

The logo for Jacada, featuring the word "Jacada" in a stylized, cursive font. The letters are colored red, yellow, and green. Below it, the word "WorkSpace" is written in a bold, black, sans-serif font.

Jacada **WorkSpace**

Administrator Guide

Version 7.3

Jacada Work Space Administrator Guide

Copyright © 2000-2016, Jacada Ltd. All rights reserved.

Notwithstanding the copyright, this documentation contains trade secrets of Jacada; use of this documentation is subject to the terms and conditions of the software license agreement between the authorized user and its licensor. No part of this publication may be reproduced, translated, stored in a retrieval system, or transmitted in any form by any means, including but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without prior written permission of Jacada. Jacada reserves the right to revise this documentation and to make changes in content from time to time without obligation on the part of Jacada to provide notification of such revision or change.

Jacada® is a registered trademark of Jacada Ltd. “Jacada WinFuse,” “Jacada HostFuse,” and “Jacada Work Space” are trademarks of Jacada Ltd. All other company and product names may be registered trademarks or trademarks of the respective companies with which they are associated.

Table of Contents

Chapter 1: Introduction	1
How This Guide is Organized.....	1
Chapter 2: Installing Jacada WorkSpace for Production: WebSphere	4
Production Requirements.....	4
Installation for Production – Overview.....	7
Installation of Jacada WorkSpace.....	9
Installation of a Clustered Production Environment.....	11
Installing the WebSphere Application Server in a Cluster.....	12
Adding Nodes and Members to a Cluster.....	13
Troubleshooting the Cluster Setup Script.....	16
Load Balancing the WebSphere Cluster.....	19
The BIG-IP Local Traffic Management System.....	19
Configuring Virtual Hosts	20
Configuring the Web Container Custom Properties for the Cluster Member.....	22
Restarting Following a Cluster Installation.....	24
Configuring the JacadaWorkSpaceCommon Application.....	26
Setting Up Licensing.....	28
Additional WebSphere Configuration Parameters for Production.....	29
Configuring the Apache Web Server.....	31

Setting Up Computer Telephony Integration.....	34
Setting Up the Agent Desktop.....	34
Installing Supplementary Files for Jacada WorkSpace.....	35
Configuring Internet Explorer for the Agent.....	35
Establishing a Dedicated HTTP Connection for Push Messaging.....	37

Chapter 3: Installing Jacada WorkSpace for Production: WebLogic 39

Production Requirements.....	39
Installation for Production - Overview.....	42
Installation of Jacada WorkSpace.....	44
Installation of a Clustered Production Environment.....	45
Configuring the WebLogic Server Cluster.....	47
Load Balancing the WebLogic Server Cluster.....	52
The BIG-IP Local Traffic Management System.....	52
Secure Communication for Load Balancer.....	54
Additional WebLogic Server Configurations.....	54
Configuring WebLogic Server to Start in Production Mode.....	55
Suspending the Generation of Diagnostic Information.....	56
Configuring the WebLogic Application Server Work Manager.....	57
Setting Up Licensing.....	58
Configuring the Apache Web Server.....	58
Configuring Redirection of Web Client Requests	59
Configuring the Apache Web Server Cache	60
Modifying the Post Timeout Setting.....	61
Configuring the Session Cookie Name	62
Adding Managed Servers to an Existing Cluster.....	62

Creating a New Cluster Member	63
Creating and Deploying a JMS Server Instance.....	65
Configuring a Topic Destination Resource	66
Creating Additional Server Domains	69
Chapter 4: Installing Jacada WorkSpace for Production:	71
Tomcat	
Production Requirements.....	71
Installation for Production - Overview.....	74
Installation of Jacada WorkSpace.....	75
Installing a Clustered Production Environment.....	76
Tomcat Cluster Configuration	77
Changing the Database Password.....	79
Load Balancing the Tomcat Cluster.....	80
Setting Up Licensing.....	81
Additional Tomcat Configuration Parameters for Production.....	81
Configuring the Apache Web Server.....	82
Chapter 5: Configuring Jacada WorkSpace in the Database	83
Database Tables.....	83
Management of Audited Entities.....	85
AUDIT_AGENT_SESSIONS Table.....	85
AUDIT_INTERACTIONS Table	86
AUDIT_PROCESSES Table.....	87
AUDIT_PROCESS_PARAMS Table.....	87
AUDIT_ACTIONS Table	88
AUDIT_ACTION_PARAMS Table.....	89
Tables of Child-Parent Relationships between Audited Events.....	89

CTI Tables.....	92
Specifying Agent CTI Information Using a Lookup Mechanism	102
Chapter 6: Configuring Authentication and Authorization	104
Authorization in Jacada WorkSpace.....	106
Configuring Settings for Authentication.....	107
Configuring Settings for LDAP Authentication.....	108
Configuring Authorization.....	109
Administering Privileges	109
Administering Roles	110
Administering Groups.....	112
Administering Skills	113
Administering Users.....	114
Chapter 7: Computer Telephony Integration (CTI)	116
Support for CTI in Jacada WorkSpace.....	116
CTI-Related Concepts.....	118
Supported Features Matrix.....	120
Configuring CTI.....	121
Configuring CTI Profiles.....	122
Configuring the Location of the CTI Gateway.....	123
Associating a CTI Provider with Each CTI Gateway.....	124
Setting Up the Call Mode of an Application.....	125
Logging Into CTI Via the Phone	125
Configuring Jacada WorkSpace to Automatically Answer Incoming Calls	125
Launching Jacada WorkSpace with a CTI Profile	126
CTI Provider-Specific Configuration.....	127

Working With Avaya Application Enablement Service (AES).....	128
Working With Avaya Elite Multichannel (EMC).....	130
Working With Avaya one_X Agent.....	142
Working With Cisco.....	143
Working With Genesys	144
Handling Loss of Connectivity to CTI.....	160
Chapter 8: Starting and Stopping the Application Server	162
Starting the Tomcat Server.....	162
Starting and Stopping the WebLogic Server.....	163
Single Server Configuration: Start and Stop Server	163
Clustered Server Configuration: Start and Stop Servers.....	165
Verify the Jacada WorkSpace Application Modules	169
Starting and Stopping the WebSphere Application Server.....	170
Clustered Server Environment: Stop and Start Servers	171
Single Server Instance: Stop and Start Server	174
Chapter 9: Deploying an Application to a Production Server	177
Deploying an Application (Tomcat).....	177
Deploying an Application (WebSphere).....	178
Deploying an Application (WebLogic).....	179
Chapter 10: Using the Jacada Management Console	183
Accessing the Jacada Management Console.....	184
Jacada Management Console User Interface.....	185
The WorkSpace Tab.....	186
Jacada WorkSpace Management Console Functions.....	188

Jacada WorkSpace Configuration.....	189
Directory Settings.....	190
Dynamic Views Settings.....	191
EMC Settings.....	192
External Script Settings.....	192
Feature Settings.....	193
Global Settings.....	194
Human Task Settings.....	198
Instant Messaging Settings.....	199
Interact Settings	200
Locales	203
Machine Settings	203
Page Mappings.....	205
RSS Feed Definitions	205
RTN Mappings	206
Script Mappings	207
Smart Pad Settings	208
Supervisor View Settings	209
Agent Disposition	209
Auditing.....	212
Authentication and Authorization.....	217
CTI.....	229
Document Search.....	246
Layout Management.....	251
Project Variables.....	269
Task Manager.....	270

Logging.....	270
Loggers	271
Log Destinations (Appenders).....	273
Monitoring Jacada WorkSpace.....	274
Chapter 11: Messaging Capabilities	276
Messaging Authorization.....	276
Message Center Messages.....	279
Creating Messages Using the Message Center.....	281
Defining Message Details	282
Defining Message Output.....	283
Editing Messages.....	285
Deleting Messages	286
Instant Messaging.....	287
Configuring the Connection to the XMPP Server.....	287
Available Tabs.....	288
Chat Session History.....	291
Chat and the Toolbar	291
Using Instant Messaging.....	293
Chapter 12: Portlet Administration	296
The Agent Tasks Portlet.....	296
Administering the Maximum Number of Tasks	297
The Document Search and Help-On-Hand Portlets.....	297
Defining Search Settings	298
The Smart Pad.....	303
Defining Smart Pad Settings.....	303

The Agent Disposition Portlet.....	304
Administration Activities.....	305
Chapter 13: Logging Capabilities	311
Producing Log Files.....	311
Jacada WorkSpace Log Configuration File.....	312
Logger Names and Appender Additivity.....	313
Log Files.....	314
Log Configuration File Settings.....	314
Logger Section.....	315
Appender Section.....	317
Supported Appenders.....	320
Session-Related Logging.....	321
The Session Dump.....	321
Nested Application Logging.....	324
WebLogic Server Logging.....	325
Chapter 14: Event Auditing Administration	329
Understanding the Audited Entities of a Project.....	330
AUDIT_AGENT_SESSIONS Table.....	330
AUDIT_INTERACTIONS Table.....	331
AUDIT_PROCESSES Table.....	332
AUDIT_PROCESS_PARAMS Table.....	332
AUDIT_ACTIONS Table.....	333
AUDIT_ACTION_PARAMS Table.....	333
AUDIT_AGENT_SESSION_PROCESSES Table.....	334
AUDIT_AGENT_SESSION_ACTIONS Table.....	335

AUDIT_INTERACTION_PROCESSES Table.....	335
AUDIT_INTERACTION_ACTIONS Table.....	336
AUDIT_PROCESS_ACTIONS Table.....	336
Set of Events for Audit.....	337
Event Auditing Administration.....	337
Aggregating Data for Reporting.....	338
Chapter 15: Side-By-Side Script Interaction	341
Administration Needed for Script Integration.....	342
Configuring the Database for Side-By-Side Script Interaction.....	342
Messages.....	348
Chapter 16: Failover and Switchover Support (WebSphere)	349
Session Replication in a Clustered Environment.....	349
Configuring a Unique Session Cookie for the Push Session.....	350
Activate Session Replication in the Application Server.....	352
Controlling the Impact of Infrastructure Failures	357
Effecting a Switchover.....	361
Chapter 17: Failover and Switchover Support (WebLogic)	362
Session Replication in a Clustered Environment.....	362
Controlling the Impact of Infrastructure Failures	363
Effecting a Switchover.....	364
Chapter 18: Troubleshooting	365
Genesys CTI.....	367
Troubleshooting Genesys CTI Connection Problems.....	367
No Access to CTI Because of a Missing Genesys License.....	369

Troubleshooting Errors in CTI Log Files.....	370
Information Required for Genesys CTI Troubleshooting.....	372

Introduction

Jacada WorkSpace enables you to create an Agent WorkSpace: the unified desktop for agents, providing a single point of access to all the applications and tools required by the agent. This guide describes how to manage an Agent WorkSpace in a production environment.

The guide is intended for administrators deploying Jacada WorkSpace to call center agent desktops. Administrators should have knowledge of:

- The Apache Tomcat server
- The Oracle WebLogic application server
- The IBM WebSphere application server, and specifically the IBM WebSphere Integrated Solutions Console

Note: If you are planning to use [Avaya EMC as your CTI provider](#), install Jacada WorkSpace using Tomcat and create the server and database using the [Server Setup Wizard](#).

How This Guide is Organized

This guide is organized as follows:

	Chapter Title	Description
2	Installing Jacada WorkSpace for Production: Websphere	How to install Jacada WorkSpace for production, both in a single server instance environment and in a clustered environment.

	Chapter Title	Description
3	Installing Jacada WorkSpace for Production: WebLogic	How to install Jacada WorkSpace for production, both in a single server instance environment and in a clustered environment.
4	Installing Jacada WorkSpace for Production: Tomcat	How to install Jacada WorkSpace for production, both in a single server instance environment and in a clustered environment.
5	Configuring Jacada WorkSpace in the Database	The configuration and settings for Jacada WorkSpace in the database.
6	Configuring Authentication and Authorization	Jacada WorkSpace support for configuring user authentication and user authorization. Authentication of users either permits or denies users' access to Jacada WorkSpace. Authorization of user assigned privileges controls both agent and supervisor use of Jacada WorkSpace features and user interface components.
7	Computer Telephony Integration (CTI)	An introduction to how Computer Telephony Integration is implemented in Jacada WorkSpace.
8	Starting and Stopping the Application Server	How to start and stop a Tomcat server/WebSphere application server.
9	Deploying an Application	How to deploy a Jacada WorkSpace application to the production server.
10	Using the Jacada Management Console	The Jacada Management Console, which is used to manage the Jacada WorkSpace runtime environment.
11	Messaging Capabilities	Messaging capabilities for the supervisor available in Jacada WorkSpace.

	Chapter Title	Description
12	Portlet Administration	Specific configuration required for portlets that require setting up by the administrator.
13	Logging Capabilities	Jacada WorkSpace logging configuration options.
14	Event Auditing Administration	How to manage auditing and nightly reports.
15	Side-By-Side Script Interaction	Integrating Jacada Interaction Manager Scripts in Jacada WorkSpace.
16	Failover and Switchover Support (WebSphere)	How to configure the environment for failover.
17	Failover and Switchover Support (WebLogic)	How to configure the environment for failover.
18	Troubleshooting	How to resolve some common problems that can be encountered.

2

Installing Jacada WorkSpace for Production: WebSphere

The following topics describe how to install Jacada WorkSpace in a IBM WebSphere Application Server production environment, in order to deploy an Agent WorkSpace:

- [Production Requirements](#)
- [Installation for Production – Overview](#)
- [Installation of Jacada WorkSpace](#)
- [Installation of a Clustered Production Environment](#)
- [Load Balancing the WebSphere Cluster](#)
- [Restarting Following a Cluster Installation](#)
- [Configuring the JacadaWorkSpaceCommon Application](#)
- [Setting Up Licensing](#)
- [Additional WebSphere Configuration Parameters for Production](#)
- [Configuring the Apache Web Server](#)
- [Setting Up Computer Telephony Integration](#)
- [Setting Up the Agent Desktop](#)

Production Requirements

This section describes the requirements for a Jacada WorkSpace production environment.

Agent Desktop Production Requirements

Jacada WorkSpace is certified with the following configuration for the call center agent desktop:

Certified Configuration

Operating System	Microsoft Windows XP Professional with Service Pack 3 Microsoft Windows 7 64 bit, Enterprise edition
Display Resolution	1280x1024 (minimum) 1366x768 and above (recommended)
Browser	• Microsoft Internet Explorer version 9 (32 bit and 64 bit) • Microsoft Internet Explorer version 10 (32 bit and 64 bit) • Microsoft Internet Explorer version 11 (32 bit and 64 bit) • Google Chrome (without hosting)

Windows Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Windows server:

Certified Configuration for Windows

Minimum Hardware	CPU	Intel Xeon 2 x 2.4GHz
	RAM	2 GB (4 GB recommended)
Software	Operating System*	Windows 2012 Server R2 64 bit
	Application Server	• IBM WebSphere Application Server version 8.5.5 (64 bit) and IBM WebSphere Application Server - Express (32 bit) • PM90932 - http://www-01.ibm.com/support/docview.wss?uid=swg24035220
	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0 • Genesys Telecommunications Laboratories Inc. – Genesys 8.0

Certified Configuration for Windows

		and 8.1 Both vendors are supported with the following switch: Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	- Oracle 11g R2 - SQL Server 2008 - MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.10 (32 bit)

*** You can work with Windows under VMware ESX.**

Linux RedHat Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Linux RedHat server:

Certified Configuration for Linux RedHat

Minimum Hardware	CPU	2 x Xeon E5420 (2.5 GHz) Quad Core
	RAM	2 GB (4 GB recommended)
Software	Operating System	Linux RedHat Enterprise 6.6 (64 bit)
	Application Server	IBM WebSphere Application Server version 8.5.5 (64 bit) and IBM WebSphere Application Server - Express (32 bit)

Certified Configuration for Linux RedHat

	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0 • Genesys Telecommunications Laboratories Inc. – Genesys 8.0 and 8.1 Both vendors are supported with the following switch: • Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	• Oracle 11g R2 • SQL Server 2008 • MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.15

Installation for Production – Overview

Jacada WorkSpace projects are deployed to the IBM WebSphere Application Server. The supported application server configurations for a production environment are, as follows:

- A clustered server environment: Used for large-scale production environments. The cluster server environment provides failover capabilities.
- A single server instance: Applicable to small-scale production environments, product testing and product staging/controlled roll-out.

Note: This guide assumes that you are familiar with the IBM Installation Manager, the IBM WebSphere-Update Installer for WebSphere v8.5.5, the IBM Profile Management Tool and the IBM WebSphere Integrated Solutions Console.

Installation Sequence

Jacada requires the following installation sequence:

Server Environment	Installation Sequence	Notes
WebSphere Cluster	- WebSphere Application Server v8.5.5. - WebSphere Application Server maintenance pack updates. - Jacada WorkSpace 7.3 without running the Jacada Server Setup wizard.	Windows Host Machine: Use WebSphere Application Server v8.5.5-Network Deployment.
		Linux RedHat Host Machine: Use WebSphere Application Server Network Deployment V8.5.5 for Linux on 64 bit.
WebSphere Single Server	- WebSphere Application Server v8.5.5. - WebSphere Application Server maintenance pack updates. - Jacada WorkSpace 7.3, including running the Jacada Server Setup wizard.	Windows Host Machine: Use WebSphere Application Server v8.5.5-Base.
		Linux RedHat Host Machine: Use WebSphere Application Server V8.5.5 for Linux on x64 bit.

Installation of Jacada WorkSpace

The Jacada WorkSpace installation includes the following activities:

- Installation of Jacada WorkSpace.
- Creation of a WebSphere Application server profile. A profile provides the environment for running a particular WebSphere server. The process also configures the database schema to be used with the WebSphere server.

If your production environment is a WebSphere cluster configuration, install only the Jacada WorkSpace product files. If your production environment is a WebSphere single server, run the server profile setup too.

Note: Before beginning the installation procedure, verify that the name of your computer does not contain an underscore character.

To install Jacada WorkSpace:

1. In the folder containing the Jacada WorkSpace installation, do one of the following:
 - If the production server is going to run under the Linux RedHat operating system, execute **/setup.bin -i console**.
 - If the production server is going to run under one of the Windows operating systems, run the **setup.exe** file.

2. Follows the installation instructions until the *Setup a Server* window displays.

When installing on Linux RedHat, this window is skipped.

When installing on a Windows operating system, do one of the following:

- If your production environment is a WebSphere cluster configuration, select the *No* option. The Jacada WorkSpace product file installation completes.
- If your production environment is a WebSphere single server, select the *Yes* option and click *Next*.

After a brief processing period, the *Server Setup Wizard Introduction* window displays.

To both create a new WebSphere profile and define the database details of the database that your WebSphere Application Server will use, continue with the next procedure: [To run the server profile setup](#).

3. When installing on a Linux RedHat operating system, when the *Install Complete* window displays, click *Done*.

- If your production environment is a WebSphere cluster configuration, take no further action.
- If your production environment is a WebSphere single server, to both create a new WebSphere profile and define the database details of the database that your WebSphere Application Server will use, continue with the next procedure: [To run the server profile setup](#).

To run the server profile setup:

1. Access the Server Setup Wizard:
 - **Linux RedHat operating system:** Run `./ServerSetupWizard.sh -i console`, located in the directory `<Jacada_installation_directory>/Jacada WorkSpace <version>/Common/Utilities/ServerSetup`.
 - **Windows operating system:** Run *Start > Programs > Jacada > Jacada WorkSpace <version> > Server Setup Wizard*.
2. In the *Select Application Server* dialog, select *IBM WebSphere 8.5.5*.
3. In the *Select Application Server Home Folder* dialog, select the WebSphere server home folder to use with Jacada WorkSpace.
4. In the *WebSphere Profile Details* dialog, if the WebSphere server will not reside on the local machine, then select a *Host Address*. Make sure that the option *Use Recommended Ports* is checked.
5. In the *Administrator User Details* dialog, specify the user name and password that will be used to log on to the IBM WebSphere Integrated Solutions Console for the profile being defined.
6. In the *Profile Details* dialog, specify the name you want for the profile.
7. In the *Database Details* dialog, select the required database from the drop-down list and dependent on the database, enter the other details.
8. In the *Schema Details* dialog, specify the name and password you want for the profile. The default schema name is the same as the profile name.
 After a processing period of several minutes, the new WebSphere profile is created. The *Setup Summary* window displays. A WebSphere server profile is created and a server configured that connects to a created database schema.
9. If you had a profile defined for a previous version of Jacada WorkSpace, in the WebSphere runtime `lib/ext` directory, remove the following `.jar` files, which were installed as global files when a profile was defined for the previous version:
 - `ConfigurationApi-<version>.jar`
 - `Logging-<version>.jar`

- ManagementUtilsInterface-<version>.jar
- MetadataInterface-<version>.jar
- SessionDumpStorage-<version>.jar
- SessionManagerInterface-<version>.jar
- WorkspaceSessionManagerInterface-<version>.jar

Installation of a Clustered Production Environment

In a production environment, Jacada WorkSpace is typically deployed in a cluster of application servers. A cluster is a deployment in which multiple IBM WebSphere Application Server instances are used to process requests. From the client's point of view, there is only one instance of WebSphere. Clusters provide efficient application administration through update synchronization across application servers. A WebSphere cluster is also referred to as a federated, WebSphere network deployment.

A WebSphere cluster is comprised of a deployment manager server, one or more managed node servers and one or more application servers, called cluster members, associated with each managed node. The managed node servers can reside on the same machine as the deployment manager server or on different machines. Jacada WorkSpace projects are deployed to the application servers of the cluster. For more information on WebSphere clustering, refer to the IBM WebSphere documentation.

To run Jacada WorkSpace projects in a clustered, WebSphere production environment, install and configure a WebSphere network deployment using the IBM WebSphere Application Server Network Deployment Install Wizard.

The following sections describe:

- [Installing the WebSphere Application Server in a Cluster](#)
- [Adding Nodes and Members to a Cluster](#)
- [Troubleshooting the Cluster Setup Script](#)

Installing the WebSphere Application Server in a Cluster

On the host machine, install WebSphere as a network deployment. By default, the initial installation creates a cell environment that includes the following network deployment elements:

- A deployment manager server and its server profile.
- A managed node, called the node agent, and its server profile.

Note: Each profile provides the environment for running a particular server.

- The node agent application server that is federated to the cell of the deployment manager.

Jacada provides a cluster property configuration file. The file contains property definitions that the Jacada cluster setup script uses to create the WebSphere cluster. Review the contents of the **<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/Utilities/ClusterSetup/WebSphere/JadCluster.properties** file. Update the property definitions to support your WebSphere cluster installation for Jacada WorkSpace.

Note: Make sure to change the *db.host* property in *JadCluster.properties* from *localhost* to the IP address of the machine hosting the database.

At this point, you are ready to transform the existing WebSphere cell into a WebSphere cluster.

To create the WebSphere cluster:

1. Open a command window and from the folder **<WebSphere_app_server_home_folder>/profiles/<deployment_manager_profile_name>/bin**, run the following command to determine the current status of the deployment manager server:

```
serverStatus dmgr
```
2. If the deployment manager server, *dmgr*, is stopped, then from the same folder, run the following command to start it:

```
startManager
```
3. From the same folder, run the following command to create the cluster:
For windows:

```
ws_ant.bat -f "<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/  
Utilities/ClusterSetup/WebSphere/SetupJacadaCluster.xml"
```

For linux:

```
./ws_ant.sh -f "<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/  
Utilities/ClusterSetup/WebSphere/SetupJacadaCluster.xml"
```

This command runs the Jacada cluster setup script that performs several main tasks using the definitions provided by the `JadCluster.properties` file. If the script does not successfully complete, separately run each cluster setup task, refer to [Troubleshooting the Cluster Setup Script](#).

Note: If the script hangs during the configuration of security, type `y` to continue.

On the host machine, the database schema, the cluster and one cluster member application server are created, the Jacada WorkSpace core applications are installed in the cluster member and the Jacada WorkSpace provided users and groups are configured.

Note: If additional WebSphere fixpacks were installed after installing the Jacada WebSphere profile, the profile needs to be augmented with JPA2.0 features. For more information, go to:
http://pic.dhe.ibm.com/infocenter/dmndhelp/v7r0mx/index.jsp?topic=%2Fcom.ibm.wdpe.install.server.doc%2Fprofiles%2Ft_aug_profiles.html.

Adding Nodes and Members to a Cluster

Note: The following section is only relevant for IBM Websphere 32 bit. For 64 bit installation, refer to the IBM documentation on how to add nodes to a cluster.

For the Jacada WorkSpace production environment, expand the IBM WebSphere Application Server cluster to include additional managed nodes and cluster members associated with each managed node. Managed nodes can reside either on the same machine as the deployment manager server

and cluster or on a remote machine.

To federate a new node to the deployment manager:

1. On the machine hosting the managed node, access the Profile Management wizard:
 - **Windows:** Run *Start > Programs > IBM WebSphere > Application Server Network Deployment v8.5.5 > Profile Management Tool*.
 - **Linux:** *./install_root/bin/ProfileManagement/pmt.sh*

The Profile Management wizard appears. Use the wizard to create additional profiles.

2. In the *Environment Selection* window, select *Custom Profile* from the list of environments to create.
3. In the *Profile Creation Options* window, select the *Advanced profile creation* option.
Choosing this option enables you to use specify your own configuration values for the profile that you are creating. During creation, the wizard always presents any default configuration values for the profile.
4. In the *Profile Name and Location* window, enter the *Profile name* and the *Profile directory* where this profile is located.
5. In the *Node and Host Names* window, enter the *Node Name* of the managed node associated with this profile and the *Host Name* of the machine where the managed node is located.
6. In the *Federation* window, enter the either the *name* or the *IP address* of the machine hosting the deployment manager, enter the deployment manager's *SOAP port number* and enter the *administrator user name* and *password* for the deployment manager.
Federate the profile's node as part of this procedure.
7. Review the information presented in the *Port Values Assignment* window. As necessary, update any configuration values.
8. Review the information presented in the *Profile Creation Summary* window. Click *Create*.

The profile creation process is initiated and the profile's node is federated to the cell of the identified deployment manager.

When notified that the profile is successfully created, uncheck the *Launch the First Steps window* option and click *Finish*.

After creating the profile and federated node, do the following:

- From an installation of Jacada WorkSpace, copy the entire contents of the folder **<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/Utilities/ClusterSetup/Common/Root** to the **<WebSphere_app_server_home_folder>/profiles/<new_node_profile_name>** folder.
- From an installation of Jacada WorkSpace, copy the file **<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/Utilities/ServerSetup/Common/Root/ehcache.xml** to the **<WebSphere_app_server_home_folder>/profiles/<new_node_profile_name>** folder.

If the new, federated node and its deployment manager do **not** reside on the same machine, do the following:

- From an installation of Jacada WorkSpace, copy the entire contents of the folder **<Jacada_installation_folder>/Jacada WorkSpace 7.3/Common/Utilities/ServerSetup/Common/Lib** to the **<WebSphere_app_server_home_folder>/lib/ext** folder on the machine hosting the managed node.
- If your WebSphere cluster is using either the IBM DB2 or the Oracle database, then from an installation of Jacada WorkSpace, copy the entire contents of the folder **<Jacada_installation_folder>/Jacada WorkSpace 7.3/WorkSpace/3rdparty/<database_name>/lib** to the **<WebSphere_app_server_home_folder>/lib/ext** folder on the machine hosting the managed node.

To start the new federated node:

1. On the machine hosting the managed node, open a command window and from the folder **<WebSphere_app_server_home_folder>/profiles/<node_profile_name>/bin**, run the following command to determine the current status of the node server:

```
serverStatus nodeagent
```

2. If the node server, *nodeagent*, is stopped, then from the same folder, run the following command to start it:

```
startNode
```

The nodeagent server of the new node is started.

To add a new cluster member to the federated node:

Note: Performing this procedure requires administrator level user privilege.

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*. The *Server clusters* page displays.

2. In the table listing the existing server clusters, click the relevant `<cluster_name>` link.
The *Configuration* tab of the `<cluster_name>` displays.
3. In the *Additional Properties* list, click the *Cluster members* link.
The table of existing cluster members displays.
4. In the table, click the *New* button.
The *Create new cluster members* page displays. Use this page to add application servers to the cluster.
5. Enter the *Member name*.
6. From the dropdown list, select the node to which you want to add the new cluster member.
7. Click the *Add Member* button.
The new cluster member is added to the table listing of cluster members
8. Click *Next*.
A table, summarizing the actions taken, displays in the *Create new cluster members* page.
9. Click *Finish*.
The *Server clusters* page displays with the updated table of existing cluster members.
10. In the *Messages* box, click the *Review* link and do the following:
 - a. Expand the *Total changed documents* entry to review the list of changed items.
 - a. Click *Save*.

The system updates the master repository configuration with your current changes, automatic synchronization mode is enabled for the new federated node and configuration synchronization completes for the cell, which includes the cluster and all its nodes.

Troubleshooting the Cluster Setup Script

Jacada provides a script to configure the application server profile with Jacada specific configurations and create a cluster with a single member as a template for all cluster members. The script consists of tasks that run sequentially. If the script fails to complete, determine the name of the specific task that failed from the script output, and follow the guidelines in the table.

Troubleshooting Table

Task	Responsibility	Possible Failure	Guideline
test.cluster.properties	Validates the properties configured in the <code>JadCluster.properties</code> file.	One of the properties in the <code>JadCluster.properties</code> file is not configured correctly.	Use the script failure message to change the relevant property and rerun the script.
common.setup.profile	Copies configuration files to the root of the node profile and all external jars to the Classpath of the application server.	No permission to copy files to the application server folders.	Verify that you have write permissions to the relevant directories and rerun the script.
jad.configure.db	Creates the database schema and loads default values.	Either the database is not running or a schema with the given name is already defined in the database.	Check the connection to the database and if necessary change database related properties in the <code>JadCluster.properties</code> file and rerun the script.
jad.setup.profile	Creates the WebSphere cluster with a single cluster member and configures the data source and JMS connections.	One of the configurations objects that this task defines is already defined in the profile.	Find the conflicting configuration and delete it manually from the console. Rerun this task and then the following tasks only.

Task	Responsibility	Possible Failure	Guideline
jad.install.applications	Deploys the JacadaWorkSpaceCommon, JacadaSessionManager, CTIGateway, dynamicviews and JMC files in the cluster.	One of the applications is already deployed in the cluster or the jad.setup.profile task did not complete successfully. This means that the target cluster may not have been configured.	Verify that the jad.setup.profile task completed successfully. If it completed successfully, deploy the EAR files manually.
jad.setup.profile.secured	Configures the Jacada WorkSpace default users and groups	One of the default users, <i>Workspace</i> or <i>Guest</i> one of the groups is already configured.	If the users and groups are already defined in your profile, skip this task, otherwise delete the conflicting configurations and rerun this task.

To run a specific task in the cluster setup script:

- Run the SetupJacadaCluster.xml specifying the task to run, as follows:

For windows:

```
ws_ant.bat -f "<Jacada_installation_dir>/JacadaWorkSpace7.3/Common/Utilities/ClusterSetup/WebSphere/SetupJacadaCluster.xml" <task>
```

where **<task>** is one of the tasks listed in the table.

For linux:

```
./ws_ant.sh -f "<Jacada_installation_dir>/JacadaWorkSpace7.3/Common/Utilities/ClusterSetup/WebSphere/SetupJacadaCluster.xml" <task>
```

where **<task>** is one of the tasks listed in the table.

Load Balancing the WebSphere Cluster

In a typical cluster configuration, a load balancer is configured as a front-end to the WebSphere Application Server cluster. For the call center agent client machines, the load balancer is the single access point to the cluster and the request distributor to the WebSphere cluster members.

The following sections describe:

- [The BIG_IP Local Traffic Management System](#)
- [Configuring Virtual Hosts](#)
- [Configuring Web Container Cluster Properties](#)

The BIG-IP Local Traffic Management System

Jacada recommends deploying the F5 Network BIG-IP Local Traffic Management (LTM) system for directing traffic to the WebSphere cluster members running Jacada WorkSpace. A BIG-IP LTM configuration consists of defining the following components:

- A virtual server (the traffic manager)
- A connection pool (a set of devices grouped together to receive traffic according to a load balancing method) associated with a virtual server
- The connection pool members (the WebSphere cluster members)

For BIG-IP LTM configuration information and procedures, see the BIG-IP guide, *Deploying the BIG-IP LTM System with IBM WebSphere Servers*.

In order for Jacada WorkSpace traffic to properly travel between the agent client machines, the LTM virtual server and the WebSphere cluster members, configure the following properties and resources in the BIG-IP LTM Configuration utility:

Default Persistence Profile – Set this resource to `cookie`. The LTM virtual server uses cookies to assign each agent client machine to a specific WebSphere cluster member in the connection pool. This setting instructs the virtual server to always direct requests from an agent client machine to the same WebSphere cluster member running Jacada WorkSpace. Access the resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.

HTTP Profile – Set this property to `http`. Access the property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

Protocol Profile (Client) and Protocol Profile (Server) – Set both these properties to the custom tcp profile of your choice. The assigned tcp profile must have its *Nagle's Algorithm* setting unchecked. By disabling this property, you prevent unnecessary TCP/IP delay of small packets during their routing. Access these properties in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

SSL Profile (Client) – Set this property to `clientssl`. This setting instructs the virtual server that communication with any agent client machines is always encrypted. Access this property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

Health Monitors – For the *Default Pool* resource of the LTM virtual server, assign as active a monitor of type TCP to its *Health Monitors* property. The assigned monitor must have its *Interval* configuration property set to 30 seconds and its *Timeout* configuration property set to 91 seconds. These resource and property configurations cause the LTM to monitor the TCP connection status of its application server pool members, when load balancing new connection requests to the cluster of servers.

- Access the *Default Pool* resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.
- Access the *Health Monitors* property in the *Main Tab > Local Traffic > Pools > Pool List > <pool_name> > Properties > Configuration > Basic* properties.
- Access the *TCP monitor Interval and Timeout* properties in the *Main Tab > Local Traffic > Monitors > Monitor List > <monitor_name> > Properties > Configuration > Basic* properties.

In a clustered, production environment that uses the F5 Network BIG-IP LTM system to provide load balancing, Jacada WorkSpace supports the failover mechanism. For a description of the various capabilities and settings required to implement failover for the Jacada WorkSpace production environment, refer to [Failover and Switchover Support](#).

Configuring Virtual Hosts

WebSphere supports establishing one or more virtual hosts. A configuration of virtual hosts enables an individual host machine to resemble multiple host machines. In other words, virtual hosts enable you to isolate, and independently manage, multiple sets of resources on the same physical machine. Each virtual host is assigned its logical name, for example, *default_host* or *admin_host*, and a set of domain name system (DNS) aliases by which the virtual host is known. The alias comprises the first part of the path for accessing a resource.

When your configuration includes a load balancer directing client machine requests to multiple WebSphere cluster members, you must add the load balancer's *port* and each cluster member's *web container default host* port as host alias entries for *default_host* virtual host.

To determine the port of the BIG-IP LTM virtual server:

1. In the *Main* tab, click the *Local Traffic* menu option, and then click the *Virtual Servers* menu option.
The *Virtual Servers* page displays.
2. From the Virtual Server list, click the *name* link of the virtual server you want.
The *Properties* page of the selected virtual server displays.
3. In the *General Properties* pane, record the value in the *Service Port* property field.
You now have the port of the BIG-IP LTM virtual server.

To determine a cluster member's Web Container Default Host port:

Note: Performing this procedure requires administrator level user privilege.

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*.
The *Server clusters* page displays.
 2. In the table listing the existing server clusters, click the relevant `<cluster_name>` link.
The *Configuration* tab of the `<cluster_name>` displays.
 3. In the *Additional Properties* list, click the *Cluster members* link.
The table of existing cluster members displays.
 4. In the cluster members table, click the relevant `<member_name>` link.
The *Configuration* tab of the `<member_name>` displays.
 5. In the *Communications* list, click the *Ports* link.
The table listing the TCP/IP ports that the cluster member uses displays.
 6. In the table, locate the port named `WC_defaulthost` and record its assigned port.
You now have the cluster member's *web container default host* port.
 7. Repeat Step 4 through Step 6 for each additional WebSphere cluster member.
- When you have obtained the port for the load balancer and the *web container default host* port of each cluster member, add them as host alias entries for *default_host* virtual host.

To add a port to the Host Alias table:

Note: Performing this procedure requires administrator level user privilege.

1. In the task menu of the WebSphere Integrated Solutions Console, select *Environment > Virtual Hosts*.

The *Virtual Hosts* page displays.

2. In the table listing the virtual hosts, click the `default_host` link.

The *Configuration* tab of the *default_host* virtual host displays.

3. In the *Additional Properties* list, click the *Host Aliases* link.

The table of defined host aliases for the *default_host* virtual host displays.

4. Click *New*.

The *Configuration* tab for a host alias displays.

5. In the *Port* field, enter the port number of either the load balancer or the *web container default host* port of a cluster member.

6. Click *Apply* and, if the information is valid and complete, click *OK*.

The updated table of defined host aliases for the *default_host* virtual host displays and contains the new host alias.

7. Repeat Step 4 through Step 6 to define each additional port.

8. In the *Messages* box, click the *Review* link and expand the *Total changed documents* entry to review the list of changed items and then click *Save*.

The master repository configuration is updated with the host aliases for the *default_host* virtual host, which includes the load balancer and *web container default host* ports of each cluster member.

Configuring the Web Container Custom Properties for the Cluster Member

Each WebSphere cluster member that has its *web container default host* port added as a host alias for the *default_host* virtual host, must have specific web container custom properties defined. A custom property specifies an arbitrary name-value pair. The value is a string that can set internal system configuration properties. The following custom properties must be configured:

- **httpsIndicatorHeader:** Set this custom property to `host`. Whenever the WebSphere cluster member identifies the value `host` in the HTTP header of a processed request and request redirection is required, the WebSphere cluster member redirects the request using an HTTPS scheme.
- **trustHostHeaderport:** Set this property to `true`. This setting causes the web container to first return the port number from the request host header and only secondarily, return the URL port that accepted the client connection.

To configure the Custom Properties for a cluster member:

Note: Performing this procedure requires administrator level user privilege.

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Application servers*.
The *Application servers* page displays.
2. In the table listing the existing application servers, click the `name` link of the relevant cluster member.
The *Configuration* tab of the cluster member `<name>` displays.
3. In the *Container Settings* list, click the *Web Container Settings* link.
Links to the available settings display.
4. Click the *Web container* link.
The Web container *Configuration* tab displays.
5. In the *Additional Properties* list, click the *Custom Properties* link.
The table of existing custom properties displays.
6. In the table, click the *New* button.
The *Configuration* tab of a new custom property displays. Use this tab to specify a name-value pair.
7. In the *Name* field, enter the custom property name.
8. In the *Value* field, enter the necessary property setting.
9. In the optional *Description* field, enter a relevant description for the custom property.
10. Click *Apply* and, if the information is valid and complete, click *OK*.
The updated table of defined custom properties displays and contains the new custom property.
11. Repeat Step 6 through Step 10 to define additional custom properties.

12. In the *Messages* box, click the *Review* link and do the following:
 - a. Expand the *Total changed documents* entry to review the list of changed items.
 - b. Click *Save*.

The master repository configuration is updated with the all the web container custom properties defined for the *default_host* virtual host per the selected WebSphere cluster member. When multiple WebSphere cluster members exist, repeat this procedure as necessary.

Restarting Following a Cluster Installation

Following completion of the WebSphere cluster installation for your Jacada WorkSpace production environment, of your WebSphere, stop and then restart all the created WebSphere network deployment elements, specifically the deployment manger, the managed nodes, the cluster and its cluster members. Performing this stop and restart ensures that all updates are appropriately applied across the WebSphere network deployment.

Remember that in a WebSphere network deployment, managed nodes can reside either on the same machine as the deployment manager server and cluster or on a remote machine.

To stop the WebSphere network deployment elements:

Note: Performing this procedure requires administrator level user privilege.

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*.
The *Servers clusters* page displays.
2. In the table listing the existing clusters, check the relevant cluster and then click *Stop*.
The selected cluster and its cluster members (application servers) stop running.
3. Open a command window.
4. Change your current directory to the profile bin folder of the managed node you want to stop, `<WebSphere_app_server_home_folder>/profiles/<node_profile_name>/bin`. For example, for the managed node whose WebSphere profile is named AppSrv01, change to:
`C:/Program Files/IBM/WebSphere/AppServer/profiles/AppSrv01/bin`
5. Run the following command:
`stopNode`
The server *nodeagent* is stopped for the managed node.
6. Repeat Step 4 through Step 5 to stop the additional managed nodes installed for this

WebSphere network deployment.

7. Change your current directory to the profile bin folder of the deployment manger you want to stop, `<WebSphere_app_server_home_folder>/profiles/<deployment_manager_profile_name>/bin`. For example, for the deployment manger whose WebSphere profile is named Dmgr01, change to:

```
C:/Program Files/IBM/WebSphere/AppServer/profiles/Dmgr01/bin
```

8. Run the following command:

```
stopManager
```

The deployment manger server *dmgr* is stopped.

At this point, all the installed WebSphere network deployment elements are stopped. Proceed with restarting these elements.

To restart the WebSphere network deployment elements:

1. Open a command window.
2. Change your current directory to the profile bin folder of the deployment manger you want to start, `<WebSphere_app_server_home_folder>/profiles/<deployment_manager_profile_name>/bin`. For example, for the deployment manger whose WebSphere profile is named Dmgr01, change to:

```
C:/Program Files/IBM/WebSphere/AppServer/profiles/Dmgr01/bin
```

3. Run the following command:

```
startManager
```

In the command window, the following system output messages are generated:

```
Starting tool with the <deployment_manager_profile_name> profile
Reading configuration for server: dmgr
Server launched. Waiting for initialization status.
Server dmgr open for e-business; process id is <process_id>
```

The deployment manger server *dmgr* is started.

4. Change your current directory to the profile bin folder of the managed node you want to start, `<WebSphere_app_server_home_folder>/profiles/<node_profile_name>/bin`. For example, for the managed node whose WebSphere profile is named AppSrv01, change to:

```
C:/Program Files/IBM/WebSphere/AppServer/profiles/AppSrv01/bin
```

5. Run the following command:

startNode

In the command window, the following system output messages are generated:

```
Starting tool with the <node_profile_name> profile
Reading configuration for server: nodeagent
Server launched. Waiting for initialization status.
Server nodeagent open for e-business; process id is <process_id>
```

The server *nodeagent* is started for the managed node.

6. Repeat Step 4 through Step 5 to start the additional managed nodes installed for this WebSphere network deployment.

Note: Performing the remaining steps requires administrator level user privilege.

7. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*. The *Servers clusters* page displays.
 8. In the table listing the existing clusters, check the relevant cluster and then click *Start*. The selected cluster and its cluster members (application servers) start running. It will require several minutes for the cluster members to change their status to started
- At this point, all the installed WebSphere network deployment elements are started.

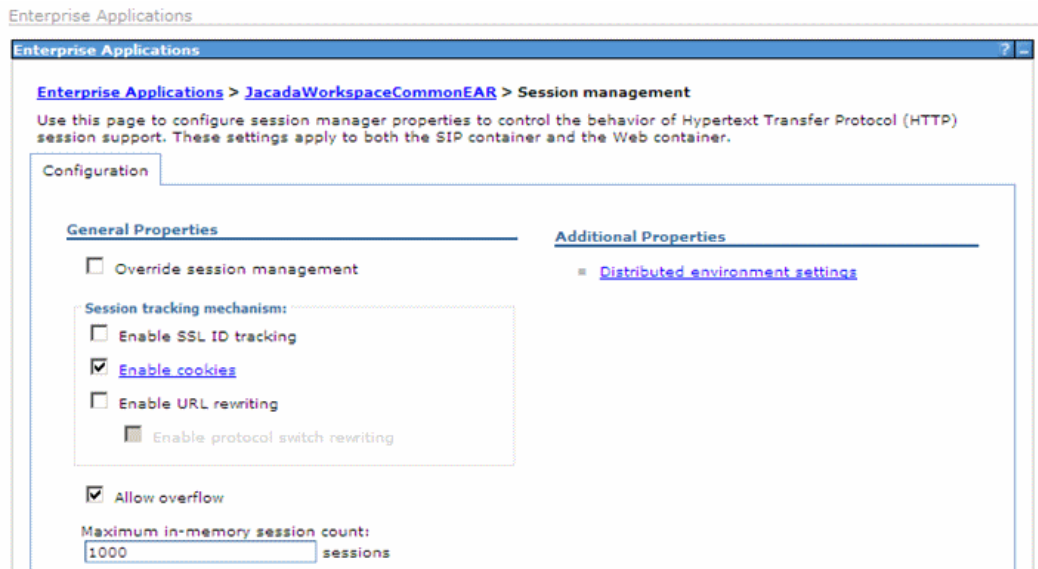
Configuring the JacadaWorkSpaceCommon Application

Following the installation of Jacada WorkSpace in a WebSphere Application Server environment, either cluster or single server instance, enable the Override Session Management option of the deployed application JacadaWorkSpaceCommonEAR and configure its use of the *PushSessionId* cookie. Perform these actions in the application server, prior to JacadaWorkSpaceCommonEAR running.

Note: After any redeployment of the JacadaWorkspaceCommon.EAR file to the WebSphere application server, make sure to re-perform these actions in the application server for JacadaWorkspaceCommonEAR.

To configure JacadaWorkspaceCommon to use the PushSessionId cookie:

1. In the task menu of the WebSphere Integrated Solutions Console, select *Applications > Enterprise Applications > JacadaWorkspaceCommonEAR > Session Management*. The *Configuration* tab for the *Session Management* page displays.



2. In the *General Properties* list, check the *Override session management* option.
3. Click *Apply*.
The JacadaWorkspaceCommon application has its *Override session management* option enabled.
4. In the *Session tracking mechanism* area, click the *Enable cookies* link.
The *Configuration* tab for *Cookies* page displays.
5. In the *Cookie name* field, replace the current field entry, *JSessionId*, with *PushSessionId*.

[Enterprise Applications](#) > [JacadaWorkspaceCommonEAR](#) > [Session management](#) > **Cookies**

Use this page to specify cookie settings for Hypertext Transfer Protocol (HTTP) session management.

Configuration

General Properties

Cookie name

☐ Restrict cookies to HTTPS sessions

Cookie domain

Cookie path

Cookie maximum age

☒ Current browser session
☐ Set maximum age seconds

6. Click *Apply*.

The JacadaWorkSpaceCommon application is configured to perform its push processing using the session assigned by the *PushSessionId* cookie.

7. Click *Save*.

Setting Up Licensing

Obtain a Jacada WorkSpace license file, `license.jacada`, from your Jacada representative. In order to correctly establish your Jacada WorkSpace licensing, do one of the following:

- For a cluster configuration, place the license file in the profile folder of each managed node in the cluster, `<WebSphere_app_server_home_folder>/profiles/<node_profile_name>`.
- For a single server instance, place the license file in the profile folder of the server, `<WebSphere_app_server_home_folder>/profiles/<server_profile_name>`.

Additional WebSphere Configuration Parameters for Production

The following sections explain parameters that need customized tuning for the production environment.

Heap Size

Initial and maximum heap size values need to be configured for each member in the cluster. The values are defined in the WebSphere Admin Console, on the page of the cluster member, from the *Server Infrastructure* section. The path to the parameters is:

Member<num> >Java and Process Management > Process Definition >Java Virtual Machine

Configure the memory according to your environment.

Generic JVM Arguments

To tune the LDAP connection pool, generic JVM arguments need to be configured for each member in the cluster. The parameter is defined in the WebSphere Admin Console, on the page of the cluster member, from the *Server Infrastructure* section. The path to the parameter is:

Member <num> >Java and Process Management -> Process Definition >Java Virtual Machine

Configure the connect pool for LDAP. For example:

```
-Dcom.sun.jndi.ldap.connect.pool.maxsize=<num>-Dcom.sun.jndi.ldap.connect.pool.prefsize=
<num> -Dcom.sun.jndi.ldap.connect.pool.timeout=<num>
```

Number of Threads

The minimum and maximum size of thread pools in the web container need to be configured for each member in the cluster. The values are defined in the WebSphere Admin Console, on the page of the cluster member, from the *Additional Properties* section. The path to the parameters is:

Member<num> > Thread Pools > WebContainer > Minimum Size / Maximum Size.

In addition, it is recommended to check the *Allow thread allocation beyond maximum thread size* checkbox.

Web Container Transport Chains

Web container transport chains need to be configured for each member in the cluster. The values

are defined in the WebSphere Admin Console, on the page of the cluster member, from the *Persistent connections* section. The path to the parameter is:

Member<num> > Web container transport chains > WCInboundDefault > HTTP inbound channel (HTTP_2).

Under *Container settings*, click the *Web container settings* link. Then, in the *Persistent connections* frame, select the *Unlimited persistent requests per connection* radio button.

Size of JMS Connection Factory Pool

Initial and maximum size values need to be configured for the JMS connection factory pool. The values are defined in the WebSphere Admin Console, on the *Topic connection factories* page. The path to the parameters is:

Resources > JMS> Process Definition >Topic Connection Factories

In the *Preferences* frame, click the *XAConnectionFactory* link. Then, on the right side of the *Topic connection factories* page, click the *Connection pool properties* link.

Configure the *Minimum connections* and *Maximum connections* values, according to your environment.

Size of JDBC Pools

Minimum and maximum size values need to be configured for JDBC pools, for both Jacada WorkSpace and Jacada Dynamic Views. The values are defined in the WebSphere Admin Console, on the *Data sources* page. The path to the parameters is:

Resources > JDBC> Data Sources

Clicking the *JACADA_JDBC_DB* link enables you to configure parameters for Jacada WorkSpace. Clicking the *metaDataDS* link enables you to configure parameters for Jacada Dynamic Views.

Under *Additional Properties* on the page of each data source, set the following configurations:

- Click on the *Connection pool properties* link, and then set the values for *Minimum connections* and *Maximum connections*, according to your environment.
- Click on the *Custom properties* link, click *New* and add the following property:
`defaultConnectionTypeOverride`. Its value should be `unshared`.
- Click on the *WebSphere Application Server Data source properties* link. Then, in the *Statements cache size* textbox, enter an appropriate value, according to your environment.
- Check the following checkboxes, and set relevant values:
 - *Validate new connections*
 - *Validate existing pooled connections*

Configuring the Apache Web Server

Jacada WorkSpace supports incorporating the Apache web server into the product architecture. The Apache Web Server is responsible for redirecting Web client requests for Web-based data to the deployed Jacada WorkSpace project in the application server.

To configure an Apache Web Server to redirect Web client requests to a WebSphere application server, perform the following actions (see the documentation for both the IBM WebSphere-Update Installer for WebSphere v8.5.5 and the Installation Wizard for Web Server Plug-ins):

- Install the WebSphere application server plug-ins and upgrade them to version 8.5.5.
- Use the IBM-provided manual installation steps and update the WebSphere plugin for Apache Web Server to the applicable 2.2.x version.

Update the file `Plugin-cfg.xml` file, located in the directory

`<WebSphere_deployment_manager_profile>/config/cells`, such that each instance of the `Transport Hostname` variable is defined with the IP address of each host machine of the WebSphere application server.

Make sure that the values of the `ConnectTimeout` and `ServerIOTimeout` are configured according to the needs of your production system.

Note: `Transport Hostname` variables must be defined, following every new generation of the `Plugin-cfg.xml` file by the administrator, for example, whenever the cluster configuration changes.

Only use the `<WebSphere_deployment_manager_profile>/config/cells/Plugin-cfg.xml` file for each web server in the configuration. Do not generate a separate plug-in for each web server in the configuration.

Cache must also be configured for use by the Apache Web Server. This cache can reduce the response time to client requests, since, when possible, the Apache Web Server will respond to the client based on its cache content and does not have to convey the client request to the application server for processing.

To configure Apache Web Server cache:

1. In the Apache HTTP server configuration file `<apache_installation_folder>\conf\httpd.conf`, make the following updates:

- a. Remove the comments from around the line:

```
LoadModule cache_module modules/mod_cache.so
```

- b. Remove the comments from around the line:

```
LoadModule disk_cache_module modules/mod_disk_cache.so
```

- c. Add these lines:

```
<IfModule mod_cache.c>
<IfModule mod_disk_cache.c>
    CacheEnable disk /
    CacheRoot "<apache_install>/cache"
    CacheDirLevels 5
    CacheDirLength 3
    CacheDefaultExpire 3600
</IfModule>
</IfModule>
```

2. In the Web server's host machine, create the directory `<apache_install>\cache`.

Use the load balancing option `Random` for each Apache web server that is configured to work with the cluster members. This option is a request routing property for web server plug-ins and applies to all requests that an Apache web server routes to WebSphere application servers. Configure the option in the WebSphere Integrated Solutions Console.

To configure the request routing property load balancing option:

1. In the task menu of the WebSphere Integrated Solutions Console, select **Servers > Web servers**.
2. In the displayed table of web servers, click on a relevant `<web_server_name>` link.
The *Configuration* tab displays.
3. In the tab's *Additional Properties* list, click the *Plug-in properties* link.
The *Configuration* tab displays.
4. In the tab's *Additional Properties* list, click the *Request routing* link.
The *Request routing Configuration* tab displays.

Web servers

[Web servers](#) > [webserver1](#) > [Plug-in properties](#) > **Request routing**

Use this page to configure request routing properties for a Webs server plug-in. These properties apply to all requests the Web server routes to application servers.

Configuration

Request routing

Load balancing option

* Retry interval
 seconds

Maximum size of request content
☒ No Limit
☐ Set Limit
 KBytes

* Maximum buffer size used when reading the HTTP request content
 KBytes

☒ Remove special headers
☐ Clone separator change

5. In the *Load balancing option* of *Request routing*, select `Random` from the dropdown menu.
6. Click *Apply*.
7. In the *Plug-in properties Configuration* tab *Additional Properties* list, click the *Request and response* link.
The *Request and response Configuration* tab displays.
8. Clear the *Enable the Nagle Algorithm for connections to the Application Server* checkbox.
9. Click *Apply*.

10. Click **Save**.
11. Repeat this configuration procedure for each web server that is configured to work with the cluster members.

Setting Up Computer Telephony Integration

The use of Jacada WorkSpace with a computer telephony integration (CTI) application is based on configuring settings in the database, in the following tables, which are maintained using the Jacada Management Console:

- **CTI_SETTINGS** – Used to configure CTI profiles. A CTI profile contains the configuration that is necessary for a specific CTI provider, or CTI provider and switch combination.
- **MACHINE_SETTINGS** – Used to associate each CTI Gateway with a CTI provider, using a CTI profile, thus enabling the use of more than one CTI provider.
- **GLOBAL_SETTINGS** – The setting **CTIGatewayLocation** identifies the host and port of the machine on which the CTI Gateway is installed.

The setting's default value, `currentInstance`, indicates that the CTI Gateway is located on the same server, or managed server in a cluster configuration, running Jacada WorkSpace. This configuration is recommended. If the CTI Gateway is located on a different machine, replace the default value with the following URL, which must contain the name or IP address of the host machine and the relevant port:

```
http://<host>:<port>/CTIGateway
```

For more information on CTI, refer to [Configuring CTI](#).

For more information on the database tables used by Jacada WorkSpace, refer to [Configuring Jacada WorkSpace in the Database](#).

Setting Up the Agent Desktop

The following sections describe tasks necessary for setting up the agent desktop for Jacada WorkSpace:

- [Installing Supplementary Files for Jacada WorkSpace](#)
- [Configuring Internet Explorer for the Agent](#)
- [Establishing a Dedicated HTTP Connection for Push Messaging](#)

Installing Supplementary Files for Jacada WorkSpace

For Jacada WorkSpace to function, you must install signed ActiveX controls via the `ApplicationHolderSetup.msi` file. The `ApplicationHolderSetup.msi` file installs the ActiveX controls. Jacada provides this installation file as a separate download.

The controls must be installed on each desktop machine that will be used to run Jacada WorkSpace.

Configuring Internet Explorer for the Agent

For Jacada WorkSpace to function with Internet Explorer, the browser settings must be configured to work with Jacada WorkSpace, and when using Internet Explorer version 8, the Jacada WorkSpace solution must open in the first tab.

Make sure that your browser settings, located in *Tools > Internet Options*, are configured as follows:

General Tab

1. In the *Browsing history* section, when using either Internet Explorer 7 or Internet Explorer 8, click the *Settings* button.
2. For the setting *Check for newer versions of stored pages*, select the *Automatically* radio button.

Security Tab

- Select the relevant Web content zone, and click the Custom Level button.

The *Security Settings* dialog box opens. The following table lists the required settings.

Security Setting Category	Security Setting Name	Required Setting Status
ActiveX controls and plug-ins	<i>Binary and script behaviors</i>	Enable
	<i>Initialize and script ActiveX controls not marked as safe</i>	Enable

Security Setting Category	Security Setting Name	Required Setting Status
	<i>Run Activex controls and Plug-ins</i>	Enable
	<i>Script ActiveX controls marked safe for scripting</i>	Enable
Miscellaneous	<i>Access data sources across domains</i>	Enable Required for integration with Jacada Integration and Automation (JIA) distributed DTA.
	<i>Allow script-initiated windows without size or position constraints</i>	Enable Required when using Internet Explorer 8 and any certified Microsoft operating system runs the agent desktop.

- Additional security settings are required when the operating system on which the Jacada WorkSpace solution runs is Windows 7. In Windows 7, Microsoft uses a User Account Control (UAC) technology and security infrastructure to improve security by limiting application software to standard user privileges, unless specified otherwise, using integrity levels.

There are three integrity levels: high, medium and low. The higher the integrity level, the greater the access privileges. A high integrity level allows administrative privileges as apposed to a low integrity level, which enables only generic execute access. A high integrity level can run a high, medium and low executable, while a medium integrity level can run a medium or low executable. When applications need to talk with each other, they must have the same integrity level. For example, medium integrity level applications can talk to a low integrity level application. However, the low integrity level application cannot respond.

Note: Applications usually spawn other application with their integrity levels.

This security mechanism in Windows 7 is used when protected mode is On. However, legacy applications are often required to run with protected mode Off. If this is the case, automation under Windows 7 functions normally and there is no need to change any settings.

Add the URL to log on to the Jacada WorkSpace solution to the *Trusted sites* zone: select the *Trusted sites* zone, and click the *Sites* button. Make sure that the *Enable Protected Mode* option is set to the default, which is off.

Advanced Tab

- Under the *HTTP 1.1 settings*, make sure that the *Use Http 1.1* option is checked.

Note: Jacada recommends not using add-ons in Internet Explorer. For example, using the Skype add-on causes a memory leak every time a web page is reloaded. This memory leak also occurs when the Web page is loaded in a tab in Jacada WorkSpace.

If you must use an add-on, check the impact on Jacada WorkSpace and if necessary nest the Internet Explorer application with the Web page in the tab and not just the frame.

When using Internet Explorer 8, make sure that the browser is set for 100%.

Restart the browser.

Establishing a Dedicated HTTP Connection for Push Messaging

You can configure Internet Explorer to establish an additional HTTP connection with the application server host. Jacada WorkSpace then exclusively uses this HTTP connection for sending Jacada WorkSpace push-related requests, making message processing more efficient and increasing overall message throughput between a Jacada WorkSpace session and the server.

Typically, Internet Explorer 7 establishes two HTTP connections with the application server; Internet Explorer 8 establishes six HTTP connections with the application server. These connections are

then used to carry all message traffic – the main application requests and responses and the push application requests and responses. Push requests often require more processing time, thereby forcing any main application requests that follow a push request to be routed over the remaining, available HTTP connection.

To configure a Push Connection URL for Jacada WorkSpace messaging:

1. Add a new entry in the organization's DNS that points to the IP address of the load balancer directing communication traffic to the Jacada WorkSpace application server host.
2. Add the URL that you provided in the *DedicatedPushConnectionURL* field in the *Global Settings* page of the Jacada Management Console to the entry added in the organization's DNS that points to the IP address of the load balancer.
3. In the Jacada Management Console navigation panel, select the *Configuration > Global Settings* option.

The *Global Settings* page displays.

4. In the *DedicatedPushConnectionURL* field, enter a URL in the format
`http://<alternate_application_server_name>:<http_port>`.
`<alternate_application_server_name>` must match the name provided in the organization's Domain Name Server (DNS). The value you provide in the `<http_port>` portion of the URL must be the Jacada WorkSpace application server's assigned HTTP communications port. For example, for IBM WebSphere, the port numbered 9080.
During TCP/IP communication processing, the desktop machine uses the organization's DNS to resolve the alternate server name. The DNS provides the IP address of the Jacada WorkSpace application server host.
5. Click *Apply*.

Installing Jacada WorkSpace for Production: WebLogic

The following topics describe how to install Jacada WorkSpace in an Oracle WebLogic Server production environment, in order to deploy an Agent WorkSpace:

- [Production Requirements](#)
- [Installation for Production: Overview](#)
- [Installation of Jacada WorkSpace](#)
- [Installation of a Clustered Production Environment](#)
- [Load Balancing the WebLogic Server](#)
- [Additional WebLogic Server Configuration](#)
- [Setting Up Licensing](#)
- [Configuring the Apache Web Server](#)
- [Setting Up Computer Telephony Integration](#)
- [Adding Managed Servers to an Existing Cluster](#)
- [Creating Additional Server Domains](#)
- [Setting Up the Agent Desktop](#)

Production Requirements

This section describes the requirements for a Jacada WorkSpace production environment.

Agent Desktop Production Requirements

Jacada WorkSpace is certified with the following configuration for the call center agent desktop:

Certified Configuration

Operating System	Microsoft Windows XP Professional with Service Pack 3 Microsoft Windows 7, Enterprise edition (32 bit and 64 bit)
Display Resolution	1280x1024 (minimum) 1366x768 and above (recommended)
Browser	• Microsoft Internet Explorer version 9 (32 bit and 64 bit) • Microsoft Internet Explorer version 10 (32 bit and 64 bit) • Microsoft Internet Explorer version 11 (32 bit and 64 bit) • Google Chrome (without hosting)

Windows Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Windows server:

Certified Configuration for Windows

Minimum Hardware	CPU	Intel Xeon 2 x 2.4GHz
	RAM	2 GB (4 GB recommended)
Software	Operating System*	Windows 2012 Server R2 64 bit
	Application Server	Oracle WebLogic server 12.1.3 (64 bit) with Sun JDK 7 update 71
	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0 • Genesys Telecommunications Laboratories Inc. – Genesys 8.0 and 8.1

Certified Configuration for Windows

		Both vendors are supported with the following switch: Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	- Oracle 11g R2 - SQL Server 2008 - MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.10 (32 bit)

*** You can work with Windows under VMware ESX.**

Linux RedHat Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Linux RedHat server:

Certified Configuration for Linux RedHat

Minimum Hardware	CPU	2 x Xeon E5420 (2.5 GHz) Quad Core
	RAM	2 GB (4 GB recommended)
Software	Operating System	Linux RedHat Enterprise 6.6 (64 bit)
	Application Server	Oracle WebLogic server 12.1.3 (64 bit) with Sun JDK 7 update 71

Certified Configuration for Linux RedHat

	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0 • Genesys Telecommunications Laboratories Inc. – Genesys 8.0 and 8.1 Both vendors are supported with the following switch: • Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	• Oracle 11g R2 • SQL Server 2008 • MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.15

Installation for Production - Overview

Jacada WorkSpace projects are deployed to the Oracle WebLogic Server. The WebLogic Server is comprised of the following server instances:

Name	Description
Administration Server	A WebLogic server instance in the server domain whose purpose is to: • Run the WebLogic Server Administration Console. • Perform server administration and application deployments. • Synchronize the managed servers of a cluster.

Name	Description
Managed Server	A WebLogic server instance in the server domain that runs the deployed Enterprise Application modules.

The supported WebLogic Server configurations for a production environment are:

- **A single server instance:** Applicable to small-scale production environments, product testing and product staging/controlled roll-out. The WebLogic single server performs the functions of both the Administration Server and the Managed Server.
- **A clustered server:** Used for a large-scale production environments. The cluster server environment provides failover capabilities. The environment includes a distinct Administration Server and a cluster of multiple Managed Servers. The Managed Servers typically reside on separate host machines.

Note: This guide assumes that you are familiar with the Oracle WebLogic Configuration Wizard and the Oracle WebLogic Server Administration Console.

Installation Sequence

Jacada requires the following installation sequence:

Server Environment	Installation Sequence
WebLogic Cluster	1. Oracle WebLogic Server 12.1.3 (64 bit) with Sun JDK 7 update 71 2. Database management system (Oracle 11g R2/SQL Server 2008/MySQL 5.5_29) 3. Jacada WorkSpace 7.3 without running the Jacada Server Setup wizard. Only after installing the product, configure the cluster using the Oracle WebLogic Configuration Wizard.
WebLogic Single Server	1. Oracle WebLogic Server 12.1.3 (64 bit) with Sun JDK 7 update 71 2. Database management system (Oracle 11g R2/SQL Server 2008/MySQL 5.5_29) 3. Jacada WorkSpace 7.3 including running the Jacada Server Setup wizard.

Installation of Jacada WorkSpace

The Jacada WorkSpace installation includes the following activities:

- Installation of Jacada WorkSpace.
- Creation of a WebLogic Server domain. A domain provides the environment for running a particular WebLogic Server. The process also configures the database schema to be used with the WebLogic Server.

If your production environment is a WebLogic Server cluster configuration, install only the Jacada WorkSpace product files. If your production environment is a single server instance of WebLogic Server, run the server domain setup too.

IMPORTANT: Before starting the installation, verify that the **beahomelist** file exists in **C:\Oracle\Middleware**. In addition, verify that the name of your computer does not contain an underscore character.

To install Jacada WorkSpace:

1. In the folder containing the Jacada WorkSpace installation, do one of the following:
 - If the production server is going to run under the Linux RedHat operating system, execute **./setup.bin -i console**.
 - If the production server is going to run under one of the Windows operating systems, run the **setup.exe** file.

The *Introduction* window opens.

2. Follows the installation instructions until the *Setup a Server* window displays.

When installing on Linux RedHat, this window is skipped.

When installing on a Windows operating system, do one of the following:

- If your production environment is a WebLogic Server cluster configuration, select the *No* option.
The Jacada WorkSpace product file installation completes.
To create the domain for a WebLogic Server cluster configuration, refer to [Configuring the WebLogic Server Cluster](#).
- If your production environment is a WebLogic single server instance, select the *Yes* option and click *Next*.
After a brief processing period, the *Server Setup Wizard Introduction* window displays.
Run this wizard to both create a new WebLogic Server domain and define the database

details of the database that your WebLogic Server will use.

Note: Under UNIX-based operating systems, the user that runs the wizard needs to have the \$HOME/OracleWebLogic/OracleWebLogichomelist information available along with appropriate privileges for creating the new domain.

If you had a domain defined for a previous version of Jacada WorkSpace and you want to use the database created for this previous version, if the database cannot be upgraded as part of the definition of the domain it can be upgraded later, before starting to work with the new version, as follows:

1. Edit the file **<Jacada_WorkSpace_installation_folder>\WorkSpace\Accessories\Schema\db.properties**.
2. In the file, provide the required database-related information, as defined for each entry.
3. Run the `schemaBuilder.cmd` script, located in the same directory.
4. The script loads the information provided in the **db.properties** file into the specified database schema.

The Jacada WorkSpace product installation is complete.

Installation of a Clustered Production Environment

In a production environment, Jacada WorkSpace is typically deployed in a cluster of managed servers. A cluster defines a group of managed server instances of Oracle WebLogic Server that work together processing requests and, thereby, increase scalability and reliability of the business' application processing. From the client's point of view, there appears to be only one instance of WebLogic Server. Clusters provide efficient application administration through update synchronization across managed servers.

The following table describes the server components associated with a WebLogic Server cluster:

Name	Required/ Optional	Description
Administration Server	1 such server is required.	A WebLogic server instance in the server domain whose purpose is to: • Run the WebLogic Server Administration Console. • Perform server administration and application deployments.

Name	Required/ Optional	Description
		Synchronize the managed servers of a cluster.
Managed Server	A minimum of 2 such servers are required.	Multiple WebLogic server instances, in the server domain, that are members of the cluster and run the deployed Enterprise Application modules.
Node Manager	Optional: 1 Node Manager process per host machine.	This utility enables using the WebLogic Server Administration Console to remotely start, shut down, and restart all Managed Servers of a server domain residing in the same host machine as the Node Manager. For information about configuring the Oracle Node Manager, contact Jacada Support.

The managed servers can reside on the same machine as the administration server or on different machines. Jacada WorkSpace projects are deployed to the managed servers of the cluster. For more information on WebLogic clustering, refer to the Oracle WebLogic documentation.

To run Jacada WorkSpace projects in a WebLogic Server cluster production environment, configure a cluster using the Oracle WebLogic Configuration Wizard.

Configuring the WebLogic Server Cluster

On the host machine, configure WebLogic Server as a cluster of managed servers. Cluster configuration process creates the following:

- An Administration Server instance, named AdminServer.
- At least one Managed Server instance.
- A WebLogic Server domain. A domain provides the environment for running the server.
- The JDBC data sources that your deployed Jacada WorkSpace applications will use.

Use the Oracle WebLogic Configuration Wizard to configure the WebLogic Server cluster. When the WebLogic Server cluster is configured across multiple host machines, then the machine hosting the Administration Server has at least one Managed Server configured in it.

When the WebLogic Server cluster resides entirely in a single host machine, configure the cluster using only the first procedure below (To configure a WebLogic Server cluster). When the WebLogic Server cluster resides in multiple host machines, perform both the following procedures.

To configure a WebLogic server cluster using the Configuration wizard:

1. Linux RedHat operating system:

a. In the host machine, change your current directory to the WebLogic server bin directory, **<WL_Home>/wlserver/common/bin** where **<WL_Home>** is the top-level installation directory for the WebLogic Server. For example, in a Windows host machine, **C:\Oracle\Middleware\Oracle_Home\wlserver**.

b. Run the **config.sh** file.

If you are running the Oracle WebLogic Configuration Wizard in console mode, run **config.sh - mode=console**.

Windows operating system:

Run *Start > Programs > Oracle > OracleHome> WebLogic Server 12c > Tools > Configuration Wizard*.

2. Select the *Create a New WebLogic domain* option and click *Next*.

The *Select Domain Source* screen is displayed.

3. Select the *custom template* option.

4. In the *Template location* field, enter the file path **<Jacada_WorkSpace_installation_folder>/Common/Utilities/ServerSetup/WebLogic/jad_cluster_template.jar** and click *Next*.

5. Enter a name for the domain.

The *Configure Administrator Username and Password* screen is displayed.

6. Enter the required information to define an administrator user and password for the WebLogic Server domain. Click *Next*.

The *Configure Server Start Mode and JDK* screen is displayed.

7. In the *WebLogic Domain Startup Mode* pane, select the *Production Mode* option. If you are using console mode select the **Other Java SDK** option.

Note: If this option is not selected and, post cluster configuration, you are ready to configure the Production Mode option, refer to [Configuring WebLogic Server To Start in Production Mode](#).

8. In the *JDK Selection* pane, select the path to `SUN JDK 1.7.0_71` which is installed on the machine and click *Next*.

If you are using console mode enter the full path to `JDK 1.7.0_71`.

The *Configure JDBC Data Sources* screen is displayed.

9. Based on the type of database being used, modify the required information to configure the JDBC data sources that your applications will use. Click *Next*.

Consider the following:

- Accept the default setting for the *JNDI name* field.
- Set the *Driver* field as follows:
 - For Microsoft SQL Server 2008, select the *Oracle's MS SQL Server Driver (Type 4) Versions: 7.0 and later* option.
 - For Oracle version 11g R2 database types, select the *Oracle's Driver (Thin) for Instance connections; Versions 9.0.1 and later* option.
- Set the *DBMS User name* field to be the same as the database schema name.

The *Select Optional Configuration* screen is displayed.

10. Enter the required information for configuring a WebLogic administration server. Accept the default setting for the *SSL Enabled* setting.
11. Click *Next*.

Note: Make sure that any port you assign, whether it be a default setting or not, is available in the host machine.

The *Configure Managed Servers* screen is displayed.

12. Enter the required information to configure a WebLogic managed server. Repeat this step for each managed server in the cluster. Click *Next*.

The *Configure Clusters* screen is displayed.

13. Click *Add* and enter the required information for configuring a WebLogic cluster. Click *Next*.

Note: Make sure to assign *IP_address* and *Port* combinations that are unique, to avoid IP traffic collision and confusion. Make sure that any port you assign, whether it be a default setting or not, is available in the host machine. For the *Cluster Addresses*, provide the *IP_address:Port* assignment of each managed server.

The *Assign Servers to Clusters* screen is displayed.

14. Select a cluster in the *Cluster* pane, then select a managed server in the *Server* pane and click the *Right Arrow*. Repeat this step for each managed server you want in the cluster.
15. Click *Next*.

The managed server is assigned to the cluster. The *Configure Machines* screen is displayed.

16. Click *Add* and do the following:

If Linux RedHat is the operating system of the server's host machine, click the link to learn about the appropriate field settings for Configuring Machines.

If Windows is the operating system of the server's host machine, accept the default settings.

17. Click *Next*.

The *Assign Servers to Machines* screen is displayed.

18. In the *Machine* pane, select a machine, then select a server in the *Server* pane and click the *Right Arrow*. Repeat this step for each server you want associated with the selected machine. Then, click *Next*.

The server is associated with the machine.

Click *Next*.

The *Creating Domain* progress window is displayed. When the process completes successfully, the *Domain Creation Summary* information is displayed.

The *Domain Created Successfully* message appears.

Note: The WebLogic Server cluster can reside in multiple host machines; In this configuration, some of the cluster's Managed Servers reside in one or more than one remote host machine, this being a machine other than the one hosting the domain's Administration Server. If this is your desired cluster configuration, continue with the next procedure.

To configure the WebLogic server cluster in additional host machines:

1. Linux RedHat operating system:

a. In the host machine, change your current directory to the WebLogic server bin directory, **<WL_Home>/wlserver/common/bin** where **<WL_Home>** is the top-level installation directory for the WebLogic Server. For example, in a Windows host machine, **C:\Oracle\Middleware\Oracle_Home\wlserver**.

b. Run the **config.sh** file.

If you are running the Oracle WebLogic Configuration Wizard in console mode, run **config.sh - mode=console**.

Windows operating system:

Run *Start > Programs > Oracle WebLogic > WebLogic Server 11gR1 > Tools > Configuration Wizard*.

2. Select the *Create a New WebLogic domain* option and click *Next*.

The *Select Domain Source* screen is displayed.

3. Select the *custom template* option.

4. In the *Template location* field, enter the file path **<Jacada_WorkSpace_installation_folder>/Common/Utilities/ServerSetup/WebLogic/jad_cluster_template.jar** and click *Next*.

5. Enter a name for the domain.

The *Configure Administrator Username and Password* screen is displayed.

6. Enter the required information to define an administrator user and password for the WebLogic Server domain. Click *Next*.

The *Configure Server Start Mode and JDK* screen is displayed.

7. In the *WebLogic Domain Startup Mode* pane, select the *Production Mode* option.

If you using console mode select the **Other Java SDK** option.

Note: If this option is not selected and, post cluster configuration, you are ready to configure the Production Mode option, refer to [Configuring WebLogic Server To Start in Production Mode](#).

8. In the *JDK Selection* pane, select the path to **SUN JDK 1.7.0_71** which is installed on the machine and click *Next*.

If you are using console mode enter the full path to **JDK 1.7.0_71**.

The *Configure JDBC Data Sources* screen is displayed.

9. Based on the type of database being used, modify the required information to configure the JDBC data sources that your applications will use. Click *Next*.

Consider the following:

- Accept the default setting for the *JNDI name* field.
- Set the *Driver* field as follows:
For Microsoft SQL Server 2008, select the *Oracle's MS SQL Server Driver (Type 4) Versions:7.0 and later* option.
For Oracle version 11g R2 database types, select the *Oracle's Driver (Thin) for Instance connections; Versions 9.0.1 and later* option.
- Set the *DBMS User name* field to be the same as the database schema name.

The *Select Optional Configuration* screen is displayed.

10. Enter the required information for configuring a WebLogic administration server. Accept the default setting for the *SSL Enabled* setting.

11. Click *Next*.

Note: Make sure that any port you assign, whether it be a default setting or not, is available in the host machine.

The *Configure Managed Servers* screen is displayed.

12. Enter the required information to configure a WebLogic managed server that is hosted in this additional machine.

Make sure of the following:

- That the name you assign to the managed server is the name assigned to the same managed server that you configured in the previous procedure.
- Select the relevant IP address of your server machine in the Listen address dropdown list.
- Assign the same Listen port to the managed server that you assigned to the same managed server configured in the previous procedure.
- That any port you assign, whether it be a default setting or not, is available in the host machine.

Repeat this step for each managed server in the cluster that is hosted in this additional machine.

13. Select a cluster in the *Cluster* pane, then select a managed server in the *Server* pane and click the *Right Arrow*. Repeat this step for each managed server you want in the cluster.

14. Click *Next*.

The managed server is assigned to the cluster. The *Configure Machines* screen is displayed.

15. Click *Add* and do the following:

If Linux RedHat is the operating system of the server's host machine, click the link to learn about the appropriate field settings for *Configuring Machines*.

If Windows is the operating system of the server's host machine, accept the default settings.

16. Click *Next*.

The *Assign Servers to Machines* screen is displayed.

17. In the *Machine* pane, select a machine, then select a server in the *Server* pane and click the *Right Arrow*. Repeat this step for each server you want associated with the selected machine. Then, click *Next*.

The server is associated with the machine.

Click *Next*.

The *Creating Domain* progress window is displayed. When the process completes successfully, the *Domain Creation Summary* information is displayed.

The *Domain Created Successfully* message appears.

Load Balancing the WebLogic Server Cluster

In a typical cluster configuration, a load balancer is configured as a front-end to the WebLogic cluster of managed servers. For the call center agent client machines, the load balancer is the single access point to the cluster and the request distributor to the cluster's managed servers.

The following sections describe:

- [BIG-IP Local Traffic Management System](#)
- [Secure Communication for Load Balancer](#)

The BIG-IP Local Traffic Management System

Jacada recommends deploying the F5 Network BIG-IP Local Traffic Management (LTM) system for directing traffic to the WebLogic managed servers running Jacada WorkSpace. A BIG-IP LTM configuration consists of defining the following components:

- A virtual server (the traffic manager)
- A connection pool (a set of devices grouped together to receive traffic according to a load balancing method) associated with a virtual server
- The connection pool members (the WebLogic cluster's managed servers)

For BIG-IP LTM configuration information and procedures, refer to the deployment guide entitled *Deploying the BIG-IP LTM System with Oracle WebLogic Servers*.

Configuring Properties and Resources

In order for Jacada WorkSpace traffic to properly travel between the agent client machines, the LTM virtual server and the WebLogic cluster's managed servers, configure the following properties and resources in the BIG-IP LTM Configuration utility:

Default Persistence Profile – Set this resource to `cookie`. The LTM virtual server uses cookies to assign each agent client machine to a specific WebLogic cluster member in the connection pool. This setting instructs the virtual server to always direct requests from an agent client machine to the same WebLogic cluster member running Jacada WorkSpace. Access the resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.

HTTP Profile – Set this property to `http`. Access the property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

Protocol Profile (Client) and Protocol Profile (Server) – Set both these properties to the custom tcp profile of your choice. The assigned tcp profile must have its *Nagle's Algorithm* setting unchecked. By disabling this property, you prevent unnecessary TCP/IP delay of small packets during their routing. Access these properties in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

SSL Profile (Client) – Set this property to `clientssl`. This setting instructs the virtual server that communication with any agent client machines is always encrypted. Access this property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

Health Monitors – For the *Default Pool* resource of the LTM virtual server, assign as active a monitor of type TCP to its *Health Monitors* property. The assigned monitor must have its *Interval* configuration property set to 30 seconds and its *Timeout* configuration property set to 91 seconds. These resource and property configurations cause the LTM to monitor the TCP connection status of its application server pool members, when load balancing new connection requests to the cluster of servers.

- Access the *Default Pool* resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.
- Access the *Health Monitors* property in the *Main Tab > Local Traffic > Pools > Pool List > <pool_name> > Properties > Configuration > Basic* properties.
- Access the *TCP monitor Interval and Timeout* properties in the *Main Tab > Local Traffic > Monitors > Monitor List > <monitor_name> > Properties > Configuration > Basic* properties.

In a clustered, production environment that uses the F5 Network BIG-IP LTM system to provide load

balancing, Jacada WorkSpace supports the failover mechanism. For a description of the various capabilities and settings required to implement failover for the Jacada WorkSpace production environment, refer to [Failover and Switchover Support](#).

Secure Communication for Load Balancer

Jacada WorkSpace provides HTTPS support for hardware-based encryption that can be employed with the load balancer.

A common, production environment configuration uses HTTPS to encrypt the communication between the instances of Jacada WorkSpace on the agent desktops and the load balancer. Additionally, this production environment configuration uses unencrypted HTTP for the communication between the load balancer and the WebLogic managed servers running the Jacada WorkSpace application. For such a configuration to work in this manner, all internal WebLogic redirection requests must use HTTP.

This production environment configuration requires the following:

- Your load balancer has a property, which when set accordingly, instructs it to always encrypt communication with any agent client machine.
- Modification to the **weblogic.xml** file. Perform this modification during Jacada WorkSpace project development. For details about the required modification, refer to the Jacada WorkSpace Developer Guide.

Additional WebLogic Server Configurations

Several additional configurations pertaining to the WebLogic application server are necessary for its proper performance in serving Jacada WorkSpace. These additional configurations are:

- [Configuring WebLogic Server to Start in Production Mode](#)
- [Suspending the Generation of Diagnostic Information](#)
- [Configuring the WebLogic Application Server Work Manager](#)

Following the configuration of either a single server instance or a cluster of Managed Servers and the installation of Jacada WorkSpace, you can start the WebLogic Server. For the start procedures, refer to [Starting and Stopping the WebLogic Server](#).

Configuring WebLogic Server to Start in Production Mode

In a production environment, WebLogic servers must start in production mode.

To configure WebLogic server instances to start in production mode:

1. Stop all managed servers.
2. Stop the administration server.
3. In the host machine of the relevant WebLogic server instance or managed server instance (cluster), access the following directory: **<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin**
where **<OracleWebLogic_Home>** is the top-level installation directory for Oracle WebLogic software. For example, **C:\Oracle\Middleware**.
4. Edit the relevant file:
 - If Linux RedHat is the operating system of the server's host machine, edit the file **setDomainEnv.sh**.
 - If Windows is the operating system of the server's host machine, edit the file **setDomainEnv.cmd**.
5. Set the `PRODUCTION_MODE` variable to the value `true`.

```
set PRODUCTION_MODE=true
```
6. Save your updates and close the file.
7. Repeat Step 3 through Step 6 for each host machine of the relevant, WebLogic managed server instance (cluster).
8. Start the relevant application servers.

Suspending the Generation of Diagnostic Information

In a production environment, halt the diagnostic information that the domain's application servers generate.

To disable diagnostic information for WebLogic server instances:

1. Stop all managed servers.
2. Stop the administration server.
3. In the host machine of the relevant WebLogic server instance or managed server instance (cluster), access the following directory: **<OracleWebLogic_Home_Home>/user_projects/domains/<domain_name>/bin**
where **<OracleWebLogic_Home>** is the top-level installation directory for Oracle WebLogic software. For example, **C:\Oracle\Middleware**.
4. Edit the relevant file:
 - If Linux RedHat is the operating system of the server's host machine, edit the file **startWeblogic.sh**.
 - If Windows is the operating system of the server's host machine, edit the file **startWeblogic.cmd**.
5. Add the following assignment to the existing Java command-line options for running the servers, %JAVA_OPTIONS%:

```
-Dcom.OracleWebLogic.wlw.netui.disableInstrumentation=true
```
6. Save your updates and close the file.
7. Repeat Step 3 through Step 6 for each host machine of the relevant, WebLogic managed server instance (cluster).
8. Start the relevant application servers.

To increase the delay of diagnostic image captures for WebLogic server instances:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Diagnostics > Diagnostic Images*.
The *Summary of Diagnostic Images* page displays.
2. In the *Server* column of the *Diagnostic Images* table, click the relevant server link.
The *Settings for <server_name>* page displays.

3. In the *Timeout* field, replace the existing value with 1440 (minutes).

4. Click *Save*.

Server settings are updated. All changes are activated and no server restarts are necessary.

5. Repeat Step 1 through Step 4 for each application server listed in the *Diagnostic Images* table.

To increase the period of diagnostic store file size checks for WebLogic server instances:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Diagnostics > Archives*.

The *Summary of Diagnostic Archives* page displays.

2. In the *Server* column of the *Diagnostic Archives* table, click the relevant server link.

The *Settings for <server_name>* page displays.

3. In the *Store Size Check Period* field, replace the existing value with 9999 (hours).

4. Click *Save*.

Server settings are updated. All changes are activated and no server restarts are necessary.

5. Repeat Step 1 through Step 4 for each application server listed in the *Diagnostic Archives* table.

Configuring the WebLogic Application Server Work Manager

A work manager defines a set of request classes and thread constraints that manage work performed by WebLogic server instances. For Jacada WorkSpace purposes, global work managers are required and are defined at the domain level.

Using the WebLogic Server Administration Console, create a global work manager for one or more WebLogic server instances or clusters.

To configure the minimum and maximum pool size, change your *startWeblogic.sh* (linux), or *startWeblogic.cmd* (windows).

Add the following parameters to your *JAVA_HOME*:

```
-Dweblogic.threadpool.MinPoolSize=100 -Dweblogic.threadpool.MaxPoolSize=100
```

Setting Up Licensing

Obtain a Jacada WorkSpace license file, `license.jacada`, from your Jacada representative. Place the license file in the domain folder of the WebLogic Server,

<OracleWebLogic_Home>/user_projects/domains/<domain_name>, where **<OracleWebLogic_Home>** is the top-level installation folder for Oracle WebLogic software. For example, in a Windows host machine, **C:\Oracle\Middleware**.

Configuring the Apache Web Server

Jacada WorkSpace supports incorporating the Apache Web Server into the product architecture. The Apache Web Server is responsible to redirect Web client requests for Web-based data to the deployed Jacada WorkSpace project in the application server.

In order for the Apache Web Server to redirect to a WebLogic application server, the Web server requires the `mod_wl.so` plug-in, which is not supplied as part of the default installation of Apache Web Server.

Note: An Oracle supplied version of the `mod_wl.so` plug-in is not adapted to work with Jacada WorkSpace.

1. Download the **WLSPlugin11g-64bit-Apache2.2-linux64-x86_64.zip** file from the Oracle website.
2. Unzip the folder to **/usr/local/lib64**.
3. Copy **/usr/local/lib64/WLSPlugin11g-64bit-Apache2.2-linux64-x86_64/lib/mod_wl.so** to **/etc/httpd/modules**.
4. Create a new file "**oracle_plugin.sh**" in the **/etc/profile.d**.
The **oracle_plugin.sh** should contain **export LD_LIBRARY_PATH="\$LD_LIBRARY_PATH:/usr/local/lib64/WLSPlugin11g-64bit-Apache2.2-linux64-x86_64/lib"**.
5. In the Apache Web Server's host machine, put the plug-in in the **<apache_installation_folder>/modules** directory .

The following sections describe:

- [Configuring redirection of Web client requests](#)
- [Configuring the cache](#)
- [Modifying the Post Timeout setting](#)
- [Configuring the Session Cookie Name](#)

Configuring Redirection of Web Client Requests

To configure an Apache Web Server to redirect Web client requests, modify, in the Web server's host machine, the Apache HTTP server configuration file `<apache_installation_folder>\conf\httpd.conf`, as follows:

- In the `Dynamic Shared Object (DSO) Support` section of the file, add the following line:

```
LoadModule weblogic_module modules/mod_wl.so
```

- Add the following lines anywhere in the file:

```
<Location <WorkSpace_application_name>>
  SetHandler weblogic-handler
</Location>

<Location /JacadaWorkspaceCommon>
  SetHandler weblogic-handler
</Location>
```

Replace `<WorkSpace_application_name>` with the name of the relevant Jacada WorkSpace project.

- Add an `<IfModule weblogic_module>` section. For a clustered WebLogic application server environment, specify the server address and port for each server in the cluster:

```
<IfModule weblogic_module>
  WebLogicCluster
    <IP_address_application_server_host1>:<HTTP_port>,
    .
    .
    .
    <IP_address_application_server_hostn>:<HTTP_port>
</IfModule>
```

For a single WebLogic application server environment, specify the server address and port as follows:

```
<IfModule weblogic_module>
  WebLogicHost <IP_address_application_server_host>
  WebLogicPort <HTTP_port>
</IfModule>
```

Additionally, in the Web server's host machine, in the file **<apache_installation_folder>\conf\weblogic.conf**, define the maximum amount of time that the plug-in waits to receive a WebLogic application server response to its request. Accomplish this by defining in the file the **WLIOTimeoutSecs** parameter with the following value:

```
WLIOTimeoutSecs=600
```

Configuring the Apache Web Server Cache

A cache must be configured for use by the Apache Web Server.

To configure the Apache Web Server cache:

1. In the Apache HTTP server configuration file **<apache_installation_folder>\conf\httpd.conf**, make the following updates:

- a. Remove the comments from around the line:

```
LoadModule cache_module modules/mod_cache.so
```

- b. Remove the comments from around the line:

```
LoadModule disk_cache_module modules/mod_disk_cache.so
```

- c. Add the following lines:

```
<IfModule mod_cache.c>
<IfModule mod_disk_cache.c>
  CacheEnable disk /
  CacheRoot "<apache_install>/cache"
  CacheDirLevels 5
  CacheDirLength 3
  CacheDefaultExpire 3600
</IfModule>
</IfModule>
```

2. In the Web server's host machine, create the directory **<apache_install>\cache**.

Modifying the Post Timeout Setting

When Apache web server is included in the production environment of a WebLogic Server cluster, modify the HTTP connection timer setting for the managed servers that are configured in the WebLogic Server domain.

To modify the Post Timeout setting for a server:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Environment*.
The *Summary of Environment* page displays.
2. In the *Summary of Environment* page, click the *Servers* link.
3. In the *Summary of Servers* page, click the relevant `<server_name>` link in the table of server instances configured in the current WebLogic Server domain.
The *Settings for <server_name>* page displays.
4. In the page, select the *Protocols* tab.
The *General* tab displays.
5. In the *General* tab, keep all the default settings.
6. Select the *HTTP* tab.
The *HTTP* tab displays.
7. Click the *Lock & Edit* button in the *Change Center* pane of the console to modify settings in this domain.
8. In the *Post Timeout* field, modify the setting's value to be 120.
This modifies the amount of time the specific server waits between receiving chunks of data in an HTTP POST data before it times out. (Used to prevent denial-of-service attacks that attempt to overload the server with POST data.)
9. Click *Save*.
The HTTP setting is updated for the server instance `<server_name>` in the WebLogic server domain.
10. Click the *Activate Changes* button in the *Change Center* pane of the console to have the modified setting take effect for the server instance.
11. Repeat this configuration procedure for each managed server that is configured in the WebLogic Server domain.

Configuring the Session Cookie Name

The default name for the cookie used to track a Jacada WorkSpace session is `jsessionId`. If desired, you can change the cookie name, to avoid cookie sharing between Jacada WorkSpace and other nested web applications.

Changing the cookie name involves modifying the **weblogic.conf** file, as shown in the following example. `@NEW_COOKIE@` should be replaced with the required name.

```
WLCookieName @NEW_COOKIE@

<Location /JacadaWorkspaceCommon>
    SetHandler weblogic-handler
    WLCookieName PushSessionId
</Location>

<Location /JacadaWorkspaceCommon/**/startPush>
    SetHandler weblogic-handler
    WLCookieName @NEW_COOKIE@
</Location>

<Location /mc>
    SetHandler weblogic-handler
    WLCookieName JSESSIONID
</Location>
```

When a cookie name other than the default (`jsessionId`) is used, verify that the following additional required configurations are done:

- Defining the `sessionCookieName` setting in the [Global Settings page](#) of the Jacada Management Console.
- Making relevant configuration changes at the project level. For more information, refer to the Jacada WorkSpace Developer Guide (Setting Up the Session Cookie Name).

Adding Managed Servers to an Existing Cluster

For an existing domain, you can create new WebLogic server instances and add them as managed server members of a cluster installation. Accomplish this cluster expansion using the following sequence of procedures:

- [Creating a New Cluster Member](#)
- [Creating and Deploying the JMS Server Instance](#)
- [Configuring a Topic Destination Resource](#)

Creating a New Cluster Member

The following procedure explains how to create a new WebLogic managed server cluster member.

To create a new WebLogic cluster member:

1. In the task menu of the application server console, select *Environment > Servers*.
The *Summary of Servers* page displays.
2. In the *Change Center* pane, click *Lock and Edit*.
3. Click *New*.
The *Create a New Server* page displays.

Create a New Server

Back Next Finish Cancel

Server Properties
The following properties will be used to identify your new server.

* Indicates required fields

What would you like to name your new server?

*Server Name: MS4

Where will this server listen for incoming connections?

Server Listen Address:

Server Listen Port: 6008

Should this server belong to a cluster?

☐ No, this is a stand-alone server.

☒ Yes, make this server a member of an existing cluster.

Select a cluster: Cluster1

☐ Yes, create a new cluster for this server.

Back Next Finish Cancel

4. In the *Server Name* field, type a name for the new server.
5. In the *Server Listen Address* field, type the IP address of the machine that is hosting the new server.
6. In the *Server Listen Port* field, type an available port number.
7. Select the option *Yes, make this server a member of an existing cluster*.
8. In the *Select a cluster* field, select a cluster from the dropdown menu.
This is the cluster to which the new server is added as a member.
9. Click *Next*.
The *Create a New Server* page displays, containing the new server's configuration information for review.
10. Click *Finish*.
The *Summary of Servers* page displays, containing the table of servers configured in the current domain.
11. In the *Change Center* pane, click *Activate Changes*.

You created a new managed server in the current domain and made it a member of the selected cluster installation.

Now, continue with the next procedure, [Creating and Deploying the JMS Server Instance](#).

Creating and Deploying a JMS Server Instance

The following procedure explains how to create a new instance of JMS server and deploy it on the new managed server.

To create and deploy a JMS server instance:

1. In the task menu of the application server console, select *Services > Messaging > JMS Servers*. The *Summary of JMS Servers* page displays.

Summary of JMS Servers

JMS servers act as management containers for the queues and topics in JMS modules that are targeted to them.

This page summarizes the JMS servers that have been created in the current WebLogic Server domain.

[Customize this table](#)

JMS Servers(Filtered - More Columns Exist)

NewDelete

Showing 1 - 3 of 3PreviousNext

☐	Name ↕	Persistent Store	Target	Current Server
☐	JMSServer-0_auto_1		MS1	MS1
☐	JMSServer-0_auto_2		MS2	MS2
☐	JMSServer-0_auto_3		MS3	MS3

NewDelete

Showing 1 - 3 of 3PreviousNext

2. In the *Change Center* pane, click *Lock and Edit*.
3. Click *New*.
The *Create a New JMS Server* page displays.

4. In the *Name* field, enter a name for the new JMS server.
 5. In the *Persistent Store* field, select the option (none) from the dropdown menu.
 6. Click *Next*.
The *Create a New JMS Server* page re-displays.
 7. In the *Target* field, select from the dropdown menu the new `<server_name>` that you created using the previous procedure ([Creating a New Cluster Member](#)).
 8. Click *Finish*.
The *Summary of JMS Servers* page displays, containing the table of JMS servers created in the current domain.
 9. In the *Change Center* pane, click *Activate Changes*.
In the current domain, you created a new JMS server and deployed it on the new server instance, previously created.
- Now, continue with the next procedure, [Configuring and Deploying a Topic Destination Resource](#).

Configuring a Topic Destination Resource

The following procedure explains how to configure a new Topic Destination resource.

To configure JMS resources:

1. From the WebLogic Admin Console, navigate to *Services > Messaging > JMS Modules*, and select *PushTopicModule*.

JMS Modules	
Click the <i>Lock & Edit</i> button in the Change Center to activate all the buttons on this page.	
New Delete	Showing 1 to 1 of 1 Previous Next
☐ Name ↕	Type
☐ PushTopicModule	System
New Delete	Showing 1 to 1 of 1 Previous Next

- The *Settings for PushTopicModule* panel opens.
- At the top of the panel, select the *Subdeployment* tab. A list of subdeployments is displayed. If JMS is deployed on a cluster, you should have one subdeployment. If JMS is deployed on a cluster and an admin server, two subdeployments are listed. For example:

Subdeployments		
Click the <i>Lock & Edit</i> button in the Change Center to activate all the buttons on this page.		
New Delete	Showing 1 to 2 of 2 Previous Next	
☐ Name ↕	Resources	Targets
☐ JMServer-0136994136	jacada.push.topic_auto_1	JMServer-0_auto_3
☐ JMServer-01932685921	dist_jacada.push.topic_auto	new_Cluster_1
New Delete	Showing 1 to 2 of 2 Previous Next	

- To deploy your JMS resources on the cluster, click the subdeployment that is targeting the cluster, and then select your cluster as a target. For example:

Settings for JMServer-01932685921

Click the **Lock & Edit** button in the Change Center to modify the settings on this page.

Save

Use this page to select the targets for the subdeployment. You can reconfigure targets later if you wish.

Servers

☐ AdminServer

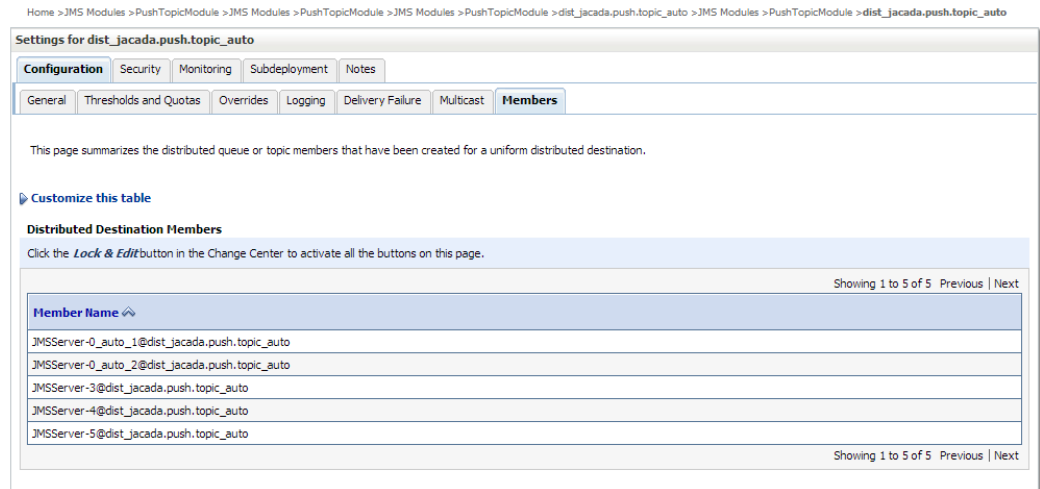
Clusters

☒ new_Cluster_1

- ☒ All servers in the cluster
- ☐ Part of the cluster
 - ☐ new_ManagedServer_2
 - ☐ new_ManagedServer_1
 - ☐ Server-4
 - ☐ Server-5
 - ☐ Server-3

Note: An error during targeting to the cluster indicates that you have duplicate targeting (to the cluster as well as to the individual servers). To remedy this situation, untarget to the individual servers, save, and then retarget to the cluster.

After performing this configuration, JMS resources needed for new JMS servers will automatically be deployed to the cluster as soon as a new JMS server is created. To view and verify destination members, navigate to *Services > Messaging > JMS Modules > PushTopicModule > dist_jacada.push.topic_auto*, and select the *Members* tab.



Creating Additional Server Domains

You may need to create additional WebLogic Server domains in the machine hosting Jacada WorkSpace. A domain provides the environment for running a particular WebLogic Server.

Creating WebLogic Server domains anytime after Jacada WorkSpace product installation requires that you run the Server Setup wizard. This process also configures the database schema to be used with the WebLogic Server.

To create a server domain for a single server instance of WebLogic Server:

1. **Linux RedHat operating system** – Run `./ServerSetupWizard.sh -i console`, located in the directory `<Jacada_WorkSpace_installation_directory>/Common/Utilities/ServerSetup`.
Windows operating system – Run `Start > Programs > Jacada > Jacada Work Space <version> > Server Setup Wizard`.
2. In the *Select Application Server* dialog, select *Oracle WebLogic 10.3.6*.
3. In the *Select Application Server Home Folder* dialog, select the WebLogic server home folder to use with Jacada WorkSpace.
4. In the *WebLogic Domain Server Details* dialog, in the *Admin server address* field, select the relevant IP address of your server machine in the *Listen address* dropdown list. The default assignment for both *Admin server port* and *Admin server SSL port* display in their respective

fields.

Note: Make sure that any port you assign, whether it be a default setting or not, is available in the host machine.

5. In the *Administrator User Details* dialog, specify the user name and password that will be used to log on to the Oracle WebLogic Console for the domain being defined.
6. In the *Domain Details* dialog, specify the name you want for the domain.
7. In the *Database Details* dialog, select the required database from the drop-down list and dependent on the database, enter the other details.
8. In the *Schema Details* dialog, specify the name and password you want for the domain. The default schema name is the same as the domain name.

After a processing period of several minutes, the new WebLogic Server domain is created.

Note: In a single server instance of WebLogic Server, the server instance functions as both the Administration Server, `AdminServer`, and the Managed Server.

Creating additional WebLogic Server cluster domains anytime after Jacada WorkSpace product installation requires that you run the Oracle WebLogic Configuration Wizard. For details, refer to [Configuring the WebLogic Server Cluster](#).

4

Installing Jacada WorkSpace for Production: Tomcat

The following topics describe how to install Jacada WorkSpace in an Apache Tomcat server production environment, in order to deploy an Agent WorkSpace:

- [Production Requirements](#)
- [Installation Overview](#)
- [Installation of Jacada WorkSpace](#)
- [Installing a Clustered Production Environment](#)
- [Load Balancing the Tomcat Cluster](#)
- [Setting Up Licensing](#)
- [Additional Tomcat Configuration Parameters for Production](#)
- [Configuring the Apache Web Server](#)
- [Setting Up CTI](#)
- [Setting Up the Agent Desktop](#)
- [Installing the EMCGateway for Avaya Elite Multichannel](#)

Production Requirements

This section describes the requirements for a Jacada WorkSpace production environment.

Agent Desktop Production Requirements

Jacada WorkSpace is certified with the following configuration for the call center agent desktop:

Certified Configuration

Operating System	Microsoft Windows XP Professional with Service Pack 3 Microsoft Windows 7, Enterprise edition (32 bit and 64 bit)
Display Resolution	1280x1024 (minimum) 1366x768 and above (recommended)
Browser	• Microsoft Internet Explorer version 9 (32 bit and 64 bit) • Microsoft Internet Explorer version 10 (32 bit and 64 bit) • Microsoft Internet Explorer version 11 (32 bit and 64 bit) • Google Chrome (without hosting)

Windows Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Windows server:

Certified Configuration for Windows

Minimum Hardware	CPU	Intel Xeon 2 x 2.4GHz
	RAM	2 GB (4 GB recommended)
Software	Operating System*	Windows 2012 Server R2 64 bit
	Application Server	Apache Tomcat server 7.0.28 with Sun JDK 7 update 11 and Apache ActiveMQ to provide the JMS. The Apache components are installed with Jacada WorkSpace.
	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0

Certified Configuration for Windows

		- Genesys Telecommunications Laboratories Inc. – Genesys 8.0 and 8.1 Both vendors are supported with the following switch: - Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	- Oracle 11g R2 - SQL Server 2008 - MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.10 (32 bit)

*** You can work with Windows under VMware ESX.**

Linux RedHat Server Production Requirements

Jacada WorkSpace is certified with the following configuration for a Linux RedHat server:

Certified Configuration for Linux RedHat

Minimum Hardware	CPU	2 x Xeon E5420 (2.5 GHz) Quad Core
	RAM	2 GB (4 GB recommended)
Software	Operating System	Linux RedHat Enterprise 6.6 (64 bit)
	Application Server	Apache Tomcat server 7.0.28 with Sun JDK 7 and Apache ActiveMQ to provide the JMS. The Apache components are installed with Jacada WorkSpace.

Certified Configuration for Linux RedHat

	Telephony	Providers • Avaya AES 6.3 • Avaya Elite Multichannel 6.3 • Cisco Systems, Inc. – Cisco UCCE 9.0 • Genesys Telecommunications Laboratories Inc. – Genesys 8.0 and 8.1 Both vendors are supported with the following switch: • Avaya S8700 (G650 Media gateway with ACM 6.3 CVLAN)
	Database	• Oracle 11g R2 • SQL Server 2008 • MySQL 5.5_29
	Web Server	Apache Web Server version 2.2.15

Installation for Production - Overview

Jacada WorkSpace projects are deployed to Tomcat. The supported application server configurations for a production environment are, as follows:

- A clustered server environment: Used for large-scale production environments. The cluster server environment provides failover capabilities.
- A single server instance: Applicable to small-scale production environments, product testing and product staging/controlled roll-out.

Installation Sequence

Jacada requires the following installation sequence:

Server Environment	Installation Sequence
Tomcat Cluster	1. Install Jacada WorkSpace 7.3. 2. Execute the Jacada Server Setup wizard for each server in the cluster.
Tomcat Single Server	1. Install Jacada WorkSpace 7.3. 2. To set up the server, execute the Jacada Server Setup wizard.

Installation of Jacada WorkSpace

The Jacada WorkSpace installation includes the following activities:

- Installation of Jacada WorkSpace.
- Creation of an Apache Tomcat server. The process also configures the database schema to be used with the Tomcat server.

Note: Before beginning the installation procedure, verify that the name of your computer does not contain an underscore character.

To install Jacada WorkSpace:

1. In the folder containing the Jacada WorkSpace installation, do one of the following:
 - If the production server is going to run under the Linux RedHat operating system, execute **/setup.bin -i console**.
 - If the production server is going to run under one of the Windows operating systems, run the **setup.exe** file.
2. Follows the installation instructions until the *Setup a Server* window displays.
 - When installing on Linux RedHat, this window is skipped.
 - When installing on a Windows operating system, select the Yes option and click Next. After a brief processing period, the Server Setup Wizard Introduction window displays. To create a new Tomcat server and define the database details of the database that your Tomcat Application Server will use, continue with the next procedure: [To run the server setup](#).

3. When installing on either a Linux RedHat operating system, when the *Install Complete* window displays, click *Done*.
 - To create a new Tomcat server and define the database details of the database that your Tomcat Application Server will use, continue with the next procedure: [To run the server setup](#).

To run the server setup:

1. Access the Server Setup Wizard:
 - **Linux RedHat operating system:** Run `./ServerSetupWizard.sh -i console`, located in the directory `<Jacada_installation_directory>/Jacada WorkSpace <version>/Common/Utilities/ServerSetup`.
 - **Windows operating system:** Run *Start > Programs > Jacada > Jacada Work Space <version> > Server Setup Wizard*.
 2. In the *Select Application Server* dialog, select the *Apache Tomcat* radio button.
 3. In the *Tomcat Details Panel* dialog:
 - a. Enter the home folder to install the Apache Tomcat server.
 - b. Enter the server HTTP port.
 - c. Enter the server JMX port.
 4. In the *Database Details* dialog, select the required database from the drop-down list and dependent on the database, enter the other details.
 5. In the *Schema Details* dialog, specify the name and password you want for the Apache Tomcat server.
 6. In the *Jacada Dynamic Views Schema Details* dialog, specify the name and password you want for the Apache Tomcat server.
 7. In the *Jacada Dynamic Views Administrator User Details* dialog, specify the name and password of the administrator.
- After a processing period of several minutes, a new Apache Tomcat server is created.

Installing a Clustered Production Environment

In a production environment, Jacada WorkSpace is typically deployed in a cluster of application servers. A cluster is a deployment in which multiple Tomcat server instances are used to process requests. From the client's point of view, there is only one instance of Tomcat.

Configuring a cluster environment involves changing port numbers in the following locations:

- Tomcat XML files:
 - Server.xml
 - Context.xml
- ActiveMQ
- JMX

For details, refer to [Tomcat Cluster Configuration](#).

Tomcat Cluster Configuration

Configuring a cluster environment involves changing port numbers in the following locations:

- Tomcat XML files:
 - **Server.xml**: For more information about this file, go to <http://tomcat.apache.org/tomcat-7.0-doc/config/index.html>
 - **Context.xml**: For more information about this file, go to <http://tomcat.apache.org/tomcat-7.0-doc/config/context.html>
- **ActiveMQ**: This message broker, which fully implements the Java Message Service, is installed by Jacada together with the Tomcat server. for more information about ActiveMQ, go to <http://activemq.apache.org/getting-started.html>
- **JMX**: A unique port is required to support Java Management Extension requests.

The following sections explain the procedures for cluster configuration in each of these elements. All of these procedures need to be done for **each** member of the cluster. After performing the procedures, you can start the cluster and work on it with the load balancer, without Apache.

Server.xml

Cluster configuration involves changing connector ports for HTTP and AJP protocols. The following port number convention should be used:

- **Tomcat1**: Use port numbers with a prefix of 8 (e.g., 8080, 8009).
- **Tomcat2**: Use port numbers with a prefix of 9 (e.g., 9080, 9009).

The following ports need to be changed:

- Tomcat listen port:
<Connector **port="9080"** protocol="HTTP/1.1" redirectPort="9443"/>

Note: To support HTTPS, add the following parameter at the end of the line:

<Connector **port="9080"** protocol="HTTP/1.1" redirectPort="9443" **scheme="https"**/>

- <Connector **port="9009"** protocol="AJP/1.3" redirectPort="9443"/>
- <Server **port="9005"** shutdown="SHUTDOWN">

In cluster configuration, for each server in the cluster, add the `jmvRoute` attribute to the `Engine` node in the **server.xml** file. The `jmvRoute` value must be unique for each Tomcat instance.

```
<Engine defaultHost="localhost" name="Catalina" jmvRoute="tomcat1">
```

Context.xml

Each JMS client connects to the broker on the localhost. The network of brokers distribute the messages to the particular client.

For cluster configuration, make the following changes in the `Context.xml` file:

- Set your IP address instead of `localhost`.
- The port for `brokerURL` should be the same as the port in the **activemq.xml**.

```
<Resource Password="manager" Username="system" auth="Container"
brokerName="LocalActiveMQBroker" brokerURL="tcp://<IP of Tomcat server>:61616"
description="JMS Connection Factory" factory="org.apache.activemq.jndi.JNDIRefere
name="XAConnectionFactory" type="org.apache.activemq.ActiveMQConnectionFactory" u
```

ActiveMQ

For cluster configuration, the ActiveMQ broker needs to be configured to accept requests on different sockets. The following port number convention should be used:

- **Tomcat1:** Use port numbers with a prefix of 5 (e.g., 51616, ...).
- **Tomcat2:** Use port numbers with a prefix of 6 (e.g., 61616, ...).

Any port in the range 3000-65000 may be used.

Note: Ports changed here must also be changed in the **Context.xml** file.

The network of brokers is used to distribute messages across the cluster. Each broker uses

multicast to publish the broker's address and port. When changing a multicast port, take care not to disturb others on the net.

To change a multicast port, make the changes indicated below in the **`${catalina.base}/webapps/activemq-web-console-5.7.0/WEB-INF/activemq.xml`** file. Make sure to update the multicast port in all servers in the cluster.

```
<transportConnectors>
  <transportConnector name="nio" uri="nio://<IP of Tomcat server>:51616"
    updateClusterClients="true"
    updateClusterFilter="web-console"
    updateClusterClientsOnRemove="true"
    discoveryUri="multicast://239.255.12.6:5155?group=activemq"
  </transportConnector>
</transportConnectors>
<networkConnectors>
  <networkConnector name="workspace" uri="multicast://239.255.12.6:5155?group=activemq"
  </networkConnector>
</networkConnectors>
```

JMX Connection

A unique port for JMX requests needs to be opened for each server in the cluster.

For cluster configuration, edit the following property in the **`${catalina.base}/Jacada/conf/management.properties`** file:

- `com.sun.management.jmxremote.port=8099`

Changing the Database Password

Changing the database password involves updating properties in the **`JacadaConfiguration.properties`** file, and then restarting the server.

To change the database password:

1. In the **`<Tomcat_home>\jacada\conf\JacadaConfiguration.properties`** file, update the following properties:

```
db.dv.password=<Dynamic Views Schema Password>
db.ws.password=<WorkSpace Schema Password>
```

2. Restart the server.

Load Balancing the Tomcat Cluster

In a typical cluster configuration, a load balancer is configured as a front-end to the Tomcat server cluster. For the call center agent client machines, the load balancer is the single access point to the cluster and the request distributor to the Tomcat cluster members.

The BIG_IP Local Traffic Management System

Jacada recommends deploying the F5 Network BIG-IP Local Traffic Management (LTM) system for directing traffic to the Tomcat cluster members running Jacada WorkSpace. A BIG-IP LTM configuration consists of defining the following components:

- A virtual server (the traffic manager)
- A connection pool (a set of devices grouped together to receive traffic according to a load balancing method) associated with a virtual server
- The connection pool members (the Tomcat cluster members)

In order for Jacada WorkSpace traffic to properly travel between the agent client machines, the LTM virtual server and the Tomcat cluster members, configure the following properties and resources in the BIG-IP LTM Configuration utility:

Default Persistence Profile – Set this resource to `cookie`. The LTM virtual server uses cookies to assign each agent client machine to a specific Tomcat cluster member in the connection pool. This setting instructs the virtual server to always direct requests from an agent client machine to the same Tomcat cluster member running Jacada WorkSpace. Access the resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.

HTTP Profile – Set this property to `http`. Access the property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

Protocol Profile (Client) and Protocol Profile (Server) – Set both these properties to the custom tcp profile of your choice. The assigned tcp profile must have its *Nagle's Algorithm* setting unchecked. By disabling this property, you prevent unnecessary TCP/IP delay of small packets during their routing. Access these properties in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Properties > Configuration > Advanced* properties.

SSL Profile (Client) – Set this property to `clientssl`. This setting instructs the virtual server that communication with any agent client machines is always encrypted. Access this property in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> >*

Properties > Configuration > Advanced properties.

Health Monitors – For the *Default Pool* resource of the LTM virtual server, assign as active a monitor of type TCP to its *Health Monitors* property. The assigned monitor must have its *Interval* configuration property set to 30 seconds and its *Timeout* configuration property set to 91 seconds. These resource and property configurations cause the LTM to monitor the TCP connection status of its application server pool members, when load balancing new connection requests to the cluster of servers.

- Access the *Default Pool* resource in the *Main Tab > Local Traffic > Virtual Servers > Virtual Server List > <virtual_server_name> > Resources > Load Balancing* resources.
- Access the *Health Monitors* property in the *Main Tab > Local Traffic > Pools > Pool List > <pool_name> > Properties > Configuration > Basic* properties.
- Access the *TCP monitor Interval and Timeout* properties in the *Main Tab > Local Traffic > Monitors > Monitor List > <monitor_name> > Properties > Configuration > Basic* properties.

Setting Up Licensing

Obtain a Jacada WorkSpace license file, `license.jacada`, from your Jacada representative. In order to correctly establish your Jacada WorkSpace licensing, place this license file under `<TOMCAT_HOME>\jacada\conf`.

Additional Tomcat Configuration Parameters for Production

The following sections explain parameters that need customized tuning for the production environment:

- Heap Size
- Number of Threads
- Size of JDBC Pools

Heap Size

The heap size value can be found in the `bin` directory, in the `catalina.sh` file, under `JACADA_OPTS`. Change the heap size to suit your production environment.

Number of Threads

The number of threads is determined by the `maxThreads` value, which is found in the **conf\server.xml** file. The recommended `maxThreads` value is 2x the number of users, per server instance.

Size of JDBC Pools

Pool size is determined by the `JACADA_JDBC_DB` value, which is found in the **conf/context.xml** file. If required, change pool sizes suit your production environment.

Configuring the Apache Web Server

The following sections describe the configuration necessary for Apache-Tomcat integration.

Tomcat Configuration

In the **conf/server.xml** file, add the `jvmRoute` attribute that identifies each tomcat instance, for example, `tomcat1`, `tomcat2`.

```
<Engine defaultHost="localhost" name="Catalina" jvmRoute="tomcat1">
```

Apache Configuration

The following lines should be added to the **httpd.conf** file:

```
ProxyPass ((.gif|.js|.css)$) !
ProxyPass / balancer://mycluster/ stickysession=JSESSIONID nofailover=On
ProxyPassReverse / balancer://mycluster/
<Proxy balancer://mycluster>
BalancerMember ajp://<ip of server machine>:<ajp port> route=tomcat1 ttl=30
BalancerMember ajp://<ip of server machine>:<ajp port> route=tomcat2 ttl=30
ProxySet lbmethod=bytraffic
</Proxy>
```

5

Configuring Jacada WorkSpace in the Database

Jacada WorkSpace relies on a database that contains various configuration options. Most of this configuration is performed using the Jacada Management Console. Some of the configuration is done directly in the database. The following topics describe the configuration performed in the database:

- [Database Tables](#)
- [Management of Audited Entities](#)
- [CTI Tables](#)

Database Tables

Jacada WorkSpace relies on a database that contains various configuration options for the product. Refer to the system requirements for a list of supported databases in a production environment.

The following table describes the Jacada WorkSpace database tables that are administered using the database management system. (For a fuller description of these tables, refer to [Management of Audited Entities](#).) All other tables are administered using the [Jacada Management Console](#), or are for internal use.

Note: The database tables should be modified only by using the Jacada Management Console, unless otherwise specified within the documentation.

Jacada WorkSpace Database Tables

Table	Description
AUDIT_ACTION_PARAMS	Contains relevant, parameter information associated with each audited agent action event.
AUDIT_ACTIONS	Contains information about each audited, agent action event performed either per agent session event, per agent interaction event or per agent process event.
AUDIT_AGENT_SESSION_ACTIONS	Contains entries that couple each audited agent action event with its parent Jacada WorkSpace session event.
AUDIT_AGENT_SESSION_PROCESSES	Contains entries that couple each audited, agent process event with its parent Jacada WorkSpace session event.
AUDIT_AGENT_SESSION	Contains information about each audited Jacada WorkSpace session event.
AUDIT_INTERACTION_ACTIONS	Contains entries that couple each audited agent action event with its parent agent interaction event.
AUDIT_INTERACTION_PROCESSES	Contains entries that couple each audited agent process event with its parent agent interaction event.
AUDIT_INTERACTIONS	Contains information about each audited agent interaction event.
AUDIT_PROCESS_ACTIONS	Contains entries that couple each audited agent action event with its parent agent process event.
AUDIT_PROCESS_PARAMS	Contains relevant parameter information associated with each audited, agent process event.

Table	Description
AUDIT_PROCESSES	Contains information about each audited agent process event performed either per agent session event or per agent interaction event.
CTI_SETTINGS	Contains CTI profiles. A CTI profile contains the configuration that is necessary for working with a specific CTI provider and switch combination.

Management of Audited Entities

When database auditing is enabled, a Jacada WorkSpace project records information for audited entities in the following tables:

- [AUDIT_AGENT_SESSIONS](#) Table
- [AUDIT_INTERACTIONS](#) Table
- [AUDIT_PROCESSES](#) Table
- [AUDIT_PROCESS_PARAMS](#) Table
- [AUDIT_ACTIONS](#) Table
- [AUDIT_ACTION_PARAMS](#) Table
- [Tables of Child-Parent Relationships between Audited Events](#)

The *Agent Session*, *Interaction*, *Process* and *Action* represent the different types of Jacada WorkSpace entities that can occur. Both *Process* and *Action* events can have associated event parameters.

AUDIT_AGENT_SESSIONS Table

The AUDIT_AGENT_SESSIONS table contains entries of the agent's Jacada WorkSpace *Session* events. Such an event represents a complete agent session from login through logout. Each AUDIT_AGENT_SESSIONS table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this agent session event.
AGENT ID	The user name used by the agent to log in to a Jacada WorkSpace session.
PROJECT ID	The name of the deployed Jacada WorkSpace project.
SESSION ID	The Jacada WorkSpace session ID.

AUDIT_INTERACTIONS Table

The `AUDIT_INTERACTIONS` table contains entries of the agent's *Interaction* events, being audited. Such an event represents the interaction between an agent and a client, for example, during a telephone conversation, a fax correspondence or an e-mail correspondence. An agent interaction event occurs within the context of an agent session event. Each `AUDIT_INTERACTIONS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this agent interaction event.
INTERACTION TYPE	The type assigned to this agent interaction event. Audited interaction types are: <i>Inbound</i> , <i>Outbound</i> , <i>Manual</i> , <i>Internal</i> and <i>Custom Item</i> .
INTERACTION NUMBER	- For interaction events of types <i>Inbound</i>, <i>Outbound</i>, <i>Manual</i> and <i>Internal</i>, this number is the telephone number of the call. - For interaction events of type <i>Custom Item</i>, this number is the Genesys Interaction ID.
START TIME	The date and time when the interaction event started, in the format YYYY-MM-DD HH:MM:SS.

Column	Description
END TIME	The date and time when the interaction event ended, in the format YYYY-MM-DD HH:MM:SS.
AGENT_SESSION_ID	The ID assigned to the agent session event, which is the parent event of this interaction event.

AUDIT_PROCESSES Table

The `AUDIT_PROCESSES` table contains entries of the agent's *Process* events, which are enabled for audit. Such an event represents a Jacada WorkSpace business process initiated by the agent that occurs within the context of either an agent session event or an agent interaction event. Each `AUDIT_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS_ID	The ID assigned to this agent process event.
PROCESS_NAME	The name given to this agent process event. Process names are unique within a Jacada WorkSpace project.
START_TIME	The date and time when the process event started, in the format YYYY-MM-DD HH:MM:SS.
END_TIME	The date and time when the process event ended, in the format YYYY-MM-DD HH:MM:SS.

AUDIT_PROCESS_PARAMS Table

The `AUDIT_PROCESS_PARAMS` table contains entries of process parameters for those *Process* events enabled for audit. A process parameter provides operational information that is integral to the *Process* event to which it is associated. For example, `Chat`, a Jacada WorkSpace-provided event

available for audit, is a *Process* event and has the following process parameters: *Initiator* and *Recipient*. Each table entry stores audit information about a single *Process* event parameter. Each `AUDIT_PROCESS_PARAMS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this process parameter.
NAME	The name given to this process parameter, e.g., <i>Initiator</i> . Process parameter names are unique within a Jacada WorkSpace project.
VALUE	The information content of the process parameter.
PROCESS ID	The ID assigned to the agent process event, which is the parent event of this process parameter.

AUDIT_ACTIONS Table

The `AUDIT_ACTIONS` table contains entries of the agent's *Action* events, which are enabled for audit. Such an event represents an operation performed by the agent that occurs within the context of either an agent session event, an agent interaction event or an agent process event. Each `AUDIT_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to this agent action event.
ACTION NAME	The name given to this agent action event. Action names are unique within a Jacada WorkSpace project.
ACTION TIME	The date and time when the action event occurred, in the format YYYY-MM-DD HH:MM:SS.

AUDIT_ACTION_PARAMS Table

The `AUDIT_ACTION_PARAMS` table contains entries of action parameters for those *Action* events enabled for audit. An action parameter provides operational information that is integral to the *Action* event to which it is associated. For example, `warmTransferClicked`, a Jacada WorkSpace-provided event available for audit, is an *Action* event and has the following action parameter: `dn`. Each table entry stores audit information about a single *Action* event parameter. Each `AUDIT_ACTION_PARAMS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this action parameter.
NAME	The name given to this action parameter. For example, <code>dn</code> (dialed number). Action parameter names are unique within a Jacada WorkSpace project.
VALUE	The information content of the action parameter.
ACTION ID	The ID assigned to the agent action event, which is the parent event of this action parameter.

Tables of Child-Parent Relationships between Audited Events

Database Auditing uses the following tables to record, and thereby manage, the child-parent relationship that exists between audited events. A many-to-one association can exist for a parent event, that is, many child events having one parent event. Each child event is uniquely associated with its parent event and never appears in any other relationship table.

The following sections describe:

- [AUDIT_AGENT_SESSION_PROCESSES Table](#)
- [AUDIT_AGENT_SESSION_ACTIONS Table](#)
- [AUDIT_INTERACTION_PROCESSES Table](#)
- [AUDIT_INTERACTION_ACTIONS Table](#)
- [AUDIT_PROCESS_ACTIONS Table](#)

AUDIT_AGENT_SESSION_PROCESSES Table

The `AUDIT_AGENT_SESSION_PROCESSES` table contains entries that identify the child-parent relationship that exists between an agent process event (the child) and the agent session event (the parent), within whose context the agent process event occurred. Each `AUDIT_AGENT_SESSION_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS ID	The ID assigned to an agent process event.
PARENT ID	The ID assigned to the agent session event, which is the parent event of this process event.

AUDIT_AGENT_SESSION_ACTIONS Table

The `AUDIT_AGENT_SESSION_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent session event (the parent), within whose context the agent action event occurred. Each `AUDIT_AGENT_SESSION_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent session event, which is the parent event of this action event.

AUDIT_INTERACTION_PROCESSES Table

The `AUDIT_INTERACTION_PROCESSES` table contains entries that identify the child-parent relationship that exists between an agent process event (the child) and the agent interaction event (the parent), within whose context the agent process event occurred. Each `AUDIT_INTERACTION_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS ID	The ID assigned to an agent process event.
PARENT ID	The ID assigned to the agent interaction event, which is the parent event of this process event.

AUDIT_INTERACTION_ACTIONS Table

The `AUDIT_INTERACTION_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent interaction event (the parent), within whose context the agent action event occurred. Each `AUDIT_INTERACTION_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent interaction event, which is the parent event of this action event.

AUDIT_PROCESS_ACTIONS Table

The `AUDIT_PROCESS_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent process event (the parent), within whose context the agent action event occurred. Each `AUDIT_PROCESS_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent process event, which is the parent event of this action event.

CTI Tables

When CTI is used, the `CTI_SETTING` table is used to configure CTI. The configuration is done in the database and not via the Jacada Management Console. The `CTI_SETTINGS` table has the following structure:

Column	Description
ACTIVE	This boolean column specifies whether the entry is active. Specifying that an entry is not active (<code>false</code>) enables you to disable an entry without removing it from the table, for testing purposes for example.

Column	Description
PROJECT	An asterisk (*) specifies that the entry affects all projects. To have an entry affect a specific project (application) only, specify the context path. For example: /myAppJacada A specific project setting takes precedence when the same setting exists for all projects.
PROFILE	The name of the profile.
NAME	The name of the setting.
VALUE	The value of the setting.
COMMENTS	Comments that provide information on the setting.

Jacada WorkSpace CTI_SETTINGS Table

Jacada WorkSpace provides a `CTI_SETTINGS` table that includes pre-configured sample CTI profiles. Each supplied CTI profile contains settings for using a specific CTI provider, or CTI provider and switch combination, with a specific configuration. You can customize the profiles according to the required configuration, by modifying existing profiles, or adding new profiles.

The following table describes the settings in the `CTI_SETTINGS` table. In the table's `Values` column, mutually exclusive options are indicated by a list of values, each on a separate line, with no separating character. When the setting can contain a list of values, commas separate between the possible valid values for the list.

Setting	Values	Description
CTIAgentStatusDuringCallSetBySwitch	true/false	Specifies whether the agent status during a call is set by the switch, and thus should not be set by the Jacada WorkSpace application.

Setting	Values	Description
CTIAllowDialPadWindow	true/false	Enables agents to use the Dial Pad window, which is accessed via the Dial button in the CTI bar of Jacada WorkSpace. If this setting is disabled, the Dial Pad window is not available via the Dial button and no <i>Dial number...</i> link appears in the <i>Dial List</i>; thereby restricting agents from dialing numbers when using the call-handling buttons (though the <i>Dial Pad</i> portlet could still be available).
CTIAllowTransferFrom Consult	true/false	Specify whether to support transferring a call from a consultation. The transfer is the default transfer type. Note that this will only work when the switch supports transferring a call from a consultation.
CTIApplicationSetACW	true/false	Specify whether Jacada WorkSpace should handle changing the agent status to After Call Work (ACW). When <code>true</code>, Jacada WorkSpace handles changing the agent status. When <code>false</code>, the status is changed by the CTI or the switch.
CTIAttachedDataKey1, CTIAttachedDataKey2, ... (Genesys only)		The names of the attached data keys to use for filtering voice items. For each attached data key, you need an entry in the database.

Setting	Values	Description
CTIAttachedDataValue1, CTIAttachedDataValue2, ... (Genesys only)		The values of the attached data keys to use for filtering voice items. These values must correspond to the names specified in the <code>CTIAttachedDataKey1,2,...</code> settings.
CTIBackupPort (Genesys and Cisco only)		The port to use to connect to the backup CTI server.
CTIBackupServer (Genesys and Cisco only)		The name or IP address of a backup CTI server, in case the <code>CTIServer</code> fails.
CTIBarEnabled	true/false	Controls the presentation of the CTI bar in Jacada WorkSpace. True is the default value.
CTIBarUrl	SYSTEM/portlets/ CTI/ CTIController.jspf	Specifies the relative project path of the file containing the CTI bar.
CTICallAutoAnswer	true/false	Specify whether Jacada WorkSpace should automatically answer calls.
CTICallMode	inbound outbound	The mode of the Jacada WorkSpace application: receiving inbound calls or dialing outbound calls.

Setting	Values	Description
CTIChangeToManualOnCTI OperationFailure	true/false	Specify whether Jacada WorkSpace should change to manual mode upon the failure of a CTI operation. When set to <code>false</code>, Jacada WorkSpace attempts to continue working despite the failure. Note: In production, it is recommended to set the value of this parameter to <code>false</code>.
CTICiscoConnectionHeartbeat		The Cisco CTI provider connection heart beat.
CTIConfigManager ApplicationName (Genesys only)		The name of the Genesys application of type 'Config Manager', as defined in the Genesys Configuration Manager Environment (CME).

Setting	Values	Description
CTIConfigurationMode	Agent Global	Specify whether the configuration mode is global or agent-specific. For each CTI provider you specify one of the modes, as follows: AES: Agent CTConnect: Agent Cisco: Agent Genesys: Global For Agent mode, you need to configure agent CTI information in the Jacada Management Console. In the sample profiles provided by Jacada WorkSpace, the configuration mode is already configured.
CTICustomItemAutoAnswer (Genesys only)	true/false	Specify whether Jacada WorkSpace should automatically answer custom items.
CTICustomItemFilterClass (Genesys only)		The fully qualified name of the class used to implement a filter for custom items.
CTICustomItemNameList (Genesys only)		A comma-separated list of the names of the Genesys Open Media types, as defined in Genesys CME. For example: WorkItem, Fax

Setting	Values	Description
CTICustomItemSkillName (Genesys only)		The name of the Genesys skill that the agent must have in order to handle the Open Media types defined in <code>CTICustomItemNameList</code> .
CTIDefaultTransferType	cold handshake warm	Specify the default transfer type to use when clicking the <i>Transfer</i> button in Jacada WorkSpace.
CTIDefaultNotReadyReasonCode	1	The default <i>Not Ready Reason</i> code (CTI Aux code). This default code value is sent to the CTI system in situations where the agent's call handling status must be updated to <i>Not Ready</i> (Busy), however, the status update request has no <i>Not Ready Reason</i> code assigned to it.
CTIEnableCustomItems (Genesys only)	true/false	Set to <code>true</code> to enable Jacada WorkSpace to handle custom items.
CTIEnableReconnect (Genesys only)	true/false	Specifies whether to enable reconnecting to CTI after a lost connection is restored.

Setting	Values	Description
CTIGatewayShouldGenerateAgentStatusEvents	true/false	This setting is relevant only when the CTISwitchReportsAgentEvents setting is false. Specifies whether the CTI Gateway application should simulate agent status events, instead of the switch. When working with Genesys, Genesys is responsible for simulating agent status events, and this setting is not relevant. In the CTI_CTC_AVAYA profile, for example, the value of this setting is <code>true</code>.
CTILink		The name of the logical link, defined in the CTI server, that defines which CTI switch to work with.
CTILoginByPhone	true/false	Set to <code>true</code> when the agent needs to perform the login to CTI via the phone, instead of automatic login as part of logging in to Jacada WorkSpace. For details, refer to the Jacada WorkSpace Developer Guide.
CTIMultiSiteSupport	true/false	Specify whether to use multi-site support.
CTINumberOfAttachedDataKeys (Genesys only)	<numeric>	The total number of attached data keys to use for filtering voice items.

Setting	Values	Description
CTIOutboundAccessKey	<numeric>	A number that needs to be dialed before dialing an outbound call (such as 9).
CTIPassword		A password that corresponds to the <code>CTIUserName</code>. When the password is entered or edited via the Jacada Management Console, the password is automatically encrypted when added to the database table, and decrypted before its use.
CTIPort (Genesys and Cisco only)		The port to use to connect to the CTI server.
CTIProvider	manual genesys cisco ctconnect aes emc	The CTI provider in use.
CTIReportAllEventsToClient	true/false	When true, specifies that every CTI event that reaches the CTI Gateway is also forwarded to an external application.
CTIServer		The name or IP address of the machine that houses the CTI server.
CTIServerTimeoutInSec (Genesys only)	60	Specify, for the CTI server, the maximum period of time for receiving responses.

Setting	Values	Description
CTISwitchAgentStatusOnLogin	ready notready	Specifies the default agent status upon login.
CTISwitchAllowsHoldIn InternalCalls	true/false	Specify whether the switch allows holding a direct-dial (internal) call. The <i>Hold/Off Hold</i> button behaves accordingly.
CTISwitchAutoAnswer	true/false	Specify whether the switch is configured to automatically answer calls.
CTISwitchReportsAgentEvents	true/false	Specify whether the switch reports agent events to the CTI Gateway. Typically, the switch does report agent events. However, there are switches that do not report agent events, such as AVAYA.
CTISwitchSendsRepeatedAgent StatusEvents	true/false	Specify whether the switch sends repeated agent status events.
CTISwitchType	avaya alcatel	The name of the switch in use.
CTISupportTransferFrom Consult	true/false	Enable transferring a call during a consultation
CTITransferTypesList	cold, handshake, warm	A comma-separated list of the transfer types supported by the CTI provider and switch in use.

Setting	Values	Description
CTIUserName		A user with access to the CTI server. Used when the CTI provider is AES, or Genesys with a multi-site configuration.
CTIVoiceFilterClass (Genesys only)		The fully qualified name of the class used to implement a filter for voice items.

Specifying Agent CTI Information Using a Lookup Mechanism

Agent-related CTI information, such as the extension number or queue number, can be defined either in the Jacada Management Console (in the `CTI > Agents` table), or can be retrieved using a lookup mechanism.

When using a lookup mechanism, you need to write the code that retrieves the agent CTI information when the agent logs in to Jacada WorkSpace.

To use a lookup mechanism for retrieving agent CTI information:

- Add the necessary code to the login method in the following file:

```
<project>\WEB-INF\src\global\Global.app
```

You **must** add the code, including the initialization of the session variables, **before** the following line:

```
super.login(request);
```

The following session variables, that correspond to the columns in the `CTI_AGENT` table, are available for use when implementing a lookup mechanism:

Session Variable	CTI Agents	Description
agentName	USer name	This variable is automatically set to the agent name used to log in to Jacada WorkSpace.
ctiExtension	Extension number	The agent's telephone extension number.

Session Variable	CTI Agents	Description
ctiGroupId	Group ID	The ID of the group (queue) the agent belongs to.
ctiOutboundExtension	Outbound extension number	The telephone extension number to use for outbound calls.
ctiPassword	CTI password	The password that corresponds to the ctiUserName.
ctiPhoneID	Agent ID	A logical ID for an agent (CTI PHONEID).
ctiProfileName	CTI profile	The name of the CTI profile used by the agent.
ctiUserName	CTI login name	When required by the CTI server, the user name for logging in to the CTI server.

Configuring Authentication and Authorization

Jacada WorkSpace includes project authentication and authorization. *Authentication* is the control of user login access to Jacada WorkSpace. *Authorization* is the control of agent and supervisor utilization of Jacada WorkSpace's features.

The following topics describe authentication and authorization:

- [Overview](#)
- [Authorization in Jacada WorkSpace](#)
- [Configuring Settings for Authentication](#)
- [Configuring Authorization](#)

Overview

Jacada WorkSpace includes project authentication and authorization to enable or restrict access to Jacada WorkSpace – authentication – and use of its features – authorization.

During development, both authentication and authorization are implemented based on information in the Jacada Management Console. In a production environment, an LDAP server is used to authenticate users, while authorization is still implemented using the Jacada Management Console.

Jacada WorkSpace has the following settings for authenticating user login, set in the Jacada Management Console:

- Standalone: The information defined in the Jacada WorkSpace database is used to authenticate user login. This setting is used during development.
- External: An external LDAP server is used to authenticate user login. Users and groups are owned and managed by the LDAP server. Jacada WorkSpace imports this information from the LDAP server. This setting is used during production.

The following table describes the entities used by a project to implements authentication and authorization. These entities are set in the Jacada Management Console. For details, refer to

[Authentication and Authorization.](#)

Entity	Description/Notes
User	A user login (name and password) that undergoes authentication to access a Jacada WorkSpace session. Once a session is established, each user's ability to work with Jacada WorkSpace, i.e., their roles and privileges, is determined by their group affiliations. Users are assigned to one or more Groups.
Group	A set containing one or more Users. Additionally, each group has a profile consisting of one or more of the following assignments: • Roles: The roles assigned to a group determine the usage rights for the users in the group, via privileges. • Supervisors of the group: Specific supervisors from the pool of supervisors who supervise this group. • Expert groups: Whether the group, and thus the users in the group, are experts in the contact center. This enables individuals groups to be defined which can be treated differently using APIs. For details about the APIs, refer to the Jacada WorkSpace Developer Guide. • Peer groups: Groups in which users have similar status. This enables individual groups to be defined which can be treated differently using APIs. For details about the APIs, refer to the Jacada WorkSpace Developer Guide.
Privilege	The usage rights defined for a feature. Privileges are specified in the project for each feature, for example, access to a specific tab/portlet or control in a JSP page, such as a button or label. If a privilege is not specified for a feature, there is no restriction on any agent using the feature. Privileges are assigned to Roles.

Entity	Description/Notes
Role	A set containing one or more Privileges. Each privilege assigned to a role can be either enabled or disabled. When the privilege is enabled, the user has full access to the specific feature. When the privilege is disabled, the user has read-only access to the specific feature. Note: Enabling or disabling a privilege applies only to input and selection controls. If the privilege is disabled on a Label control, the label text is not displayed.
Skill	A set defining one or more areas of expertise related to the customer's business and the Jacada WorkSpace project. Skills are assigned to Groups.

Authorization in Jacada WorkSpace

Jacada WorkSpace provides a base set of users, groups, roles and privileges.

Entity	Value
Users	guest, workspace (both with password workspace)
Groups	Agents, CTIUsers, Supervisors
Roles	AgentRole, CTIUserRole, SupervisorRole
Privileges	Jacada WorkSpace provides privileges for the various features of Jacada WorkSpace, including portlets, menu options, CTI options and tabs. For a list of the Jacada WorkSpace-provided privileges, refer to Privileges .

The user `guest` is authorized to use basic Jacada WorkSpace features. The user `workspace` is authorized to use the entire range of Jacada WorkSpace features. Details about which groups, roles

and privileges are assigned to these users can be seen in the Jacada Management Console.

To assign privileges to users:

1. In the Jacada Management Console, select the *Configuration > Authentication-Authorization* option.
2. In the *Users* option, assign a valid group to the Jacada WorkSpace user.
3. In the *Groups* option, assign valid roles to the group.

Note: Peer groups, supervisor users and expert groups can also be assigned to a group.

4. In the *Roles* option, assign valid privileges to the roles.
5. In the *Roles* option, edit the assigned privileges per role and enable them.

To review a role's assigned privileges:

1. In the Jacada Management Console navigation menu, select the *Configuration > Authentication-Authorization > Roles* option.
2. In the *Roles* page, click the *Edit* link of a role listed in the displayed table.

Configuring Settings for Authentication

For general configuration of authentication, select the *Configuration > Authentication-Authorization > Settings* option in the Jacada Management Console. In the *Settings* page, configure the following fields:

- HTTPS Port
- Security Mode – Defines the information source that Jacada WorkSpace uses to authenticate users logging into Jacada WorkSpace. If Standalone is defined, Jacada WorkSpace uses the Jacada WorkSpace database to authenticate users. If External is defined, Jacada WorkSpace uses an LDAP server to authenticate users.

Note: When transitioning from using `Standalone` mode to using `External` mode, all users and groups must be deleted from the database. Perform this activity using the Jacada Management Console.

If the database is being used to provide user authentication, the administrator must define, for authentication purposes, users in the Jacada Management Console ([Administering Users](#)). If an LDAP server is being used to provide user authentication, groups and users are owned and managed by the LDAP server. The administrator must define LDAP related information in the Jacada Management Console.

- *Server Administrator Name* and *Server Administrator Password* – Do not edit these fields. In a single application server environment, this information is automatically populated in the database during the Jacada WorkSpace installation. In a cluster environment, these fields are not relevant.

Configuring Settings for LDAP Authentication

When an LDAP server is being used to provide user authentication for Jacada WorkSpace, the administrator is required to perform the following activities, using the Jacada Management Console:

- Define the LDAP settings that enable Jacada WorkSpace to access the LDAP server and to search for and retrieve user and group related information.
- Import the relevant LDAP groups into Jacada WorkSpace. Jacada WorkSpace recognizes imported groups as its own.

To define LDAP settings:

1. In the Jacada Management Console navigation menu, select the *Configuration > Authentication-Authorization > LDAP > Settings* option.
2. In the *Settings* page, configure all the fields.
3. To test the connection to the LDAP server, click the *Test Connection* button on the *Settings* page.

To import relevant LDAP groups into Jacada WorkSpace, select the *Configuration > Authentication-Authorization > LDAP > Groups* option in the Jacada Management Console navigation menu. The *Groups* page displays only those LDAP groups that have not yet been imported into Jacada WorkSpace. In the *Groups* page, select the checkboxes of those LDAP groups that you want to import into Jacada WorkSpace, and then click *Import*.

Configuring Authorization

User authorization of Jacada WorkSpace features requires that the following entities are defined:

- Privileges
- Roles
- Groups
- Users

The following authorization entities are optionally defined:

- Peer groups
- Supervisors
- Expert groups
- Skills

Administering Privileges

A privilege is the right to use a specific Jacada WorkSpace feature or set of features. Privileges are assigned to roles. Privileges are managed via the *Configuration > Authentication-Authorization > Privileges* option in the Jacada Management Console. The *Privileges* page enables the following:

- Adding a privilege (defining its name and describing it)
- Deleting a privilege

Note: A privilege assigned to one or more roles cannot be deleted.

- Editing a privilege (updating its description)

Administering Roles

A role is assigned one or more privileges. Roles are assigned to groups. Roles are managed via the *Configuration > Authentication-Authorization > Roles* option in the Jacada Management Console navigation menu. The *Roles* page enables the following:

- Adding a role, which includes defining its name, assigning privileges to the role and enabling or disabling any of the role's assigned privileges. By default, assigned privileges are enabled.

Note: Enabling or disabling a privilege only applies to input and selection controls in a JSP page, defined using specific Jacada tags. For details, refer to the Jacada WorkSpace Developer Guide.

- Deleting a role.

Note: A privilege assigned to one or more roles cannot be deleted.

- Editing a role, which includes updating the role's assigned privileges and updating the enabled or disabled status of any of the role's assigned privileges.

Privileges, Roles, and User Authorization

In project code, privileges are used with features in order to support authorization of their use. The relationship between privileges, roles and user authorization is as follows:

Condition	User Authorization
Role X is assigned the privilege required for the use of the feature. This condition is relevant to the following features: menu, menu item, portlet, tab and toolbar button.	The user, who is assigned role X, has full use of the feature.

Condition	User Authorization
Role X is assigned the privilege required for the use of the feature and the privilege per role is enabled. This condition is relevant to the following features: button, checkbox, checkbox group, combo box, label, radio button, radio button group, text input field and text input area.	The user, who is assigned role X, has full use of the feature.
Role X is assigned the privilege required for the use of the feature and the privilege per role is disabled. This condition is relevant to the following features: button, checkbox, checkbox group, combo box, label, radio button, radio button group, text input field and text input area.	The feature is displayed in Jacada WorkSpace; however, the user, who is assigned role X, cannot use it.
Role X is not assigned the privilege required for the use of the feature.	The feature is not available to the user, who is assigned role X
No privilege is required for the use of the feature.	The user, regardless of assigned role, has full use of the feature.

Roles with Special Functions

The following Jacada WorkSpace provided roles have special functions:

- **SupervisorRole:** This role must be assigned to any group that you want to designate as a group of supervisors. All users assigned to a group with this role can be assigned as supervisors. Users who are reassigned tasks based on a specified role must also be assigned the SupervisorRole role.
- **AgentRole:** In order for a user to be able to log in and authenticate, the user must have this role. The AgentRole must be assigned to every group to enable the group's assigned users to log in to Jacada WorkSpace.

Jacada recommends having a maximum of 5% of all users as supervisors. If you want to assign supervisor privileges to more than 5% of all users, create a new role, assign it the relevant privileges,

and then assign the users to this role.

Administering Groups

The group is the bridge of authorization; it joins authenticated users together with roles. Multiple users and multiple roles can be assigned to a group. For details about user assignment to groups, refer to [Administering Users](#). Additionally, a group can be defined with one or more of the following assignments:

- Supervisors of the group
- Peer groups
- Expert groups, which can include groups assigned one or more skills

Groups are managed via the *Configuration > Authentication-Authorization > Groups* option in the Jacada Management Console navigation menu. The *Groups* page enables the following:

- Adding a group, which includes:
 - Defining its name
 - Assigning it peer groups and expert groups
 - Assigning it roles

Note: `AgentRole` and `SupervisorRole` roles have [special functions](#).

- Assigning supervisors of the group

Note: Jacada WorkSpace requires a large amount of processing resources to prepare the *Supervisors* and *Experts* lists of instant messaging contacts per user logging in. Based on your total agent population, limit the total number of supervisor users to 5% and expert users to 3%.

- Unassigning all users from the group

When an LDAP server is being used to provide user authentication, adding a group is not available in the *Groups* page, since groups and users are owned and managed by the LDAP server. Jacada WorkSpace imports its groups from the LDAP server ([Configuring Settings for LDAP Authentication](#)).

- Deleting a group – A group cannot be deleted if any of the following conditions are true:
 - Users are assigned to it
 - It is assigned as a peer group to another group
 - It is assigned as an expert group to another group
- Setting Group Search Order – Define the order in which Jacada WorkSpace searches the groups to find a user's supervisor. Groups are searched, one by one, from the top entry in the list to its bottom entry until a supervisor is found, selected and the search stops. When a user is a member of multiple groups, the first supervisor found is selected. Should a group being searched contain multiple supervisors, Jacada WorkSpace randomly selects a supervisor from among the candidates.
- Editing a group

Administering Skills

A skill defines one or more areas of expertise related to the customer's business and the Jacada WorkSpace project. Skills enable you to formally classify a group's expertise. Skills are assigned to groups. Multiple skills can be assigned to a group, and multiple groups can be assigned to a skill.

Note: Skill assignment to a group is not required in order for the group to be assigned as an expert group. Be informed about skills that are used in your Jacada WorkSpace projects. Projects can access and evaluate skill related information.

Skills are managed via the *Configuration > Authentication-Authorization > Skills* option in the Jacada Management Console navigation menu. The *Skills* page enables the following activities:

- Adding a skill, which includes:
 - Defining its name and describing it
 - Assigning the skill to groups
- Deleting a skill

Note: A skill assigned to one or more groups cannot be deleted.

- Editing a skill, which includes:
 - Updating its description
 - Updating the skill's group assignments

Administering Users

In order to access Jacada WorkSpace, users undergo authentication upon login. Therefore, if the Jacada WorkSpace database is being used to provide user authentication, the administrator defines users.

On establishing a Jacada WorkSpace session, each user's ability to work with Jacada WorkSpace is determined by the groups to which they are assigned. Users must be assigned to at least one group and can be assigned to multiple groups.

Users are managed via the *Configuration > Authentication-Authorization > Users* option in the Jacada Management Console navigation menu. The *Users* page enables the following:

- **Adding a user**, which includes defining the user's name and password and assigning the user to groups. Each user must be a member in a group that is assigned the role AgentRole. For a user to be assigned as a supervisor, in the Edit Group page, the user must be a member in a group that is assigned the role SupervisorRole.

When an LDAP server is being used to provide user authentication, adding a user is not available in the *Users* page, since users are owned and managed by the LDAP server. Jacada WorkSpace requests from the LDAP server all users that are designated as supervisors of the user logging in to Jacada WorkSpace, and enters these users in the database.

- **Deleting a user**

Note: A user assigned as a supervisor to one or more groups cannot be deleted.

- **Editing a user**, which includes updating the user's password and the user's group assignments.

Each user must be a member in a group that is assigned the role `AgentRole`. For a user to be assigned as a supervisor, in the Edit Group page, the user must be a member in a group that is assigned the role `SupervisorRole`.

When an LDAP server is being used to provide user authentication, editing a user is not available in the *Users* page, since users are owned and managed by the LDAP server. Jacada WorkSpace requests from the LDAP server all users that are designated as experts to the user logging in to Jacada WorkSpace, and enters these users in the database.

7

Computer Telephony Integration (CTI)

The following topics provide information about support for Computer Telephony Integration (CTI) in Jacada WorkSpace:

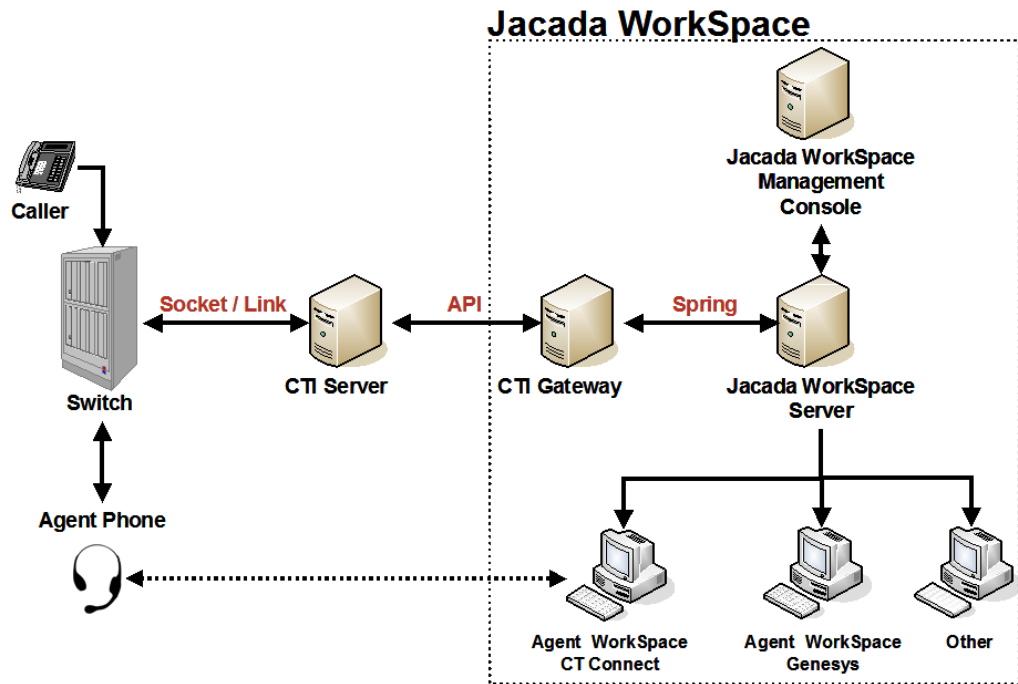
- [Support for CTI in Jacada WorkSpace](#)
- [CTI-Related Concepts](#)
- [Supported Features Matrix](#)
- [Configuring CTI](#)
- [CTI Provider-Specific Configuration](#)
- [Handling Calls in Jacada WorkSpace](#)

The functionality supported in Jacada WorkSpace depends on the CTI provider, and may vary between providers. These topics describe the general support provided by Jacada for all the supported CTI providers.

Support for CTI in Jacada WorkSpace

Jacada WorkSpace provides a CTI Gateway to support computer telephony integration in Jacada WorkSpace. The CTI Gateway is packaged as an EAR file that is deployed during installation to the application server. The CTI Gateway uses APIs and Spring remoting services to pass information between the Jacada WorkSpace application and the CTI server in use.

The following diagram demonstrates the Jacada WorkSpace architecture for using CTI.



Jacada WorkSpace provides gateways to support the following CTI providers:

- Avaya Application Enablement Service (AES)
- Avaya Elite Multichannel
- Cisco
- Genesys Telecommunications Laboratories Inc.

The functionality supported in Jacada WorkSpace depends on the CTI provider, and may vary between providers.

CTI-Related Concepts

This section introduces the following CTI-related concepts:

- Multi-Channel Support
- Call Handling Modes
- Call Transfer and Consultation Types
- Direct-Dial Calls

Multi-Channel Support

The Jacada WorkSpace CTI Gateway provides support for a multi-channel environment, which includes the following channels for interacting with customers:

- **Voice:** Support for handling phone calls. Call-handling capabilities are integrated into Jacada WorkSpace.
- **Custom Item:** Support for items, such as faxes or scanned documents.
- **Chat:** Support for handling incoming web chat messages.
- **Email:** Support for handling incoming email messages.

Call Handling Modes

When using the voice channel, Jacada WorkSpace supports the following call handling modes:

- **Manual:** No CTI support. Jacada WorkSpace is not connected to the telephone system. The agent uses a telephone, and is responsible for manually starting and then ending the handling of a call in Jacada WorkSpace.
- **CTI Inbound:** CTI is used for inbound calls. When the agent is ready to receive calls, calls are automatically directed to Jacada WorkSpace by the CTI Gateway. This option also supports making outbound calls.
- **CTI Outbound:** CTI is used for outbound calls. The agent initiates outbound calls in Jacada WorkSpace, via the CTI Gateway.

In production, the application switches to manual mode if there are CTI connectivity problems.

Call Transfer and Consultation Types

When handling a customer call, it may be necessary to transfer the call to another agent or a supervisor. The manner in which the call is transferred depends on the transfer types supported by the CTI provider and the switch in use. Depending on the CTI provider, Jacada WorkSpace supports the following call transfer types:

- **Cold transfer:** Transfers the call directly. The agent is disconnected from the call.
- **Handshake transfer:** Ensures there is a response from the other side before transferring the call. While waiting for the response, the caller is put on hold. The following scenarios are possible:
 - If there is a response, the call is transferred.
 - If the line is busy, the call is returned to the agent.
 - To stop the transfer, the agent can press Off Hold.
- **Warm transfer:** Enables talking to another agent or a supervisor, before transferring the caller. The caller is put on hold, the transfer number is dialed, and when the transfer is answered, the agent and supervisor talk. When they reach a decision, the agent can either transfer the caller or return to the caller.
- **Consult:** Enables talking to another agent or a supervisor before going back to the caller, transferring the caller or including the caller in a party conference call. The caller is put on hold, the supervisor's number is dialed, and when the supervisor answers, the agent and supervisor talk. After the consultation, the agent can transfer the caller to the supervisor, return to the caller, or turn the consultation into a three-party conference call. While the number dialed does not answer, the Off Hold button is used to cancel the consult and return to the original call.
- **Transfer connect:** Enables a transfer to be implemented by sending Dual Tone Multiple Frequencies (DTMF) tones to the switch.

Direct-Dial Calls

A direct-dial call is a call that was dialed directly to the agent extension (in contrast to a call that was redirected by a queue to the agent extension). An example is a supervisor calling an agent extension directly.

Jacada WorkSpace treats direct-dial calls as non-business calls, which means that the call-handling script is not loaded for this type of call, and that the call counters are not active. In Jacada WorkSpace, a direct-dial call is referred to as an *Internal Call* in the CTI *Status* field.

Calls received in the following cases are defined by Jacada WorkSpace as direct-dial calls:

- When Jacada WorkSpace status is *Not Ready*
- During call disposition

Supported Features Matrix

Some Jacada WorkSpace features are not supported in all CTI providers. The following table lists these features and indicates whether they are supported in each CTI provider:

Feature	AES	EMC	CISCO CTI	Genesys
Attached data support – retrieving	+	-	+	+
Attached data support – setting	-	-	-	+
Busy reason codes	-	+	+	+
Cold transfer	+	+	+	+
Conference	+	+	+	+
Consult	+	+	+	+
Custom item support	-	-	-	+
Filtering CTI events	-	-	-	+
Handshake transfer	+	-	+	+
Inbound calls	+	+	+	+
Internal calls	+	+	+	+
Login to CTI from phone	+	+	-	+

Feature	AES	EMC	CISCO CTI	Genesys
Multi-channel support	-	+	-	+
Multi-site configuration support	-	-	+	+
Operations from phone reflected in Jacada WorkSpace	-	+	+	+
Outbound calls	+	+	-	+
Outbound calls while in inbound mode	-	+	-	+
Passing CTI events to an external application	-	+	+	+
Reconnect to CTI	-	-	+	+
Transfer Connect	-	-	+	+
Transfer data from Smart Pad when transferring a call	+	-	-	+
Warm transfer	+	+	+	+

Configuring CTI

The use of Jacada WorkSpace with CTI is based on configuring settings in the database in the `CTI_SETTINGS`. These settings are used to configure CTI profiles and the *Machine Settings* page in the Jacada Management Console. These configurations are used to associate each CTI Gateway with one or more CTI profiles, thus enabling use of different CTI Gateways with different CTI providers. A CTI profile contains the configuration that is necessary for a specific CTI provider, or CTI provider and switch combination.

To use Jacada WorkSpace with CTI, you need to:

1. [Configure the necessary CTI profiles](#) in the `CTI_SETTINGS` table, only if a special configuration is needed. Otherwise, use the existing profiles.
2. [Configure the location of the CTI Gateway](#) in the `CTI_SETTINGS` table.
3. [Associate the CTI Gateways](#) with the suitable CTI profiles in the *Machine Settings* page in the Jacada Management Console.
4. [Launch Jacada WorkSpace](#) with the suitable CTI profile.

Configuring CTI Profiles

Jacada WorkSpace enables working with multiple CTI providers and switches, using the same database. This is done by defining CTI profiles in the `CTI_SETTINGS` table. A CTI profile contains the configuration that is necessary for working with a specific CTI provider and switch combination.

CTI profiles are defined at different levels, ranging from general profiles, to profiles specific to a CTI provider, to profiles for a specific CTI provider and switch combination. The CTI profiles are defined in a hierarchical manner, where a specific profile inherits the settings of a more general profile, and adds to these settings, or overrides them, as is the case. The inheritance mechanism is implemented by defining a parent profile for each CTI profile.

Note: For information on the `CTI_SETTINGS` keys and their values, refer to [CTI Tables](#).

When it is necessary to customize a profile, it is recommended to create a new profile that inherits the profile to customize, and make changes in the new profile. If no special configuration is needed, it is recommended to use the preconfigured profiles.

Preconfigured CTI Profiles

Jacada WorkSpace provides a set of preconfigured CTI profiles in the `CTI_SETTINGS` table, such as:

Profile	Description	Parent Profile
CTI	A root profile, with general CTI settings, that are not related to any specific CTI provider or switch.	-
CTI_AES	Settings for Avaya AES.	CTI
CTI_AES_AVAYA	Settings for Avaya AES using the AVAYA switch.	CTI_AES
CTI_CISCO	Settings for Cisco	CTI
CTI_GENESYS	Settings for Genesys.	CTI
CTI_GENESYS_ALCATEL	Settings for Genesys using the Alcatel switch.	CTI_GENESYS
CTI_GENESYS_AVAYA	Settings for Genesys using the Avaya switch.	CTI_GENESYS
MANUAL_AVAYA_ONEX	Settings for Avaya using Avaya one-X Agent	MANUAL
CTI EMC_AVAYA	Settings for Avaya Elite Multichannel.	CTI EMC

Configuring the Location of the CTI Gateway

By default, each Jacada WorkSpace session uses the CTI Gateway deployed on the application server, or on the managed server in a clustered environment, running that session. This is configured by default in the `CTI_SETTINGS` table, by specifying the literal token `CurrentInstance` as the value of the `CTIGatewayLocation` setting.

However, since the Jacada CTI Gateway can be deployed on any application server, when the CTI Gateway is located on a different machine, it is necessary to specify the following URL as the value of the `CTIGatewayLocation` setting:

```
http://<IP>:<port>/CTIGateway/InvocationService
```

Associating a CTI Provider with Each CTI Gateway

The [Machine Settings page](#) in the Jacada Management Console is used to associate each CTI Gateway with one or more CTI profiles, thus enabling use of different CTI Gateways with different CTI providers. When a CTI Gateway starts, it searches for its own host address and port details in the *Machine Settings* page, and locates the most specific match.

The value of the host and the port can be either a specific value, or an asterisk (*) that represents any value. The following search order is used when searching for a host and port match:

1. [specific host, specific port]
2. [specific host, port=*]
3. [host=*, specific port]
4. [host=*, port=*]

By default, the *Machine Settings* page is configured so all CTI Gateways are associated with the `MANUAL` profile, which starts Jacada WorkSpace in manual mode, with no CTI support.

You need to configure the *Machine Settings* page according to the CTI providers in use. In the following example, all CTI Gateways connect to Genesys using the `CTI_GENESYS` profile, except for the gateway that resides on 12.34.56.78:9080, that uses the `CTI_GENESYS_AVAYA` profile:

ACTIVE	HOST	PORT	NAME	VALUE	COMMENTS
true	*	*	PROFILES	CTI_GENESYS	
true	12.34.56.78	9080	PROFILES	CTI_GENESYS_ALCATEL	NULL

Note: In a simple configuration, where all CTI Gateways connect to the same profile, it is only necessary to add to the default row (*,*,...) the name of the profile to use.

Setting Up the Call Mode of an Application

The call mode is defined via the `CTICallMode` setting in the `CTI_SETTINGS` table. The valid values are:

- `inbound`
- `manual`

Logging Into CTI Via the Phone

By default, logging into or out of the CTI is done automatically as part of logging into or out of Jacada WorkSpace.

However, when using the **voice channel only**, in some cases, you may need the agent to perform the login to CTI via the phone, not automatically. Examples are to take advantage of switch-specific login options, or due to switch requirements. To log in to the CTI voice channel using the phone, you use the `CTILoginByPhone` setting in the `CTI_SETTINGS` table.

When logging in via the phone, the agent must first log in to Jacada WorkSpace, and only then log in to CTI via the phone.

Note: When the agent logs into CTI via the phone, the agent **must also log out of CTI via the phone**.

Configuring Jacada WorkSpace to Automatically Answer Incoming Calls

Answering calls automatically is done either by the switch, or by Jacada WorkSpace, in the database.

To configure calls to be answered automatically:

1. To specify that the Jacada WorkSpace application should automatically answer **all** calls, in the `CTI_SETTINGS` table, set the `CTICallAutoAnswer` setting to `true`.
2. Specify whether the switch is configured to automatically answer ACD calls, using the `CTISwitchAutoAnswer` setting in the `CTI_SETTINGS` table.

When you do not want calls to be answered automatically, set the `CTICallAutoAnswer` setting to

`false`, and then verify with your switch administrator that the switch is also configured accordingly. In this case, the value of the `CTISwitchAutoAnswer` setting is not used.

Launching Jacada WorkSpace with a CTI Profile

In order to use CTI with Jacada WorkSpace, you need to launch Jacada WorkSpace with the suitable CTI profile. The profile to use can be specified at different levels, increasing in their scope: per invocation, per agent (supported for CT Connect and AES only), or per project.

Note: If you do not specify a CTI profile, the default profile, `DEFAULT`, is used, which launches Jacada WorkSpace in manual mode (with no CTI support).

The following table describes the different ways to launch Jacada WorkSpace with a CTI profile.

Scope	Specify profile in	Details
Per invocation	URL that launches Jacada WorkSpace	Use the <code>CTIProfile</code> parameter in the URL that launches Jacada WorkSpace. For example: <pre>http://<ip>:<port>/myApp?CTIProfile=CTI_GENESYS</pre> When a <code>CTIProfile</code> is not specified, the <code>DEFAULT</code> profile, defined in the default database settings, is used.
Per agent (supported for CT Connect and AES only)	<code>CTI_AGENT</code> table	Specify a value for the <code>CTIProfile</code> setting in the <code>CTI_AGENT</code> table, to define a persistent profile per agent, without the need to specify the profile in the URL. For more information, refer to CTI Tables.

Scope	Specify profile in	Details
Per project	login method	Enables custom-coding the profile for a project. The method is located in: <pre><project_name>\Java Resources\src \USER.global.managers.ExtensionManagersjava</pre> The following example shows how to set the profile programmatically: <pre>@GlobalAction(actionName=GlobalActionNames.LOGIN,order=GlobalActionOrder.BEFORE) public void login(HttpServletRequest request){ globalApp.getCTIManager().getVoiceInformation().setCti ProfileName("<prof_name>"); }</pre>

CTI Provider-Specific Configuration

The following sections describe the basic configuration necessary to start using Jacada WorkSpace with the supported CTI providers:

- [Working With Avaya Application Enablement Service \(AES\)](#)
- [Working With Avaya Elite Multichannel \(EMC\)](#)
- [Working With Avaya one-X Agent](#)
- [Working With Cisco](#)
- [Working With Genesys](#)

Working With Avaya Application Enablement Service (AES)

Jacada WorkSpace supports using Avaya Application Enablement Service (AES) as the CTI provider.

To use AES as the CTI provider for Jacada WorkSpace:

1. Create a file named `TSAPI.PRO` with the following line:

```
<IP of AES machine>=<Port of AES machine>
```

For example:

```
12.34.56.78=450
```

Place the `TSAPI.PRO` file in the relevant folder:

- **Tomcat:** `<Tomcat_home>/jacada/conf`
- **WebSphere:** `<WebSphere_Application_Server_Home>/profiles/<profile_name>/properties`
- **WebLogic:** Before relocating the file, compress the file as a .zip, and change the file type to .jar. Then, place the file in `<OracleWebLogic_Home>/user_projects/domains/<domain_name>`.

Note: A sample `TSAPI.PRO` file is available in `<Jacada_WorkSpace_installation_folder>/Workspace/Accessories`.

2. In the `CTI_SETTINGS` table, in the AES CTI profile that you want to use, such as one of the existing profiles, `CTI_AES` or `CTI_AES_AVAYA`, do the following:
 - a. Configure `CTIUserName` and `CTIPassword` to a user and password defined for your AES machine. For more information, refer to [Configuring CTI Profiles](#).
 - b. Adjust any additional settings according to your environment. For more information, refer to [CTI Tables](#).
3. By default, each Jacada WorkSpace session uses the CTI Gateway deployed on the server, or on the managed server in a clustered environment, running that session. This is configured in `Global Settings` in the Jacada Management Console, by specifying the literal token `CurrentInstance` as the value of the `CTIGatewayLocation` setting. However, since the Jacada CTI Gateway can be deployed on any machine that houses an application server, when the CTI Gateway is located on a different machine, it is necessary to specify the following URL as the value of the `CTIGatewayLocation` setting:

```
http://<IP>:<port>/CTIGateway/InvocationService
```

4. In the *Machine settings* page in the Jacada Management Console, make sure the CTI Gateway you are connecting to is configured to use the AES profile specified in Step 2, above. If all the gateways connect to AES, then you can add the CTI_AES profile to the default row in the table. For more information, refer to [Machine Settings](#).
5. Make sure that the CTI agent information is accessible. This can be performed, for example, by implementing a lookup mechanism in the project code or in the Jacada Management Console, in *CTI > Agents*, with the following information:
 - User name** – The application server or LDAP user name
 - CTI Profile** – CTI profile to use, from the CTI_SETTINGS table
 - Extension number** – The agent's extension number
 - Group ID** – For internal use
 - CTI login name** – Switch login ID
 - CTI password** – The agent switch password
 - Outbound extension number** – For internal use
 - Agent ID** – Switch login ID
6. Make sure that user authentication, the application server and LDAP user name, is properly configured.
7. For changes to take effect, restart the server.

You can verify that the CTI Gateway is connected to the AES machine, by reviewing the messages in the log file.

Configuring Call Disposition (AES)

When an agent ends a call, the next state is determined by the Jacada Workspace configuration.

When `CTIApplicationSetACW` is set to `true`, the default call disposition starts automatically, and the agent moves to the *After-Call-Work* state automatically.

When `CTIApplicationSetACW` is set to `false`, there is no call disposition, and the agent does not move to *After-Call-Work* state.

Working With Avaya Elite Multichannel (EMC)

Jacada WorkSpace supports using Avaya EMC as the CTI provider.

Note: When using Avaya EMC, [install Jacada WorkSpace using Tomcat](#) and create the server and database using the [Server Setup Wizard](#).

Installing the EMCGateway for Avaya Elite Multichannel

The following sections describe how to set up communication between Jacada WorkSpace and Avaya EMC, using the EMCGateway bridge. The EMCGateway supports IIS 7.0 and IIS 8.5.

Before you begin, verify that:

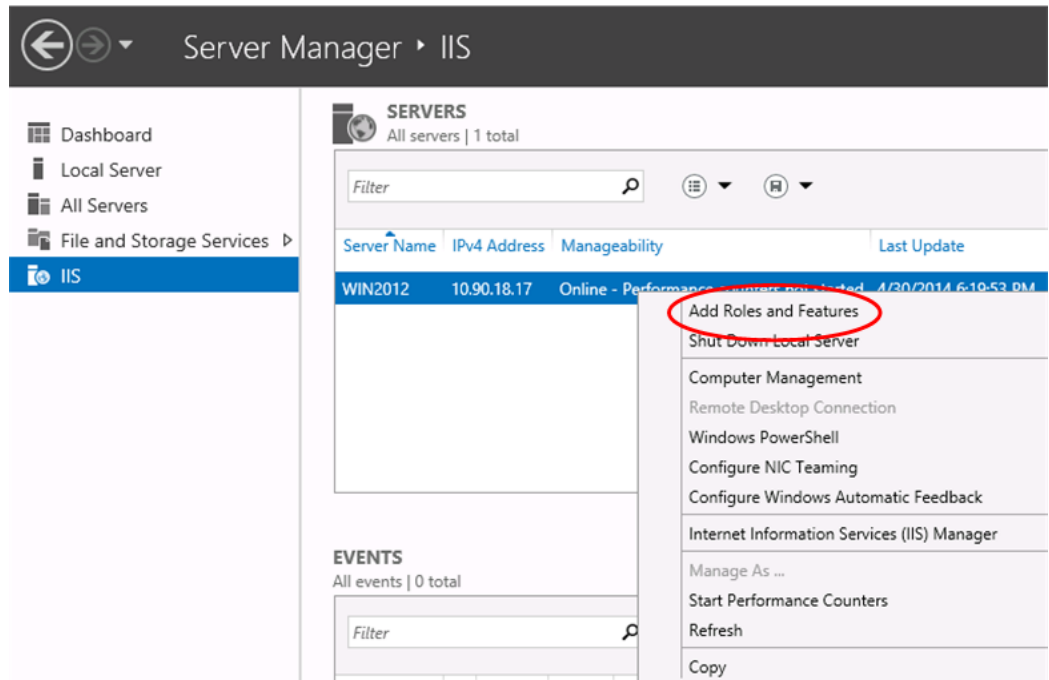
- .NET 4.5 is installed.
- Avaya Media Proxy is installed, and that the Web Deploy Extension is installed on your IIS server (<http://www.iis.net/downloads/microsoft/web-deploy>).
- The system locale of your operating system is set to English. (To check this setting, from the control panel, navigate to the *Region and Language* settings.)

Server Configuration

The following procedure explains how to set the required roles to be installed on the server.

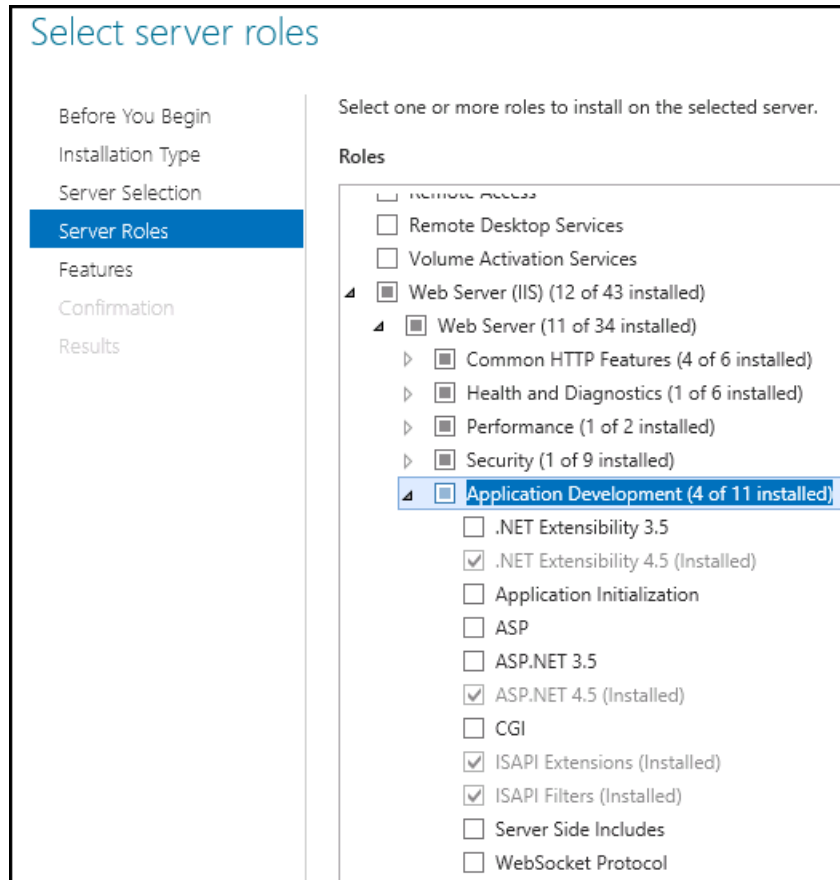
To configure server settings:

1. From the Server Manager, select *IIS > Add Roles and Features*.



The Add Roles and Features wizard opens.

2. Navigate to *Server Roles > Web Server (IIS) > Web Server > Application Development*, and select the following checkboxes:
 - .Net Extensibility
 - ASP .Net 4.5
 - ISAPI Extensions
 - ISAPI Filters



3. Save your settings.

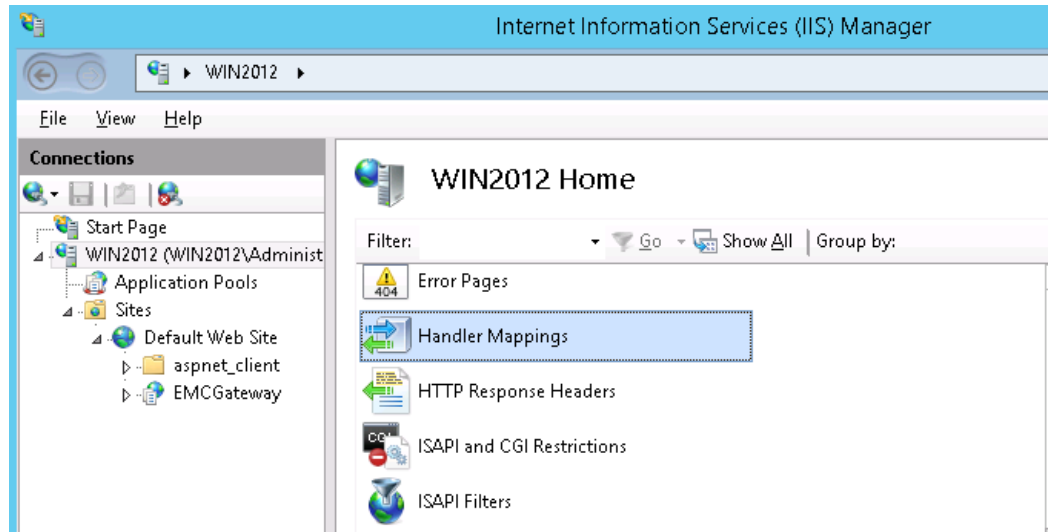
IIS Configuration

The following sequence of procedures explain how to perform the required IIS configuration. The procedures are done from the IIS Manager.

Handler Mapping and Script Map

To add a new handler mapping and script map:

1. From the *Connections* pane of the IIS Manager, select the machine that hosts the IIS. Then, from the *Features* view, select *Handler Mappings*.

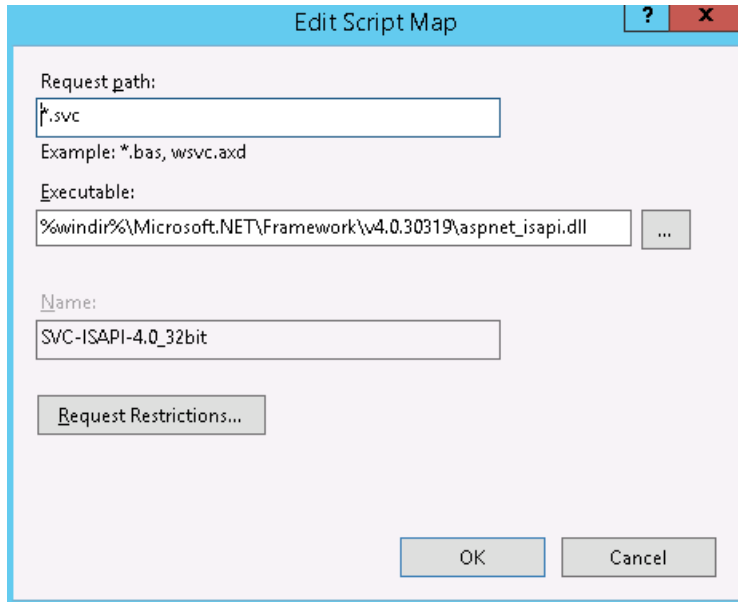


The *Handler Mappings* page opens.

2. From the *Actions* pane at the right side of the page, select *Add Script Map*.
The *Edit Script Map* popup opens.

3. In the popup, enter the following settings:

- **Request path:** *.svc:
- **Executable:** %windir%\Microsoft.NET\Framework\v4.0.30319\aspnet_isapi.dll
- **Name:** SVC-ISAPI-4.0_32bit

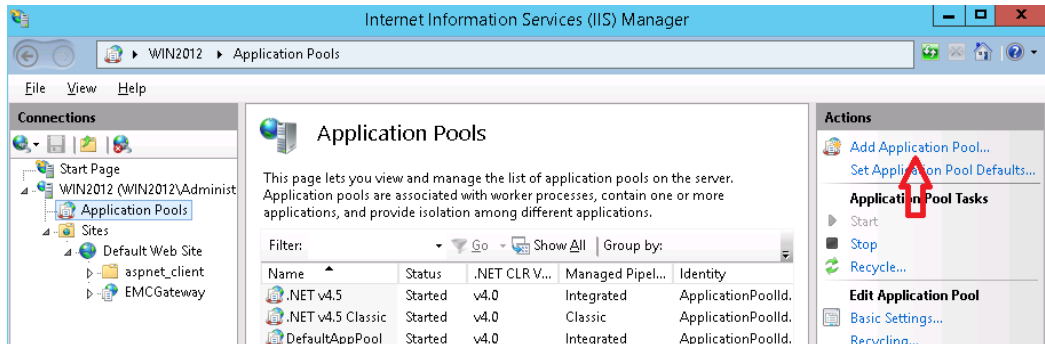


4. Click *Request Restrictions*, and configure settings on each of the tabs:
 - **Mapping:** Clear the checkbox.
 - **Verbs:** Select the *All verbs* radio button.
 - **Access:** Select the *Script* radio button.
5. Save your settings.

Adding and Configuring the Application Pool

To add an application pool:

1. From the *Connections* pane of the IIS Manager, select *Application Pools*.
The *Application Pools* page opens.
2. From the *Actions* pane at the right side of the page, select *Add Application Pool*.



The *Add Application Pool* popup opens.

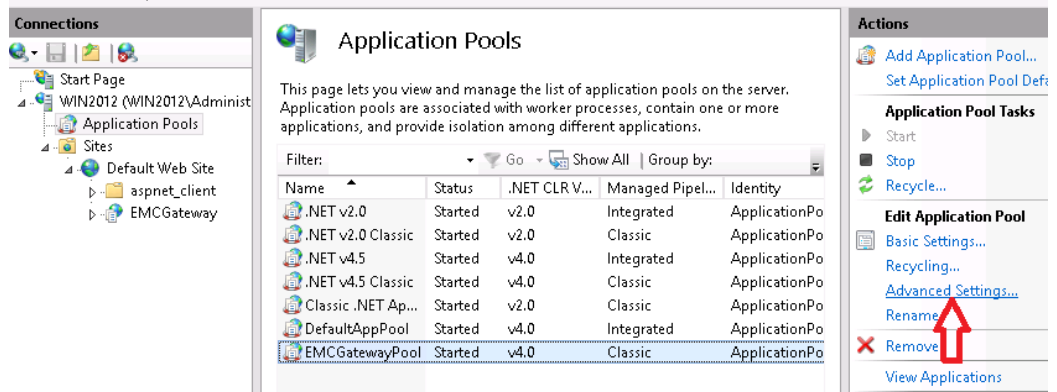
3. In the popup, configure the following settings:
 - In the *Name* field, enter *EMCGatewayPool*.
 - From the *.NET CLR version* dropdown list, select *.NET CLR Version V4.0.30319*.
 - From the *Managed pipeline mode* dropdown list, select *Classic*.
 - Select the *Start application pool immediately* checkbox.

The screenshot shows the 'Add Application Pool' dialog box. The 'Name' field contains 'EMCGatewayPool'. The '.NET CLR version' dropdown is set to '.NET CLR Version v4.0.30319'. The 'Managed pipeline mode' dropdown is set to 'Classic'. The 'Start application pool immediately' checkbox is checked. At the bottom are 'OK' and 'Cancel' buttons.

4. Save your settings.

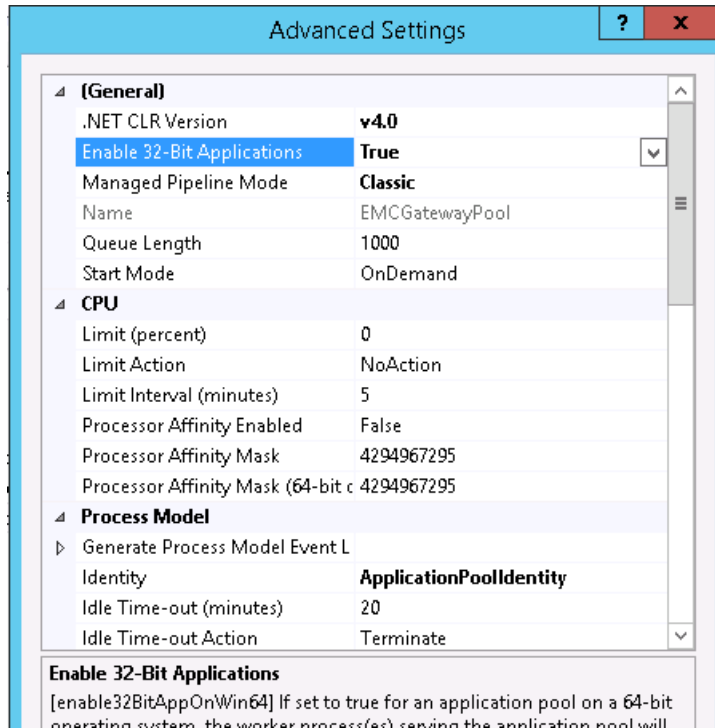
To configure the application pool:

1. From the *Application Pools* page of the IIS Manager, select *EMCGatewayPool*. Then, from the *Actions* pane at the right side of the page, select *Advanced Settings*.



The *Advanced Settings* popup opens.

2. In the popup, configure the following settings:
 - **.NET CLR Version:** 4.0
 - **Enable 32-Bit Applications:** True
 - In the *Process Model* section, set the *Idle Time-out (minutes)* to 0.
 - In the *Recycling* section, set the *Regular Time Interval (minutes)* to 0.



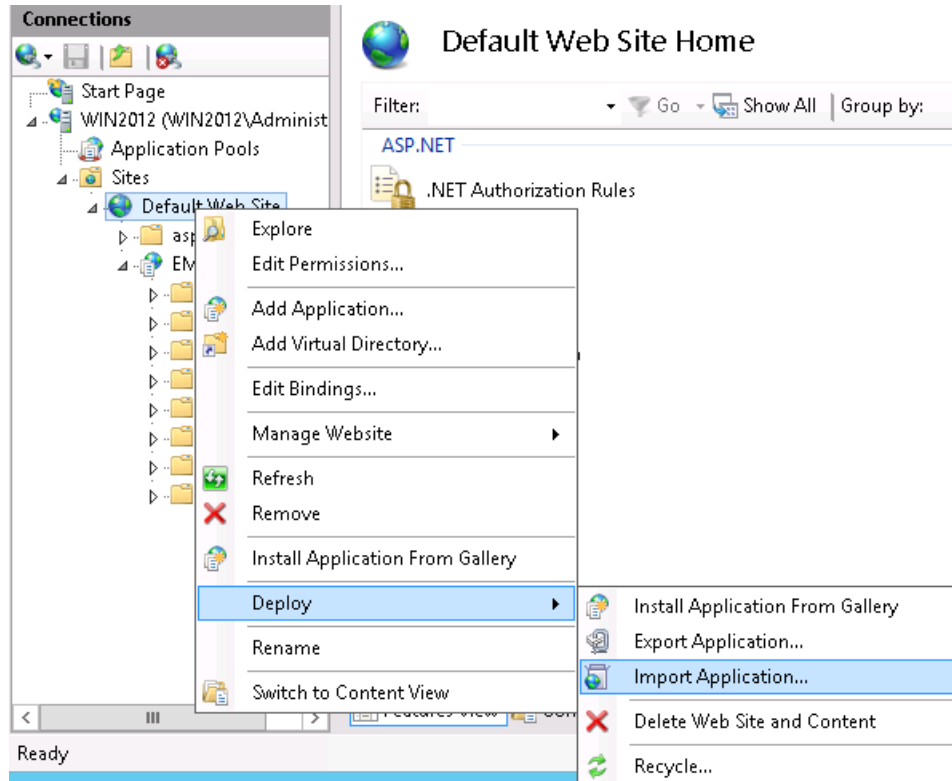
3. Save your settings.

Deploying the EMCGateway Application

The **EMCGatewayService.zip** file is used to deploy the application.

To deploy the EMCGateway application:

1. From the *Connections* pane of the IIS Manager, select *Default Web Site*. Then, from the context menu, select *Deploy > Import Application*.

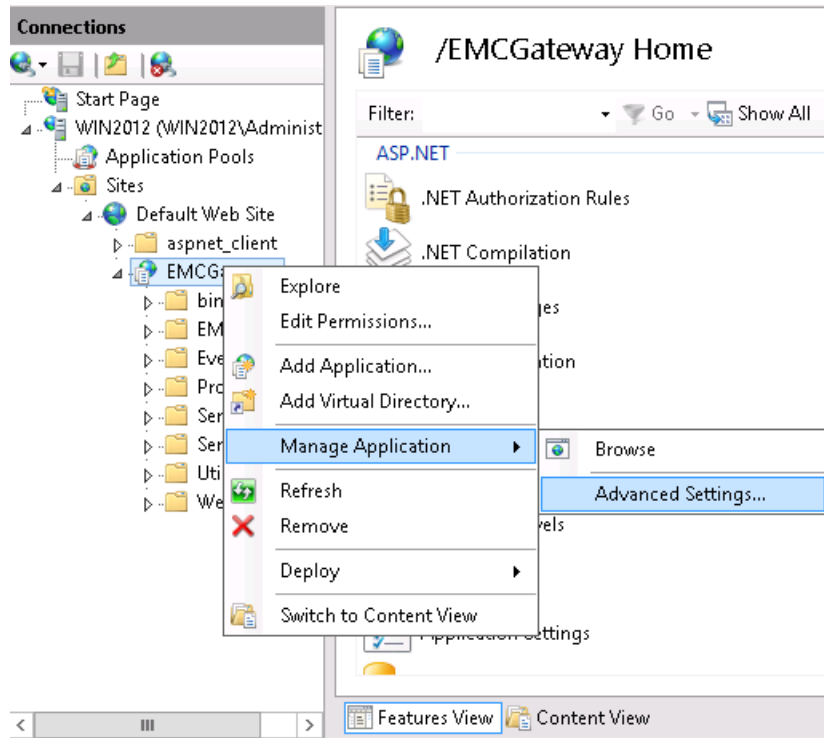


2. Navigate to the **EMCGatewayService.zip** file, and then continue with the default options.

Assigning the Application Pool to EMCGateway

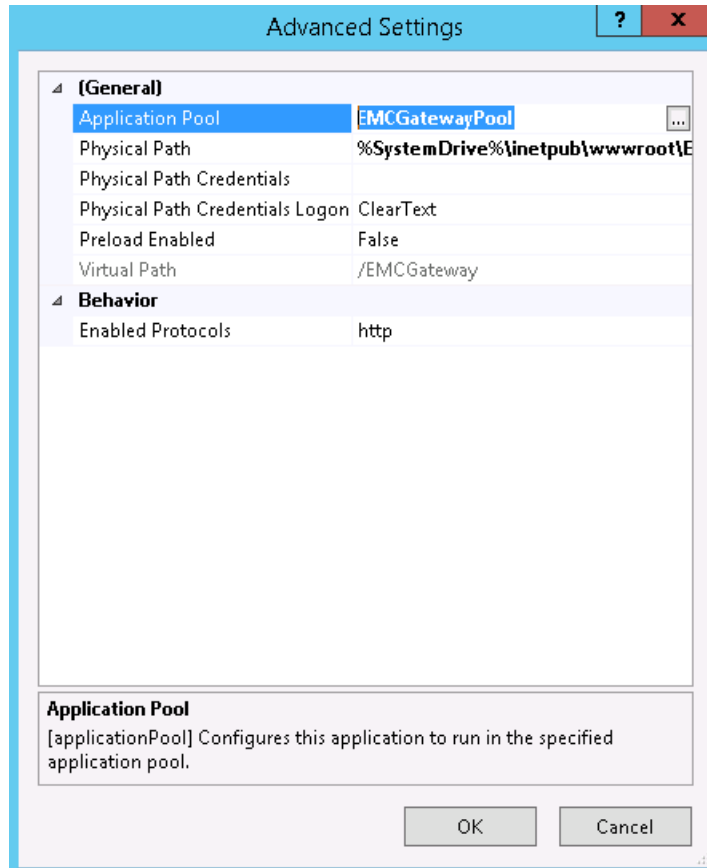
To assign the application pool:

1. From the *Connections* pane of the IIS Manager, select *EMCGateway*. Then, from the context menu, select *Manage Application > Advanced Settings*.



The *Advanced Settings* dialog opens.

2. Set the *Application Pool* setting to *EMCGatewayPool*.



3. Click **OK** and save your configuration settings.

Testing Your Installation

To test the effectiveness of your installation, open a browser and go to <http://localhost/EMCGateway/RestServiceImpl.svc/version>

The response received includes a build number assigned by developers, and a timestamp showing when the code was compiled. For example:

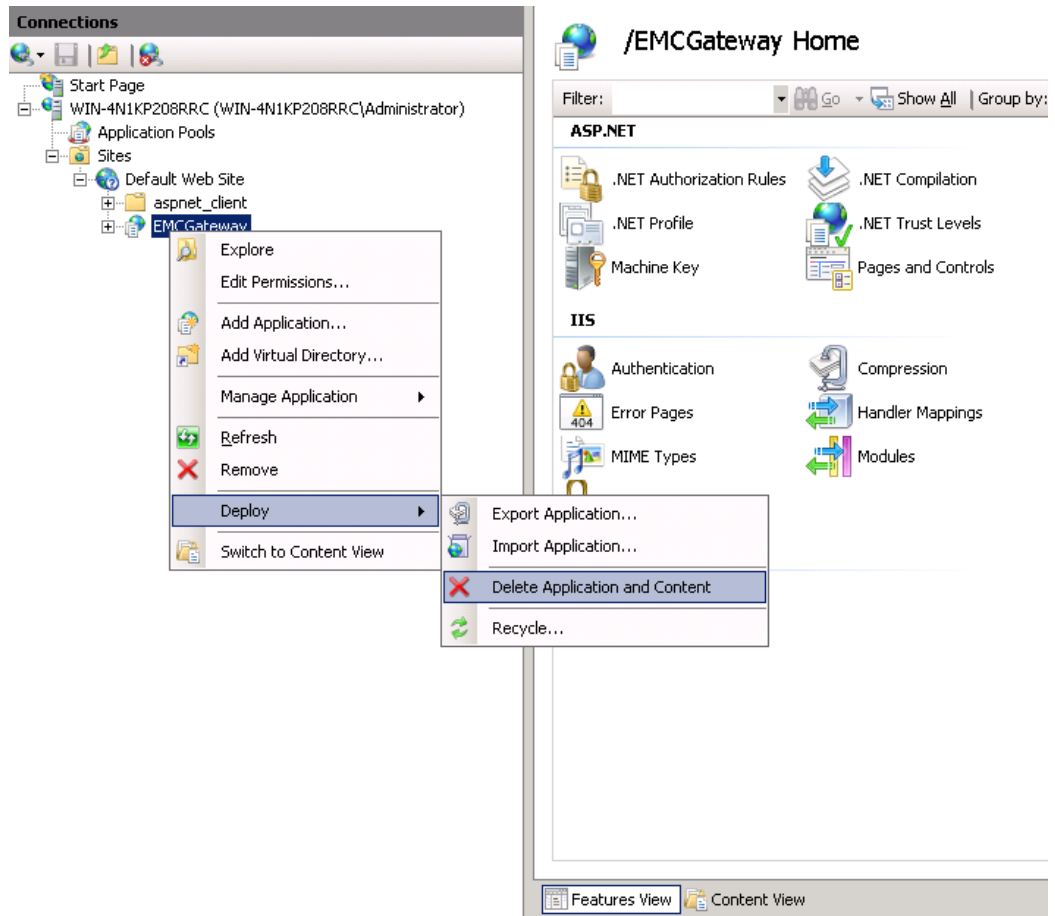
Jacada EMC Gateway 1.0.0.22 30/04/2014 04:02:31

Handling New Builds and Upgrades

Before installing a new build or version, it is necessary to delete your current application.

To perform an upgrade:

1. From the *Connections* pane of the IIS Manager, select *EMCGateway*. Then, from the context menu, select *Deploy > Delete Application and Content*.



2. If you are prompted with any questions from the IIS Manager, select *Yes*.
3. Verify that the application is deleted.
4. To install the new build, from the *Connections* pane of the IIS Manager, select *Default Web Site*. Then, from the context menu, select *Deploy > Import Application*.
5. Navigate to the **EMCGatewayService.zip** file, and then continue with the default options.

Configuring Avaya EMC as the CTI Provider

To use Avaya EMC as the CTI provider for Jacada WorkSpace:

1. In the EMC Profile Settings table, in the CTI EMC Avaya profile, define values for the following settings:

Setting	Description	Notes
CTIProvider	emc	This predefined value must not be modified.
ProxyIP	IP address of the proxy server	localhost
ProxyPort	Port number of the media proxy server	29079
ServerIP	IP address of the Media Director	localhost
ServerPort	Port number of the Media Director	29096

2. In the *Machine settings* page in the Jacada Management Console, make sure the CTI Gateway you are connecting to is configured to use the EMC profile specified in Step 1, above. If all the gateways connect to EMC, then you can add the CTI_EMCM_AWAYA profile to the default row in the table. For more information, refer to [Machine Settings](#).
3. For changes to take effect, restart the server.
You can verify that the CTI Gateway is connected to the EMC machine by reviewing the messages in the log file.

Working With Avaya one_X Agent

Jacada WorkSpace supports using Avaya one-X Agent as a CTI provider.

When integrating with Avaya one-X Agent, verify that the `OneXPort` setting is defined in the `CTI_SETTINGS` table. This setting defines the port number that the client registers on (ranges from 60000-61000). The exact port number can be found in **HKCU/Software/AvayaonexAgent/**

Settings/APIPort.

Working With Cisco

Jacada WorkSpace supports using Cisco as a CTI provider.

To configure Cisco as the CTI provider for Jacada WorkSpace:

1. In the `CTI_SETTINGS` table, in the Cisco CTI profile that you want to use (such as `CTI_CISCO`), configure the following fields:
 - CTIServer** – The IP address of the Cisco server.
 - CTIPort** – The port where the Cisco server runs.
 - CTISBackupServer** – The IP address of a backup Cisco server.
 - CTIBackupPort** – The port where the backup Cisco server runs.
 - CTICiscoConnectionHeartBeat** – The Cisco connection heart beat.
2. By default, each Jacada WorkSpace session uses the CTI Gateway deployed on the server, or on the managed server in a clustered environment, running that session. This is configured in the `GLOBAL_SETTINGS` table, or in `Global Settings` in the Jacada Management Console, by specifying the literal token `CurrentInstance` as the value of the `CTIGatewayLocation` setting. However, since the Jacada CTI Gateway can be deployed on any machine that houses an application server, when the CTI Gateway is located on a different machine, it is necessary to specify the following URL as the value of the `CTIGatewayLocation` setting:
`http://<IP>:<port>/CTIGateway/InvocationService`
3. Optionally, specify the following in the `CTI_AGENT` table:
 - AGENTID** – The agent id for a specific agent.
 - CTIPASSWORD** – The password for the agent.
 - EXTNUMBER** – The phone extension number for the agent.These values can also be defined in the project code, called during the login procedure, using the following APIs:
 - agentInformation.setCtiPhoneID()** – The agent id for a specific agent.
 - agentInformation.setCtiAgentPassword()** – The password for the agent.
 - agentInformation.setExtension()** – The phone extension number for the agent.
4. In the project code, use the `agentInformation.setCtiAgentPeripheralId()` API, called during the login procedure, to specify the site or switch ID used by the agent.
5. In the `MACHINE_SETTINGS` table, make sure the CTI Gateway you are connecting to is configured to use the `CTI_CISCO` profile. If all the Gateways connect to `CTI_CISCO`, then you can add the

CTI_CISCO profile to the default row in the table.

6. Make sure that the CTI agent information is accessible, for example, by implementing a lookup mechanism in your code, in the `login` method in `<project_name>\Java Resources\src\USER.global.managers.ExtensionManager.java` or in the `CTI_AGENT` table.
7. Make sure that user authentication is properly configured.
8. For your changes to take effect, restart the server.
You can verify that the CTI Gateway is connected to the Cisco machine by reviewing the messages in the server console.
9. Whether or not there is call disposition depends on the switch settings. To enable call disposition to start automatically, where agents move to the *After-Call-Work* state automatically, the switch must be configured as follows:
Log into the Avaya switch and enter the command `change hunt-group n`, where *n* is the hunt group relevant to the site. Also, set the `Timed ACW Interval (sec)` value, as follows:
 - To move to call disposition without a time limit, set the value to 9999.
 - To turn off ACW, set the value to spaces.

Working With Genesys

Jacada WorkSpace supports using Genesys as the CTI provider. To use Genesys as the CTI provider, you need to configure Jacada WorkSpace, and perform additional configuration in Genesys, as described in the following sections.

The CTI Gateway uses the AIL client to communicate with the Genesys Configuration Manager Environment. The AIL client keeps a configuration cache, which is fully dynamic: a modification in the Configuration Layer is immediately updated in the cache. The connection between the AIL client and the Genesys Configuration Manager is a long running connection, and must not be terminated in any intermediate layer. If the connection is terminated, changes made in the Genesys Configuration Manager will not be reflected in the AIL client and the CTI Gateway.

Note: Jacada WorkSpace prevents customers who do not have a license for Genesys from accessing Genesys via the Jacada CTI Gateway. When a license file does not exist, it is not possible to use Genesys CTI, only manual mode.

The CTI Gateway licensing for Genesys is implemented within the Jacada WorkSpace License Server. Thus, if access to the License Server is not available, access to Genesys CTI will also not be available.

A Jacada license that supports Genesys can be obtained from your support representative.

For further information, refer to [Additional Genesys-Specific Configuration](#).

To configure Jacada WorkSpace for Genesys:

1. In the `CTI_SETTINGS` database table, in the Genesys CTI profile that you want to use, such as one of the existing profiles, `CTI_GENESYS_ALCATEL` or `CTI_GENESYS_AVAYA`, do the following:
 - a. Configure the `CTIServer` and `CTIPort` of the Genesys Configuration Server.

The `CTIServer` is the Genesys Configuration Manager server host name. If your machine cannot access this host, you need to add the host name to the operating system `hosts` file.

The default `CTIPort` is 2020.

For more information, refer to [Configuring CTI Profiles](#).
 - b. Adjust any additional settings according to your environment. For more information, refer to [CTI Tables](#).
2. By default, each Jacada WorkSpace session uses the CTI Gateway deployed on the server, or on the managed server in a clustered environment, running that session. This is configured in `Global Settings` in the Jacada Management Console, by specifying the literal token `CurrentInstance` as the value of the `CTIGatewayLocation` setting. However, since the Jacada CTI Gateway can be deployed on any machine that houses an application server, when the CTI Gateway is located on a different machine, it is necessary to specify the following URL as the value of the `CTIGatewayLocation` setting:
`http://<IP>:<port>/CTIGateway/InvocationService`
3. In the *Machine settings* page in the Jacada Management Console, make sure the CTI Gateway

you are connecting to, is configured to use the Genesys CTI profile specified in Step 1, above. If all the gateways connect to Genesys, then you can add the `CTI_GENESYS` profile to the default row in the table. For more information, refer to [Machine Settings](#).

4. For your changes to take effect, restart the server.

You can verify that the CTI Gateway is connected to the Genesys machine, by reviewing the messages in the server console.

Note: For details on working in a multi-channel environment, which includes support for Genesys Open Media, refer to [Configuring the Database to Handle Custom Items](#).

For information on configuring the Genesys application for a multi-site configuration, refer to [Support for a Genesys Multi-Site Configuration](#).

Some switches may require a password when logging in to the switch. In this case the password needs to be set in the code. In order to supply a password, you need to add code to the `login` method of the application, as described in [Support for Agent Passwords at Switch Level](#).

Genesys T-Server Configuration

In the T-Server, set the flag `set-otherdn-trunk-info` to `true`.

In Genesys, the following configuration options define whether the T-Server should assign direct-dial incoming calls to, and outgoing calls from, agents as business calls:

`inbound-bsns-call` (Genesys default: `false`)

`outbound-bsns-call` (Genesys default: `false`)

When both these settings are set to `true`, the T-Server, and not Jacada WorkSpace, is responsible for moving the agent to call disposition, which ensures that waiting work is not routed to the agent.

Working in Inbound Mode

When Jacada WorkSpace is in inbound mode, the recommended configuration is:

```
inbound-bsns-call: true
```

```
outbound-bsns-call: false
```

In this configuration, business inbound calls are automatically moved by the T-Server to call disposition. Outbound calls, which are only allowed when working with custom items, will not be followed by call disposition, and move to the ready state.

Working in Outbound Mode

When Jacada WorkSpace is in outbound mode, the recommended configuration is:

```
inbound-bsns-call: not relevant
```

```
outbound-bsns-call: true
```

When the `outbound-bsns-call` setting is set to `true`, the T-Server is responsible for moving the agent to call disposition.

When the `outbound-bsns-call` setting is set to `true`, the T-Server is responsible for moving the agent to call disposition. In this case, Jacada WorkSpace is responsible for exiting the disposition stage.

When the T-Server is down, an agent logging in receives a “pc down, initializing” message until the T-Server is up.

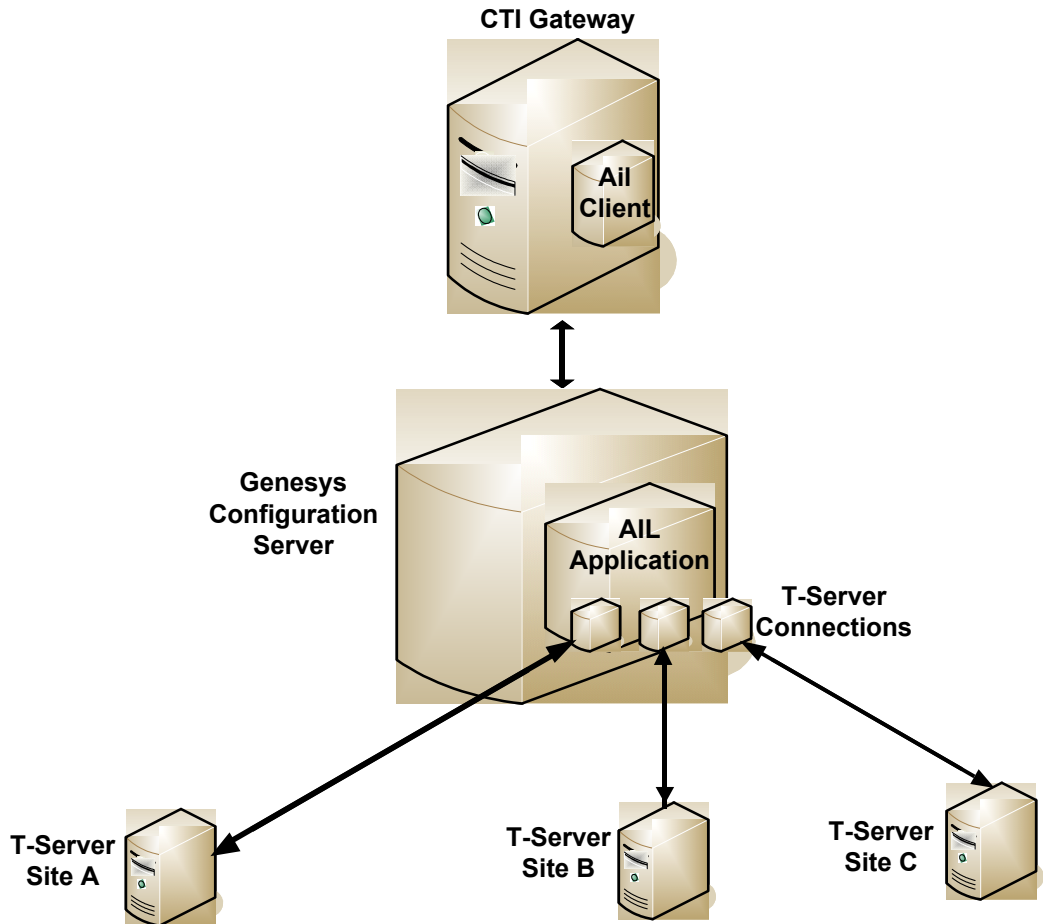
When the `outbound-bsns-call` setting is set to `false`, and the agent dials an outbound call, when this call ends, Jacada WorkSpace tries to move to call disposition. In order to enable moving to call disposition, you must make sure that there are no intermediate statuses between ending the call and Jacada WorkSpace regaining control of the agent status. For example, in Alcatel, in case of non-internal calls, there is an intermediate status: *Charge Information*, which comes immediately after the call ends. In this status, the agent status cannot be changed. In order to regain control immediately after the call ends, it is necessary to reduce the duration of the *Charge Information* status to a minimum, that is, set it to 1 unit (which equals 100 milliseconds).

Support for a Genesys Multi-Site Configuration

Genesys supports a multi-site configuration, in which it is possible to work with multiple switches, located at multiple sites, that connect to multiple T-Servers. The T-Servers ensure that critical call data (such as ANI, DNIS, agent information and general attached data) is maintained as the call travels from one switch to another. In order for a multi-site configuration to operate, the switches must share the Genesys Configuration Server.

Jacada WorkSpace provides support for a Genesys multi-site configuration. The Jacada CTI

Gateway accesses the Genesys Configuration Server in order to determine to which switch (site) to transfer a call.



When working with multiple T-Servers, when one T-Server fails, only the agents connected to that specific T-Server (switch) are switched to manual mode. Once the T-Server is functional again, the CTI Gateway provides a reconnect option for the agents of that T-Server. For more information about the reconnect option, refer to [Handling Loss of Connectivity to CTI](#).

Configuring Jacada WorkSpace for a Genesys Multi-Site Configuration

The following sections describe how to configure Jacada WorkSpace to work with a Genesys multi-site configuration:

Note: It is assumed that the Genesys multi-site environment is already configured. For more information on configuring Genesys, contact your Genesys administrator.

Configuring the Database

In a Genesys multi-site configuration, in order to enable the Jacada CTI Gateway to connect to the Genesys Configuration Server, it is necessary to configure settings in the Jacada WorkSpace database, according to the Genesys configuration in use. The following table lists the database settings (located in the `CTI_SETTINGS` table) that are necessary to connect to the Genesys Configuration Server:

Setting	Description
CTIMultiSiteSupport	Specify whether to use Genesys multi-site support (true/false).
CTIServer	The name or IP address of the machine that houses the CTI server.
CTIPort	The port to use to connect to the CTI server. For Genesys, port 2020 is typically used.
CTIUserName	A user with access to the Genesys Configuration Server.
CTIPassword	A password that corresponds to the user above.
CTIConfigManagerApplicationName	The name of the Genesys application of type 'Config Manager', as defined in the Genesys Configuration Manager Environment (CME).

Configuring the Genesys Application

Genesys requires an AIL application, defined in the Genesys Configuration Manager Environment,

that serves as an access point for Jacada WorkSpace. For information on creating a Genesys application, refer to [Creating a Genesys Application](#). The Genesys application needs to be configured to support a multi-site configuration.

To configure an existing Genesys application for a multi-site configuration:

1. In the *Connections* tab, add all the T-Servers that are part of the multi-site configuration.
2. In the *Options* tab, under the *dn-at-switch* section, the `enabled` attribute must be set to `true`.

Details of a Successful Genesys Multi-Site Connection

When the connection to CTI is successful in a Genesys multi-site configuration, the following details are listed in the console, as shown in the example below:

- CTI server host name (`genesys`) and port (`2020`)
- Application name (`JacadaGW_10.9.8.9_7001`)

```
** Initialized connection to Genesys Configuration Server Protocol **
Agent Interaction Layer version is 7.2.020.00
Primary config server host is genesys
Primary config server port is 2020
Backup config server host is
Backup config server port is 0
Config server login is null
Application name is JacadaGW_10.9.8.9_7001
Agent Interaction Layer will work as Server
Period is 20 seconds.
Timeout is 21 seconds.
```

The following details are listed in the logs, as shown in the example below:

- A notice informing that a connection to the Genesys Configuration Server Protocol has been initialized
- Connection information regarding various Genesys components such as AIL and Telephony.

```
*** Initialized Genesys AIL ***
*** serviceStatusChanged: service_type=Stat; service_name=null; service_status=Absent
*** serviceStatusChanged: service_type=Is; service_name=IsService (no client); service_status=Absent
*** serviceStatusChanged: service_type=AIL; service_name=Agent Interaction Layer; service_status=On
*** serviceStatusChanged: service_type=Database; service_name=null; service_status=Absent
*** serviceStatusChanged: service_type=Config; service_name=confserv; service_status=On
*** serviceStatusChanged: service_type=Telephony; service_name=VoiceTServerSimulation; service_status=Off
*** serviceStatusChanged: service_type=Telephony; service_name=Generic TServer_651; service_status=Off
*** serviceStatusChanged: service_type=Stat; service_name=null; service_status=Absent
*** serviceStatusChanged: service_type=Is; service_name=IsService (no client); service_status=Absent
*** serviceStatusChanged: service_type=AIL; service_name=Agent Interaction Layer; service_status=On
*** serviceStatusChanged: service_type=Telephony; service_name=Generic TServer_651; service_status=On
*** serviceStatusChanged: service_type=Telephony; service_name=VoiceTServerSimulation; service_status=On
```

In a multi-site configuration, there is a Telephony line for each T-Server. In the above example, the T-Servers are: `Generic_TServer_651` and `VoiceTServerSimulation`.

Note: When connecting to Genesys CTI, exceptions may appear in the console or logs. Check the connectivity to your Genesys CTI. For more information, refer to [Genesys CTI](#).

Supported ISCC Transaction Type

T-Server supports multi-site operations using Inter Server Call Control (ISCC). As switches of different types provide calls with different sets of information parameters, a single mechanism for passing call data between the switches is not feasible in some cases. Therefore, the ISCC feature supports a number of mechanisms for passing call data along with calls between locations.

Jacada WorkSpace supports the default ISCC Transaction Type `route`, aliased as `route-notoken`.

Setting Up a List of Reasons for the Busy State (Aux Codes)

The CTI bar includes a label to the left of the call-handling buttons that provides a user-friendly interpretation of the busy reason or Aux code. The label is displayed when the agent is in *Not Ready* mode, after the agent has clicked the *Busy* button. To see the label displayed for the busy reason the agent must have the `BusyReasonCode` privilege.

You can set up the CTI system so that clicking the *Busy* button displays a list from which the agent needs to select the reason for not being ready to receive additional calls. If, when pressing *Busy*, the agent is in the process of handling a call, the *Not Ready* status takes effect only after the current call is completed.

Genesys supports specifying a set of reasons, each one associated with a reason code. By default, no reasons are defined, and clicking the *Busy* button uses the reason code '1', defined in the `CTIDefaultNotReadyReasonCode` in the database. When more than one reason is defined, clicking the *Busy* button displays the list of reasons for the agent to select from. If only one reason is defined, clicking *Busy* does not display the list.

The list of reasons is defined in the Jacada Management Console or in the `login` method as described in the Jacada WorkSpace Developer Guide. When the list of reasons is defined both in the `login` method and in the Jacada Management Console, the `login` method reasons take precedence over the Jacada Management Console settings.

To set up the busy reason in the Jacada Management Console:

1. In the Jacada Management Console navigation menu, select the *Configuration > CTI > Busy Reason Codes* option.

The *Busy Reason Codes* page displays.

2. Click *Add*.

The *Edit CTI Busy Reason* page displays.

3. On the page, enter the busy reason code, the busy reason description and the busy reason display name, and then click *Save*.
4. Set the `CTIDefaultNotReadyReasonCode` in the `CTI_SETTINGS` table to the value you want displayed when the agent does not select a value from the list, or when no busy reason code is defined.
5. In the Jacada Management Console navigation menu, select the *Authentication-Authorization > Groups* option.

The *Groups* page displays.

6. Click *Edit* next to the group to which the agents are assigned.

Note: The groups are mapped from the groups in the LDAP Active Directory.

The *Edit Groups* page displays, showing the details of the group, including the role assigned for the group.

7. In the Jacada Management Console navigation menu, select the *Authentication-Authorization > Roles* option.

The *Roles* page displays.

8. Click *Edit* next to the role to which the agent group is assigned.

The *Edit Roles* page displays, showing the list of privileges assigned for the role.

9. Make sure the role is assigned the `BusyReasonCode` privilege, and then click *Save*.

Using the Busy Reason in Jacada WorkSpace

When the agent logs on, the default status is `Not Ready`. In this case the default label defined in the `defaultBusyReason` field in the `CTI_SETTINGS` table is displayed.

When the agent is in `Ready` status and clicks the **Busy** button, a drop-down list is displayed with the list of busy reason codes defined in the database. The agent selects the relevant busy reason from the list. The selected reason is then displayed as a label to the left of the buttons.

Configuring Call Disposition

When an agent ends a call, the next state is determined by the CTI and switch configurations. Depending on CTI and switch configurations, the agent may or may not move to the *After-Call-Work* status. If agents are *not* in the *After-Call-Work* status, they can return to the previous state they were in before the call ended or alternatively, move to another preferred state.

Note: The *After-Call-Work* state defines that the agent cannot receive calls, but is currently working.

To enable call disposition to start automatically, where the agents moves to the *After-Call-Work* state automatically, the switch must be configured as follows:

- **Avaya:** Log in to the Avaya switch and enter the command `change hunt-group n`, where *n* is the hunt group relevant to the site. Also, set the `Timed ACW Interval (sec)` value, as follows:
 - To move to call disposition without a time limit, set the value to 9999. In the Genesys configuration, add the `tacw` parameter with this value (9999), under the relevant queue.
 - To turn off ACW, set the value to spaces.

Note: With the Avaya switch, these settings only apply to inbound calls.

Note: With Genesys and the Avaya switch and ACW enabled, the following agent status issue occurs: The agent is logged in and *Ready* (Inbound mode) and receives a call from the queue. If the caller disconnects while on hold, call-handling does not always end with call disposition. Instead, the agent is sometimes moved to the *Ready* state.

- **Alcatel:** Change the following T-Server flags:

```
wrap-up-time = 1000
```

```
real-agent-pause-time = 1000
outbound-bsns-calls = true
```

Configuring Custom Item Support

Before configuring Jacada WorkSpace to handle custom items, verify that the Genesys environment is configured to support Open Media items:

- You have a running Open Media server, that supplies Genesys with Open Media items, such as faxes.
- The Open Media Type is defined in the *CME > Business Attribute > Media Type*.
- There is a skill that represents the ability to handle Open Media items. This skill must be defined in the *CME > Skills*, and must be assigned to the agents (Persons) that need to handle Open Media items.
- There is routing that directs incoming work to the agents (Persons).
- All the agents (defined in Open Media as *Persons*) that have the skill to handle Open Media items have their *Capacity Rule* defined as 1, meaning that they can only receive one call or one custom item at a time.

Configuring the Database to Handle Custom Items

You configure Jacada WorkSpace to handle custom items in the `CTI_SETTINGS` table in the database. To work with custom items, the following settings are required:

DB Setting	Value
CTIEnableCustomItems	True
CTICustomItemNameList	A comma-separated list of the names of the Genesys Open Media types, as defined in Genesys CME. For example: WorkItem, Fax
CTICustomItemSkillName	The name of the Genesys skill that the agent must have in order to handle the Open Media types defined in <code>CTICustomItemNameList</code> .

Configuring the Answer Mode for Custom Items

A custom item can either be auto-answered by Jacada WorkSpace without a prompt, or else the agent receives an indication when a new call or custom item arrives and needs to accept the item by clicking the *Answer* button.

Configuring the answer mode for custom items can be done either in the database, or at the code level. When the configuration for all agents is the same, use the database. In projects that require more complex configuration, use the code level.

Note: The code level overrides the database setting. When the answer mode is not configured in the database or in the code, the default is `true`, meaning that custom items are automatically answered.

To configure the answer mode for custom items in the database:

- In the `CTI_SETTINGS` table, use the following setting: `CTICustomItemAutoAnswer`.
Set the value of this setting to `true` or `false` in the relevant profile (such as `CTI_GENESYS`).

Note: The developer can configure the answer mode at the code level. For details, refer to the Jacada WorkSpace Developer Guide.

Additional Genesys-Specific Configuration

To use Genesys with Jacada WorkSpace, the following configuration also needs to be performed:

- Creating a Genesys Application
- Defining Users for Jacada WorkSpace
- Associating Between an Agent and a Place
- Support for Agent Passwords at Switch Level
- Prefer-Ready Flag in a Multi- Channel Environment
- Configuration for Smart Pad Entry Transfer
- Moving to After-Call-Work Mode From the Phone
- After-Call-Work Mode in Genesys Alcatel

Creating a Genesys Application

To use Genesys, it is necessary to create an AIL application, in the Genesys Configuration Management Environment, that will serve as an access point for Jacada WorkSpace. The application must be based on the AIL Server version template, and must be named as follows:

```
JacadaGW_<Application IP>_<Application Port>
```

where the `IP` and `Port` refer to server machine running the Jacada WorkSpace CTI Gateway. When using clustering, it is necessary to create such an application for each server.

To create a Genesys application:

- In the Genesys *Configuration Manager* create a new application using the following guidelines:
 - a. Select the following template: *Agent_Interaction_server_720*
 - b. In the *General* tab, specify the application name, using the following format:
`JacadaGW_<Application IP>_<Application Port>`
For example: `JacadaGW_10.20.30.123_7001`
 - c. In the *Start Info* tab, specify a period (.) in both the *Working Directory* and *Command Line* fields.
 - d. In the *Connections* tab, specify the T-Server application to use with Jacada WorkSpace.
 - e. In the *Options* tab, configure an appropriate license for Genesys.
 - f. To obtain the location of the license file, contact your Genesys administrator.

Note: For information on configuring the Genesys application for a multi-site configuration, refer to [Support for a Multi-Site Configuration](#).

Defining Users for Jacada WorkSpace

When defining users for Jacada WorkSpace with Genesys, you must make sure that the same users (or persons) defined in Genesys are also defined in the authentication provider used by Jacada WorkSpace, such as Microsoft Active Directory. Note that user names are case-sensitive.

Associating Between an Agent and a Place

When an agent logs in to Jacada WorkSpace, it is necessary to associate between the agent and the place, which defines the extension used by the agent and other capabilities (such as e-mail) when available.

When fixed seating is used in the call center, each agent is associated with a place. The association between an agent and a place is defined in the Genesys Configuration Manager, where you define a Default Place for each agent (or person). When the agent logs in to Jacada WorkSpace, as part of the login to CTI, the place information is retrieved from Genesys.

When free seating is used in the call center, the place is defined per machine, not per agent. Defining the place per machine is done using the `MachineName` parameter in the URL that is used to activate Jacada WorkSpace. The `MachineName` parameter can be used in one of the following manners:

- Define places in Genesys according to the machine names (not machine IPs) used by the agents. In this manner, the same URL is used to activate Jacada WorkSpace on all agent machines. For example:

```
http://localhost:7001/MyAppJacada?MachineName=%MACHINENAME%
```

Note: When the `MachineName` parameter is assigned an environment variable, `%MACHINENAME%`, the URL must be passed as a command line argument to Internet Explorer. For example,

```
"C:\Program Files\Internet Explorer  
\iexplore.exe" http://localhost:7001/  
MyAppJacada?MachineName=%MACHINENAME%
```

- Specify the place name that is defined in Genesys as the value of the `MachineName` parameter in the URL. This option is useful when you want to use existing place definitions, but requires specifying a different URL on each machine. For example:

http://localhost:7001/MyAppJacada?MachineName=Place_123

Support for Agent Passwords at Switch Level

Some switches may require a password when logging in to the switch. The login to the switch is performed immediately after the login to Jacada WorkSpace. In order to supply a password, you need to add code to the `login` method of the application.

To set the agent password for the switch:

- In `<project_name>\Java Resources\src\USER.global.managers.ExtensionManagersjava`, add code to the `login` method such as the following:

```
@GlobalAction(actionName=GlobalActionNames.LOGIN,order=GlobalActionOrder.BEFORE)
public void login(HttpServletRequest request){
    globalApp.getCTIManager().getAgentInformation().setCtiAgentPassword("1234");
}
```

Prefer-Ready Flag in a Multi- Channel Environment

When working in a multi channel environment (which includes voice and custom items), the `CTIDisplayReadyWhenSingleChannelIsReady` parameter is used to show the ready state in Jacada WorkSpace, as follows:

- When `CTIDisplayReadyWhenSingleChannelIsReady` is set to `true`, and only one of the channels is available, the *Ready* button will be active.
- When `CTIDisplayReadyWhenSingleChannelIsReady` is set to `false`, and only one of the channels is available, the *Not Ready* button will be active.

Configuration for Smart Pad Entry Transfer

To enable Smart Pad entry transfer, configure the following options in the Genesys T-Server:

- `consult-user-data = inherited`
- `merged-user-data = merged-over-main`

Moving to After-Call-Work Mode From the Phone

From the telephone, agents can change their status to *After-Call-Work* while in the *Ready* or *Not Ready* states. Jacada WorkSpace goes into call disposition depending on the type of status. When the agent moves to the *After-Call-Work* status, the Jacada WorkSpace status changes to *After-Call-Work* status in *Wrap Up* and the call disposition script may or may not be displayed according to the required configuration. Refer to the Jacada WorkSpace Developer Guide for further details on configuring the disposition script. For each case the agent can do one of the following:

- Change original state
 - When the agent was in the *Ready* state before call disposition began, clicking the *Busy* button moves the agent to the *Not Ready* state after call disposition ends.
 - When the agent was in the *Not Ready* state before call disposition began, clicking the *Ready* button moves the agent to the *Ready* state after call disposition ends.

- Move to previous state

When call disposition default script **is** displayed, the agent clicks *Next* to end the call disposition and reverts to the previous state before the call disposition started.

When call disposition is **not** displayed, one of following APIs is used to end the call disposition and move the agent to the previous state before the call disposition started:

JavaScript:

```
$W().cti.onDispositionScriptEnded();
```

Java:

```
PushHelper.pushMessage  
(getSessionInformation().getSession(),  
"callDispositionEnded", "");
```

After-Call-Work Mode in Genesys Alcatel

The After-Call-Work mode in Genesys Alcatel in the handshake, warm transfer complete and consultation/transfer from consultation scenarios is the "special" ACW "In Wrap up". These states have the same behavior as when [moving to ACW from the phone](#).

Handling Loss of Connectivity to CTI

When an error occurs, such as loss of connectivity to CTI, or if Jacada WorkSpace is not synchronized with the CTI, Jacada WorkSpace switches to manual mode. The loss of connectivity is handled differently, according to the CTI provider.

Handling Loss of Connectivity to CTI in AES and CT Connect

When using AES or CT Connect and an error occurs, such as loss of connectivity to CTI, or if Jacada WorkSpace is not synchronized with the CTI, Jacada WorkSpace switches to manual mode. In manual mode, it is recommended to complete handling the current call, log out of Jacada WorkSpace, and then login again before starting to handle the next call.

Handling Loss of Connectivity to CTI in Cisco

When any Cisco component fails (PG, CTI server, CTIOS server, etc.), Cisco automatically moves the user to a temporary mode and then to the backup system.

If a Cisco component fails while an agent is logged in and is in one of the following states: *Ready*, *Not Ready*, *In Call*, *In Hold*, or *In Wrap Up*, the agent state changes to “*Initializing...*”, and the CTI bar is disabled.

When the failed component is restarted, or when Cisco backup components become active, the agent is synchronized with the previous CTI state.

Note: Failover is not supported during consult/conference/warm transfer.

Handling Loss of Connectivity to CTI in Genesys

When using Genesys, the handling of the lost CTI connectivity depends on the reason for the following problem:

- When the connection to CTI is lost because the Genesys T-Server is down or is offline:

When the connection to CTI is lost, Jacada WorkSpace switches automatically to manual mode.

Once the connection to CTI is recovered and the T-Server is running and accessible, Jacada WorkSpace displays a *Reconnect Phone* button in the call-handling bar.

To continue working with CTI once again, the agent needs to click the *Reconnect Phone* button. However, it is not recommended to click the *Reconnect Phone* button while in the process of handling a call, as information may be lost as a result of the reconnection. For example, when the agent ends the call the disposition is started, but without a script and has to click the *Busy/*

Ready button to change the state.

After pressing the *Reconnect Phone* button, a disabled *Reconnecting...* button is displayed, until the CTI connection is recovered.

Note: To enable the reconnect mechanism, the setting **CTIEnableReconnect** in the `CTI_SETTINGS` table in the database, must be set to `true` in the profile in use.

Starting and Stopping the Application Server

Jacada WorkSpace runs on either the IBM WebSphere, Oracle WebLogic or the Apache Tomcat server. The server configuration in a production environment can be either a cluster configuration or a single server instance. Jacada WorkSpace is certified to run on Windows and Linux RedHat host machines.

The following topics describe how to start and stop a specific application server:

- [Starting the Tomcat Server](#)
- [Starting and Stopping the WebLogic Server](#)
- [Starting and Stopping the WebSphere Application Server](#)

Starting the Tomcat Server

The following procedures explain how to start the Apache Tomcat server. The procedures vary according to operating system.

From the command line or linux console, navigate to the <Apache Tomcat Home Directory>/bin folder and run the following command:

For windows:

```
catalina.bat run
```

For linux:

```
./catalina.sh run
```

Starting and Stopping the WebLogic Server

The procedures for starting and stopping the Oracle WebLogic Server running Jacada WorkSpace vary, based on the server configuration used in the production environment:

- [Single Server Configuration](#)
- [Clustered Server Configuration](#)
- [Verify the Jacada WorkSpace Application Modules](#)

Jacada WorkSpace runs on the WebLogic Server that is hosted in either a Windows machine or a Linux RedHat machine. Stopping and starting the server configuration is done from either the host machine's command line or via the WebLogic Server Administration Console.

The WebLogic Server stop and start commands (issued from the command line) for a Windows based host machine and a Linux RedHat based host machine are identical except for the filename extension:

- **Windows:** `<command_name>.cmd`
- **Linux Redhat:** `<command_name>.sh`

Single Server Configuration: Start and Stop Server

For a single server configuration of WebLogic Server (also referred to as a stand-alone configuration), the server startup and shutdown commands are:

- `startWebLogic`
- `stopWebLogic`

These commands are located in the following directory:

<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin

where **<OracleWebLogic_Home>** is the top-level installation directory for the Oracle WebLogic software.

For example, in a Windows host machine, **C:/Oracle\Middleware**.

Starting a WebLogic Server

The following procedures explain different ways to start the server. After the initial start of WebLogic Server following its setup, verify that the Administration Server has its general logging settings

appropriately configured for your production environment. Pay particular attention to the severity level setting of each message destination.

View and update these logging settings in the WebLogic Server Administration Console, as follows:

- In the *Domain Structure* navigation pane, select *Environment* > *Servers* > *Servers* table <server_name> link > *Logging* tab > *General* page.

To start a WebLogic Server on a Windows host machine via the Start menu:

- *Start* > *Programs* > *Oracle WebLogic* > *User Projects* > <domain_name> > *Start Admin Server for WebLogic Server Domain*

A command window opens. In the window, startup messages display. The WebLogic server instance is started. This server instance functions as both the Administration Server (*AdminServer*) and the Managed Server. The WebLogic Server Administration Console is available for server administration.

To start a WebLogic Server from the command line:

1. In the host machine where your server instance is installed, change your current directory to the domain bin directory of the server you want to start, **<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin**.
2. From the command line, run the following command:

```
startWebLogic
```

The WebLogic server instance is started. This server instance functions as both the Administration Server (*AdminServer*) and the Managed Server. The WebLogic Server Administration Console is available for server administration.

Stopping a WebLogic Server

The following procedures explain different ways to stop the server.

To stop a WebLogic Server on a Windows host machine via the Start menu:

- *Start* > *Programs* > *Oracle WebLogic* > *User Projects* > <domain_name> > *Stop Admin Server*

A command window opens. In the window, shutdown messages display. The WebLogic server instance, including both its Administration Server and Managed Server functions, is stopped and the WebLogic Server Administration Console is not available.

To stop a WebLogic Server from the command line:

1. In the host machine where your server instance is installed, change your current directory to the domain bin directory of the server you want to stop, **<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin**.
2. From the command line, run the following command:

```
stopWebLogic
```

The WebLogic server instance, including both its Administration Server and Managed Server functions, is stopped and the WebLogic Server Administration Console is not available.

Clustered Server Configuration: Start and Stop Servers

For a clustered server configuration of WebLogic Server, the server startup and shutdown commands are as follows:

- `startWebLogic`
- `startManagedWebLogic`
- `stopWebLogic`
- `stopManagedWebLogic`

These commands are located in the following directory: **<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin**, where **<OracleWebLogic_Home>** is the top-level installation directory for Oracle WebLogic software.

For example, in a Windows host machine, **C:/Oracle\Middleware**.

- `startNodeManager`

This command is located in the following directory: **<WL_Home>/server/bin**, where **<WL_Home>** is the top-level installation directory for WebLogic Server. For example, in a Windows host machine this directory could be **C:/bea/wlserver_10.0**.

For a description of the server components associated with a WebLogic Server cluster, refer to [Installation of a Clustered Production Environment](#).

Starting Server Components

Start the WebLogic Server cluster configuration in the following order:

1. The Administration Server

Note: In a Windows host machine, the cluster's Administration Server can be configured to run as a Windows service. For information about configuring the Administration Server to run as a Windows service, contact Jacada Support.

2. The Node Manager
3. Each Managed Server

To start the Administration Server and Node Manager on a Windows host machine:

1. Select *Start > Programs > Oracle WebLogic > User Projects > <domain_name> > Start Admin Server for WebLogic Server Domain*.

A command window opens. In the window, startup messages display. The Administration Server server instance, *AdminServer*, is started. The WebLogic Server Administration Console is available for server administration.

2. Select *Start > Programs > Oracle WebLogic > WebLogic Server 11g > Tools > Node Manager*.

A command window opens. In the window, startup messages display. The Node Manager process is started.

To start the Administration Server and Node Manager from the command line:

1. In the host machine where your Administrator Server instance is installed, from the command line run the following command:

```
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin./startWebLogic
```

2. When prompted by the start process, enter the cluster domain's Administrator user name and password.

The Administration Server server instance, *AdminServer*, is started. The WebLogic Server Administration Console is available for server administration.

3. Start the Node Manager by running the following command from the command line:

```
<WL_Home>/server/bin/startNodeManager
```

The Node Manager process is started. One Node Manager process is started per host machine of WebLogic Server.

To start the Managed Servers:

1. In the host machine for each Managed Server instance, from the command line, run one of the following commands:

- If the Managed Server being started and the Administration Server reside on the same host machine:

```
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin/startManagedWebLogic
```

- If the Managed Server being started and the Administration Server reside on different host machines:

```
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin/startManagedWebLogic
```

where:

`host` is either the IP or machine name of the machine hosting the Administration Server.

`Port` is the *Listen Port* of the Administration Server in the cluster.

2. When prompted by the start process, enter the cluster domain's Administrator user name and password.

The Managed Server server instance, `<managed_server_name>`, is started.

3. Repeat Steps 1 and 2 to start each additional Managed Server in the cluster domain.

The WebLogic Server cluster configuration is started.

After the initial start of a WebLogic Server cluster following its setup, verify that the started server instances, these being the Administration Server and any Managed Servers, have their general logging settings appropriately configured for your production environment. Pay particular attention to the severity level setting of each message destination.

View and update these logging settings in the WebLogic Server Administration Console, as follows:

- In the *Domain Structure* navigation pane, select *Environment > Servers > Servers* table `<server_name>` link > *Logging* tab > *General* page.

Note: The first time you start the cluster, the connection test must be configured:

Launch the weblogic admin console and navigate to Services->Data Sources.

Click the JACADA_JDBC_DB data source, select the connection pool tab, click Advanced and Change Test Table Name property value to SQL SELECT 1 FROM DUAL.

Repeat the above steps for metadataDS data source.

Stopping Server Components

The following procedures explain how to stop components of the WebLogic Server cluster configuration.

Stopping the WebLogic Server cluster can be done in any order. When stopping a cluster, there are no dependencies among any server components (the Administrative Server, the Node Manager, and the Managed Servers). Stopping a server component does not affect any of the other components.

Important: If you kill the JVM, the server immediately stops all processing. In this case, navigate to your database `server_connection_information` table and delete all records.

To stop the Administration Server on a Windows host machine:

- Select *Start > Programs > Oracle WebLogic > User Projects > <domain_name> > Stop Admin Server*.

A command window opens. In the window, shutdown messages display. The Administration Server server instance, *AdminServer*, is stopped. The WebLogic Server Administration Console is not available.

To stop the Administration Server from the command line:

1. In the host machine where your Administrator Server instance is installed, change your current directory to the domain bin directory of the cluster you want to stop:
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin.
2. From the command line, run the following command:

```
stopWebLogic
```

3. When prompted by the stop process, enter the cluster domain's Administrator user name and password.

The Administration Server server instance, *AdminServer*, is stopped. The WebLogic Server Administration Console is not available.

To stop the Node Manager and Managed Servers:

1. Stop the Node Manager process by closing the command shell (Linux RedHat host machine) or the command window (Windows host machine) in which the process is running.
2. In the host machine where each Managed Server instance is installed, from the command line, run one of the following commands:
 - If the Managed Server being started and the Administration Server reside on the same host machine:

```
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin/stopManagedWebLogic
```

- If the Managed Server being started and the Administration Server reside on different host machines:

```
<OracleWebLogic_Home>/user_projects/domains/<domain_name>/bin/stopManagedWebLogic
```

where:

host is either the IP or machine name of the machine hosting the Administration Server.

Port is the *Listen Port* of the Administration Server in the cluster.

3. When prompted by the stop process, enter the cluster domain's Administrator user name and password.

The Managed Server server instance, *<managed_server_name>*, is stopped.

4. Repeat Steps 2-3 to stop each additional Managed Server in the cluster domain.

The WebLogic Server cluster configuration is stopped.

Verify the Jacada WorkSpace Application Modules

After starting the WebLogic Server for a domain, verify that the deployed Jacada WorkSpace application modules have the status *Active*. For a WebLogic single server configuration, verify only after the Administration Server, *AdminServer*, has the status *Running*. For a WebLogic Server cluster configuration, verify only after all of the cluster's Managed Servers have the status *Running*.

The following procedure explains how to verify application module status. Performing this procedure

requires administrator level user privileges.

To verify Jacada WorkSpace application module status:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Deployments*.

The *Summary of Deployment* page displays.

2. Verify the following:

- In the *Deployments* table, the following Jacada WorkSpace application modules display with a state of *Active*:
 - o CTIGatewayEAR
 - o JacadaSessionManager
 - o JacadaWorkSpaceCommonEAR
 - o dynamciviews.war
 - o JacadaManagentConsole.ear
- If a Jacada WorkSpace project was previously deployed to the server, then the project's application module `<project_nameEAR>` displays with a state of *Active*.

After verifying that the Jacada WorkSpace application modules, including the project's application module, have the status *Active*, you can access and log in to JacadaWorkSpace.

Note: When it is necessary to redeploy an application (e.g., after installing a patch), you need to specify the deployment order. When redeploying the Jacada Management Console, the deployment order should be 200, so the MC will be loaded last.

Starting and Stopping the WebSphere Application Server

The procedure for starting and stopping the IBM WebSphere Application Server running Jacada WorkSpace varies, based on the configuration used in the production environment:

- Cluster of servers (network deployment)
- Single server instance

In the procedures that follow, commands that are issued from a command window are similar for a Windows host machine and a Linux RedHat host machine. The only difference is in their filename extension, that is `<command_name>.bat` for Windows and `<command_name>.sh` for Linux RedHat.

Clustered Server Environment: Stop and Start Servers

After applying configuration and application updates to the WebSphere cluster for your Jacada WorkSpace production environment, you need to stop and then start all the WebSphere network deployment elements, specifically the deployment manger, the managed nodes, the cluster and its cluster members (the application servers running the Jacada WorkSpace application modules). Performing this stop and restart ensures that all updates are in effect across the WebSphere cluster.

Remember that in a WebSphere cluster, managed nodes can reside either on the same machine as the deployment manager server and cluster or on a remote machine.

If you need to stop and start all the WebSphere network deployment elements, perform the procedures in the order presented. If you only need to stop and start the WebSphere cluster, perform only the procedures relevant to stopping and starting the cluster.

To stop the WebSphere cluster:

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*. The *Servers clusters* page displays.

Note: Performing this procedure requires administrator level user privilege.

2. In the table listing the existing clusters, check the relevant cluster and then click *Stop*. The selected cluster and all its cluster members (application servers) stop running.
3. Verify the status of the WebSphere cluster. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*. The *Servers clusters* page displays and the *Status* field of the relevant cluster contains the stopped indicator, .
4. Verify the status of the cluster members. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Application servers*. The *Application servers* page displays and the *Status* fields of the application servers associated with the relevant WebSphere cluster contain the stopped indicator, .

To stop the Managed Nodes and the Deployment Manager:

1. Open a command window.
2. Change your current directory to the profile bin folder of the managed node you want to stop,

```
<WebSphere_app_server_home_folder>/profiles/<node_profile_name>/bin.
```

3. Run the following command:

```
stopNode
```

WebSphere requests you to enter the administrator user name and password. After you provide these details, the `stopNode` request completes and the server *nodeagent* is stopped for the managed node.

4. Repeat Step 2 through Step 3 to stop the additional managed nodes installed for this WebSphere network deployment.
5. Change your current directory to the profile bin folder of the deployment manger you want to stop, `<WebSphere_app_server_home_folder>/profiles/<deployment_manager_profile_name>/bin.`

6. Run the following command:

```
stopManager
```

The deployment manger server *dmgr* is stopped.

At this point, the cluster and all its cluster members, the managed nodes and the deployment manager of the WebSphere network deployment are stopped.

To start the Deployment Manager and the Managed Nodes:

1. Open a command window.
2. Change your current directory to the profile bin folder of the deployment manger you want to start, `<WebSphere_app_server_home_folder>/profiles/<deployment_manager_profile_name>/bin.`

3. Run the following command:

```
startManager
```

In the command window, the following system output messages are generated:

```
Starting tool with the <deployment_manager_profile_name>
profile
Reading configuration for server: dmgr
Server launched. Waiting for initialization status.
Server dmgr open for e-business; process id is <process_id>
```

The deployment manger server *dmgr* is started.

4. Change your current directory to the profile bin folder of the managed node you want to start, `<WebSphere_app_server_home_folder>/profiles/<node_profile_name>/bin.`
5. Run the following command:

startNode

In the command window, the following system output messages are generated:

```
Starting tool with the <node_profile_name> profile
Reading configuration for server: nodeagent
Server launched. Waiting for initialization status.
Server nodeagent open for e-business; process id is
<process_id>
```

The server *nodeagent* is started for the managed node.

6. Repeat Step 4 through Step 5 to start the additional managed nodes installed for this WebSphere network deployment.

At this point, the deployment manager and all the managed nodes of the WebSphere network deployment are started.

To start the WebSphere cluster:

1. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*.
The *Servers clusters* page displays.

Note: Performing this procedure requires administrator level user privilege.

2. In the table listing the existing clusters, check the relevant cluster and then click *Start*.
The selected cluster and its cluster members (application servers) start running. It will require several minutes for the cluster members to change their status to started.
3. Verify the status of the WebSphere cluster. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Clusters*.
4. The *Servers clusters* page displays and the *Status* field of the relevant cluster contains the started indicator, .
5. Verify the status of the cluster members. In the task menu of the WebSphere Integrated Solutions Console, select *Servers > Application servers*.
The *Application servers* page displays and the *Status* fields of the application servers associated with the relevant WebSphere cluster contain the started indicator, .

At this point, the cluster and all its cluster members, the deployment manager and all the managed nodes of the WebSphere network deployment are started.

Single Server Instance: Stop and Start Server

After applying configuration and application updates to the WebSphere Application Server for your Jacada WorkSpace production environment, you need to stop and then start the server to have the updates be in effect for the WebSphere single server instance.

To stop the WebSphere server:

1. Open a command window.
2. Change your current directory to the profile bin folder of the server you want to stop,
`<WebSphere_app_server_home_folder>/profiles/<server_profile_name>/bin.`
3. Run the following command:

```
stopServer <server_name>
```

Note: The default WebSphere server name for a single server instance is `server1`.

WebSphere requests you to enter the administrator user name and password. After you provide these details, the `stopServer` request completes. In the command window, the following system output messages are generated:

```
Starting tool with the <server_profile_name> profile
Reading configuration for server: server1
Server stop request issued. Waiting for stop status.
Server server1 stop completed.
```

The server `<server_name>` is now stopped.

To start the WebSphere server:

1. Open a command window.
2. Change your current directory to the profile bin folder of the server you want to start,
`<WebSphere_app_server_home_folder>/profiles/<server_profile_name>/bin.`
3. Run the following command:

```
startServer <server_name>
```

Note: The default WebSphere server name for a single server instance is `server1`.

In the command window, the following system output messages are generated:

```
Starting tool with the <server_profile_name> profile
Reading configuration for server: server1
Server launched. Waiting for initialization status.
Server server1 open for e-business; process id is
<process_id>
```

The server `<server_name>` is now started.

Verify the WebSphere Application Server and the Jacada WorkSpace Application Modules

To verify the status of the single instance WebSphere Application Server:

1. Open a command window.
2. Change your current directory to the profile bin folder of the server whose status you want to verify, `<WebSphere_app_server_home_folder>/profiles/<server_profile_name>/bin`.
3. Run the either of the following server status commands:

```
serverStatus <server_name>
```

Note: The default WebSphere server name for a single server instance is `server1`.

WebSphere requests you to enter the administrator user name and password. After you provide these details, the server status request completes.

In the command window, either one of the following system output messages are generated:

```
Starting tool with the <server_profile_name> profile
Retrieving server status for server: server1
The Application Server "server1" is STARTED.
```

or

```
Starting tool with the <server_profile_name> profile
Retrieving server status for server: server1
The Application Server "server1" is cannot be reached. It
appears to be stopped.
```

To verify the Jacada WorkSpace application modules status:

1. In the task menu of the WebSphere Integrated Solutions Console, select *Applications >*

Enterprise Applications.

Note: Performing this procedure requires administrator level user privilege.

The *Enterprise Applications* page displays..

Note: When the WebSphere Application Server is stopped, the WebSphere Integrated Solutions Console for the specific server cannot be accessed.

2. In the *Application Status* fields of the Jacada WorkSpace application modules – CTIGatewayEAR, JacadaSessionManager and JacadaWorkSpaceCommonEAR – the started indicator, , displays.

Deploying an Application to a Production Server

To run in a production environment, Jacada WorkSpace projects are deployed to one or more production servers, that is either to a single server instance or to a cluster.

Each Jacada WorkSpace project is installed as a new application in its target production servers, in the following form:

- **Tomcat:** WAR archive file.
- **WebSphere:** EAR archive file. Project EAR archive files are created in the Jacada WorkSpace project workspace of the eclipse IDE.
- **WebLogic:** EAR archive file. Project EAR archive files are created in the Jacada WorkSpace project workspace of the eclipse IDE.

The following sections describe:

- [Deploying an Application \(Tomcat\)](#)
- [Deploying an Application \(WebSphere\)](#)
- [Deploying an Application \(WebLogic\)](#)

Deploying an Application (Tomcat)

Prior to deploying a Jacada WorkSpace project to one or more production Tomcat servers, the following is required:

- A WAR archive file, `<project_name>.war`, exists for the project.

For details about creating WAR archive files and, separately, exporting them from the IDE, refer to the Jacada WorkSpace Developer Guide.

- The production Tomcat server must be started. If Jacada WorkSpace runs in a Tomcat cluster, the cluster and all its cluster members must be started.

To deploy an application:

- Copy the WAR file to the **webapps** folder, located in the **<Tomcat_home>** directory.

Deploying an Application (WebSphere)

Prior to deploying a Jacada WorkSpace project to one or more production WebSphere Application Servers (WAS), the following is required:

- An EAR archive file, `<project_nameEAR>.ear`, exists for the project.
For details about creating EAR archive files and, separately, exporting them from the IDE, refer to the Jacada WorkSpace Developer Guide.
- The production WAS must be started. If Jacada WorkSpace runs in a WAS cluster, the cluster and all its cluster members must be started.

To deploy an application:

1. From an Internet Explorer browser, access WebSphere Integrated Solutions Console via the administrative console port established for the relevant WAS deployment manager.
2. In the task menu of the Integrated Solutions Console, select the *Applications > Install New Applications* menu option.
The *Preparing for the application installation* page displays.
3. In the *Path to the new application* pane, select the applicable file system and enter the full path to the location of the Jacada WorkSpace project's EAR archive file.
4. Click *Next*.
The *Select installation options* window displays.
5. Accept the default option settings and click *Next*.
The *Map modules to Servers* window displays.
6. In the list of *Clusters and Servers*, do the following:
 - a) If the production environment is a WAS single server instance, select the relevant server entry where you are installing the application modules.
 - b) If the production environment is a WAS cluster, select the relevant cluster where you are installing the application modules.

- c) If the production environment incorporates Apache Web Servers, select the relevant Apache Web Server entry where you are installing the application modules.
7. In the Map modules to servers table, check the relevant *Module* to install in the selected servers (WAS single server/WAS cluster, Apache Web Server) and click *Apply*.
The Map modules to servers table displays containing the updated module to server entry.
8. Click *Next*.
The *Summary* window displays.
9. Click *Finish*.
The installation proceeds and status messages displays. When the installation successfully completes the message *Application <application_name> installed successfully*.
10. Click the *Review* link and do the following:
 - a) Expand the *Total changed documents* entry to review the list of changed items.
 - b) Click *Save*.
The system updates the master repository configuration with the application modules that were installed in the selected server or cluster.
11. In the task menu of the Integrated Solutions Console, select the *Applications > Enterprise Applications* menu option.
The *Enterprise Applications* page displays. The newly installed application appears in the table of installed applications.
12. In the table, check the newly installed application entry and click *Start*.
After a brief processing delay, the *Application Status* of the newly installed application displays a green circle indicating that it is started.

Deploying an Application (WebLogic)

Prior to deploying a Jacada WorkSpace project to one or more production WebLogic Servers, the following is required:

- An EAR archive file, <project_nameEAR>.ear, exists for the project.
For details about creating EAR archive files and, separately, exporting them from the IDE, refer to the Jacada WorkSpace Developer Guide.
- The production WebLogic Server must be started. If Jacada WorkSpace runs in a WebLogic Server cluster, all the cluster's managed servers must be started.

Following the deployment of the Jacada WorkSpace project as an Enterprise application, start it in

the server instances of the WebLogic Server domain.

The following procedures explain how to deploy an application to a production server, and start the deployed Enterprise application in the server. Performing these procedures requires administrator level user privileges.

To deploy an application to a production server:

1. Access the Oracle WebLogic Server Administration Console.
2. In the *Domain Structure* navigation pane of the console, select *Deployments*.
The *Summary of Deployments* page displays and contains the *Deployments* table. The table lists the application modules that are installed in the WebLogic Server domain.
3. In the *Change Center* pane, click *Lock & Edit*.
The *Deployments* table re-displays with its *Install* button enabled.
4. Click *Install*.
The *Install Application Assistant* page displays.
5. If the Jacada WorkSpace project's EAR archive file resides in the local machine, use the *Location* table and select the full path to the folder containing the desired file.
The *Location* field displays the selected path, and the *Location* table displays all the EAR archive files found in the folder.
6. If the Jacada WorkSpace project's EAR archive file is remotely located, use the upload your file(s) link to access the desired file and upload it to the Admin Server.
 - a. In the *Deployment Archive* field, click *Browse* and locate the EAR archive file to deploy.
 - b. In the *Deployment Archive* field, enter the full path to the file.
 - c. Click *Next*.
The *Install Application Assistant* page redisplay with the following message: *The file <project_name>EAR.ear has been uploaded successfully to <OracleWebLogic_Home>/user_projects/domains/<domain_name>/servers/AdminServer/upload.*
The *Location* field displays the upload folder path and the *Location* table displays all the EAR archive files found in the upload folder.
7. In the *Location* table, select the EAR archive file you want to install and click *Next*.
The *Install Application Assistant* page re-displays with the *Choose* targeting style options.
8. Select the installation target option to use (*Install as an application* is most commonly used) for installing the EAR archive file and click *Next*.
The *Install Application Assistant* page re-displays with the *Select deployment target* options.

9. From the available installation targets, select the servers and/or clusters to which you want to deploy the specified EAR archive file. Click *Next*.

The *Install Application Assistant* page re-displays with the *Optional Settings* for modification.

10. Accept the default settings as presented and click *Next*.

The *Install Application Assistant* page re-displays with a summary of your deployment selections to be used when installing the EAR archive file.

11. Click *Finish*.

The *Settings for <project_nameEAR>* page displays and presents the Enterprise Application's general configuration information and all its modules and components.

12. Click *Save*.

The following message displays: *Settings updated successfully*. Pending configuration changes exist for the WebLogic Server domain.

13. In the *Change Center* pane, click *Activate Changes*.

The following message displays: *All Changes have been activated*.

The Weblogic Server domain configuration is updated such that the EAR archive file, *<project_nameEAR>*, is installed as an Enterprise application in the specified server and/or cluster targets. The deployment is complete.

To start the deployed Enterprise application in the production server:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Deployments*.

The *Summary of Deployments* page displays. In the *Control* tab, the newly installed application, *<project_nameEAR>*, appears in the *Deployments* table.

2. In the *Deployments* table, verify that the state of the application is *Prepared*.

3. In the *Deployments* table, check the application and click *Start* and select the *Servicing all requests* option.

The *Start Application Assistant* page displays and asks you to confirm your request to start the deployed Enterprise application.

4. Click *Yes*.

The *Summary of Deployments* page displays with the following message: *Start requests have been sent to the selected Deployments*. In the *Control* tab's *Deployments* table, the state of the application is *start Running*.

5. In the *Domain Structure* navigation pane, select *Deployments* to re-display the *Control* tab's

Deployments table.

The state of the application is *Active*, which indicates that the deployed Enterprise application is started in all running server instances of the WebLogic Server domain.

Using the Jacada Management Console

The Jacada Management Console is a Web-based application that enables administration, configuration and monitoring of the Jacada WorkSpace runtime environment.

Using the Jacada Management Console, you can configure Jacada WorkSpace global settings, logging and RSS Feeds as well as view session statistics. The Jacada Management Console is deployed on the same application server on which your Jacada WorkSpace is deployed; this can be either a single application server instance or an application server in a cluster of managed application servers.

The following topics describe:

- [Accessing the Jacada Management Console](#)
- [Jacada Management Console User Interface](#)
- [Jacada WorkSpace Management Console Functions](#)
- [Jacada WorkSpace Configuration](#)
- [Logging](#)
- [Monitoring Jacada WorkSpace](#)

Accessing the Jacada Management Console

The Jacada Management Console and Jacada WorkSpace run in the same application servers. Access the Jacada Management Console using one of the following methods:

- **Via Internet Explorer:** Use a Microsoft Internet Explorer browser and supply the applicable URL. This access method is available for both cluster and single server configurations. This method provides both local and remote access of the management console.
- **Via the WorkSpace IDE:** This method allows you to access the Jacada Management Console directly from the development environment, when using an Apache Tomcat or an IBM WebSphere server.

To access the Jacada Management Console via Internet Explorer:

1. In the browser's URL field, enter the following:

`http://<host>:<Port>/mc`

where:

host is either the IP or machine name of the machine hosting the Jacada Management Console.

In a cluster configuration, *Port* is the web container default host port, `WC_defaulthost`, of any application server in the cluster.

2. In a single server configuration, *Port* is the HTTP transport port of the application server.
3. In the Jacada Management Console *Login* dialog, enter the *user name* and *password* established for the Jacada Management Console administrator. The default values are *admin*, *admin*.

To access the Jacada Management Console from the development environment:

1. From the Eclipse IDE, right-click on the relevant project, and select *Jacada WorkSpace> Management Console*.

The Jacada Management Console *Login* page appears in a browser page.

2. Log in, using the credentials established for the Jacada Management Console administrator. The default values are *admin*, *admin*.

Jacada Management Console User Interface

The Jacada Management Console contains the following elements:

Console Explorer

A navigation menu for moving through the Jacada Management Console pages. The current menu option is displayed in bold font. Clicking the current option refreshes the current page.

Help link

Displays help about the Jacada Management Console.

Logout link

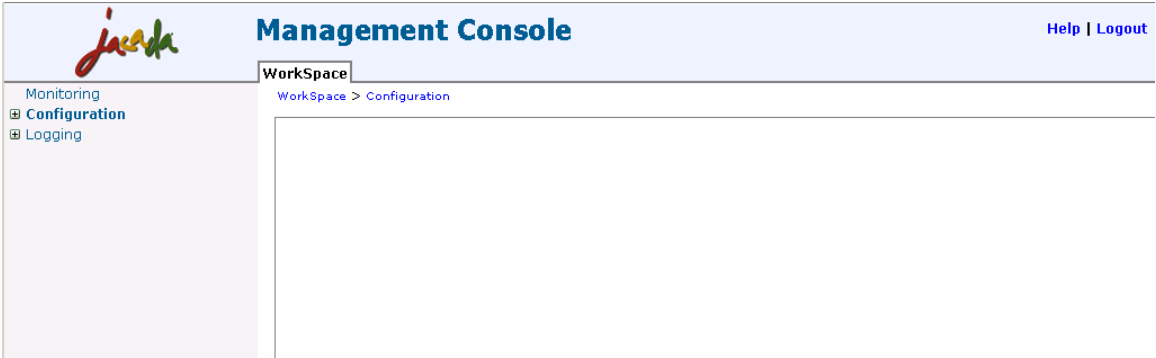
Logs you out of the Jacada Management Console.

WorkSpace tab

The main working area, where you configure and monitor Jacada WorkSpace. For more information, refer to [The WorkSpace Tab](#).

Breadcrumb navigation

Displays the path to the currently displayed topic.



The Workspace Tab

The Workspace tab enables you to configure, manage and monitor the Jacada Workspace environment.



Management Console

Help | Logout

Workspace

Monitoring

Configuration

- Dynamic Views Settings
- External Script Settings
- Feature Settings
- Global Settings
- Instant Messaging Settings
- Locales
- Machine Settings**
- Page Mappings
- RSS Feed Definitions
- RTN Mappings
- Script Mappings
- Smart Pad Settings
- Supervisor View Settings
- Tabs Configuration
- Transport
- WinFuse Distributed Services

Workspace > Configuration > Machine Settings

Show table properties >>

☐	Host▼	Port	Name	Value	Comments	Active	Action
☐	*	*	PROFILES	MANUAL		true	Edit

Add

Remove

When the Workspace tab displays a table, the *Show table properties* link appears above the upper left corner of the table. Clicking this link displays the *Table Properties* pane, which enables you to view table properties and customize the way in which the table is displayed.



Management Console

Monitoring

Configuration

Dynamic Views Settings

External Script Settings

Feature Settings

Global Settings

Instant Messaging Settings

Job Assignments

Locales

Machine Settings

Page Mappings

RSS Feed Definitions

RTN Mappings

Script Mappings

Smart Pad Settings

Supervisor View Settings

WinFuse Distributed Services

Agent Disposition

Auditing

Authentication-Authorization

CTI

Document Search

Layout Management

Layouts

Portlets

Tabs

Transport

Project Variables

Task Manager

Logging

Workspace

Workspace > Configuration > RTN Mappings

<< Hide table properties

Displayed columns:

Project ID

RTN

Script ID

>>

>

<

<<

Undisplayed columns:

Window Snip

Filter column:

RTN

Filter value:

Max items per page:

10

Apply

	Project ID ▼	RTN	Script ID	Action
☐	*	1234567	NONE	Edit
☐	*	8888888	NONE	Edit

Add

Remove

Table Properties Pane Options

The *Table Properties* pane options are described in the following table.

Option	Description/Notes
Displayed columns/Undisplayed columns frames	Determine which of the available columns actually appear in the table. Select relevant column names, and use the buttons between the two frames to customize column appearance in the table. Then, click <i>Apply</i>.

Option	Description/ Notes
<i>Filter column drop-down/Filter value textbox</i>	Enable you to filter the table to display only rows in which a specified value appears. Select a column name from the drop-down list, and enter the relevant filter string in the textbox. Then, click <i>Apply</i>.
<i>Max items per page drop-down</i>	Determines the maximum number of rows displayed in each page of the table. Select a value from the drop-down list, and then click <i>Apply</i>.
<i>Hide table properties link</i>	Hides the <i>Table Properties</i> pane.

Jacada WorkSpace Management Console Functions

Use the Jacada Management Console to configure Jacada WorkSpace. The options of the navigation menu provide access to the following:

- [Jacada WorkSpace Configuration](#)
- [Logging](#)
- [Monitoring Jacada WorkSpace](#)

Jacada WorkSpace Configuration

Using the Jacada Management Console navigation menu option *Configuration*, configure the following:

- [Directory Settings](#)
- [Dynamic Views Settings](#)
- [EMC Settings](#)
- [External Script Settings](#)
- [Feature Settings](#)
- [Global Settings](#)
- [Human Task Settings](#)
- [Instant Messaging Settings](#)
- [Interact Settings](#)
- [Locales](#)
- [Machine Settings](#)
- [Page Mappings](#)
- [RSS Feed Definitions](#)
- [RTN Mappings](#)
- [Script Mappings](#)
- [Smart Pad Settings](#)
- [Supervisor View Settings](#)

The following nodes have entries under them:

- [Agent Disposition](#)
- [Auditing](#)
- [Authentication and Authorization](#)
- [CTI](#)
- [Document Search](#)
- [Layout Management](#)
- [Project Variables](#)
- [Task Manager](#)

Directory Settings

Use the *Directory Settings* page to update the search provider type used for searching contacts and customers.

Field	Description
Contact Search Provider Type	The type of directory search provider. Valid values are: • <code>ldap</code> • <code>custom</code> (This type is implemented by Professional Services, in CustomSearchProvider.java.) • <code>emc</code> (Searches for contacts in the Avaya EMC database)
Conversation History Search Provider Type	The type of Conversation History search provider. Valid values are: • <code>custom</code> (This type is implemented by Professional Services, in CustomConversationSearchProvider.java.) • <code>emc</code> (Searches for customer conversation history in the Avaya EMC database)
Customer Auto Search	This flag indicates whether or not an automatic customer search is performed when a new email, chat or business call is accepted by an agent.

Field	Description
Customer Search Provider Type	The type of search provider used for performing customer searches. Valid values are: <code>custom</code> (This type is implemented by Professional Services, in CustomSearchProvider.java.) <code>emc</code> (Searches for contacts in the Avaya EMC database)

Dynamic Views Settings

Use the *Dynamic Views Settings* page to update the credentials used to log into Jacada Dynamic Views from Jacada WorkSpace.

Field	Description
Dynamic Views password	The password with which Jacada WorkSpace logs into Jacada Dynamic Views.
Dynamic Views user	The username with which Jacada WorkSpace logs into Jacada Dynamic Views.
Jacada WorkSpace session timeout	The time period, in minutes, after which a Jacada WorkSpace session is automatically terminated, following no interaction with the Dynamic Views designer.

EMC Settings

Use the *EMC Settings* page to update EMC related settings.

Field	Description
Database Host	The EMC database host.

External Script Settings

Use the *External Script Settings* page to control both the function and display of an external, side-by-side, script.

Field	Description
ExternalScriptAreaCollapsedWidth	Defines the size of a collapsed external script area expressed as a percentage of the tab's total area. This setting is only relevant when the <code>SideBySideLayout</code> setting is set to vertical.
ExternalScriptAreaCollapsible	Defines whether the user can collapse the external script area. This setting is only relevant when the <code>SideBySideLayout</code> setting is set to vertical.
ExternalScriptAreaWidth	Defines the size of the external script area expressed as a percentage of the tab's total area. This setting is only relevant when the <code>SideBySideLayout</code> setting is set to vertical.

Field	Description
SideBySideEnabled	Defines whether Jacada WorkSpace displays both an external script and Jacada project-defined Web pages, as part of call handling. When enabled, Jacada WorkSpace expects to display, as part of call handling, both an external script and one or more project-defined Web pages in a side-by-side layout. The external script is the controlling script, which instructs Jacada WorkSpace to display the project-defined Web pages. The external script interacts with the currently displayed Web page. A Jacada Interaction Manager script is an example of an external script, that is a script which exists external to the Jacada project.
SideBySideLayout	Defines the layout of the external script in Jacada WorkSpace, either vertical or horizontal. In a vertical layout, the external script and Jacada project-defined Web pages share the <i>Script</i> tab display area. In a horizontal layout, the external script is displayed in a horizontal workspace area, above the <i>Script</i> tab.

Feature Settings

Use the *Feature Settings* page to control whether or not main features of Jacada WorkSpace are displayed.

Field	Description
Supervisor View Enabled	Defines whether the Supervisor View (represented by the <i>Dashboard</i> tab) is displayed.

Field	Description
TickerTape Enabled	Defines whether the ticker tape is displayed. The ticker tape is a progress bar that monitors length of calls.
Timeline Enabled	Defines whether the timeline is displayed. The timeline displays a record of events horizontally, in chronological order.
Toolbar Enabled	Defines whether the toolbar is displayed.

Global Settings

Use the *Global Settings* page to manage the global settings for Jacada WorkSpace projects.

Field	Description
AdministratorEmail	Email address used to send an email to the Jacada WorkSpace administrator with information on any problems encountered regarding the Jacada WorkSpace license.
CTIGatewayEngineTimeout	The time period, in seconds, after which a CTI Gateway engine session is automatically terminated, following no use of the WorkSpace CTI bar.
CTIGatewayLocation	Location of the CTI Gateway.

Field	Description
CTISettingsTableRefreshInSec	Length of interval in seconds between automatic refreshments of the CTI_SETTINGS table. No server restart is required for these refreshments. The default value, zero, means the table is not automatically refreshed. For production, keep the default value. For development, it is possible to configure a period of time after which the CTI_SETTINGS table is automatically refreshed, without a need to restart the server.
DedicatedPushConnectionURL	Provides a URL containing an alternate name for the Jacada WorkSpace application server. This alternate name causes the IE browser, on the agent desktop, to open a new http connection. This new connection is dedicated to transporting only Jacada WorkSpace push messages to the application server. (The <i>alternate_application_server_name</i> is resolved either by each desktop's Hosts file or by a DNS.)
DefaultLocale	Determines the default language for all text displayed in Jacada WorkSpace.
DurableSubscription (Relevant for WebSphere Only)	Causes the deployed Jacada WorkSpace project to maintain its subscription to the Java messaging service in the WebSphere application server host. In the event of a disconnection from the messaging service, for example, due to a server going down, the Java messaging service continues to retain messages intended for the subscribed project. Upon reconnection with the messaging service, Agent receives any messages that were published for it during the disconnection period.

Field	Description
DynamicallyChangeFavoritesTabTitle	Indicates whether or not the <i>Favorites</i> tab title changes to display the opened favorite's name.
Environment	Specifies the environment from which to take any project variables, defined in <i>Project Variables</i> , which are described in Project Variables.
IsSessionDumpEnabled	Specifies whether a session dump log file will be created when a session terminates due to an unexpected error.
LayoutByContext	Specifies whether the layout changes according to the context of agent activity.
MailServerHost	Mail server host ip address used to send an email to the Jacada WorkSpace administrator with information on any problems encountered regarding the Jacada WorkSpace license.
MailServerPort	Mail server port used to send an email to the Jacada WorkSpace administrator with information on any problems encountered regarding the Jacada WorkSpace license.

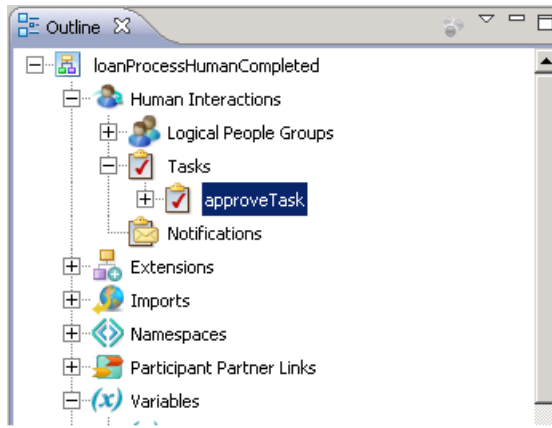
Field	Description
OnDuplicateLogin_AutomaticallyAllowLogin	Defines Jacada WorkSpace handling of an attempt to log in to Jacada WorkSpace when a session currently exists for the provided user name. When set to true, Jacada WorkSpace terminates the existing agent session and allows the agent's log in to a new session. When set to false, Jacada WorkSpace prompts the agent to decide: log in to a new session and terminate the existing session or not.
openMenuOnClick	Defines whether users need to click on a menu to display menu options. By default, the value is set to false, and menu options are displayed when users hover the mouse pointer over a menu.
PushTimeoutInMs	Time in milliseconds that a request from the client waits to receive an event from the server. If no event is received during this time, the server responds with a timeout event, and the client resubmits the request.
sessionCookieName	The name of the cookie used for session tracking. This setting is supported only for WebLogic users, and it needs to be defined only if you do not want to use the default cookie name (<code>jsessionId</code>). When a cookie name other than the default is used, some additional configuration is required. For details, refer to: • Configuring the Session Cookie Name • Jacada WorkSpace Developer Guide (Setting Up the Session Cookie Name).

Field	Description
Theme	Default theme name.
TickerTapePosition	Defines both the placement and the display status of the ticker tape in Jacada WorkSpace.
welcomeTimeoutInSeconds	Time in seconds that the Welcome page is displayed before automatically closing. When the timeout expires, if the agent has started to open messages on the Welcome page, the value is reset. If there are mandatory messages on this page, the value has no effect until after the agent has opened the mandatory messages.

Human Task Settings

Use the *Human Task* page to map a Jacada workflow human task to the location of the task WSDL in the Jacada workflow project.

Field	Description
Project ID	Name of a Jacada WorkSpace project. An asterisk (*) specifies that the entry affects all projects.
Task Name	The unique task name in the Jacada workflow project. The task name needs to be identical to the name assigned in the ActiveVOS Designer (see example below).
Task Location	The location of the task WSDL in the Jacada workflow project. Copy the location from the <i>Resource Detail</i> page of the ActiveVOS Admin Console (<i>Catalog > Contributions > <name of project> > <relevant WSDL></i>). Note: Do not include the <i>project</i> prefix in the location (see example below).



CATALOG ADMIN

Resource Detail

Type:	WS http://schemas.xmlsoap.org/wsdl/
Location:	project/loan/wSDL/loanApproval.wSDL
Version:	1.0.7
Target Namespace:	http://docs.active-endpoints.com/sample/wSDL/loanApproval/2008/02/loanApproval.wSDL
Process Group:	
Description:	
Last Modified	2014/01/03 08:48 PM

Resource Definition

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://docs.active-endpoints.com/sample/wSDL/loanApproval/2008/02/loanApproval.wSDL"
  xmlns:loanmsg="http://docs.active-endpoints.com/sample/wSDL/loanMessages/2008/02/loanMessages.wSDL"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:loan="http://schemas.active-endpoints.com/sample/LoanRequest/2008/02/loanRequest.xsd"
  name="LoanApproval"
  targetNamespace="http://docs.active-endpoints.com/sample/wSDL/loanApproval/2008/02/loanApproval.wSDL">
  <wsdl:import location="loanMessages.wSDL"
    namespace="http://docs.active-endpoints.com/sample/wSDL/loanMessages/2008/02/loanMessages.wSDL"/>
  <wsdl:types>
    <xsd:schema>
      <xsd:import namespace="http://schemas.active-endpoints.com/sample/LoanRequest/2008/02/loanRequest.xsd"
        schemaLocation="../../schemas/loanRequest.xsd"/>
      </xsd:import>
    </xsd:schema>
  </wsdl:types>
```

Update

Instant Messaging Settings

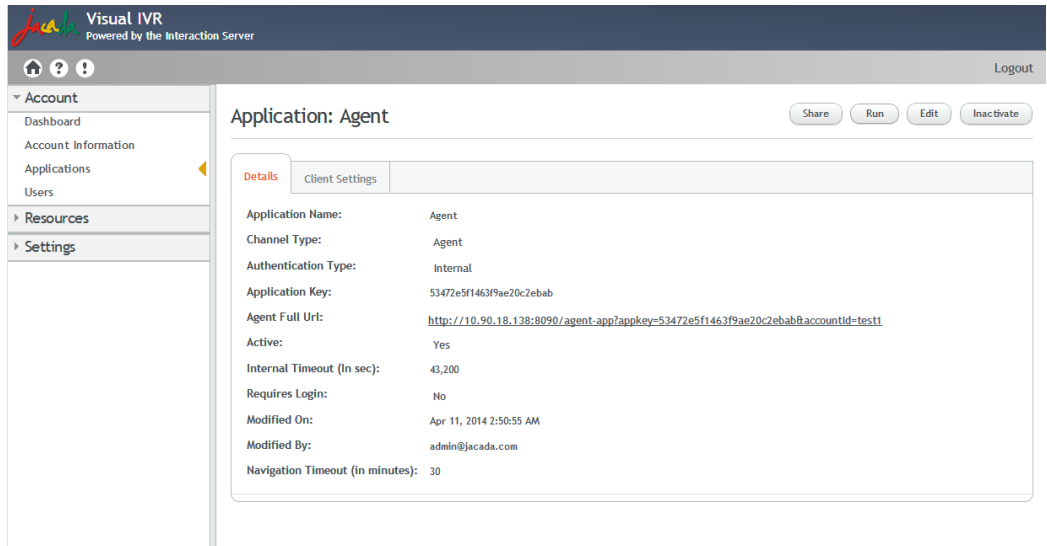
Use the *Instant Messaging Settings* page to define the required XMPP server settings to enable supervisors and agents to chat with each other using their *Chat* portlet in Jacada WorkSpace.

Field	Description
Server Name	The name or the IP address of the machine hosting the XMPP server.
Server Port	The port used by the XMPP server.

Interact Settings

Use the *Interact Settings* page to define the settings required for successful integration with Jacada Interaction Server.

Field	Description
Interact Application Key	The value of the application key used to access the Agent Application via the Jacada Interact Server. This setting can be copied from the <i>Applications</i> menu of the Jacada Interact Server Admin Console (see figure below).
Interact Server URL	The host and port URL of the Jacada Interact Server.
Interact Account	The name of your account on the Jacada Interact Server.



Jacada Interaction Server Integration

For a WorkSpace project to be successfully integrated with Jacada Interaction Server, settings of the Agent Application need to be configured properly in the Interaction Server Admin Console. Settings are accessed from the menu bar of the Admin Console by selecting *Account > Applications*, and then selecting the relevant Agent Application from the list of applications on the *Applications* page.

Note: It is recommended to have a dedicated Agent Application instance for each WorkSpace project.

Settings that are relevant to WorkSpace integration are described in the following table.

Setting	Description/Notes
Authentication Type	This option should be set to <i>Custom</i> .

Setting	Description/Notes
Authentication Link URL	This setting should point to the location of the WorkSpace project. The structure is as follows: <code>http://<host>:<port>/<name of WorkSpace project>/rest/interact</code> Note: The <code>/rest/interact</code> appenders should not be changed.
Requires Login	This option should be set to Yes.

IMPORTANT: If you are working with Jacada Interact on the cloud, and you have a WebLogic or WebSphere application server, you will need to import a security certificate into the keystore of your application server. For details, refer to [Configuring Application Server Settings for Cloud Integration](#).

Configuring Application Server Settings for Cloud Integration

If you are working with a WebLogic or WebSphere application server, it is necessary to import a security certificate for successful cloud integration. To import the certificate into the keystore of your application server, follow the relevant procedure below.

To configure WebLogic server settings for cloud integration:

1. From the WebLogic Admin Console, navigate to *Home > Summary of Servers > AdminServer*, and select the *Keystores* tab.
2. In the *Java Standard Trust Keystore Passphrase* field, enter *changeit*. Then, enter *changeit* in the *Confirm Java Standard Trust Keystore Passphrase* field.
3. Save the settings, and then select the *SSL* tab.
4. At the bottom of the *Advanced* frame, select the *Use JSSE SSL* checkbox.
5. Save your settings, and then restart the server.

To configure WebSphere server settings for cloud integration:

1. From the WebSphere Admin Console, navigate to *SSL certificate and key management > Key*

stores and certificates > NodeDefaultTrustStore > Signer certificates.

2. At the top of the grid, click *Retrieve from port*.
The *General Properties* page opens.
3. Configure the following settings:
 - In the *Host* field, enter *gointeract.io*
 - In the *Port* field, enter *443*
 - In the *Alias* field, enter *GoDaddyC2CA*
4. Click *Retrieve signer information*. In the *Issued By* field, the following line should be displayed:
OU=Go Daddy Class 2 Certification Authority, O="The Go Daddy Group, Inc.", C=US
5. Click *Apply* and save your settings.

Locales

The *Locales* page provides the ability to manage locale information per Jacada WorkSpace project.

Field	Description
Project ID	Name of a Jacada WorkSpace project to which the entry applies. An asterisk (*) specifies that the entry applies to all projects.
Locale	Any Java Locale code, for example <code>en</code> for English, <code>zh</code> for Chinese (traditional) or <code>pt_BR</code> for Brazilian Portuguese.

Click the *Add* button to define a new locale entry.

Locale entries cannot be edited. Existing entries must be deleted and then defined anew.

Machine Settings

Use the *Machine Settings* page to associate each CTI Gateway with one or more CTI profiles, thus enabling the use of different CTI Gateways with different CTI providers. The page contains a list of one or more host/port combinations that serve as the key to the table. When a CTI Gateway starts, it searches for its own host address and port details in the page and locates the most specific match.

The value of the host and the port can be either a specific value, or an asterisk (*) that represents any value. The following search order is used when searching for a host and port match in the *Machine Settings* page:

1. [specific host, specific port]
2. [specific host, port=*]
3. [host=*, specific port]
4. [host=*, port=*]

You configure the page according to the CTI providers in use.

Field	Description
Host	The IP of the server on which the CTI Gateway resides. An asterisk (*) specifies any IP.
Port	The port number of the server on which the CTI Gateway resides. An asterisk (*) specifies any port.
Name	Reserved for future use. The value must always be PROFILES.
Value	A list of profiles supported by the CTI Gateway. The profile names are separated by commas. All gateways support the MANUAL profile for agents, enabling work in manual mode.
Comments	Free text to provide information about the setting.
Active	A boolean value specifying whether the entry is active or not. Specifying that an entry is not active, disables the entry. This is useful for testing purposes.
Action	Actions, such as <i>Edit</i> , available for use with each machine setting entry.

Note: In a simple configuration, where all CTI Gateways connect to the same profile, it is only necessary to add to the default row (*,*,...) the name of the profile to use.

Page Mappings

Use the *Page Mappings* page to add, delete and edit page mappings. Jacada WorkSpace uses these mappings to locate a project-defined Web page for display in the *Script* tab. During call handling and when side-by-side script interaction is enabled, an external script instructs Jacada WorkSpace as to the specific Web pages to display.

Field	Description
Page ID	The Web page name. For example, <code>gameWinnerPage</code> or <code>wrongSelectionPage</code> .
Context Path	The relative project path to where the Web page is stored. For example, <code>USER\resources\jsp\wrongSelectionPage.jsp</code> .
Action	Actions, such as <i>Edit</i> , available for use with each page mapping entry.

RSS Feed Definitions

Use the *RSS Feed Definitions* page to manage RSS feeds for Jacada WorkSpace projects.

Field	Description
Project ID	ID of the project.
Name	Name assigned to the RSS feed.

Field	Description
Resource location	URL of the RSS server.
Refresh rate	The interval in minutes between refreshes of the feed. All messages, including from an RSS feed are created in chunks and displayed in the ticker tape. When a feed is refreshed, it is not displayed in the ticker tape until the current content of the ticker tape finishes, which can be a little more than the actual refresh time specified.
Time to live	The time, in minutes, that the Ticker Tape continues to display an RSS feed item, after the RSS feed has stopped publishing that item.
Max number of items	The maximum number of RSS feed items to be displayed in the Ticker Tape from the specified RSS server resource.
Action	Actions, such as <i>Edit</i> , available for use with each defined RSS feed in the table.

RTN Mappings

Use the *RTN Mappings* page to map routing telephone numbers (RTNs) to the script that will be used during agent call handling. The scripts available for Jacada WorkSpace call handling are managed in the *Script Mappings* page.

Field	Description
Project ID	Name of a Jacada WorkSpace project. An asterisk (*) specifies that the entry affects all projects. This field cannot be modified in an existing RTN Mapping.
RTN	The routing telephone number. This field cannot be modified in an existing RTN Mapping.

Field	Description
Script ID	A call handling script name defined in the <i>Script Mapping</i> page, described in Script Mappings.
Action	Actions, such as <i>Edit</i> , available for use with each defined RTN in the table.

Script Mappings

Use the *Script Mappings* page to manage the call handling scripts that are available to use with Jacada WorkSpace. For agent call handling, one or more routing telephone numbers (RTN) are associated with an available script in the *RTN Mappings* page.

Field	Description
Script ID	Call handling script name. This value is used to match the script with the RTN described in RTN Mappings.
Context Path	The path of the call-handling script relative to the root of the Jacada WorkSpace project or the URL of the Jacada Interaction Manager call-handling script. When the prefix <code><currentInstance></code> or <code><CurrentInstance></code> appears at the front of the CONTEXT PATH entry, it indicates that the entry is an external path to a Jacada Interaction Manager call-handling script that requires Jacada WorkSpace to resolve the host and port info for this <code><currentInstance></code>.
Action	Actions, such as <i>Edit</i> , available for use with each defined script in the table.

Smart Pad Settings

The *Smart Pad Settings* page provides the ability to manage the settings that manage both the use and the display of the *Smart Pad* portlet in Jacada WorkSpace:

Field	Description
Empty Entry Position	Valid values are <i>Top</i>, <i>Bottom</i> and <i>None</i>. The setting's default value is <i>Bottom</i>. Defines whether or not the Smart Pad always provides a blank entry and, if so, the position in which the blank entry always appears.
Hide Columns	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>true</i>. Defines whether or not users can hide Smart Pad columns from display.
Maximum Entries	Valid value: A number between 0 – 250. Limits the number of entries that can be added by the agent, as follows: If the number of existing entries is $\Rightarrow$ value, the agent cannot add entries. If the number of existing entries is $<$ value, the agent can add up to value.
Sort Columns	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>true</i>. Defines whether or not users can sort the Smart Pad entries by a specified column.

Supervisor View Settings

The *Supervisor View Settings* page allows you to enable/disable specific features of the *Dashboard* tab (Supervisor View) in Jacada WorkSpace. The *Dashboard* tab itself is enabled/disabled using the *Feature Settings* page (Feature Settings).

The Supervisor View allows supervisors to see only the information that is relevant to agents under their supervision.

Field	Description
Agent Task View Enabled	Defines whether the <i>Agent Task</i> portlet is displayed (allowing supervisors to view the tasks of their agents).
Allow chat initiation	Defines whether supervisors are able to initiate chat sessions with agents under their supervision.
CTI Status Enabled	Defines whether the <i>Status</i> column appears in the <i>Agents List</i> portlet. The <i>Status</i> column lists the current status of agents (Offline, etc.)

Agent Disposition

The Jacada Management Console enables managing the *Agent Disposition* portlet in Jacada WorkSpace. *Agent Disposition* provides the following options:

- [Disposition Codes](#)
- [Settings](#)

Disposition Codes

The *Disposition Codes* page provides the ability to manage the set of disposition codes displayed in the *Agent Disposition* portlet. The page enables you to add, edit and remove disposition codes.

Field	Description
Disposition Code	A unique code identifying specific disposition information.
Display Value	The value displayed to the agent for the current disposition code. When no value is provided, then, by default, this field is set to the disposition code value.
Parent Code	Identifies the parent disposition code of this current disposition code. When the Code Hierarchy setting is true, the parent code impacts the content displayed in the disposition value fields of the <i>Agent Disposition</i> portlet. By defining this field for a disposition code, you establish a hierarchical relationship between it and the identified parent code. To establish the current disposition code as a top-tier parent code, when the codes are related hierarchically, select the value <i>None</i> in the field's dropdown list.
Action	Actions, such as <i>Edit</i> , available for use with each defined disposition code in the table.

Click the *Add* button to define a new disposition code.

The Jacada Management Console does not delete selected disposition codes while they are defined as a parent code in other disposition codes. To delete a disposition code, ensure that it is not defined as a parent code in any other disposition codes.

Settings

The *Settings* page provides the ability to manage the settings that manage the use and display of disposition codes:

Field	Description
Enable Automatic Call Disposition	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>true</i>. Indicates whether or not the <i>Automatic Call Disposition</i> popup is available.
Code Hierarchy	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>false</i>. Indicates whether the disposition code fields share a hierarchical relationship or are discrete, unrelated fields. When fields are related hierarchically, they display in a hierarchal list. When fields are unrelated, they display in a left-justified, top-to-bottom list.
Number of Disposition Value Fields	Valid integer values starting at 1. The setting's default value is 3. Identifies the number of fields in the Disposition Values area that the Agent Disposition portlet must provides.
Disposition Codes Enabled	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>false</i>. Indicates whether or not disposition codes are available for agent use in the Agent Disposition portlet. When not available, only the <i>Notes</i> pane is available for use by the agent in the portlet.

Field	Description
Only Display Parent Codes	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>true</i>. Indicates to only display, in the Agent Disposition portlet, those disposition codes that are defined as parent codes. Relevant when your disposition codes are hierarchically related (Code Hierarchy = <i>true</i>).

Auditing

The Jacada Management Console enables auditing events in Jacada WorkSpace. *Auditing* provides the following options:

- [Events](#)
- [Settings](#)

Events

Use the *Auditing Events* page to manage auditing for Jacada WorkSpace events. Select from the following set of events:

Event Name	Auditing Status	Description
Add Task	Enabled (default)/ Disabled	The event generated when a new task is added.
Add Task to BPM	Enabled (default)/ Disabled	The event generated when Jacada WorkSpace adds a task to an external BPM system.

Event Name	Auditing Status	Description
After Call Work	Enabled (default)/ Disabled	The code generated event that moves the agent, currently in Ready status, directly to After Call Work status.
Answer	Enabled (default)/ Disabled	The event generated by the agent answering an incoming, business call and auto-answer is not enabled.
Answer Custom Item	Enabled (default)/ Disabled	The event generated by the agent answering an incoming, custom item call and custom item auto-answer is not enabled.
Busy	Enabled (default)/ Disabled	The event generated when the agent, in CTI inbound mode, requests to move to the Busy status.
Chat	Enabled (default)/ Disabled	The event generated by the agent or supervisor beginning a chat session. This event contains information that reports about both the beginning and the ending of the chat session.
Chat Message	Enabled (default)/ Disabled	The event generated by the agent or supervisor sending a chat message.
Cold Transfer	Enabled (default)/ Disabled	The event generated by the agent's cold transfer of a call.
Complete Transfer	Enabled (default)/ Disabled	The event generated by the agent completing the warm transfer of a call.
Consult	Enabled (default)/ Disabled	The event generated by the agent moving a call into a two-party consult call, which occurs between either two agents or an agent and a supervisor.

Event Name	Auditing Status	Description
Delete Task	Enabled (default)/ Disabled	The event generated when a task is deleted.
Dial	Enabled (default)/ Disabled	The event generated by the agent dialing a call.
End Call	Enabled (default)/ Disabled	The event generated by the agent ending a call.
End Disposition	Enabled (default)/ Disabled	The event generated when a call disposition script ends.
End Item	Enabled (default)/ Disabled	The event generated by the agent ending a custom item call.
Handshake Transfer	Enabled (default)/ Disabled	The event generated by the agent's handshake transfer of a call.
Handshake Transfer Connect (Genesys only)	Enabled (default)/ Disabled	The event generated by the agent's handshake transfer connect of a call.
Hold	Enabled (default)/ Disabled	The event generated by the agent placing a call on hold.
Off Hold	Enabled (default)/ Disabled	The event generated by the agent removing a call from hold.
Join	Enabled (default)/ Disabled	The event generated by the agent joining the caller to the consult call, thereby establishing a three-party conference call.

Event Name	Auditing Status	Description
Login	Enabled (default)/ Disabled	The event generated by an agent or a supervisor logging into Jacada WorkSpace.
Logout	Enabled (default)/ Disabled	The event generated by an agent or a supervisor logging out from Jacada WorkSpace.
Ready	Enabled (default)/ Disabled	The event generated when the agent, in CTI inbound mode, requests to move to the Ready status.
Resynch (Genesys only)	Enabled (default)/ Disabled	The event generated by the agent re-establishing the connection between a Jacada WorkSpace session and CTI.
Return	Enabled (default)/ Disabled	The event generated by the agent returning to the caller, during either a warm transfer of a call or a two-party consult call.
Start	Enabled (default)/ Disabled	The event generated by the agent, in manual mode, starting a call.
Start Disposition	Enabled (default)/ Disabled	The event generated by the starting of the call disposition script.
Start Outbound	Enabled (default)/ Disabled	The event generated by the agent starting an outbound call.
Switch Tab	Enabled (default)/ Disabled	The event generated by the agent switching, while in a business call, between two Jacada WorkSpace tabs.
Transfer From Consult	Enabled (default)/ Disabled	The event generated by the agent transferring the caller, during a two-party consult call, directly to a supervisor.

Event Name	Auditing Status	Description
Update Task	Enabled (default)/ Disabled	The event generated when a task is updated.
Warm Transfer	Enabled (default)/ Disabled	The event generated by the agent's warm transfer of a call.
Warm Transfer Connect (Genesys only)	Enabled (default)/ Disabled	The event generated by the agent's warm transfer connect of a call.

Settings

Using the *Auditing Settings* page, manage the auditing feature for your project.

Setting Name	Value	Description
AuditingEnabled	true/ false (default)	Controls the auditing feature in Jacada WorkSpace. When enabled, database auditing is available for use and project level auditing can be developed.
DBAuditingEnabled	true (default) / false	Controls database auditing. When enabled, the following information is saved in the database, during Jacada WorkSpace operation: - Jacada WorkSpace-provided events that are enabled for audit. - Any project level events that are explicitly coded for database auditing.

Authentication and Authorization

The Jacada Management Console enables managing the authentication and authorization that is available with Jacada WorkSpace. *Authentication-Authorization* provides the following options:

- [Groups](#)
- [Privileges](#)
- [Roles](#)
- [Settings](#)
- [Skills](#)
- [Users](#)
- [LDAP](#)

Groups

A group can be assigned any of the following:

- Multiple peer groups.
- Multiple roles.
- Multiple supervisors.
- Multiple expert groups.
- Multiple users – user group assignments are performed in the *Authentication-Authorization > Users* option.

The *Groups* page enables managing Jacada WorkSpace groups. The Jacada WorkSpace-provided groups are: *Supervisors*, *CTIUsers* and *Agents*.

Field	Description
Name	Name of a Jacada WorkSpace group.
Action	Actions, such as <i>Edit</i> , available for use with each defined group in the table.

Click the *Add* button to define a new group.

Clicking the *Set Group Search Order* button displays a dialog page to modify the group search order that Jacada WorkSpace uses whenever it performs a search for a user's supervisors. The list of groups displays in descending search order.

When external authentication is the security mode in use by Jacada WorkSpace for user login authentication (*Authentication-Authorization > Settings* page), selecting a group's checkbox in the table enables the *Synchronize* button. Clicking *Synchronize* causes the Jacada Management Console to retrieve, from the established LDAP server, any new users that are assigned to groups designated as either supervisor or expert and update both the Jacada database and the established XMPP server with the new users.

Privileges

A privilege is assigned to one or more than one role.

Jacada WorkSpace-provided Privileges

The following Jacada WorkSpace-provided privileges are available:

Privilege	Description
AgentChat	Ability to chat with other agents.
AgentDispositionPortlet	Use of the <i>Agent Disposition</i> portlet.
BusyReasonCode	Ability to set a busy reason for current call
ChatHistory	Availability of the <i>Chat History</i> tab.
CTIBar	Availability of the CTI bar.
CTIBusyReady	Ability to press the <i>Busy/Ready</i> button in the CTI bar.
CTIConsult	Ability to press the <i>Consult</i> button in the CTI bar.
CTIDial	Ability to press the <i>Dial</i> button in the CTI bar.

Privilege	Description
CTIEnd	Ability to press the <i>End Call</i> button in the CTI bar.
CTIHold	Ability to press the <i>Hold</i> button in the CTI bar.
CTIStart	Ability to press the <i>Start</i> button in the CTI bar.
CTIStartOutboundCall	Ability to press the <i>Start Call</i> button in the CTI bar.
CTITransfer	Ability to press the <i>Transfer</i> button in the CTI bar.
CTITransferConnect	Ability to press the <i>Transfer Connect</i> button in the CTI bar.
DeleteTasks	Allows agent to delete tasks.
DocumentsSearch	Use of the <i>Document Search</i> portlet.
DynamicViews	Availability of Dynamic Views.
EditFellowTeamMemberTasks	Allows agents to edit tasks of other team members.
ExpertChat	Ability to chat with experts.
HelpOnHand	Use of the <i>Help On Hand</i> portlet.
JMCAgentDisposition	Ability to view the <i>Agent Disposition</i> node in the JMC (Jacada Management Console).
JMCAuditing	Ability to view the <i>Auditing</i> node in the JMC (Jacada Management Console).

Privilege	Description
JMCAuthentication Authorization	Ability to view the <i>Authentication-Authorization</i> node in the JMC (Jacada Management Console).
JMCCTI	Ability to view the <i>CTI</i> node in the JMC (Jacada Management Console).
JMCDocumentSearch	Ability to view the <i>Document Search</i> node in the JMC (Jacada Management Console).
JMCDynamicViews	Availability of the <i>Dynamic Views Settings</i> node in the JMC (Jacada Management Console).
JMCExternalScriptSettings	Availability of the <i>External Script Settings</i> node in the JMC (Jacada Management Console).
JMCFeatureSettings	Availability of the <i>Feature Settings</i> node in the JMC (Jacada Management Console).
JMCGlobalSettings	Availability of the <i>Global Settings</i> node in the JMC (Jacada Management Console).
JMCInstantMessagingSettings	Availability of the <i>Instant Messaging Settings</i> node in the JMC (Jacada Management Console).
JMCInteract	Availability of Interact integration in the JMC (Jacada Management Console).
JMCLayoutManagement	Ability to view the <i>Layout Management</i> node in the JMC (Jacada Management Console).
JMCLocales	Availability of the <i>Locales</i> node in the JMC (Jacada Management Console).
JMCLogging	Ability to view the <i>Logging</i> node in the JMC (Jacada Management Console).

Privilege	Description
JMCMachineSettings	Availability of the <i>Machine Settings</i> node in the JMC (Jacada Management Console).
JMCMonitoring	Ability to view the <i>Monitoring</i> node in the JMC (Jacada Management Console).
JMCPageMappings	Availability of the <i>Page Mappings</i> node in the JMC (Jacada Management Console).
JMCProjectVariables	Ability to view the <i>Project Variables</i> node in the JMC (Jacada Management Console).
JMCRSSFeedDefinitions	Availability of the <i>RSS Feed Definitions</i> node in the JMC (Jacada Management Console).
JMCRTNMappings	Availability of the <i>RTN Mappings</i> node in the JMC (Jacada Management Console).
JMCScriptMappings	Availability of the <i>Script Mappings</i> node in the JMC (Jacada Management Console).
JMCSmartPad	Availability of the <i>Smart Pad Settings</i> node in the JMC (Jacada Management Console).
JMCSupervisorView	Availability of the <i>Supervisor View</i> node in the JMC (Jacada Management Console).
JMCTabsConfiguration	Availability of the <i>Tabs</i> node in the JMC (Jacada Management Console).
JMCTaskManager	Ability to view the <i>Task Manager</i> node in the JMC (Jacada Management Console).
MenuBarItemFavorites	Availability of the <i>Favorites</i> menu and the ability to use the <i>Manage Favorites</i> menu item.

Privilege	Description
MenuBarItemHelp	Availability of the <i>Help</i> menu.
MultiChat	Ability to hold simultaneous chat sessions.
PeerChat	Ability to chat with peers.
PortletChat	Use of the <i>Chat</i> portlet.
PortletDialPad	Use of the <i>Dial Pad</i> portlet.
PortletFavorites	Use of the <i>Favorites</i> portlet.
PortletGuidancePortlet	Use of the <i>Guidance</i> portlet.
PortletSamplePortlet	Use of the <i>Sample</i> portlet.
PortletSmartPad	Use of the <i>Smart Pad</i> portlet.
ReportingSuperUser	Authorization to run reports for all available groups.
ReportingTab	Availability of the <i>Reports</i> tab.
SampleTab	Availability of the <i>Sample</i> tab.
SearchResultTab	Availability of the <i>Search Results</i> tab.
SuperChat	Ability to chat with supervisors.
SupervisorDashboardTab	Availability of the <i>Dashboard</i> tab.
TabFavoritesTab	Availability of the <i>Favorites</i> tab.

Privilege	Description
TabMessageBoard	Availability of the <i>Message Board</i> tab.
TabMessageCenter	Availability of the <i>Message Center</i> tab.
TabTickerTapeTab	Availability of the <i>Ticker Tape</i> tab.
TaskManager	Use of the <i>Agent Tasks</i> portlet.
TasksTeamView	Availability of the Task Manager Team View filter.
TickerTapeFeature	Availability of the Ticker Tape feature.
TimelineFeature	Availability of the Timeline feature.
ToolbarFeature	Availability of the Toolbar feature.

Privileges Page Functions

The *Privileges* page provides the ability to manage Jacada WorkSpace privileges.

Field	Description
Name	Name of a Jacada WorkSpace privilege.
Action	Actions, such as <i>Edit</i> , available for use with each defined privilege in the table.

Click the *Add* button to define a new privilege.

Roles

A role is assigned to one or more than one group. The Jacada WorkSpace-provided roles are: *SupervisorRole*, *CTIUserRole* and *AgentRole*.

Defining a Role

Field	Description
Name	Name of a Jacada WorkSpace role.
Action	Actions, such as <i>Edit</i> , available for use with each defined role in the table.

Click the *Add* button to define a new role.

Edit Assigned Privileges Link

Clicking the *Edit Assigned Privileges* link, available when defining a new role and when editing an existing role, displays a dialog page to manage the permission to use the privileges that are assigned to the role. The page displays a table listing the role's assigned privileges.

Field	Description
Privilege	The name of the assigned privilege.
Enabled	A checkbox that, by default, is selected. Indicates whether or not the assigned privilege is permitted to be used by the role. When enabled is selected, the role has full use of any component that implements this privilege. When enabled is not selected, the role is restricted to only viewing any component that implements this privilege.

Settings

The *Settings* page provides the ability to manage system level security settings for Jacada WorkSpace:

Field	Description
Security Mode	Valid values are <i>standalone</i> and <i>external</i>. The setting's default value is <i>standalone</i>. Indicates the authentication mode that Jacada WorkSpace uses for user login authentication, as follows: • Standalone – Jacada WorkSpace authenticates user log in attempts via the user information configured in its database. • External – Jacada WorkSpace authenticates user log in attempts via accessing an external LDAP server that hosts the organization's Active Directory of users. Note: Modifying this setting's value must be followed by the restart of all the application servers hosting Jacada WorkSpace.
Server Administrator Name	The administrator username with which Jacada WorkSpace logs into a specified server.
Server Administrator Password	The administrator password with which Jacada WorkSpace logs into a specified server. The password is recorded in its encrypted form.

Skills

Skills identify the acquired knowledge, capabilities and experience available within your Jacada WorkSpace users. Skills are then assigned to one or more groups. This enables distinguishing groups, and the users assigned to them, by their collective expertise.

The *Skills* page provides the ability to manage Jacada WorkSpace skills. Out of the box, Jacada

WorkSpace does not provide any skills. Click the *Add* button to define a new skill.

When a minimum of a single skill is defined, the page displays a table listing the defined skills.

Field	Description
Name	Name of a Jacada WorkSpace skill.
Description	Description of the skill.
Action	Actions, such as <i>Edit</i> , available for use with each defined skill in the table.

Click the *Add* button to define a new skill.

Users

Jacada WorkSpace authenticates user login in standalone authentication mode by accessing user information defined in the *Users* page. The Jacada WorkSpace-provided users are: *workspace* (supervisor) and *guest* (basic).

Field	Description
Name	User name of a Jacada WorkSpace user.
Action	Actions, such as <i>Edit</i> , available for use with each defined user in the table.

Click the *Add* button to define a new user, including the user's Jacada WorkSpace group assignments.

LDAP

Jacada WorkSpace supports user login authentication in external authentication mode by accessing an external LDAP server that hosts the organization's Active Directory of users. *LDAP* provides the following options:

- Groups
- Settings

Groups

The *Groups* page provides the ability to import groups from the established, external LDAP server into Jacada WorkSpace. The page displays a table listing only those LDAP groups that have not yet been imported into Jacada WorkSpace.

Clicking the *Import* button, imports all of the LDAP groups whose checkbox is selected in the table. Once imported, groups become valid Jacada WorkSpace groups.

Settings

The *Settings* page provides the ability to manage all the necessary information about the external LDAP server to enable Jacada WorkSpace to access this server and its relevant data in order to provide user login authentication in external authentication mode.

Field	Description
Base Dn	The branch in the LDAP directory structure from which to begin a data search. No spaces are allowed between the provided Base Dn values.
Group common name	An LDAP attribute that is always used to find group related data in the LDAP server.
Group object class name	The object class value that is used, during an LDAP search, to identify the LDAP entry as a group.
Group search base	Relative to the provided Base Dn, the location in the LDAP directory structure from which to begin a search for groups.

Field	Description
Group Membership Search Level	Specifies how many levels of group membership can be searched.
MemberOf attribute for groups	The value, employed in the LDAP search, that retrieves all the users that are associated with a specific group.
MemberOf attribute for users	The value, employed in the LDAP search, that retrieves all the groups that are associated with a specific user.
LDAP Password	The associated user password, defined in the LDAP server, for Jacada WorkSpace log in to the server.
User search filter	LDAP user attribute that is employed in the search to find a specific user.
LDAP URL	URL path to the LDAP server used by Jacada WorkSpace. Provide a URL in the following format: ldap://IP address:Port
User common name	An LDAP attribute that is always used to find user related data in the LDAP server.
User display name	An LDAP attribute that is always used for a user display name in the LDAP server.
LDAP Username	The user, defined in the LDAP server, for Jacada WorkSpace log in to the server.
User object class name	The object class value that is used, during an LDAP search, to identify the LDAP entry as a user.

Field	Description
User search base	Relative to the provided Base Dn, the location in the LDAP directory structure from which to begin a search for users.

Note: Modifying the value of any LDAP setting must be followed by the restart of all the application servers hosting Jacada WorkSpace.

For testing purposes, clicking the *Test Connection* button executes a communication connection with the identified LDAP server, based on the values provided in the page's *LDAP URL*, *LDAPusername* and *LDAPpassword* fields.

CTI

The Jacada Management Console enables managing and reviewing the information necessary for computer telephone integration (CTI). *CTI* provides the following options:

- [Agents](#)
- [Busy Reason Codes \(Genesys and Cisco\)](#)
- [Dial List](#)
- [Settings](#)

Agents

The *Agents* page provides the ability to manage the agents using CTI.

Field	Description
User name	User name of an agent defined in the authentication provider, such as Microsoft Active Directory.
CTI profile	Name of the CTI profile used by the agent.

Field	Description
Extension Number	Telephone extension number of the CTI Agent.
Group ID	ID of the group to which the agent belongs.
CTI login name	When required by the CTI server or the switch, the name used to log into the extension.
CTI password	When required by the CTI server or the switch, the password used to log into the extension.
Outbound extension number	Telephone extension to use for outbound calls. The value must be numeric.
Agent ID	Agent's logical ID (CTI PHONEID)
Action	Actions, such as <i>Edit</i> , available for use with each defined agent in the table. Clicking on the action, an active link, displays a dialog page where you edit the fields of the table entry.

Click the *Add* button to define a new CTI agent entry.

Busy Reason Codes (Genesys and Cisco)

The *Busy Reason Codes* page provides the ability to define the codes used by agents to indicate a reason when updating their call handling status to “busy”.

Field	Description
Busy Reason Code	An alphanumeric code representing a specific busy reason for an agent. The valid code length is specific to your switch configuration. The defined busy reason code and name display as an entry in the dropdown menu of available busy reasons that is associated with the <i>Busy</i> button in the CTI bar.

Field	Description
Busy Reason Description	Description of the busy reason, for example, <i>20 minute recess for the afternoon shift agent</i> . Maximum length is 255 characters.
Busy Reason Display Name	Display name assigned to the busy reason code, for example <i>Shift recess-afternoon</i>. Maximum length is 255 characters. The defined busy reason code and name display as an entry in the dropdown menu of available busy reasons that is associated with the <i>Busy</i> button in the CTI bar.
Action	Actions, such as <i>Edit</i> , available for use with each defined code in the table.

Click the *Add* button to define a new busy reason code entry.

Dial List

The *Dial List* page provides the ability to manage the telephone numbers used by agents to dial their calls.

Field	Description
Dial List Number	The destination number without the DTMF code. The valid number length is specific to your switch configuration. The defined number, name, and number description display as an entry in the dropdown menu of available outbound numbers.
Transfer Type (Genesys only)	Defines whether the transfer of the number occurs by standard transfer or by <i>transfer connect</i> (sending of Dual Tone Multi-Frequency (DTMF) tones to the configured switch). Select either <i>Standard</i> or <i>Connect</i> from the dropdown menu.

Field	Description
Privilege	Defines the Jacada WorkSpace privilege required for the agent in order for the defined transfer number to display as an entry in the dropdown menu of available outbound numbers. Select a privilege from the dropdown menu.
Name	The name assigned to the defined number. Maximum length is 64 characters. The defined number, name, and number description display as an entry in the dropdown menu of available outbound numbers.
Description	The description assigned to the defined number. Maximum length is 255 characters. The defined number, name, and number description display as an entry in the dropdown menu of available outbound numbers.
Use access code	Defines whether to prefix the access code, which is defined in the CTI setting <code>CTIOutboundAccessKey</code>, to the defined Dial List number. Select either <i>True</i> or <i>False</i> from the dropdown menu.
Filter 1	Filter 1 value to match the Dial List number.
Filter 2	Filter 2 value to match the Dial List number.
Action	Actions, such as <i>Edit</i> , available for use with each defined transfer number in the table.

Click the *Add* button to define a new transfer number entry.

Settings

Use the CTI *Settings* pages to review the CTI settings, per supported CTI provider and switch. Changing these settings is done directly in the database.

Call-handling Profiles

Jacada WorkSpace provides the following call-handling profiles:

Profile Name	Description
DEFAULT	This profile is assigned, during log in to a Jacada WorkSpace session, to all users that do not specify their call-handling profile in the URL of their browser. By definition, the DEFAULT profile has no profile settings of its own. DEFAULT inherits the settings of the parent profile to which it is assigned, for example, MANUAL, CTI or AES.
CTI	The CTI profile is the top level profile for call-handling that is automated for an agent by a CTI. In the Jacada Management Console navigation menu, access its profile page via the <i>Configuration > CTI > Settings > Default > CTI</i> option. The CTI profile provides the following hierarchy of children profiles: • CTC profile • AVAYA profile – for use when CTC interfaces with an Avaya switch • NORTEL profile – for use when CTC interfaces with a Nortel switch. • OUTBOUND profile – for use when CTC interfaces with a Nortel switch that supports outbound calling.

Profile Name	Description
CTI (continued)	The CTI profile provides the following children profiles (continued): CISCO profile – for use when Cisco interfaces with an Avaya switch • GENESYS profile • AVAYA profile – for use when Genesys interfaces with an Avaya switch • ALCATEL profile – for use when Genesys interfaces with an Alacatel switch • AES profile • AVAYA profile – for use when AES interfaces with an Avaya switch
MANUAL	The MANUAL profile is the top level profile for Jacada WorkSpace call-handling that is manually performed by an agent.

Each call-handling profile supplies a collection call-handling settings. A child profile inherits its parent profile settings. For example, the AES profile inherits the settings of its CTI parent. The AVAYA profile inherits the settings of its AES parent.all profiles under the *CTI* profile inherit the settings of the CTI profile.

Jacada Management Console Profile Page

Each call-handling profile has a corresponding Jacada Management Console profile page. This page displays:

- A *Profile Settings* table listing the settings that this profile provides. The entries can be edited. The table contains the following columns:

Field	Description
Key	Name of the call-handling setting. Clicking on the setting name, an active link, displays a dialog page to edit setting information. Call-handling setting information to edit: value, comments and active (setting activation indicator). After editing, clicking <i>Apply</i> updates the Jacada WorkSpace database.
Value	The value assigned to the defined the call-handling setting. Maximum length is 255 characters.
Comments	Comments related to the defined the call-handling setting. Maximum length is 255 characters.
Active	Valid values are <i>checked</i> and <i>unchecked</i>. The setting's default value is <i>checked</i>. Indicates whether the defined call-handling setting, provided by the profile, is activated for use (<i>checked</i>) or not (<i>unchecked</i>).

- An *Inherited Profile Settings* table listing the settings that this profile inherits from its parent profile. These setting entries are view-only.

Jacada WorkSpace-provided Profile Settings

The following table lists all the Jacada WorkSpace-provided profile settings that are available for use by all projects (listed alphabetically):

Setting	Values	Description
CTIAgentStatusDuringCallSetBySwitch	<code>true</code> <code>false</code>	Specifies whether the agent status during a call is set by the switch, and thus should not be set by the Jacada WorkSpace application.
CTIAllowDialPadWindow	<code>true</code> <code>false</code>	Enables agents to use the Dial Pad window, which is accessed via the Dial button in the CTI bar of Jacada WorkSpace. If this setting is disabled, the Dial Pad window is not available via the Dial button and no <i>Dial number...</i> link appears in the <i>Dial List</i>., thereby restricting agents from dialing numbers when using the call-handling buttons (though the Dial Pad portlet could still be available). For details, refer to the Jacada WorkSpace Developer Guide.
CTIAllowTransferFromConsult	<code>true</code> <code>false</code>	Specify whether to support transferring a call from a consultation. The transfer is the default transfer type. Note that this will only work when the switch supports transferring a call from a consultation. For details, refer to the Jacada WorkSpace Developer Guide.

Setting	Values	Description
CTIApplicationSetACW	<code>true</code> <code>false</code>	Specify whether Jacada WorkSpace should handle changing the agent status to After Call Work (ACW). When <code>true</code>, Jacada WorkSpace handles changing the agent status. When <code>false</code>, the status is changed by the CTI or the switch.
CTIAttachedDataKey1, CTIAttachedDataKey2, ... (Genesys only)		The names of the attached data keys to use for filtering voice items. For each attached data key, you need an entry in the database. For details, refer to the Jacada WorkSpace Developer Guide.
CTIAttachedDataValue1, CTIAttachedDataValue2, ... (Genesys only)		The values of the attached data keys to use for filtering voice items. These values must correspond to the names specified in the <code>CTIAttachedDataKey1, 2, ...</code> settings. For details, refer to the Jacada WorkSpace Developer Guide.
CTIBackupPort (Genesys and Cisco only)		The port to use to connect to the backup CTI server.
CTIBackupServer (Genesys and Cisco only)		The name or IP address of a backup CTI server, in case the <code>CTIServer</code> fails.

Setting	Values	Description
CTICallAutoAnswer	<code>true</code> <code>false</code>	Specify whether Jacada WorkSpace should automatically answer calls. For details, refer to the Jacada WorkSpace Developer Guide.
CTICallMode	<code>inbound</code> <code>outbound</code>	The mode of the Jacada WorkSpace application: receiving inbound calls or dialing outbound calls. For details, refer to the Jacada WorkSpace Developer Guide.
CTIChangeToManualOnCTI OperationFailure	<code>true</code> <code>false</code>	Specify whether Jacada WorkSpace should change to manual mode upon the failure of a CTI operation. When set to <code>false</code>, Jacada WorkSpace attempts to continue working despite the failure. Note: In production, it is recommended to set the value of this parameter to <code>false</code>.
CTIConfigManager ApplicationName (Genesys only)		The name of the Genesys application of type 'Config Manager', as defined in the Genesys Configuration Manager Environment (CME).

Setting	Values	Description
CTIConfigurationMode	- Agent - Global	Specify whether the configuration mode is global or agent-specific. For each CTI provider you specify one of the modes, as follows: AES: Agent CTConnect: Agent Cisco: Agent Genesys: Global For Agent mode, you need to configure agent CTI information in the <code>CTI_AGENT</code> table (see Agents). In the sample profiles provided by Jacada WorkSpace, the configuration mode is already configured.
CTICustomItemAutoAnswer (Genesys only)	- true - false	Specify whether Jacada WorkSpace should automatically answer custom items. For details, refer to the Jacada WorkSpace Developer Guide.
CTICustomItemFilterClass		The fully qualified name of the class used to implement a filter for custom items. For details, refer to the Jacada WorkSpace Developer Guide.

Setting	Values	Description
CTICustomItemNameList (Genesys only)		A comma-separated list of the names of the Genesys Open Media types, as defined in Genesys CME. For example: <code>WorkItem, Fax</code> For details, refer to the Jacada WorkSpace Developer Guide.
CTICustomItemSkillName (Genesys only)		The name of the Genesys skill that the agent must have in order to handle the Open Media types defined in <code>CTICustomItemNameList</code>. For details, refer to the Jacada WorkSpace Developer Guide.
CTIDefaultTransferType	<code>cold</code> <code>handshake</code> <code>warm</code>	Specify the default transfer type to use when clicking the <i>Transfer</i> button in Jacada WorkSpace. For details, refer to the Jacada WorkSpace Developer Guide.
CTIDisplayLoginLink (Genesys only)	<code>true</code> <code>false</code>	Specifies whether or not the <i>CTILogin</i> and <i>CTILogout</i> links are displayed in Jacada WorkSpace. This flag is relevant only when the <i>CTISeparateLogin</i> flag is set to <code>true</code>.
CTIEnableCustomItems (Genesys only)	<code>true</code> <code>false</code>	Set to <code>true</code> to enable Jacada WorkSpace to handle custom items. For details, refer to the Jacada WorkSpace Developer Guide.

Setting	Values	Description
CTIEnableDTMFButton	• <code>true</code> • <code>false</code>	Specifies whether or not the <i>DTMF</i> button is enabled. The <i>DTMF</i> button enables agents to send <i>DTMF</i> tones during a call.
CTIEnableReconnect (Genesys only)	• <code>true</code> • <code>false</code>	Specifies whether to enable reconnecting to CTI after a lost connection is restored.
CTIEnableSoftReturnButton (Genesys only)	• <code>true</code> • <code>false</code>	Specifies whether or not the <i>Toggle</i> button is enabled. The <i>Toggle</i> button allows agents to switch between an active call and a call on hold.
CTIGatewayShouldGenerateAgentStatusEvents	• <code>true</code> • <code>false</code>	This setting is relevant only when the CTISwitchReportsAgentEvents setting is <code>false</code>. Specifies whether the CTI Gateway application should simulate agent status events, instead of the switch. When working with Genesys, Genesys is responsible for simulating agent status events, and this setting is not relevant. In the <code>CTI_CTC_AVAYA</code> profile, for example, the value of this setting is <code>true</code>.
CTIHandshakeTransferConnectDTMFCode		Specifies the transfer code (e.g., <code>*9</code>) associated with a handshake transfer connect.

Setting	Values	Description
CTILink		The name of the logical link, defined in the CTI server, that defines which CTI switch to work with.
CTILoginByPhone	• <code>true</code> • <code>false</code>	Set to <code>true</code> when the agent needs to perform the login to CTI via the phone, instead of automatic login as part of logging in to Jacada WorkSpace.
CTIMultiSiteSupport (Genesys and Cisco only)	• <code>true</code> • <code>false</code>	Specify whether to use Genesys multi-site support.
CTINumberOfAttachedDataKeys (Genesys only)	<code><numeric></code>	The total number of attached data keys to use for filtering voice items.
CTIOutboundAccessKey	<code><numeric></code>	A number that needs to be dialed before dialing an outbound call (such as 9).
CTIPassword		A password that corresponds to the <code>CTIUserName</code>. When the password is entered or edited via the Jacada Management Console, the password is automatically encrypted when added to the database table, and decrypted before its use.
CTIPort (Genesys and Cisco only)		The port to use to connect to the CTI server.

Setting	Values	Description
CTIProvider	• manual • genesys • cisco • ctconnect • aes	The CTI provider in use.
CTIReportAllEventsToClient	• true • false	When true, specifies that every CTI event that reaches the CTI Gateway is also forwarded to an external application. For details, refer to the Jacada WorkSpace Developer Guide.
CTISeparateLogin (Genesys only)	• true • false	Specifies whether or not agents may log in and out of CTI independently from logging into WorkSpace. The default value is <code>false</code> (agents do not have separate CTI login ability).
CTIServer		The name or IP address of the machine that houses the CTI server.
CTIServerTimeoutInSec (Genesys only)	60	Specify, for the CTI server, the maximum period of time, in seconds, for receiving responses.
CTISetNumberOfRecentlyDialed Numbers	Integer	Defines the number of recently dialed numbers that are displayed in the Dial List. The default value is 5.

Setting	Values	Description
CTISwitchAgentStatusOnLogin	- ready - notready	Specifies the default agent status upon login.
CTISwitchAllowsHoldInInternalCalls	- true - false	Specify whether the switch allows holding a direct-dial (internal) call. The <i>Hold/Off Hold</i> button behaves accordingly. For details, refer to the Jacada WorkSpace Developer Guide.
CTISwitchAutoAnswer	- true - false	Specify whether the switch is configured to automatically answer calls. For details, refer to the Jacada WorkSpace Developer Guide.
CTISwitchReportsAgentEvents	- true - false	Specify whether the switch reports agent events to the CTI Gateway. Typically, the switch does report agent events. However, there are switches that do not report agent events, such as AVAYA.
CTISwitchSendsRepeatedAgentStatusEvents	- true - false	Specify whether the switch sends repeated agent status events.
CTISwitchType	- avaya - alcatel	The name of the switch in use.

Setting	Values	Description
CTITransferTypesList	• cold • handshake • warm	A comma-separated list of the transfer types supported by the CTI provider and switch in use. For details, refer to the Jacada WorkSpace Developer Guide.
CTIUserName		A user with access to the CTI server. Used when the CTI provider is AES, or Genesys with a multi-site configuration.
CTIVoiceFilterClass (Genesys only)		The fully qualified name of the class used to implement a filter for voice items. For details, refer to the Jacada WorkSpace Developer Guide.
CTIWarmTransferConnectDTMFCode		Specifies the transfer code (e.g., *9) associated with a warm transfer connect.
OneXPort (Avaya one-X Agent only)	60000 (default)	The port number that the client registers on (can range from 60000-61000). The exact port number can be found in HKCU/Software/AvayaonexAgent/Settings/APIPort.

Document Search

Document Search enables managing the document search related capabilities that are available in Jacada WorkSpace:

- [Directories](#)
- [Indexing](#)
- [Settings](#)

To ensure that results from search portlets are displayed in the correct destination, for example, in a new browser window, each agent machine is required to have its *Windows Folder Option File Type* setting enabled with the *Browse in same window* option.

To enable the File Type setting option for each applicable document file type:

1. In the Jacada WorkSpace machine, right-click the *Start* menu.
2. In the menu, select the *Explore* option.
3. In the displayed *Start Menu* window, select *Tools > Folder Options... > File Types* tab.
4. In the *File Types* tab, select from the *Registered file types* pane a file type for those source documents that are available to be searched by Jacada WorkSpace. For example, *Microsoft Office Word Document*.
5. In the *Details for <selected_file_type> extension* pane, click the *Advanced* button.
6. In the displayed *Edit File Type* dialog, check the *Browse in same window* option.

Note: This option is not available for every document file type. For example, the *Adobe Acrobat Document* file type. Such a file type does not require any option enabled.

7. Click *OK*.
8. Repeat Step through Step , per applicable document file type of the documents available for search.
9. Click *Close*.

Directories

The *Directories* page enables defining, per project, the directories containing the documentation to be searched.

Field	Description
Project ID	Name of a Jacada WorkSpace project to which the document search directory entry applies. An asterisk (*) specifies that the entry applies to all projects.
Document Directory	Defines the full path to a document search directory that is used with the defined project. A valid path is any path that is accessible from the host machine of the application server in which the defined project is deployed.
Indexing Language	Defines the main language used in the documentation residing in the document search directory entry. Identifying documentation language enables Jacada WorkSpace to use indexing techniques, when generating a project document index, that improve document searching. Select a Java Locale code from the dropdown menu, for example <code>en</code> for English, <code>zh</code> for Chinese (traditional) or <code>pt_BR</code> for Brazilian Portuguese. See Locales about locales available to a project.
Action	Actions, such as <i>Edit</i> , available for use with each document search directory entry in the table.

Indexing

The *Indexing* page enables indexing of the documents contained in their defined document search directories (described in Directories).

Search Directories

Field	Description
Project ID	The project name of a Jacada WorkSpace project deployed in the same application server as is deployed the Jacada Management Console.
Indexing Status	Displays the indexing status of the documents residing in the project's defined document search directories.
Next Scheduled Indexing	Valid time format is 00:00 through 23:59. The setting's default value is 00:00. Defines the time of day during which daily document indexing is scheduled to occur for the project's documents. For newly deployed projects, make sure to edit this setting and customize its value.
Schedule	Valid values are <i>true</i> and <i>false</i>. The setting's default value is <i>false</i>. Defines whether the <i>Next Scheduled Indexing</i> of project documents is activated to occur or not. For newly deployed projects, make sure to edit this setting and customize its value; <i>check</i> to activate scheduled indexing (<i>true</i>) and <i>uncheck</i> to deactivate scheduled indexing (<i>false</i>).
Action	Actions, such as <i>Edit</i> , available for use with each project indexing information entry in the table.

- Clicking the *Refresh* button updates the *Indexing Status* of all the projects displayed in the table.
- Clicking the *Index* button launches, for the selected projects, an unscheduled indexing of the

documents contained in their defined document search directories, described in Directories. This action generates a new index for the project's documents. Index processing, per document search directory, may require several minutes to complete, depending upon directory size.

Note: Due to indexing's heavy processing, Jacada recommends that indexing should only be performed during periods of low system utilization.

Settings

The *Settings* page provides the ability to manage the settings for both the use and the display of the *Document Search* and *Help-On-Hand* portlets:

Field	Description
Document Display Destination	Valid values are <i>tab</i> and <i>browser</i>. The setting's default value is <i>tab</i>. Defines the destination in which Jacada WorkSpace displays a selected search result. Setting applies to both the <i>Document Search</i> and the <i>Help On Hand</i> portlets. Based on the current setting, either one of the following media opens for display: • The <i>Document Display</i> tab of Jacada WorkSpace. • An Internet Explorer browser window/tab. Note: Regardless of this setting's value, the user's <i>Windows Folder Option File Type</i> setting can cause an unintended display destination. To prevent this from occurring, the File Type setting for each applicable document file type must have the <i>Browse in same window</i> option enabled.

Field	Description
Maximum Help-On-Hand Results	Valid integer value starting at 1. The setting's default value is 3. Defines the maximum number of results, per search, that the <i>Help On Hand</i> portlet displays.
Indexing Root Directory	Defines the path to the root directory where Jacada WorkSpace saves its document indexing results. Setting applies to both the <i>Document Search</i> and the <i>Help-On-Hand</i> portlets.
Results Page Size	Valid integer value starting at 1. Defines the maximum number of results per page, in a document source results pane, that the <i>Document Search</i> portlet displays.
Results Mode	Valid values are <i>combine</i> and <i>separate</i>. The setting's default value is <i>separate</i>. Mode used by the <i>Document Search</i> portlet to display its search results. Either combine results from all document sets into one results pane or separate results into multiple result panes, based on their document source. Combining results produces only one page of results in the results pane.

Layout Management

The *Layout Management* node enables you to create and modify WorkSpace layouts, and manage properties of portlets. Layout Management consists of the following:

- [Layouts](#)
- [Portlets](#)
- [Tabs](#)
- [Transport](#)

Layouts

The layout of Jacada WorkSpace defines which available portlets appear, where on the page they are displayed, whether their location and/or size is adjustable, etc. The *Layouts* page lists available layouts, and allows you to add and modify layouts.

Layouts Page

Field	Description
Layout ID	Unique identifier for the layout. Generally, this identifier is created by combining the Layout ID and the Project ID to build a unique compound ID.
Project ID	Name of a Jacada WorkSpace project to which the layout applies. An asterisk (*) specifies that the layout applies to all projects.
Default	Indicates whether or not the layout is the default layout (when several layouts are available). A layout can be set as the default for a specific project (e.g., Default=true, Project ID=projA) or for all projects (Default=true, Project ID=*).

Edit Layout Page

Clicking *Edit* or *Add* displays the *Edit Layout* page, which lists the following additional parameters:

Field	Description
Layout ID	Unique identifier for the layout. Generally, this identifier is created by combining the Layout ID and the Project ID to build a unique compound ID.
Display Name	Localized layout name for management console.
Project ID	Name of a Jacada WorkSpace project to which the layout applies. An asterisk (*) specifies that the layout applies to all projects.
Interaction Type	Enables you to specify a context in which the layout is displayed (<i>Voice</i> , <i>Work Item</i> , or <i>Tasks</i>). By default, the setting is <i>General</i> (i.e., the layout is always displayed).
Default	Indicates whether or not the layout is the default layout (when several layouts are available). A layout can be set as the default for a specific project (e.g., Default=true, Project ID=projA) or for all projects (Default=true, Project ID=*).
Enable Areas Splitters	Indicates whether or not areas within the layout (e.g., left area, bottom area, etc.) can be resized using splitters.
Enable Tabs Drag and Drop	Indicates whether or not the order of tabs can be rearranged using drag-and-drop.

The lower portion of the *Edit Layout* page enables you to customize a layout by selecting the portlets that appear in each area of the workspace (top, bottom, left, and right). Design each area by transferring the relevant portlets for that area from the *Available Portlets* list to the *Selected Portlets*

list.

Note: Width and height of a specific area are determined by the input in the *Width* and *Height* textboxes. Inputs can be expressed as a percentage, or as a number (of pixels). (For example, 30%, or 250.)

Portlets

The *Portlets* page lists available portlets. It also allows you to add new portlets, and modify portlet properties.

By default, the following portlets are supplied out-of-the-box:

- **Guidance:** Enables agents to access self service Interactions from the Jacada Interaction Server.
- **Sample:** A generic portlet that is used as a starting point for developing new portlets.
- **SmartPad:** Enables agents to store and view information required during a call.

The screenshot shows the Jacada Management Console interface. The top header features the Jacada logo and the title "Management Console". Below the header, a breadcrumb trail reads "WorkSpace > Configuration > Layout Management > Portlets". A left-hand navigation menu lists various configuration options, with "Portlets" highlighted at the bottom. The main content area displays a table of portlets with columns for "Portlet ID", "Project Id", "Privilege", and "Action". The table lists three portlets: "Guidance", "SamplePortlet", and "SmartPadPortlet". Each row includes a checkbox, the portlet name, a project ID of "*", a privilege name, and an "Edit" link. Below the table are "Add" and "Remove" buttons.

	Portlet ID ▼	Project Id	Privilege	Action
☐	Guidance	*	PortletGuidancePortlet	Edit
☐	SamplePortlet	*	PortletSamplePortlet	Edit
☐	SmartPadPortlet	*	PortletSmartPad	Edit

[Show table properties >>](#)

Portlets Page Settings

Field	Description
Portlet ID	Unique identifier for the portlet.
Project Id	Name of a Jacada WorkSpace project to which the layout applies. An asterisk (*) specifies that the layout applies to all projects.
Privilege	The name of the privilege required for accessing the portlet. If no privilege is specified, all agents are able to view the portlet. For more information about privileges, refer to Privileges.

Additional Parameters

Clicking *Edit* or *Add* displays another page, which lists the following additional parameters:

Field	Description
Portlet ID	Unique identifier for the portlet.
Title	Header of the portlet, as it appears in Jacada WorkSpace.
Project ID	Name of a Jacada WorkSpace project to which the layout applies. An asterisk (*) specifies that the layout applies to all projects.

Field	Description
URL	URL of the portlet's content. The field supports: • Project variables, e.g., <code>\${param_name}</code> • A URL to a .jsp file, e.g., <code>SYSTEM/portlets/sample/index.jsp</code> • A fully qualified ExtJS class name, e.g., <code>Jacada.system.ui.sample.MyPortlet</code> When Dynamic View dashboards are defined in the database schema, the <i>Dynamic View Dashboard</i> dropdown list appears, and is automatically populated with the names of the defined dashboards. When a dashboard is selected from the dropdown list, the URL of the relevant Dynamic View appears automatically in the <i>URL</i> field.
URL Parameters	Additional URL parameters (list of key=value pairs, separated by &).
Privilege	The name of the privilege required for accessing the portlet. If no privilege is specified, all agents are able to view the portlet. For more information about privileges, refer to Privileges.
Collapsible	Indicates whether or not the portlet can be collapsed and expanded. The default value is <code>true</code>.
Width	The width of the portlet. The width can be expressed as a percentage, or as a number (of pixels). (For example, 30%, or 250.)
Height	The height of the portlet. The height can be expressed as a percentage, or as a number (of pixels). (For example, 30%, or 250.)

Tabs

The *Tabs* page lists available tabs, and enables you to configure the order of their appearance in Jacada WorkSpace. It also provides functionalities for editing the parameters of existing tabs, and creating new tabs.

Field	Description
Tab Id	Unique identifier for the tab.
Project Name	Name of a Jacada WorkSpace project to which the tab applies. An asterisk (*) specifies that the tab applies to all projects.

Clicking *Set Tab Appearance Order* displays the *Tab Appearance Order* page, which enables you to use *Up/Down* buttons to arrange the order of tab appearance in WorkSpace.

Adding and Modifying Tabs

Clicking *Edit* or *Add* displays the *Edit Tab Settings* page, which enables you to update tab parameters, or create new tabs. Tabs can contain Web applications or Windows applications. General settings apply to all tabs, while other settings vary according to tab content type.

General Settings

General settings are relevant to tabs of all application types.

Field	Description
Tab Id	Unique identifier for the tab.
Header	Title of the tab, as it appears in Jacada WorkSpace.
Access Key	The keyboard shortcut for opening the tab.

Field	Description
Privilege	The name of the privilege required for accessing the tab. If no privilege is specified, all agents are able to view the tab. For more information about privileges, refer to Privileges.
Interaction Type	Enables you to specify a context in which the tab is visible (<i>Voice</i>, <i>Work Item</i>, or <i>Tasks</i>). By default, the setting is <i>General</i> (i.e., the tab is always displayed).
Tab Type	Type of tab content: • Web [Frame]: Indicates a Web application. • Screen Pop [Frame]: Indicates a tab that is displayed only in a special context (e.g., in Voice context, the tab could display customized information about the caller). • Windows [Hosting]: Indicates a nested Windows application.
Removable	Indicates whether or not the tab can be removed from the tabs area of WorkSpace.
Maximizable	Indicates whether or not the tab can be maximized (to occupy the entire work area).
Project Name	Name of the project in which the application is based.

Web Application Settings

The following configuration settings are relevant to tabs containing Web applications:

Field	Description
Frame Id	Unique identifier for the frame.

Field	Description
URL	URL of the tab's content. The field supports: • Project variables, e.g., <code>\${param_name}</code> • A URL to a .jsp file, e.g., <code>SYSTEM/portlets/sample/index.jsp</code> • A fully qualified ExtJS class name, e.g., <code>Jacada.system.ui.sample.MyTab</code> If an absolute path is used, include the protocol path (http://). When a predefined Dynamic View is selected from the <i>Choose a Dynamic View</i> dropdown list, the URL of the relevant Dynamic View appears automatically in the <i>URL</i> field. Note: If the HTTPS protocol is used, a Windows application tab needs to be defined, using IE as a nested application. For details, refer to Using Internet Explorer as a Nested Application. A URL for a Screen Pop tab can contain placeholders that can be replaced at the project level. Placeholders in the URL must be unique. The following placeholders will be replaced automatically by the product when EMC is the provider: Caller name (CallerNameValue), Caller number (CallerNumberValue), Prompted Digits (PromptedDigitsValue), UII (UIIValue), Email (EmailValue), DateTime (DateTimeValue). The placeholders for the values must be entered exactly as presented above. For example: <i>USER/portlets/sample/PrintURLRequestParameters.jsp?callerNumber=CallerNumberValue&dateTime=DateTimeValue</i>
Load tab content on user login	Indicates whether or not tab content is automatically loaded when a user logs into Workspace. This setting does not appear for Screen Pop tab types.
Hide tab on user login	Indicates whether or not the tab is hidden when the agent logs in. This setting does not appear for Screen Pop tab types.

Windows Application Settings

Configuration settings for tabs containing Windows applications are described in the table below.

Note: When working with nested applications, only 32-bit applications are supported.

Field	Description
Hosting Id	The numeric identifier of the machine hosting the application. This property must not contain white spaces.
Application Source	Indicates whether the application is retrieved from the file system or from the registry. For further details about file system and registry settings, refer to File System Settings and Registry Settings.
Link Text	The text of the link that reopens a closed tab. This parameter is relevant only when the <i>Automatically Reopen</i> checkbox is cleared.
Background Color	The color of the background of the tab. The default background color is black.
Detect Busy	Indicates whether or not agents are able to proceed with other work while a nested application is busy. When the checkbox is cleared, agents are unable to initiate or continue tasks until the busy application becomes available.
Busy Message	The text of the message displayed when the application is busy. This parameter is relevant only when the Detect Busy checkbox is checked. When the checkbox is checked, and a message is not specified, a default message is loaded.

Field	Description
Show Load Message	Indicates whether or not a message is displayed while the nested application is loading into the tab.
Loading Message	The text of the message displayed while the application is loading. This parameter is relevant only when the Show Load Message checkbox is checked. When the checkbox is checked, and a message is not specified, a default message is loaded.
Center Application in Tab	Defines which windows of the nested application are centered in the tab. Select one of the following options: • All windows are centered. • No windows are centered. • Only the windows not containing menus are centered. • Only windows that do not have a parent window are centered.
Hide application windows	A list of windows (within a nested application) that should not be displayed in the tab. Windows are identified by Class:Name,Caption. Multiple windows should be separated by a semicolon, e.g., Centura:Dialog,File;Centura:Dialog,Section.
Automatically Reopen	Indicates whether or not the application reopens automatically when an agent closes it.
Provide Scrolling	Indicates whether or not a scrollbar can be created in the tab.
Monitor application in its tab	Indicates whether or not a busy application is monitored, so that when it becomes available it opens in its designated tab only.

Field	Description
Exclude application processes (Hide)	Names of processes running in the background that might prevent the application from closing. Processes listed in the field are ignored by Windows, allowing the application to close. Multiple processes must be comma separated.
Open applications windows outside nested area	Indicates whether nested windows (e.g., dropdown lists, etc.) always open within the area of the host window. To open windows outside the nested area, the string <code>AvoidPlaceInsideHost</code> must appear in the relevant window class.

File System Settings

The following settings are relevant when a Windows application for a tab is accessed through the file system.

Note: The nested window application should be a 32-bit application.

At the top of the *Windows Application Settings* frame, in the *Application Source* dropdown list, verify that *File System* is selected.

Field	Description
Executable Name	Name of the executable file that runs the desired application. The field also supports project variables, e.g., <code>\${param_name}</code> .
Executable Location	Location of the executable file. The field also supports project variables, e.g., <code>\${param_name}</code> .
Executable Arguments	Other additions/specifications for the file name. The field also supports project variables, e.g., <code>\${param_name}</code> .

Registry Settings

The following settings are relevant when a Windows application for a tab is accessed through the Windows registry. At the top of the *Windows Application Settings* frame, in the *Application Source* dropdown list, verify that *Registry* is selected.

All registry keys must be defined in the registry under the **HKEY_LOCAL_MACHINE** root. However, when entering the registry keys in the *Tabs* page, the file paths should **not** include the root folder.

Field	Description
Executable Registry Key	The registry key specifying the name of the executable file of the desired application. The field also supports project variables, e.g., \${param_name}.
Location Registry Key	The registry key specifying the location of the executable file. The field also supports project variables, e.g., \${param_name}.
Arguments Registry Key	The registry key specifying other additions/specifications for the file name. The field also supports project variables, e.g., \${param_name}.

Using Internet Explorer as a Nested Application

When a secured URL is used, or when you want to display a web application that cannot be contained in an i-frame, Internet Explorer should be used as a nested application. The required URL is then loaded in the nested IE.

Field	Description
Executable Name	iexplore.exe
Executable Location	C:/Program Files (x86)/Internet Explorer

Field	Description
Executable Arguments	-nomerge -k <a href="https://<your URL>">https://<your URL> For example: -nomerge -k https://vpn.jacada.com

Transport

Clicking the *Transport* node displays the Transport Wizard, which enables developers to share layout-related content. Content is transferred via an automatically generated file called a Transport Package. Transport Packages can contain one or more of the following layout-related categories:

- Layouts
- Portlets
- Dashboards (Dynamic Views)
- Tabs

A Transport Package automatically incorporates all of the entities of a selected category that are defined in the database schema (e.g., all portlets, etc.).

The following sections describe:

- [Exporting Transport Packages](#)
- [Importing Transport Packages](#)
- [Upgrading Layout-related Components](#)

Backing Up and Restoring Layout Components

The *Backup and Restore* page enables you to:

- Back up all layout components in your database into an .xml package.
- Load the content of a backup .xml package into your database. When this content is loaded, all existing data in the database is overwritten.

To back up layout components:

1. Navigate to the *Backup and Restore* page.
2. In the *Backup the Layout* frame, in the *Package Name* textbox, enter a relevant name for the Transport Package. It is not necessary to specify a file extension; this is generated automatically.
If no name is entered, the Transport Package is given a default name.
3. If desired, in the *Author* and *Comments* fields, enter appropriate additional information about the package.
4. Click *Backup*.
The Transport Package is created, and the *File Download* window appears.
5. Save the file to the desired location.

Note: If an error occurs during the backup process, an error message appears, and the process stops. Partially compiled Transport Packages are not exported.

To back up layout components:

1. Navigate to the *Backup and Restore* page.
2. In the *Restore the Layout* frame, click *Browse*, and select the desired Transport Package (.xml file).
The file path of the selected package appears in the *Path* field.
3. Click *Restore*.
The entities in the selected package are imported into your database schema. When this content is loaded, all existing data in the database is overwritten.

Note: If an error occurs during the backup process, an error message appears, and the process stops. Partially compiled Transport Packages are not exported.

Exporting Transport Packages

Exporting a Transport Package involves selecting the package's content categories, and then downloading the Transport Package to a specified location. Content categories that can be exported are layouts, portlets, dashboards, and tabs.

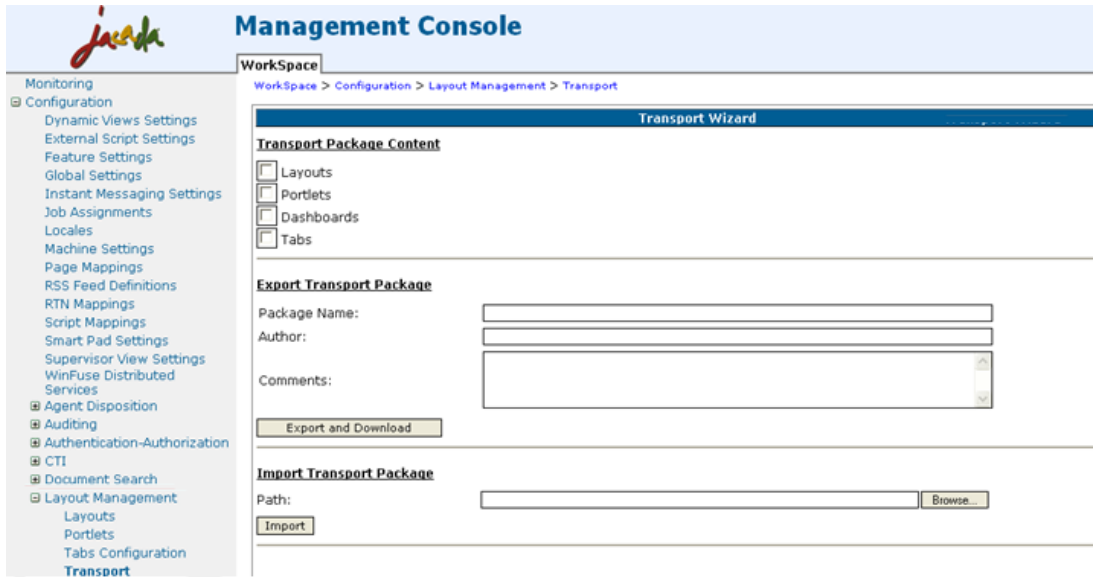
Note: Only dashboards that are marked as published, and dashboards created using administrative credentials can be exported.

To export a Transport Package:

1. At the top of the *Transport Wizard* page, in the *Transport Package Content* frame, verify that the checkboxes of the layout categories to be included in the Transport Package are checked. (By default, all categories are checked.)

The components (and nested components) of layout categories that are not selected are not exported.

Note: When an exported layout contains dashboards, but the *Dashboards* layout category is not selected (i.e., only the *Layouts* category is selected), sample dashboards may appear in the layout after it is imported into another database schema. This is a result of the system's attempt to repair broken links to the dashboard.



2. In the *Export Transport Package* frame, in the *Package Name* textbox, enter a relevant name for the Transport Package. It is not necessary to specify a file extension; this is generated automatically.
If no name is entered, the Transport Package is given a default name.
3. If desired, in the *Author* and *Comments* fields, enter appropriate additional information about the package.
4. Click *Export and Download*.
The Transport Package is created, and the *File Download* window appears.
5. Save the file to the desired location.

Note: If an error occurs during the export process, an error message appears, and the process stops. Partially compiled Transport Packages are not exported.

Importing Transport Packages

Importing a Transport Package involves specifying the relevant package, and selecting the package content categories to be imported.

When you import a Transport Package, its data is automatically incorporated into your database

schema, and imported content is merged with existing layout-related content in your schema. Imported portlets, tabs, and layouts that have the same Project ID and <component> ID as the entities in your schema override existing entity definitions. (Dashboards having the same key as imported dashboards are also overridden.)

Dashboards that are referenced in tabs and/or portlets are imported together with the imported tab or portlet. Portlets that are part of an imported layout are imported together with the layout.

Note: When an exported layout contains dashboards, but the *Dashboards* layout category was not selected upon export (i.e., only the *Layouts* category was selected), sample dashboards may appear in the layout after it is imported into another database schema. This is a result of the system's attempt to repair broken links to the dashboard.

Generally, all imported layouts are brought into the database schema with a **Default** value of False. The desired default layout can then be configured later.

To import a Transport Package:

1. Navigate to the *Import Layout* page.
2. In the *Upload Transport Package* frame, click *Browse*, and select the desired Transport Package (.xml file).
The file path of the selected package appears in the *Path* field.
3. Click *Upload*.
The content of the selected Transport Package is loaded to the screen.
4. Select the checkboxes of the layout elements that you want to import (*Layouts*, *Portlets*, *Tabs*). Then, select specific dashboards to be imported.
5. Click *Import*.
The entities in the selected layout categories are imported into your database schema.

Note: If an error occurs during the import process (e.g., importing an entity that you do not have privileges to view), an error message appears, and the process stops. Partially imported data not appear in your schema.

Upgrading Layout-related Components

The upgrade feature of the Transport Wizard enables developers to Import portlet and layout-related definitions to the database after upgrading to a newer version of WorkSpace.

The process involves importing data from a .zip package containing portlet and layout-related .xml files from the WorkSpace project of the previous version.

Note: The current version does not support upgrade of tab definitions from **tabs.jspf**. Tabs need to be added to the database schema using the *Tabs* node of the Jacada Management Console. For details, refer to Tabs.

Creating the Upgrade Package

The content of the .zip package is based on a specific WorkSpace project. The .zip file must be named **layout_%PROJECT_NAME%.zip**, where **%PROJECT_NAME%** is the name of the WorkSpace project in the previous version. The project name parameter is used to register layout components to the database schema, and to assign the components to a specific Project Id.

The .zip package must contain the following directories:

- **Layouts:** Contains **layouts.xml** and all custom layout .xml files, located under **/%PROJECTID%/WebContent/USER/resources/layouts**.
- **Portlets:** Contains **portlets.xml** and all portlet-related .xml files, located under **/%PROJECTID%/WebContent/USER/resources/**.

Note: Verify that all relevant .xml files are included in the .zip package. If some files are missing, the upgrade process fails.

Importing the Upgrade Package

Importing the .zip package is done using the Import function of the Transport Wizard.

To import an upgrade package:

1. At the bottom of the *Transport Wizard* page, in the *Import Transport Package* frame, click *Browse*, and select the desired .zip package.

The file path of the selected package appears in the *Path* field.

Note: It is not necessary to select content categories at the top of the page. All components included in the upgrade package are automatically imported.

2. Click *Import*.

Content of the upgrade package is imported into the database.

Note: If an error occurs during the upgrade process, an error message appears, and the process stops.

Project Variables

Project Variables enables managing the settings in the project code that are project cycle dependent. The variables are defined per project cycle, in the following available project cycles:

Common – Variables that apply to all project cycles. The values specified for these variables are inherited by all the project cycles, unless overridden by a different value specified for a specific project cycle.

Development – Variables that apply to the development cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

Pre-production – Variables that apply to the pre-production cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

Production – Variables that apply to the production cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

QA – Variables that apply to the QA cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

System Test – Variables that apply to the system test cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

Training – Variables that apply to the training cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

User Acceptance Test— Variables that apply to the user acceptance test cycle. The values specified for these variables are specific to this cycle. In addition, variables which do not have a specific value specified for this cycle are inherited from the common variables.

Task Manager

The *Task Manager > Settings* page enables you to control specific settings of the following portlets in Jacada WorkSpace:

- Agent Tasks
- Agent Disposition

Field	Description
Number of Days to Show Due Tasks	The number of days (previous to the current date) for which past-due tasks are listed in the Agent Tasks portlet. For example, if the current date is October 1, all tasks due on and after September 1 (regardless of their status) appear in the Agent Tasks portlet. All tasks with due dates in the future are listed in the Agent Tasks portlet.
Display a Callback Button	Defines whether or not the <i>Add Callback</i> button appears in the <i>Agent Disposition</i> portlet. Clicking the button displays the <i>Add Callback</i> window, which provides a template for designing a callback task. The callback task can then be added to the tasks list in the <i>Agent Tasks</i> portlet.

Logging

Manage the Jacada WorkSpace logging configuration using the Jacada Management Console's *Logging* navigation menu option. Configure nested application logging in each, specific agent desktop that is installed with the required ActiveX control for nested application operation in Jacada WorkSpace tabs. For information on Jacada WorkSpace logging, see [Logging Capabilities](#).

Note: The `JacadaLogConfiguration.xml` contains loggers, for 3rd party software, such as Beehive, Hibernate, and Spring, and `Jacada.Debug.JFAP.DEV_INTERNAL`. These loggers are for internal use only. Do not modify these loggers.

Loggers

The page provides the following:

Field	Description
Select logger	The name of the logger. The table of logging destinations displays the list of log destinations available with the default destination for the selected logger.
Log level	Determines what information is written to the logs. The ALL level allows output of all log messages. At the other end of the scale, the OFF level turns off all output. The levels between ALL and OFF place increasingly growing limitations on the output of messages. When going from one level to the next, less messages are output and they are a result of error situations of increasing severity.
JFAPSingleFileAdmin	A RollingFileAppender that writes Jacada WorkSpace administration messages.
JFAPSingleFileScript	A RollingFileAppender that writes call script specific messages.
JFAPCTIFile	A RollingFileAppender that writes CTI messages.
FusionFile	Backs up the log file when it reaches a certain size and creates a new log file.

Field	Description
JFAPCTIGatewayFile	A RollingFileAppender that writes CTI Gateway messages.
AilAppender	A RollingFileAppender that writes Genesys AIL messages.
HibernateFile	A RollingFileAppender that writes Hibernate messages.
SpringFile	A RollingFileAppender that writes Spring messages.
RollingFileAppender	Extends the FileAppender and backs up the log file when it reaches a certain size. At that point, it also creates a new log file.
DailyRollingFileAppender	Extends the FileAppender and backs up the log file at a frequency specified by the user.
ConsoleAppender	Writes log events to <code>System.out</code> or <code>System.err</code> .
ClusterDailyRollingFileAppender	Similar to the DailyRollingFileAppender. Supports cluster logging and rolls over log file at a frequency specified by the user.
ClusterRollingFileAppender	Similar to the RollingFileAppender. Supports cluster logging.
LanguagesDailyRollingFileAppender	Similar to the DailyRollingFileAppender. Enables writing non-English characters to the log.
LanguagesFileAppender	Extends the standard FileAppender to support writing UTF-8 encoding (non-English characters) to the log.
LanguagesRollingFileAppender	Similar to the RollingFileAppender. Enables writing non-English characters to the log.

Field	Description
ClusterFileAppender	Similar to the FileAppender. Supports cluster logging.

Log Destinations (Appenders)

The page provides the following:

Field	Description
Select Log Destination	The name of the Log Destination
File	The path and name of the log file.
Append	A boolean value. If true, at server startup messages are appended to an existing log file. If false, a new log file is created at server startup. The default value is true.
Max file size(MB)	Specifies the maximum file size of the log file. When the value is reached, the file is backed up and a new log file is created.
Conversion Pattern	Defines the format of a log message.
Max Backup Index	Specifies how many backups of the log file are kept. By default, there is one backup file.

Monitoring Jacada WorkSpace

The *Monitoring* page enables monitoring of existing WorkSpace sessions, termination of sessions and changing WorkSpace session log levels.

The table contains the following:

Field	Description
Agent Name	The name of the agent logged in.
Application name	Name of the application active for the session.
Script name	Name of the currently running script.
CTI Extension	Agent's CTI Extension.
CTI Status	Agent's CTI status.
Session Start Time	Session start time.
Log Level	Logging level for the session.
Server Name	Name of the host machine that the session is running on.

The following controls are located below the Monitoring table:

Refresh

Refreshes the current view, updating the table with the recent information

Terminate Session

Terminates the selected sessions

Change Log Level

Sets the selected log level for the selected sessions.

Messaging Capabilities

This section describes the messaging capabilities of Jacada WorkSpace . During runtime, system messages are managed from the *Message Center* tab in Jacada WorkSpace. In addition to the unidirectional *Message Center* messages issued by supervisors, Jacada WorkSpace supports instant messaging that provides interactive chat sessions between Jacada WorkSpace users.

The following sections present:

- [Messaging Authorization](#)
- [Message Center Messages](#)
- [Creating Messages Using the Message Center](#)
- [Instant Messaging](#)

Messaging Authorization

To use the *Message Center* tab, the *Message Board* tab, and Instant Messaging (*Chat* portlet), you must assign the required Jacada WorkSpace authorization privilege to the agent. User authorization is defined and managed using the Jacada Management Console. For details, refer to [Authentication and Authorization](#).

You can modify Jacada WorkSpace roles and privileges. In addition, you can define custom roles and privileges that are used in the project to allow or restrict the use of the *Message Center*, Instant Messaging or both.

Authorization Privileges for Messaging

Jacada WorkSpace provides the following authorization privileges for its messaging capabilities:

Messaging Capability	Privilege	Description
Message Center	TabMessageCenter	When enabled, permits use of the <i>Message Center</i> tab. By default, this privilege is assigned to supervisors.
Message Board	TabMessageBoard	When enabled, permits use of the <i>Message Board</i> tab.
Instant Messaging	PortletChat	When enabled, permits use of the <i>Chat</i> portlet.
	SupervisorChat	When enabled, permits the user to chat with Jacada WorkSpace users that are defined as supervisors. This privilege is effective only when enabled in combination with <code>PortletChat</code>.
	ExpertChat	When enabled, permits the user to chat with Jacada WorkSpace users that are defined as experts. This privilege is effective only when enabled in combination with <code>PortletChat</code>.
	PeerChat	When enabled, permits the user to chat with Jacada WorkSpace users that are defined as peers. This privilege is effective only when enabled in combination with <code>PortletChat</code>.
	AgentChat	When enabled, permits the user to chat with Jacada WorkSpace users that are categorized as agents. This privilege is effective only when enabled in combination with <code>PortletChat</code>.

Messaging Capability	Privilege	Description
	MultiChat	When enabled, permits the user to simultaneously conduct multiple chat sessions. This privilege is effective only when enabled in combination with any of the privileges <code>SupervisorChat</code>, <code>ExpertChat</code>, <code>PeerChat</code> or <code>AgentChat</code>. By default, this privilege is assigned to supervisors.
	ChatHistory	When enabled, permits use of the <i>Chat History</i> tab to view user chat activity of the past 24 hours and the chat session log. By default, this privilege is assigned to supervisors.

WorkSpace Groups and Message Related Privileges

The following are the Jacada WorkSpace provided groups and their messaging related privileges:

Group	Relevant Assigned Role	Messaging Related Privileges
Agents	AgentRole	AgentRole assigns the Agents group the following privileges: • <code>PortletChat</code> • <code>SupervisorChat</code> • <code>ExpertChat</code> • <code>TabMessageBoard</code>

Group	Relevant Assigned Role	Messaging Related Privileges
CTIUsers	AgentRole	AgentRole assigns the CTIUsers group the following privileges: • PortletChat • SupervisorChat • ExpertChat • TabMessageBoard
Supervisors	AgentRole, CTIUserRole, SupervisorRole	AgentRole,CTIUserRole and SupervisorRole assign the Supervisors group the following privileges: • PortletChat • SupervisorChat • ExpertChat • PeerChat • AgentChat • MultiChat • ChatHistory • TabMessageBoard • TabMessageCenter

Message Center Messages

Messages created by a supervisor using the Jacada WorkSpace *Message Center* can display in any of the following Jacada WorkSpace locations:

- The Message Board
- The Ticker Tape
- The *Welcome* page

For example, messages that agents must read can be directed for display on the *Welcome* page,

which appears to the user upon log in to a Jacada WorkSpace session.

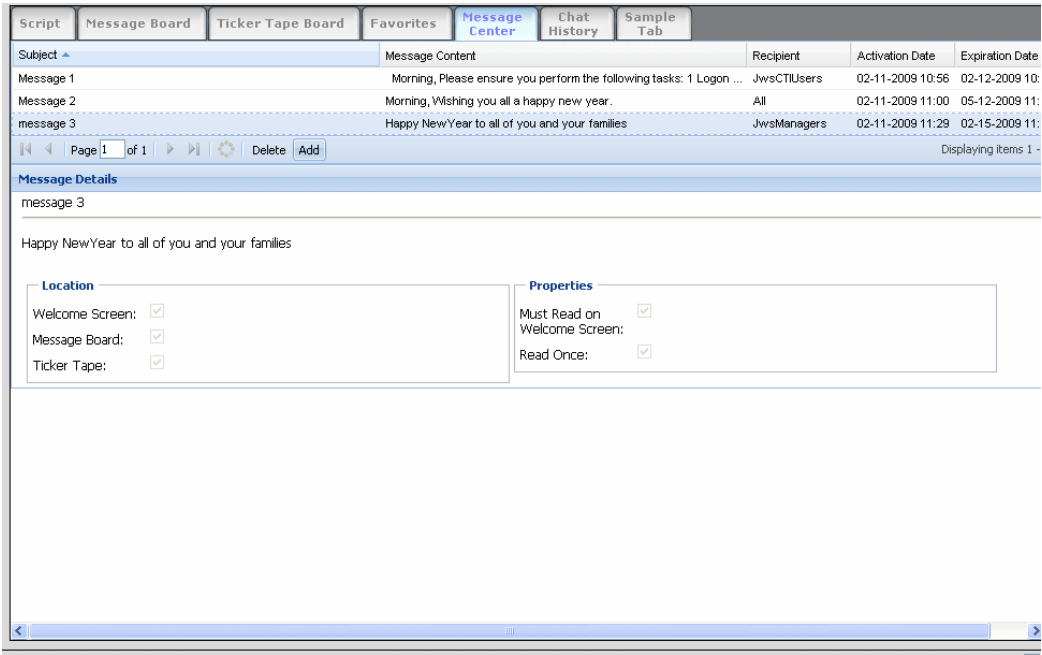
The different message locations are described below.

- **Welcome Page Messages:** After a call center agent logs in, Jacada WorkSpace displays the *Welcome* page. This page lists all current messages. Some of the messages on the Welcome page contain the *Must Read on Welcome Screen* attribute ([Creating Messages Using the Message Center](#)). The agent must open these messages before leaving the *Welcome* page.
- **Message Board Messages:** Messages can be displayed on the *Message Board* tab. The *Message Board* is updated with new messages when the user logs in to Jacada WorkSpace and when the agent ends a call.
- **Ticker Tape Messages:** Messages created in the *Message Center* tab can be displayed on the *Ticker Tape*. The ticker tape properties are defined when adding or editing messages ([Defining Message Output](#)). These include defining the message priority, color, the display date and time range and more.
- **Ticker Tape Board:** The *Ticker Tape Board* tab displays the full message of a message on the ticker tape. The content of this message is replaced each time a ticker tape message is selected.

In addition, the *Ticker Tape Board* tab displays RSS feeds in a separate line of the ticker tape. Before a new RSS item appears on the ticker tape, there is a delay of one minute for the ticker tape to complete a cycle, where the last item to display on the tape finishes.

Creating Messages Using the Message Center

The *Message Center* tab, available to supervisors, enables creating messages and saving them in the database. Messages can be sent to defined groups of Jacada Workspace users.



Supervisors can define message attributes, such as where to display the message (message board, ticker tape board or welcome screen), whether the message appears once or is repeated, the color and frequency of the message on the ticker tape, and so on.

Note: Supervisors can sort messages in the list and add fields to display. These include activation date (the date from which the message is displayed) and expiration date (the date after which the message is no longer displayed). It is recommended to set the activation date to the previous day, so clock inconsistencies are not relevant.

The following sections describe the main features and functions of the *Message Center* tab:

- [Defining Message Details](#)
- [Defining Message Output](#)
- [Editing Messages](#)
- [Deleting Messages](#)

Defining Message Details

This section describes how to define general message details. These include the recipients and the message display range, as well as the subject and content of the message.

To define message details:

1. In the *Message Center* tab, click *Add*. The *Add Message* dialog appears, with the *Mandatory* tab displayed by default.

Add Message

Mandatory Optional

Recipients

Send to: All Group: Select group...

Display Range

Active From: 11-19-2008 13:59

Expires On: 11-20-2008 13:59

Subject:

Message Content:

Tahoma B I U A⁺ A⁻ A¹ ab [List Icons]

Save Cancel

2. In the *Recipients* area, define the message recipients:

- From the *Send to* dropdown list, select the required option, as follows:
 - *All*: Select this option to send the message to all the users in the system.
 - *Group*: Select this option to send the message to selected groups. Each group is a collection of Jacada WorkSpace users. The groups appearing in this dropdown list are defined in the *Groups* page of the Jacada Management Console.

Note: When *All* is selected, the *Group* option is disabled.

3. In the *Display Range* area, set the date and time range to display the message:

- In the *Active From* field, select the starting date and time from which to display the message.
- In the *Expires On* field, select the end date and time for displaying the message.

4. Define the message subject and content in the relevant fields.

Note: Use the formatting bar to format the message. To enter HTML code you must click the *Source Edit*  button.

Defining Message Output

This section describes how to define the message output. These include the display location of the message and its read properties.

To define the message output:

1. Select the *Optional* tab, as follows:

Add Message

Mandatory Optional

Location

- ☒ Welcome Screen
- ☒ Message Board
- ☒ Ticker Tape

Properties

- ☒ Read Once
- ☒ Repeat
- ☒ Must Read on Welcome Screen: 1 day

Ticker Tape

- ☒ Flash Message
- ☒ Pause Message at Center

Priority: Regular Color:

Save Cancel

- In the *Location* area, define where the message is displayed, as follows:
 - To display the selected message on the *WorkSpace Welcome* page, select the *Welcome Screen* checkbox.
 - To display the selected message on the *WorkSpace Message Board*, select the *Message Board* checkbox.
 - To display the selected message on the ticker tape, select the *Ticker Tape* checkbox.
- In the *Properties* area, define the read properties, as follows:
 - To display the message until it is read, check the *Read Once* checkbox. This means, the message will not appear anymore after then next login.

Note: When this checkbox is selected, the *Repeat* field becomes enabled, where you can define whether you want to repeat the message daily, weekly, or monthly, as described below.

- To define the period for repeating the message check the *Repeat* checkbox and then

select the required repeat option from the dropdown list.

- If you want Jacada WorkSpace to require the agent to open this message in the *Welcome* screen listing of messages, check the *Must Read on Welcome Screen* checkbox.

Note: This checkbox is only enabled if the *Welcome Screen* checkbox is selected in the *Location* area.

4. If the *Ticker Tape* checkbox was selected in Step , define the ticker tape properties in the *Ticker Tape* area, as follows:

- To display the message as a flash message on the ticker tape, select the *Flash Message* checkbox.
- To pause the message at the center of the ticker tape, select the *Pause Message at Center* checkbox.
- To define the priority, select the required priority level (*Urgent*, *Important*, *Regular*, *Low*, *Very Low*) from the *Priority* checkbox. Meaning, messages containing the *Urgent* priority level will appear on the ticker tape the most number of times and messages containing the *Very Low* priority level, will appear on the ticker tape the least number of times.

Note: This option is used when there is no space to display all messages on one ticker tape cycle.

- To define the ticker tape message color, either select the required color code from the *Color* dropdown list or enter a color code.

5. Click *Save*.

Editing Messages

After creating a message, the supervisor can edit existing message properties from the *Edit Message* dialog, as required.

To edit a message:

1. From the *Message Center* main window, double-click on the message to edit.
2. The *Edit Message* dialog is displayed.

3. Edit the message content as described in Defining Message Details and Defining Message Output.
4. To save the edited message in the database, click **Save**.

Deleting Messages

The supervisor can remove messages from the system as required. A confirmation message is displayed confirming the removal.

To delete a message:

1. From the *Message Center* tab, select the messages to delete.
2. From the *Message Center* toolbar click **Delete**. The *Confirm Delete* confirmation dialog
3. Click **Yes**. The selected messages are removed from the message list.

Instant Messaging

The Jacada WorkSpace instant messaging feature, provided by the *Chat* portlet in Jacada WorkSpace and supported by an XMPP server, enables supervisors and agents to initiate chat sessions and send interactive messages to one another. By default, you open the *Chat* portlet using the following button, available in the toolbar: 

The *Chat* portlet provides the following tabs:

- The *Contact List* tab
- The *Add Contacts* tab
- *Chat* tabs

By default, agents can maintain one active chat session at a time and can only initiate a chat session with either a supervisor user or an expert user. Supervisors can both initiate and receive an unlimited number of chat sessions. To modify Instant Messaging related privileges, see [Messaging Authorization](#).

Supervisors can override chat sessions at any time and can chat with agents, even when the agent is in a call.

The following sections describe the main administrative topics related to the Instant Messaging feature:

- [Configuring the Connection to the XMPP Server](#)
- [Available Tabs](#)
- [Chat Session History](#)
- [Chat and The Toolbar](#)
- [Using Instant Messaging](#)

Configuring the Connection to the XMPP Server

Instant messaging is supported by an XMPP server, for example, the Openfire XMPP Server. The XMPP server maintains the instant messaging contact list of Jacada WorkSpace users and manages the instant messaging connections between Jacada WorkSpace users. Jacada WorkSpace must know about the machine that is hosting the XMPP server, so that it can connect to relevant machine and communicate with the server.

Using the Jacada Management Console, define the required XMPP server settings to enable

supervisors and agents to chat with each other using their *Chat* portlet in Jacada WorkSpace. For details, see the [Instant Messaging Settings](#).

Note: When an OpenFire XMPP server is used for instant messaging, after adding a contact to the contact list if an exception is generated, restart the OpenFire XMPP server.

Synchronizing Users

The XMPP server maintains the instant messaging contact list of Jacada WorkSpace users. Jacada WorkSpace requests the XMPP server to add new Jacada WorkSpace users and their authorizations but does not request the XMPP server to either modify or delete user information.

Given this relationship between the systems, if a user resigns from the organization or the user's group assignment changes, make sure to remove that user from the XMPP server. This enables the user, when relevant, to be appropriately, added anew via the established Jacada WorkSpace–XMPP server request interaction.

Available Tabs

The *Chat* portlet makes the following tabs available:

Contact List tab – Contains the user's lists of relevant chat contacts; agents, peers, experts and supervisors are maintained in separate contact lists. The user searches these lists for a needed contact with whom to initiate a chat session.

Add Contacts tab – Used to search for and add users to the appropriate contact list in the *Contact List* tab. The *Add Contacts* tab is accessed via the *Add Contacts...* link; this link is provided in the *Contact List* tab.

Multiple chat tabs – Jacada WorkSpace presents each chat session in a separate tab in the *Chat* portlet. Each chat tab displays the contact name with whom the user is in session.

For users, typically agents, assigned the Jacada WorkSpace instant messaging related privileges *SupervisorChat* and *ExpertChat*, via the groups to which they belong, Jacada WorkSpace provides the *Supervisors* list and the *Experts* list of contacts in the *Contact List* tab.

For users, typically supervisors, assigned the Jacada WorkSpace instant messaging related privileges *SupervisorChat*, *ExpertChat*, *PeerChat* and *AgentChat*, via the groups to which they

belong, Jacada WorkSpace provides the *Supervisors* list, the *Experts* list, the *Peers* list and the *Agents* list of contacts in the *Contact List* tab. The *Agents* and *Peers* lists are initially empty. Users add and delete contacts to these lists as needed.

Upon log in to a Jacada WorkSpace session, Jacada WorkSpace populates the user's *Supervisors* list and *Experts* list with those users that are assigned as supervisors or experts to the groups to which the user belongs, as defined in the Jacada Management Console's *Authentication and Authorization* option.

Note: Jacada WorkSpace requires a large amount of processing resources to prepare these contact lists per user logging in. Based on your total agent population, limit the total number of supervisor users to 5% and expert users to 3%. Generally, keep to a minimum the number of supervisor and expert users that are associated with individual users.

Jacada WorkSpace registers supervisors and experts in the XMPP server upon their initial log in to Jacada WorkSpace. In order for supervisors and experts to appear in a user's applicable contact list, the user must have logged in to Jacada WorkSpace following the initial log in to Jacada WorkSpace of the relevant supervisors and experts.

The *Add Contacts* tab provides the ability to add a user as a chat contact. Use of this tab requires the user to have one or more of the enabled privileges: *PeerChat* and *AgentChat*. The tab's contact name input field is for data entry and the tab's pane displays all relevant Jacada WorkSpace user names that match the text entered in the field.

Note: Relevant user names displayed in the pane include those that are either:

- Assigned as peers to the searching user
- Categorized as agents to the searching user; not assigned to the searching user as a supervisor, expert or peer.

Each user is associated with and appears in only one contact list, be it *Experts*, *Supervisors*, *Peers* or *Agents*.

Instant Messaging Icons

The following are the instant messaging icons that are used in the contact lists to indicate the current, chat availability of each contact:

Icon	Description
	The agent is currently online and available to chat. The icon's tooltip displays this chat status.
	The agent is currently engaged in a call. Only supervisors can chat with agent. The icon's tooltip displays this chat status and the elapsed time that the agent has been in call.
	The agent is currently in call disposition. Only supervisors can chat with agent. The icon's tooltip displays this chat status and the elapsed time that the agent has been in call disposition.
	The agent is currently in a chat session. Only supervisors can override the agents' chat session. The icon's tooltip displays this chat status.
	The agent is currently offline. The icon's tooltip displays this chat status.

Should a user either remove the *Chat* portlet from the Jacada WorkSpace layout or log out from Jacada WorkSpace, Jacada WorkSpace takes the following actions:

- Closes all chat sessions of the user.
- Notifies the user's chat mates.
- Changes the user's chat status to offline.

Chat Session History

Jacada WorkSpace provides a *Chat History* tab. Use of this tab requires the user to have the enabled privilege `ChatHistory`. *Chat History* provides the supervisor with sub-tabs:

- *Chat Activity: 24 Hour Window*.
- *Session Log*.

Chat History enables supervisors to view the current number of active chat sessions. Via its sub-tabs, supervisors view statistical data regarding agent chat sessions, as follows:

- The *Chat Activity* tab provides a bar graph, with a rolling window display, showing you the total number of chat sessions that occurred in a hour of the day, during the previous 24 hours.
- The *Session Log* tab provides a listing of chat session entries where each displays the following information: chat initiator, chat recipient, chat session start time and the length of the chat session.
- Update the information for a sub-tab in view by clicking *Refresh*.

Jacada WorkSpace uses the chat related data supplied in the database's auditing tables to populate the *Session Log* tab with information, therefore, Jacada WorkSpace must be auditing chat events.

To enable the auditing of chat events:

1. In the Jacada Management Console navigation menu, open the *Configuration > Auditing > Settings* option and configure the `AuditingEnabled` setting to be `true`.
2. In the same page, make sure that the `DBAuditingEnabled` setting is `true` (this is the default value).
3. Open the *Configuration > Auditing > Events* option and make sure that both the *Chat* and the *Chat Message* audit events are enabled (these events are enabled by default).

Chat and the Toolbar

By default, a user opens the *Chat* portlet by clicking the chat button, , available in the Jacada WorkSpace toolbar.



When the *Chat* portlet is closed, the chat toolbar button will indicate when the portlet has received any of the following notifications from Jacada WorkSpace:

- Start of a chat session with the user.
- Arrival of a new chat message for the user.
- End of an chat session.

Note: When the Chat portlet is available via its toolbar button and the portlet is closed, the portlet is actually only hidden from view. This means that during the Jacada WorkSpace session, the user remains online and the portlet continues to receive start chat notices, new messages and end chat notices that are sent by other users.

The chat button indicates its receipt of these notifications by changing its background color (to dark blue) and flashing several times.

Note: During project development, the default *Chat* portlet access can be changed. Instead of using its toolbar button, the portlet can be made available for Jacada WorkSpace display in the standard portlet manner. For development details, see the *Jacada WorkSpace Developers Guide*.

Using Instant Messaging

When available in the *Contact List* tab, the *Peers* and *Agents* contact lists are initially empty. Users must add and delete contacts to these lists as needed. Use of the *Peers* and *Agents* contact lists requires the user to have one or more of the enabled privileges: *PeerChat* and *AgentChat*.

To add a contact:

1. In the *Contact List* tab, click the available *Add Contacts...* link.
 2. In the contact name input field of the *Add Contacts* tab, type in text (part or all of a user name).
 3. In the pane of the *Add Contacts* tab, select a displayed user name.
The selected user name is added to the appropriate contact list, either *Peers* or *Agents*, in the *Contact List* tab.
 4. Following user name selection, the *Contacts List* tab redisplay in the *Chat* portlet.
- Supervisors can override chat sessions at any time and can chat with agents, even when the agent is in a call.

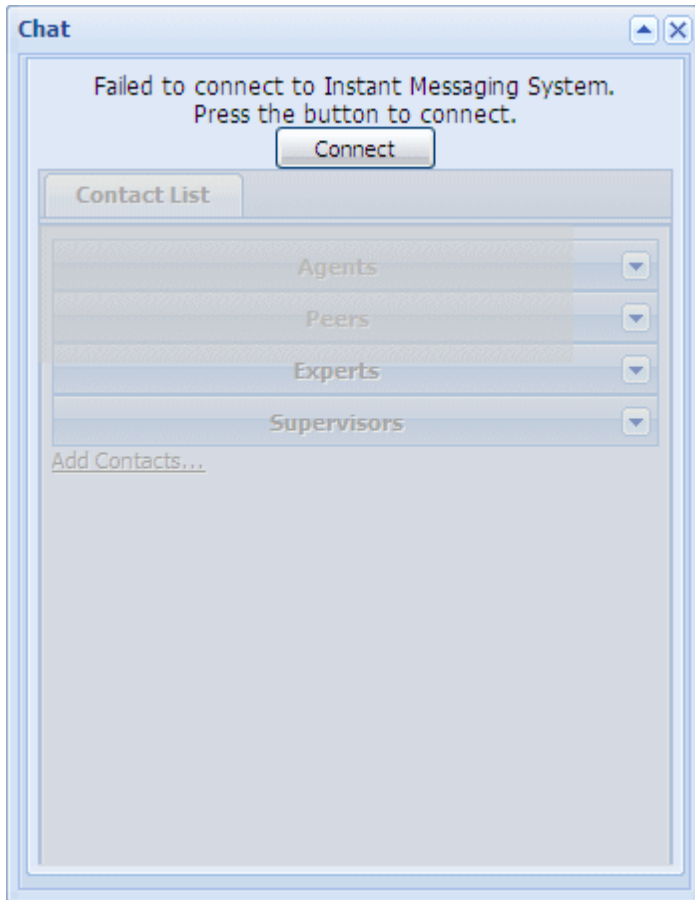
To initiate a chat session:

1. In the toolbar, click the chat button, .
The *Chat* portlet opens and displays on top of Jacada WorkSpace.
2. In the *Contact List* tab, open an available contact list (*Supervisors*, *Experts*, *Peers* or *Agents*).
3. From the contact list, select the user name of the user with whom you want to chat.
A chat tab, dedicated to this specific chat session, opens in the *Chat* portlet.
4. In the lower pane of the chat tab, enter your message and click *Send*.

5. To end the chat session, click *End Chat*.

Each open chat tab displays the contact name with whom the user is in session: the chat mate. The upper pane of a chat tab displays a history of the chat session, where each user message is identified by a blue headline and each chat mate message is identified by a red headline.

Jacada WorkSpace maintains an active communication connection with the Instant Messaging server. If Jacada WorkSpace cannot establish an initial connection with the Instant Messaging server, a message is displayed in the portlet:



To have Jacada WorkSpace make another attempt at establishing a communication connection with the Instant Messaging server, click the available *Connect* button.

If Jacada WorkSpace's existing connection with the Instant Messaging server drops, a message

displays in the *Chat* portlet and Jacada WorkSpace automatically attempts to re-establish its connection with the Instant Messaging server. Until the connection is re-established, the *Chat* portlet is disabled.

Portlet Administration

Some of the portlets provided require configuration before they can be used by an agent. The following topics describe the specific configuration required for these portlets:

- [The Agent Tasks Portlet](#)
- [The Document Search and Help-On-Hand Portlets](#)
- [The Smart Pad](#)
- [The Agent Disposition Portlet](#)

The Agent Tasks Portlet

The *Agent Tasks* portlet enables agents to create and manage their own personal list of tasks. Agent tasks are saved in the database and, therefore, are maintained across Jacada WorkSpace sessions.

The portlet is comprised of the two panes, one above the other:

Task pane – Displays basic information associated with each task entry: a description, due date, due time, reminder and status. A task can be added manually by the agent or by the developer via the project code. Tasks added by the developer in the code, by default, are always editable by the agent. The developer can make an added task not editable in the Agent Tasks portlet.

Additional information pane– Displays any additional information that exists in the database for a task currently selected in the task pane.

Additional information entries are created, when a task is added, using custom processing in the Jacada WorkSpace project code. Additional task information, when it exists, is read-only and cannot be modified by the agent.

Administering the Maximum Number of Tasks

The number of tasks that an *Agent Tasks* portlet can contain is controlled by the Jacada WorkSpace database setting `maximumAgentTasks`. The setting's default value is 30. An administrator modifies this setting using the [Global Settings page](#) in the Jacada Management Console.

To modify the maximum number of tasks that an Agent Tasks portlet can contain:

1. In the Jacada Management Console navigation panel, select the *Configuration > Global Settings* option.

The *Global Settings* page displays.

2. In the *maximumAgentTasks* field, replace its current value with the desired value for the maximum number of tasks that the Agent Tasks portlet can contain.

3. Click Apply.

The *Global Settings* are updated in the Jacada WorkSpace database with the modified value provided for the `maximumAgentTasks` setting.

The Document Search and Help-On-Hand Portlets

Portlets are supplied that either enable searching documents by the agent for information or automatically search documents for information relevant to a call which is then displayed to help the agent:

Document Search – Provides the agent with ability to conduct a text search within documents in specified document sources. This portlet is available from the toolbar.

Help On Hand – Conducts background, text searches, using the project's configured document sources, to proactively provide relevant search results to the agent. This portlet is available from the toolbar.

Jacada WorkSpace is certified to search documents of the following file formats: Microsoft Word (.doc and .docx), Adobe Acrobat (.pdf), rich text format (.rtf) and Microsoft Power Point (.ppt and .pptx).

You specify the document search directories and the scheduling of document indexing for the Jacada WorkSpace document sources in the Jacada Management Console by accessing its navigation menu option *Configuration > Document Search*. The following sections describe each of these administration activities:

- [Defining Search Settings](#)
- [Defining Search Directories](#)
- [Scheduling Document Indexing](#)

Defining Search Settings

Manage document search settings via the *Configuration > Document Search > Settings* option in the Jacada Management Console navigation menu. In the *Settings* page, define and modify the following settings that are relevant to the function of the portlets.

Settings

Setting	Applicable Portlet	Description
Document Display Destination	• Document Search • Help-On-Hand	Setting is used by both portlets. Defines the destination to display selected search results, either in a Jacada WorkSpace tab or a new browser window.
Maximum Help-On-Hand Results	Help-On-Hand	Defines the maximum number of results, per search, that the portlet lists.

Setting	Applicable Portlet	Description
Indexing Root Directory	- Document Search - Help-On-Hand	Required for a Jacada WorkSpace project should either its <i>Document Search</i> portlet, its <i>Help On Hand</i> portlet or both these portlets use the internal document sources, that is those defined for access by the Jacada WorkSpace search engine, in order to provide their document search results. Setting defines the path to the root directory where Jacada WorkSpace saves its document indexing results. In a clustered application server environment using multiple host machines, the defined path must be accessible by all of the cluster's host machines, for example, located in a shared network drive. In a single application server environment, if this setting is not defined, the default indexing root directory used is the <code>indexes</code> directory, which located under the directory where the application server is defined. In the indexing root directory, an index directory is created for each project and its document languages of the form <code>index_<Java_locale_code>_<project_name></code>. For example, <code>index_de_SolarEnergy</code> or <code>index_es_PropertyInsurance</code>.

Setting	Applicable Portlet	Description
Results Page Size	Document Search	Defines the maximum number of results per page, in a document source results pane, that the <i>Document Search</i> portlet displays.
Results Mode	Document Search	Mode used by the <i>Document Search</i> portlet to display its search results. Either combine results from all document sets into one results pane or separate results into multiple result panes, based on their document source. Combining results produces only one page of results in the results pane.

To ensure that results from search portlets are displayed in the correct destination, for example, in a new browser window, the PCs where Jacada WorkSpace runs must have the *Windows Folder Option File Type* setting enabled with the *Browse in same window* option.

To enable the File Type setting option for each applicable document file type:

1. In the Jacada WorkSpace host machine, right-click the *Start* menu.
2. In the menu, select the *Explore* option.
3. In the displayed *Start Menu* window, select *Tools > Folder Options... > File Types* tab.
4. In the *File Types* tab, select from the *Registered file types* pane a file type for those source documents that are available to be searched by Jacada WorkSpace. For example, *Microsoft Office Word Document*.
5. In the *Details for <selected_file_type> extension* pane, click the *Advanced* button.
6. In the displayed *Edit File Type* dialog, check the *Browse in same window* option.

Note: This option is not available for every document file type. For example, the *Adobe Acrobat Document* file type. Such a file type does not require any option enabled.

7. Click *OK*.
8. Repeat Step through Step , per applicable document file type of the documents available for search by both the *Document Search* and the *Help On Hand* portlets.
9. Click *Close*.

Defining Search Directories

This administration activity is required for a Jacada WorkSpace project should either the *Document Search* portlet, the *Help On Hand* portlet or both portlets use the internal document sources, that is those defined for access by the Jacada WorkSpace search engine, to provide their document search results.

Define the Jacada WorkSpace document sources of directories to be searched by the portlet. Documents written in different languages can be searched however, Jacada's guideline for organizing your document search directories is the following: a directory should only contain documents written in the same language. Organizing directories by document language allows for advanced document indexing, based on the language used in the document, which results in sophisticated document searching within the directory.

Administer the document search directories via the *Configuration > Document Search > Directories* option in the Jacada Management Console navigation menu. The *Directories* page makes the following activities available:

Note: For document indexing purposes, directories to be searched cannot contain files whose size exceeds 100 megabytes.

- Add a directory – Use the dropdown menu to select the project whose Document Search portlet uses the directory being added. Selecting the asterisk (*) makes the directory available to all projects for document search.

In a clustered application server environment using multiple host machines, the defined document directory path must be accessible by all of the cluster's host machines, for example, located in a shared network drive.

Use the dropdown menu to select the Java locale code, that is the language, of the documents contained in the directory.

- Delete a directory.
- Edit a directory – Update the document directory, its associated project and indexing language.

Scheduling Document Indexing

This administration activity is required for a Jacada WorkSpace project should either the *Document Search* portlet, the *Help On Hand* portlet or both portlets use the internal document sources, that is those defined for access by the Jacada WorkSpace search engine, to provide their document search results.

In order for documents to be searched, they must undergo a process called indexing. In order for Jacada WorkSpace to index a project's search documents, the project's document indexing schedule must be defined. The indexing performed is for those documents defined in the Jacada WorkSpace document sources.

Schedule document indexing for a project via the *Configuration > Document Search > Indexing* option in the Jacada Management Console navigation menu. The *Indexing* page presents a table listing the projects deployed in the application server and makes the following activities available:

- Edit a project's indexing schedule – update the daily start time of document indexing for the project and either activate or suspend this schedule.

Note: The maximum size of any file to be indexed cannot exceed 100 megabytes.

- Refresh the indexing status information of all projects in the page's table.
- Launch indexing – manually launch indexing for selected projects or all projects listed in the table, using the Index button. Indicate all projects either by all checkboxes cleared or by all selected.

Note: Due to indexing's heavy processing, Jacada recommends that indexing should only be performed during periods of low system utilization.

In the table, the *Indexing Status* field reflects any one of the following states:

- Indexing not yet performed.
- Indexing in progress.
- Indexing completed successfully.
- Indexing performed and failed.

The Smart Pad

The Smart Pad enables agents to store and view temporary information that is required during a call, such as information that is received from the caller or extracted from various applications. The application can be an application that is nested in Jacada WorkSpace or an application on the agent's desktop. Agent and supervisor use Smart Pad to perform the following actions:

- Paste values from the Smart Pad into an application.
- Copy selected text from an application and store it in the Smart Pad.

Thus, the Smart Pad is a central repository for temporary call information, replacing the need to write such information on a piece of paper or in other applications such as Notepad.

Smart Pad also enables designating entries for transfer and then transferring those entries to the Smart Pad of another agent or supervisor, as part of performing certain types of call transfers to those users.

Defining Smart Pad Settings

Manage *Smart Pad* settings via the *Configuration > Smart Pad Settings* option in the Jacada Management Console navigation menu. In the *Smart Pad Settings* page, modify the following portlet display settings as necessary:

Smart Pad Settings

Setting	Description
Empty Entry Position	Required. Defines whether Smart Pad maintains an empty row or not and the position of the empty row in the portlet. The field's default value is <code>Bottom</code>.

Setting	Description
Hide Columns	Required. Defines whether the user has the ability to hide the Smart Pad columns from display or not. The field's default value is <code>True</code>.
Maximum Entries	Required. Limits the number of entries that can be added by the agent, as follows: • If the number of existing entries is $\Rightarrow$ value, the agent cannot add entries. • If the number of existing entries is $<$ value, the agent can add up to value. Valid value: A number between 0 - 250.
Sort Columns	Defines whether the user has the ability to sort the Smart Pad column displays or not. The field's default value is <code>True</code>.

Modifications made to setting values take effect in the *Smart Pad* portlet upon the user's next log in to Jacada WorkSpace.

The Agent Disposition Portlet

The *Agent Disposition* portlet enables the agent to record the following information:

Notes – a *Notes* pane is always available for the agent to record information and submit the information to Jacada WorkSpace for project-specific processing.

Disposition Values – one or more fields are optionally available for the agent to record disposition values, while in the call wrap-up state, and submit this disposition information to Jacada WorkSpace for project-specific processing.

By default, the *Agent Disposition* portlet is available from the toolbar using the  button.

Administration Activities

Administration activities associated with the *Agent Disposition* portlet are performed in the Jacada Management Console by accessing its navigation menu option *Configuration > Agent Disposition*. The following sections describe each of these administration activities:

- [Defining Disposition Settings](#)
- [Defining Disposition Codes](#)

Defining Disposition Settings

To enable the *Agent Disposition* portlet to provide disposition values to the agent, define the following settings using the Jacada Management Console by accessing the *Configuration > Agent Disposition > Settings* page:

Settings

Setting	Value	Description
Disposition Codes Enabled	Valid values are <code>true</code> and <code>false</code> . The default value is <code>true</code> .	Indicates whether or not disposition codes are available for agent use in the <i>Agent Disposition</i> portlet. When the setting value is: • <code>True</code> – disposition codes are available; the portlet is only available for use by the agent during call handling. The agent can only submit recorded notes and disposition information to Jacada WorkSpace while in the call wrap-up state. In the portlet's <i>Disposition Values</i> area, disposition value fields are provided. Each field is a dropdown list containing valid

Setting	Value	Description
		disposition codes for selection by the agent. • <code>False</code> – the <i>Notes</i> pane is always available for use by the agent in the portlet. The <i>Disposition Values</i> area does not appear in the portlet.
Number of Disposition Value Fields	Valid integer values starting at 1. The default value is 3.	Identifies the number of fields in the <i>Disposition Values</i> area that the <i>Agent Disposition</i> portlet must provide.

Setting	Value	Description
Code Hierarchy	Valid values are <code>true</code> and <code>false</code> . The default value is <code>false</code> .	Indicates whether the disposition code fields share a hierarchical relationship or are discrete, unrelated fields. When the setting value is: • <code>True</code> – the portlet displays the provided disposition value fields in a hierarchical list in the <i>Disposition Values</i> area. The top tier field's dropdown list contains all disposition values that do not have a defined parent code. The second tier field's dropdown list contains all the disposition values with a defined parent code matching the selected value in the top tier field. All following tier field's dropdown lists are similarly related to their antecedent tier. • <code>False</code> – the portlet displays the provided disposition value fields in a left-justified, top-to-bottom list. Each field's dropdown list contains all defined disposition values.
Only Display Parent Codes	Valid values are <code>true</code> and <code>false</code> . The default value is <code>true</code> .	Indicates to only display, in the Agent Disposition portlet, those disposition codes that are defined as parent codes. Relevant when your disposition codes are hierarchically related (Code Hierarchy = <code>true</code>).

Change any of these settings as necessary. Changes made to these settings become effective in an agent's *Agent Disposition* portlet, according to the following conditions:

- If agents access this portlet from the toolbar in Jacada WorkSpace and
 - The portlet's toolbar button attribute `reloadContentOnNextOpen` is set to `true`, then the next time the portlet is opened during the current Jacada WorkSpace session, the portlet's display reflects any changes to these settings.
 - The portlet's toolbar button attribute `reloadContentOnNextOpen` is set to `false`, then the next time an agent logs in to a Jacada WorkSpace session, the portlet's display reflects any changes to these settings.

Note: Verify the portlet's `reloadContentOnNextOpen` setting in the project file `<project_name>/WebContent/USER/resources/toolbar.xml`.

- If agents access this portlet from the *Edit* menu > *Layout* option in Jacada WorkSpace, then the next time an agent logs in to a Jacada WorkSpace session, the portlet's display reflects any changes to these settings.

Defining Disposition Codes

When `Disposition Codes Enabled` is set to `true` (see Defining Disposition Settings), the *Agent Disposition* portlet requires disposition values to display in the fields in the *Disposition Values* area. Define disposition values for portlet display using the Jacada Management Console by accessing the *Configuration > Agent Disposition > Disposition Codes* page.

The *Disposition Codes* page enables you to:

- Add new disposition codes.
- Edit existing codes.
- Delete existing codes.

For details about the *Disposition Codes* page of the Jacada Management Console, see [Disposition Codes](#).

Note: Language localization is supported for disposition code entries by simply providing the text, in the required languages, in the relevant field of the *Disposition Codes* page. For example, if a disposition code's *Display Value* requires both an English and a Spanish version of the text **Invalid Billing Address**, in the entry's *Display Value* field, type the following text: Invalid Billing Address / Dirección de facturación no válido.

Change disposition codes, as necessary. Changes made to disposition codes are reflected in the *Agent Disposition* portlet's disposition value fields, according to the following conditions:

- If agents access the portlet from the toolbar in Jacada Workspace and:
 - The portlet's toolbar button attribute `reloadContentOnNextOpen` is set to `true`, then the next time the portlet is opened during the current Jacada Workspace session, the disposition value fields reflect any disposition code changes.
 - The portlet's toolbar button attribute `reloadContentOnNextOpen` is set to `false`, then the next time an agent logs in to a Jacada Workspace session, the disposition value fields reflect any disposition code changes.

Note: Verify the portlet's `reloadContentOnNextOpen` setting in the project file `<project_name>/WebContent/USER/resources/toolbar.xml`.

- If agents access this portlet from the *Edit* menu > *Layout* option in Jacada WorkSpace, then the next time an agent logs in to a Jacada WorkSpace session, the disposition value fields reflect any disposition code changes.

Logging Capabilities

Jacada WorkSpace provides a mechanism for logging Jacada WorkSpace session information. The following topics describe:

- [Producing Log Files](#)
- [Jacada WorkSpace Log Configuration File](#)
- [Log Configuration File Settings](#)
- [Supported Appenders](#)
- [Session-Related Logging](#)
- [WebLogic Server Logging](#)

Producing Log Files

Jacada WorkSpace uses the Apache Logging Services APIs to produce log files. The main components of the Apache logging APIs are *loggers*, *appenders*, and *layouts*.

Loggers are the logic components that describe the target for log messages. Appenders are the components that define where and how to write the message, for example, to a console or to a log file. Layouts define the formats of the loggers output.

A single logger can have several appenders. An important feature of loggers is that they have a hierarchical structure. This means that they inherit a property from the first ancestor for which this property is defined.

Loggers, appenders, and layouts are specified in XML log configuration files. By changing the values of their parameters, you can customize the logging to create different log files.

Jacada WorkSpace uses log4j as the logging infrastructure. For more information on the log4j APIs, follow the links to the individual projects in the Apache Logging Services Project home page at: <http://logging.apache.org/>

In addition to the logging related to the project deployed in the application server, Jacada WorkSpace provides the following:

- A session dump log file. A session dump is generated when a Jacada WorkSpace session terminates due to an unexpected error. The session dump contains session-specific information to help understand the cause of the error.
- A nested application log file. Nested applications – any Windows-based or Web-based application that is incorporated into a Jacada WorkSpace tab – write to their own, common log file. The logging performed by nested applications occurs in and resides on the agents' desktop.

Jacada WorkSpace Log Configuration File

A log configuration file consists of `<logger>` and `<appender>` sections, that define the properties of each logger and each appender. Each logger is associated with one or more appenders. An appender includes where to which to write the messages.

The log configuration can be accessed and modified through the Jacada Management Console. For details about log configuration access using the management console, refer to [Logging](#). The log configuration is saved in the following location:

- **WebSphere:** `<WebSphereHome>\profiles\<profile>\JacadaLogConfiguration.xml`.
- **WebLogic:** `<OracleWebLogicHome>\user_projects\domains\<domain_name>\JacadaLogConfiguration.xml`
- **Tomcat:** `<TomcatHome>\jacada\conf\JacadaLogConfiguration.xml`

The default Jacada WorkSpace log configuration is as follows:

Logger	Appender	Log File
Ail^a	AilAppender	logs\Ail.log
Jacada.Admin	FusionFile	logs\Fusion.log
Jacada.Debug	FusionFile	logs\Fusion.log
Jacada.Admin.JFAP.ADMIN	JFAPSingleFileAdmin	logs\JFAP-Admin.log
Jacada.Debug.JFAP.CTI	JFAPCTIFile	logs\CTI.log

Logger	Appender	Log File
Jacada.Debug.JFAP.CTIGateway	JFAPCTIGatewayFile	logs\CTIGateway.log
Jacada.Debug.JFAP.SCRIPT	JFAPSingleFileScript	logs\JFAP-Script.log

a. For use with Genesys CTI.

Note: The `JacadaLogConfiguration.xml` contains loggers, for 3rd party software, such as Beehive, Hibernate, and Spring, and `Jacada.Debug.JFAP.DEV_INTERNAL`. These loggers are for internal use only. Do not modify these loggers.

Logger Names and Appender Additivity

The logger names implement a hierarchical structure by which a logger is said to be an ancestor of another logger if its name followed by a dot is a prefix of the descendant logger name. A logger is said to be a parent of a child logger if there are no ancestors between itself and the descendant logger.

For example, `Jacada.Debug` is a parent of `Jacada.Debug.JFAP`, which in turn is a parent of `Jacada.Debug.JFAP.SCRIPT`.

Log4j implements an “appender additivity” feature, by which appender accumulation is additive. This means that when the additivity flag is set to true, each logging request for a given logger is forwarded to all the appenders in that logger as well as the appenders of the parent logger in the hierarchy.

For example, in Jacada Workspace all the `Jacada.Debug.JFAP.<NAME>` loggers are set to use additivity, which means that in addition to writing to the log files defined by their own appenders, they all also write to the `JFAP-Debug.log` which is defined by the appender of their parent logger `Jacada.Debug.JFAP`. In a similar manner, all the Jacada loggers write to the `Fusion.log`.

Note: If an ancestor of logger E, for example C, has the additivity flag set to false, then E's output will be directed to all the appenders in E and it's ancestors up to and including C but not the appenders in any of the ancestors of C.

Log Files

The log files are used as follows:

File Name	Contains
Fusion.log	All log messages.
JFAP-Debug.log	Debug-specific information.
JFAP-Admin.log	Application information.
JFAP-Script.log	Script-specific information.
CTI.log	CTI information.
CTIGateway.log	CTI Gateway information.

Log Configuration File Settings

The following sections describe in detail the settings in a logger section and in an appender section in the log configuration file:

- [Logger Section](#)
- [Appender Section](#)

Logger Section

A <logger> section contains the following settings:

Setting Name	Description
name	The name of the logger. The logger name has a hierarchical structure, which together with the <code>additivity</code> setting, determines which appenders are used and which log files the logger writes to. For more information, see Logger Names and Appender Additivity .
additivity	A boolean parameter which specifies whether the logger output is sent to the appenders of the parent logger in addition to its appenders. When set to <code>false</code>, the logger outputs to its appenders only. When the parameter is not specified it is implied to be <code>true</code>.
level	The debug level of the logger. See Debug Levels, for details.
appender-ref	The name of an appender to which the logger sends its output. Since this is the setting that links the logger to the appender, an appender with this name must be specified in the file.

Debug Levels

The different debug levels determine what information is written to the logs. You can set the debug level of the file to one of the levels listed in the table below.

The **ALL** level allows output of all log messages. At the other end of the scale, the **OFF** level turns off all output. The levels between **ALL** and **OFF** place increasingly growing limitations on the output of messages. When going from one level to the next, less messages are output and they are a result of error situations of increasing severity.

Level	Description
ALL	Turns on all logging.
TRACE	Logs all the server HTTP requests.
DEBUG	Logs informational events that are most useful to debug an application.
INFO	Logs informational messages that highlight the progress of the application.
WARN	Logs potentially harmful situations.
ERROR	Logs error events that might still allow the application to continue running.
FATAL	Logs severe error events that lead the application to abort.
OFF	Turns off logging.

Example

The following is an example of a logger definition from the `JacadaLogConfiguration.xml`:

```
<logger name="Jacada.Admin.JFAP.ADMIN" additivity="true">
  <level value="INFO"/>
  <appender-ref ref="JFAPSingleFileAdmin" />
</logger>
```

TRACE Debug Level

When using the `TRACE` level, you must add the `com.jacada.logging.TraceLevel` class attribute in the logger definition. For example:

```
<logger name="Jacada.Debug.JFAP.ADMIN" additivity="true">
  <level value="TRACE"
class="com.jacada.logging.TraceLevel"/>
  <appender-ref ref="JFAPSingleFileAdmin" />
</logger>
```

For all other debug levels, the class attribute is not specified.

Appender Section

Appenders are the output destinations of loggers.

All `<appender>` sections contain the following basic settings:

Setting	Description
name	The name of the appender.
class	The Java class that imports the appender. In <code>JacadaLogConfiguration.xml</code> , the full class name is specified (for example, <code>org.apache.log4j.FileAppender</code>).
layout	A section that defines the log's format.

For a list of the appenders supported by Jacada WorkSpace, see [Supported Appenders](#).

Layout Section

Each appender contains a `<layout>` section, that defines the output format of the appender.

Jacada WorkSpace uses different layouts defined by the following classes:

Class Name	Description
PatternLayout	Formats a logging event and returns the result as a string.

Class Name	Description
TTCCLayout	Format consists of time, thread, category and nested diagnostic context information.

Note: In `JacadaLogConfiguration.xml`, the layout is specified by the full class name (for example, `<class="org.apache.log4j.PatternLayout">`).

PatternLayout Properties

The `PatternLayout` returns a logging event as a string. The string's format is specified in the `ConversionPattern` parameter. This parameter is composed of literal text and format control expressions (conversion specifiers).

Each conversion specifier starts with a percent sign (%) and is followed by optional format modifiers and a conversion character. The conversion character specifies the type of data, such as category, priority, date, thread name, etc. The format modifiers control such things as field width, padding, left and right justification.

The following conversion characters are used in the log configuration file:

Conversion Character	Effect
c	Used to output the category of the logging event. The category conversion specifier can be optionally followed by a precision specifier, that is a decimal constant in brackets. If a precision specifier is given, only the corresponding number of right most components of the category name will be printed. By default the category name is printed in full. For example, for the category name "a.b.c" the pattern <code>%c{2}</code> will output "b.c"

Conversion Character	Effect
d	Used to output the date of the logging event. The date conversion specifier may be followed by a date format specifier enclosed between braces. For example, <code>%d{HH:mm:ss,SSS}</code> or <code>%d{dd MMM yyyy HH:mm:ss,SSS}</code>. If no date format specifier is given then ISO8601 format is assumed. For best results it is recommended to use the log4j date formatters. These can be specified using one of the strings “ABSOLUTE”, “DATE” and “ISO8601” for specifying AbsoluteTimeDateFormat, DateTimeDateFormat and ISO8601DateFormat. For example, <code>%d{ISO8601}</code> or <code>%d{ABSOLUTE}</code>.
m	Used to output the application supplied message associated with the logging event.
n	Outputs the platform dependent line separator character or characters.
p	Used to output the priority of the logging event.
t	Used to output the name of the thread that generated the logging event.

For more information, see <http://logging.apache.org/>.

Supported Appenders

Jacada WorkSpace supports the following appenders:

- **ConsoleAppender:** Causes log events to be written to the console.
- **FileAppender:** Causes log events to be written to a file.
- RollingFileAppender
- DailyRollingFileAppender
- ExternallyRolledFileAppender
- Cluster Appenders

RollingFileAppender

The **RollingFileAppender** extends `FileAppender` to backup the log file when it reaches a certain size and create a new log file. The following settings determine how big the log file is before it is rolled over and how many backups are kept:

MaxFileSize – Specifies the maximum file size of the log file. When the value is reached, the file is backed up and a new log file is created. The default value is 10 Mb.

MaxBackupIndex – Specifies how many backups of the log file are kept. There is one backup file by default.

DailyRollingFileAppender

The **DailyRollingFileAppender** extends `FileAppender`. It backs up the log file at a specified frequency, as opposed to the `RollingFileAppender` which depends on the file size. The rolling schedule is specified by a **DatePattern** parameter. This pattern follows simple date format conventions such as yyyy-MM-dd or dd-MM-yy. The date pattern is used as the suffix for the rolled file name.

For example, if the **file** parameter is set to `/foo/bar.log` and the **DatePattern** set to `'yyyy-MM-dd'`, on 2001-02-16 at midnight, the logging file `/foo/bar.log` will be copied to `/foo/bar.log.2001-02-16` and logging for 2001-02-17 will continue in `/foo/bar.log` until it rolls over the next day. Note the period separator is specified in quotes.

ExternallyRolledFileAppender

The **ExternallyRolledFileAppender** extends `RollingFileAppender`.

This appender listens on a socket on the port specified by the `port` property for a “RollOver” message. When such a message is received, the underlying log file is rolled over and an acknowledgment message is sent back to the process initiating the roll over.

Cluster Appenders

In a clustered environment all the managed servers on a given machine use the same `JacadaLogConfiguration.xml` file. Therefore all the servers try to write to the same log files. In order to avoid this situation, Jacada provides an appender that enables each managed server in a cluster environment to write its own log by creating a separate log directory for each managed server.

Each appender extends a different log4j file appender. The name of the cluster appender is identical to the name of the log4j appender with the prefix `cluster` added. For example, the `ClusterDailyRollingFileAppender` extends the log4j `DailyRollingFileAppender`.

These appenders add the name of the server to the path of the log file in the `file` parameter as the last directory name, so that every server writes its logs to a different directory.

Session-Related Logging

In addition to the logging related to the deployed project, Jacada WorkSpace provides the following, Jacada WorkSpace session-related logging:

- [The Session Dump](#)
- [Nested Application Logging](#)

The Session Dump

When a Jacada WorkSpace session terminates due to an unexpected error, session-specific information is written to a log file, in order to help understand the reason for the error. The log file, referred to as a session dump, contains environment information, session information and lists session resources too.

Each session dump is written to a separate log file in the following location:

- **WebSphere:** <WebSphereHome>\profiles\<profile>\logs
- **WebLogic:** <OracleWebLogicHome>\user_projects\domains\<domain_name>\logs
- **Tomcat:** <TomcatHome\jacada\logs>

The name of the log begins with the 'ws_' prefix followed by the WorkSpace session ID.

The generated file is listed in an index file, `index.html`, in the same directory. This file contains pointers to all the session dump log files generated on the same server.

The following setting in the `GLOBAL_SETTINGS` table is used to determine whether to generate the session dump log files: `IsSessionDumpEnabled`. By default, the value of the setting is `true`, meaning that session dump log files are generated.

The session dump consists of the following segments:

- General environment information
- General session information
- Accumulated session information

General Environment Information

Information	Description
Server information	The IP address and operating system details of the application server, and the server memory consumption.
Jacada WorkSpace information	The version of Jacada WorkSpace, and the name of the running project.
Java version	The JDK version and additional details.

General Session Information

Information	Description
Session ID	The WorkSpace session ID and project ID.

Information	Description
Agent ID	The user name of the agent who was logged in to the terminated WorkSpace session and related details.
Login time	The time when the agent logged in to Jacada WorkSpace.
Session termination time	The time when the WorkSpace session terminated.
CTI agent details	The CTI agent name, extension number and other CTI details. If CTI is not in use (working in manual mode), a message indicating this will be shown instead.

Accumulated Session Information

This segment includes information accumulated during the session lifetime, to help understand the cause of the session dump:

Information	Description
Dump title	A short description of the cause for the dump, such as “An unexpected exception occurred”.
Dump description	A more detailed description of the dump.
Dump date	The date and time when the dump started.
Log messages	A list of recent log messages for the WorkSpace session, including messages not written to the log because of the log level.
Last exceptions thrown	A list of recent exceptions, which were thrown by the WorkSpace session.
Last accessed pages	A list of recent <code>URLs</code> , of pages which were visited by the WorkSpace session.

Information	Description
Fatal exceptions	A list of the exceptions that caused the error condition which generated the dump.
Last push messages sent	A list of recent push messages, with keyword and parameters, which were sent by the WorkSpace session.

Nested Application Logging

In each agent desktop that has installed the `JacadaApplicationHolder` ActiveX control, nested applications, running in Jacada WorkSpace tabs, will perform logging.

During a Jacada WorkSpace session, each nested application that is running in a Jacada WorkSpace tab writes entries to the following log file, which is located in the agent desktop:
`${ALLUSERSPROFILE}\Jacada\JAD\LogFiles\Nesting.log`. Nested applications write to this file via the `JacadaApplicationHolder` ActiveX control.

Upon an agent logging in to a Jacada WorkSpace session and the start of the ActiveX control, the Jacada WorkSpace project issues the current application server time and the WorkSpace session ID, which the ActiveX control then records in the `Nesting.log` file.

Each entry provides the following information about the nested application:

Local time in the agent desktop – for example, 26 Nov 2008 19:06:44,723.

Information debug level – for example, INFO.

Source file and line number – for example, jadnesting.cpp: 343.

Thread and thread ID – for example, 5,520.

Time in the application server – for example, 2008/11/26-18:05:30.

Recorded log message – for example, Effective desktop Rect {205,354,662,451}.

The following is an example of an entry recorded in the `Nesting.log`:

```
26 Nov 2008 19:06:44,723 [INFO] [jadnesting.cpp: 343] [5,520] [2008/11/26-18:05:30] Effective desktop Rect {205,354,662,451}
```

Configuration of Nested Application Logging

The `NestingLogConfiguration.xml` file contains the configuration definitions necessary for the logging performed by nested applications in Jacada WorkSpace. `NestingLogConfiguration.xml` exists in each agent desktop that has installed the `JacadaApplicationHolder` ActiveX control, which supports the nesting of applications in Jacada WorkSpace tabs. By default, this configuration file resides in `<Jacada_install_folder>\JacadaApplicationHolder\Configuration\Logs`.

`NestingLogConfiguration.xml` establishes the use of the `TimeRollingFile` appender with the `DailyRollingFileAppender` class for the `Nesting.log` file. Nested application back-up log files are located in the same folder as the `Nesting.log` file and are uniquely identified by the naming convention of `Nesting.log.yyyy-MM-dd`. The default Debug level of information written to the log file is `ERROR`.

Update any of the provided logging configurations directly in the `NestingLogConfiguration.xml` file; the Jacada Management Console does not currently support updating nested application logging configuration. In order that updates made in the configuration file take effect for the logging performed by a nested application, it is necessary to close and then re-open the nested application in the tab.

WebLogic Server Logging

Each server configured in a WebLogic Server domain maintains several logs to archive messages received from its subsystems. Subsystems within a WebLogic Server generate log messages to communicate their status. In addition to the several server-specific logs, such as the *Events Log*, the *Metrics Log*, the *Server Log*, the *Web Server Log*, and the *Web Application Log*, the Administration Server maintains a domain-wide log, *DomainLog*, that summarizes messages from the individual server logs.

Each server log, `<server_name>.log`, records information about events such as the startup and shutdown of servers, the deployment of new applications, or the failure of one or more subsystems. For example, when you start a server instance, the Security subsystem writes a message to report its initialization status.

Also, each server instance, based on its default logging configuration settings, logs certain types of transaction processing. An example of WebLogic Server default logging is HTTP logging; in a separate log file, the server saves HTTP transactions processed by the HTTP subsystem. Lastly, each server instance maintains a Standardout log file and a Standarderror log file.

For information about viewing and configuring WebLogic Server logs, see the *Oracle WebLogic Server Administration Console Help*.

The following procedures administer the required logging for the servers configured in a WebLogic Server domain, either a single server instance or a cluster of Managed Servers. Performing these procedures requires administrator level user privileges.

To update logging levels and disable HTTP request logging for WebLogic servers:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Environment > Servers*.
The *Summary of Servers* page displays.
2. In the *Servers* table, click the relevant `<server_name>` link.
The *Configuration – General* tab of the `<server_name>` displays.
3. Choose the *Logging* tab.
4. In the tab, choose the *General* tab.
5. In the *General* tab, click on the *Advanced* link.
The *Advanced* options display.
6. From the dropdown list for the *Minimum severity to log* option, select the setting *Warning*.
7. Go to the *Message destination(s)* area of the *Advanced* options and do the following:
 - a. From the *Severity level* dropdown list for the *Log file* destination, select the setting *Warning*.
 - b. From the *Severity level* dropdown list for the *Standard out* destination, select the setting *Error*.
 - c. From the *Severity level* dropdown list for the *Memory buffer* destination, select the setting *Error*.
8. Click *Save*.
Server logging settings are updated. All changes are activated and no server restarts are necessary.
9. Choose the *HTTP* tab under *Logging*.
10. Clear the *HTTP access log file enabled* checkbox.
Server logging of HTTP requests is disabled.
11. Click *Save*.
Server logging settings are updated. All changes are activated and no server restarts are necessary.
12. Repeat Step 1 through Step 11 for each application server listed in the *Servers* table on the

Summary of Servers page.

To enable security-related logging for a WebLogic server:

1. In the *Domain Structure* navigation pane of the WebLogic Server Administration Console, select *Environment > Servers*.

The *Summary of Servers* page displays.

2. In the *Servers* table, click the relevant `<server_name>` link.

The *Configuration – General* tab of the `<server_name>` displays.

3. Choose the *Debug* tab.

In the tab, the *Debug settings for this Server* table displays.

4. Expand the *weblogic* entry.

All the debug scopes for the *weblogic* entry display.

5. Expand the *security* debug scope entry.

All the debug attribute categories of the *security* debug scope display.

6. Expand the *adjudicator* debug attribute category.

The *DebugSecurityAdjudicator* attribute displays.

7. Expand the *atn* debug attribute category.

The *DebugSecurityAtn* attribute displays.

8. Expand the *atz* debug attribute category.

The *DebugSecurityAtz* attribute displays.

9. Expand the *realm* debug attribute category.

The *DebugSecurityRealm* attribute displays.

10. Expand the *rolemap* debug attribute category.

The *DebugSecurityRoleMap* attribute displays.

11. In the *State* column, verify the current state of these attributes.

12. If the state of any of these attributes is *Disabled*, in the *Change Center* pane, click *Lock & Edit*.

The *Debug settings for this Server* table re-displays with its state modification buttons enabled.

13. For each of these attributes whose state is *Disabled*, check the attribute.

14. Click *Enable*.

The following message displays: *Please click Activate Changes button to activate your changes*.

15. In the *Change Center* pane, click *Activate Changes*.

The following message displays: *All Changes have been activated. No restarts are necessary.*

The debug security attributes *DebugSecurityAdjudicator*, *DebugSecurityAtn*, *DebugSecurityAtz*, *DebugSecurityRealm* and *DebugSecurityRoleMap* are enabled for the WebLogic server. Security-related logging is now issued to the server's Standardout file by all installed application modules running in the server.

Event Auditing Administration

Jacada WorkSpace audits project events that are enabled for auditing by the administrator. Jacada WorkSpace provides for the auditing of project events in the following ways:

- Using the Jacada Management Console, administer the pre-defined set of project events. From this set of events, enable events to be processed by the Jacada auditing manager.
- Project events, other than the Jacada WorkSpace pre-defined set of events, are explicitly coded to be processed by the Jacada auditing manager.

When the auditing feature is enabled, then database auditing can be enabled and any custom project auditing can be invoked during runtime of the project.

Note: Auditing can impact performance and create many database entries, requiring regular checking of the database size.

Database Auditing – During a Jacada WorkSpace session, events identified for audit are recorded in the database by the DB Auditor process. For a description of the relevant tables and the event information recorded with database auditing of project events, see [Management of Audited Entities](#).

Custom Project Auditing – During project development, you can override any of the supplied auditing mechanisms, for example, override database auditing of project events by extending the `DBAuditor`. Also, you can develop new auditors that provide custom auditing treatment of events.

For a detailed description about developing custom project auditing, refer to the Jacada WorkSpace Developer Guide.

The following topics describe:

- [Understanding the Audited Entities of a Project](#)
- [Set of Events for Audit](#)
- [Event Auditing Administration](#)
- [Aggregating Data for Reporting](#)

Understanding the Audited Entities of a Project

When database auditing is enabled, Jacada WorkSpace records in the database information for audited entities. In order for an information system to retrieve the necessary audited project data, understand the relevant schema as defined in the following database tables:

- [AUDIT_AGENT_SESSIONS Table](#)
- [AUDIT_INTERACTIONS Table](#)
- [AUDIT_PROCESSES Table](#)
- [AUDIT_PROCESS_PARAMS Table](#)
- [AUDIT_ACTIONS Table](#)
- [AUDIT_ACTION_PARAMS Table](#)

The *Agent Session*, *Interaction*, *Process* and *Action* represent the different types of Jacada WorkSpace entities that can occur. Both *Process* and *Action* events can have associated event parameters.

AUDIT_AGENT_SESSIONS Table

The AUDIT_AGENT_SESSIONS table contains entries of the agents' Jacada WorkSpace *Session* events. Such an event represents a complete agent session from login through logout. Each AUDIT_AGENT_SESSIONS table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this agent session event.

Column	Description
AGENT ID	The user name used, by the agent, to log in to a Jacada WorkSpace session.
PROJECT ID	The name of the deployed Jacada WorkSpace project.
SESSION ID	The Jacada WorkSpace session ID.

AUDIT_INTERACTIONS Table

The `AUDIT_INTERACTIONS` table contains entries of the agents' *Interaction* events, being audited. Such an event represents the interaction between an agent and a client, for example, during a telephone conversation, a fax correspondence or an e-mail correspondence. An agent interaction event occurs within the context of an agent session event. Each `AUDIT_INTERACTIONS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this agent interaction event.
INTERACTION TYPE	The type assigned to this agent interaction event. Audited interaction types are: <i>Inbound</i> , <i>Outbound</i> , <i>Manual</i> , <i>Internal</i> and <i>Custom Item</i> .
INTERACTION NUMBER	For interaction events of types <i>Inbound</i> , <i>Outbound</i> , <i>Manual</i> and <i>Internal</i> , this number is the telephone number of the call. For interaction events of type <i>Custom Item</i> , this number is the Genesys Interaction ID.
START TIME	The date and time when the interaction event started, in the format YYYY-MM-DD HH:MM:SS.
END TIME	The date and time when the interaction event ended, in the format YYYY-MM-DD HH:MM:SS.

Column	Description
AGENT_SESSION_ID	The ID assigned to the agent session event, which is the parent event of this interaction event.

AUDIT_PROCESSES Table

The `AUDIT_PROCESSES` table contains entries of the agents' *Process* events, which are enabled for audit. Such an event represents a Jacada WorkSpace business process initiated by the agent that occurs within the context of either an agent session event or an agent interaction event. Each `AUDIT_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS_ID	The ID assigned to this agent process event.
PROCESS_NAME	The name given to this agent process event. Process names are unique within a Jacada WorkSpace project.
START_TIME	The date and time when the process event started, in the format YYYY-MM-DD HH:MM:SS.
END_TIME	The date and time when the process event ended, in the format YYYY-MM-DD HH:MM:SS.

AUDIT_PROCESS_PARAMS Table

The `AUDIT_PROCESS_PARAMS` table contains entries of process parameters for those *Process* events enabled for audit. A process parameter provides operational information that is integral to the *Process* event to which it is associated. For example, `Chat`, a Jacada WorkSpace-provided event available for audit, is a *Process* event and has the following process parameters: `Initiator` and `Recipient`. Each table entry stores audit information about a single, *Process* event parameter. Each `AUDIT_PROCESS_PARAMS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this process parameter.
NAME	The name given to this process parameter. For example, <code>Initiator</code> . Process parameter names are unique within a Jacada WorkSpace project.
VALUE	The information content of the process parameter.
PROCESS ID	The ID assigned to the agent process event, which is the parent event of this process parameter.

AUDIT_ACTIONS Table

The `AUDIT_ACTIONS` table contains entries of the agents' *Action* events, which are enabled for audit. Such an event represents an operation performed by the agent that occurs within the context of either an agent session event, an agent interaction event or an agent process event. Each `AUDIT_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to this agent action event.
ACTION NAME	The name given to this agent action event. Action names are unique within a Jacada WorkSpace project.
ACTION TIME	The date and time when the action event occurred, in the format YYYY-MM-DD HH:MM:SS.

AUDIT_ACTION_PARAMS Table

The `AUDIT_ACTION_PARAMS` table contains entries of action parameters for those *Action* events enabled for audit. An action parameter provides operational information that is integral to the *Action* event to which it is associated. For example, `warmTransferClicked`, a Jacada WorkSpace-

provided event available for audit, is an *Action* event and has the following action parameter: `dn`. Each table entry stores audit information about a single, *Action* event parameter. Each `AUDIT_ACTION_PARAMS` table entry is comprised of the following elements:

Column	Description
ID	The ID assigned to this action parameter.
NAME	The name given to this action parameter. For example, <code>dn</code> (dialed number). Action parameter names are unique within a Jacada WorkSpace project.
VALUE	The information content of the action parameter.
ACTION ID	The ID assigned to the agent action event, which is the parent event of this action parameter.

Database Auditing uses the following tables to record, and thereby manage, the child-parent relationship that exists between audited events. A many-to-one association can exist for a parent event, that is many child events having one parent event. Each child event is uniquely associated with its parent event and never appears in any other relationship table.

- [AUDIT AGENT SESSION PROCESSES Table](#)
- [AUDIT AGENT SESSION ACTIONS Table](#)
- [AUDIT INTERACTION PROCESSES Table](#)
- [AUDIT INTERACTION ACTIONS Table](#)
- [AUDIT PROCESS ACTIONS Table](#)

AUDIT_AGENT_SESSION_PROCESSES Table

The `AUDIT_AGENT_SESSION_PROCESSES` table contains entries that identify the child-parent relationship that exists between an agent process event (the child) and the agent session event (the parent), within whose context the agent process event occurred. Each `AUDIT_AGENT_SESSION_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS ID	The ID assigned to an agent process event.
PARENT ID	The ID assigned to the agent session event, which is the parent event of this process event.

AUDIT_AGENT_SESSION_ACTIONS Table

The `AUDIT_AGENT_SESSION_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent session event (the parent), within whose context the agent action event occurred. Each `AUDIT_AGENT_SESSION_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent session event, which is the parent event of this action event.

AUDIT_INTERACTION_PROCESSES Table

The `AUDIT_INTERACTION_PROCESSES` table contains entries that identify the child-parent relationship that exists between an agent process event (the child) and the agent interaction event (the parent), within whose context the agent process event occurred. Each `AUDIT_INTERACTION_PROCESSES` table entry is comprised of the following elements:

Column	Description
PROCESS ID	The ID assigned to an agent process event.

Column	Description
PARENT ID	The ID assigned to the agent interaction event, which is the parent event of this process event.

AUDIT_INTERACTION_ACTIONS Table

The `AUDIT_INTERACTION_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent interaction event (the parent), within whose context the agent action event occurred. Each `AUDIT_INTERACTION_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent interaction event, which is the parent event of this action event.

AUDIT_PROCESS_ACTIONS Table

The `AUDIT_PROCESS_ACTIONS` table contains entries that identify the child-parent relationship that exists between an agent action event (the child) and the agent process event (the parent), within whose context the agent action event occurred. Each `AUDIT_PROCESS_ACTIONS` table entry is comprised of the following elements:

Column	Description
ACTION ID	The ID assigned to an agent action event.
PARENT ID	The ID assigned to the agent process event, which is the parent event of this action event.

Set of Events for Audit

Jacada WorkSpace supplies a set of events that can be enabled for auditing, using the Jacada Management Console. Enabled events become auditable by all registered, project auditors, for example, the `DBAuditor`, and any custom developed auditors. The following table describes the Jacada WorkSpace set events.

Jacada WorkSpace Set Events

* Auditing of custom items: When reviewing custom item event information, distinguish between the following: custom items that have both a recorded Answer Custom Item event and End Item event versus custom items that have only a recorded End Item event. The former occurrence identifies custom items that were handled by an agent, while the latter occurrence identifies items that were not answered.

Event Auditing Administration

The Jacada WorkSpace administrator manages event auditing using the Jacada Management Console. Via the Management Console, the administrator can accomplish the following:

- Enable or disable the Jacada WorkSpace auditing feature.
Enabling this feature makes available for use any of the supplied auditors and custom project auditing development.
- Enable or disable any of the supplied auditors, for example the `DBAuditor`, which provides database auditing of project events.
- Enable or disable any of the events contained in the Jacada WorkSpace-provided set of events for audit.

Auditing administration updates take effect in those Jacada WorkSpace sessions that are established following the updates.

To enable auditing:

1. In the Jacada Management Console navigation panel, select the *Configuration > Auditing > Settings* option.
The *Settings* page displays.
2. In the *AuditingEnabled* field, type *true*.

3. Click *Apply*.

The auditing feature is now enabled for Jacada WorkSpace projects, thereby making available for use any of the supplied auditors and custom project auditing development.

To enable database auditing:

1. In the Jacada Management Console navigation panel, select the *Configuration > Auditing > Settings* option.

The *Settings* page displays.

2. In the *DBAuditingEnabled* field, make sure that its value is *true* (default). If not, type *true*.
3. Click *Apply*.

Database auditing is now enabled for Jacada WorkSpace projects. The *DBauditor* will provide database auditing of all audited project events.

To enable events from the set of events to audit:

1. In the Jacada Management Console navigation panel, select the *Configuration > Auditing > Events* option.

The *Events* page displays.

2. For each listed project event that requires audit, make sure that its associated dropdown menu displays *Enabled* (default). If not, in the dropdown menu, select *Enabled*.
3. Click *Apply*.

Events marked *Enabled* are now auditable by all project auditors, during a Jacada WorkSpace session.

Aggregating Data for Reporting

Report generation is based on aggregated auditing data. Jacada WorkSpace provides two scripts for data aggregation, located in the installation package under:

<Workspace Home folder>\Workspace\Accessories\Reporting\WS-Nightly-Scripts

To enable data aggregation, after installing WorkSpace and creating a database schema, you need to run the *script_<db type>_definitions.sql* script. This script creates the required functions and stored procedures that aggregate the data. It is located under **\Workspace\Accessories\Reporting\WS-Nightly-Scripts\sql**.

The script should be executed on a nightly basis. Follow the relevant procedure below to run the

script.

To run the auditing data aggregation script (Windows):

1. After verifying that you are using Java 1.6 or above, add the Java to the PATH system variables.
2. Configure the properties in the **db.properties** file to match your database details. For `db.schema.password`, enter the plain password for your database schema. Then, ensure that the `db.password.encrypted` value is set to `false`.
3. Execute the script by executing the following command:

```
WS>wsreporting.cmd
```

By default, the script aggregates the data of the previous day. To specify another date for aggregation, enter the relevant date following the command in the following format: `dd/mm/yyyy`.
For example:

```
WS>wsreporting.cmd 21/10/2014
```
4. Create a schedule task to execute the script on a nightly basis.

To run the auditing data aggregation script (Unix/Linux):

1. After verifying that you are using Java 1.6 or above, invoke the following command:

```
export PATH=<JAVA_HOME>/bin:$PATH
```
2. Configure the properties in the **db.properties** file to match your database details. For `db.schema.password`, enter the plain password for your database schema. Then, ensure that the `db.password.encrypted` value is set to `false`.
3. Execute the script by executing the following command:

```
WS>wsreporting.sh
```

By default, the script aggregates the data of the previous day. To specify another date for aggregation, enter the relevant date following the command in the following format: `dd/mm/yyyy`.
For example:

```
WS>wsreporting.hs 21/10/2014
```
4. Create a schedule task to execute the script on a nightly basis.
For more information on Unix/Linux scheduling, go to http://kevin.vanzonneveld.net/techblog/article/schedule_tasks_on_linux_using_crontab/

Nightly Scheduling on Linux

On Linux, nightly reporting is scheduled as a cron task. The reporting script requires the full path for the script's folder. By default, this path is

<JACADA_INSTALLATION>\WorkSpace\Accessories\Reporting\WS-Nightly-Scripts.

To copy the **WS-Nightly-Scripts** folder to a different location, open **wsreporting.sh** for editing, and change all the paths to the new path.

To schedule a cron task:

1. Verify that you are running **dos2unix wsreporting.sh**.
2. To make the scripts runnable, execute the following commands:
 - `chmod 777 wsreporting.sh`
 - `chmod 777 ant/bin/ant`
3. Schedule the task using crontab --e. Make sure you set your PATH in the crontab editor and that it includes both the java home and ant folder (full path). For example:
PATH=<JACADA_INSTALLATION>/Workspace/Accessories/Reporting/ WS-Nightly-Scripts/ant/:usr/java/jre1.6.0_24/bin:usr/kerberos/sbin:usr/kerberos/bin:usr/local/sbin:usr/local/bin:sbin:bin:usr/sbin:usr/bin:root/bin

The following example represents the cron task of executing reports every day at 23:30:

```
23 30 * * * <JACADA_INSTALLATION>/Accessories/Reporting /WS-Nightly-Scripts/wsreporting.sh
```

Side-By-Side Script Interaction

Jacada WorkSpace supports the side-by-side interaction of Jacada WorkSpace call handling processing together with a Jacada Interaction Manager script. This supported integration operates as follows: During call handling, the Jacada Interaction Manager script is the initial script to be invoked and displayed; at one or more specific processing points in the flow of the Jacada Interaction Manager script, the script instructs Jacada WorkSpace as to which project Web page to display in its *Script* tab.

With side-by-side interaction mode, there are two script presentation layouts:

Vertical – The Jacada WorkSpace *Script* tab is vertically divided between a Jacada Interaction Manager script display area and a Web page display area. The Jacada Interaction Manager script displays on the left side and the Web pages display on the right side of the *Script* tab.

Horizontal – The Jacada Interaction Manager script display area is positioned horizontally above the Jacada WorkSpace *Script* tab.

The following topics describe:

- [Administration Needed for Script Integration](#)
- [Messages](#)

Administration Needed for Script Integration

In order for Jacada WorkSpace to integrate the running of a Jacada Interaction Manager script into its call handling, the administrator must perform the following actions:

- Configure a Jacada Interaction Manager web application (a .war file).
- Deploy the configured Jacada Interaction Manager web application to an application server that is certified to support Jacada WorkSpace.
- Publish a Jacada Interaction Manager script to the Jacada Interaction Manager web application.

(For a detailed description of the above actions, refer to the Jacada Installation Guide for the Developer.)

- Using the Jacada Management Console, configure the required database tables to reference the Jacada Interaction Manager script that must be run for a given project and telephone number.

Configuring the Database for Side-By-Side Script Interaction

Enable Jacada WorkSpace to execute side-by-side script interaction for call handling by configuring the Jacada WorkSpace database. This involves working with the following Jacada Management Console *Configuration* pages:

- The *RTN Mappings* page – Maps Jacada WorkSpace project entries to their associated telephone number and script. Jacada WorkSpace requires this table to be configured to support running a Jacada Interaction Manager script, either in solo mode or in side-by-side interaction mode.
- The *Script Mappings* page – Maps script entries to their associated URL. The Jacada Interaction Manager script name and the URL to its location are provided in this table. Jacada WorkSpace requires this table to be configured to support running a Jacada Interaction Manager script, either in solo mode or in side-by-side interaction mode.
- The *External Script Settings* page – Configures the settings that are specific to the side-by-side interaction mode, including if this mode is enabled and the associated layout and display in Jacada WorkSpace of a Jacada Interaction Manager script operating in this mode.

When the side-by-side layout in effect is vertical, the external script area, that is the Jacada Interaction Manager script display area, by default configuration, is collapsible. Clicking on the Web page display area in the *Script* tab of Jacada WorkSpace results in the collapse of the Jacada Interaction Manager script display and the expansion of the Web page display area. Clicking on the collapsed Jacada Interaction Manager script display area (the external script area) in the *Script* tab, expands the Jacada Interaction Manager script display and returns both it and the Web page display back to their configured display area sizes for the *Script* tab.

- The Page Mapping page – Contains the mapping of project Web pages to their relative project path. For example, `/USER/resources/jsp/myPage.jsp`. Based on processing states, Jacada WorkSpace can display one or more of the table's Web page entries. In support of the side-by-side interaction mode of a Jacada Interaction Manager script, this table must contain all the project Web pages that Jacada WorkSpace displays in the *Script* tab as instructed by the Jacada Interaction Manager script. The following default .jsp pages are provided:

Default .jsp Pages

Page ID	Description
collapsedExternalscriptDisplayPage	The page displayed by Jacada WorkSpace over the collapsed Jacada Interaction Manager script display area in the script tab. The default page can be replaced with a custom developed page.

Page ID	Description
horizontalExternalscriptDisplayPage	The page displayed in the Jacada Interaction Manager script display area, prior to a Jacada Interaction Manager script starting, when the horizontal layout is used for the display area. The default page can be replaced with a custom developed page.
missingMappingPage	The page displayed in the script tab by Jacada WorkSpace whenever Jacada WorkSpace cannot locate the Web page that the Jacada Interaction Manager script instructed it to display. The default page can be replaced with a custom developed page.
scriptStartPage	The page initially displayed in the Web page display area of the script tab by Jacada WorkSpace when the Jacada Interaction Manager script starts. The default page can be replaced with a custom developed page.

For descriptions of the Jacada Management Console user interface and how to access it and the relevant database tables, refer to [Using the Jacada Management Console](#).

To configure the RTN Mapping required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > RTN Mappings* option.
The *RTN Mappings* page displays.
2. Click *Add*.
The *RTN Mapping add entry* page displays.

3. In the *Project ID* field, type the name of a Jacada WorkSpace project.

Note: Providing the asterisk in this field, *, specifies that the current entry applies to all projects.

4. In the *RTN* field, type a routing telephone number that is associated with the provided project and that its call handling will be by side-by-side script interaction.
5. From the *Script ID* dropdown menu, select a Jacada Interaction Manager script from the list.
You are assigning the Jacada Interaction Manager script that Jacada WorkSpace will run when an agent receives a call from the provided routing telephone number.
6. Click *Save*.
The *RTN Mappings* page displays containing the added entry in the table of current RTN mapping entries. The RTN mappings are updated in the Jacada WorkSpace database with the entry you added.
7. Repeat Step 2 through Step 6, as necessary.

To modify the script of an existing RTN Mapping entry required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > RTN Mappings* option.
The *RTN Mappings* page displays the table of current RTN mapping entries.
2. In the *Action* column of the RTN mapping entry you want to modify, click the *Edit* link.
The *RTN Mapping edit entry* page displays.
3. From the *Script ID* dropdown menu, select one of the call handling scripts listed.
You are modifying the assigned call handling script that Jacada WorkSpace will run when an agent receives a call from the provided routing telephone number.
4. Click *Save*.
The *RTN Mappings* page displays containing the modified entry in the table. The RTN mappings are updated in the Jacada WorkSpace database with the entry you modified.
5. Repeat Step 2 through Step 4, as necessary.

To configure the Script Mapping required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > Script Mappings* option.

The *Script Mappings* page displays. If there are no current script mapping entries in the Jacada WorkSpace database, the following message appears in the page: *There is no data to display.*

2. Click *Add*.

The *Script Mapping add entry* page displays.

3. In the *Script ID* field, type the name of a Jacada Interaction Manager script to be run by Jacada WorkSpace.
4. In the *Context path* field, type the URL to the location of the provided Jacada Interaction Manager script.
5. Click *Save*.

The *Script Mappings* page displays containing the added entry in the table of current script mapping entries. The script mappings are updated in the Jacada WorkSpace database with the entry you added.

6. Repeat Step 2 through Step 5, as necessary.

To modify the context path of an existing Script Mapping entry required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > Script Mappings* option.

The *Script Mappings* page displays the table of current script mapping entries.

2. In the *Action* column of the script mapping entry you want to modify, click the *Edit* link.
The *Script Mapping edit entry* page displays.

3. In the *Context Path* field, modify the URL of the Jacada Interaction Manager script.
4. Click *Save*.

The *Script Mappings* page displays containing the modified entry in the table. The script mappings are updated in the Jacada WorkSpace database with the entry you modified.

5. Repeat Step 2 through Step 4, as necessary.

To modify the External Script Settings required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > External Script Settings* option.

The *External Script Settings* page displays. Jacada WorkSpace supplies default values for these settings.

2. In the *SideBySideEnabled* field, type `true`.

3. In the *SideBySideLayout* field, type the script layout you want for side-by-side interaction mode. The options are either `vertical` or `horizontal`.
4. If *SideBySideLayout* field is `vertical`, the following fields are relevant to configure:
 - a. *ExternalScriptAreaWidth* – type a percentage (whole number) between 1 and 100.
 - b. *ExternalScriptAreaCollapsible* – type `true` to make the Jacada Interaction Manager script display area collapsible or type `false` to make this display area not collapsible.
 - c. *ExternalScriptAreaCollapsedWidth* – type a percentage (whole number) between 1 and 100.
5. Click *Apply*.

The external script settings are updated in the Jacada WorkSpace database with any setting values you modified.

To configure the Page Mapping required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > Page Mappings* option.

The *Page Mappings* page displays the table of current page mapping entries. Jacada WorkSpace supplies several, default .jsp pages.
2. Click *Add*.

The *Page Mapping add entry* page displays.
3. In the *Page ID* field, type the name of a project Web page that Jacada WorkSpace will display in the *Script* tab as instructed by a Jacada Interaction Manager script.
4. In the *Context Path* field, type the relative project path to where the Web page is stored.

For example, `\USER\resources\jsp\wrongSelectionPage.jsp`.
5. Click *Save*.

The *Page Mappings* page displays containing the added entry in the table of current page mapping entries. The page mappings are updated in the Jacada WorkSpace database with the entry you added.
6. Repeat Step 2 through Step 5, as necessary.

To modify the context path of an existing Page Mapping entry required for side-by-side script interaction:

1. In the Jacada Management Console navigation panel, select the *Configuration > Page Mappings* option.

The *Page Mappings* page displays the table of current script mapping entries.
2. In the *Action* column of the page mapping entry you want to modify, click the *Edit* link.

The *Page Mapping edit entry* page displays.

3. In the *Context Path* field, modify the relative project path to where the Web page is stored.
4. Click *Save*.

The *Page Mappings* page displays containing the modified entry in the table. The page mappings are updated in the Jacada WorkSpace database with the entry you modified.

5. Repeat Step 2 through Step 4, as necessary.

Messages

When side-by-side interaction is the Jacada Interaction Manager script mode of operation, Jacada WorkSpace issues messages in the event of processing errors. Relevant messages are displayed, as follows:

- Presented to the agent in the Web page display area of the Jacada WorkSpace *Script* tab.
- Presented to the agent in the Jacada Interaction Manager script display area of the Jacada WorkSpace *Script* tab.
- Messages are generated to the relevant Jacada WorkSpace log files.

Refer to the Jacada WorkSpace Developer Guide for relevant message information.

Failover and Switchover Support (WebSphere)

Failover is the capability of a product architecture's components to automatically switch processing responsibility from the active server, database or network that just failed to a standby server, database or network. Switchover is the capability to manually switch processing responsibility from the active server, database or network to a standby server, database, or network, upon the failure of the formerly active server, database or network.

Jacada WorkSpace provides a failover mechanism that enables session replication in a clustered environment. Failover is certified using the F5 Networks BIG-IP load balancer. For more information on BIG-IP, refer to [Load Balancing the WebSphere Cluster](#).

The following sections describe how to use the failover and switchover capabilities:

- [Session Replication in a Clustered Environment](#)
- [Controlling the Impact of Infrastructure Failures](#)
- [Effecting a Switchover](#)

Note: These topics are relevant only to users working with a WebSphere application server.

Session Replication in a Clustered Environment

An application server supports failover for Web applications by replicating the session state currently contained in memory. The server instance copies its session state in memory to another, designated server instance in the cluster. If the first server fails, then the replicated session state on the secondary server is located and used to continue session processing without any loss of the session's information. To support failover, Jacada WorkSpace enables you to work with the application server's session replication.

Note: As session replication may impact performance, Jacada recommends consulting your Jacada representative before implementing session replication.

Configuring a Unique Session Cookie for the Push Session

During periodic push timeout, the push session, which exists between the JacadaWorkSpaceCommon application in the application server and Jacada WorkSpace, enters a wait state. During this wait state, the WebSphere application server suspends performing any session replication for the specific Jacada WorkSpace session, identified by its assigned cookie *JSessionId*. To enable the WebSphere application server to perform its session replication, unaffected by the push session wait state, you must establish a session id for the push session that is separate from the session id that is assigned to the deployed Jacada WorkSpace project. Accomplish this separation of sessions by configuring the JacadaWorkSpaceCommon application to use a different cookie, namely the *PushSessionId* cookie.

Note: This configuration is required as part of the post-installation actions for Jacada WorkSpace in a WebSphere Application Server. Verify that JacadaWorkSpaceCommonEAR's *Override session management* option is enabled and that its use of the *PushSessionId* cookie is configured. If not, use the provided procedure to accomplish these actions.

To configure JacadaWorkSpaceCommon to use the PushSessionId cookie:

1. In the task menu of the WebSphere Integrated Solutions Console, select *Applications > Enterprise Applications > JacadaWorkSpaceCommonEAR > Session Management*. The *Configuration* tab for the *Session Management* page displays.

Enterprise Applications > [JacadaWorkspaceCommonEAR](#) > Session management

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These settings apply to both the SIP container and the Web container.

Configuration

General Properties	Additional Properties
☐ Override session management	= Distributed environment settings
Session tracking mechanisms: ☐ Enable SSL ID tracking ☒ Enable cookies ☐ Enable URL rewriting ☐ Enable protocol switch rewriting	
☒ Allow overflow	
Maximum in-memory session count: 1000 sessions	

2. In the *General Properties* list, check the *Override session management* option.
3. Click *Apply*.
4. In the *Session tracking mechanism* area, click the *Enable cookies* link.
The *Configuration tab* for *Cookies* page displays.
5. In the *Cookie name* field, replace the current field entry, *JSessionId*, with *PushSessionId*.

Enterprise Applications > **JacadaWorkspaceCommonEAR** > **Session management**

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These settings apply to both the SIP container and the Web container.

Configuration

General Properties	Additional Properties
☐ Override session management	☒ Distributed environment settings
Session tracking mechanisms: ☐ Enable SSL ID tracking ☒ Enable cookies ☐ Enable URL rewriting ☐ Enable protocol switch rewriting	
☒ Allow overflow	
Maximum in-memory session count: 1000 sessions	

- Click *Apply*.

The JacadaWorkSpaceCommon application is configured to perform its push processing using the session assigned by the *PushSessionId* cookie.

- Click *Save*.

Activate Session Replication in the Application Server

In order for a IBM WebSphere application server to perform session replication for each Jacada WorkSpace session, configure Memory-to-Memory HTTP Session Replication for the deployed Jacada WorkSpace project using the WebSphere Integrated Solutions Console.

Note: Each redeployment of the Jacada WorkSpace project to the WebSphere application server de-activates its session replication. After deploying a Jacada WorkSpace .ear file to the WebSphere application server, re-configure session replication for the application.

To configure Memory-to-Memory HTTP Session Replication:

- In the task menu of the WebSphere Integrated Solutions Console, select *Applications* > *Enterprise Applications* > *<JacadaWorkSpace_application_name>* > *Session Management*.

The *Configuration* tab for the *Session Management* page displays.

Enterprise Applications [Close page](#)

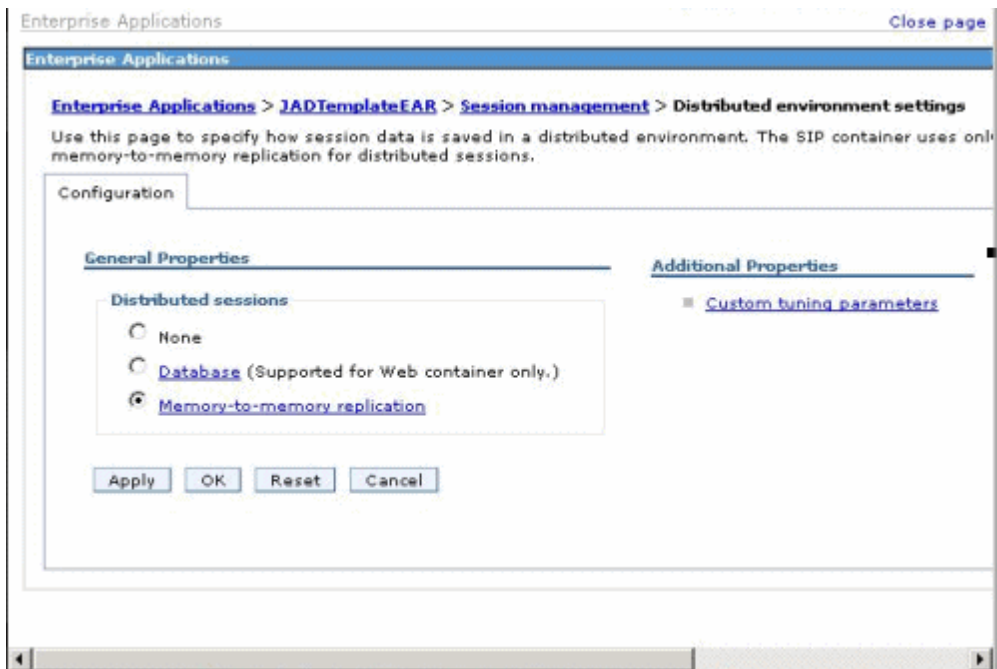
Enterprise Applications > [CommonAgentCallFusionEAR](#) > **Session management**

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These settings apply to both the SIP container and the Web container.

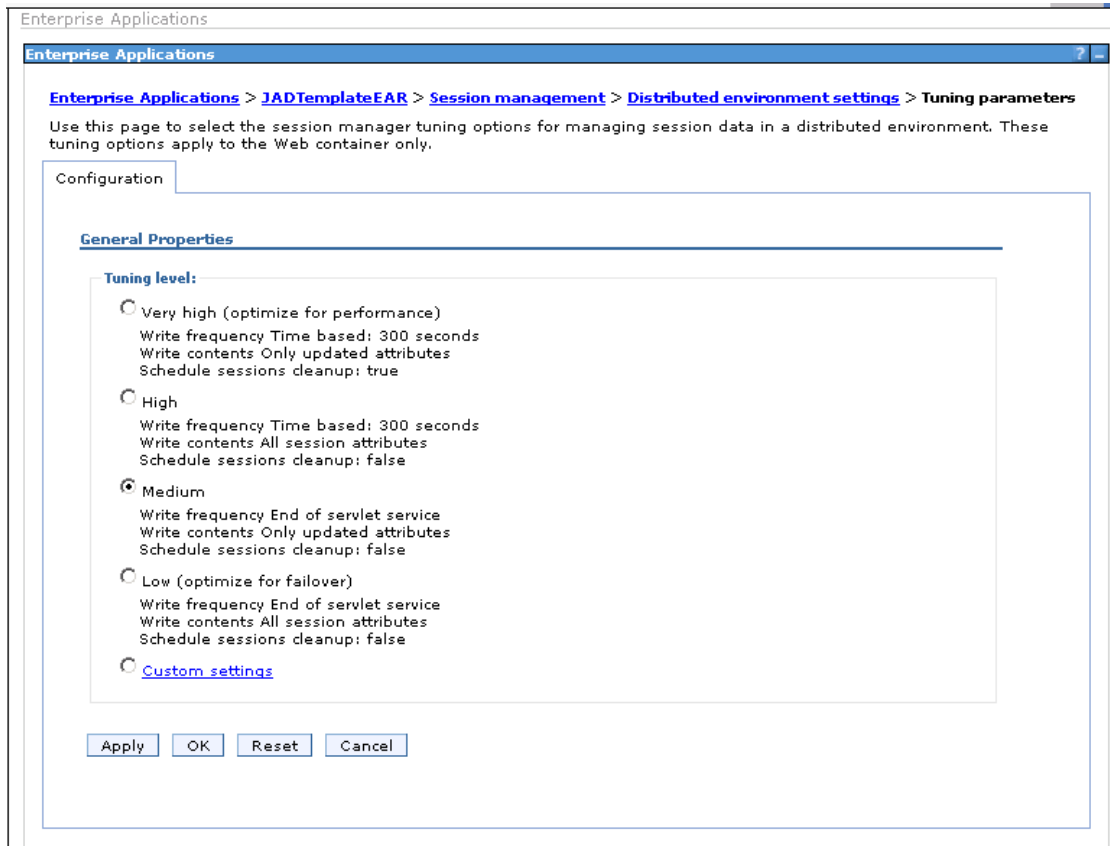
Configuration

General Properties	Additional Properties
☐ Override session management	☒ Distributed environment settings
Session tracking mechanism:	
☐ Enable SSL ID tracking	
☒ Enable cookies	
☐ Enable URL rewriting	
☐ Enable protocol switch rewriting	
☒ Allow overflow	
Maximum in-memory session count: 1000 sessions	

2. In the *General Properties* list, check the following options: *Override session management* and *Enable URL rewriting*.
3. Click *Apply*.
4. Click *Save*.
5. In the *Additional Properties* list, click the *Distributed environment settings* link.
The *Configuration* tab for the *Distributed environment settings* page displays.



6. In the *Additional Properties* list, click the *Custom tuning parameters* link.
The *Configuration* tab for the *Tuning Parameters* page displays.



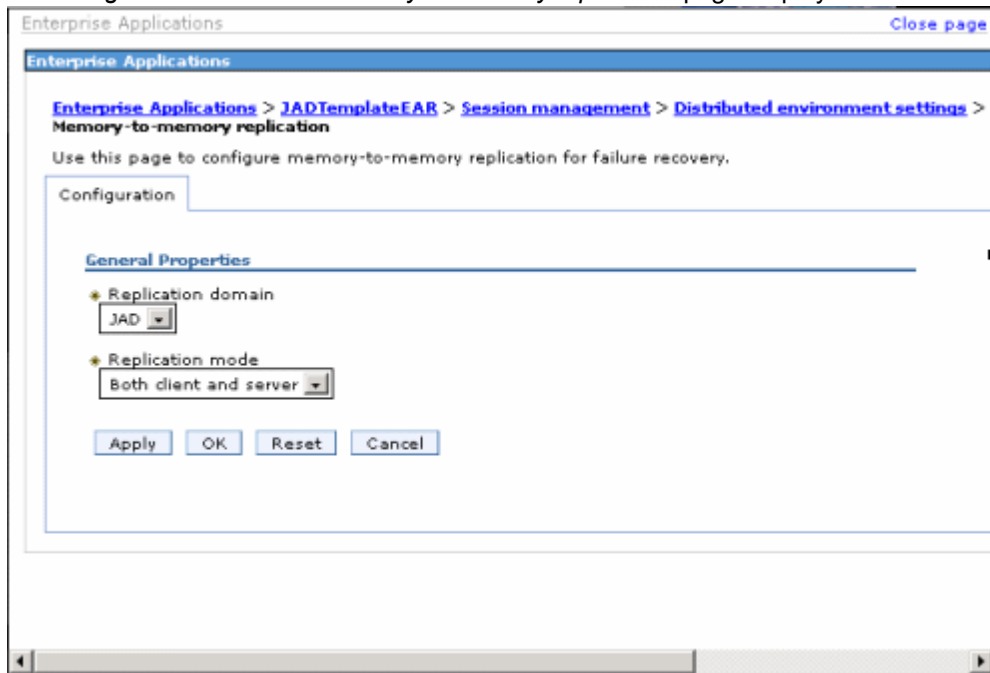
7. In the *General Properties* list, select the *Medium* tuning level option.
8. Click *Apply*.
9. Return to the *Distributed environment settings* page.
10. In the *General Properties* list, select the *Memory-to-memory replication* option.
11. Click *Apply*.
 - If this is the first time session replication is being configured for the deployed `<JacadaWorkspace_application_name>`, then the *Replication Domain* page displays. Continue with Step 12.
 - Else, the *Configuration* tab for the *Memory-to-memory replication* page displays. Continue with Step 15.
12. In the *Replication Domain* page, click *New*.

The *Configuration* tab for the *Replication Domain* page displays.

13. In the *Name* field of the *General Properties* list, provide a replication domain name to be used for Jacada WorkSpace applications.

14. Click *Apply*.

The *Configuration* tab for the *Memory-to-memory replication* page displays.



The screenshot shows a web browser window titled "Enterprise Applications" with a "Close page" link in the top right. The breadcrumb navigation path is "Enterprise Applications > JADTemplateEAR > Session management > Distributed environment settings > Memory-to-memory replication". Below the breadcrumb, a message states: "Use this page to configure memory-to-memory replication for failure recovery." The "Configuration" tab is selected. Under the "General Properties" section, there are two dropdown menus: "Replication domain" with "JAD" selected, and "Replication mode" with "Both client and server" selected. At the bottom of the configuration area are four buttons: "Apply", "OK", "Reset", and "Cancel".

15. In the *Replication domain* dropdown menu of the *General Properties* list, select the relevant, replication domain (the one created for Jacada WorkSpace).
16. In the *Replication mode* dropdown menu of the *General Properties* list, select the mode *Both Client and sever*.
17. Click *Apply*.
18. Click *Save*.

Memory-to-memory HTTP session replication is now established for the Jacada WorkSpace application.

Controlling the Impact of Infrastructure Failures

The following sections describe:

- Re-establishing push messaging following a communication loss
- Using durable subscription to retain messages

Re-establishing Push Messaging Following a Communication Loss

Upon your BIG-IP Local Traffic Management (LTM) system's re-route of traffic to an application server with functioning communications, following the detection of a loss of communications with a supported application server, it is vital that push messaging between any Jacada WorkSpace sessions (client-side) and Jacada WorkSpace(server-side) is re-established as quickly as possible. Accomplish the quick re-establishment of push messaging between each client-side and the server-side by reducing the push message timeout. Both a server-side setting and a client-side setting exist to control timeout for push messages. **To produce the required result, reduce both settings.** The client-side setting is modified in your project using the method `setTimeout()`. See the Jacada WorkSpace Developer Guide for the required code modification to reduce the client-side timeout setting for push messages.

To reduce the server-side timeout setting for push messages:

1. In the Jacada Management Console navigation panel, select the *Configuration > Global Settings* option.
The *Global Settings* page displays.
2. In the *PushTimeoutInMs* field, select the default setting of 120000 milliseconds (2 minutes) and replace it by typing 25000 milliseconds (25 seconds).
3. Click *Apply*.

The project's server-side timeout setting for push messages is updated in the Jacada WorkSpace database with value you provided in the *PushTimeoutInMs* field. Jacada WorkSpace, deployed in the application server, will operate using the updated setting value with any subsequent Jacada WorkSpace sessions that are established with it.

Using Durable Subscription to Retain Messages

In support of failover, employ the Java messaging service's durable subscription capability in all the application servers in the cluster. Using durable subscription causes the deployed Jacada WorkSpace project to maintain its subscription to the Java messaging service in the WebSphere application server host. A failover event causes the deployed Jacada WorkSpace project to

disconnect from the messaging service. With durable subscription activated, the Java messaging service continues to retain messages intended for the subscribed project. Upon the project's reconnection with the messaging service, following the failover event, Jacada WorkSpace receives any messages that were published for it during the disconnection period. WebSphere application servers require explicit configuration to activate durable subscription.

To activate durable subscription for push messaging and for the deployed Jacada WorkSpace:

1. In the Jacada Management Console navigation panel, select the *Configuration > Global Settings* option.

The *Global Settings* page displays.

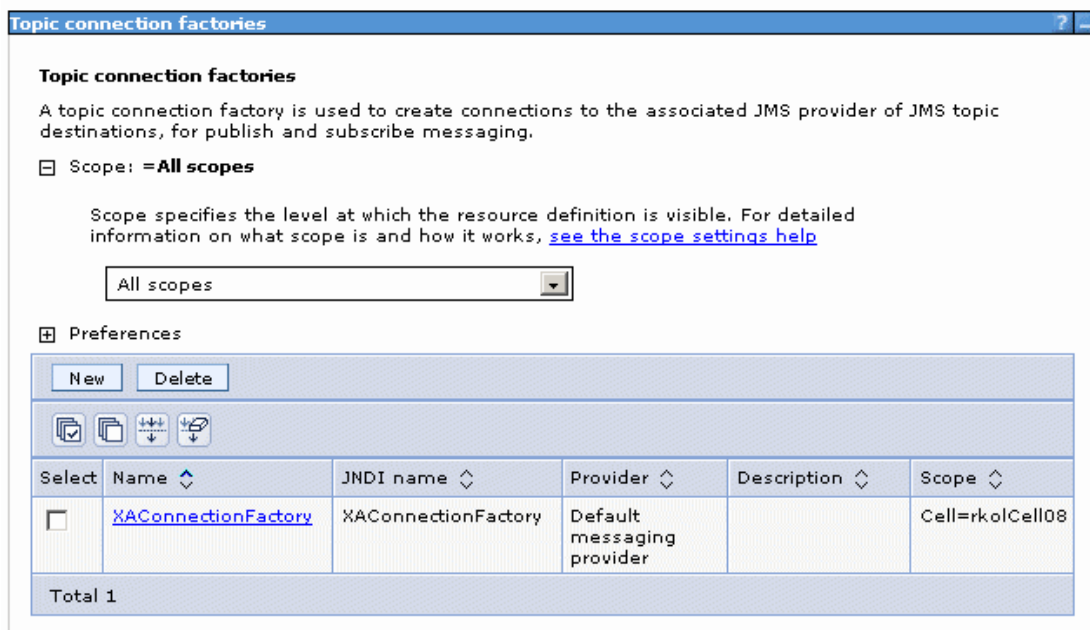
2. In the *DurableSubscription* field, replace the default setting of `false` with `true`.

3. Click *Apply*.

The `GLOBAL_SETTINGS` table in the Jacada WorkSpace database is updated with the value you provided in the *DurableSubscription* field. Jacada WorkSpace, will operate using durable subscription with any subsequent Jacada WorkSpace sessions.

4. In the task menu of the WebSphere Integrated Solutions Console, select *Resources > JMS > Topic connection factories*.

The *Topic connection factories* page displays listing the existing topic connection factories.



5. In the table of topic connection factories, click the *XAConnectionFactory* link. This is the Jacada WorkSpace topic connection factory created by the product installation.
The *Configuration* tab of the *XAConnectionFactory* displays.

Topic connection factories > **XAConnectionFactory**

A JMS topic connection factory is used to create connections to the associated JMS provider of JMS topics, for publish/subscribe messaging. Use topic connection factory administrative objects to manage JMS topic connection factories for the default messaging provider.

Configuration

General Properties

Administration

Scope
Cell=ekoCell08

Provider
Default messaging provider

* Name
XAConnectionFactory

* JNDI name
XAConnectionFactory

Description

Category

Additional Properties

■ [Connection pool properties](#)

Related Items

■ [JAAS - J2C authentication data](#)

■ [Buses](#)

Connection

* Bus name

- In the *General Properties* list, access the *Durable Subscription* pane.

Durable Subscription

Client identifier
pushSubscription

Durable subscription home
JadCluster.000-jadBus ▼

Quality of Service

Nonpersistent message reliability
Express nonpersistent ▼

Persistent message reliability
Reliable persistent ▼

- In the *Client identifier* field of the *Durable Subscription* pane, type `pushSubscription`.

8. Click *Apply*.
9. Click *Save*.

Durable subscription is now activated. The Java messaging service will treat the deployed Jacada WorkSpace's subscription messaging topic connection as a durable subscription. In doing so, Java messaging service will retain push messages published to the topic connection for Jacada WorkSpace should it disconnect from this service.

Effecting a Switchover

In a production environment that implements session replication, the system administrator may need to remove from service specific managed servers, for example, in order to perform system maintenance on them. To accomplish this, the administrator must switchover from those servers designated to be placed out of service to other, active, managed servers in the cluster.

To effect a processing switchover of managed application servers in a cluster:

1. In the relevant, BIG IP Load Traffic Manager load balancing pool, disable the managed servers that are designated for service removal.
2. Wait 30 seconds.
3. Shutdown these designated, managed servers.

To return managed application servers to active service:

1. Start the managed servers that are currently shutdown and out of service.
2. In the relevant, BIG IP Load Traffic Manager load balancing pool, enable these managed servers.

Failover and Switchover Support (WebLogic)

Failover is the capability of a product architecture's components to automatically switch processing responsibility from the active server, database or network that just failed to a standby server, database or network. Switchover is the capability to manually switch processing responsibility from the active server, database or network to a standby server, database, or network, upon the failure of the formerly active server, database or network.

Jacada WorkSpace provides a failover mechanism that enables session replication in a clustered environment. Failover is certified using the F5 Networks BIG-IP load balancer. For more information on BIG-IP, refer to [Load Balancing the WebLogic Server Cluster](#).

The following sections describe how to use the failover and switchover capabilities:

- [Session Replication in a Clustered Environment](#)
- [Controlling the Impact of Infrastructure Failures](#)
- [Effecting a Switchover](#)

Note: These topics are relevant only to users working with a WebLogic application server.

Session Replication in a Clustered Environment

An application server supports failover for Web applications by replicating the session state currently contained in memory. The server instance copies its session state in memory to another, designated server instance in the cluster. If the first server fails, then the replicated session state on the secondary server is located and used to continue session processing without any loss of the session's information. To support failover, Jacada WorkSpace enables you to work with the application server's session replication.

Note: As session replication may impact performance, Jacada recommends consulting your Jacada representative before implementing session replication.

Activate Session Replication in the Application Server

Activating session replication for a Oracle WebLogic application server requires a modification to the project code, in the **weblogic.xml** file. Perform this modification during Jacada WorkSpace project development. For details about the required modification, refer to the Jacada WorkSpace Developer Guide.

Controlling the Impact of Infrastructure Failures

The following sections describe techniques to limit the impact of infrastructure failures.

Re-establishing Push Messaging Following a Communication Loss

Upon your BIG-IP Local Traffic Management (LTM) system's re-route of traffic to an application server with functioning communications, following the detection of a loss of communications with a supported application server, push messaging between any Jacada WorkSpace sessions (client-side) and Jacada WorkSpace (server-side) must be re-established as quickly as possible. Accomplish the quick re-establishment of push messaging between each client-side and the server-side by reducing the push message timeout. Both a server-side setting and a client-side setting exist to control timeout for push messages. **To produce the required result, reduce both settings.** The client-side setting is modified in your project using the method `setTimeout()`. Refer to the Jacada WorkSpace Developer Guide for the required code modification to reduce the client-side timeout setting for push messages.

To reduce the server-side timeout setting for push messages:

1. In the Jacada Management Console navigation panel, select the *Configuration > Global Settings* option.
The *Global Settings* page displays.
2. In the *PushTimeoutInMs* field, select the default setting of 120000 milliseconds (2 minutes) and replace it by entering 25000 milliseconds (25 seconds).
3. Click *Apply*.

The project's server-side timeout setting for push messages is updated in the Jacada WorkSpace database with value you provided in the *PushTimeoutInMs* field. Jacada WorkSpace,

deployed in the application server, will operate using the updated setting value with any subsequent Jacada WorkSpace sessions that are established with it.

Using Durable Subscription to Retain Messages

In support of failover, employ the Java messaging service's durable subscription capability in all the application servers in the cluster. Using durable subscription causes the deployed Jacada WorkSpace project to maintain its subscription to the Java messaging service in the WebLogic application server host.

A failover event causes the deployed Jacada WorkSpace project to disconnect from the messaging service. With durable subscription activated, the Java messaging service continues to retain messages intended for the subscribed project. Upon the project's reconnection with the messaging service, following the failover event, Jacada WorkSpace receives any messages that were published for it during the disconnection period.

WebLogic application servers automatically use the durable subscription capability in a clustered environment. No explicit configuration is required to activate the durable subscription capability.

Effecting a Switchover

In a production environment that implements failover/session replication, the system administrator may need to remove from service, specific managed servers, for example, in order to perform system maintenance on them. To accomplish this operation in a production environment, the administrator must effect a processing switchover from those servers designated to be placed out of service to other, active, managed servers in the relevant, load balancing pool.

To effect a processing switchover of managed application servers in a cluster:

1. In the relevant, BIG IP Load Traffic Manager load balancing pool, disable the managed servers that are designated for service removal.
2. Wait 30 seconds.
3. Shutdown these designated, managed servers.

To return managed application servers to active service:

1. Start the managed servers that are currently shutdown and out of service.
2. In the relevant, BIG IP Load Traffic Manager load balancing pool, enable these managed servers.

Troubleshooting

The following topics describe some common problems with Jacada WorkSpace and how to resolve them, including items specific to Genesys CTI.

General Problems

Problem	Solution
The agent logs in to Jacada WorkSpace during a call and is logged in, in manual mode.	Due to CTI provider limitations, the agent cannot log in during a call.
Jacada WorkSpace hangs when changing to ready.	If the problem happens in the very first login to Jacada WorkSpace, check with the administrator that the credentials used for authentication are correct. Otherwise, check that after you click the <i>Ready</i> button, the status on the phone changes to <i>Ready</i> . If it does, log out and log in again.
An inbound call is made while the agent is handling a custom item. The counters continue to run even after the agent has finished handling the item.	The agent cannot receive an inbound call while a custom item is being handled. The agent must disconnect from the call.
The load balancer is not balancing efficiently.	Ensure that any keep-alives by the load balancer are set to send a GET or a POST request and not a HEAD request (which is the default request type for most load balancers).

Problem	Solution
A nested application in a tab is not displayed.	Ensure that the agent has read/write permission to the registry, or during set up of the agent machine, the registry key <pre>HKEY_LOCAL_MACHINE\SOFTWARE \Microsoft\Windows NT \CurrentVersion\Windows \AppDataInit_DLLs</pre> includes the value JFAPAppInitDLL.
Jacada WorkSpace displays ??? instead of text.	Ensure that the <code>DefaultLocale</code> setting in the <code>GLOBAL_SETTINGS</code> database table is set to a valid 2-letter value, such as <code>en</code> for English or <code>fr</code> for French.
Jacada WorkSpace displays unreadable/ corrupted message text, containing multiple question marks, in any of the following places: the <i>Welcome</i> page, the <i>Message Center</i> tab, the <i>Message Board</i> tab, the <i>Ticker Tape</i> and the <i>Ticker Tape Board</i> tab.	Ensure that the database is properly configured to support either the language in which the message text is meant to appear or UTF-8 text encoding.
Agent performs any one of the following call transfers: • Handshake • Warm • Conference + transfer • Conference + join	This scenario is not a problem. Smart Pad entry transfer depends upon <i>attach data</i>. Smart Pad entry transfer only occurs for call transfers where attach data is available. When transferring or consulting a call outside the switch (dials required access code + number), attach data does not appear and, therefore, Smart Pad entry transfer does not occur.

Problem	Solution
to a number outside of the switch (dials required access code + number). The call itself transfers; however, the agent's Smart Pad entries that are marked for transfer do not transfer. Note: Smart Pad entry transfer with call transfers is only supported for the CTI providers: AES and Genesys.	

Genesys CTI

The following sections provide information on troubleshooting Genesys CTI:

- [Troubleshooting Genesys CTI Connection Problems](#)
- [No Access to CTI Because of a Missing Genesys License](#)
- [Troubleshooting Errors in CTI Log Files](#)
- [Information Required for Genesys CTI Troubleshooting](#)

Troubleshooting Genesys CTI Connection Problems

Genesys provides information on the CTI connection by writing messages to a log file.

You begin troubleshooting a Genesys CTI connection problem, by reviewing the connection details. In order to troubleshoot the problem, it is necessary first to know what a successful connection looks like.

Details of a Successful Genesys CTI Connection

When the connection to CTI is successful, the connection details are listed in the application server output. These details include the CTI server host name and port, the name of the application, and connection information regarding various Genesys components such as `AIL` and `Telephony`.

Note: When connecting to Genesys CTI, exceptions may appear more than once. These exceptions can be ignored, as long as a successful connection to the Genesys CTI is ultimately established, as indicated by the following message: `Initialized Genesys AIL`. Otherwise, check the connectivity to your Genesys CTI.

Genesys CTI Connection Problems

When there is a problem in the connection to CTI, you may see an exception in the application server output, such as `ConfigTimeoutException` or `ServiceException`.

To troubleshoot a Genesys CTI connection problem, use the following guidelines:

1. Make sure the parameters displayed in the application server output are correct, including the Genesys server host name and port. These parameters are defined in the Jacada WorkSpace database, in the [CTI SETTINGS table](#).
2. Make sure the application is set up correctly in Genesys, including the application name, the T-Server and license configuration. For more information, refer to [Genesys T-Server Configuration](#).
3. Make sure that the status of the `AIL` and `Telephony` components is `On`.
4. Make sure that the Genesys T-Server is running.
5. Check the information in the Genesys log file `Ail.log`.
For details, refer to [Troubleshooting the Ail.log](#).
6. If all of the above items are correct, redeploy the CTI Gateway.

Troubleshooting the Ail.log

The `Ail.log` file is a Genesys log file, that is located by default in `<WebSphere_Home>\profiles\<profile>\logs`.

The information in the log file may indicate that one of the following problems exists:

- A network connectivity problem
- The host name is not recognized

Handling Network Connectivity Problems

When there is a network connectivity problem that prevents the Jacada WorkSpace server from connecting to the Genesys server, the following error may appear in the `Ail.log` file:

Cannot open socket: Connection refused

The problem may be in one of the ports used for maintaining a Genesys connection. Genesys uses a set of fixed ports: 3000, 2020, and 7260; and an additional varying port, which is used for the Genesys license daemon.

To check whether the fixed ports are open:

- Use the following Telnet command for each fixed port:

```
telnet <hostname> <port>
```

where <hostname> is the name of the host.

If a telnet session opens, the port is open.

If the command gets stuck, there is a problem accessing the port. The typical reason for this is a firewall. In order to use Genesys, the port must be open. Consult your network administrator.

To check the Genesys license daemon port:

- When a firewall is in use, check with the Genesys administrator that the Genesys license daemon is configured to use a fixed port.

For more information on Genesys licensing, go to <http://genesyslab.com/support/home/index/> (requires user and password).

Handling an Unrecognized Host Name

When the Jacada WorkSpace server does not recognize the Genesys host name, make sure the host name and IP appear in the platform's `hosts` file. In Windows, for example, the `hosts` file is located at:

```
C:\WINDOWS\system32\drivers\etc
```

No Access to CTI Because of a Missing Genesys License

Jacada WorkSpace prevents customers who do not have a license for Genesys from accessing Genesys via the Jacada CTI Gateway. When a license file does not exist, it is not possible to use Genesys CTI, only manual mode.

Troubleshooting Errors in CTI Log Files

The following CTI-related errors may appear in the `CTI.log` and the `CTIGateway.log` files:

- `ailFactory Error`
- `Timer Expired Error`
- `Operation Failed Error`
- `Agent Already Logged in Error`

Note: Log files are located by default in
`<WebSphere_Home>\profiles\<profile>\logs`

ailFactory Error

When the login of an agent to Jacada WorkSpace is unsuccessful, and results in CTI manual mode, the following message may be written to the log:

```
ailFactory could not create the agent ext:john.smith with agent id:john.smith
```

To troubleshoot this error, use the following guidelines:

1. Make sure the parameters displayed in the application server console are correct, including the Genesys server host name and port. These parameters are defined in the Jacada WorkSpace database, in the [CTI_SETTINGS table](#).
2. Make sure the application is set up correctly in Genesys, including the application name, the T-Server and license configuration. For more information, refer to [Genesys T-Server Configuration](#).
3. Make sure that the Genesys T-Server is running.
4. Check the information in the Genesys log file `Ail.log`.
For details, refer to [Troubleshooting the Ail.log](#).
5. Check the Agent/Person configuration in Genesys ([Working With Genesys](#)).
6. Use the Genesys Agent Desktop to try logging in with the same agent details, and if successful, then contact your Jacada support representative for further assistance.
7. If all of the above items are correct, redeploy the CTI Gateway.

Timer Expired Error

When the timeout for receiving an event is exceeded, the following message may be written to the `cti.log`:

The timer <timer name> for the action login has timed out. The following events were not received:ready, not-ready

where <timer name> is a string such as the following:

```
com.jacada.jad.cti.event.EventTimer@10f17ef
```

To troubleshoot this error:

- Check if the action specified in the message succeeded. (For example, `login` in the above message.) Use the `CTIGateway.log` for additional information.
- If the action succeeded, then check if the events specified in the message were received, using the `CTIGateway.log`.

Operation Failed Error

When a CTI operation fails, the following message may be written to the log:

```
In checkCTIResult got error result: <specific error message>
```

Troubleshooting such errors is complex as there is a wide range of possible causes for the error. This type of error may occur as a result of a CTI or switch-dependent restriction, such as:

- CTI or switch configuration does not allow using *Hold* button in internal calls
- CTI or switch configuration does not allow changing the state of the extension during a call

In another example, a message may be written to the log when the agent performs an action via the telephone, and then performs the same action via Jacada Workspace.

When an operation failed error occurs, if it is not a result of CTI or switch configuration, or a result of agent actions, consult your Jacada support representative for further assistance.

Agent Already Logged in Error

When an agent is already logged in to another extension, or the extension is already used by another agent, the following message may be written to the log:

```
Error in GenWrapper.checkIsLoggedIn() the following Dn is already logged in by
```

another agent. ... <details> ...

The details of the extension, Dn, and agent appear in the message, to enable you to resolve the problem.

Information Required for Genesys CTI Troubleshooting

When encountering problems with Genesys CTI that require assistance from Jacada Support, collect the following information:

- IP of the Genesys configuration server
- Genesys CTI connection details (screenshot of the application server console, or log file)
- Name or extension of the agent encountering problems
- Date and time of the problem
- Log files: Ail.log, CTI.log, CTIGateway.log
- An export of the application definition in the Configuration Manager Environment (CME), or screenshots if an export is not available