Oracle® Application Server

CDC Adapters for SQL Server User's Guide

10*g* Release 3 (10.1.3.4)

E13836-01

January 2009

ORACLE®

Oracle Application Server CDC Adapters for SQL Server User's Guide, 10*g* Release 3 (10.1.3.4)

E13836-01

Copyright © 2008, 2009, Oracle and/or its affiliates. All rights reserved.

Primary Author: David Goldman

Contributor: Sami Zeitoun, Arie Kremer, Dror Harari

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

1 Introduction

OracleAS CDC Adapter for SQL Server Overview 1-1
OracleAS Adapter for SQL Server CDC Agent Architecture 1-2
OracleAS CDC Adapter for SQL Server Architecture 1-3
The Staging Area 1-5

2 Installing Oracle Connect and Oracle Studio

Before you Begin 2-1
Microsoft SQL Server Requirements 2-1
Windows Hardware and Software Requirements 2-1
Hardware Requirements 2-1
Software Requirements 2-2
UNIX Requirements 2-2
Installing Oracle Connect on Windows 2-2
Installing Oracle Connect on a Windows Platform 2-2
Installing Oracle Connect on UNIX 2-3
Installing Oracle Connect on a UNIX Platform 2-3
Preinstallation Tasks 2-3
Installing Oracle Connect 2-4
Installation Tasks 2-4
Post-Installation Tasks 2-5
Installing Oracle Studio 2-6
Oracle Studio Requirements 2-6
Installing Oracle Studio on Windows 2-7
Installing Oracle Studio on Linux 2-7
Installing Oracle Studio with the Wizard (SH installation) 2-7
Installing with a Silent Installation (RPM) 2-7

3 Configuring the OracleAS CDC Adapter for SQL Server

Setting Up a Windows Computer in Oracle Studio 3-1
Securing Access to Oracle Connect 3-2
Setting Password Access to Oracle Studio 3-3
Specifying Users with Administrative Rights 3-4
Setting Up Run-Time User Access 3-5
Setting Up a Change Data Capture with the OracleAS CDC Adapter for SQL Server 3-7

Enabling SQL Replication 3-7
SQL Server 2000 Replication 3-8
SQL Server 2005 Replication 3-8
Configuring Security Properties 3-8
Setting up Log On Information 3-9
Setting up the Database 3-10
SQL Server 2000 Settings 3-10
SQL Server 2005 Settings 3-10
Setting Up the TLOG Miner (LGR) 3-11
Call the LGR Service Interface 3-11
Configuring the Template Input File 3-12
Registering the TLOG Miner (LGR) Service 3-14
Setting the Recovery Policy 3-15
Setting up a Change Data Capture in Oracle Studio 3-15
Create a CDC Project 3-16
Set up the CDC Server 3-18
Set up the Staging Area Server 3-22
Deploying a Change Data Capture 3-32
Activating and Deactivating Solution Workspaces 3-36
Testing the OracleAS CDC Adapter for SQL Server CDC Solution 3-36
Handling Metadata Changes 3-37
Environment Verification 3-38
Verify the SQL Server Version 3-38
Ensure that the Service is Registered 3-39
Verify that the LGR Service is Running 3-40
Viewing the Service Greetings 3-40
Check the Output Files 3-41

4 Oracle BPEL Process Manager Deployment and Integration

Overview of Oracle BPEL Process Manager Integration 4-1
Configuring the SQL Server CDC Adapter In the Application Server 4-1
CDC Stream Positions Table Definition 4-2
Configuring the SQL Server CDC Adapter Connection Factory 4-3
Configuring the Oracle BPEL Process Manager 4-4
Setting up the Connection to the Oracle Connect Server Metadata 4-4
Using JDeveloper to Integrate a CDC Inbound Process to the Inbound Application 4-4
Configuring a CDC Adapter Partner Link 4-5
Configuring WSDL 4-5
Troubleshooting Rollback Loops 4-8

5 Runtime Tasks and Troubleshooting

Adding and Removing Tables 5-1
Handling Metadata Changes 5-2
Staging Area Maintenance 5-3
Monitoring the Change Data Capture 5-4
Service Context Table 5-4
Monitoring the Status 5-5

Daemon Life-Cycle Tasks 5-7
Starting the Daemon 5-7
Shutting Down the Daemon 5-7
Monitoring the Daemon During Runtime 5-7
Daemon (Computer) Options 5-8
Workspace Options 5-8
Server Options 5-9
Daemon Logs 5-9
The Log Monitor 5-10
Resolving Communication Errors 5-11
Resolving Specific Errors 5-12

6 Advanced Features of OracleAS CDC Adapter for SQL Server

Configuring the Daemon for High Availability 6-1
Adding a New Daemon Workspace Configuration 6-1
Editing the Workspace 6-2
Configuring the Server Mode 6-2
Configuring a Binding Environment 6-6
Debug 6-7
General 6-8
Language 6-9
Modeling 6-9
ODBC 6-9
OLEDB 6-9
Optimizer 6-9
Parallel Processing 6-10
Query Processor 6-10
Transaction 6-13
Tuning 6-13
XML 6-14
Migration Considerations 6-15
Security Considerations 6-15
Setting Design Time Security 6-16
Setting Runtime Security 6-16
Transaction Support 6-17

A Advanced Tuning of the Metadata

Metadata for the SQL Server Data Source A-1
General Tab A-2
Columns Tab A-3
Column Definition Section A-4
Column Properties A-5
Indexes Tab A-7
Table Information A-7
Properties A-8
Statistics Tab A-8

Generating Statistics A-10
Advanced Tab A-11
Adapter Metadata A-13
Adapter Metadata General Properties A-13
Adapter Metadata Interactions A-15
Interaction Advanced Tab A-16
Adapter Metadata Schema Records A-18
Adapter Metadata XML A-20

B Advanced Tuning of the Daemon

Control B-1
Logging B-3
Security B-6
Workspaces B-8
General B-8
Server Mode B-11
Security B-16

C Editing Properties for the OracleAS CDC Adapter for SQL Server

Editing Properties in Oracle Studio C-1
Editing the OracleAS CDC Adapter Properties C-1
Configuration Properties C-2
CDC Logger Properties C-2
OracleAS CDC Adapter Configuration Properties C-2
OracleAS CDC Adapter for SQL Server Properties C-2
CDC Queue Adapter Properties C-3
Common CDC Adapter Properties C-3
Change Router Properties C-4
Referential Integrity Considerations C-7
Access to Change Events C-9
Transaction Support C-10
Security C-10
Data Types C-10
User Defined Data Types (UDT) C-11

D Globalization Settings

Defining the Language and Codepage D-1

E Editing XML Files in Oracle Studio

Preparing to Edit XML Files in Oracle Studio E-1
Making Changes to the XML File E-2
Remove Objects E-2
Add DTD Information E-2
Edit Namespaces E-3
Add Elements and Attributes E-5

Replace an Element E-5

Index

Preface

This guide is the primary source of user and reference information on the OracleAS CDC adapter for SQL Server, which enables Oracle users to use change data capture in their integration architecture, working with the Oracle Application Server.

This document describes the features of the OracleAS CDC adapter for SQL Server.

This preface covers the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Conventions

Audience

This manual is intended for Oracle integration administrators and developers who perform the following tasks:

- Installing and configuring OracleAS CDC Adapters for SQL Server
- Using OracleAS for developing integration solutions using change data capture

> **Note:** You should understand the fundamentals of OracleAS, OC4J, the UNIX and Microsoft Windows operating system before using this guide to install or administer OracleAS Adapters for SQL Server.

- Diagnosing errors

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at `http://www.oracle.com/accessibility/`.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at `http://www.consumer.att.com/relay/tty/standard2.html`. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

Related Documents

For more information, see the following documents in the Oracle Other Product One Release 10.0 documentation set or in the Oracle Other Product Two Release 6.1 documentation set:

- *Oracle Application Server Adapter Concepts Guide*
- *Oracle Application Server Adapter Installation Guide*
- *Oracle Application Server Adapter Concepts Guide*
- *Oracle Application Server Containers for J2EE User's Guide*
- *Oracle Application Server Containers for J2EE Services Guide*
- *Oracle Application Server Containers for J2EE Security Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 Introduction

Change data capture (CDC) is an integration technology that enables efficient near real-time integration by capturing changes made to enterprise data sources and providing them for processing in integration middleware, such as Oracle Fusion Middleware.

The Oracle CDC Adapter for SQL Server provides such capabilities for SQL Server 2000 and 2005, capturing changes in a non-intrusive manner leveraging the SQL Server Transaction Logs (TLOG). The CDC Adapter can be used by Oracle SOA Suite (BPEL, ESB), and Oracle Data Integrator (ODI).

This section provides an overview of the features and architecture of the OracleAS CDC adapter for SQL Server. It includes the following topics:

- OracleAS CDC Adapter for SQL Server Overview
- OracleAS Adapter for SQL Server CDC Agent Architecture
- OracleAS CDC Adapter for SQL Server Architecture

OracleAS CDC Adapter for SQL Server Overview

The OracleAS CDC Adapter for SQL Server captures and delivers changes (such as insert, update, and delete operations) made to tables in SQL Server 2000 and 2005. By using the CDC Adapter, you can build efficient data integration solutions that use SQL Server data, synchronize systems to ensure data integrity, update data warehouses and operational data stores in near real-time, and enable event-driven business activity monitoring and processing.

The OracleAS CDC adapter for SQL Server has the following capabilities:

- **Non-intrusive change capture**: The OracleAS CDC Adapter for SQL Server is a log-based CDC technology. It leverages the SQL Server Transaction Logs (TLOG) to identify changes and does not add triggers or tables into the database.

  The CDC agent accesses a logstream that is polled to access the changes. The agent begins polling from a specific point called the stream position (or agent context). The last position read is saved as the stream position. This next time the agent polls the logstream for information, it starts at the saved stream position.

- **Real-time data capture**: The OracleAS CDC Adapter for SQL Server captures changes to data in near-real-time for applications that demand zero latency and require the most up-to-date data. Unlike scheduled CDC solutions, near real-time data capture guarantees that a change event is can be made available in a matter of seconds at the consumer. The term near-real-time data capture is used because after a period of inactivity changes may be available after a few seconds.

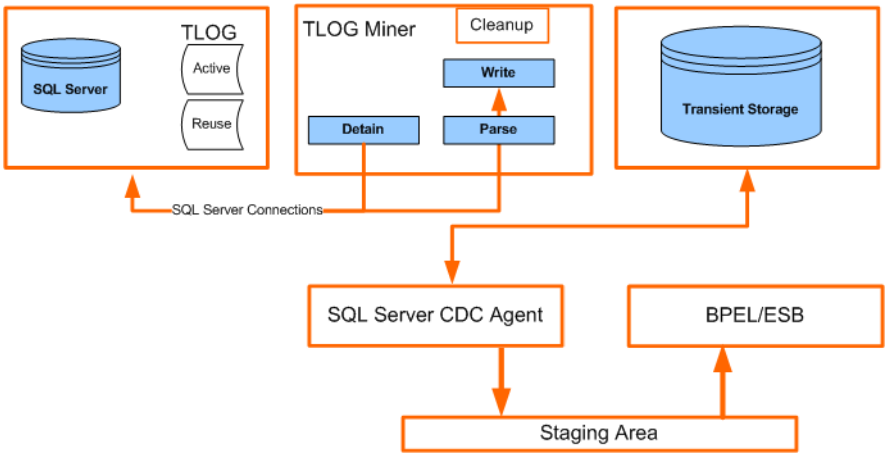
- **Change storage**: The Oracle CDC Adapter for SQL Server provides a staging area that stores the captured changes in a format that is easy to access and process.The staging area ensures that changes are persisted until they are processed, and provides facilities to clean up old change records.
- **Change access and delivery**: The Oracle CDC Adapter for SQL Server includes a JCA Resource Adapter that can be configured in BPEL and ESB as an inbound endpoint, delivering changes in XML format.
- **Filtering**. The OracleAS CDC Adapter for SQL Server enables you to filter captured data based on the type of operation (INSERT, UPDATE, DELETE), based on column selection and even based on specific column values.
- **Reliable delivery and recovery**: The Oracle CDC Adapter for SQL Server stores changes in a reliable persistent storage and uses the concept of a 'stream position' that indicates the location in the change stream that was successfully moved and processed. By storing stream positions, the CDC Adapter can always restart after failure, recover, and continue moving and delivering changes from wherever it was stopped
- **Ease of use**: The OracleAS CDC Adapter for SQL Server provides an easy and intuitive way to confugure CDC solutions using Oracle Studio.

Using the OracleAS CDC Adapter for SQL Server makes ETL (extract, transform, and load) processes more efficient and less disruptive to the operation of the source database systems, eliminating the need for the traditional ETL downtime window

OracleAS Adapter for SQL Server CDC Agent Architecture

The OracleAS CDC Adapter for SQL Server CDC agent uses the normal SQL Server transactin logs (TLOG) and a TLOG miner to extract the necessary data and place it into the staging area where it is available for consumption by BPEL/ESB. The following diagram shows this process:

Figure 1–1 SQL Server CDC Agent Architecture



The agent is a part of the overall OracleAS Adapter architecture that is described in OracleAS Adapter for SQL Server CDC Agent Architecture. The following describes the components shown in the diagram:

- **SQL Server**: The Microsoft SQL server creates transaction logs that log server activity for recovery purposes. When using Replication, the logs hold information in a format that is consumable by Oracle's CDC agents.

The active section of the TLOG contains the changes made to the currently active transaction.

The reusable or inactive section has the information from older transactions, which do not require further processing by MS SQL server. This space is reusable. It is possible to back up the data or the MS SQL Server might truncate the log to create more space. When a TLOG is truncated, some of the data is dropped and is no longer available for the CDC agent to use. The Oracle CDC solution for the MS SQL Server is designed to prevent potential data loss.

- **TLOG miner** (LGR): This component is installed as a stand-alone Microsoft Windows service. It reads the TLOG file and extracts or mines the change data and stores it in the Transient Storage area. It has two parts.
  - TLOG Detainer: The Microsoft SQL Server management policy periodically reorganizes the log files when necessary. In this case, the log files are truncated when their data is no longer active. The truncated data is erased from the system and cannot be used. Occasional truncation of the transaction LOG can expose the LGR to potential loss of data. The detainer is used to ensure the TLOG files are not truncated before being read by the TLOG miner.

    Truncation only takes place in the non-active section of the TLOG. The detainer places a detained transaction behind the logged records to be read. This creates a limit for the TLOGs active portion, which protects records from being truncated before they are read.
  - TLOG Parser: The parser parses the TLOG information and then writes it into Transient Storage.
- **Transient Storage**: This is a set of binary files located in a specified working directory where the LGR stores the data it mines from the TLOG into Transient Storage. The log records are kept in Transient Storage according to the LGR cleanup policy and the maximum size limit specified at coniguation time.

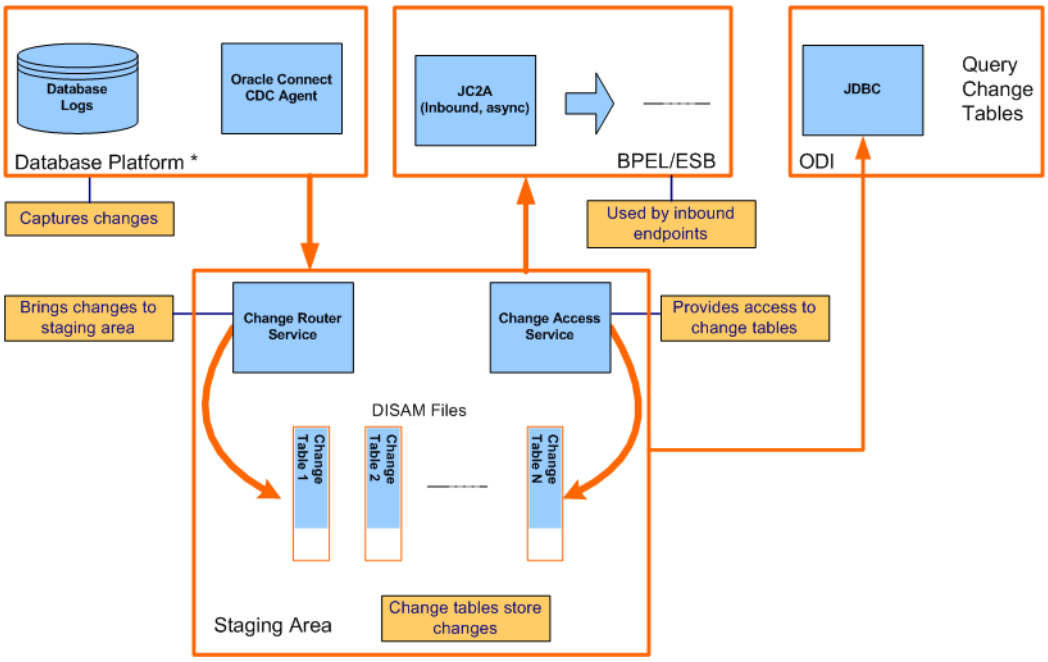
  The OracleAS CDC Adapter for SQL Server CDC agent reads the changes from Transient Storage, parses it, orders it in committed transaction order, and places it in the staging area where it can be consumed by BPEL and ESB.

  For more information see, Setting Up a Change Data Capture with the OracleAS CDC Adapter for SQL Server.
- **SQL Server CDC agent**: This is part of the OracleAS Adapter for SQL Server CDC Agent Architecture.
- **BPEL/ESB**: This is part of the OracleAS Adapter for SQL Server CDC Agent Architecture.
- **Staging Area**: This is part of the OracleAS Adapter for SQL Server CDC Agent Architecture.

OracleAS CDC Adapter for SQL Server Architecture

The following figure illustrates the system components used for change data capture using the OracleAS CDC adapter for SQL Server.

Figure 1–2 Component Architecture used with the SQL Server CDC adapter



The OracleAS CDC adapter for SQL Server component architecture includes the following components:

- **Database Platform**: The database platform is the data source that contains the data to be captured.
- **Database Log**: The database log is a log stream that contains the raw change data.
- Oracle Connect: Oracle Connect runs on the backend system and handles from the special J2CA adapter that runs on the Oracle Application Server.
- **CDC agent**: This is an OracleAS CDC Adapter for SQL Server component that runs under the control of the daemon (a listener process) on the computer where the backend data is located. The capture agent provides access to change events in the transaction log for the change router.

  The agent is located on the same computer as the changes to be captured. It maintains the last position read in the journal (the stream position or context) and starts at this point the next time it polls the journal. The context is stored in the repository where the adapter definition is stored. The adapter definition includes a single primary interaction which is used to access the appropriate journal and includes the list of tables to monitor for changes.
- **BPEL/ESB**: This is where the change events are consumed by means of a JCA provider that sends change records to an end point configured in either ESB or BPEL.
- **JC2A Resource Adapter**: A special version of the J2CA 1.5 SQL Server adapter that can read change records from the staging area. It implements inbound endpoints for BPEL and ESB that pulls change events from the staging area and deliver them as XML change documents. Each endpoint handles change records for a single captured table at the source database.

  The J2CA adapter uses multiple connections to the staging area to be able to process changes from multiple tables in parallel.
- **JDBC Driver**: Enables access to changes in the staging area using SQL queries.

- **OC4J / BPEL Components**: Any OracleAS component that hosts and gets messages from the JC2A adapter.
- **Staging Area**: This is a set of change tables where change records are stored before being sent to BPEL/ESB. Old records from the Staging Area are deleted based upon the specified aging policy.

  The Staging Area contains a DISAM data source for SQL access and a Database Events adapter for the XML access.

  See The Staging Area for more information.

  The staging area includes the following components:

  - **Change Router**: This is an OracleAS CDC Adapter for SQL Server service that runs under the control of the daemon (a listener process) on the Oracle platform. The change router reads the change stream from the CDC agent, sorts them in transaction/commit order eliminating rolled-back changes, and distrubutes the changes into the various change tables.

    Only one instance of the change router is active at a time against a captured database. The change router also deletes old changes from the change tables according to a configurable aging policy. policy (48 hours is the default, which means that records that have been in the staging area for more than 48 hours will be deleted).
  - **Change Tables**: The staging area is made of change tables, each containing changes to one captured table in the source database. These change tables maintain change events in transaction order and in occurrence order within a transaction for each captured table. The change tables are DISAM files that are indexed to support fast access to the change data, that can be accessed directly using the stream position.
  - **Change Access Service**. This is an instance Oracle Connect that is installed on a separate computer or the same computer as BPEL/ESB that allows access to the change tables. This instance of Oracle Connect is configure as the staging area.
- **Oracle Studio**: Provides the developer or administrator with a graphical user interface for setting up the CDC Agent and Staging Area,using intuitive wizards. Oracle Studio can be installed on Linux, UNIX, or Windows.

The Staging Area

The staging area is an area used by Oracle Connect to store captured data from a journal. The journal is scanned once and changes for every required table read during that scan are passed to the staging area, where they are stored. This means that the journal is scanned once each time it is polled. After the changes are written to the staging area, processing of these changes is carried out independently of the journal.

The staging area is a benefit when transactions are used. The changed data is not written to the change queue until the transaction is committed. Thus, if the transaction fails, there is no overhead of having to back out any processing done with the steps in the failed transaction.

The staging area can be on any computer with a Windows, Linux, or UNIX operating system that is running the Oracle Application Server and does not need to be run on the same server as the SQL Server CDC adapter. Once the information is extracted from the journal and written to the staging area, processing of changes is carried out on the staging area only. Thus, the staging area should be setup to consider the network configuration and where the consumer application runs.

The staging area maintains the last position read by the consumer application (the staging area context) and starts at this point the next time a request from the consumer application is received. The context is stored in the repository where the staging area is maintained.

The staging area is indexed so that access to the staging area for a specific stream is quick.

The staging area is cleared by default every 48 hours. All events that have been in the staging area for more than 48 hours are deleted.

2

Installing Oracle Connect and Oracle Studio

This section describes how to install Oracle Connect and Oracle Studio.

> **Note:** In addition to the installation procedures described in this section, the J2CA 1.5 adapter must be installed with Oracle Application Server. Installing the J2CA 1.5 adapter is described in the *Oracle Application Server Adapter Installation Guide*.

This section includes the following topics:

- Before you Begin
- Installing Oracle Connect on Windows
- Installing Oracle Connect on UNIX
- Installing Oracle Studio

Before you Begin

Before installing OracleAS Adapter for SQL Server, ensure that you have carried out the following:

- Microsoft SQL Server Requirements
- Windows Hardware and Software Requirements
- UNIX Requirements

Microsoft SQL Server Requirements

Microsoft SQL Server 2000 or 2005 (Standard, Enterprise, or SBS editions). For MS SQL Server 2005 you must use Service Pack 1 or higher. You must install SQL Server where you install Oracle Connect.

Windows Hardware and Software Requirements

This section describes the following requirements for installing Oracle Connect:

- Hardware Requirements
- Software Requirements

Hardware Requirements

The following table summarizes the hardware requirements for Oracle Connect.

Table 2–1 Oracle Connect Hardware Requirements

Hardware Component	Requirements
Processor	An Intel or 100% compatible computer, based on a Pentium processor
Memory	256 MB of RAM
Disk Space for Oracle Connect	40 MB of free disk space

Software Requirements

The following table summarizes the software requirements for Oracle Connect.

Table 2–2 Oracle Connect Software Requirements

Software Component	Requirements
Operating System	Microsoft 2000 with service pack 2 or higher, Microsoft Windows XP, or Microsoft Windows Server 2003.
Microsoft	Network transport protocol software, TCP/IP, included with Microsoft Windows.

UNIX Requirements

The following UNIX operating systems are supported:

- Red Hat Linux
- Linux SUSE
- HP-UX
- IBM AIX
- Sun Solaris

For all of the operating systems, the required disk space is 48MB.

Installing Oracle Connect on Windows

You must install Oracle connect on the Windows computer where you install Microsoft SQL Server and on any machine that you are using for a staging area for the CDC adapter. This can be the same computer or different computers.

This following sections explain how to install Oracle Connect on a Windows machine.

- Installing Oracle Connect on a Windows Platform

Installing Oracle Connect on a Windows Platform

This section explains how to install Oracle Connect. Do the following to install Oracle Connect.

Note: If you are upgrading from version 10.1.3.3, you must use the upgrade option in the installation wizard. Do not uninstall Version 10.1.3.3 and then install the newer version. If you do this, you will lose all data in the `Def` directory.

---

1. Copy the following installation files into a folder on the Windows computer where you are installing Oracle Connect.
   - `OCL_10134_windows.exe`
   - `brand.bin`
2. Open the Windows command line interface.
3. Change to the directory where you installed the installation files.
4. Type the following at the command prompt to install Oracle Connect.

   ```
   "OCL_10134_windows.exe" -a -bBRAND=".\brand.bin" -bSERVER
   ```

   The install wizard opens. Follow the directions in the insallation wizard to complete the installation.

   Note: If you are installing Oracle Connect on a Windows XP computer, you cannot use a logical drive as the destination folder for the installation.

Installing Oracle Connect on UNIX

You must install Oracle connect on any UNIX machine that you are using for a staging area for the CDC adapter. The following sections explain how to install Oracle Connect on a UNIX machine.

- Installing Oracle Connect on a UNIX Platform

Installing Oracle Connect on a UNIX Platform

This section explains how to install Oracle Connect on a UNIX platform. This section includes the following:

- Preinstallation Tasks
- Installing Oracle Connect
- Installation Tasks
- Post-Installation Tasks

Preinstallation Tasks

Before starting the installation procedure, ensure that you have the following information is available:

- The root directory where you want to install Oracle Connect.

  Notes: The root directory cannot be a system root directory, `/var` or `/tmp`.

- The account name where Oracle Connect will run.
- Whether the installation source media is removable media or a disk archive file.

  If you are installing from the removable media that is not a default device in the system, you need to know the media device name (such as `/dev/rmt/0m`). See the operating system manuals or ask the system administrator to find out the device name for the site.

If you are installing from the disk archive file, you need to know the name of the Oracle Connect disk archive file (such as /tmp/nav.2.0.tar).

- The shell being used: C-shell Korn,-shell, or Bourne-shell. The installation creates a startup file according to the indicated shell.

When running the installation below, use the following as the `<filename>`:

- For RedHat Linux: `OCL10134-linuxrh.tar.Z`
- For Linux Suse: `OCL10134-linuxsuse.tar.Z`
- For HP-UX: `OCL10134-hpux.tar.Z`
- For IBM AIX: `OCL10134-ibmaix.tar.Z`
- For Solaris Operating System (SPARC): `OCL10134-sunsol2.8.tar.Z`

Installing Oracle Connect

Perform the following steps to install Oracle Connect:

1. Transver the tar.Z file to th esystem.
2. Decompress the file using the following command:

   ```
   uncompress <filename>
   ```

3. Run the tar command, as shown in the following example:

   ```
   tar xvf <filename> nav_install
   ```

   The following message is displayed:

   ```
   x nav_install, nnnn bytes, mmmm tape blocks
   ```

Note: Ensure that the directory used to run the installation files has `WRITE` privileges.

Installation Tasks

Perform the following steps to install Oracle Connect:

1. Run the following command:

   ```
   ./nav_install
   ```

   This command initiates the installation procedure. The installation procedure is displayed in a series of screen prompts and responses.

2. Enter the full path of the disk archive (.tar) file, and press **Enter**.
3. Enter the root directory name for the installation, and press **Enter**. You must have a `WRITE` permission for this directory. The default directory is the users home directory.

Notes:

- The root directory cannot be a system root directory or `/var` or `/tmp` directory.
- Oracle Connect is installed into a fixed directory named `navroot`.

---

4. Confirm the directory name in which Oracle Connect will be installed, and press **Enter**.
5. Enter the account name where you want Oracle Connect to run, and press **Enter**. This account name will be used for anonymous access to the server by clients. It can be changed after the installation is complete.
6. Confirm the account name, and press **Enter**.
7. Specify the required shell, under which Oracle Connect should run, and press **Enter**. The following options are displayed:
   - C-shell (`/bin/csh`).
   - Korn-shell (`/bin/ksh`)
   - Bourne-shell (`/bin/sh`)
8. Enter the account name for a user with administrative authorization. Optionally, press **Enter** to enable any user to administer Oracle Connect.

Post-Installation Tasks

After installing Oracle Connect, perform the following post-installation tasks:

- Configuring the Oracle Connect Environment
- Configuring the Oracle Connect Script
- Starting the Oracle Connect Daemon

Configuring the Oracle Connect Environment

When Oracle Connect is installed on a UNIX platform, using FTP, copy the `brand.bin` file to the Oracle Connect computer, to *`NAVROOT`*`/bin`.

Where *NAVROOT* is the directory where Oracle Connect is installed.

Configuring the Oracle Connect Script

The program that manages Oracle Connect server processes (`nav_server`) is accessed by a symbolic link to a file for the C-shell, Bourne and Korn shells.

To set up nav_server, perform the following steps:

1. In the `bin` directory, under the directory where Oracle Connect is installed, delete the existing link to nav_server using the following command:

   ```
   rm nav_server
   ```

2. In the `bin` directory, under the directory where Oracle Connect is installed, link to the required version of nav_server as follows:
   - C-shell: `ln -s nav_server.csh nav_server`
   - Bourne: `ln -s nav_server.sh nav_server`
   - Korn: `ln -s nav_server.ksh nav_server`

   Note: Instead of renaming files, use a symbolic link.

The Oracle Connect `nav_login` procedure defines the default environment when Oracle Connect runs. If you want site-dependent variables to be included in the environment, create a file called `site_nav_login` and save this file in the bin

directory under the Oracle Connect root directory. `nav_login` runs `site_nav_login` automatically.

`nav_login` must be invoked to run Oracle Connect. It can be invoked from the user login script.

The command line for invoking `nav_login` varies according to the shell the user is running. The following table lists the different options for invoking the command line:

Shell	nav_login Command
CSH	`source` *`root`*`/bin/nav_login`
Bourne	`.` *`root`*`/bin/nav_login.sh`
Korn	`.` *`root`*`/bin/nav_login.sh`

In the nav_login command, *root* represents the root directory of the Oracle Connect installation. After running the login procedure, the environment variable `NAVROOT` points to this root directory.

Ensure that users have `READ` and `EXECUTE` permissions on the Server files. Use the `chmod` command to change the permissions.

Starting the Oracle Connect Daemon

The Oracle Connect daemon must run on a server for client/server access to Oracle Connect. To start the daemon with the system startup, add the following command invoking the daemon to the end of the `/etc/inittab` file:

```
nv:3:once:navroot/bin/irpcd -l ip:2551 start >/dev/console 2>&1
```

In this command, the symbol *navroot* should be replaced with the directory where Oracle Connect is installed and *ip* replaced by the ip address of the computer.

> **Note:** To allow automatic client/server access to Oracle Connect, start the daemon at system startup time from a super user account.

Installing Oracle Studio

This following sections explain how to install Oracle Studio.

- Oracle Studio Requirements
- Installing Oracle Studio on Windows
- Installing Oracle Studio on Linux

> **Note:** If you have Oracle Studio version 10.1.3.4 or higher installed on your computer because you are using an OracleAS legacy adapter or OracleAS CDC adapter , you do not need to reinstall it. If you have an older version of Oracle Studio, you must install the newest version.

Oracle Studio Requirements

The following are the hardware requirements for Oracle Studio:

- Processor: Intel or 100% compatible computer, based on a Pentium processor

- Memory: 256 MB
- Disk space: 120 MB of free disk space

You can install Oracle Studio on the following operating systems:

- Windows XP with Service Pack 2 or higher

  Windows 2003

  Windows Vista
- Linux OS with GTK

Installing Oracle Studio on Windows

Oracle Studio is installed with a standard install wizard. Do the following to install Oracle Studio.

- Run the installation file, either using the Run option in the Windows Start menu or through Windows Explorer. Follow the instructions on the wizard screen.

Installing Oracle Studio on Linux

The Oracle Studio installation on Linux can be caried out by one of the following:

- Installing Oracle Studio with the Wizard (SH installation)
- Installing with a Silent Installation (RPM)

Installing Oracle Studio with the Wizard (SH installation)

You can install Oracle Studio by using the installation wizard. This allows you to easily carry out any standard or custom installation. Do the following for the Linux SH installation.

1. Install into a directory where you have permission.
2. Change the mode to execute mode. Type in:

   ```
   Chmod +x <file name>
   ```
3. Enter the following:

   ```
   ./ <file name>
   ```
   If you are not installing to the current directory, enter the full path.

Installing with a Silent Installation (RPM)

You can use the Linux RPM method to install Oracle Studio. Do the following to carry out a silent installation on Linux.

- Enter the following:

  ```
  rpm -i <file name>
  ```
  You can use RPM parameters to create a custom inslallation.

3

Configuring the OracleAS CDC Adapter for SQL Server

This section describes how to configure Oracle Connect using Oracle Studio.

All modeling of Oracle Connect is performed using Oracle Studio. To use Oracle Studio, you first configure it to enable access to a Windows computer where the SQL Server database resides.

This section includes the following topics:

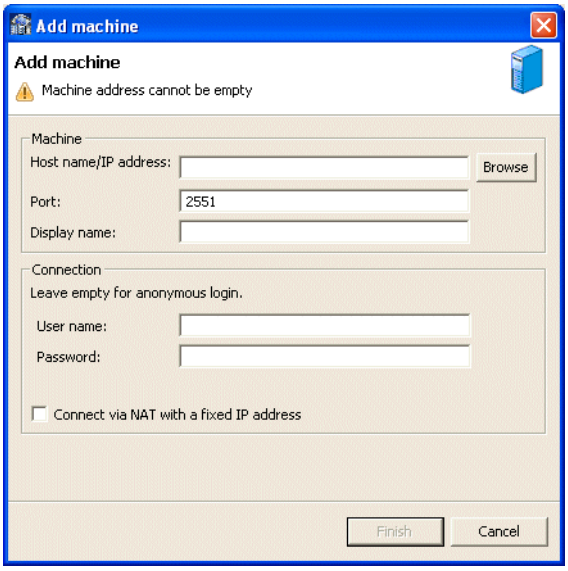
- Setting Up a Windows Computer in Oracle Studio
- Securing Access to Oracle Connect
- Setting Up a Change Data Capture with the OracleAS CDC Adapter for SQL Server
- Testing the OracleAS CDC Adapter for SQL Server CDC Solution
- Handling Metadata Changes
- Environment Verification

Setting Up a Windows Computer in Oracle Studio

Using Oracle Studio, perform the following steps to configure the Windows computer:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens, displaying the Design perspective.
2. Right-click **Machines** in the Configuration Explorer and select **Add Machine**. The Add Machine screen is displayed.

Figure 3–1 The Add Machine screen



3. Enter the name of the computer you want to connect to, or click **Browse** to select the computer from the list of computers that is displayed and which use the default port (2551).
4. Specify the username and password of the user who was specified as the administrator when Oracle Connect was installed.

> **Note:** Selecting **Anonymous connection** enables anyone having access to the computer to be an administrator, if this was defined for the computer.

5. Click **Finish**.

   The computer is displayed in the Configuration Explorer.

Securing Access to Oracle Connect

Oracle Studio includes mechanisms to secure access to Oracle Connect both during modeling and at run time.

During modeling, the following security mechanisms can be applied:

- Setting Password Access to Oracle Studio
- Specifying Users with Administrative Rights

At run time client access to Oracle Connect is provided by the user profile:

- Setting Up Run-Time User Access

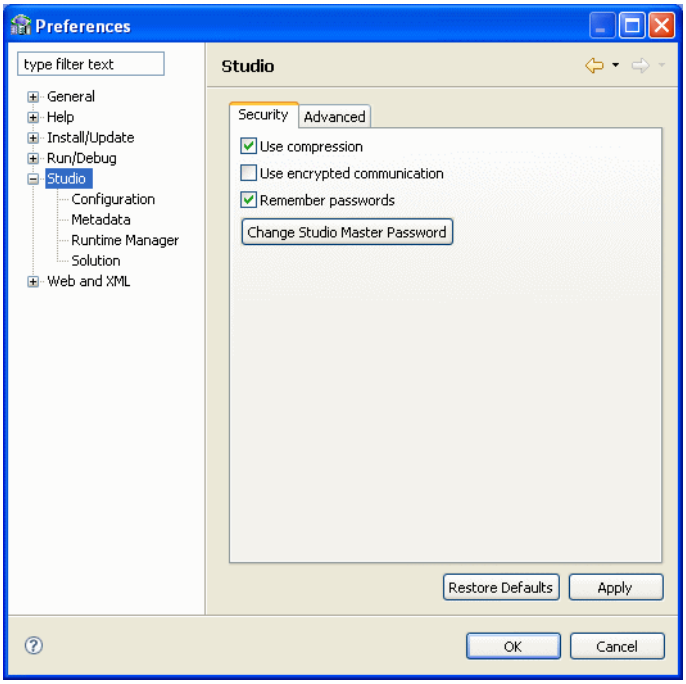
Setting Password Access to Oracle Studio

Initially, any operation performed using Oracle Studio does not require a password. You can set a password so that the first operation that involves accessing the server from Oracle Studio requires a password to be entered.

Perform the following steps to set password access to Oracle Studio:

1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens.
2. Select **Window** from the menu bar, and then select **Preferences**. The Preferences screen is displayed.
3. Click **Studio** in the left pane as shown in the following figure:

Figure 3–2 The Preferences screen



4. Click **Change Studio Master Password**. The Change Master Password screen is displayed, as shown in the following figure:

Figure 3–3 The Change Master Password Screen

5. Leave the **Enter current master password** field blank and type a new master password in the **Enter new master password field**.
6. Enter the new passoword again in the **Confirm new master password** field.
7. Click **OK**.

Specifying Users with Administrative Rights

By default, only the user who was specified during the installation as an administrator has the authorization to modify settings on that computer from Oracle Studio. This user can then authorize other users to make changes or to view the definitions for a selected computer. Adding a computer to Oracle Studio is described in "Setting Up a Windows Computer in Oracle Studio".

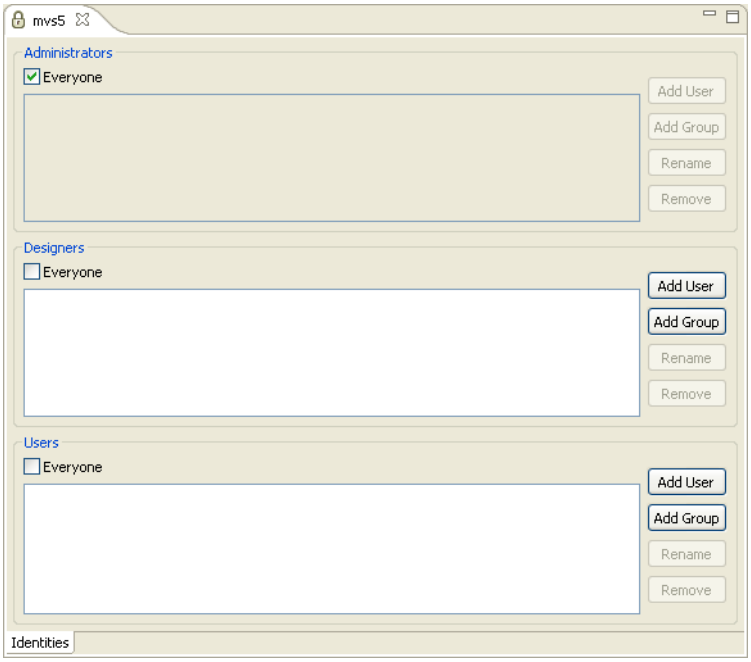
Note: The default during installation is to enable all users to be administrators.

---

1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens.
2. In the Design perspective Configuration view, Right-click the computer and select **Administration Authorization**.

   The Administration Authorization screen is displayed as shown in the following figure:

Figure 3–4 The Administration Authorization screen



The screen has the following sections:

Administrators: Administrators can view and modify all the definitions in Oracle Studio for the selected computer. On initial entry to Oracle Studio, every user is defined as a system administrator.

Designers: Designers can view all the definitions for the computer in Oracle Studio and can modify any of the definitions under the **Bindings** and **Users** nodes for the selected computer. For example, Oracle Studio database administrator can add new data sources and adapters and can change metadata definitions for a table in a data source.

Users: Users can view all the definitions for the computer in Oracle Studio for the selected computer. Regular users cannot modify any of the definitions.

3. Add users or groups of users by clicking **Add User** or **Add Group** for the relevant sections.

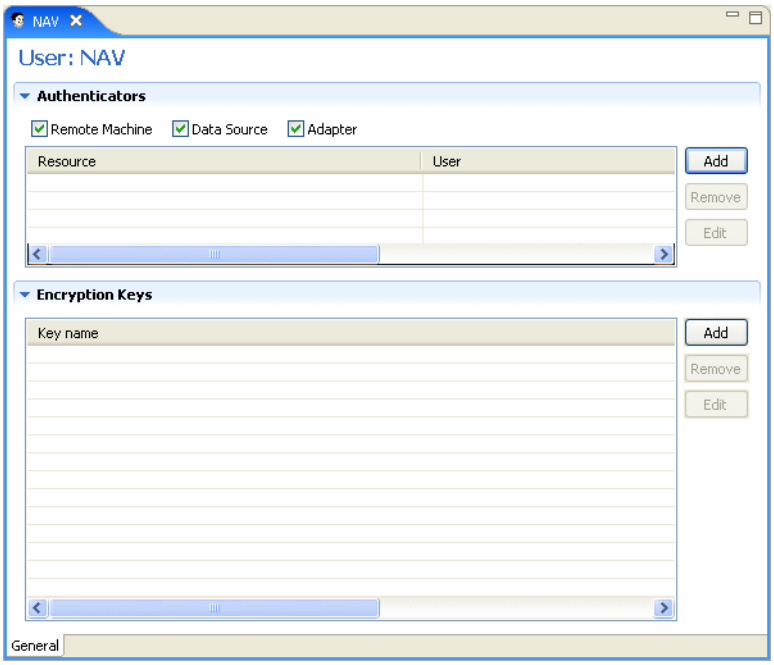
   The user or group that is added must be recognized as a valid user or group for the computer. Once a name has been added to a section, only the user or group who logs on with that user name has the relevant authorization.

Setting Up Run-Time User Access

During run time, client access to Oracle Connect is provided by the user profile. A user profile contains name and password pairs that are used to access a computer, data source or application at run time, when anonymous access is not allowed.

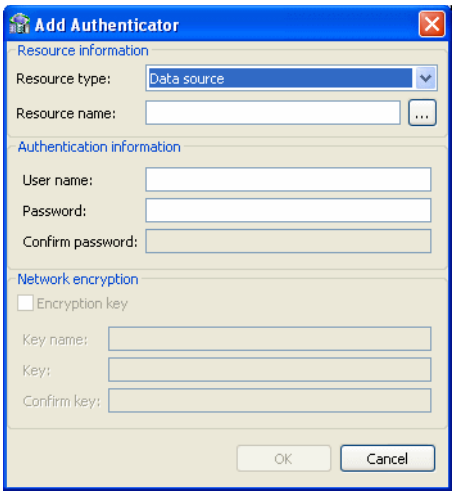
1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**. Oracle Studio opens.
2. From the Design perspective, Configuration view, expand the **Machines** folder, then expand the machine where you want to set the user name and password.
3. Expand **Users**.
4. Right-click the **NAV** user profile and select **Open**. The NAV user profile editor is displayed:

Figure 3–5 The User Editor



5. In the User editor, click **Add**. The Add Authenticator screen is displayed:

Figure 3–6 The Add Authenticator screen



6. Select **Remote Machine** from the **Resource Type** list.
7. Enter the name of the Windows computer defined in Oracle Studio.
8. Enter the name and password used to access the computer and confirm the password.
9. Click **OK**.

Setting Up a Change Data Capture with the OracleAS CDC Adapter for SQL Server

You must set up the Oracle Connect SQL Server CDC adapter on a Windows computer to handle capture changes to the SQL Server data. To work with the SQL Server CDC adapter, you must various tasks on the computer with the SQL database and then configure the change data capture using the Oracle Studio CDC Solution perspective. Oracle Studio can be installed on a Windows or UNIX computer.

Perform the following steps to setup the change data capture and configure the CDC adapter:

- Enabling SQL Replication
- Configuring Security Properties
- Setting up Log On Information
- Setting up the Database
- Setting Up the TLOG Miner (LGR)
- Setting up a Change Data Capture in Oracle Studio
- Deploying a Change Data Capture

Enabling SQL Replication

When logging an `UPDATE` operation, the MS SQL Server records only changed data to the TLOG. This is not enough information for the CDC agent to provide before and after images for `UPDATE` statements. There is not enough information to provide the values for changed columns with primary keys, which is the minimum requirement for a CDC agent.

To solve this problem, you must turn Replication on in the MS SQL Server. The log will be able to report update changes as usual. In addition, Before Image results are also supported in this mode. Replication is valid only for tables with primary keys. Therefore, the MS SQL Server CDC Agent works only with tables that have a primary key.

The Microsoft Replication solution used by Oracle's MS SQL Server CDC must be enabled by a qualified system administrator. The system administrator must use the tools provided with the MS SQL Server to enable replication.

The following sections explain how to set up replication for SQL Server 2000 and SQL Server 2005.

SQL Server 2000 Replication

In SQL Server 2000, open the SQL Server's Publishing wizard in the Microsoft SQL Server's Enterprise Manager. and follow the instructions provided by the wizard or see the SQL Server documentation.

The following should be added to the database's definitions:

- A new Distribution database
- A replication entry
- A replication monitor entry

SQL Server 2005 Replication

In SQL Server 2005, in the Microsoft SQL Server's Management Studio, follow the instructions provided by the Configure Distribution wizard to set up replication or see the SQL Server documentation.

To open the wizard from Microsoft SQL Server 2005:

- In the Microsoft SQL Server Management Studio, right-click the Replication folder and select **Configure Distribution**.

  The Configure Distribution wizard opens.

You should make the following selections in the wizard:

- In the Distributor step, select **<SQL Server Name> will act as its own distributor; SQL Server will create a distribution database and log**
- In the SQL Server Agent Start step, select **Yes, configure the SQL Server agent to start automatically**

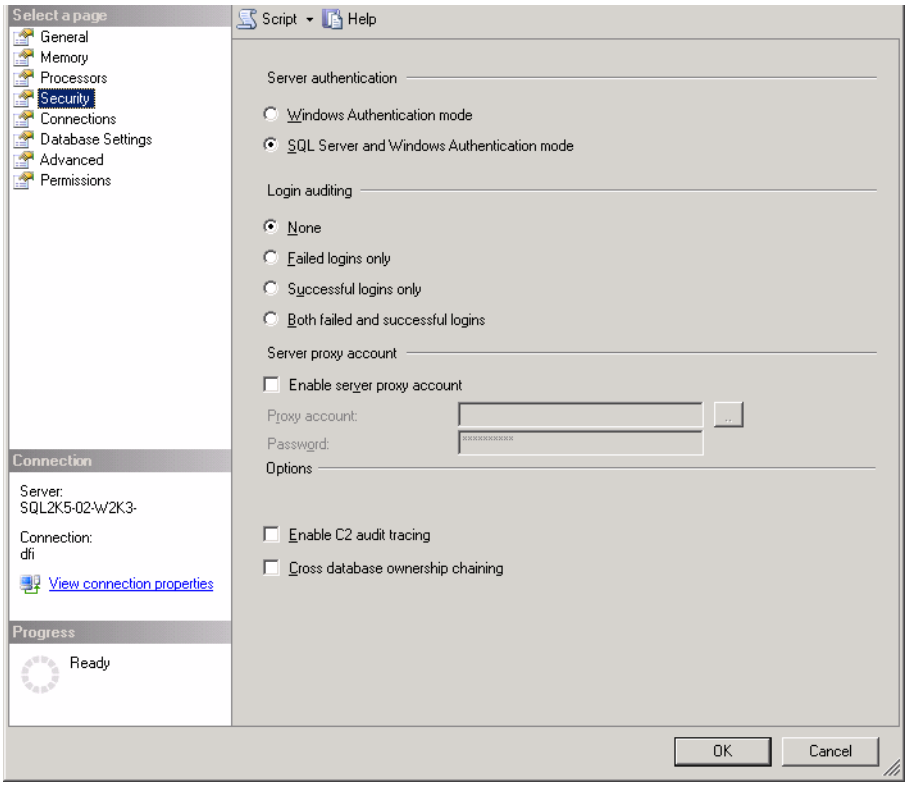
Configuring Security Properties

To ensure that the security requirements for the OracleAS SQL server CDC adapter are met, do the following:

- Set up an account with administrator rights to run the SQL Server andMS SQL Server CDC components.
- Ensure that the OracleAS SQL Server CDC adapter and the TLOG Miner (LGR) are executed as members of the `sysadmin` server role.

In addition, set the following security properties from the SQL Server Properties dialog box. The following figure shows the Security tab. The example below may look different on your machine depending on the version of SQL Server you are using. The following example is from SQL Server 2005.

Figure 3–7 Security Settings



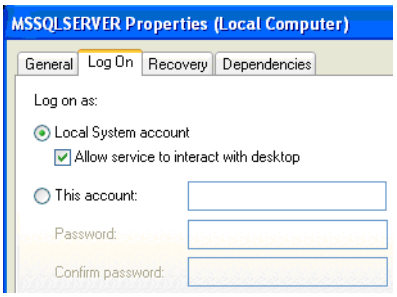
- Set the Authentication settings to **SQL Server and Windows**.
- Set the Audit level to **None**.

Setting up Log On Information

The SQL Service Login account information must match the configuration information for the Oracle IRPCD and Oracle LGR services. The logon setups should be entered in a way that allows the Oracle Services to access the SQL Server database. In most cases, the services log on at the Local System account. You should enter the following information in the SQL Server (MSSQLSERVER) Properties Log On tab. You access this through the Windows services control panel. The following describes how to set up the log on information:

1. From the Windows **Start** menu, select **Control Panel**.
2. Double-click **Administrative Tools**.
3. Double-click **Services**. The Services control panel is displayed.
4. From the Services list, right-click **SQL Server (MSSQLSERVER)** and select Properties.
5. Configure the system as shown in the figure below.

Figure 3–8 Log On Properties



The configuration shown is the default for the IRPCD and LGR services, which allows anonymous access.

If you want the SQL Server (MSSQLSERVER) to Log on using Windows authentication, the system administrator must enter the correct settings to log On to accounts for Oracle services.

Setting up the Database

You must make sure that some of the database set up and configurations are set so that the OracleAS CDC Adapter for SQL Server will consume the changes made to the database. This section describes the properties that must be set for the correct operation of the adapter.

SQL Server 2000 Settings

Set the following properties in the SQL Server Enterprise Manager.

- In the database properties **Options** tab, set the Recovery Model to **Full**. In this mode, the transaction Log is more durable and truncation occurs less frequently.
- Create enough log space to handle the size of the published database.
- In the database properties **Transaction Log** tab, select the correct setting for **File Growth** based on the application's capacity profile.
- Set the `trunc. log on chkpt` property to `FALSE`. To set this property, enter the following in the SQL Query Analyzer:

  ```
  EXEC sp_dboption '<database name>', 'trunc. log on chkpt.', 'FALSE'
  ```
- Make sure that all tables that will be consumed by the SQL Server CDC have a primary key.

> **Note:** See the documentation provided with Microsoft SQL Server for information on how to set the above properties correctly.

SQL Server 2005 Settings

Set the following properties in the SQL Server Management Studio.

- From the Object Explorer, right click the database and select **Properties**. In the **Options** tab, set the **Recovery model** to **Full**. In this mode, the transaction Log is more durable and truncation occurs less frequently.
- Create enough log space to handle the size of the published database.

- From the Object Explorer, right click the database and select **Properties**. In the Files tab, set the initial size and growth parameters for the log files based on the application's capacity profile.
- Set the `trunc. log on chkpt` property to `FALSE`. To set this property, run the following query:

  ```
  EXEC sp_dboption '<database name>', 'trunc. log on chkpt.', 'FALSE'
  ```
- Make sure that all tables that will be consumed by the SQL Server CDC have a primary key.

> **Note:** See the documentation provided with Microsoft SQL Server for information on how to set the above properties correctly.

Setting Up the TLOG Miner (LGR)

SQL Server handles logs in a way that is not fully compatible with standard CDC solutions. For example, the MS SQL Server will truncate a TLOG after a period of inactivity to make more space available for logging operations. Uncontrolled LOG truncation could cause a loss of the truncated data. To solve these problems, the OracleAS CDC Adapter for SQL Server uses a TLOG miner. This component is initiated as a Microsoft Windows service. It mines the data and sends it to a Transient Storage area. The OracleAS Adapter for SQL Server uses the data in Transient Storage to consume changes.

The Log reader (LGR) is the component that actually reads the MSQL transaction LOG. All logged data that is affected by MS replication is read and placed at the transient storage folder. It is implemented as an independent standalone Windows Service. Since its functionality is highly sensitive, it has high availability features and fault tolerance and attempts to be always up.

The following sections describe the procedures necessary for setting up the TLOG Miner service:

- Call the LGR Service Interface
- Configuring the Template Input File
- Registering the TLOG Miner (LGR) Service
- Setting the Recovery Policy

Call the LGR Service Interface

You must call the service interface. Enter the following command at the service command prompt to call the service interface.

```
>>>sqlcdclgr -?
```

The service interface is displayed. The service interface shows commands that you can use. The following is an example of the service interface that is displayed.

```
SQLCDCLGR Transaction LOG mining service controller:
----------------------------------------------------
sqlcdclgr -s register -a <service-name> <input-file> Register a service and its
input file
sqlcdclgr -s unregister -a <service-name> Unregister a service
sqlcdclgr -s start -a <service-name> Start service execution
sqlcdclgr -s stop -a <service-name> Stop service execution
sqlcdclgr -s restart -a <service-name> Restart service execution (=refresh
parameters)
sqlcdclgr -p name <service-name> Display input file P_arameter name registered for
```

```
a service
sqlcdclgr -p contents <service-name> Display input file P_arameter contents
registered for a service
sqlcdclgr -p help Display help for parameters values assignment
sqlcdclgr -t T_ype an input file template
sqlcdclgr -b <input-file> Run the service in an online 'B_locking' mode, using
input file
sqlcdclgr [-h|-?] Display this H_elp banner

Service input is held at: HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\Oracle
Connect\Services
```

Configuring the Template Input File

The configuration template defines some basic configuration parameters. You must define some of these parameters manually in the template. Generate the template and then edit the parameters. Enter the following at the command prompt to generate the configuration template.

```
<your drive>:\<full path>\sqlcdclgr>sqlcdclgr -t >sqlcdclgr_pars.xml
```

The following is an example of the configuration template that opens.

```
<serviceConfig>
<cdcOrigin server='?xxx?' database='?xxx?' user='sa' password=''defaultOwner=''/>
password=' ' useWindowsAuthentication='false' defaultOwner='dbo'/>
<transientStorage directory='?xxx?'maxFileSize='1' totalSize='100'
                                lowThreshold='65' highThreshold='85'/>

<logging directory='?xxx?'/>

<control batchSize='50000' retryInterval='1' debugLevel='none'
traceDBCC='false' traceStatistics='false'/>

<detainer detainingTimeInterval='300' detainerTxnDurationLimit='2147483647'
traceActivity='false'/>
</serviceConfig>
```

You must enter the correct values for some of the parameters in this file. These parameters are shown as placeholders ?*xxx*? in the example above. Enter the current information for your system where the placeholders are shown. The following table describes the parameters to be changed.

Table 3–1 Configuration Parameters

Property	Parameter	Parameter Description
`Origin`	`database`	Enter the name of the SQL server database you are using. The name given for the database must be in the same literal form as the name given to the **dbName** when you Set up the Data Source.
	`server`	Enter the name of the server machine where SQL Server is installed. The name given for the server must be in the same literal form as the name given to the **SQL Server Name** when you Set up the Data Source.

Table 3–1 (Cont.) Configuration Parameters

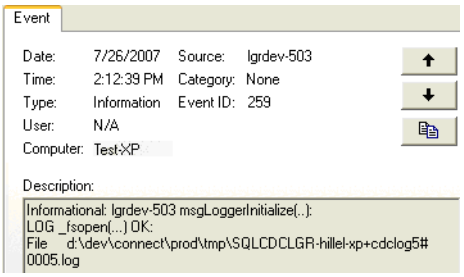
Property	Parameter	Parameter Description
	`user`	Enter the name of the authorized user for the server. **Note**: The user entered must have `sysadmin` permissions in the SQL Server database.
	`password`	Enter the password for the user entered in the User parameter.
	`useWindowsAuthentication`	The default value for this property is `false`. Change this property to `true` if you want to use Windows authentication. In this case, when you start the LGR service you do not need to provide credentials to sign in to SQL Server.
`transientStorage`	`directory`	Enter the full path to the directory where the transient storage files are located.
	`maxFileSize`	The maximum size (in MB) allowed for a single transient storage file. For this parameter, you can change the default value.[a]
	`totalSize`	The maximum size (in MB) allowed for all of the transient storage. For this parameter, you can change the default value.[a]
	`lowThreshold`	For this parameter, you can change the default value.[a]
	`highthreshold`	For this parameter, you can change the default value.
`logging`	`directory`	Enter the full path to the directory where he log files are located. LOG files are named by adding the leading prefix, SQLCDCLGR, then the server machine identifier and the database name. An example of an LGR file name is: `SQLCDCLGR-192_168_165_167+CDClog5#0002.log` You can view the information about the log file for an LGR instance in the Windows Event Properties dialog box. 
control	`batchSize`	The limit of the batch size for records being read upon a single LGR scan pass.

Table 3–1 (Cont.) Configuration Parameters

Property	Parameter	Parameter Description
	`retryInterval`	The time interval in seconds that the system waits for additional information when the number of lines is less than defined in the `batch` parameter.
	`debugLevel`	Set the level for the debugging log.
	`traceDBCC`	Set `true` or `false` to determine if trace information is returned to the log.
	`traceStatistics`	Set `true` or `false` to determine if tracing statistics are returned to the log.
detainer	`detainingTimeInterval`	Set the amount of time is seconds that data is held in the TLOG before it can be truncated. You can change the default for this parameter.
	`detainerTxnDuratinLimit`	This is not an active parameter.
	`traceActivity`	Set true or false to determine whether to trace the detainer activity and send the information to the log.

[a] Transient storage management is space oriented. It is based on a maximum allowed allocated space (default:100MB) and upper/lower thresholds. After every LOG scan, the LGR checks whether the transient storage space is close to exceeding its upper threshold (by default it checks to see if the storage space it at 85% or more). If it is close to exceeding the maximum space, a cleanup is started. The cleanup reduces the occupied space to the `lowThreshold` parameter (default 65%) of the total size specified. Cleanup activity is always reported in the LGR log file for all debug/trace settings.

Notes:

The `-p help` option displays a list of these parameters and an explanation for each.

All paths must be fully qualified. You cannot use logical names.

Registering the TLOG Miner (LGR) Service

Do the following to register the LGR service:

1. Provide a name for the service. You should use the same name as the name of the database that you are using.
2. Register the service by entering the following at the system prompt:

```
C:\Program Files\Oracle\Connect\tmp>sqlcdclgr -s register -a <service name>
C:\Program Files\Oracle\Connect\def\sqlcdclgr_pars.xml
```

Note: You must enter the *full path* to the configuration template file as the last parameter, as shown above.

The following is an example of the system response:

```
| SQLCDCLGR Transaction LOG mining feature.
| Associated program is : C:\Program Files\Oracle\Connect\BIN\sqlcdclgr.exe
+-----------

Install(): Service 'SQLCDC' installed
```

```
setServiceParameter(): Parameter 'C:\Program
Files\Oracle\Connect\def\sqlcdclgr_pars.xml' has been set for Service 'SQLCDC'
at HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\Oracle Connect\Services
addEventSource(): Key (+values) added : HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Application\SQLCDC
```

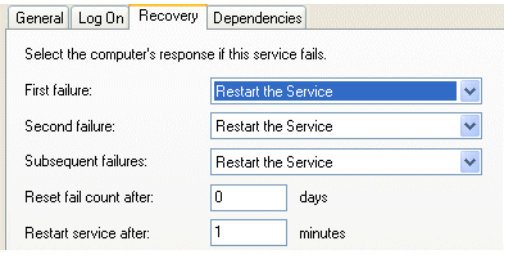
Setting the Recovery Policy

You must also set the Recovery policy for the service. The Recovery Policy is set in the Service Properties. Follow these steps to set the Recovery Policy for the new TLOG Miner (LGR) service:

Do the following to set the recovery policy:

1. From the Windows **Start** menu, select **Control Panel, Administrative Tools** and double-click the **Services** icon.
2. In the Windows Services control Panel, right click your new TLOG Miner service and select **Properties**.
3. In the Properties screen, click the **Recovery** tab.
4. Select the following computer response for each failure:
   - First failure: **Restart the Service**
   - Second failure: **Restart the Service**
   - Subsequent failures: **Restart the Service**

Figure 3–9 Recovery Tab



Setting up a Change Data Capture in Oracle Studio

You set up the change data capture in Oracle Studio. Oracle Studio can be installed on Windows XP or Vista operating systems, or on UNIX. For more information, see Installing Oracle Studio.

A change data capture is defined in the CDC Solution perspective, which contains a series of links to guide you through the CDC set up process. The CDC solution perspective guides display the following symbols in front of a link to show you what tasks should be done, and what tasks were completed.

- Triangle: This indicates that there are subtasks associated with this link. When you click the link, the list expands to display the subtasks.
- Asterisk (*): This indicates that you should click that link and carry out the tasks and any subtasks presented. If more than one link has an asterisk, you can carry out the marked tasks in any order.
- Check mark (✓): This indicates that the tasks for this link and any sublink are complete. You can double click the link to edit the configuration at any time.
- Exclamation mark (!): This indicates a potential validation error.

Carry out the Following to set up a change data capture:

- Create a CDC Project
- Set up the CDC Server
- Set up the Staging Area Server

Create a CDC Project

Do the following to create a CDC Project

1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**.
2. Open the CDC Solution perspective, click the **Perspective** button on the perspective toolbar and select **CDC Solution** from the list.

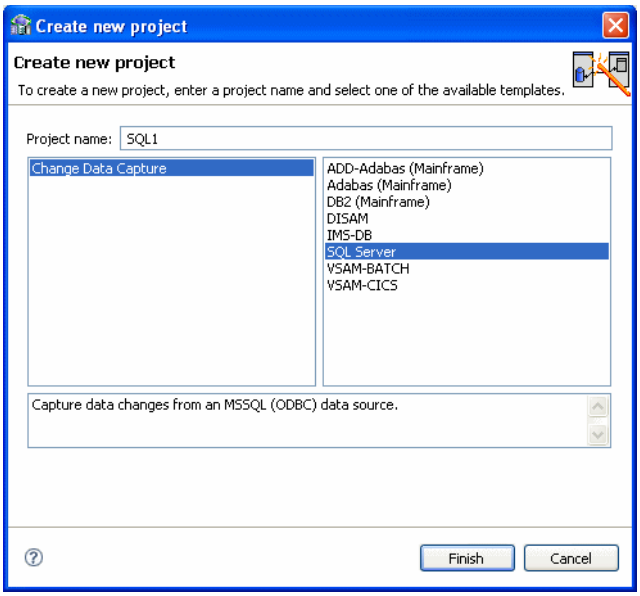
   The CDC Solution perspective opens with the Getting Started guide in the left pane of the workbench.
3. Click **Create new project**.

   The Create new project screen opens.
4. In the **Project name** field, enter a name for your project.

   The types of projects available are listed in the left pane just below.
5. Select **Change Data Capture**.

   From the the right pane, select **SQL Server**.

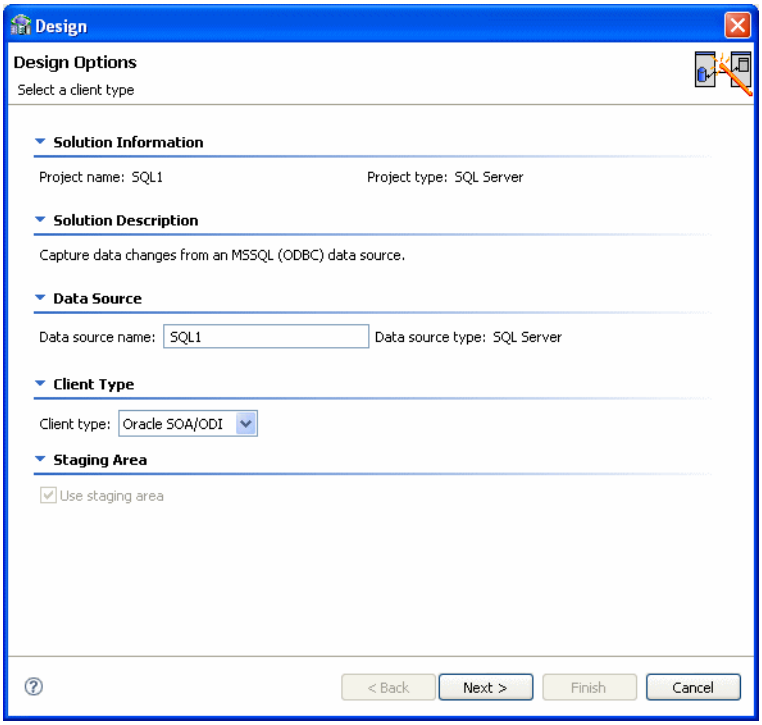
Figure 3–10 Create New Project



6. Click **Finish**. The Project Overview guide is displayed in the left pane.
7. Click **Design**. The Design wizard opens. Use this wizard to enter the basic settings for your project.

Note: The wizard screens are divided into sections. Some sections provide information only and other sections let you to enter information about the project. If you do not see any information or fields for entering information, click the triangle next to the section name to expand the section.

Figure 3–11 Design Wizard (Design Options)



8. In the **Client Type** you can select **Oracle SOA/ODI** only. The **Use staging area** is selected and cannot be changed, you must use a staging area with the OracleAS CDC Adapter for SQL Server.

9. Click **Next**.

   The Design Wizard's second screen is displayed. In this step you configure the machines used in your solution. Enter the information for the following machines:

   - **Server Machine Details**: Information about the machine where Oracle Connect is installed. The selection here is always **Server Machine** and **Microsoft Windows**.

   - **Staging Area Details**: Information about the machine platform where the staging area is located.

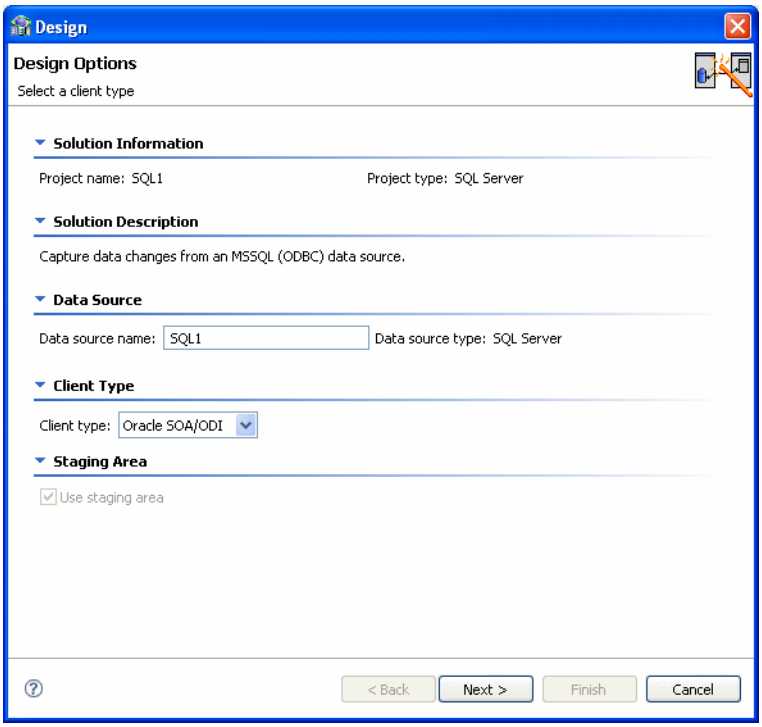
     In the **Name** list, you should select **Server Machine**. This will create the Staging Area on the same machine specified as the Server Machine. Although this is not the default selection, and you can select a different machine, for guaranteed delivery reasons, it is recommended to use a One Machine solution when working with the OracleAs SQL Server CDC adapter.

     In the **Platform** list, select the operating system for the staging area. This can be Windows, Linux or UNIX. The available options are:

     – **Microsoft Windows**

- **HP-UX**
- **IBM AIX**
- **Sun Solaris**
- **Linux** (Red Hat)
- **Suse** (Linux)

Figure 3–12 Design Wizard (Configure Solution Machines)



10. Click **Finish**. The wizard closes.

Set up the CDC Server

Click **Implement** in the **Getting Started** guide to open the Implementation guide.

In the Implementation guide, do the following to set up the CDC server:

- Set up the Machine
- Set up the Data Source
- Set up the CDC Service

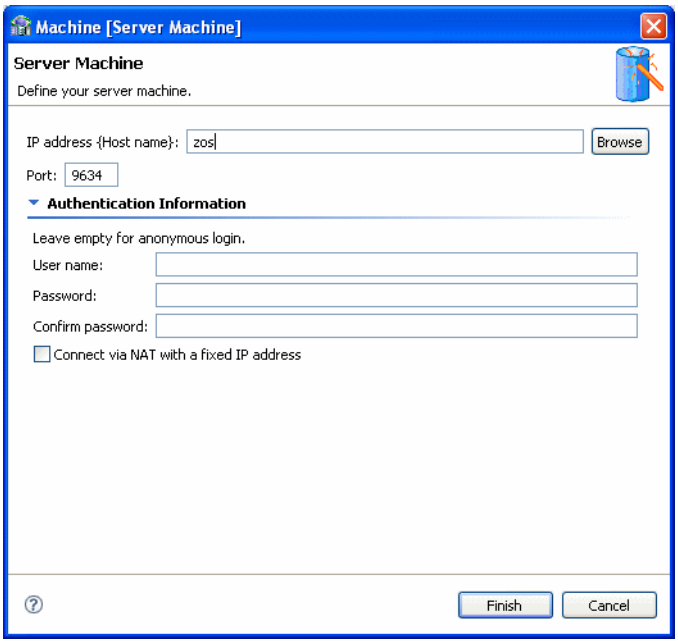
Set up the Machine

You do the following to define the IP Address/host name and Port for the CDC server machine.

1. Click **Machine**.

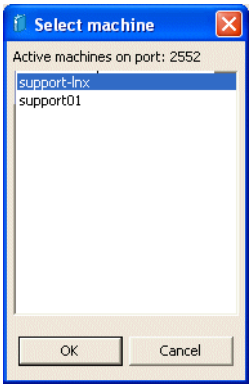
The machine definition screen is displayed:

Figure 3–13 Machine Definition



2. In the **IP address/host name** field, do *one* of the following:
   - Enter the server machine's numeric IP address.
   - Click the **Browse** button and select the host machine from the ones presented, then click **Finish**.

Figure 3–14 Select Machine



Note: The machine you enter must be compatible with the platform designated in the Design Wizard (Configure Solution Machines) screen.

---

3. Enter the port number.

   The default port number is **2551**.

4. If you want to connect with user authentication, enter a user name and password, with confirmation, in the Authentication Information area.

5. Select the **Connect via NAT with a fixed IP address** checkbox if you are using Network Access Translation and want to always used a fixed IP address for this machine.
6. Click **OK**.

Continue setting up the CDC Server on the Windows computer.

Set up the Data Source

In this step, set up the SQL Server datasource that you are using. Enter the following information in this step:

- **SQL Server Name**: Enter the name of the SQL server database you are using.

  The name given for the database must be in the same literal form as the name given to the `database` parameter when Configuring the Template Input File.
- **dbName**: Enter the name of the server machine where the SQL Server is installed.

  The name given for the server must be in the same literal form as the name given to the `Server` parameter when Configuring the Template Input File.

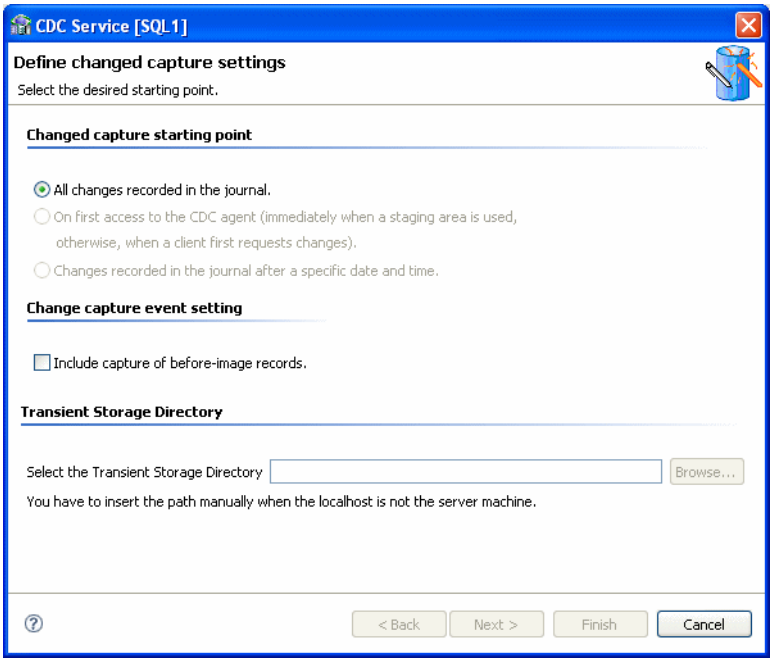
Continue setting up the CDC Server.

Set up the CDC Service

In this step you define the starting point or event for the change capture and then indicate the name of the change logger. Do the following to set up the CDC service.

1. In the Solution perspective, click **Implement**.
2. In the Server Configuration section, click **CDC Service**. The CDC Service wizard is displayed. The **Define changed capture settings** is the first step. It has three sections.

Figure 3–15 Define changed Capture Settings



3. In the **Change capture starting point** section, select *one* of the following to determine the Change Capture starting point:

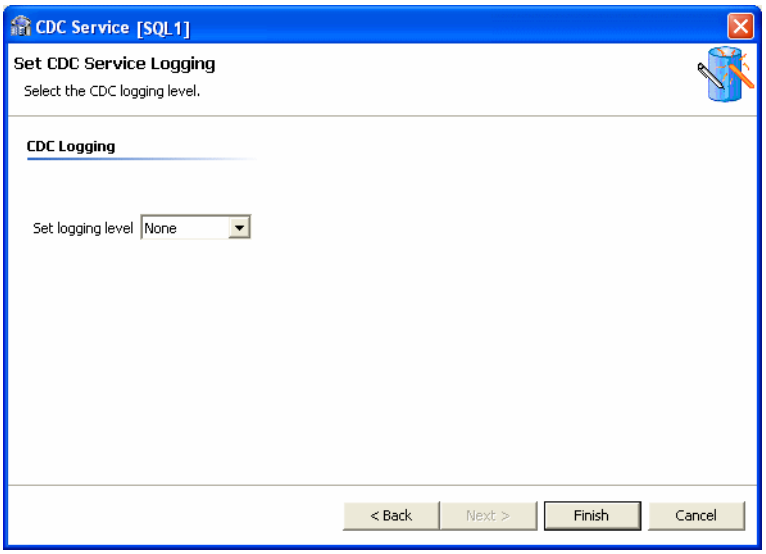
- **All changes recorded to the journal**
- **On first access to the CDC (immediately when a staging area is used, otherwise, when a client first requests changes**
- **Changes recorded in the journal after a specific date and time**.

  When you select this option, click **Set time**, and select the time and date from the dialog box that is displayed.

4. In the **Change capture envent setting** section, Select the **Include capture of before image records** if you want to include these records in the CDC solution.
5. In the **Transient Storage Directory** section, Enter the path to the folder where the transient storage is located. This should be the same location as defined in the **transientStorage** parameter when Configuring the Template Input File.

   The transient storage should be on the same machine where the OracleAS CDC Adapter for SQL Server is defined.
6. Click **Next** to go to the next step where you set the CDC Service Logging. Select the log level to use from the **Logging level** list.

Figure 3–16 Logging Level



Select *one* of the following from the list:

- None
- API
- Debug
- Info
- Internal Calls

7. Click **Finish**.

Continue setting up the CDC Server on the Windows computer.

Set up the Staging Area Server

Click **Implement** in the **Getting Started** guide to open the Implementation guide.

In the Implementation guide, do the following under the **Stream Service Configuration** section, to set up the staging area server:

- Set Up the Staging Area Machine
- Set up the Stream Service
- Configure the Access Service Manager

Set Up the Staging Area Machine

To set up the machine for the staging area, do the following.

1. Under the Stream Service Configuration section, click **Machine**.
2. Use the same configurations used to Set up the Machine for the CDC server.

Continue setting up the staging area Server.

Set up the Stream Service

In this step you set up the stream service. The Stream Service configures the following:

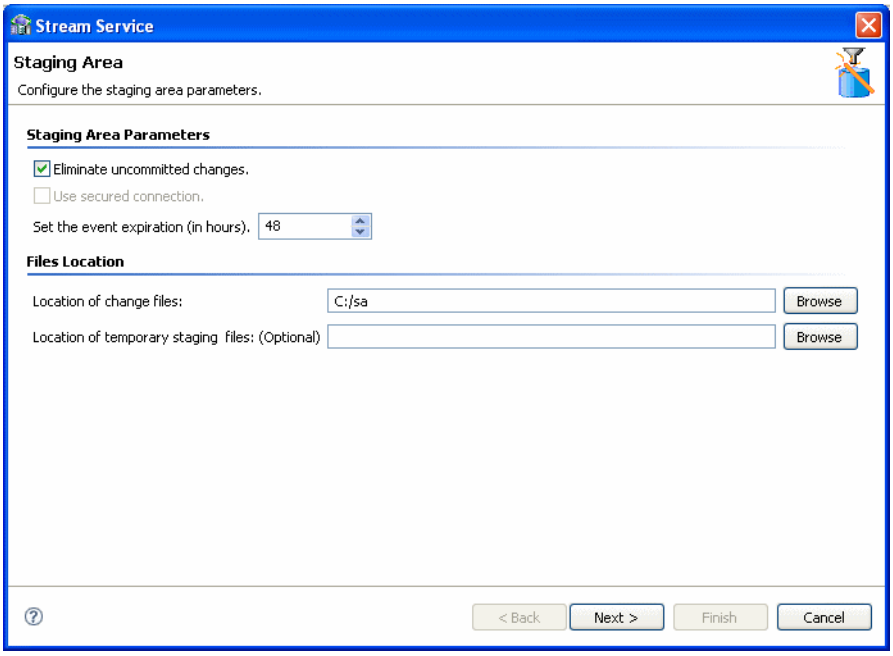
- Staging area
- Filtering of changed columns
- Auditing

Note: Null filtering is currently unsupported. Filtering empty values is supported. Space values are truncated and are handled as empty values.

---

1. Click **Stream Service**. The Stream Service wizard opens.

Figure 3–17 Staging Area

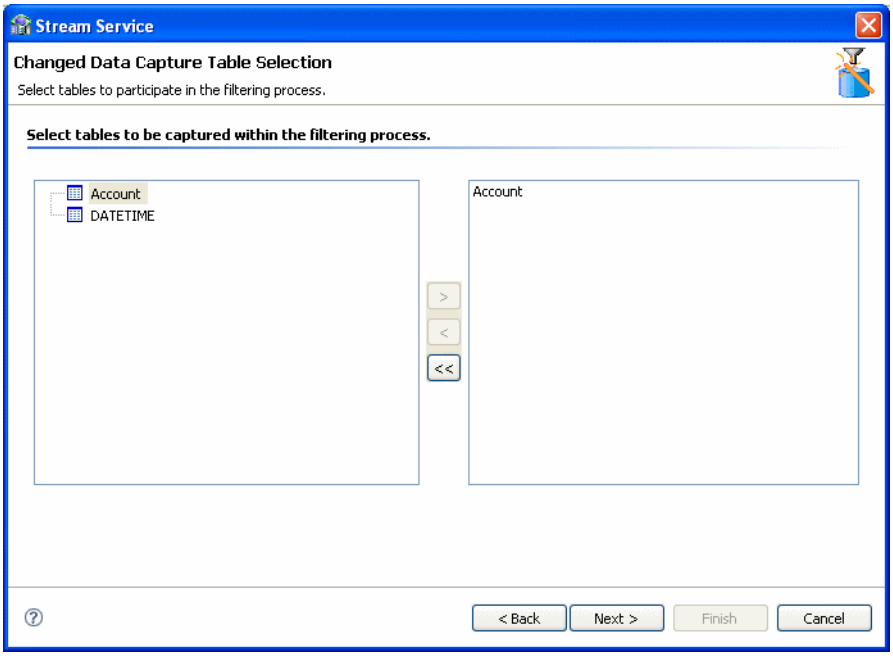


Note: This screen will only appear if you selected the inclusion of a staging area in your solution.

---

2. You can configure the following parameters in this screen:
   - Select **Eliminate uncommitted changes** to eliminate uncommitted changes from your CDC project.
   - Select the **Use secured connection** check box to configure the staging area to have a secured connection to the server. This is available only if you logged into the server using user name and password authentication.
   - Set the event expiration time in hours.
   - Under File Locations, click the **Browse** buttons to select the location of the changed files, and temporary staging files, if necessary.
3. Click **Next** to select the tables to include in the filtering process.

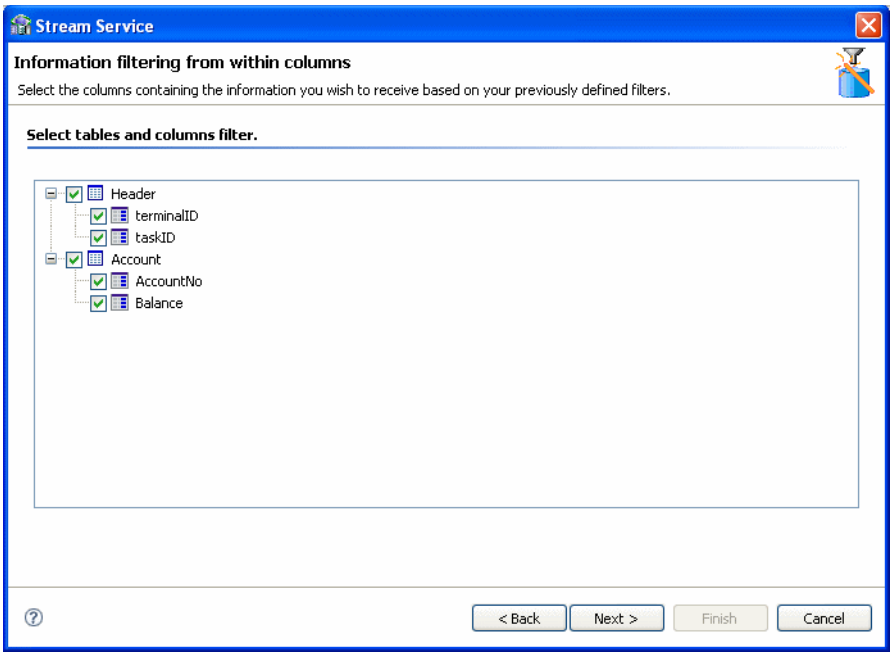
Figure 3–18 Select Tables



4. Click the required tables in the left pane and move them to the right pane using the arrow keys.

   Note: You can remove the tables and add new ones to be captured after you have already added the tables to the right pane. For more information, see Adding and Removing Tables.
5. Click **Next**. From the tables selected above, select the columns that receive changes. Select the check box next to the table to use all columns in the table.

Figure 3–19 Column Selection

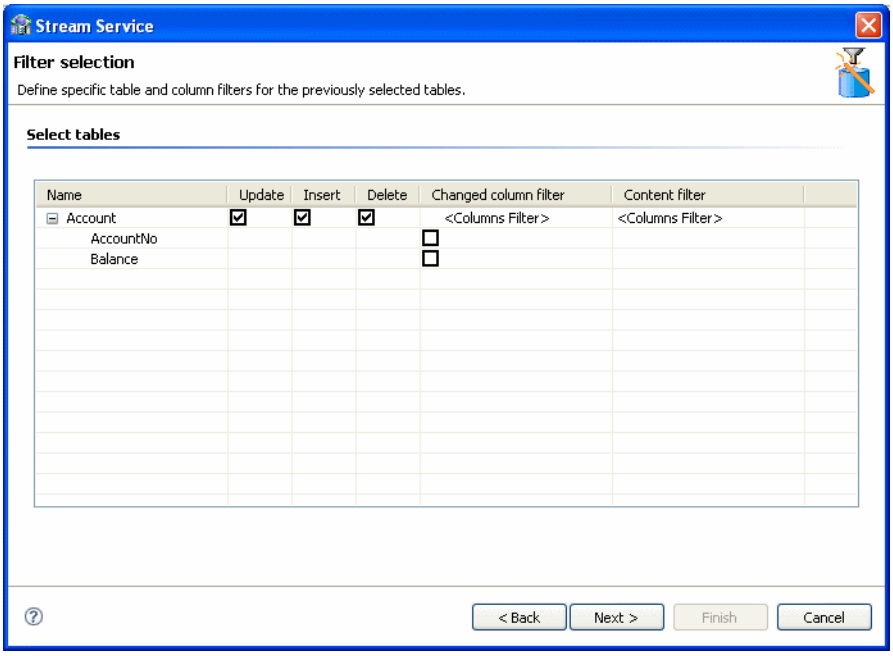


Note: Table headers will appear grouped together in a separate table at the beginning of the list. You can also request the receipt of changes in the headers' columns.

Any data changes in the columns selected will be recorded.

6. Click **Next**. The Filter selection screen is displayed. the types of changes you want to receive in the tables and which columns to display.

Figure 3–20 Filter Selection



7. You can do the following in this screen:

   Select the actions from which you want to receive change information:

   - Update
   - Insert
   - Delete

   Note: These items are all selected by default.

   - Under the Changed Columns Filter column, select the columns for which you want to receive notification of changes.

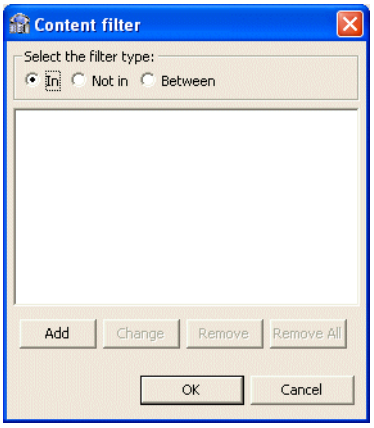
   Notes:

   - If you do not select a column, you will receive notification of *all* changes.
   - If you select only one, you will receive change information only if the field selected undergoes a change.
   - If you select more than one, but not all, then you will receive change information only if any or all of the selected fields undergo a change

8. In the **Content Filter** column of the Filter screen, double-click a table column and then click the **Browse** button to filter content from the selected column.

   The Content Filter screen is displayed.

Figure 3–21 Content Filter



9. Select a filter type:

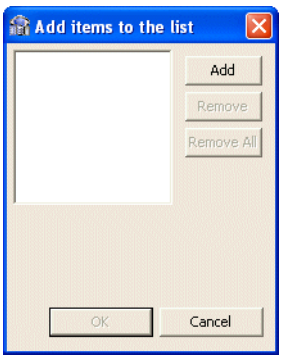
   - Select **In** for events to be returned where the relevant column value equals the values you specify (if a column is NULL, it is not captured).
   - Select **Not In** for events to be returned where the column value is *not* in the values you specify (if the column is NULL, it is captured).
   - Select **Between** for when the column value is between the two values you specify (if a column is NULL, it is not captured).

10. Click **Add** in the lower-left corner of the Content Filter screen.

> **Note:** If you select more than one condition, you will receive the change information as long as one of the conditions is true.

11. Depending on your selection, do *one* of the following:
    - If you selected **In/Not In**, continue with step12.
    - If you selected **Between**, continue with step 14.
12. Click **Add** in the Add items to the list screen. Enter a value for events to be returned where the column value appears (or does not appear) in that value. To filter empty values (”) for the **Not In** filter type, leave this field blank in the dialog box that is displayed.

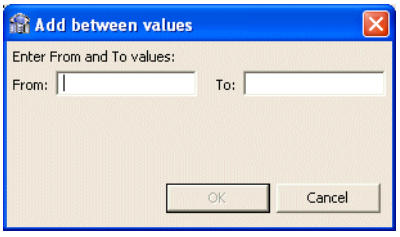
Figure 3–22 Add Items (In or Not In)



13. Repeat steps 12 as many times as necessary, and then proceed to step 16.
14. Click **Add** in the Add items to list screen.

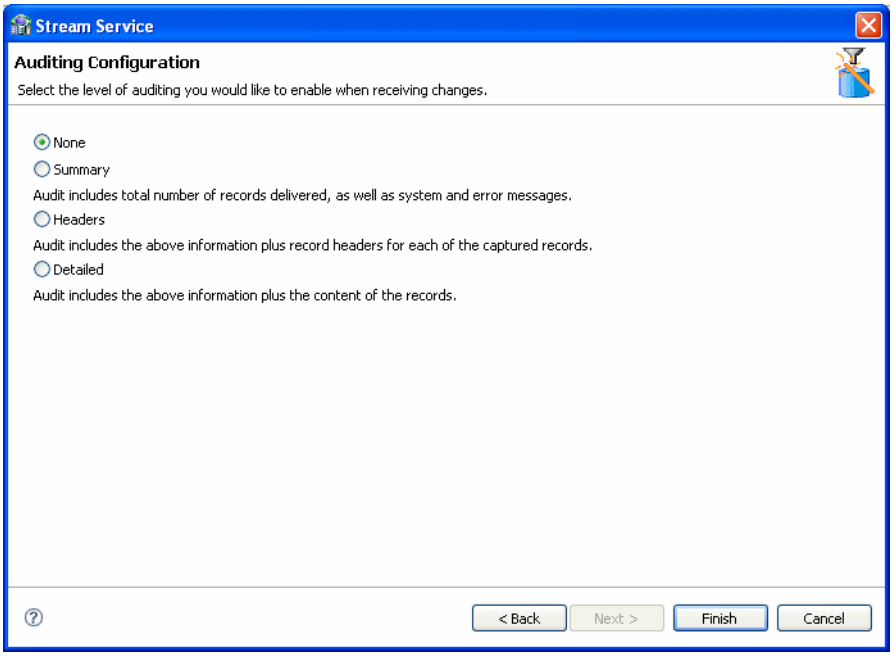
    The Add between values screen is displayed.

Figure 3–23 Add Items (Between)



15. Enter values for events to be returned where the column value is between the two values you specify.
16. In the Content Filter screen, click **Next**.

Figure 3–24 Auditing Configuration



17. Select the required auditing level when receiving changes. Your options are:
    - **None**: For no changes.
    - **Summary**: For an audit that includes the total number of recorded delivered, as well as system and error messages.
    - **Headers**: For an audit that includes the total number of records delivered, system and error messages, and the record headers for each captured record.
    - **Detailed**: For an audit that includes the total number of records delivered, system and error messages, the record headers for each captured record, and the content of the records.
18. Click **Finish**.

Continue setting up the staging area Server.

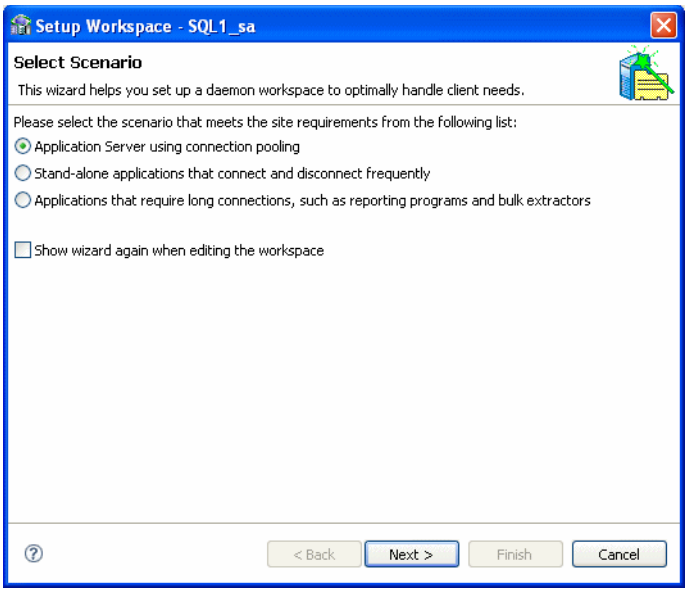
Configure the Access Service Manager

In this step you set up a daemon workspace for the CDC adapter. Do the following to configure the access service manager.

1. Click **Access Service Manager**.

   The Setup Workspace wizard opens.

Figure 3–25 Select Scenario



2. Select the scenario that best meets your site requirements:
   - **Application Server using connection pooling**
   - **Stand-alone applications that connect and disconnect frequently**
   - **Applications that require long connections, such as reporting programs and bulk extractors**
3. Click **Next**.

   The next screen that is used to create a workspace server pool. The parameters available depend on the selection you made in the first screen. The following are the available parameters:

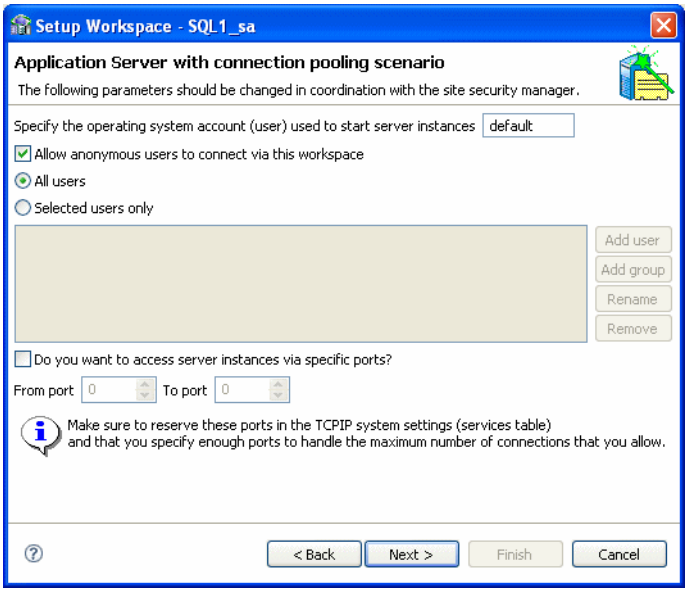
   - If you selected **Application Server using connection pooling**:
     - **What is the average number of expected concurrent connections?** Enter the number of expected connections, which cannot be greater than the number of acutal available connections.
     - **What is the maximum number of connections you want to open?** Enter the number of connections you want opened.
   - If you selected **Stand-alone applications that connect and disconnect frequently**, in addition to the choices listed in the item above, you can also set the following:
     - **What is the minimum number of server instances you want available at any time?** Enter the number of instances, which cannot be greater than the number of actual available instances.
     - **What is themaximum number of server instances you want available at any time?** Enter the number of instances you want to be available.
   - If you selected **Stand-alone applications that connect and disconnect frequently**:
     - **How many connections do you want to run concurrently?** Enter the number of concurrent connections that you want to run.

4. Click **Next**. In the next screen you set time out parameters. These parameters should be changed if the system is slow or overloaded. The parameters are:
   - **How long do you want to wait for a new connection to be established?** Enter the amount of time you want to wait for a new connection to be established (in seconds).
   - **How long do you want to wait for a response that is usually quick?** Change this parameter if you have a fast connection. Enter the amount of time to wait for a response (in seconds).
5. Click **Next**. In the next screen you set security parameters. You should consult with the site security manager before changing these parameters.

   Edit the following parameters in this screen:
   - Enter the operating system account (user name) used to start server instances.
   - Select **Allow anonymous users to connect via this workspace**, if you want to allow this option.
   - Enter the permissions for the workspace. You can allow **All users** to access the workspace, or select **Selected users only** to allow only the users/groups that you want to have exclusive access.
   - Select **Do you want to access server instances via specific ports,** if you want to allow this options. If this option is cleared, the defaults are used.

     If you select this option, indicate the **From port** and **To port** and make sure that you reserve these ports in the TCP/IP system settings.

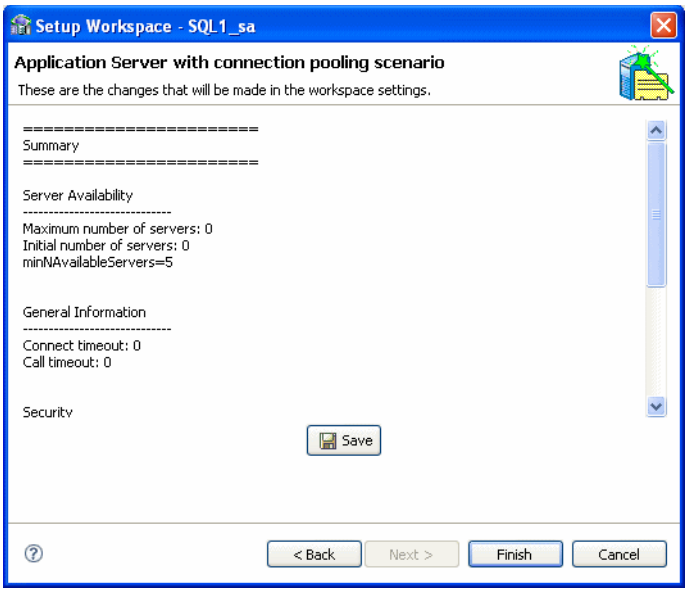
Figure 3–26 Site Security



6. Click **Next**.

   The summary screen opens.

Figure 3–27 Workspace Setup Summary



7. Click **Save** and then click **Finish**.

When you complete all the Implementation operations, a check mark (✓) is displayed next to every link. Click **Done** to return so you can begin Deploying a Change Data Capture.

Deploying a Change Data Capture

After you complete the design and implementation guides, the following procedures are available.

- **Deployment Procedure**: This section is used to deploy the project.
- **Control**: This section is used to activate or deactivate workspaces after the project is deployed and you are ready to consume changes. In this section, you can deactivate the workspace anytime you want to suspend consumption of changes from the staging area.

Note: Before deploying the solution, make sure you have carried out the following tasks, in addtion to setting up the CDC solution in Oracle Studio:

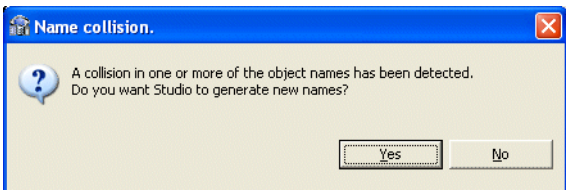
- Enabling SQL Replication
- Configuring Security Properties
- Setting up Log On Information
- Setting up the Database
- Setting Up the TLOG Miner (LGR)

Do the following to deploy the CDC solution:

1. Click **Deploy**. The Deployment Procedure and Control sections are displayed in the Deployment view.
2. Click **Deploy** in the **Deployment Procedure** section.

Studio processes the naming information. This may take a few minutes. If there are naming collisions, a message is displayed asking if you want Oracle Studio to resolve them.

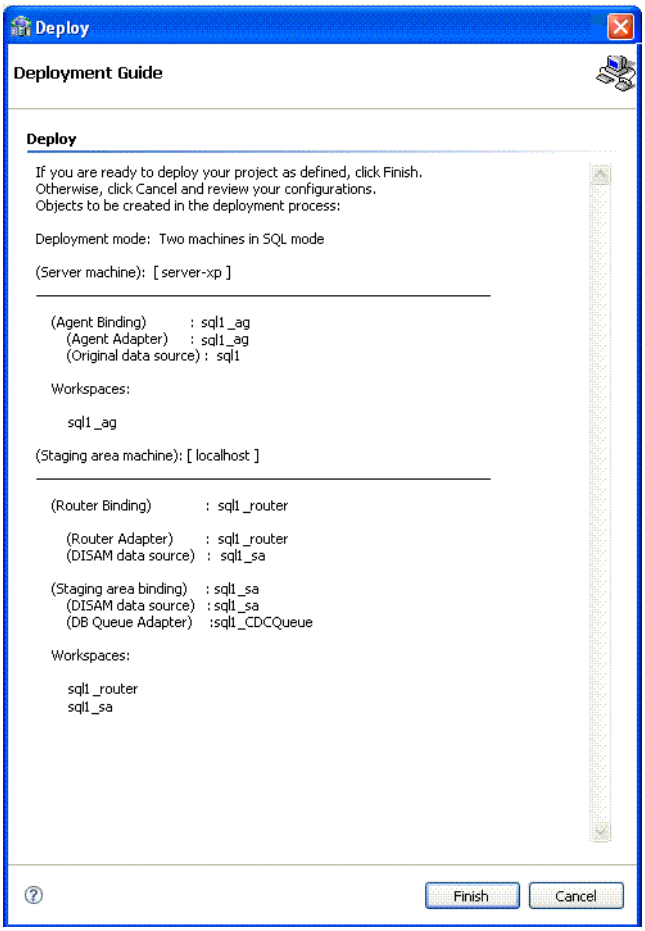
Figure 3–28 Resolve Naming Collision



3. Click **Yes** to resolve any naming collisions.

   The Deployment Guide is displayed.

Figure 3–29 Deployment Guide



4. If you are ready to deploy, click **Finish**.

   Otherwise, click **Cancel** and you can return to Create a CDC Project, Set up the CDC Server, or Set up the Staging Area Server to make any changes.

   If this project was deployed previously, you will be notified that re-deployment will override the previous instance.

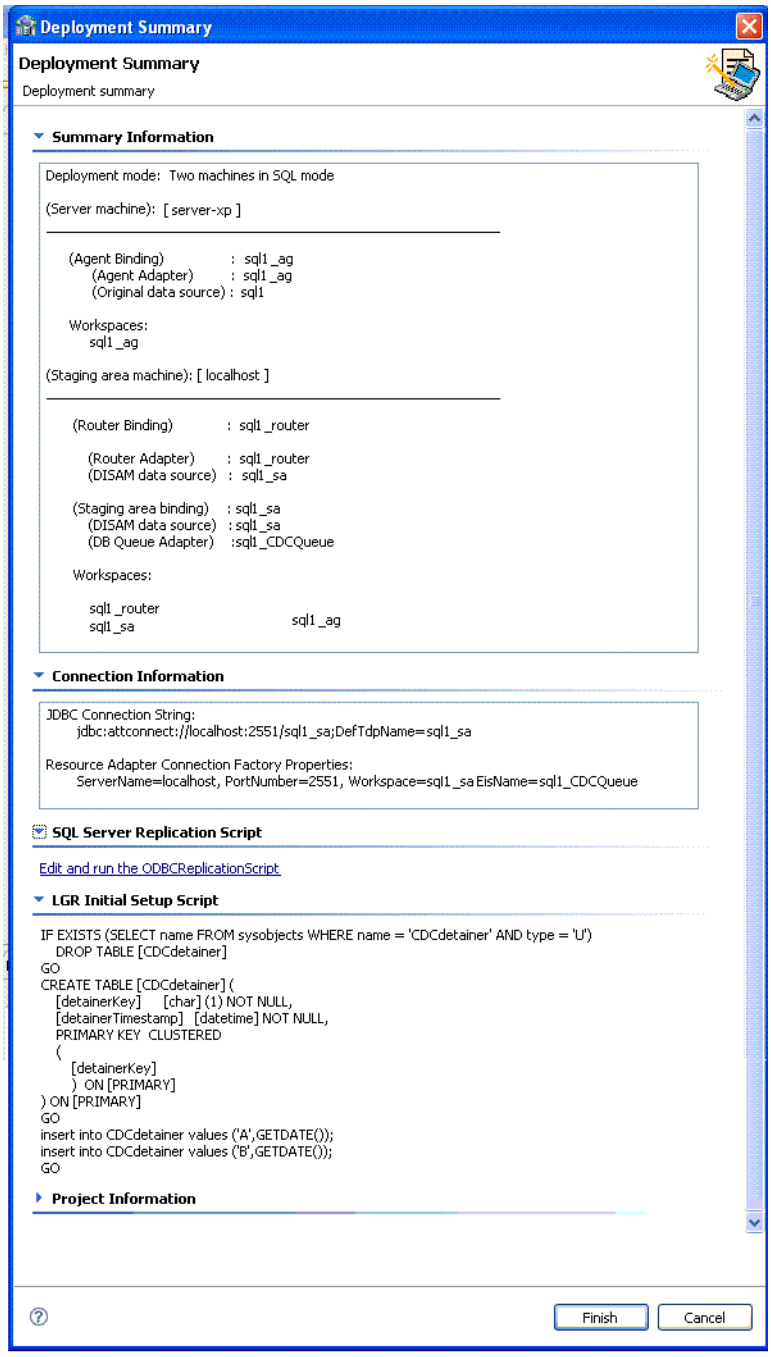
Notes:

- When you redeploy a project where the metadata is changed, the Staging Area (SA) tables should be deleted so that no incorrect information is reported.
- When you redeploy a solution, the a new binding is created for the solution. The new binding is created with the default parameters only. Any temporary features that were added are lost.

5. Where applicable, click **OK** to redeploy.
6. Click the **Deployment Summary** link.

   The Deployment Summary is shown. It includes the ODBC connection string and JDBC connection string, as well as specific logger scripts to enable CDC capturing.

Figure 3–30 Deployment Summary



7. Cut and paste any information required from the Deployment Summary screen to your environment as necessary.

> **Note:** The scripts generated in the deployment summary is template. Before running the scripts, they should be reviewed and modified by a DBA.

8. If there is nothing wrong with your deployment results, click **Finish**.

If you found problems, click **Cancel** and to return Create a CDC Project, Set up the CDC Server, or Set up the Staging Area Server to modify the solution.

Note: If you are redeploying a solution you must follow these directions to make sure that the `context` and `agent_context` fields of the `SERVICE_CONTEXT` table should be saved. Follow these directions to save the fields:

1. In the staging area data source run: select context, `agent_context from SERVICE_CONTEXT`; and save the returned values.
2. Delete the `SERVICE_CONTEXT` table physical files.
3. Redeploy the solution.
4. Activate the router to create the `SERVICE_CONTEXT` table.
5. Disable the router.
6. In the staging area datasource run: `insert into SERVICE_CONTEXT (context, agent_context) values('XXX', 'YYY')`. This will Insert the saved values to the SERVICE_CONTEXT table.
7. Activate the solution.

Activating and Deactivating Solution Workspaces

In the Project guide for your OracleAS CDC adapter solution, click **Deploy**, then do *one* of the following to activate or deactivate the workspaces for a solution

- To activate workspaces, under the Control section iclick the **Activate Workspaces** link.
- To deactivate workspaces, click the **Deactivate Workspaces** link.

During the activation/deactivation process, you may receive messages indicating that the daemon settings on one or more of the machines involved in your solution have changed. Click **Yes** to proceed.

Testing the OracleAS CDC Adapter for SQL Server CDC Solution

Check the following to ensure that the OracleAS CDC Adapter for SQL Server will operate correctly.

- The system contains a Temporary Transient working folder
- All consumed tables are "articled" within at least one Replication/publication definition
- All consumed tables have a primary key
- Verify that the TLOG Miner components are running (see Environment Verification)

Handling Metadata Changes

When you make changes to the source tables in your SQL Server CDC Solution, you need to be sure that the CDC solution can recognize the changes and work with them. This section provides you with a procedure to handle the metadata in your OracleAS CDC adapter for SQL Server CDC solution if changes are made after deploying the solution. You should carry out these steps at a time when there is little or no activity in the system. If you want to receive new events with a new structure, consume the

changes for the table you are updating before carrying out any the steps in this process.

Do the following to handle changes to metadata:

1. Deactivate the Solution using Oracle Studio.
2. Update the metadata on the backend database for the table you are working with.

   In Microsoft SQL Server 2005, an inconsistency between the modified metadata and the data layout can appear because of the changes made to the metadata. To handle this inconsistency:

   - If a clustered index is defined for the table, run:

     ```
     DBCC DBREINDEX ('<table name>',<clustered index>)
     ```

     Where `<table name>` is the table with updated metadata, and `<clustered index>` is the name of its clustered index.
   - If no clustered index is defined, reload the table.
3. Update the metadata in the Staging Area by doing *one* of the following:

   - If you made manual changes to the CDC solution after deployment, or if you do not want to redeploy the solution, then on the Router's (Staging Area) machine, do the following:

     Run Oracle Studio, and open the Design perspective.

     Edit the Metadata for the Router's Data source.

     Expand the table list and edit the metadata for the table.

     If you are adding a new column, make sure to add it to the end of the `COLUMN` list. This operation can also be done using the Source view. Make sure you select the correct datatype.

     If you are modifying a datatype, make sure to select the corresponding data type when making the modification.

     Save the metadata.

     For more information, see Advanced Tuning of the Metadata.
   - For cases where you can redeploy the solution:

     Run Oracle Studio, and open the Solution perspective.

     Open the CDC solution project.

     Click **Implement** and then click **Stream Service**.

     Run the wizard.

     Redeploy the solution, but do not activate it.

     For more information, see Deploying a Change Data Capture.
4. Delete the physical files that represent the modified tables from the Staging Area. Make sure *not* to delete the SERVICE_CONTEXT and CONTROL_TABLE files.
5. Reactivate the solution using Oracle Studio.

Environment Verification

The following topics show how to ensure that the SQL Server CDC solution components are configured properly. This section has the following topics.

- Verify the SQL Server Version
- Ensure that the Service is Registered
- Verify that the LGR Service is Running
- Viewing the Service Greetings
- Check the Output Files

Verify the SQL Server Version

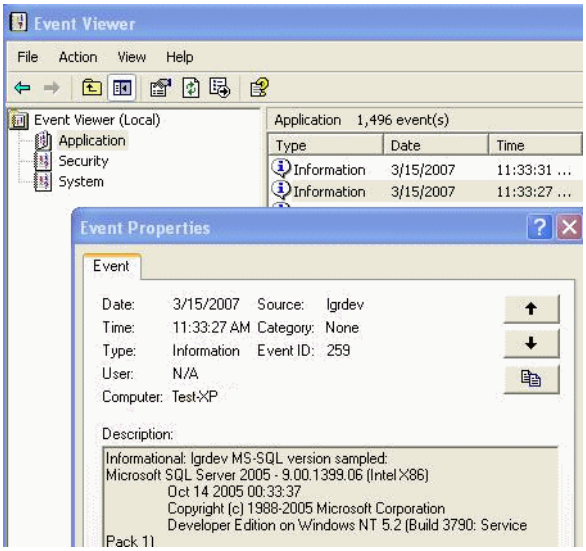
During setup, the LGR checks to see which version of the Microsoft SQL Server you are using to be sure that it works with the correct standards. You may need to check to verify that the correct SQL Server version is recognized by the LGR service. To verify the version, do one of the following:

- In the initial setup section of the LGR log file, find the backend version stamping as follows:

```
<<20070315-113327>>     Module:sqlcdclgr/Line:697       MS-SQL version sampled:
Microsoft SQL Server 2005 - 9.00.1399.06 (Intel X86)
         Oct 14 2005 00:33:37
         Copyright (c) 1988-2005 Microsoft Corporation
         Developer Edition on Windows NT 5.2 (Build 3790: Service Pack 1)
```

- In the Windows Event Viewer, do the following:
  - Find the `lgrdev` source for any **Information** entry.
  - Double-click the entry to display the following. Check to see if the message describes the SQL version. If not, try another entry.

Figure 3–31 Verify SQL Server Version



Ensure that the Service is Registered

Use the System Registry (`REGEDIT`) to ensure that:

- The TLOG Miner service is registered
- The TLOG Miner service is assigned as a Windows event log source

Do the following to check that the TLOG Miner service is registered:

In the System Registry (`REGEDIT`), approve the service and its parameters. To access the registry:

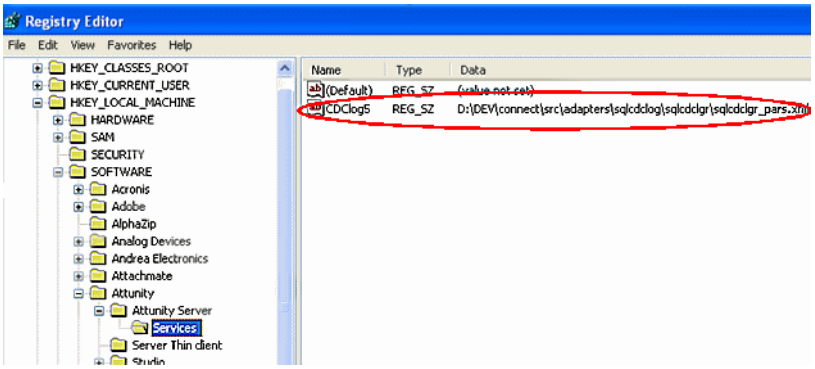
- Click **Start**, click **Run**, type **regedit**, and then click **OK**.

  Scroll thorough the registry tree by expanding the folders that lead to the root folder where you installed Oracle Connect. The path listed here assumes that you installed Oracle Connect in the default location:

  `HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\Oracle Connect\Services`. Be sure that the LGR service registration is listed on the right side.

  The following is an example of how the registry may look:

Figure 3–32 Registry



For checking if the TLOG Miner service is assigned as a Windows event log source, follow this procedure.

Do the following to check that the LGR service is assigned as a Windows event log source:

In the System Registry (`REGEDIT`), browse to the `CDClog` folder. To access the registry:

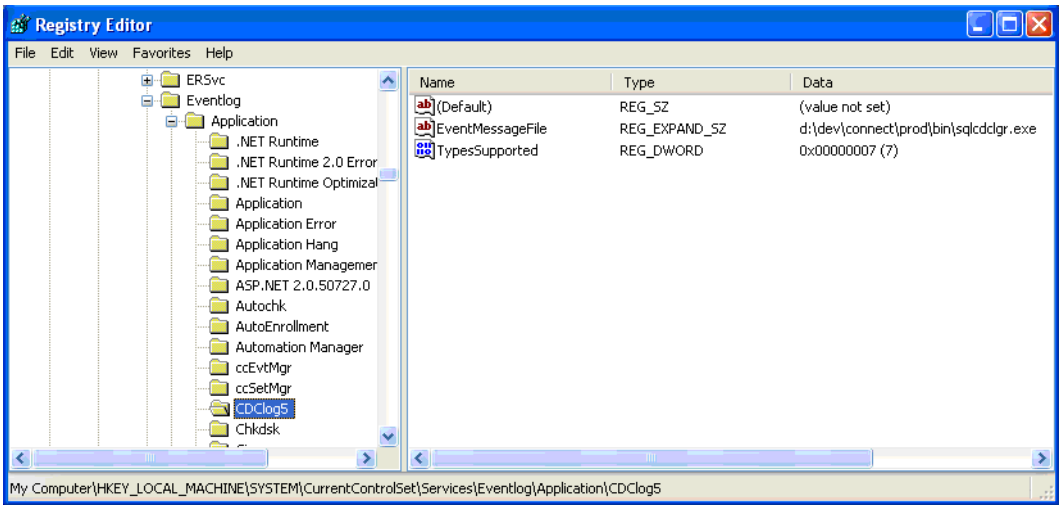
- Click **Start**, click **Run**, type **regedit**, and then click **OK**.

  Scroll thorough the registry tree by expanding these folders:

  `SYSTEM\CurrentControlSet\Services\EventLog\Application\CDClog5`. Be sure that the LGR service registration is listed on the right side.

  The following is an example of how the registry may look:

Figure 3–33 Registry CDCLog folder



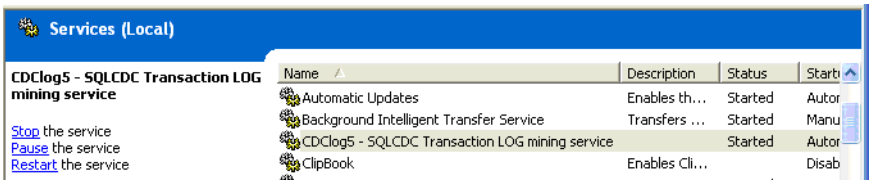
Verify that the LGR Service is Running

You should carry out the following to be sure the TLOG Miner service is running:

- Ping the Service
- Start the service for a period of time, and then stop it. To start the service:
  - From the Windows **Start** menu, select **Control Panel, Administrative Tools** and double-click the **Services** icon.

    Find the service is listed in the **Name** list and click **Stop the service**. To start the service click **Restart the service**. See the example below.

Figure 3–34 Start and Stop the Service



Viewing the Service Greetings

Open the Event Viewer and view the messages in the Event Properties dialog box.

Do the following to view the event properties:

1. From the Windows **Start** menu, select **Control Panel, Administrative Tools** and double-click the **Events** icon.
2. Select System on the left side of the viewer.
3. From the right pane, right click and event from the SQL Server CDC and select **Properties**. The following figure is a sample of the information that is displayed:

Figure 3–35 Service Greeting

Check the Output Files

You should check the following files:

- LGR Service log files: These files are in the folder or directory that is selected in the `Logging` parameter of the template input file.
- Transient storage output file: This file is in the folder or directory that is selected in the `transientStorage` parameter of the template input file.

For information on where to define these parameters, see Configuring the Template Input File.

4

Oracle BPEL Process Manager Deployment and Integration

To deploy and integrate OracleAS CDC Adapters for SQL Server with Oracle BPEL Process Manager, you need to configure the BPEL Process Manager.

This section includes the following topics:

- Overview of Oracle BPEL Process Manager Integration
- Configuring the SQL Server CDC Adapter In the Application Server
- Configuring the Oracle BPEL Process Manager

Overview of Oracle BPEL Process Manager Integration

Oracle BPEL Process Manager provides a comprehensive solution for creating, deploying, and managing BPEL business processes. Oracle BPEL Process Manager is based on the Service Oriented Architecture (SOA) to provide enterprises with flexibility, interoperability, reusability, extensibility, and rapid implementation of Web services and business processes. It reduces the overall costs of management, modification, extension, and redeployment of existing business processes. Each business activity is a self-contained, self-describing, and modular application whose interface is defined by the WSDL, and the business process is modeled as a Web Service.

A Web Service is first published and then composed or orchestrated into business flows. Publishing a service is implemented by taking a function within an existing application or system and making it available in a standard way, while orchestration is implemented by composing multiple services into an end-to-end business process. The CDC solutions that are defined as part of the configuration of the OracleAS CDC Adapter for SQL Server are integrated into the orchestration as PartnerLinks. Every PartnerLink is linked to a WSDL that describes the CDC table.

See Also: *Oracle Application Server Adapter Concepts Guide*.

Configuring the SQL Server CDC Adapter In the Application Server

You can use the OC4J or WebLogic application servers to deploy the Oracle Connect Legacy adapter. This section describes how to deploy the Legacy adapter, set up the necessary connections, and build the CDC Stream Positions table.

This section includes the following topics:

- CDC Stream Positions Table Definition

- Configuring the SQL Server CDC Adapter Connection Factory

CDC Stream Positions Table Definition

Each CDC inbound process is defined for one source table only. You create one CDC Stream Positions table for each CDC adapter. The CDC Stream Positions table contains one row for each process (for each table) to control its stream position and XA transaction. The following SQL statement is used to define the table:

```
create table CDC_STREAM_POSITIONS(TABLE_NAME varchar(127) not null,
STREAM_POSITION varchar(127), PREPARED_STREAM_POSITION varchar(127),
ROLLBACK_STREAM_POSITION varchar(127), NEW_STREAM_POSITION varchar(127),
XID_FORMAT int, XID_BRANCH varchar(128), XID_GLOBAL varchar(128));
create unique index CDC_STREAM_POSITIONS_INDEX on CDC_STREAM_POSITIONS
(TABLE_NAME);
```

You can change the name of the table, but not the columns.

The following table describes the CDC Stream Positions table columns.

Table 4–1 CDC Stream Positions Table

Columns	Description
`TABLE_NAME`	The name of the CDC table for the inbound process. You should not change the data in this column.
`STREAM_POSITION`	The last committed stream position of the CDC process for the current table. You should not change the data in this column.
`PREPARED_STREAM_POSITION`	The last prepared stream position of the CDC process for the current table. You should not change the data in this column.
`ROLLBACK_STREAM_POSITION`	In case of a Rollback Loop, this column contains the stream position of the event that occured at the time of the rollback.
`NEW_STREAM_POSITION`	You can enter a valid stream position value in this column to manually change the current stream position. Before you update this column, deactivate the corresponding CDC process. **Note**: Make sure that the `XID_FORMAT` column has a null value. If the `XID_FORMAT` column is not null, do not enter a value for this column.

`XID_FORMAT`
`XID_BRANCH`
`XID_GLOBAL`
	The last prepared XID fields. If the `XID_FORMAT` column is null, the stream position is committed. In this case the `XID_BRANCH` and `XID_GLOBAL` columns will contain the last committed XID fields.

You should configure the native data source that contains the CDC Stream Positions tables in your application server. The following example shows an example of a data source definition in OC4J (in the `data-sources.xml` file):

```
<native-data-source
  name="CdcContextOracleDS"
  jndi-name="jdbc/CdcContextOracleDS"
  description="Native CDC Context DataSource"
  data-source-class="oracle.jdbc.pool.OracleDataSource"
  user="system"
```

```
  password="manager"
  url="jdbc:oracle:thin:@//server1-xp:2551/ora10202">
</native-data-source>
```

Configuring the SQL Server CDC Adapter Connection Factory

Deploy the Oracle Connect Legacy Adapter RAR into the OC4J or WebLogic server with the name `jca-legacy-adapter`.

Create the appropriate connection factories in the OC4J or WebLogic application server.

This table provides a description of the connection factory properties.

Table 4–2 Connection Properties for CDC Inbound Process

Property	Description
`eisName`	Required. Sets the name of the adapter to use. The adapter is defined in the Oracle Connect server using Oracle Studio.
`serverName`	Required. Enter the name of the server with the TCP/IP address or host name where the Oracle Connect daemon is running. See Also: Advanced Tuning of the Daemon for details about the daemon.
`workspace`	Required. Specifies the name of an Oracle Connect server workspace to use. See Also: Workspaces for details about workspaces.
`portNumber`	Optional. Specifies the TCP/IP port where the Oracle Connect daemon is running on the server. The default port is `2551`.
`userName`	Optional. Specifies a user who can access the Oracle Connect server. The user is defined in the Oracle Connect daemon configuration. See also: daemon Security and workspace Security for details about users allowed to access an Oracle Connect server
`password`	Optional. Specifies a valid password for the user.
`persistentConnection`	By defalult this is set to `true`. You should not change this value.
`keepAlive`	By defalult this is set to `true`. You should not change this value.
`firewallProtocol`	Optional. Specifies the firewall protocol used: either `none` or `fixedNat` (the Nat protocol using a fixed address for the daemon). The default is set to `none`.
`connectTimeout`	Optional. Specifies the connection timeout in seconds. The default is 0, indicating that there is no connection timeout.
`encryptionProtocol`	Optional. Specifies the name of encryption protocol to use. The default is set to `RC4`. If the value of the property is not defined, the RC4 protocol is used.
`encryptionKeyName`	Optional. Specifies the name of the symmetric encryption key to use.
`encryptionKeyValue`	Optional. Specifies the value of the symmetric encryption key to use.

Table 4–2 (Cont.) Connection Properties for CDC Inbound Process

Property	Description
`useNamespace`	By defalult this is set to `true`. You should not change this value.
`exposeEventStreamMetadata`	By defalult this is set to `true`. You should not change this value.
`retryInterval`	Optional. The amount of time the resource adapter waits if no events are found in the server before issuing the next request. The default value is `15` seconds.
`jdbcDataSource`	Required. The JNDI name of the JDBC native data source that contains `cdcStreamPositionsTable`.
`cdcStreamPositionsTable`	The SQL table that is used to control the CDC inbound process. See CDC Stream Positions Table Definition for more information.

Configuring the Oracle BPEL Process Manager

This section includes the following topics:

- Setting up the Connection to the Oracle Connect Server Metadata
- Using JDeveloper to Integrate a CDC Inbound Process to the Inbound Application

Setting up the Connection to the Oracle Connect Server Metadata

Perform the following steps to set up the connection to the Oracle Connect server to provide an automatic BPEL application built by JDeveloper:

1. Open the Oracle BPEL Admin window.
2. On the Server tab, on the Configuration tab, specify the following:
   - `LegacyServer`: The IP address of the server where Oracle Connect is installed.
   - `LegacyPort`: The port number of the server where Oracle Connect is installed.
3. Repeat the previous step for each Oracle Connect server to be used by Oracle BPEL Process Manager. Use a comma as a separator between the different servers and ports.
4. Click **Apply**.
5. Restart the Oracle SOA Suite.

Using JDeveloper to Integrate a CDC Inbound Process to the Inbound Application

The following sections describe how to integrate the CDC inbound process to the inbound application.

- Configuring a CDC Adapter Partner Link
- Configuring WSDL

Configuring a CDC Adapter Partner Link

Carry out the following steps in Oracle JDeveloper to configure a CDC Adapter Partner Link.

1. Drag a Partner Link into one of the Services lanes of the visual editor.
2. In the Create Partner Link dialog box, click the **Service Explorer** button (the second button under **WSDL Settngs**.
3. In the Service Explorer, expand **Adapter Services**.
4. Expand the connection server on which you deployed the SQL Server resource adapter (this is the server where you created your CDC solution).
5. Under **Adapters**, expand **Legacy** to view a list of the Oracle Connect servers that you defined by using the Oracle BPEL Admin window.
6. Expand the Oracle Connect server with the metadata you want to check, to view a list of workspaces.

   Expand the CDC staging area workspace that you created when . This workpace will have the name with the suffix `_sa`.
7. Expand the CDCQueue adapter to view a list of captured tables.
8. Select the table you are working with and click **OK**.
9. Define the **Partner Role** and **My Role** then click **OK**.

Configuring WSDL

When you build an inbound Partner Link, Oracle BPEL Process Manager automatically creates the WSDL that corresponds to the captured table, including the properties of the `AttuCDCActivationSpec` class. These properties are the only relevant properies for a CDC inbound process. These properties, except for the `TableName`, can also be provided using the Connection Factory. For information on how to configure the Connection Factory, see Configuring the SQL Server CDC Adapter Connection Factory.

The WSDL specifies the name of the adapter's connection factory as the value of the `adapterInstanceJndi` attribute of the `<jca:address>` element in the `<service>` section. If a connection factory exists on the applicatio server, its properties are taken. Otherwise, the properties specified by the `ActivationSpec` are used. If a value is specified by both the connection factory and the `ActivationSpec`, the `ActivationSpec` property overrides the value in the connection factory. If you want to use the value specified in the connection factory, you have to delete the property from the WSDL.

The following is an example of a CDC inbound WSDL:

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<definitions name="nation" targetNamespace="http://xmlns.oracle.com/pcbpel/dlg1_
CDCQueue/nation" xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:legacyReq="noNamespace://dlg1_CDCQueue"
xmlns:tns="http://xmlns.oracle.com/pcbpel/dlg1_CDCQueue/nation"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
xmlns:pc="http://xmlns.oracle.com/pcbpel/" xmlns:legacyRes="noNamespace://DEMO#">
    <types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="noNamespace://dlg1_CDCQueue" targetNamespace="noNamespace://dlg1_CDCQueue"
elementFormDefault="qualified" attributeFormDefault="unqualified">
```

```
          <xsd:element name="nationStream" type="nationStreamDescription"/>
            <xsd:complexType name="nationStreamDescription">
                <xsd:sequence>
                    <xsd:element name="nation" type="nationOut" minOccurs="1"
maxOccurs="unbounded"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:element name="nationOut" type="nationOut"/>
            <xsd:complexType name="nationOut">
                <xsd:sequence>
                    <xsd:element minOccurs="0" maxOccurs="1" name="header"
type="nation__header"/>
                    <xsd:element minOccurs="0" maxOccurs="1" name="data"
type="nation__data"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:element name="nation__data" type="nation__data"/>
            <xsd:complexType name="nation__data">
                <xsd:attribute name="n_nationkey" type="xsd:int" use="required"/>
                <xsd:attribute name="n_name" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="25"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="n_regionkey" type="xsd:int" use="required"/>
                <xsd:attribute name="n_comment" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="152"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:complexType>
            <xsd:element name="nation__header" type="nation__header"/>
            <xsd:complexType name="nation__header">
                <xsd:attribute name="timestamp" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="26"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="tableName" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="64"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="operation" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="12"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="transactionID" use="required">
                    <xsd:simpleType>
```

```
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="24"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="context" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="32"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="agent_context" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="64"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="terminalID" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="4"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
                <xsd:attribute name="taskID" use="required">
                    <xsd:simpleType>
                        <xsd:restriction base="xsd:string">
                            <xsd:maxLength value="8"/>
                        </xsd:restriction>
                    </xsd:simpleType>
                </xsd:attribute>
            </xsd:complexType>
        </xsd:schema>
    </types>
    <message name="event">
        <part name="event_nation" element="legacyReq:nationStream"/>
    </message>
    <portType name="nationPortType">
        <operation name="nation">
            <input name="Event_nation" message="tns:event"/>
        </operation>
    </portType>
    <binding name="nationJCABinding" type="tns:nationPortType">
        <pc:inbound_binding/>
        <operation name="nation">
            <jca:operation
ActivationSpec="com.oracle.adapter.oracle.AttuCDCActivationSpec"
TableName="nation" JdbcDataSource="jdbc/CdcContextOracleDS"
CdcStreamPositionsTable="" EisName="dlg1_CDCQueue" ServerName="server1-xp"
PortNumber="2551" UserName="" Password="" Workspace="dlg1_sa" RetryInterval="15"
ConnectTimeout="0"/>
            <input/>
        </operation>
    </binding>
    <service name="nationService">
        <port name="nationPort" binding="tns:nationJCABinding">
            <jca:address
ResourceAdapterClassName="com.oracle.adapter.AttuResourceAdapter"
```

```
adapterInstanceJndi="eis/legacy/dlg1_CDCQueue"/>
        </port>
    </service>
    <plt:partnerLinkType name="nationPartnerLinkType">
        <plt:role name="nationRole">
            <plt:portType name="tns:nationPortType"/>
        </plt:role>
    </plt:partnerLinkType>
</definitions>
```

Note: IIn this example, the element `<nation>` is defined as an array. The maximum number of elements in this array is defined by the `maxRecords` property in the corresponding CDC Queue adapter.

Troubleshooting Rollback Loops

When a problem occurs in the CDC resource adapter work thread, the thread will wait for a defined period of time (`retryInterval`) then it will retry the last operation. If you want to stop the adapter, you must deactivate the endpoint. The behavior of the resource adapter is different according to the type of rollback loop. In all cases, the resource adapter writes the corresponding error messages to the resource adapter log file and sends the message to the message listener `onAlert` method.

- If the connection to Oracle Connect is lost, the resource adapter always waits for a defined period of time (`retryInterval`) until the connection is reestablished.
- In all other cases, the resource adapter retrys the last operation immediately and if the problem returns, the resource adapter waits for acsending periods of time, beginning with the `retryInterval` and doubling the wait time for each successive attempt, but not for more than one minute.

  If the OracleAS resource manager invokes a rollback before a 2PC `prepare` operation, the CDC resource adapter take steps to carry out a special operation that assists you in troublshooting the problem. In this case, the resource adapter separately sends each event from the last event array to the BPEL endpoint, which saves the stream position of the last problematic event in the `ROLLBACK_STREAM_POSITION` column.

  If you notice that a rollback loop occurs in the CDC resource adapter work thread, check the value of the `ROLLBACK_STREAM_POSITION` is not null. If the value is not null, check whether there was a data problem. You can select the problematic event from the staging area, using the current stream position value. In this way you can analyze the data to determine what the problem is. In addition, you can skip the problematic event. In this case you stop the endpoint and update the `NEW_STREAM_POSITION` column with the value of the `ROLLBACK_STREAM_POSITION`.

5

Runtime Tasks and Troubleshooting

Runtime tasks are executed after you install and configure the OracleAS CDC Adapter for SQL Server. This chapter describes how to carry out basic maintenance tasks, and make changes to CDC solutions. These tasks include:

- Adding and Removing Tables
- Handling Metadata Changes
- Staging Area Maintenance
- Monitoring the Change Data Capture
- Daemon Life-Cycle Tasks
- Resolving Communication Errors
- Resolving Specific Errors

Adding and Removing Tables

After you deploy the OracleAS CDC adapter solution and start working with it, you may want to change the tables that are being monitored. You do this in the CDC Solution perspective of Oracle Studio. The following describes the main steps that you need to follow to make changes to the table. During this operation you will need to go back and change some of the configurations you made when you set up the OracleAS CDC adapter solution during design time. Do the following to change tables:

1. From the **Start** menu, select **Start**, **Programs**, **Oracle**, and then select **Studio**.
2. From the **Window** menu, click the **Open Perspective** button and select **CDC Solution**.
3. In the Getting Started pane, under Recent projects, click the OracleAS CDC solution that you are changing.

   If you do not see the CDC solution you want to edit in the list under Recent projects, click Open an existing project and select your solution from the drop-down list in the dialog box that is displayed.
4. Click **Implement**.
5. Under the **Stream Service Configuration** section, click **Stream Service**.
6. In the first screen of the Stream Service wizard, you can change the location of the change files or click **Next** to make changes to the tables to be captured.
7. Select one or more tables from either pane.

- Select a table from the right pane then click the left arrow to move a selected table into theleft pane and remove it from the list of tables to be captured.
- Select a table from the left pane then click the right arrow to move a selected table into the right pane and add it to the list of tables to be captured.

8. Click **Next**, in the screen displayed, clear check boxes from the columns you no longer want to capture. Select check boxes from columns that are not currently included in the change data capture to include them.
9. Click **Next**, in the screen displayed, clear and select the check boxes to change the filters used in the change data capture.
10. Click **Next**, in the final screen you can makde changes to the auditing configuration.
11. Click **Finish** to close the wizard and save the changes.
12. You must re-deploy the solution for the changes to be recognized at runtime. See Deploying a Change Data Capture.

For more information on configuring the Stream Service, see Set up the Stream Service.

Note: When adding a new table, be sure to create a corresponding endpoint for capturing changes. See Using JDeveloper to Integrate a CDC Inbound Process to the Inbound Application.

Handling Metadata Changes

When you make changes to the metadata, you need to be sure that the solution recognizes the changes and works with them. This section provides you with a procedure to handle the metadata when working with an OracleAS CDC adapter for SQL Server solution if changes are made after deploying the solution. You should carry out these steps at a time when there is little or no activity in the system. If you want to receive new events with a new structure, consume the changes for the table you are updating before carrying out any the steps in this process. Do the following to handle any changes to the metadata:

1. Deactivate the Solution workspaces using the CDC Solution perspective in Oracle Studio.
2. Update the metadata on the backend database for the table you are working with.
3. Update the metadata in the Staging Area by doing *one* of the following:
   - If you made manual changes to the CDC solution after deployment, or if you do not want to redeploy the solution, then on the Router's (Staging Area) machine, do the following:

     Run Oracle Studio, and open the Design perspective.

     Edit the Metadata for the Router's Data source.

     Expand the table list and edit the metadata for the table.

     If you are adding a new column, make sure to add it to the end of the `COLUMN` list. This operation can also be done using the Source view. Make sure you select the correct datatype.

     If you are modifying a datatype, make sure to select the corresponding data type when making the modification.

     Save the metadata.

- For cases where you can redeploy the solution:

  Run Oracle Studio, and open the Solution perspective.

  Open the CDC solution project.

  Click **Implement** and then click **Stream Service**.

  Run the wizard.

  Redeploy the solution, but do not activate it.

  For more information, see Deploying a Change Data Capture.

4. Delete the physical files that represent the modified tables from the Staging Area. Make sure *not* to delete the SERVICE_CONTEXT and CONTROL_TABLE files.
5. Reactivate the solution using Oracle Studio. See Activating and Deactivating Solution Workspaces.

Staging Area Maintenance

The Staging Area files for the OracleAS CDC adapter solution are DISAM files that store changes until the client application consumes them. When you delete old changed records they are actually only marked as deleted. New changes continue to be written to the deleted records.

In busy production sites the DISAM files can get very large, which can affect the performance of the system.

To ensure that system performance is not degraded, you should defragment the Staging Area files to better maintain the Staging Area repository. Use the following procedures and suggestions when you carry out the defragmentation of the staging area:

- Defragment the DISAM files at least once a week. The frequency can vary depending on the amount of changes in the staging area.
- In addition to defragmenting the staging area files, you should also check for corrupt DISAM files.
- You should carry out the maitenance processes when the lowest possible activity in your system occurs.

To defragment and maintain the staging area, use the DCHECK and DPACK DISAM utilities. The DCHECK utility is used to check for corruption of DISAM files and rebuild the bad indexes and the DPACK utility defragments the DISAM files and rebuilds them without the deleted records. This reduces the size of the files and ensures that the DISAM files contain only active and relevant records.

Do the following to carry out the maintenance activities:

1. Deactivate the OracleAS CDC Solution with the staging area you working with. See Activating and Deactivating Solution Workspaces.
2. Run the DCHECK utility on each file to make sure it's not corrupted. For example, at the command prompt enter the following to check the DIASM file that represents the table called **employees**.

   ```
   dcheck employees
   ```

   If the file is corrupt you can use the `-b` switch to rebuild all indexes.

3. Defragment each file using the DPACK utility. For example, at the command prompt enter the following to defragment the DIASM files that represent the tables called **employees** and **salaries**.

```
dpack employees
dpack salaries
...
```

4. Reactivate all solutions. See Activating and Deactivating Solution Workspaces.

Monitoring the Change Data Capture

After you deploy the OracleAS Change Data Capture, you can monitor its progress. Monitoring provides you with information about the OracleAS CDC adapter's status, troubleshooting and tuning. This section contains the following topics that explain monitoring in a CDC.

- Service Context Table
- Monitoring the Status

Service Context Table

A control table is maintained by the event router that reports its current state and other important statistics. It can be accessed with any tool that supports SQL access.

The control table is called SERVICE_CONTEXT. This table has a single row with the following columns:

Table 5–1 SERVICE_CONTEXT Table Structure

Column Name	Data Type	Description
context	string (32)	The context value of the last change record in the most recently committed transaction. This value can be used to synchronize retrieval of transactions among different tables.
agent_context	string (64)	This is the agent context that the staging area would return to if it were to restart for whatever reason. The agent context value is calculated as follows: ▪ If there are pending uncommitted transactions, the `agent_context` value is the agent context of the first event of the oldest uncommitted transaction. ▪ If there are no pending uncommitted transactions the agent context of the last event of the most recently committed transaction, prefixed with'`next` and indicates that on recovery, the next event after that is to be processed. The staging area maintains an internal `agent_context` that is more advanced than the one stored in the SERVICE_CONTEXT table. The staging area uses memory to speed up change processing and when stopped it may revert back to an earlier agent context. The amount of extra work depends on the existence of long-running transactions.
start_time	timestamp	The time when the staging area started.
status	string (16)	Staging area status. For more information, see Monitoring the Status.
`sub_status`	string (64)	A second level status. For more information, see Monitoring the Status.
status_message	string (80)	Message that is returned that describes the staging area status.

Table 5–1 (Cont.) SERVICE_CONTEXT Table Structure

Column Name	Data Type	Description
status_time	timestamp	The time that the status is updated.
completed_transactions	uint4	Number of transactions processed.
active_transactions	uint4	Number of transactions in progress (in memory, not yet committed or rolled back).
timedout_transactions	uint4	Number of transactions that have timed out (were in memory for too long, declared to have timed out and written to a file).
rolledback_transactions	uint4	Number of rolled back transactions.
processed_change_events	uint4	Number of change events written out.
deleted_change_events	uint4	Number of change events deleted from change table.
bytes_written	uint4	Accumulated size in bytes of change records written.
opened_files	uint4	Current number of physically opened files by the staging area.
opened_files_virtual	uint4	Current number of logically opened files by the staging area.
memory_usage	uint4	Amount of memory currently allocated for staging.
`node_id`	uint4	A two-digit identifier with the same value as the `nodeID` config property when in multi-router mode. In regular mode, this column has no value.
`last_transaction_timestamp`	string (26)	The time of the last transaction.
`Version`	uint4	The version number for the router.
errors	uint4	Total number of errors reported.
Reduced_transactions	uint4	The number of transactions reduced to disc.
`compensation_records`	uint4	The number of compensation records captured.

The CONTROL table is also used by the event router to persist its state for purpose of recovery. This table must not be modified by the users.

Monitoring the Status

The following table describes the status for the CDC adapters when they are running. The status is defined as a state in the SERVICE_CONTEXT table. The table describes the different statuses available for a CDC adapter.

Table 5–2 SERVICE_CONTEXT Status States

State	Sub State	State Details	Description
Active	Processing	■ Reads the change events ■ Writes the change events ■ Reduces the timed-out transaction to disc ■ Deletes any expired change events	The router is connected to the CDC adapter and is processing or waiting for the change events.
	Idle	Waits for new change events	The router's adapter reaches the end of its journal and does not have any new change events.
error	router.discWriteError	Detailed error text	This indicates that the change router operation involving writing to disk failed. The most common reason is not enough disk space. Other reasons such as permissions, wrong path, or locking can also cause this.
	component.error This error type occurs in adapters and routers. The following are the errors that are returned for this error type: ■ `xmlError` ■ `requestError` ■ `noActiveConnection` ■ `resourceLimit` ■ `noSuchResource` ■ `authenticationError` ■ `noSuchInteraction` ■ `noSuchConnection` ■ `notImplemented` ■ `autogenRejected` ■ `resourceNotAvailable` ■ `authorizationError` ■ `configurationError` ■ `noSuchStream` ■ `temporarilyUnavailable` ■ `dataError` ■ `interventionRequired`	Detailed error text	The prefix *component* (Adapter/Router) indicates where the error happened. The *error* in the `sub_state` column identifies the error.
	Disconnected	Detailed error text	This indicates that the change router operation with the CDC adapter failed and cannot be restored.

Table 5–2 (Cont.) SERVICE_CONTEXT Status States

State	Sub State	State Details	Description
Paused	N/A		Operator manually paused the change router using the `sqlrtr_pause` control reset.
Down	N/A	Down message (orderly shutdown or abort message)	This indicates that the change router is not running.

Daemon Life-Cycle Tasks

Most of the daemon runtime tasks between Oracle Application Server and OracleAS CDC Adapter for SQL Server are carried out using Oracle Studio, which is used to monitor the daemon and server activity and control what happens to the daemon and server processes.

> **See Also:** Appendix B, "Advanced Tuning of the Daemon" for details about the configuration settings.

This section contains the following topics:

- Starting the Daemon
- Shutting Down the Daemon
- Monitoring the Daemon During Runtime
- Daemon Logs

Starting the Daemon

The daemon is started when OracleAS Adapter for SQL Server is installed. In case you have shut down the daemon, as described in Shutting Down the Daemon, you can restart the daemon as described in the following task.

> **Note:** The daemon is started on the Windows computer where Oracle Connect is installed. It cannot be started remotely using Oracle Studio.

Shutting Down the Daemon

To shut down the daemon use Oracle Studio, as follows:

1. From the **Start** menu, select **Start**, **Programs**, **Oracle**, and then select **Studio**.
2. Expand the **Machines** folder.
3. Right-click the computer defined in Setting Up a Windows Computer in Oracle Studio and select **Open Runtime Perspective**.
4. In the Runtime Explorer, right-click the daemon you want to shut down and select **Shutdown Daemon**.

Monitoring the Daemon During Runtime

Use the Runtime Manager perspective of Oracle Studio to monitor the daemon during run time.

Perform the following steps:

1. From the **Start** menu, select **Start**, **Programs**, **Oracle**, and then select **Studio**.
2. Right-click the computer defined in Setting Up a Windows Computer in Oracle Studio, and select **Open Runtime Perspective**.

You can manage the daemon by expanding the relevant node, daemon, workspace or server process, as described in the following sections.

Daemon (Computer) Options

Right-click the daemon to display the options available for it, including the ability to display the daemon log.

The following table lists the available daemon options:

Table 5–3 Daemon Options

Option	Description
Edit Daemon Configuration	Opens the daemon editor, which enables you to reconfigure the daemon. See Also: Appendix B, "Advanced Tuning of the Daemon" for details about the configuration settings.
Status	Checks the status of the daemon. The information about the daemon includes the name of the daemon configuration used, the active client sessions, and logging information.
Reload Configuration	Reloads the configuration after any changes. Any servers currently started are not affected by the changed configuration. See Also: Appendix B, "Advanced Tuning of the Daemon" for details about the configuration settings.
View Log	Displays the daemon log. For details see Daemon Logs.
View Events	Displays the daemon events log.
Daemon Properties	Displays information about the computer where the daemon is running, such as the physical address and any username and password needed to access the computer.
Recycle servers	Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.
Kill servers	Immediately closes all active and unused servers. **Note**: It is recommended to use this option with caution, as it may lead to data loss.
Shutdown Daemon	Shuts down the daemon on the computer.
Rename	Enables changing the name of the daemon displayed in the Runtime Explorer.
Remove	Removes the daemon from the Runtime Explorer.
Refresh	Refreshes the display.

Workspace Options

Right-click a workspace to display the options available for the workspace, including the ability to display the workspace log.

The following table lists the available options:

Table 5–4 Workspace Options

Option	Description
Edit Workspace Configuration	Opens the daemon editor to enable you to reconfigure the workspace. See Also: Appendix B, "Advanced Tuning of the Daemon" for details about the configuration settings.
Status	Checks the status of the workspace whether it is available or not.
Reload Configuration	Reloads the configuration of the workspace after any changes. Any servers currently started are not affected by the changed configuration.
View Log	Displays the log for all servers for the workspace. For details see Daemon Logs
View Events	Displays the workspace events log.
Recycle Servers	Closes all unused servers and prepares all active servers to close when the client disconnects. New connection requests are allocated with new servers.
Kill Servers	Immediately closes all active and unused servers. **Note**: It is recommended to use this option with caution, as it may lead to data loss.
Remove	Removes the selected workspace from the Runtime Explorer.
Disable	Disables the selected workspace.
Refresh	Refreshes the display.

Server Options

Right-click a server to display the options available for the server, including the ability to display the server log.

The options available at the server level are listed in the following table:

Table 5–5 Server Options

Option	Description
Status	Checks the status of the server. The information about the server includes the server mode and the number of active client sessions for the server.
View Log	Displays the server log. For details see Daemon Logs.
View Events	Displays the server events log.
Kill server	Ends the server process, regardless of its activity status. **Note**: It is recommended to use this option with caution, as it may lead to data loss.
Refresh	Refreshes the display.

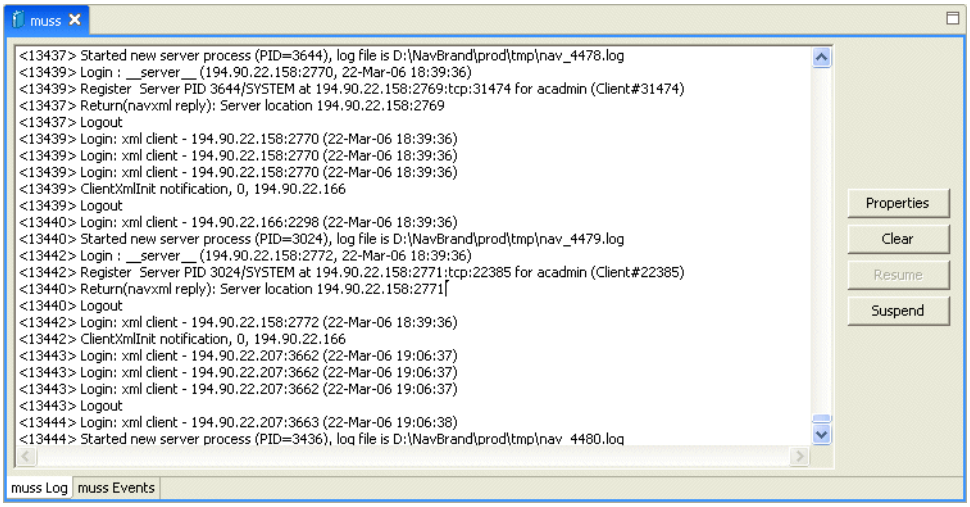
Daemon Logs

Oracle Connect produces a number of logs that you can use to troubleshoot problems. The daemon manages the following logs:

- Daemon

- Workspace
- Server process

The Runtime Manager perspective of Oracle Studio provides a monitor for these logs as shown in the following figure:



To display the required log, do the following:

1. In Oracle Studio, Runtime perspective,right-click, expand the Daemons folder

   If you want to view the workspace log, then expand the daemon with the workspace you want to view.

   If you want to view the server, right-click the workspace with the server you want to view.

2. Right click the daemon, workspace, or server)and select **View Log**.

Each log is displayed in a different tab. You can browse the different logs by clicking the tab at the bottom of the screen.

The Log Monitor

The logs display daemon, workspace, or server events as they happen. You can view the follwoing types of logs in themonitor:

Daemon logs: Display activity between clients and the daemon, including clients logging in and logging out from the daemon.

Workspace logs: Display information about the workspace being used by the client.

Server logs: Display activity between clients and the server process used by that client ot handle the client request.

You can change the logging level. To change the logging level, click **Properties**. The following levels of logging are available in the dialog box:

- **none**: The log displays users that log in and out.
- **error**: The log displays users that log in and out and any errors that have been generated.
- **debug**: The log displays users that log in and out, any errors that have been generated, and any tracing that was configured. For information on configuring the tracing options, see daemon Logging and workspace General.

You can start and stop the logging display.

- Click **Suspend** to stop collecting logging information.
- Click **Resume** to start collecting logging information.

You can remove the inforation displayed in the log.

To remove the information, click **Clear**.

If logging is enabled, new information will continue to be displayed. The cleared information cannot be viewed again.

Resolving Communication Errors

When Oracle Studio disconnects from the Windows computer that you are working with, the computer is displayed in Oracle Studio with an X in a red circle. If this situation occurs, try to access the computer later.

The following table describes the various scenarios that may exist when Oracle Application Server disconnects from the server.

Table 5–6 Scenarios When a Client Is Disconnected

Scenario	Idle (Not Processing a Client Request)	Processing a Client Request
Explicit Disconnect (client explicitly closes connection or client program terminates)	The server is immediately notified of the disconnect and either becomes available for use by another client or terminates (if it is not reusable).	The server does not know that the client has disconnected and continues processing. When processing completes, the server tries to reply to the client and immediately gets an error that the connection was lost. The server either becomes available for use by another client or terminates (if it is not reusable).
Abrupt Disconnect (client closed without proper shutdown or client system hanged and communication disconnected)	The server does not know that the client has disconnected and remains in the idle state. After timing out based on whichever comes first of the value for the client idle timeout daemon workspace parameter or the TCP/IP `KEEPALIVE` parameter, the server is notified of the disconnect and either becomes available for use by another client or terminates (if it is not reusable).	The server does not know that the client has disconnected and continues processing. When processing completes, the server tries to reply to the client. After an interval (typically several minutes, depending on the TCP/IP configuration), during which the TCP/IP subsystem retries sending the message to the client, the server assumes that the client has terminated and notifies the server that the connection has been closed. The server either becomes available for use by another client or terminates (if it is not reusable).

To troubleshoot client/server communication problems, you need to be familiar with the following:

- Daemon configuration settings
- Oracle Connect security
- TCP/IP subsystem. Oracle Application Server Adapter for SQL Server uses TPC/IP for internal intercomputer communications.
- System details, such as the account name and password of the administrator account, the IP address of the computers involved and whether a portmapper is being used.

Resolving Specific Errors

The following error messages relate to errors received from Oracle Connect.

C007: Server initialization failed.

Cause: The daemon failed to start its network service.

Action: Check the processes being run on the system to see whether another daemon or program is using the port specified in the oc4j-ra-xml file for the adapter.

Action: Check the TCP/IP subsystem on the current computer by trying to ping it or run FTP or Telnet to or from it.

Action: Check whether the daemon has privileges to use the TCP/IP services on the current computer with the port specified in the oc4j-ra-xml file for the adapter.

C008: Setting server event handler failed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C009: IRPCD process has been terminated by user request.

Cause: This message is informational only. The daemon successfully shut down.

Action: No action required.

C00A: Application %s not found.

Cause: The requested workspace does not exist.

Action: Check that the workspace defined in the oc4j-ra-xml file is also defined in the daemon configuration on the Windows computer. Use the Status option in the Runtime Manager perspective.

C00B: Invalid IRPCD client context.

Cause: A non-Oracle Connect program is trying to connect to the daemon.

Action: Check the processes and kill the relevant process with a system command.

C00C: Daemon request requires a server login.

Cause: A non-Oracle Connect server or program was trying to use a daemon service reserved for Oracle Connect servers.

Action: Check the processes and kill the relevant process with a system command.

C00D: Daemon request requires a client login.

Cause: The requested daemon requires a valid client login, which was not supplied.

Action: Reissue the command and specify a username and password.

Action: Edit the user profile in Oracle Studio to specify a valid username and password for the Windows compuer with Oracle Connect.

See Also: Setting Up a Windows Computer in Oracle Studio.

C00E: Daemon request requires an administrator login.

Cause: The requested daemon service requires an administrative login.

Action: Edit the daemon security in Oracle Studio to specify a valid administrator username and password.

See Also: Daemon Security.

C00F: Anonymous client logins are not allowed.

Cause: The daemon is configured to require a valid username and password, which were not supplied.

Action: Enable anonymous client access in daemon security in Oracle Studio.

See Also: Daemon Security.

Action: Edit the user profile in Oracle Studio to specify a valid username and password for the Windows computer with Oracle Connect.

See Also: Setting Up a Windows Computer in Oracle Studio.

C010: Anonymous server logins are not allowed.

Cause: Internal error.

Action: Contact Oracle Support Services.

C011: Client has already timed out.

Cause: A server process was started on behalf of a client and the client has timed out before the server completed its startup.

Action: Increase the Connect timeout value for the server workspace in the General tab of the workspace configuration.

See Also: Workspace General.

C012: Invalid username/password.

Cause: Invalid username/password supplied when logging on to the daemon.

Action: See the daemon log file for the reason that the username/password were not accepted.

Action: Edit the user profile in Oracle Studio to specify a valid username and password for the Windows computer with Oracle Connect.

See Also: Setting Up Run-Time User Access.

Action: Make sure the daemon is started from an APF-authorized account that is allowed to check for system usernames and passwords.

C014: Client connection limit reached - try later.

Cause: The maximum number of server processes for the workspace has been reached, and none of the active servers could accept the client connection.

Action: Increase the value of the **Number of sub-tasks** in the Server Mode section of the workspace configuration.

See Also: Workspace Server Mode.

Action: Try running the command later.

C015: Failed to start server process.

Cause: The Oracle Connect daemon failed to start a server process or the started server failed upon starting up.

Action: See the daemon and server logs for the reason the server did not start. For example, you might receive an message with a reason specified in the log file similar to the following: `[C015] Failed to start NAVIGATOR server process: No server account name defined for anonymous client; code: -1601: SQL code: 0`

Action: If you use impersonation, check the user profile on the client. Also see C069.

To set impersonation: APF authorize all the steplibs in the server script on a z/OS computer. For example:

```
setprog... ada622-volume adavol
       CICS.CICS.SDFHEXCI - p390dx
       INSTROOT.load - 111111
       INSTROOT.loadaut - 111111
```

INSTROOT is the high level qualifier where Oracle Connect is installed.

In the **Security** tab of the Navigator workspace, under the daemon node in the Configuration view, select **Use specific workspace account** and clear the **Workspace account** field of all values.

C016: Unexpected server state.

Cause: Internal error.

Action: Contact Oracle Support Services.

C017: Active daemon clients exist. Shutdown canceled.

Cause: One or more clients are still connected to the daemon.

Action: Wait until all the clients log off the daemon and then retry the shutdown operation.

C019: Request is not granted because someone else is locking it.

Cause: A request to lock a resource managed by the daemon was denied because another user has locked the resource.

Action: Wait for the other user to release the resource.

C01A: Lock %s not found.

Cause: A request to free a resource was denied because the caller did not lock that resource (for example, another user shut down the daemon you are working with).

Action: Contact Oracle Support Services.

C01B: Unexpected error in %s.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01C: Cannot update configuration without _APPLICATIONS lock.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01D: Need to lock the application first.

Cause: Internal error.

Action: Contact Oracle Support Services.

C01F: Cannot set configuration of a deleted application.

Cause: Internal error.

Action: Contact Oracle Support Services.

C020: Failed in looking up host name (gethostname())

Cause: Cannot connect to the remote computer.

Action: Check that the name specified for the computer in the oc4j-ra-xml file is correct.

Action: Check that a domain name server (DNS) is available to look up the host name.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

C021: Required variable %s not found

Cause: An environment variable required by the Oracle Connect server was not defined when the server started up.

Action: Check whether the startup script makes any changes to the environment variables used by Oracle Connect.

Action: Check whether the system-defined environment size is sufficiently large for Oracle Connect.

C022: Server failed to connect and register with the daemon.

Cause: An Oracle Connect server started by the daemon was not able to connect or register back with the daemon.

Action: Try to connect again.

Action: Increase the Connect timeout value for the server workspace in the General tab of the workspace configuration.

> **See Also:** Workspace General.

Action: Check that the startup script for the workspace launches the correct version of Oracle Connect.

Action: Increase the value of the **Set maximum number of server processes** parameter for the Clients per server limit in the Server Mode tab of the workspace configuration.

> **See Also:** Workspace Server Mode.

C023: Call made to unregistered module %d.

Cause: Internal error.

Action: Contact Oracle Support Services.

C024: Failed to create a socket.

Cause: An error occurred within the TCP/IP subsystem.

Action: Check whether you have sufficient system privileges.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

C025: Failed to set socket option %s

Cause: An error occurred within the TCP/IP subsystem.

Action: Check whether you have sufficient system privileges.

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

C026: Failed to bind server to port %s

Cause: An Oracle Connect server or daemon was not able to bind to the specified port.

Action: Check whether another program is holding the port that was specified in the oc4j-ra-xml file for the adapter.

Action: Check whether you have sufficient system privileges.

C027: Cannot create TCP service for %s

Cause: An error occurred within the TCP/IP subsystem

Action: Check the TCP/IP subsystem on the computer by trying to ping it or run FTP or Telnet to or from it.

C028: Unable to register (%s, %d, tcp)

Cause: This error may happen when a portmapper is used (*host*:a) but the portmapper is not available.

Action: Enable the portmapper.

Action: Avoid using the portmapper (by not using :a when starting the daemon).

C029: Failed to create a server thread

Cause: Internal error.

Action: Contact Oracle Support Services.

C02A: Server thread failed to start

Cause: Internal error.

Action: Contact Oracle Support Services.

C02B: Stopping the %s server - no client

Cause: A server that was started by the Oracle Connect daemon to service a client did not get a client connection request within one minute. The server terminates.

Action: In most cases, the client was terminated by a user request, so no specific action is required.

Action: If no client can connect to the server, it may be that the server has multiple network cards and the Oracle Connect daemon is not aware of this. In this case, start the daemon with an IP address.

C02C: Unexpected event - a termination signal intercepted

Cause: Internal error.

Action: Contact Oracle Support Services.

C02D: Modified transport, context unknown/lost

Cause: Internal error.

Action: Contact Oracle Support Services.

C02F: Corrupted arguments passed to procedure

Cause: Internal error.

Action: Contact Oracle Support Services.

C030: Unable to free arguments for %s() of %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C031: Cannot register a non-module RPC %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C032: An IRPCD program is required

Cause: Internal error.

Action: Contact Oracle Support Services.

C033: An IRPCD super-server is required for module events

Cause: Internal error.

Action: Contact Oracle Support Services.

C034: An invalid super-server module ID was specified, %d

Cause: Internal error.

Action: Contact Oracle Support Services.

C035: Out of memory

Cause: Not enough memory to service a client request.

Action: Increase process memory quota or add memory to the system.

C036: Failed to register RPC procedure module %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C037: Failed to register an invalid RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C038: Cannot reregister RPC procedure number %x

Cause: Internal error.

Action: Contact Oracle Support Services.

C042: Remote call to %s failed; %s

Cause: Remote call to API failed.

Action: Check the daemon log file.

Action: If necessary, change the level of detail written to the log file to help resolve the problem.

See Also: Daemon Logging.

C043: Failed to connect to host %s;%s

Cause: The remote host is not correctly defined to Oracle Connect or is not working.

Action: Check the remote computer definition in the oc4j-ra-xml file for the adapter.

Action: Check that the daemon is up on the Windos computer with Oracle Connect. Use the Status option in the Runtime Manager perspective.

Action: Check the network connection by trying to ping the host computer or run FTP or Telnet to or from it.

C047: %s out of memory

Cause: Not enough memory was available to Oracle Connect to complete a requested operation.

Action: Terminate unnecessary processes running on the server.

Action: Add more memory to the system.

Action: Allow the process to use more memory.

Action: Limit the number of processes the daemon may start. If the demand for servers exceeds the number of available servers, clients get a message telling them the maximum number of servers has been reached and asking them to try again later.

C066: Communication error with the server%s

Cause: Connection to the Oracle Connect daemon or server failed, or an established session with a server has failed.

Action: Check the remote computer definition in the oc4j-ra-xml file.

Action: Check that the daemon is up on the Windows compuer with Oracle Connect. Use the Status option in the Runtime Manager perspective.

Action: In case of a network problem, check the network connection by trying to ping the host computer or run ftp or telnet to or from it.

C067: Unexpected error occurred in server function %s

Cause: One of the server functions has exited with an exception (such as an abend, or an Invalid Instruction).

Action: Contact Oracle Support Services.

C068: Fail to login daemon

Cause: The daemon is not running on the server computer.

Action: Use the Status in Oracle Studio Runtime Manager perspective to check whether a daemon is running on the server

Action: Have the system administrator reinstall Oracle Connect on the server.

C069: Fail to get server

Cause: The Oracle Connect daemon on the server computer could not start a server process to serve the client. A separate message provides more detail on why the server process could not start.

Action: There are many possible causes of this error. If the cause is not clear from the related message, see the Oracle Connect daemon log file on the server

Action: The resolution to this error is highly dependent on the particular cause. The following are some typical causes and resolutions.

Action: Some process creation quota was exceeded. Either try again later or increase the quota or the other relevant system resources.

Action: The server startup script failed.

Action: The username given is not allowed to use the requested server. Use an authorized username.

Action: A limit on concurrent clients for a server has been reached. Try again later.

Action: If you use impersonation, check the user profile on the client. Also see C015.

C06A: Failed to connect to server

Cause: The server assigned to the client did not accept the client connection. A separate message provides more detail about why the server process did not accept the connection.

Action: See the daemon and server log files for the reason that the server was not available to accept its assigned client.

C06B: Disconnecting from server

Cause: A network failure, or a server computer failure or a server program failure caused the connection to abort. The currently active transaction is aborted as well.

Action: Oracle Connect automatically tries to reestablish a connection with a server upon the next SQL command issued against the server. Once the network or computer failure is corrected, the connection to the daemon is reestablished automatically.

C070: Server failed to send reply to the client

Cause: Server terminated unexpectedly.

Action: Unless the client was intentionally stopped (for example, using Control-C), contact Oracle Support Services.

C071: Connection to server %s was disconnected. Cursors state was lost.

Cause: Either a network failure, a server computer failure or a server program failure caused the connection to abort. The currently active transaction is aborted as well.

Action: Normally, Oracle Connect automatically tries to create a new session with the server upon the next attempt to access the server. If the network and server are accessible, the next operation should succeed. Otherwise, the network or server computer should be fixed before connection can be resumed.

Action: In case of a server crash not related to callable user code, contact Oracle Support Services.

C072: Reconnect to server %s

Cause: This is an informational message only. The client has reestablished its connection with the server.

Action: No action is required.

C073: The parameters passed to the admin server are invalid: %s

Cause: Internal error.

Action: Contact Oracle Support Services.

C074: No authorization to perform the requested operation (%s)

Cause: User/account has insufficient privileges.

Action: Grant administrative privileges to the user/account using the Administrator parameter of the Security tabs in the daemon and workspace configurations.

See Also: Damon Security or Workspace Security.

C075: Failed to register daemon in the TCP/IP service table

Cause: Registration of the daemon in the TCP/IP services file has failed.

Action: Check that the account running the daemon has the permissions to update the TCP/IP services file.

E001: Failed in lock/release operation

Cause: A lock or release operation of a global resource has failed. A separate message provides more details. The separate message specifies the cause of this error.

Action: There are various causes for this error, including lack of sufficient privileges or a system resource shortage.

J0006: Operation on already closed connection was requested

Cause: A request using a connection that was closed was attempted.

Action: Reopen the connection and try again.

J0028: Internal Error: Unknown XML tag %s

Cause: Internal error.

Action: Contact Oracle Support Services.

J0030: Internal Error: Method %s needs to be overwritten

Cause: Internal error.

Action: Contact Oracle Support Services.

J0031: Internal Error: Required attribute %s not found in %s verb

Cause: Internal error.

Action: Contact Oracle Support Services.

J0032: Internal Error: %s ACP object was returned instead of %s as expected

Cause: Internal error.

Action: Contact Oracle Support Services.

J0033: Internal Error: Attempt to work with closed socket

Cause: Internal error.

Action: Contact Oracle Support Services.

J0034: Internal Error: corrupted message; %s bytes read instead of %s as expected

Cause: XML sent from the client to the server has become corrupted.

Action: Check compression settings for XML transferred from the client to the server. If the setting are OK, retry sending the request from the client to the server.

J0035: Internal Error: Invalid redirection address %s returned by daemon

Cause: Internal error.

Action: Contact Oracle Support Services.

J0036: %s: %s

Cause: One of the following errors was received from the server: 0 - server.internalError, 1 - client.xmlError, 2 - client.requestError, 3 - client.noActiveConnection, 4 - server.resourceLimit, 5 - server.redirect, 6 - client.noSuchResource, 7 - client.authenticationError, 8 - client.noSuchInteraction, 9 - client.noSuchConnection, 10 - server.notImplemented, 11 - server.xaProtocolError, 12 - server.xaUnknownXID, 13 - server.xaDuplicateXID, 14 - server.xaInvalidArgument, 15 - client.autogenRejected, 16 - server.xaTransactionTooFresh, 17 - server.resourceNotAvailable, 18 - client.authorizationError, 19 - server.configurationError

Action: Review the server log file to determine the problem.

J0037: Internal Error: No ACP response when %s was expected

Cause: Internal error.

Action: Contact Oracle Support Services.

J0039: Internal Error: ACP root is not found in the XML

Cause: Internal error.

Action: Contact Oracle Support Services.

J0040: Internal Error: Input record is required for interaction %s execution

Cause: Internal error.

Action: Contact Oracle Support Services.

J0048: Invalid metadata type %s is passed to %s function

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

J0050: Key of the put method must be of type string

Cause: In either a GET or PUT operation, the key must be a string.

Action: Change the key used in the operation to a valid key.

J0059: Value %s is invalid for attribute %s

Cause: A request for metadata was not fulfilled.

Action: Check the validity of the request.

J0068: Value must be of type string

Cause: In a PUT operation, the value must be a string.

Action: Change the value used in the operation to a valid value.

J0069: Value must be of type MappedRecord

Cause: In a PUT operation, the value must be a mapped record.

Action: Change the value used in the operation to a valid value.

J0070: Value must be of type MappedRecord[]

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the value used in the operation to a valid value.

J0071: Bad key for mapped record, #element or #element[] is required

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the key used in the record to a valid key.

J0072: Value must be of type Object[]

Cause: In a PUT operation, the value must be mapped record array.

Action: Change the value used in the operation to a valid value.

J0078: In nonpersistent connection and the nonkeep alive encryption is not supported - ignored

Cause: Encryption is not supported for nonpersistent connections.

Action: There is no action to take. This warning can be ignored.

J0079: Invalid argument passed to %s - Argument: %s, Value: %s

Cause: The value pass.

Action: Change the argument used to a number.

6

Advanced Features of OracleAS CDC Adapter for SQL Server

Oracle Connect includes a number of tuning parameters that can improve performance. Specifically, the daemon can be configured to optimize communication between the Windows operating system and a client.

In addition, the binding environment can be tuned to optimize the request handling.

This section contains the following topics:

- Configuring the Daemon for High Availability
- Configuring a Binding Environment
- Migration Considerations
- Security Considerations
- Transaction Support

Configuring the Daemon for High Availability

The daemon workspace is responsible for allocating server processes to clients. You can configure a workspace to use a pool of server processes so that a server process is always available for a client request.

Use Oracle Studio to maintain daemon and daemon workspace parameters to control the allocation of server processes and their management in a pool.

You can also have a number of daemon workspace configurations. This lets you create individual workspaces for use with different adapters or CDC adapters.

Adding a New Daemon Workspace Configuration

Use Oracle Studio to add a new daemon configuration. You can set up different daemon configurations for different situations.

Carry out the following steps to add a new daemon workspace configuration:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Design Perspective Configuration view, expand the **Machines** folder and then expand the machine where you want to add the workspace.
3. Expand the **Daemons** folder. The daemon available on this computer are listed.
4. Right-click **IRPCD** and select **New Workspace**. The New Daemon Workspace screen is displayed.

5. Enter a name for the new workspace and then enter a description, if desired.
6. Select whether to use default settings or copy the properties of an existing workspace.

   To copy the properties of an existing workspace, click the Browse button and select the workspace from which you want to copy the properties.
7. Click **Next**. The Select Scenario screen is displayed.
8. Select **Application Server using connection pooling** and click **Next**.
9. Continue through the wizard, entering the required values for the workspace.
10. Click **Finish**.

The workspace is displayed under the **IRPCD** daemon node.

Editing the Workspace

You edit a workspace by using the tabs described in the following table:

Table 6–1 Workspace Properties tabs

Tab	Description
General	Specifies general information including the server type, the command procedure used to start the workspace, the binding configuration associated with this workspace (which dictates the data sources and applications that can be accessed) the timeout parameters, and logging information (which dictates the data sources and applications that can be accessed), the timeout parameters, and logging information.
Server Mode	Contains the workspace server information including features that control the operation of the servers started up by the workspace and allocated to clients.
Security	Specifies administration privileges, user access, ports available to access the workspace and workspace account specifications.

Use Oracle Studio to access these tabs, as follows:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Design perspective Configuration view, expand the **Machines** folder and then expand the machne where you want ot edit the workspace.
3. Expand the **Daemons** folder. The daemon available on this computer are listed.
4. Expand the **IRPCD** daemon. The daemon workspaces are listed.
5. Right-click the workspace you are editing and select **Open**.
6. Click the tab that contains the information you want to edit. For full details of the tabs and the fields in these tabs, see Workspaces.
7. After editing the workspace, click **Save**.

Configuring the Server Mode

The server mode defines how the daemon starts up new processes. The daemon supports the following server modes:

- **singleClient**: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace.

  This mode enables servers to run under a particular user account and isolates clients from each other (because each receives its own process). However, this server mode incurs a high overhead due to process startup times and may use a lot of server resources (because it requires as many server processes as concurrent clients).

- **multiClient**: Clients share a server process and are processed serially. This mode has low overhead because the server processes are already initialized. However, because clients share the same process, they may impact one another, especially if they issue lengthy queries.

  The number of clients that share a process is determined by the Clients per server limit (the maximum number of concurrent clients a server process for the current workspace accepts).

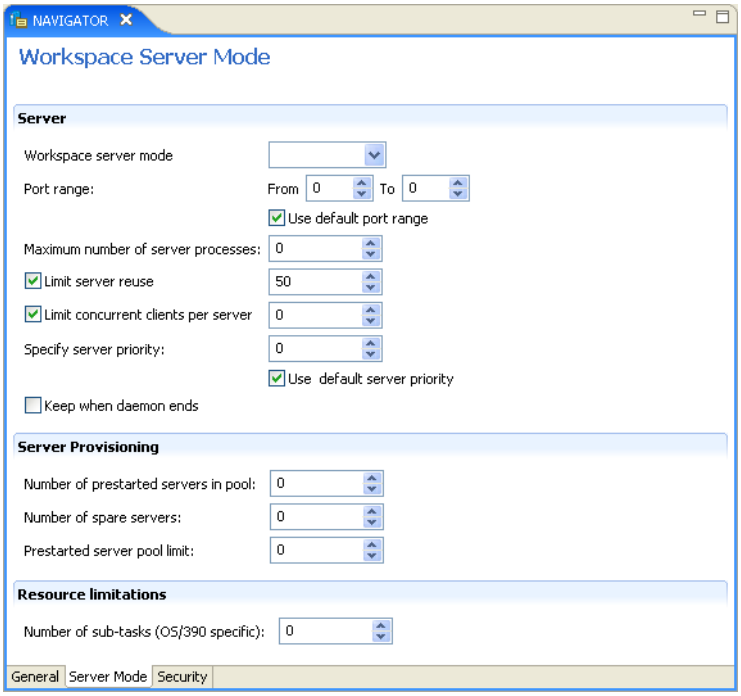
- **reusable**: This is an extension of the single client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead.

  This mode does not have the high overhead of single client mode because the servers are already initialized. However, this server mode may use a lot of server resources (because it requires as many server processes as concurrent clients).

  The other modes can be set so that the server processes are reusable by setting the number of times a process can be reused with the `Reuse limit` value (the maximum number of times a particular server process can be reused or how many clients it can serve before it is retired). Reuse of servers enhances performance because it eliminates the need to repeat initializations. However, reuse runs a risk of higher memory leakage over time. The default value for the `Reuse limit` field is `None`, indicating that no reuse limit is enforced.

Set the server mode in the **Server Mode** tab of the daemon workspace editor as shown in the following figure:

Figure 6–1 The Server Mode tab



When using any of the server modes you can specify a pool of server processes. These server processes are started when the daemon starts and are maintained in a pool. The server processes are available for use by new client requests from the pool, saving initialization time. Instead of starting a new server process each time one is requested by a client, the client receives a process immediately from the pool of available processes. When the client finishes processing, this server process either dies, or if reusable servers have been specified, it is returned to the pool.

You set up a pool of server processes by specifying the following parameters in the **Server Mode** tab.

- **Port Range**: Select the range for specific firewall ports through which you access the workspace. Determines the range of ports available for this workspace when starting server processes. Use this option when you want to control the port number, so that Oracle Connect can be accessed through a firewall.

  Enter the port range in the following fields:

  - **From**: Enter the highest numbered port in the range
  - **To**: Enter the lowest numbered port in the range

- **Use Default Port Range**: Select this to use the port range that is defined in the daemon.

- **Maximum number of server processes**: Enter the maximum number of server processes that can run at the same time.

- **Limit server reuse**: Select this if you want to limit the number of servers that can be reused. If this is selected, the **Reuse limit** parameter is available.

  If **Limit server reuse** is selected, in the field next to the check box, enter the maximum number of times a server can be reused. Select the maximum of clients accepted in a server process.

A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization.

This parameter is not available if the Limit server reuse parameter is not selected.

This parameter is not available if the server mode value is **singleClient**.

- **Limit Concurrent clients per server**: Select this to limit the number of clients that a server can accept for the current workspace process.

  If this is not selected, the number of clients is unlimited.

  If **Limit concurrent clients per server** is selected, in the field next to the check box, enter the maximum number of clients that a server process for the current workspace accepts. The default for this field is **None**, indicating that the number of clients for each server is unlimited. This field is available if the server mode value is **multiClient** or **multiThreaded**.

- **Specify Server Priority**: Set the priority for servers. For example, a workspace for applications with online transaction processing can be assigned a higher priority than a workspace that requires only query processing. The lower the number, the higher the priority. For example, workspaces with a priority of `1` are given a higher priority than workspaces with a priority of `2`.

  Note: This is unavailable if **Use default server priority** is selected.

- **Use default server priority**: Sets the priority to 0. There is no specific priority for this workspace. Clear this check box to set a priority in the **Specify Server Priority** parameter.

- **Keep when daemon ends**: Select this to kill all servers started by that daemon when a daemon is shutdown, even if they are active. Select this if you want the servers for the workspace to remain active, even after the daemon has been shut down. If selected, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level.

- **Number of prestarted servers in pool**: The number of server processes that are prestarted for this workspace when the daemon starts up. These are available for use by new client processes with minimal initialization time. Instead of starting a new server process each time one is requested by a client, the daemon immediately allocates (to the client) a server from a pool of available servers. When the number of available server processes drops lower than the value specified in the Minimum number of available servers field, the daemon again starts server processes until the specified number of available servers is reached. The default for this parameter is `0`, meaning that no servers are prestarted for this workspace.

- **Number of spare servers**: The minimum number of server processes in the prestarted server's pool before the Oracle Connect daemon resumes creating new server processes (up to the number specified in the Initial number of servers field value, described earlier). If this parameter is set to a value greater than the Initial number of servers field value, the daemon considers the value to be the same as the value specified in the Initial number of servers field. In this case, a new server process is started and added to the pool each time a server process is removed from the pool and allocated to a client). The default for this parameter is `0`, which means that new servers are created only when there are no other available servers.

- **Number of sub-tasks**: The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts up. In addition to setting up a pool of server processes as described earlier, you can set additional server processes as

sub-tasks by specifying this parameter. If you set 10 servers and 10 prestarted sub-tasks then 100 tasks are started (10 sub-tasks for each process).

Configuring a Binding Environment

Each binding configuration includes the following information:

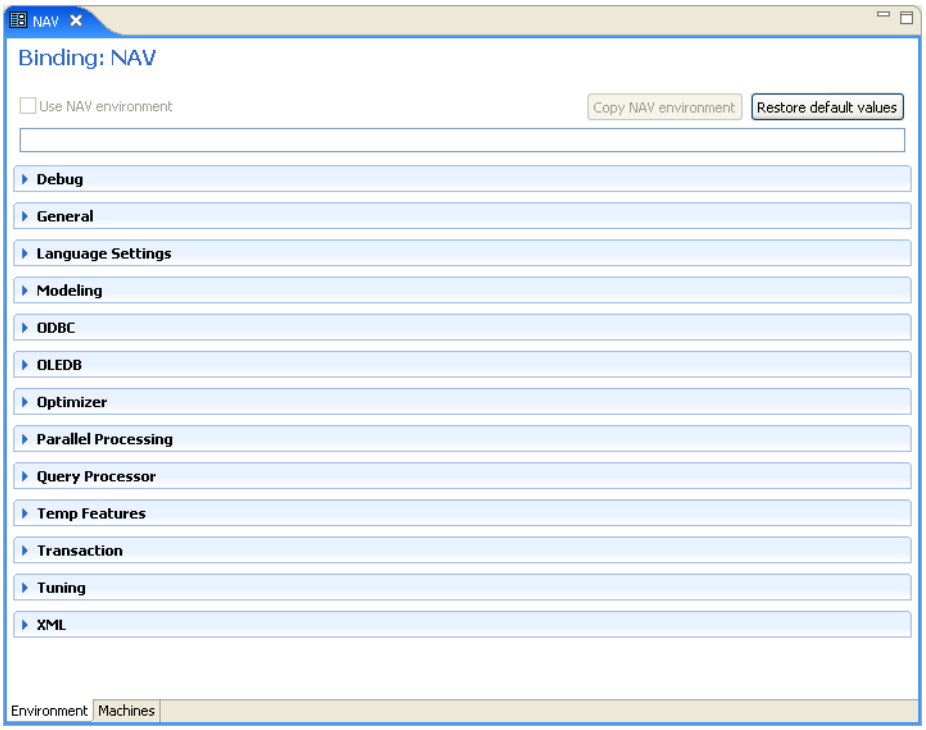
- Environment settings, which are used to configure the environment used by any of the adapters or CDC adapters defined in the binding.
- Application adapters on the current computer.

To configure environment settings in Oracle Studio, perform the following steps:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Design perspective Configuration view, expand the **Machine** folder, then expand the machine where you want to configure the binding.
3. Expand the **Bindings** folder. The binding available on this computer are listed.
4. Right-click **NAV** and select **Open**.
5. In the **Environment** tab, edit the environment settings as needed. To edit an environment setting, expand the property category and click the value to edit.

   The binding **Environment** tab is shown in the following figure:

Figure 6–2 The binding Properties tab



The binding environment is divided into the following categories:

- Debug
- General
- Language

- Modeling
- ODBC
- OLEDB
- Optimizer
- Query Processor
- Transaction
- Tuning
- XML

Debug

The following table lists the parameters that define debugging and logging operations:

Table 6–2 Debug Parameters

Parameter	Description
ACX trace	Select this for the the input xml sent to the back-end adapter and the output xml returned by the back-end adapter to be written to the log.
GDB Trace	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
General trace	Select this to log general trace information. The default writes only error messages to the log. **Note**: Changing the default setting can degrade performance.
Query warnings	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
Add timestamp to traced events	Select this to add a timestamp on each event row in the log.
Query Processor trace	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
Binary XML Log Level	Select the binary XML log level from the list. The following logging levels are available: ■ None ■ API ■ Info ■ Debug
Log file	The high-level qualifier of the log file for messages. The following type of message are written to the log: ■ Error messages ■ Trace information and information about the query optimization strategy, if **General Trace** is selected.
Trace Directory	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
Optimizer trace	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
Transaction extended logging	Select this for the transaction manager to write additional information about transactions to the log.

General

The following table lists the parameters that define various operations the directory where temporary files are written.

Table 6–3 General Parameters

Parameter	Description
NAV_UTIL editor	This parameter is not applicable for use with OracleAS CDC Adapter for SQL Server.
Temporary Dir	The directory where temporary files are written, including the temporary files created for use by hash joins and for sorting files. The default is the current high-level qualifier.
Year 2000 policy	Determines the way two-digit years are converted into four-digit years. When the `year2000Policy` parameter is not set, or when it is set to a value outside the range of values defined for the policy, as described in the following paragraphs, a default value of `5` and the Sliding Base Year policy is used. Two policies are provided: Fixed Base Year: `year2000Policy` is set to a value greater than, or equal to 1900. In this case, the value of `year2000Policy` is the first four-digit year after 1900 that can be represented by a two-digit year. For example, if `year2000Policy` is set to 1905, the years 2000->2004 will be represented by 00->04. All other two digits will map to 19xx. This solution is most required if there is live data at the low end (close to the year 1900), which the user wants to keep with the current two-digit format. The user will probably change the base date only after ensuring that these old dates have been deleted from the data source. Sliding Base Year: `year2000Policy` is set to a positive value less than 100. In this case, the value of `year2000Policy` represents the number of years ahead of the current year that can be represented by a two-digit number. With each passing year the earliest year that can be represented by a two-digit number changes to a year later.
Cache buffer size	Enter the number of bytes to be used for a memory buffer on a client machine, which is used by the Oracle Connect client/server to store read-ahead data. The default is `200000`

Language

The following table lists the parameters that define globalization support:

Table 6–4 Language Settings Parameters

Parameter	Description
Language	Identifies the application language. A default codepage is selected based om the value specified for this parameter. See also: Appendix D, "Globalization Settings".
Code Page	For use with globalization support to identify the codepage for the workspace. See also: Appendix D, "Globalization Settings".

Table 6–4 (Cont.) Language Settings Parameters

Parameter	Description
NLS String	Specifies the codepage used by a field whose data type is defined as `nlsString`. Use this for a field whose codepage is other than that of the computer codepage. This parameter includes the following values: ■ The name of the codepage. ■ Whether the character set reads from right to left (as in middle eastern character sets). The default is `false`.

Modeling

The **Modeling** parameters are not applicale with OracleAS CDC Adapter for SQL Server.

ODBC

The **ODBC** parameters are not applicable for use with OracleAS CDC Adapter for SQL Server.

OLEDB

The **OLEDB** parameters are not applicable for use with OracleAS CDC Adapter for SQL Server.

Optimizer

The following parameters enable you to customize the performance:

Optimizer goal: The optimization policy. Select one of the following from the list to the right:

- **none** (default): All row optimization is used.
- **first**: First row optimization is performed based on the assumption that the results produced by the query are used as the rows are retrieved. The query optimizer uses a strategy that retrieves the first rows as fast as possible, which might result in a slower overall time to retrieve all the rows.
- **all**: Optimization is performed based on the assumption that the results produced by the query are used after all the rows have been retrieved. The query optimizer uses a strategy that retrieves all the rows as fast as possible, which might result in a slower time to retrieve the first few rows.

Aggregate queries automatically use all row optimization, regardless of the value of this parameter.

All other optimizer parameters are not applicable for use with the OracleAS CDC Adapter for SQL Server.

Parallel Processing

This following list shows the parallel processing properties. The parallel processing properties control how parallel processes are handled in the binding.

Table 6–5 Parallel Processing Parameters

Parameter	Description
Disable threads	.Select this to disable multi-threading. If this is selected, the following properties are disabled: ■ **Disable threaded read ahead (QP)**: Select this to disable read-ahead functionality. ■ **Disable query read ahead (QP)**: Select this to disable read-ahead functionality for components using Query Processor services. ■ **ODBC async execution** property to enable ODBC asynchronous execution ■ **Disable QP parallel execution**: Select this to disable parallel processing for query execution. This option is available only if both **Disable threaded read ahead (QP)** and **Disable query ready ahead (QP)** are *not* selected. ■ **Hash parallelism**: Select this to read both sides of hash joins at the same time. By default, this property is selected. If you do not want this behavior, clear the check box.

Query Processor

The following table lists the parameters that enable you to fine tune how queries are processed:

Table 6–6 Query Processor Parameters

Parameter	Description
Disable command reuse	Select this to disable Query Processor cach;ing the executed state of a query for reuse.
Disable DS property cache	Select this to disable caching data source properties.
Disable insert parameterization	Select this to disable parameterization constants in INSERT statements.
Disable metadata caching	Select this to disable caching object metadata. If this is selected, the object metadata is taken from the from the original data source instead of the cache.
Disable query parametarization	Select this to not convert constants into parameters when accessing data sources.
Disable row mark field fetch	Select this for OLE DB `getRows` errors to be marked and reshown on every `getRows`, if the rowset is active.
Compile after load	Select this to always compiles an Oracle Connect procedure or view after it is read.
Ignore segments bind failure	This property determines how Oracle Connect responds when the execution of one of the segments of a segmented data source fails: ■ Select this to Log a message and continue execution. This is the default setting. ■ Clear the check box to Log a message and stop execution. By default, this property is selected. If you want to stop execution after sending a message, clear this check box.
Prompt database-user password	Select this to configure Oracle Connect to prompt the user for security information when accessing a data source.

Table 6–6 (Cont.) Query Processor Parameters

Parameter	Description
Use alternate qualifier	Select this to use the @ symbol instead of a colon (:) when connecting to multiple data sources.
Use table filter expression	Select this to enable the use of tables that have filter expressions specified in their metadata.
Write empty string as null	Select this to replace empty strings in a SET clause of an UPDATE statement or in a VALUES list of an INSERT statement with null values.
Optimistic for update	Select this to use optimistic locking as the default locking behavior on queries with a FOR UPDATE clause.
Disable compilation cache	Select this to disable saving successfully compiled statements in the cache.
Maximum SQL cache	Enter the maximum number of SQL queries that can be stored in cache memory. This property's value is ignored if **Disable compilation cache** is selected. The default is 3.
First tree extensions	Enter the maximum size allowed for an SQL query after compilation. The default is 150.
Maximum columns in parsing	Enter the maximum number of columns that a query can reference. The default is 500.
Maximum segmented database threads	Enter the maximum number of open threads allowed, when working with segmented databases.
Minimum number of parameters allocated	Enter the minimum number of parameters that can be used in a query.
Continuous query retry interval	Enter the number of seconds that the query processor waits before executing a query again, when no records are returned. The default is 2.
Continuous query timeout	Enter the number of seconds that the query processor will continue to issue queries, when no records are returned. The default is 3600 (one hour), which indicates that after an hour without new messages the continuous query will end. Enter 0 to indicate that there is no timeout and the continuos query will not end automatically.
Continuous query prefix	Enter a prefix to replace the $$ prefix that is used to identify the continuous query special columns. For example, if you enter ##, then the continuous query alias is '##StreamPosition' and the control command alias is '##ControlCommand'.

Table 6–6 (Cont.) Query Processor Parameters

Parameter	Description
Arithmetic fixed precision	Enter an integer determine the precision scale factor for floating decimal position. The default is 0, which indicates that the exact arithmetic function is not used. When the value is set to a small positive integer, the special precise floating point arithmetic is used in the query processor. The value determines the precision scale factor (for example, a value of 2 indicates two digits decimal precision). Setting this parameter can be done at a workspace level and it affects all queries running in that workspace with no change to the query or to the underlying data source. The query processor ADD(), SUBTRACT() and SUM() functions that currently use double arithmetic for both floating and decimal types will use this logic. When the value is set to the default, 0, the exact arithmetic function is not used. This property is used to set the Exact Arithmetic function. The `qpArithmeticFixedPrecision` property is an integer value that determines the fixed precision the Oracle Connect query processor uses for precise floating point arithmetic. It is used to create an accurate result when using the SUM function. Because floating point datatypes are not accurate their results over time does not correspond to the expected arithmetic sum. In other words, in the floating point representation, values such as 0.7 cannot be represented precisely. If there are eight precision digits, there is usually imprecision in the least significant digit so the number is actually approximately 0.699999995. The`qpArithmeticFixedPrecision` property corrects this imprecision by using an exact floating point.
Parser depth	The maximum depth of the expression tree. The default is 500
Token size	Enter the maximum length of a string in an SQL query. The minimum value is 64. The default value is 350.
Insert from select commit rate	Enter the commit rate to use when executing an INSERT-FROM-SELECT operation. If a value more than 0 is entered, a commit is performed automatically after inserting the indicated number of rows. For example, if the value is 5,a commit is performed every time 5 rows are inserted.
Disable SQS cache	Select this to always read compiled Oracle Connect procedures and views from a disk. In this case, they are not saved in the cache.
Procedures cache size	Enter the number of stored queries created with a CREATE PROCEDURE statement that can be kept in cache memory. This property's value is ignored if **Disable SQS cache** size is selected.
Expose XML fields	Expose XML fields: Select this to display data returned for a query as XML, representing the true structure of the result. This is useful when querying a data source table that contains arrays or variants. ■ **XML field name**: Enter the name used in a query to indicate that the data is returned as XML, instead of the keyword XML. This is available only if **Expose XML fields** is selected.

Transaction

The following table lists the parameters that define transaction support:

Table 6–7 Transaction Parameters

Parameter	Description
Transaction extended logging	Select this to write extended information about transactions to the transaction manager log files.
Commit on destroy	Select this to commit all single-phase commit transactions opened for a data source, if a connection closes while the transaction is still open.
Disable 2PC	Select this to disable two-phase commit capabilities, even in drivers that support two phase commit.
User commit confirm table	This parameter is not applicable for use with OracleAS CDCAdapter for SQL Server.
Transaction log file	The high-level qualifier and name of the log file that logs activity when using transactions.
Recovery Delay	The number of minutes from the start of a transaction before any recovery operation on that transaction can be attempted. The default is `15 minutes`.
Time limit	Enter the time to wait for a transaction to complete before an error is returned. This parameter is also used when performing a RECOVERY, and it then indicates the number of minutes to wait before a forced activity can be performed, since the last transaction activity.
Conversions Select one:	
	No conversion: Select this if you want all transactions to remain as sent. This is selected by default **Convert all to distributed**: Select this to convert all simple transactions into distributed transactions. **Convert all to simple**: Select this to convert all distributed transactions into simple transactions.

Tuning

The following table lists the parameters that define the tuning:

Table 6–8 Tuning Parameters

Parameter	Description
Dsm maximum buffer size	Enter the maximum size of a cache memory. This cache is used when memory is required on a temporary basis (as when Oracle Connect sorts data for a query output, for a subquery, or for aggregate queries). This cache size is not used for hash joins and lookup joins (see the `hashBufferSize` parameter). The default value is `1000000` bytes.
Dsm maximum Sort buffer size	Enter the maximum size of the sort buffers. Use this parameter instead of `dsmMaxBufferSize` for sorts only. The default value is `1000000` bytes.

Table 6–8 (Cont.) Tuning Parameters

Parameter	Description
Dsm middle buffer size	Enter the maximum size of the index cache. This cache is not used for hash joins and lookup joins. The default value is `100000` bytes.
File pool size	Enter the maximum number of files that can be opened in the file pool. The default is 10.
File pool size per file	Enter the size of the file in the pool. The default is 3.
File close on transaction	Select this if you want the File Pool to close when a transaction is committed.
Use global file pool	Select this to use a global file pool is. When the workspace server mode parameter is set to multiClient or reusable, this parameter also indicates whether the file pool closes upon the client disconnection.
Hash buffer size	Enter the amount of cache memory that is available for each hash join or lookup join. The default is `1000000` bytes.
Hash max open files	Enter the maximum number of files that a query can open at one time for use when performing hash joins. The number assigned to this parameter must not exceed the system maximum. The default isset to `90` files. **Note**: The hash join optimization strategy results in a number of files being opened to perform the join. The larger the table size, the more files are opened. By adjusting this parameter you can disable hash joins on very large tables, while allowing hash joins for small tables.
Hash primary extent size	Enter the primary extent size.
Hash secondary extent size	Enter the secondary extent size.
Hash enable RO	Select this for the QP to store the first hash bucket in memory instead of a sequential file.

XML

The following table lists the parameters that define XML support:

Table 6–9 XML Category Parameters

Parameter	Description
COM maximum XML in memory	Specifies the maximum size of an XML document held in memory. The default is `65535` bytes.
COM maximum XML size	Specifies the maximum size of an XML document passed to another computer. The default is `65535` bytes. **Note**: When you increase this value for this property, you may need to increase the value for the **Maximum XML in memory** property in the daemon. For more information on daemons, see Control.
COM XML transport buffer size	Enter the maximum size of the internal communications buffer. The default value (`-1`) indicates there is no size limit.

Table 6–9 (Cont.) XML Category Parameters

Parameter	Description
XML date format	Enter the date format to use for XML. The options are: ■ `ISO` (the default): The date format is: `YY-MM-DDThh:mm:ss[.ss..]` ■ `ODBC`: The date format is: `YYYY-MM-DD HH:MM:SS[.NNN...]`
Replace invalid XML characters	Select this to replace invalid XML characters with a `'?'`. This property is used for diagnostic and troubleshooting purposes.
XML trim char columns	Select this to enable padded spaces to be trimmed from XML string columns when the record format is fixed. By default this is selected, and padded spaces are trimmed for fixed size character columns. If you do not want this behavior, clear this check box.

Migration Considerations

You can migrate an adapter configuration from one platform to another. The configuration information is stored in the Oracle Connect repository on the source platform and is exported to an XML file which can then be imported to the target platform.

Note that when migrating a configuration, any file names and paths that are specific to the source platform must be changed to valid files on the target platform.

To migrate an adapter configuration using Oracle Studio, perform the following steps:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. In the Design perspective, Configuration view, expand the **Machines** folder.
3. Right-click the required computer, and select **Export XML definitions**.
4. In the **File name** field, type the path and name of the XML file where the XML representation of the computer and its complete configuration is stored.
5. Edit any paths in the XML definition to the paths required on the target platform. For example, the setting for the serverLogFile might need changing, depending on the platform.
6. Set up the target platform in Oracle Studio in the same way you set up the source platform, as described in Setting Up a Windows Computer in Oracle Studio.
7. In the Configuration Explorer, right-click the target computer and select **Import XML definitions**.
8. Import the XML file to the target platform.

Security Considerations

Oracle Connect works within the confines of the platform security system.

In addition, Oracle Connect provides the following security:

- A binary XML encryption mechanism, which is activated as follows:
  1. The client's first message to the server includes a pre-defined shared key, includeing the key name and value in the connection string. The server gets the key value for the key name passed from the client from the native object store (NOS).

2. The server generates a random 128-bit RC4 session key which is returned encrypted to the client, using the shared key. If no predefined shared key is provided, then a predefined, hardcoded key is used (this key is hardcoded on the client and on the server).
3. Passwords are always encrypted when passed over the wire, using an RC4, 128-bit session key, regardless of whether the entire session is encrypted or not.
4. If a predefined shared key was provided, then the entire session is encrypted. Otherwise, only the password exchange is encrypted (using the hardcoded key).

- Credentials: Passwords and usernames exchanged over the network are encrypted using a pre-defined, hardcoded, 128-bit RC4 session key.
- Design Time: Security within Oracle Studio to grant access to Oracle Studio itself and to grant access to computers, user profiles and workspaces.
- Run time: Security used to access SQL Server, including controlling the daemon for the access.

Setting Design Time Security

Setting design time security is described in the following sections:

- Securing access to Oracle Studio is described in Setting Password Access to Oracle Studio.
- Securing rights to configure a computer in Oracle Studio is described in Specifying Users with Administrative Rights.
- Securing access to user profiles is accomplished by right-clicking the relevant user profile in Oracle Studio and selecting **Change Master Password**. In the dialog box that is displayed, specify a password that must be provided in the future to edit the specific user profile.
- Securing access to workspaces is accomplished by right-clicking the relevant workspace in Oracle Studio and selecting **Set Authorization**. In the dialog box that is displayed, specify a valid user and password that must be provided in the future to edit the specific workspace.

Setting Runtime Security

During run time, security considerations are implemented as follows:

- When the client request accesses the legacy platform through the daemon, either anonymous access is allowed or a valid user name and password must be provided for the computer in the user profile. The `userName` and `password` properties in the J2CA 1.5 SQL Server adapter are used at this stage to access the daemon.

  Note: The user name used to access the daemon must also be the name of a user profile used.

- Access by the client must be through a valid port, according to the port range entered in the **Server** section of the workspace **Server Mode** tab in Oracle Studio. For more information, see the explanation of the workspace Server Mode.

> **Note:** Access to the legacy platform through a firewall using the NAT protocol is specified when the computer is added to Oracle Studio.

- To be allocated a server process, the client must be granted anonymous access to the workspace or be listed in the **Authorized Workspace Users** section of the workspace **Security** tab in Oracle Studio. For more information see the explanation of the workspace Security tab.
- The ability to run commands on the daemon, such as starting or stopping a daemon or ending server processes is available only to administrators who have been registered in Oracle Connect as a daemon administrator. A client is registered as a valid daemon administrator in theAdministrator provileges section of the daemon **Security** tab in Oracle Studio. For more information, see explanation of the daemon Security tab.

> **Note:** You can also specify administrators who can run commands only at the level of the workspace. Select these administrators in the workspace **Security** tab, as described in the explanation of the workspace Security.

Transaction Support

OracleAS CDC Adapter for SQL Server supports global transactions and can fully participate in a distributed transaction.

To work with global transactions, select **Convert all to distributed** in the Transaction section of the binding environemtal properties. See Transaction in the Configuring a Binding Environment section.

You can use SQL Server with its two-phase commit capability both under MTS, and directly through an XA connection. In both cases, Microsoft DTC must be running on the server.

If you are working under MTS, start an OLE transaction. The SQL Server data source is automatically included in the distributed transaction.

If the connection to the data is through an XA connection, the connection is made automatically. The daemon server mode must be configured to Single-client mode (see Server Mode).

To use distributed transactions from an ODBC-based application, ensure that AUTOCOMMIT is set to 0.

A

Advanced Tuning of the Metadata

Oracle Studio enables you to work directly with the data source metadata and to define outbound adapter interactions. In addition, Oracle Studio defines input and output structures used for these interactions. The interactions and input and output structures are maintained as metadata by Oracle Studio in the **Metadata** tab of the Design perspective.

This appendix contains the following sections:

- Metadata for the SQL Server Data Source
- Adapter Metadata

Metadata for the SQL Server Data Source

Using Oracle Studio, perform the following steps to maintain the metadata for the SQL Server data source:

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. Expand the **Machines** folder, then expand the CDC Server machine that you defined when you Set up the CDC Server.
3. Expand the **Bindings** folder.
4. Expand the binding with the name of your CDC solution and the suffix `_ag`.
5. Expand the **Data sources** folder to display the data sources.
6. Right-click the SQL Server data source and select **Show Metadata View** to display the **Metadata** view.
7. Right-click the table you want to view in the Metadata Explorer and select **Edit**.

The metadata editor opens, displaying the **General** tab, with general table details. The following tabs are used to view and edit the metadata:

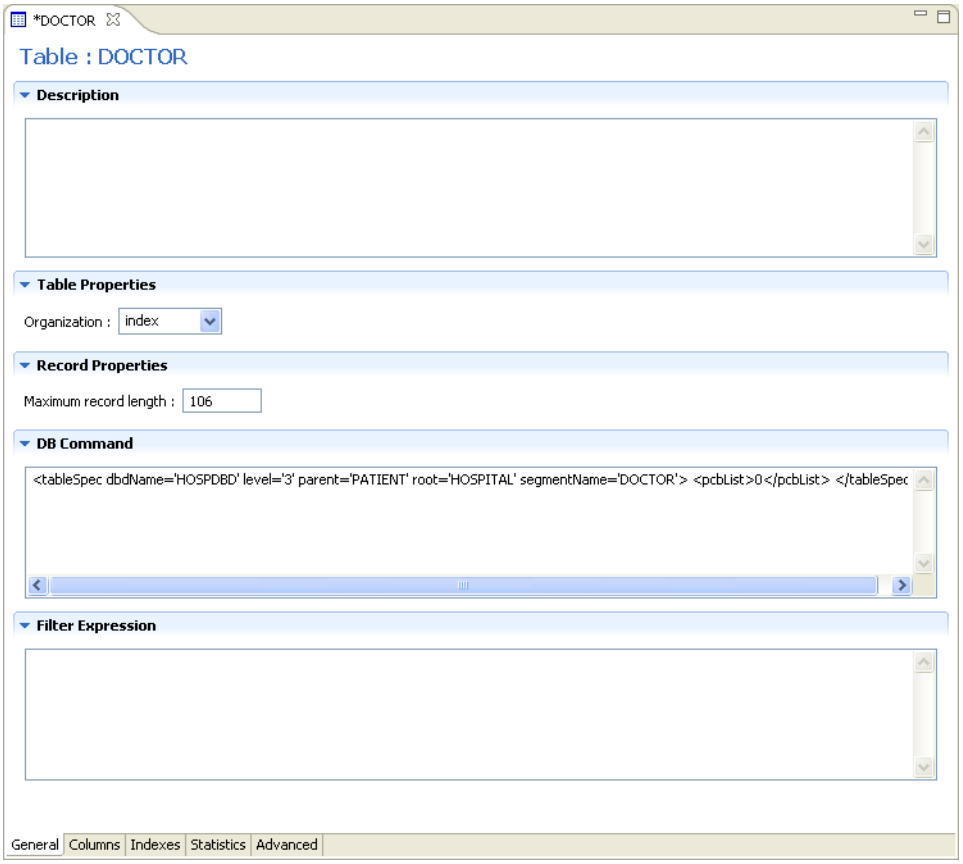
- General Tab
- Columns Tab
- Indexes Tab
- Statistics Tab
- Advanced Tab

General Tab

Use the **General** tab to maintain information about the whole table, such as the table name and the way the table is organized.

The **General** tab is shown in the following figure:

Figure A–1 The General Tab



The **General** tab comprises fields, as listed in the following table:

Table A–1 General Tab Components

Field	Description
Description	An optional description of the table.
Table Properties	
Data file location	The name of the file that contains the table. You must enter the full path and include the file extension for the file. For example, `D:\COBOL\orders.cob`. You can click Browse and browse to find and enter the location of the table file. Note: Do not enter the file extension for DIASM or CIASM files.

Table A–1 (Cont.) General Tab Components

Field	Description
Organization	Select how the record represented by the table is organized. The options that are displayed depend on the record. The following options are available: ■ Index ■ Sequential ■ Relative: Used with RRDS files. Access to a specific record number of a relative file is performed by using a pseudo column to specify the record position. The hash symbol (#) is used to specify a pseudo column. For example: `SELECT * FROM colleges WHERE # = 6`
Record format	Specifies how the record, represented by the table, is formatted. The options that are displayed depend on the record. The SQL Server records have a fixed format.
Maximum record length	The maximum allowable size of a record (in bytes).
DB Command	SQL Server specific commands are displayed in the DB Command section of the **General** tab. The information displayed is generated automatically when the metadata is generated and should not be modified.
Filter Expression	A `WHERE` clause. This clause is added to every query accessed using this metadata. The filter is specified without the `WHERE` keyword. Specify a filter when more than one logical table is stored in the same physical file.

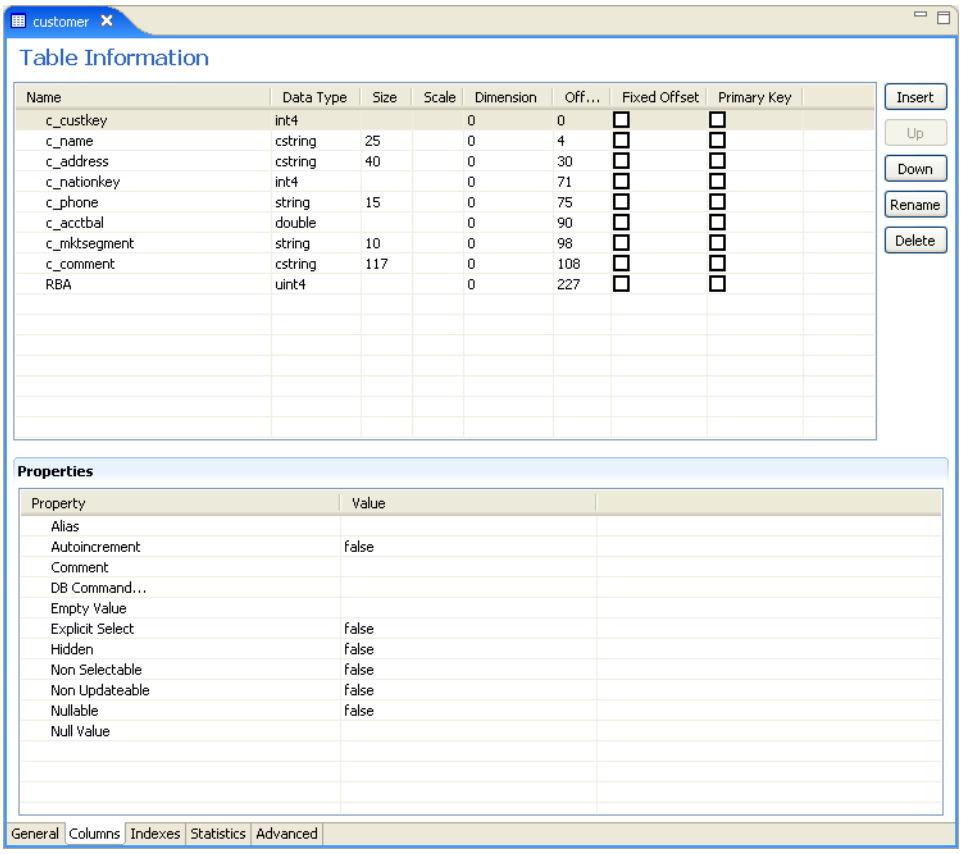
Columns Tab

Use the **Columns** tab to specify metadata that describe the table columns. The tab is divided into the following:

- Column Definition Section
- Column Properties

The **Columns** tab is shown in the following figure:

Figure A–2 The Columns Tab



Column Definition Section

The top section of this tab lets you define the columns in the source data. You can click in any row (which represents a column in the data base table) to edit the information. The following table describes this section.

Table A–2 Metadata Column Tab Definitions

Field name	Description
Name	The name of the column
Data type	The data type of the column. Selecting this field displays a drop-down box listing the possible data types.
Size	The size of the column
Scale	The information entered in this field depends on the data type: For decimal data types, this is the number of digits to the right of the decimal place. This number must not be greater than the number of digits. The default value is **0**. For scaled data types, this is the total number of digits. The number must be negative.
Dimension	The maximum number of occurrences of a group of columns that make up an array. The (+) to the left of a column indicates a group field. This type of field will have a Dimension value. Click (+) to display the group members.
Offset	An absolute offset for the field in a record.

Table A–2 (Cont.) Metadata Column Tab Definitions

Field name	Description
Fixed offset	This column lets you determine whether to calculate the offset. There are two options: ■ Calc offset: If you clear this check box, the absolute offset for each of the columns is calculated. ■ Fixed offset: When you select this check box, you will have a fixed offset. The offset of a field is usually calculated dynamically by the server at runtime according the offset and size of the proceeding column. Select the check box in this column to override this calculation and specify a fixed offset at design time. This can happen if there is a part of the buffer that you want to skip. By selecting the check box, or by editing the offset value you pin the offset for that column. The indicated value is used at runtime for the column instead of a calculated value. Note that the offset of following columns that do not have a fixed offset are calculated from this fixed position.
Primary Key	Select this to indicate that this column is a primary key.

The buttons on the right side of the tab are used to manipulate the data in this section of the tab. The following table describes how you can move around in this section.

Table A–3 Definition Section Buttons

Button	Description
Insert	Inserts a column to the table. You can insert a new column. If the table has arrays, you can add a new child column.
Up	Moves your selection to the column directly above where the currently selected column.
Down	Moves your selection to the column directly below where the currently selected column
Rename	Lets you rename the selected column.
Delete	Deletes the selected column.
Find	Click this button to open a list of all columns in the database. Select a column and click **OK** to select it in the table.

Column Properties

You can change the property value by clicking in the **Value** column. Follow these steps for displaying the column properties.

To display the column properties

- Select a column from the Column Definition (top) section.

  The properties for the column are displayed at the bottom of the tab.

The following table shows some of the properties available for selected columns.

Table A–4 Metadata Properties

Property	Description
Alias	A name used to replace the default virtual table name for an array. Virtual table names are created by adding the array name to the record name. When an array includes another array the name of the nested array is the name of the record and the parent array and the nested array. When the default generated virtual table name is too long, use an Alias to replace the long name.
Autoincrement	The current field is updated automatically by the data source during an `INSERT` statement and is not explicitly defined in the `INSERT` statement. The `INSERT` statement should include an explicit list of values. This attribute is used for fields such as an order number field whose value is incremental each time a new order is entered to the data source.
Comment	A short note or description about the column.
DB command	SQL Server specific commands for the column The information displayed is generated automatically when the metadata is generated and should not be modified
Empty value	The value for the field in an insert operation, when a value is not specified.
Explicit Select	When `true`, the current field is not returned when you execute a `SELECT * FROM...` statement. To return this field, you must explicitly ask for it in a query, for example, `SELECT NATION_ID, SYSKEY FROM NATION` where `SYSKEY` is a field defined with Explicit Select. You cannot use an asterisk (*) in a query where you want to retrieve a field defined with the Explicit Select value.
Hidden	The current field is hidden from users. The field is not displayed when a `DESCRIBE` statement is executed on the table.
Non Selectable	When `true`, the current field is never returned when you execute an SQL statement. The field is displayed when a `DESCRIBE` statement is executed on the table.
Non Updateable	If `true`, the current field cannot be updated.
Nullable	This value allows the current field to contain `NULL` values.
Null value	The null value for the field during an insert operation, when a value is not specified.
Chapter of	This property shows that the set member field is a chapter of an owner field. A value for this property must be used when accessing a set member as a chapter in an ADO application. This property is used for DBMS metadata
OnBit	The position of the bit in a `BIT` field and the starting bit in a `BITS` field.
Subfield of	The value is generated automatically when you generate metadata from SQL Server data that includes a superdescriptor based on a subfield. A field is created to base this index on, set to the offset specified as the value of the `Subfield start` field. If no value is entered in the `Subfield start` field, the subfield is set by default to an offset of 1.
Subfield start	The offset within the parent field where a subfield starts.

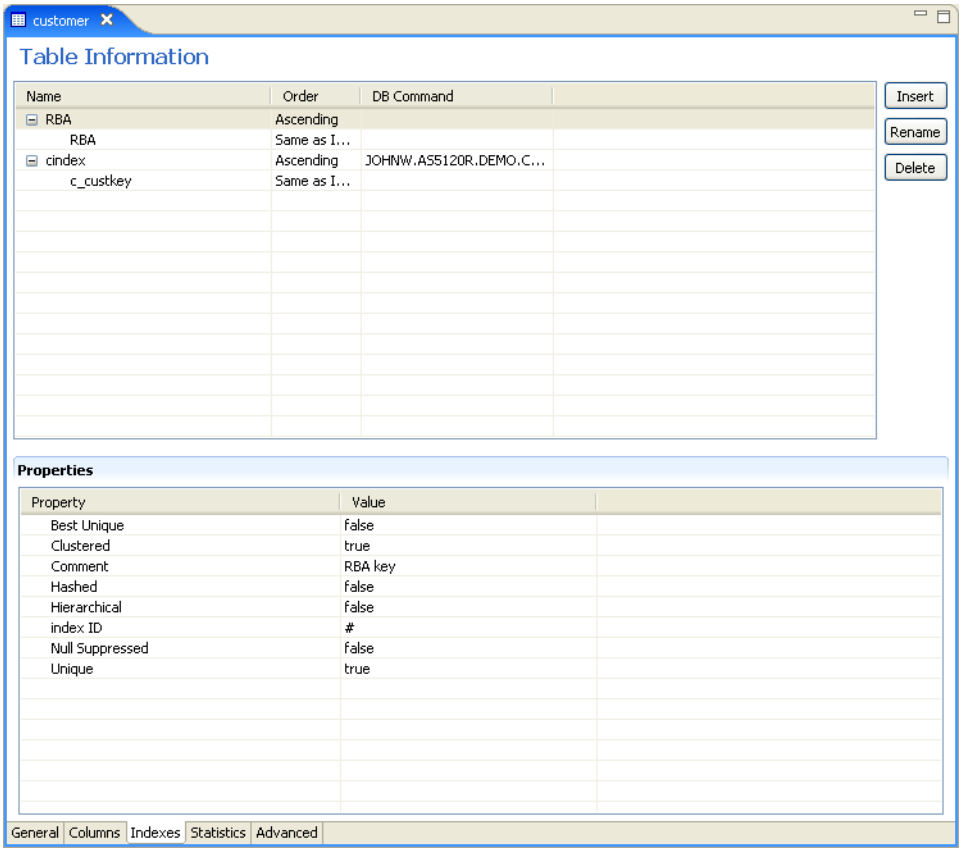
Indexes Tab

Use the **Indexes** tab to specify metadata describing the indexes of a table.

> **Note:** The Indexes tab contains information only if the **Organization** field in the Table tab is set to **Index**.

The **Indexes** tab is shown in the following figure:

Figure A–3 The Indexes Tab



This tab has two sections. The first section lets you define the index keys for the columns in the table. The bottom of the tab lists the properties for each of the columns at the top.

The following sections describe the **Indexes** tab:

- Table Information
- Properties

Table Information

The following table describes the fields for the top part of the tab, which defines the indexes used for the table.

Table A–5 Indexes Tab Components

Field	Description
Name	The names of existing indexes for the current table.
Order	The ordering of the rows retrieved by the index.
DB Command	SQL Server-specific commands for the index. The information displayed is generated automatically when the metadata is generated and should not be modified.

The buttons on the right side of the tab are used to manipulate the data in this section of the tab. The following table describes how you can move around in this section.

Table A–6 Index Definition Buttons

Button	Description
Insert	Inserts an index to the table.
Rename Index	Lets you rename the selected index.
Delete	Deletes the selected index.

Properties

You can index properties for each index column. Follow these steps for displaying the properties for each index.

To display the index properties

- Select a column from the Index Definitions (top) section.

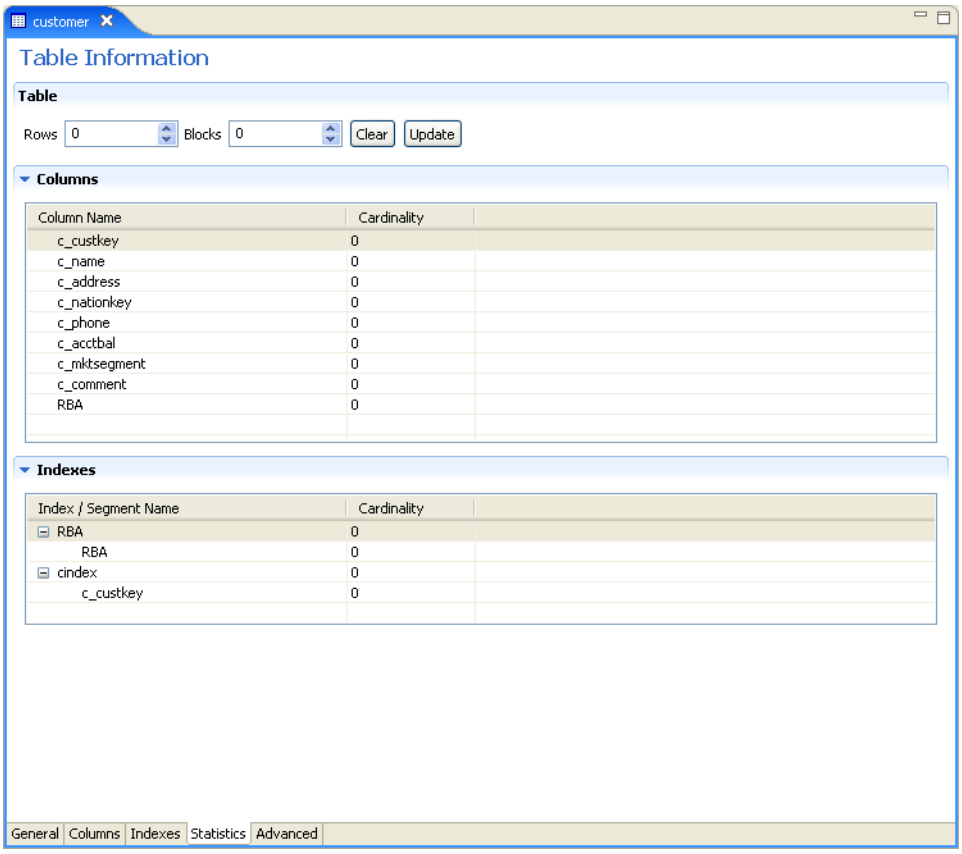
  The properties for the column are displayed at the bottom of the tab.

This properties displayed at the bottom of the tab describe the index or segment. The properties available depend on the data source.

Statistics Tab

Use the **Statistics** tab to update metaedata statistics for a table. The **Statistics** tab is shown in the following figure:

Figure A–4 The Statistics tab



The **Statistics** tab has three sections. The following tables describe each section.

Table A–7 Table Information

Field	Description
Rows	Enter or use arrows to select the approximate number of rows in the table. If the value is `-1`, then the number of rows in the table is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that this table is empty.
Blocks	Enter or use arrows to select the approximate number of blocks in the table. **Note**: If neither the number of rows nor the number of blocks is specified for a table, queries over the table might be executed in a nonoptimal manner.

Use the Columns group box to specify cardinality for each of the columns in the table:

Table A–8 Columns group Components

Field	Description
Column name	The columns in the table.

Table A–8 (Cont.) Columns group Components

Field	Description
Cardinality	The number of distinct values for the column. If the value is `-1`, then the number of distinct values for the column is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that there are no distinct values for the column.

Use the Indexes group box to specify cardinality for the columns in each of the indexes in the table:

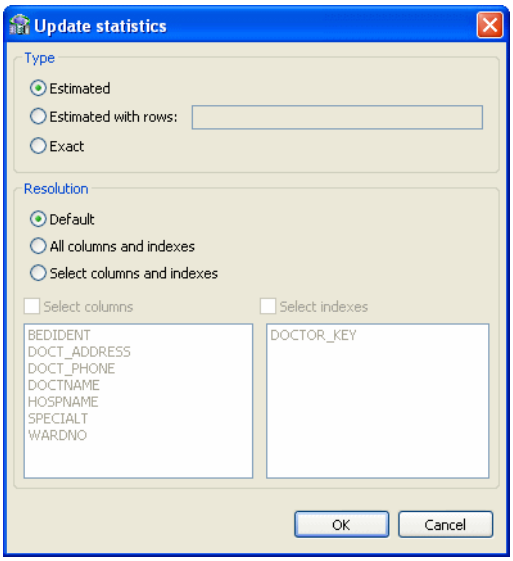
Table A–9 Indexes group Components

Field	Description
Indexes and segments	The indexes and segments in the table.
Cardinality	The number of distinct key values in the index. If the value is `-1`, then the number of distinct key values in the index is unknown (a value was not supplied and the update statistics utility was not run to update the value). A value of `0` indicates that there are no distinct key values in the index.

Generating Statistics

Click **Update** in the **Statistics** tab to generate updated statistics for the table. The Update Statistics screen is displayed, as shown in the following figure:

Figure A–5 The Update Statistics screen



The following tables list the fields that are used to update statistics:

The **Type** section lets you edit the following:

Table A–10 Type Components

Field	Description
Estimated	An estimation of the amount of statistical information returned.
Estimated with rows	An estimation of the amount of statistical information returned. The estimate includes an estimation of the number of rows in the table. Specify the number in the text box. This number is used to shorten the time to produce the statistics, assuming that the value specified here is the correct value, or close to the correct value.
Exact	The exact statistical information returned. Note that this can be a lengthy task and can lead to disk space problems with large tables.

The **Resolution** section lets you specify the statistical information returned:

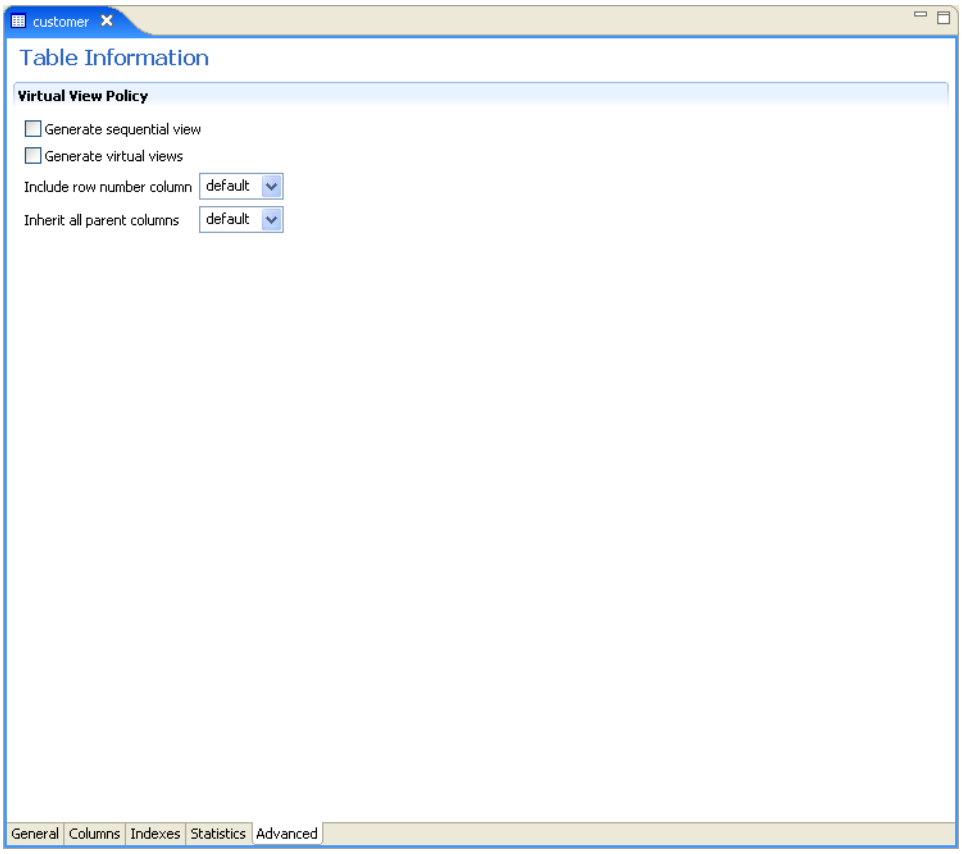
Table A–11 Resolution Components

Field	Description
Default	Only information about the table and indexes is collected. Information for partial indexes and columns is not collected.
All columns and indexes	Information about the table, indexes, partial indexes and columns is collected.
Select columns and indexes	Enables you to select the columns and indexes for which you want to collect statistics. In the enabled list of columns or indexes, left click those columns you want included (you can use shift-click and control-click to select a number of columns or indexes).

Advanced Tab

The **Advanced** tab lets you enter information about the virtual view policy for arrays. These parameters are valid only if you are using virtual array views. The configurations made in this editor are for the selected table, only. The same parameters are configured on the data source level in the data source editor.

Figure A–6 Data Source Metadata Advanced Tab



Enter the following information in this tab:

- **Generate sequential view**: Select this to map non-relational files to a single table.
- **Generate virtual views**: Select this to have individual tables created for each array in the non-relational file.
- **Include row number column**: Select one of the following:
  - **true**: Select **true**, to include a column that specifies the row number in the virtual or sequential view. This is true for this table only, even in the data source is not configured to include the row number column.
  - **false**: Select **false**, to not include a column that specifies the row number in the virtual or sequential view for this table even if the data source is configured to include the row number column.
  - **default**: Select **default** to use the default data source behavior for this parameter.
- **Inherit all parent columns**: Select one of the following:
  - **true**: Select **true**, for virtual views to include all the columns in the parent record. This is true for this table only, even in the data source is not configured to include all of the parent record columns.
  - **false**: Select **false**, so virtual views do not include the columns in the parent record for this table even if the data source is configured to include all of the parent record columns.

– **default**: Select **default** to use the default data source behavior for this parameter.

Adapter Metadata

Use Oracle Studio to maintain the metadata for your CDC Queue adapter.

1. From the **Start** menu, select **Programs**, **Oracle**, and then select **Studio**.
2. Expand the **Machines** folder, then expand the machine with your staging area.
3. Expand the **Bindings** folder.
4. Expand the binding for your staing area (with the suffix `_sa`).
5. Expand the **Adapters** folder to display the adapters list.
6. Right-click the CDC Queue adapter, and select **Show Metadata View** to open the Metadata view.

   You can create and edit the adapter metadata as follows:

   - Adapter Metadata General Properties: Enter and edit information about the adapter, such as the adapter name and the way in which you connect to the adapter. You make these changes in the Design perspective, Metadata view.
   - Adapter Metadata Interactions: Enter details of an interaction. The interaction Advanced tab is displayed for some adapters only, such as the Database adapter and includes more details about the interaction.
   - Adapter Metadata Schema Records: The input and output record structure for a record in the adapter definition.

Adapter Metadata General Properties

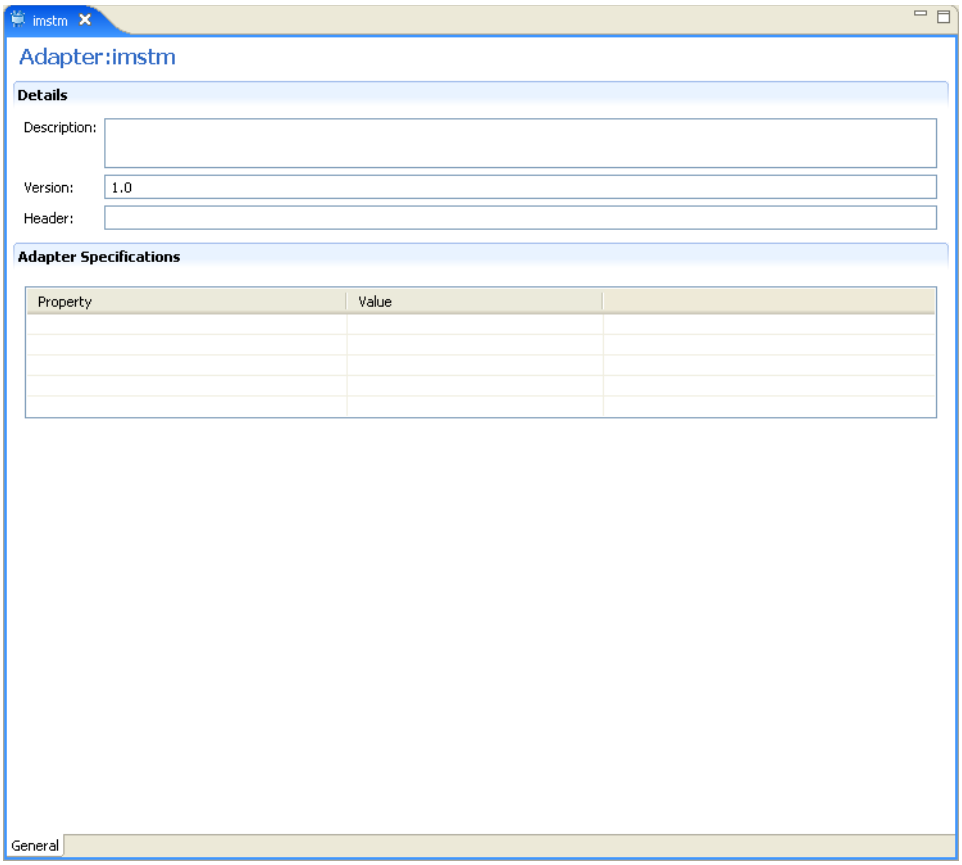
You can enter and edit information about the adapter, such as the adapter name and the way in which you connect to the adapter. You make these changes in the Design perspective, Metadata view. The following describes how to open the Adapter General Properties editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Right-click the adapter that you want to edit, and select **Open**.

   The General properties editor is displayed.

The Adapter General Properties editor is shown in the following figure:

Figure A–7 Adapter Metadata General Properties



The **General** properties are listed in the following table:

Table A–12 General tab Components

Field	Description
Description	Specifies an identifying description of the adapter.
Authentication mechanism	Specifies the authentication to access the adapter. The available mechanisms are: ■ kerbv5 ■ none ■ basic password
Max request size	Specifies the maximum size in bytes for an XML request or reply. Larger messages are rejected with an error.
Max active connections	Specifies the maximum number of simultaneous connections for an adapter (per process).
Max idle timeout	Specifies the maximum time, in seconds, that an active connection can stay idle. After that time, the connection is closed.
Adapter Specifications	Specifies the adapter-specific properties for an interaction. The SQL Server CDC Queue adapter does not have any adapter-specific properties.

Adapter Metadata Interactions

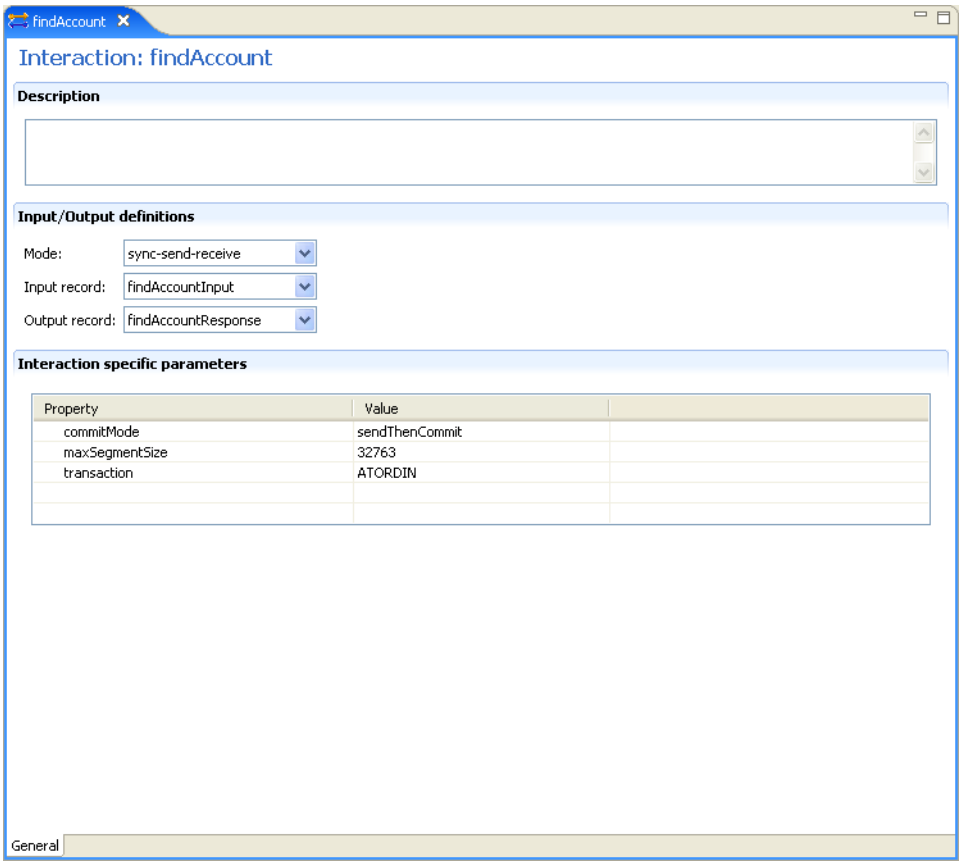
The Adapter Metedata Interactions editor defines an interaction and its input and output definitions. The following describes how to open the Adapter Metadata editor.

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the adapter with the interaction that you want to edit.
3. Expand the **Adapter** folder.
4. Right-click the adapter you want to edit and select **Open**.

   The Adapter Metadata Interactions editor is displayed.

The Adapter Metadata Interactions editor is shown in the following figure:

Figure A–8 Adapter Metadata Interactions



The Adapter Metadata Interaction properties are listed in the following table:

Table A–13 Interaction General tab Components

Field	Description
Description	Provides a descriptive identifier for the interaction.

Table A–13 (Cont.) Interaction General tab Components

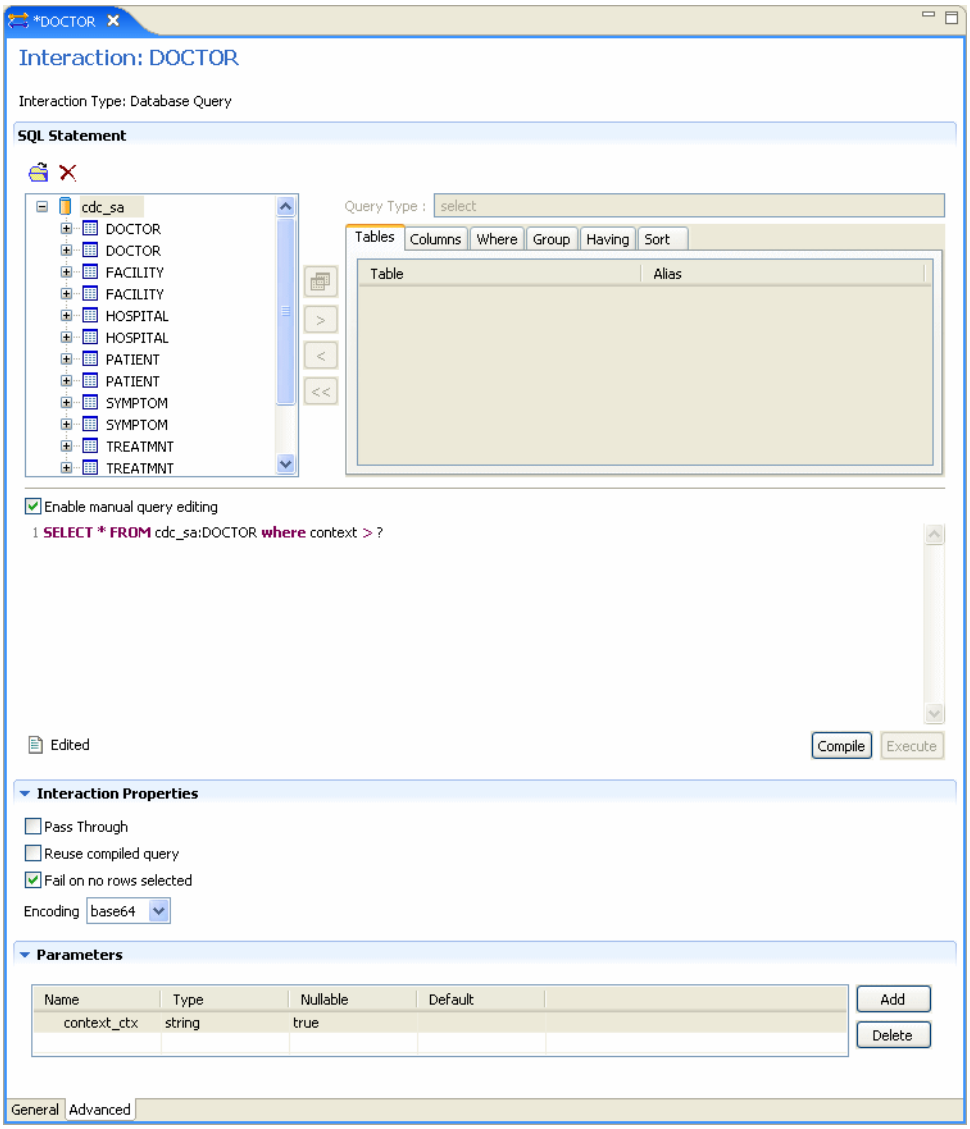
Field	Description
Mode	Determines the interaction mode. The following interaction modes are available: ▪ sync-send-receive: The interaction sends a request and expects to receive a response. ▪ sync-send: The interaction sends a request and does not expect to receive a response. ▪ sync-receive: The interaction expects to receive a response.
Input record	Identifies an input record.
Output record	Identifies an output record for the results of an interaction.
Interaction Specific Parameters	Specific properties for the interaction. When an Interaction **Advanced** tab is used, this section is not displayed.

Interaction Advanced Tab

In the Adapter Metadata Interactions editor, click **Advanced** at the bottom of the editor to open the Interaction **Advanced** tab. Use this to enter advanced details for the interaction or to create interaction manually.

The following figure shows the Interactions **Advanced** tab.

Figure A–9 Interaction Advanced Tab



This tab has three sections.

The **SQL Statement** section lets you build any valid SQL statement. Use the tabs to select the tables and statement types and build a statement. When you select **Enable manual query editing**, you can manually enter a valid SQL statement at the bottom of the screen.

The **Interaction Properties** section lets you select any of the following:

- **Pass Through**: Select this to pass a query directly to the SQL Server data.
- **Reuse compiled query**: Select this to save query objects created in the previous execution to the cache. This allows the objects to be reused.
- **Fail on no rows returned**: Select this if you want to the system to return an error if no rows are selected.
- **Encoding**: Select one of the following from the list:
  - **base64**: Select this for base 64 encoding

  - **hex**: Select this for hexadecimal encoding

The **Parameters** section lets you create parameters to use for the interaction. To create a parameter, click **Add**. Enter a name in the dialog box and click **OK**. The parameter is entered in the list. You can edit the following properties for each parameter you create. Click in the corresponding cell to edit the properties.

- **Name**: The name of the parameter. This is automatically entered when you create a new parameter. You can click in the cell to change this parameter.
- **Type**: The type of parameter. Select one of the following types from the list:
  - **string**
  - **number**
  - **timestamp**
  - **binary**
  - **xml**
- **Nullable**: Select True or False to determine whether the parameter can be nullable.
- **Default**: Enter a default value for the parameter, which is used if the parameter attribute is missing in the input record.

Notes:

- If a field is not nullable and a default value is not supplied in the schema part of the Adapter Definition, an error occurs if the parameter attribute is missing in the input record.
- The parameters must be entered in the same order as they are used in the SQL statement.

Adapter Metadata Schema Records

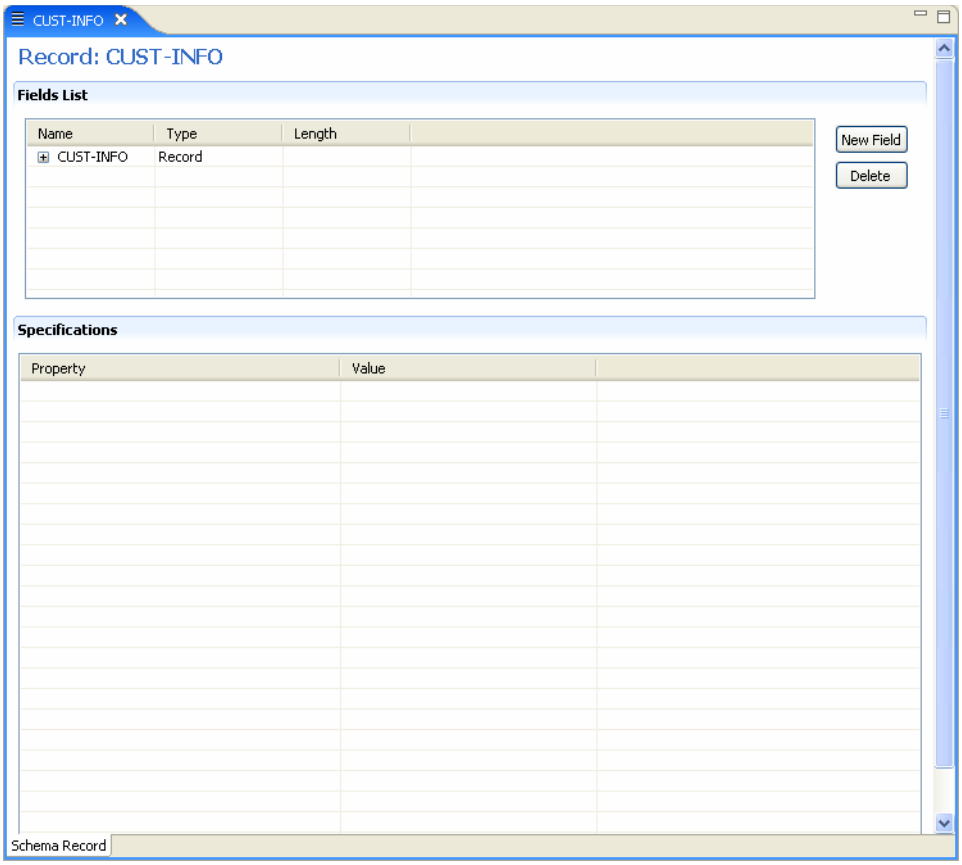
The Adapter Metadata Schema Records editor defines the general details of the input and output record structures for the interaction.The following describes how to open the Adapter Metadata Schema Records editor:

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Expand the adapter with the schema records that you want to edit.
3. Expand the **Schemas**.
4. Right-click the schema you want to edit and select **Open**.

   The Adapter Metadata Schema Records editor is displayed.

The Adapter Metadata Schema Records editor is shown in the following figure:

Figure A–10 Adapter Metadata Schema Records



The Adapter Metadata Schema Records properties are listed in the following tables:

Table A–14 Schema Record Tab

Field	Description
Fields list	Defines the single data item within a record. This section has a table with the following three columns: ▪ **Name**: The name of the field ▪ **Type**: The data type of the field. See the Valid Data Types table for a list of the valid data types. ▪ **Length**: The size of the field including a null terminator, when the data type supports null termination (such as the string data type).
Specifications	Defines specific field properties. To display the properties, select the specific field in the Fields list.

The following table describes the valid data types that can be used when defining these specifications in the Schema Record editor.

Table A–15 Valid Data Types

Binary	Boolean	Byte
Date	Double	Enum
Float	Int	Long

Table A–15 (Cont.) Valid Data Types

Binary	Boolean	Byte
Numeric[(p[,s])]	Short	String
Time	Timestamp	

Adapter Metadata XML

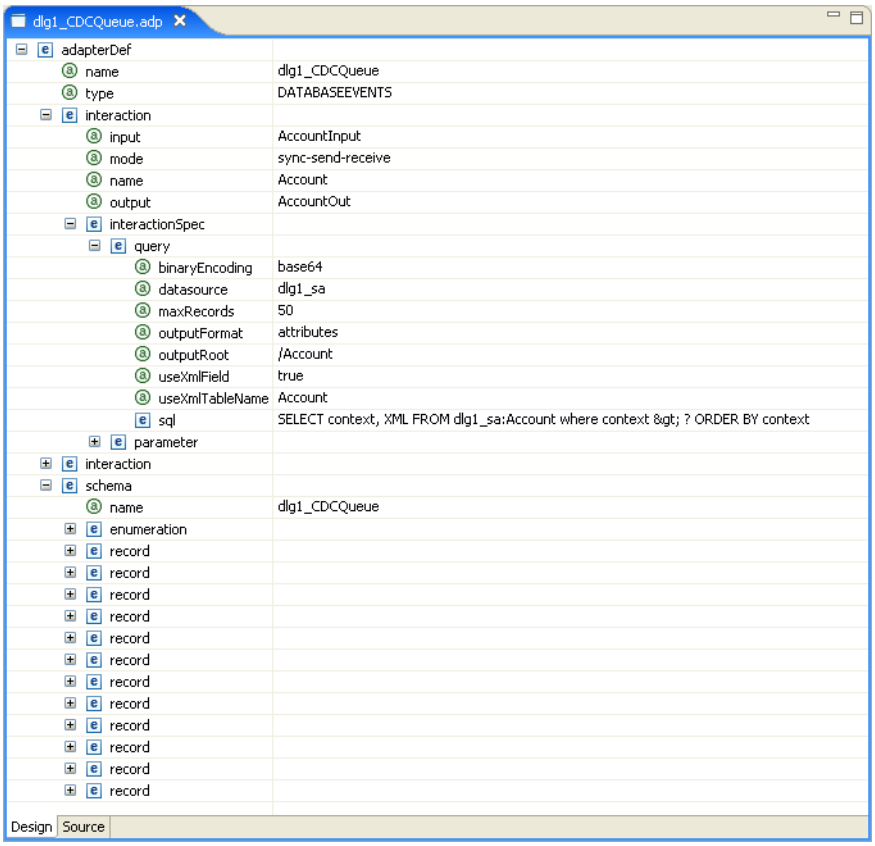
You can also edit the adapter metadata by viewing its XML schema. The following describes how to open the adapter metadata XML editor:

1. In Oracle Studio Design perspective, Metadata view, expand the **Adapters** folder.
2. Right-click the adapter that you want to edit, and select **Open as XML**.

   The adapter XML editor is displayed in the Design view.

The adapter XML editor is shown in the following figure:

Figure A–11 Adapter Metadata XML Editor



For information on how to edit the properties in the XML editor Design view, see Appendix E, "Editing XML Files in Oracle Studio".

See CDC Queue Adapter Properties for an explanation of which attributes you can edit for the CDC Queue adapter.

B

Advanced Tuning of the Daemon

The daemon configuration is managed using Oracle Studio. Daemon configuration is divided into the following groups:

- Control
- Logging
- Security
- Workspaces

Control

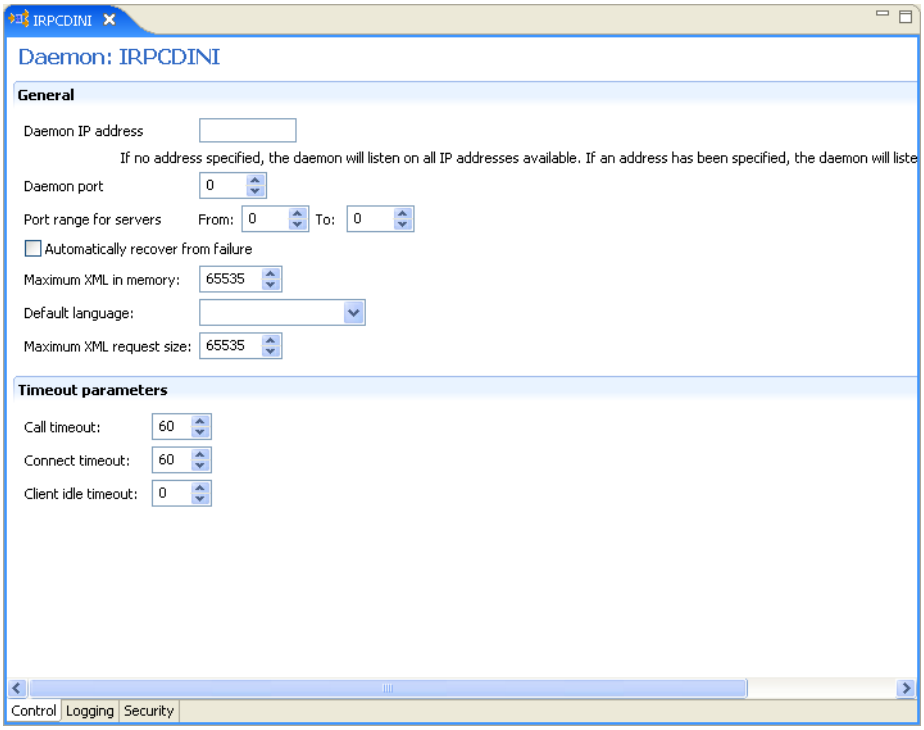
Using the **Control** tab for the daemon, you define various daemon control options. The **Daemon Control** tab is accessed as follows:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the required daemon in the Runtime Explorer and select **Edit Daemon Configuration**. The **Control** tab for the damon is displayed in the editor.
5. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

> **Note:** You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
>
> Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

The **Control** tab is shown in the following figure:

Figure 6–3 The Control tab



The following table shows the paremters that can be set in the **Control** tab:

Table 6–10 Daemon Control tab Components

Field	Description
Daemon IP Address	Enter the IP address of the machine(s) where the daemon is listening. If no IP address is entered, the daemon will listen on all available IP addresses.
Daemon port	Enter the port where the daemon is listening. If no port is entered, the daemon listens on all available ports.
Automatically recover from failure	The daemon restarts automatically if it fails for any reason (any error that causes the daemon process to terminate, such as network process lost or the CPU running the daemon crashes and the backup daemon is defined on another CPU). All available and unconnected servers are terminated and any connected servers are marked and terminated on release. Also the backup starts a backup for itself. The backup appends a new log file to the log of the original daemon, adding a line indicating that a backup daemon was started.
Maximum XML request size	The maximum number of bytes that the daemon handles for an XML document.
Default language	The language that the daemon supports. This setting is used when working with a client with a code page different from the server code page.
Maximum XML in memory	The maximum amount of space reserved for the XML in memory.

Table 6–10 (Cont.) Daemon Control tab Components

Field	Description
Call timeout	The timeout period for *short* calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as `DESCRIBE` should be completed in a few seconds as opposed to call like a `GETROWS` call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds. Specifying the timeout in a workspace overrides the value set in this field for that workspace.
Connect timeout	The time the client waits for a daemon server to start. If the daemon server does not start within this period, then the client is notified that the server did not respond. The value specified for this parameter serves as the default timeout for all the workspaces listed in the daemon configuration. The default value for this parameter is 60 seconds. **Notes**: ■ Entering the timeout in a workspace overrides the value set in this field for that workspace. ■ Even if the XML source does not list this parameter in the workspace section, the workspace gets it using the default value. If you want to prevent a workspace from using the default value, you must enter a value of zero for this parameter in the workspace section.
Client idle timeout	The maximum amount of time any daemon client may be idle before the connection with the server is closed. Specifying the timeout in a Workspace overrides this setting for that workspace.

Logging

Using the **Logging** tab, you define the daemon log file settings, the log file structure and the location where the log is saved. In addition, use it to define the data that is logged and traced in the file.

The following describes how to open the **Logging** tab.

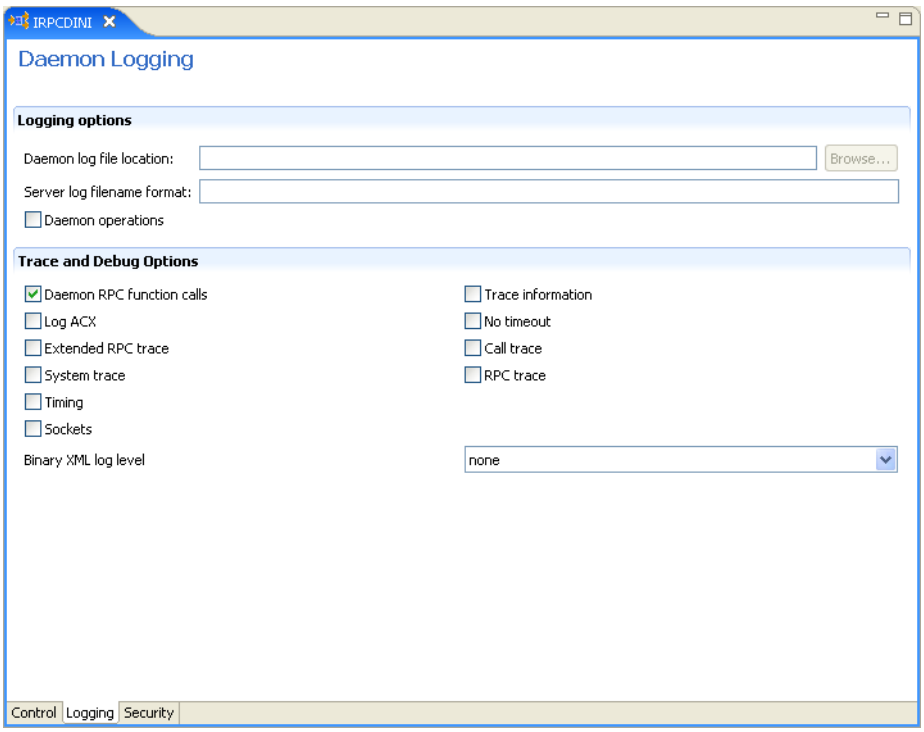
1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click a computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit Daemon Configuration**.
5. Click the **Logging** tab.
6. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

> **Note:** You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
>
> Changes made to the daemon configuration are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager perspective.

7. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Logging** tab for the daemon is shown in the following figure:

Figure 6–4 The Logging tab



The **Daemon Logging** tab comprises fields, as listed in the following table:

Table 6–11 Daemon Logging tab Components

Field	Description
Logging options	
Daemon log file location	Specifies the daemon produces its log data. The full path must be specified.
Server log filename format	Defines the name and location of the server log file. The field must specify the full path name. If no directory information is provided for the log file, then it will be located in the login directory of the account running Oracle Connect workstation.
Daemon operations	Select this to log all of the daemon operations.

Table 6–11 (Cont.) Daemon Logging tab Components

Field	Description
Tracing and debug options	
Daemon RPC function calls	Select this to log all daemon RPC function calls.
Log ACX	Select this if you want to log requests and processes.
Extended RPC trace	Generates a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server.
System trace	Generates system-specific tracing of various operations.
Timing	Generates a timestamp for every entry to the server log file.
Sockets	Generates a message in the server log file for each socket operation.
Trace information	Select this to log low-level RPC operations.
No timeout	Disables the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging.
Call trace	Generates a message in the server log file for each RPC function called. This is useful for troubleshooting the server.
RPC trace	Enables debugging messages on the server.
Binary XML log level	Sets the binary XML log level. Your options are: ■ debug ■ none (the default) ■ api ■ info

The following tokens can appear in the log file template and will be replaced accordingly:

- `%A`: workspace name
- `%D`: date (yymmdd)
- `%I`: instance number of the given workspace server
- `%L`: server account login directory
- `%P`: server process ID
- `%T`: time (hhmmss)
- `%U`: server account name (username)

For example, `%L/server_%A%I.log` may produce a log file such as: `/usr/smith/server_sales15.log`.

The default log file template is `%L/server_%A%I.log`.

Security

The following **Security** tab for the daemon is used to:

- Grant administration rights for the daemon.

- Determine access to the computer.

The following shows how to open the Daemon **Security** tab:

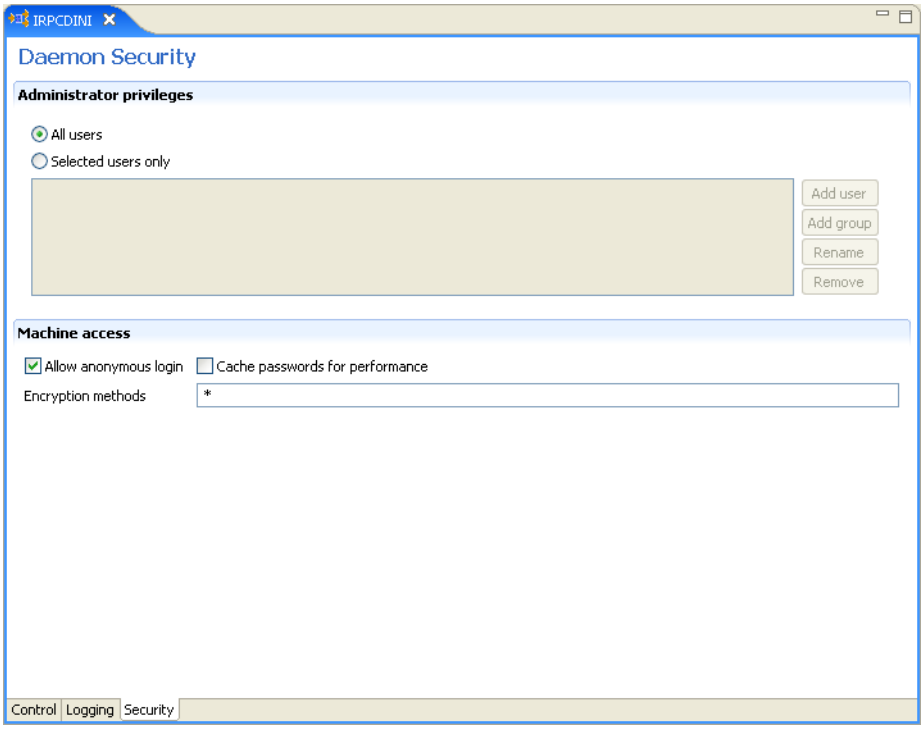
1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Right-click the daemon in the Runtime Explorer and select **Edit Daemon Configuration**.
5. Click the **Security** tab.
6. After making changes to the daemon, right-click the daemon and select **Reload Configuration**.

> **Note:** You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
>
> Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **Recycle servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The Daemon **Security** tab is shown in the following figure:

Figure 6–5 The Daemon Security tab



The **Daemon Security** tab comprises fields, as listed in the following table:

Table 6–12 Daemon Security tab Components

Field	Description
Administrators privileges	Identifies the users (accounts) allowed to perform administrative tasks (tasks that require administrative login).
All users	Enables all users to access the daemon and change the settings.
Selected users only	Identifies the names of users (accounts) and groups that can be administrators.[1] If a user is not specified, the account from which the daemon was started is considered the administrator. Note that the daemon does not require the user to log in to the account on the system, but to log in to the daemon using the account name and password.
Machine access	Manages access to the computer.
Allow anonymous login	Whether workspaces allow anonymous logins (without user name/password entries). For the optimal level of security, keep this option unchecked and define a username for the Daemon Administrators parameter. If unchecked, then no workspace can have an anonymous client. If checked, then a particular workspace allows anonymous clients.
Cached password	Enables login passwords to be cached. This enhances performance by reducing login times for future connections from the same client in a session.
Encryption methods	Specifies the encryption method used to send information across the network. The default is an asterisk (*), meaning that all methods are acceptable. If an encryption method is specified, it must be used. The RC4 and DES3 protocols are currently supported.

[1] The name is prefixed with '@', to utilize the operating system GROUP feature.

Workspaces

A daemon can include a number of workspaces. A workspace defines the server processes and environment that are used for the communication between the client and the server for the duration of the client request. Each workspace has its own definition. The workspace definition is divided into the following groups:

- General
- Server Mode
- Security

General

Using the **General** tab, you enter general information about the features that control the operation of the workspace, such as the server type, the command procedure used to start the workspace and the binding configuration associated with this workspace.

Do the following to open the **General** tab:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. From the Design perspective Configuration view, expand the **Machines** folder.

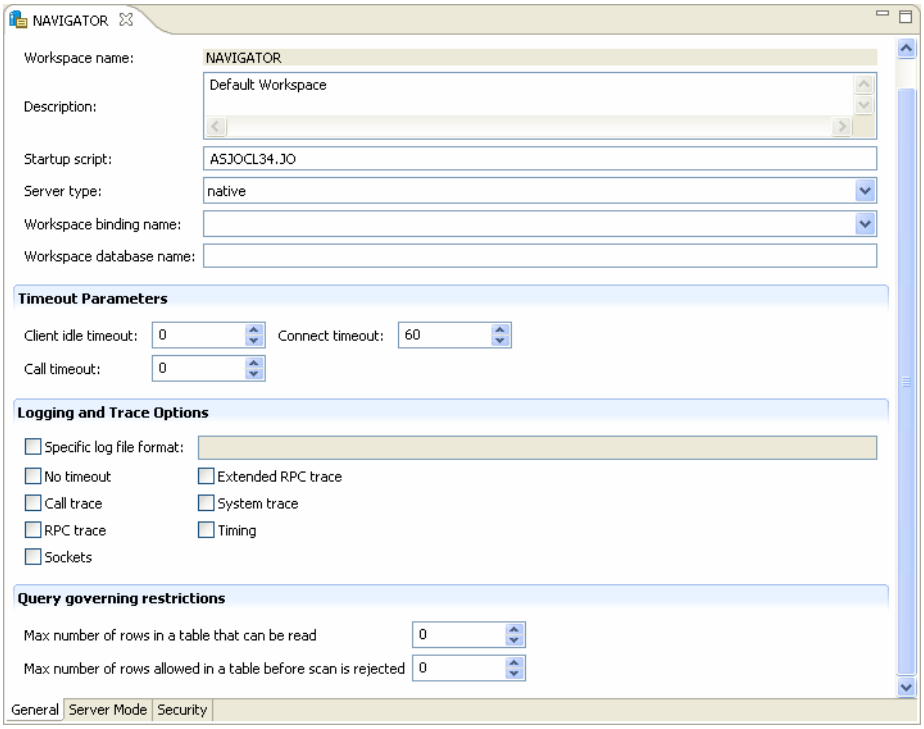
3. Right-click a computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**. The **General** tab opens.
6. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

> **Note:** You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
>
> Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

7. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **General** tab is shown in the following figure:

Figure 6–6 The General tab



The **General** tab comprises fields, as listed in the following table:

Table 6–13 General Tab Components

Field	Description
Info	

Table 6–13 (Cont.) General Tab Components

Field	Description
Workspace name	The name used to identify the workspace. **Note**: The default configuration includes the default Navigator workspace. This workspace is automatically used if a workspace is not specified as part of the connection settings.
Description	A description of the workspace.
Startup script	The full path name of the script that starts the workspace server processes. The script specified here must always activate the `nav_login` procedure and then run the server program (svc). If you do not specify the directory, the startup procedure is taken from the directory where the daemon resides. Oracle Connect includes a default startup script, which it is recommended to use.
Server type	This field is not applicable for use with OracleAS Adapters for SQL Server.
Workspace binding name	This field is not applicable for use with OracleAS Adapters for SQL Server.
Timeout parameters	The time the client waits for the workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond. Specifying the timeout here overrides the default setting, specified in the Control section. See Also: Control for details about the Daemon Control section.
Client idle timeout	The maximum amount of time a workspace client can be idle before the connection with the server is closed.
Connect timeout	The time the client waits for a workspace server to start. If the workspace server does not start within this period, then the client is notified that the server did not respond.
Call timeout	The timeout period for *short* calls for all daemons. The definition of a short call is a call that should be completed in a few seconds. For example, most calls to a database such as `DESCRIBE` should be completed in a few seconds as opposed to call like a `GETROWS` call, which can take a long time. In heavily loaded or otherwise slow systems, even short calls such as calls to open a file, may take a significant amount of time. If a short call takes more than the specified time to complete, then the connection is stopped. The default value for this parameter is 60 seconds. Values of less than 60 seconds are considered to be 60 seconds. Specifying the timeout in a workspace overrides the value set in this field for that workspace.
Logging and Trace Options	

Table 6–13 (Cont.) General Tab Components

Field	Description
Specific log file format	Defines the name and location of the server log file if you want the data written to a file instead of SYSOUT for the server process. The parameter must specify the name and the high level qualifier. The following tokens can appear in the log file template and will be replaced accordingly: ■ `%A`: workspace name ■ `%D`: date (yymmdd) ■ `%I`: instance number of the given workspace server ■ `%L`: server account's login directory ■ `%P`: server's process ID ■ `%T`: time (hhmmss) ■ `%U`: server's account name (username)
Trace options	Specifies the type of tracing to be performed. The following tracing options are available: ■ **No timeout**: Select this to disable the standard RPC timeouts, setting them to a long duration (approximately an hour) to facilitate debugging. ■ **Call trace**: Selecct this to generate a message in the server log file for each RPC function called. This is useful for troubleshooting the server. ■ **RPC trace**: Select this to enable debugging messages on the server. ■ **Sockets**: Select this to generate a message in the server log file for each socket operation. This is useful for troubleshooting client/server communication - providing a detailed trace of every client/server communication. ■ **Extended RPC trace**: Select this to generate a verbose message in the server log file for each low-level RPC function called. This is useful for troubleshooting the server. ■ **System trace**: Select this to generate operating system-specific tracing. ■ **Timing**: Select this to generate a timestamp for every entry to the server log file.
Query governing restrictions	
Max number of rows in a table that can be read	Select the maximum number of table rows that are read in a query. When the number of rows read from a table exceeds the number stated the query returns an error.
Max number of rows allowed in a table before scan is rejected	Select the maximum number of table rows that can be scanned. This parameter has different behavior for query optimization and execution. ■ For query optimization, the value set is compared to the table cardinality. If the cardinality is greater than the value, the scan strategy is ignored as a possible strategy (unless it is the only available strategy). ■ For query execution, a scan is limited to the value set. When the number of rows scanned exceeds the number entered, the query returns an error.

Server Mode

The Server Mode tab lets you configure the features that control the operation of the servers started up by the workspace and allocated to clients.

For example, you can configure the workspace to start up a number of servers for future use, prior to any client request, instead of starting each server when a request is received from a client.

Do the following to open the **Server Mode** tab:

1. From the **Start** menu, select **Programs, Oracle,** and then select **Studio**.
2. From the Design Perspective Configuration view, expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **Server Mode** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

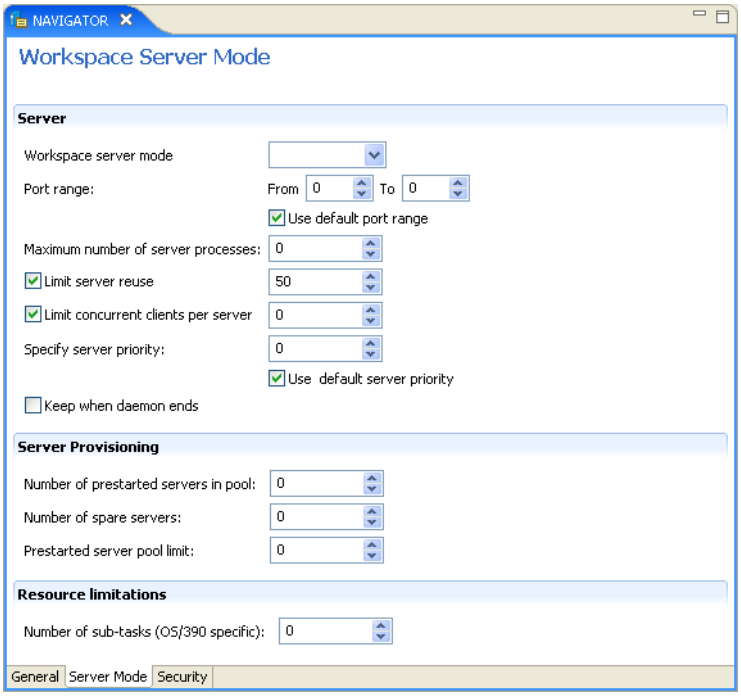
Notes:

- You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
- Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

8. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Server Mode** tab is shown in the following figure:

Figure 6–7 The Server Mode tab



The **Server Mode** tab has the following fields:

Table 6–14 Server Mode Tab Components

Field	Description
Workspace server mode	Specifies the type of new server processes that the daemon starts up. The daemon supports the following server modes: ▪ `singleClient`: Each client receives a dedicated server process. The account in which a server process runs is determined either by the client login information or by the specific server workspace. This mode enables servers to run under a particular user account and isolates clients from each other, as each receives its own process. However, this server mode incurs a high overhead due to process startup times and can use a lot of server resources as it requires as many server processes as concurrent clients. ▪ `multiClient`: Clients share a server process and are processed serially. This mode has low overhead because the server processes are already initialized. However, because clients share the same process, they can impact one another, especially if they issue lengthy queries. The number of clients that share a process is determined by the Clients per server limit field. ▪ `multiThreaded`: This mode is not applicable for use with OracleAS Adapter for SQL Server. ▪ `reusable`: An extension of single-client mode. Once the client processing finishes, the server process does not die and can be used by another client, reducing startup times and application startup overhead. This mode does not have the high overhead of single-client mode because the servers are already initialized. However, this server mode can use a lot of server resources as it requires as many server processes as concurrent clients. **Note**: The other modes can be set so that the server processes are reusable. The number of times a process can be reused is controlled by the Reuse limit field value.
Reuse limit	Sets the maximum number of times a particular server can be reused. A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization. The default for this field is none (0), indicating that server reuse is unlimited. This parameter is disabled only if the server mode value is `singleClient`.
Clients per server limit	Sets the maximum number of clients a server process for the current workspace accepts. The default for this field is none (0), indicating that the number of clients for each server is unlimited. This field is enabled only if the server mode value is `multiClient` or `multiThreaded`.

Table 6–14 (Cont.) Server Mode Tab Components

Field	Description
Server availability	Specifies the number of servers in a pool of servers, available to be assigned to a client. The following options are available: ■ Initial number of servers: The number of server processes that are prestarted for this workspace when the daemon starts up. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0. ■ Minimum number: The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0. ■ Keep when daemon ends: When a daemon is shutdown, all the servers started by that daemon are also killed, even if they are active. Set this field to `true` if you want the servers for the workspace to remain active, even after the daemon has been shut down. If this field is set to `true`, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level. ■ Set maximum number of servers: The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if a number of server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.
Port range	the workspace. Determines the range of ports available for this workspace when starting server processes. Use this option when you want to control the port number, so that Oracle Connect can be accessed through a firewall. Enter the port range in the following fields: ■ **From**: enter the lowest numbered port in the range. ■ **To**: Enter the highest numbered port in the range.
Use default port range	Select this to use the port range that is defined in the daemon. This is defined in the Port range for servers field in the daemon Control tab.
Maximum number of server processes	Enter the maximum number of server processes that can run at the same time.
Limit server reuse	Select this if you want to limit the number of servers that can be reused. If this is selected, the Reuse limit parameter is available.

Table 6–14 (Cont.) Server Mode Tab Components

Field	Description
	If **Limit server reuse** is selected, in the field next to the check box, enter the maximum number of times a server can be reused. Select the maximum of clients accepted in a server process. A one-client server can be reused after its (single) client has disconnected. Reuse of servers enhances startup performance because it avoids the need to repeat initialization. This parameter is not available if the Limit server reuse parameter is not selected. This parameter is not available if the server mode value is **singleClient**.
Limit concurrent clients per server	Select this to limit the number of clients that a server can accept for the current workspace process. If this is not selected, the number of clients is unlimited. If Limit concurrent clients per server is selected, in the field next to the check box, enter the maximum number of clients that a server process for the current workspace accepts. The default for this field is **None**, indicating that the number of clients for each server is unlimited. This field is available if the server mode value is **multiClient** or **multiThreaded**.
Specify Server Priority	Set the priority for servers. For example, a workspace for applications with online transaction processing can be assigned a higher priority than a workspace that requires only query processing. The lower the number, the higher the priority. For example, workspaces with a priority of `1` are given a higher priority than workspaces with a priority of `2`. **Note**: This is unavailable if **Use default server priority** is selected.
Use default server priority	Sets the priority to `0`. There is no specific priority for this workspace. Clear this check box to set a priority in the **Specify server priority** parameter.
Keep when daemon ends	Select this to kill all servers started by that daemon when a daemon is shutdown, even if they are active. Select this if you want the servers for the workspace to remain active, even after the daemon has been shut down. If selected, it is the responsibility of the system operator or manager to ensure that the servers are eventually killed. This must be done at the system level.
Server Provisioning	
Number of prestarted servers in pool	**Initial number of servers**: The number of server processes that are prestarted for this workspace when the daemon starts up. When the number of available server processes drops lower than the value specified in the Minimum number field, the daemon again starts server processes until this number of available server processes is reached. The default for this field is 0.

Table 6–14 (Cont.) Server Mode Tab Components

Field	Description
Number of spare servers	The minimum number of server processes in the prestarted pool before the daemon resumes creating new server processes (to the value specified in the Initial number of servers field). If this field is set to a value higher than the Initial number of servers field, the daemon uses the value specified in the Initial number of servers field. The default for this field is 0.
Prestarted server pool limit	The maximum number of available server processes. Once this number is reached, no new nonactive server processes are created for the particular workspace. For example, if a number of server processes are released at the same time, so that there are more available server processes than specified by this field, the additional server processes higher than this value are terminated. The default for this field is zero, meaning that there is no maximum.
Resource limitations	
Nuber of sub-tasks	The number of sub-tasks for a server that are prestarted for this workspace when the daemon starts up. In addition to setting up a pool of server processes as described earlier, you can set additional server processes as sub-tasks by specifying this parameter. If you set 10 servers and 10 prestarted sub-tasks then 100 tasks are started (10 sub-tasks for each process).

Security

The **Security** tab lets you configure the seciruty level for a workspace. this lets you set the security options for the workspace only. The Security tab is used to:

- Grant administration rights for the workspace
- Determine access to the workspace by a client

Do the following to open the **Security** tab:

1. From the **Start** menu, select **Programs, Oracle,** and then select **Studio**.
2. From the Design Perspective Configuration view, expand the **Machines** folder.
3. Right-click the computer and select **Open Runtime Perspective**.
4. Expand the Daemons node to display the workspaces in the Runtime Explorer.
5. Right-click the workspace and select **Edit Workspace Configuration**.
6. Click the **Security** tab.
7. After making changes to the workspace, right-click the daemon and select **Reload Configuration**.

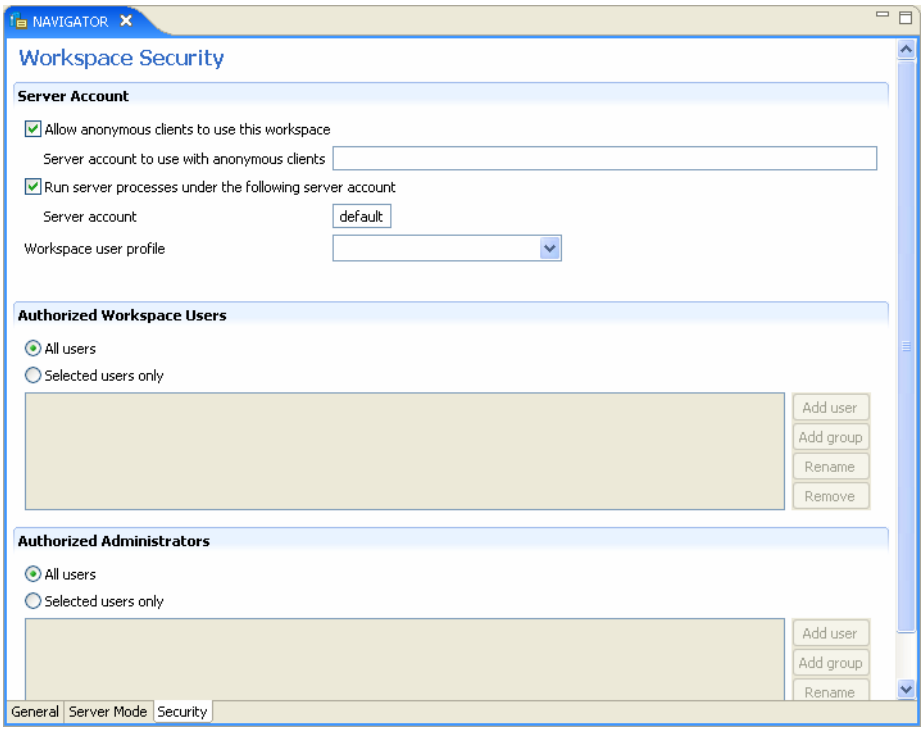
Notes:

- You can also change daemon settings using the Configuration Explorer, by selecting a computer and scrolling the list to the required daemon. Right-click the daemon and select **Edit Daemon**.
- Changes made to the daemon configuration are not implemented. They are only implemented after the configuration is reloaded using the **Reload Configuration** option in the Runtime Manager.

8. Right-click the daemon and select **Recycle Servers**. Any servers in the connection pool are closed and new servers start with the new configuration.

The **Security** tab is shown in the following figure:

Figure 6–8 The Security tab



The **Security** tab has the following fields:

Table 6–15 Security Tab

Field	Description
Server Account	This section defines the users (accounts) allowed to access the workspace, firewall access ports, workspace account, and anonymous login permissions.
Use specific workspace account	Select this if you want to define the operating system account used for the workspace. If selected, enter the name of the workspace account in the **workspace account** field. If not selected, the account name that was provided by the client is used.
Allow anonymous clients to use this workspace	Select this if you want to allow this workspace to be invoked without authentication. If selected, enter the name of the workspace account in the **Server account to use with anonymous clients** field.
Authorized Workspace users	Indicate which users have permission to use the workspace. Select one of the following ■ **All user**s: Any user who has logged on to the daemon may use the workspace ■ **Selected users only**: Select this to allow only users (or accounts) with specific permission to use the workspace. When this is selected, add the names of users (or accounts) and groups that can be use the workspace in the field below. **Note**: If no user is specified, any user who has logged on to the daemon may use the workspace.
Authorized Administrators	Identifies the users (accounts) with administrator privileges. Select one of the following: ■ **All users**: Indicates that anyone can access the workspace and change the settings. ■ **Selected users only**: Select this to allow only users (or accounts) with specific permission to be administrators. When this is selected, add the names of users (or accounts) and groups that can be workspace administrators. If no user is specified, any user who has logged on to the daemon may administrator this workspace.

C

Editing Properties for the OracleAS CDC Adapter for SQL Server

The OracleAS CDC Adapter for SQL Server has a number of configuration properties. You can edit the properties in Oracle Studio after Setting Up a Change Data Capture with the OracleAS CDC Adapter for SQL Server. This chapter describes the configuration properties and how to edit them. It contains the following sections:

- Editing Properties in Oracle Studio
- Configuration Properties
- Access to Change Events
- Transaction Support
- Security

Editing Properties in Oracle Studio

After you create the OracleAS CDC solution, you can also edit the properties. The solution create two adapters, the adapter and the CDC Queue adapter. The adapter is created on the server machine and the CDC Queue adapter is created on the staging area machine. For more information, see Setting up a Change Data Capture in Oracle Studio.

Editing the OracleAS CDC Adapter Properties

To edit the CDC adapter properties, open the Oracle Studio Design perspective and find the binding for the CDC solution on the server machine. Then open the adapter, which contains the name of the CDC solution with the suffix `_ag` added to it. Changes to adapter properties are reset when the CDC solution is redeployed, therefore these changes must be reapplied following solution deployment. For information on deploying a solution, see Deploying a Change Data Capture.

Do the following to edit the CDC adapter properties.

1. From the **Start** menu, select, **Programs**, **Oracle**, and then select **Studio**.
2. Expand the **Machines** folder.
3. Expand the server machine that you created when Setting up a Change Data Capture in Oracle Studio.
4. Expand the **Bindings** folder, and then expand the binding the name of the CDC solution with the suffix `_ag` added to it.
5. Expand the **Adapter** folder.

6. Right-click the adapter the adapter with the name of the solution and the suffix `_ag` and select **Open**.

   The adapter configuration editor opens in the editor, which displays the properties for the adapter.

7. Edit the adapter parameters as required.

   See OracleAS CDC Adapter Configuration Properties for a description of the properties.

Configuration Properties

This section describes the configuration properties for the SQL Servre CDC adapter. You can edit the properties using Oracle Studio. The SQL Server CDC has the following types of properties:

- CDC Logger Properties
- OracleAS CDC Adapter Configuration Properties
- Change Router Properties
- Referential Integrity Considerations

CDC Logger Properties

`Logger Name`: the name of the logstream used for the data capture.

OracleAS CDC Adapter Configuration Properties

This section describes the common configuration properties for Oracle CDC adapters and the change router configuration properties, which is located on the staging area machine.

To edit the CDC adapter properties, open the Oracle Studio Design perspective and find the binding for the CDC solution you created. The binding contains the name of the CDC solution with the suffix `_ag` added to it. Open the adapter with the name of the solution and the suffix `_ag` to edit the properties. For information on how to edit adapter properties in Oracle Studio, see Editing Properties in Oracle Studio. Changes to adapter properties are reset when the CDC solution is redeployed, therefore these changes must be reapplied following solution deployment.

See the following topics for a description of the configuration properties:

- OracleAS CDC Adapter for SQL Server Properties
- CDC Queue Adapter Properties
- Common CDC Adapter Properties

OracleAS CDC Adapter for SQL Server Properties

The following describes the properties for the OracleAS CDC Adapter for SQL Server.

There are also additional properties that are common to all Oracle CDC adapters. For a description of the common Oracle CDC adapter properties, see Common CDC Adapter Properties.

- `transientStorageDirectory`: Enter the full path to the folder with the Transient Storage file. This property is also defined in the CDC solution wizard when Set up the CDC Service.

CDC Queue Adapter Properties

The CDC Queue adapter is a data base adapter that is found in the staging area. It is created automatically when Setting up a Change Data Capture in Oracle Studio. It has three properties that can be viewed in the adapter's editor in Oracle Studio:

- `connectString`
- `defaultDatasource`
- `multipleResults`

To ensure that the queue adapter works properly with your CDC solution, do not change the values for these properties.

The CDC Queue adapter may have additional properties that can be viewed in the adapter's XML schema. For infomation on how to view the XML, see Adapter Metadata XML.

The following property, which is viewed in the XML only can be edited:

- `maxRecords`: The maximum number of records that can be returned.

For information on how to edit XML records in Oracle Studio, see Editing XML Files in Oracle Studio.

Common CDC Adapter Properties

The following table describes the common configuration properties for Oracle CDC adapters. For information on specific CDC adapter properties for the OracleAS CDC Adapter for SQL Server, see OracleAS CDC Adapter for SQL Server Properties.

Table 6–16 CDC Adapter Configuration Properties

Parameter	Type	Default	Description
`datasource`	string		The name of the data source for the OracleAS CDC adapter.
`routers`			A list of users who can connect to the adapter and get change events from it for processing. If no routers are specified, any valid user for the workspace can get change events from the Oracle Connect adapter. To add the list of users in Oracle Studio, expand the `router` property then right-click `users`. A new entry called Item(#) is added to the Property column. In the Value column, enter the User Name for this router.
`retryInterval`	int	2	The polling interval for the database journal When no events are received in the database journal, the adapter waits for the amount of time (in seconds) that is indicated for this property.
`getTransactionInfo`	boolean	`true`	When set to `true`, transaction information (`begin`, `commit`, `rollback`) is returned.
`getBeforeImage`	boolean	`false`	When `true`, before image information is returned.

Table 6–16 (Cont.) CDC Adapter Configuration Properties

Parameter	Type	Default	Description
`realTime`	boolean	`true`	When `true`, this reduces latency in getting change events, however it also increases the polling of the database journal. The change router asks for N events from the adapter. If the adapter finds fewer than N events in the database journal and `realTime` is `true`, these events are immediately returned to the change router. If `realTime` is false, the adapter polls the journal again after waiting for the number of seconds indicated in the `retryInterval`.
`logLevel`			The logging level. The following are the available log levels: ▪ `none` ▪ `api` ▪ `internalCalls` ▪ `info` ▪ `debug`
`checkTimeoutEveryNEvents`	int	`100`	The number of events that occur before the CDC adapter checks the `timeout` value. For example, if this property is set to `100`, the adapter will check the `timeout` value after 100 events have taken place. If the amount of time set in the `timeout` property is past, the adapter will timeout.

Change Router Properties

The following table describes the SQL-based change event router configuration parameters.

To edit the router properties, open the Oracle Studio Design perspective and find the binding for the CDC solution you created. The binding contains the name of the CDC solution with the suffix `_router` added to it. Open the adapter with the name of the solution and the suffix `_router` to edit the properties. For information on how to edit adapter properties in Oracle Studio, see Editing Properties in Oracle Studio.

Table 6–17 Change Router Configuration Parameters

Parameter	Type	Default	Description
`cdcDatasource`	string		The OracleAS Change Data Source.

Table 6–17 (Cont.) Change Router Configuration Parameters

Parameter	Type	Default	Description
`eliminateUncommittedChanges`	Boolean	false	When set to `true`, only committed change records are moved to the Change Table. If `false`, all change records are moved to the change tables (in which case, memory usage is minimal) hence the change table may contain rolled back data. For most adapters, following the RI considerations (see Referential Integrity Considerations) results in rolled-back changes eliminated naturally by means of compensating change records generated by the adapter in case of a rollback. Consult the respective CDC adapter documentation for details.
`eventExpirationHours`	int	48	Indicates how long change records are kept in change tables within the staging area. After the indicated time, change records are deleted. You can set a value between `0` and `50000`. A value of `0` means that the records are never deleted. A value of 1 indicates that the records are kept for one hour.
`logLevel`	enum		The logging level for the router. The following are the available log levels: ▪ `none` ▪ `api` ▪ `internalCalls` ▪ `info` ▪ `debug`.
`maxDeletedEventsInBatch`	int	`500`	Controls how many expired change records are delete in a single pass. This number may need to be lowered in some rare cases in order to reduce latency when a large number of change events is continuously being received.
`maxOpenfiles`	int	`200`	Controls the number of physical files opened by the router.
`maxTransactionMemory`	int (in Kb)	`1000`	Specifies how much memory can be stored in memory per transaction before it is off-loaded to disk. This number should be higher than the average transaction size so that the slower-than-memory disk is not used too often.
`maxStagingMemory`	int (in Kb)	`100000`	Specifies how much memory in total can be used for storing active transactions (ones that have not yet committed or rolled back).

Table 6–17 (Cont.) Change Router Configuration Parameters

Parameter	Type	Default	Description
`sourceEventQueue`	Structure:		Connection information to the OracleAS CDC adapter.
▪ `server`	string		
▪ `workspace`	string		
▪ `adapter`	string		
▪ `eventWait`	int	`30`	
▪ `maxEventsAsBlocks`	int	`250`	
▪ `reconnectWait`	int	`15`	
▪ `fixedNat`	boolean	`false`	
`stagingDirectory`	string		Specifies the directory where the staging area change files will be stored. This directory also stores off-loaded transactions as well as timed-out transactions and error files.
`transactionTimeout`	int (in seconds)	`3600`	Specifies how long can a transaction be active without getting new events. This parameter should be set according to the corresponding setting of the captured database. In particular, this setting must not be lower than the database's transaction time-out setting as this may lead to the loss of transactions.
`useType`	enum	sqlBbased Cdc	This parameter must be set to this value.
`routers`			A list of users who can connect to the change event router and get change events from it for processing. If no routers are specified, any valid Oracle Connect user for the workspace may get change events from the adapter. To add the list of users in Oracle Studio, expand the `router` property then right-click `users`. A new entry called Item(#) is added to the Property column. In the Value column, enter the User Name for this router.
`senders`			A list of users who can connect to the change event router and send change events to it for processing. If no routers are specified, any valid Oracle Connect user for the workspace may get change events from the adapter. To add the list of users in Oracle Studio, expand the `sender` property then right-click `users`. A new entry called Item(#) is added to the Property column. In the Value column, enter the User Name for this router.

Table 6–17 (Cont.) Change Router Configuration Parameters

Parameter	Type	Default	Description
subscribeAgentLog	Boolean	False	When set to true, the change router writes the contents of the OracleAS CDC adapter's log into its own log. Do not set this property to true if the logLevel property is set to debug because the large amount of information that is sent in this mode will cause performance delays.
nodeID	string		ID for a node when using multi-router mode. Each node represents a router.
alternativeOwnerSeparator	string	..	This is the value of the separator that is used to separate the name of the owner and the suffix table. In Oracle Studio, the default separator for the staging area is an underscore (_). For example, owner.table. When using SSIS to configure a CDC solution, an underscore will cause an error. When you change the default separator in the Oracle Studio Preferences window, the new value is entered in this property.

Referential Integrity Considerations

Some related tables have referential integrety (RI) constraints enforced on them. For example, with OrderHeader and OrderLines one cannot have OrderLines without an associated OrderHeader.

When processing change events by the table (which is how an OracleAS CDC works) as opposed to by transaction, referential integrity cannot be maintained properly. For example, when first handling all OrderHeader records and then all OrderLines records then a deleted OrderHeader may be applied long before the required delete of the associated OrderLines records.

In order to reduce the potential referential integrity to a known time frame after which referential integrity is restored, a somewhat different process is needed (compared with Reading the Change Tables').

A special SYNC_POINTS table should be added to maintain a common sync-point for use with multiple related tables. The table is defined as follows:

Table 6–18 SYNC_POINTS Table Structure

Column Name	Data Type	Description
application_name	string (64)	The application for which the processing is done.
table_name	string (64)	The name of the synchronization point
context	String (32)	A stream position that can be safely used as an upper bound for event retrieval of all related tables

This table's primary unique key is the concatenation of application_name + sync_name. The use of this table is not mandatory but it is part of the recommended use pattern of SQL-based CDC.

The SYNC_POINTS table is created with the following definition (where filename is changed into an actual path):

```
<?xml version='1.0' encoding='UTF-8'?>
<navobj>
  <table name='SYNC_POINTS'
         fileName='<staging-directory-path>SYNC_POINTS' organization='index'>
    <fields>
      <field name='application_name' datatype='string' size='64'/>
      <field name='sync_name' datatype='string' size='64'/>
      <field name='context' datatype='string' size='32'/>
    </fields>
    <keys>
      <key name='Key0' size='128' unique='true'>
        <segments>
          <segment name='application_name'/>
          <segment name='sync_name'/>
        </segments>
      </key>
    </keys>
  </table>
</navobj>
```

The following procedure describes how to ensure RI is regained at the end of a group of ETL rounds. It is an extension of the procedure described earlier for consuming change records. Here we assume that tables T1, T2 and T3 are related with RI constraints and that A is the application we are working under.

To create a stream position

1. This is a one-time setup step aimed to create a stream position record for T [1/2/3] + A in the STREAM_POSITIONS table. The following SQL statement creates that record:

   ```
   insert into STREAM_POSITIONS values (‘A’, ‘T1’, ‘’);
   insert into STREAM_POSITIONS values (‘A’, ‘T2’, ‘’);
   insert into STREAM_POSITIONS values (‘A’, ‘T3’, ‘’);
   ```

2. This step is performed at the beginning of a group of ETL rounds processing (that is before starting to process change events for T1, T2 and T3). The goal here is to get a shared sync point for retrieval of T1, T2 and T3. This is done by sampling the ‘context’ column of the SERVICE_CONTEXT table. This value is the stream position of the last change record in the most recently committed transaction. This is done as follows:

   ```
   insert into SYNC_POINTS
       select 'A' application_name, 'T123' sync_name, context from SERVICE_
   CONTEXT;
   ```

   Here, T123 is the name chosen for the synchronization [points of tables T1, T2, and T3.

3. This step is where change data is actually read. It occurs on each ETL round.

   ```
   select n.* from T t, STREAM_POSITIONS sp, SYNC_POINTS sy where
       sp.application_name = 'A' and
       sp.table_name = 'T' and
       sy.application_name = sp.application_name and
       sy.sync_name = 'T123' and
       n.context > sp.context and n.context <= sy.context order by n.context;
   ```

Note that "`n.context <= sy.context`" is used because the context represents a change record to be processed and processing should include the change record associated with `sy.context`, too.

This query retrieves change records starting from just after the last handled change record but stopping at a common sync point. "`n.*`" can be replaced with an explicit list of columns, however it is important that the 'context' column must be selected as this is the change record stream position which is required for the next step.

This step occurs at the end of each ETL round once all change records were retrieved and processed for a table Ti. Let's assume that the value of the 'context' column of the last change record was 'C'. This value needs to be stored back into the STREAM_POSITION table for the next ETL round. This is done with:

```
update STREAM_POSITIONS set context='C' where application_name 'A' and table_
name = 'Ti';
```

This value can be stored more frequently during the ETL process as needed. The general guideline is that once change record data has been committed to the target database, the stream position should be updated as well.

Access to Change Events

Changes are captured and maintained in a change table. The table contains the original table columns and CDC header columns. The header columns are described in the following table:

Table 6–19 Header Columns

Column Name	Description
context	The change record stream position in the staging area. The column is defined as primary unique index. It is a 32-bytes string with the following structure: `<yyyymmdd>T<hhmmss>.<nnnn><cccccc>` Where: ■ `<yyyymmdd>T<hhmmss>` is the commit processing timestamp as generated in the staging area when starting to process the `Commit` event. ■ `<nnnn>` is a unique number to differentiate between transactions committed during the same second (up to 99,999 are assumed). ■ `<cccccc>` is a counter for the change events in the transaction making every stream position unique (up to 9,999,999 are assumed).
agent_context	The original change record stream position from the agent (non-numeric). This column is defined as alternate, descending unique index. It is used: ■ To ensure that a change event does not appear more than once in the change table. ■ To allow scanning of a change table backwards, peeking easily for the last `N` change events. ■ When working with complex records, multiple records may result from a single back-end change record. This column enables the user to associate these records with the single change record.

Table 6–19 (Cont.) Header Columns

Column Name	Description
operation	This column lists the operations available for the CDC adapter. The available operations are: ■ INSERT ■ DELETE ■ UPDATE ■ BEFOREIMAGE ■ COMMIT ■ ROLLBACK
transactionID	The operation's transaction ID. The transaction ID is increased each time a new transaction starts and a BEGIN_AXACT record is logged.
tableName	The name of the table where the change was made. For INSERT, UPDATE, and BEFOREIMAGE operations, the owner name and then the table name are displayed. For COMMIT and ROLLBACK operations, this value is the same as the `OPERATION` value.
rowID	An identifier for the row.
timestamp	The date and time of the occurrence.

Transaction Support

The OracleAS CDC Adapter for SQL Server supports transactions.

Security

The following are some specific security requirements for this CDC agent:

- You must have an account with administrator rights to run the MS SQL Server and MS SQL Server CDC components.
- The CDC agent and the TLOG Miner (LGR) must be executed as members of the `sysadmin` server role.

Data Types

The following table shows the Oracle Connect data types and their SQL equivalent that are supported by the OracleAS CDC Adapter for SQL Server. In addition to the data types listed in this table, there is some limited support for User Defined Data Types (UDT).

Table 6–20 Supported Data Types

Char	SQLCHAR
Char	SQLCHAR
Datetime	SQLDATETIME
Decimal	SQLDECIMAL
Double	SQLFLT8
Float	SQLFLT8

Table 6–20 (Cont.) Supported Data Types

Char	SQLCHAR
Int	SQLINT4
Money	SQLMONEY
Nchar	SQLNCHAR
Nvarchar	SQLNVARCHAR
Numeric	SQLNUMERIC
Real	SQLFLT4
Smalldatetime	SQLDATETIM4
Smallint	SQLINT2
Smallmoney	SQLMONEY4
Tinyint	SQLINT1
Bigint	SQL_BIGINT
Binary	SQLBINARY
Varchar	SQLCHAR
Bit (Limited to one column per table)	SQLBIT
Ntext- Value processing is skipped. The filed is exposed as NULL	
Text - Value processing is skipped. The filed is exposed as NUL.	
Uniqueidentifier	SQL_UNIQUEIDENTIFIER

Note: The OracleAS CDC Adapter for SQL Server supports only the data types listed in the Supported Data Types table.

User Defined Data Types (UDT)

The OracleAS CDC Adapter for SQL Server does not provide complete support for UDT.

To handle a UDT, Oracle Connect maps it onto its base type and handles it as if it were the base data type.

D

Globalization Settings

The OracleAS CDC Adapter for SQL Server provides the globalization support for the following languages:

- Arabic
- English (the default)
- French
- German
- Greek
- Hebrew
- Italian
- Japanese
- Korean
- Portuguses
- Simple Chinese
- Spanish
- Traditional Chinese
- Turkish

This appendix describes how to define the language support.

Defining the Language and Codepage

The language and codepage parameters are accessed from the computer where Oracle Studio is installed.

Perform the following steps to define the required language and codepage:

1. From the **Start** menu, select **Programs, Oracle**, and then select **Studio**.
2. In the Design perspective **Configuration** view, expand the **Machines** folder.
3. Expand the machine for which you want to set the language.
4. Expand the Bindings folder and right-click the **NAV** binding.
5. Select **Open**.
6. Expand the **Language Settings** and do the following:

- From the **Language** list, select the NLS supported language to use in this binding. Valid values are are listed in the Language Name column of NLS Language Codes table.
- From the **Codepage** list, select the codepage that you want to use with this language. The code pages available are determined by the Language that is selected. If you have additional code pages available, you can manually enter them in this field.

  Note: If you change the language, the code page will also change. Check to be sure that you want to use the selected code page with the language you selected.

  If no codepage is selected, the default codepage for the selected language is used.
- From the **NLS string** list, select the NLS string for this language and code page. The NLS strings available are determined by the code page that is selected. If you have additional NLS strings available, you can manually enter them in this field.

  The codepage is used by a field with a data type defined as `nlsString`. This parameter is used for a field with a codepage that is different than the machine's codepage. This property includes values for the name of the codepage and whether the character set reads from right to left (as in middle-eastern character sets).

  For example, the following specifies a Japanese EUC 16-bit codepage:

  ```
  <misc nlsString="JA16EUC,false"/>
  ```

7. Save the change. New servers will use the language selected.

The following table lists the codepages:

Table 6–21 NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
English US	ENUS	Windows-1252	ISO-8859-15	IBM1140	IBM285, IBM500, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
English UK	ENUK	Windows-1252	ISO-8859-15	IBM1146	IBM285, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)

Table 6–21 (Cont.) NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
French	FRE	Windows-1252	ISO-8859-15	IBM1147	IBM297, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Latin International	LAT	Windows-1252	ISO-8859-15	IBM1148	IBM500, IBM037, IBM1140, IBM1047, ISO-8859-1 (ASCII based)
Spanish	SPA	Windows-1252	ISO-8859-15	IBM1145	IBM284, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
German	GER	Windows-1252	ISO-8859-15	IBM1141	IBM273, IBM037 ,IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Portuguese	POR	Windows-1252	ISO-8859-15	IBM1140	IBM037, IBM500, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Italian	ITL	Windows-1252	ISO-8859-15	IBM1144	IBM280, IBM037, IBM500, IBM1140, IBM1148, IBM1047, ISO-8859-1 (ASCII based)
Greek	GRK	Windows-1253	ISO-8859-7	IBM875	-
Russian[1]	RUS	Windows-1251	ISO-8859-5	IBM1154	IBM1025

Table 6–21 (Cont.) NLS Language Codes

Language Name	Language Code	Windows Default	ASCII Platforms (Default)	EBCDIC Platforms (Default)	Alternative Codepages (EBCDIC based unless noted otherwise)
Turkish[2]	TUR	Windows-1254	ISO-8859-9	IBM1155	IBM1026
Hebrew	HEB	Windows-1255	ISO-8859-8	IBM424	IBM 862
Arabic	ARA	Windows-1256	ISO-8859-6	IBM420	
Japanese	JPN	SJIS	SJIS	IBM939	EUC-JP=EUC, VMS-JP
Chinese - Simplified	SCHI	GBK	GBK	IBM935	-
Chinese - Traditional	TCHI	BIG5	BIG5	IBM937	-
Korean	KOR	MS949	EUC-KR	IBM933	MS949

[1] Russian users who use ANSI 1251 Cyrillic as their Windows codepage must edit the `RUS.TXT` file and compile it to RUS.CP using the NAV_UTIL CODEPAGE.

[2] To work with solutions in Oracle Studio, when using Turkish, add the `-nl en` switch to the Target path in the Oracle Studio shortcut properties. For example: `"C:\Program Files\Oracle\Studio1\studio.exe -nl en"`

E

Editing XML Files in Oracle Studio

In many cases you must manually edit the metadata to configure parts of a solution or composition. Metadata is created in XML format. You define aspects of a solution by changing the values of the elements and attributes of the XML files that belong to the solution. Oracle Studio provides a graphical interface where you can define the various aspects of a solution. This interface lets you make changes easily without having to manually edit the XML file.

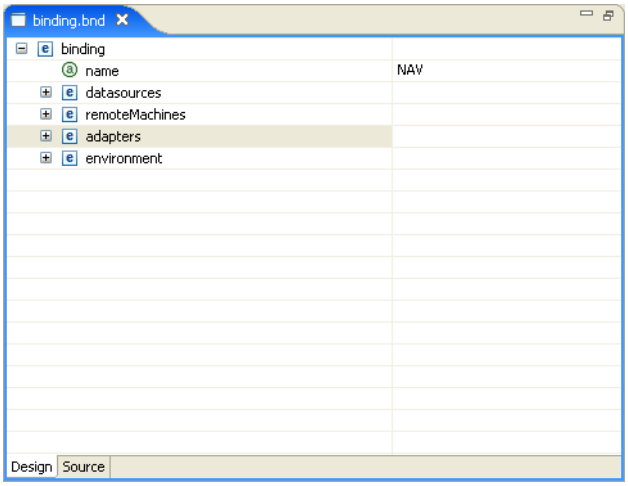
Preparing to Edit XML Files in Oracle Studio

You can edit XML files for the following items in Oracle Studio:

- Machines
- Bindings.
- Daemons
- Users

When you open an XML file, a graphical representation of the file is opened in the editor. The editor displays the elements and attributes in the file in the first column and their corresponding values in the second column. Each entry has an icon that indicates whether the entry is an element or an attribute. Click the **Source** tab to view the file in its native format. The following figure is an example of the editor's view of an XML file.

Figure 6–9 XML Graphical Display



Do the following to edit an XML file in Oracle Studio

1. In the Design perspective, open the Navigator view.
2. In the Navigator view, find the item with the XML file that you want to edit. This can be a machine, binding, daemon, or user.
3. Right-click the item and select **Open as XML**. A graphical list of the file's elements and attributes opens in the editor.
4. Find the element or attribute (property) that you want to change.
5. Click in the right column next to the property you are changing and edit or add the value.
6. Save the file, then select it again in the Project Explorer and press F5 to refresh.

   The XML file is updated automatically.

Making Changes to the XML File

You can also make the following changes to XML files in Oracle Studio:

- Remove Objects
- Add DTD Information
- Edit Namespaces
- Add Elements and Attributes
- Replace an Element

Remove Objects

You can delete an element, attribute, or other object from the XML file.

Do the following to remove an object

1. Right-click an object from the list in the editor.
2. Select **Remove**.

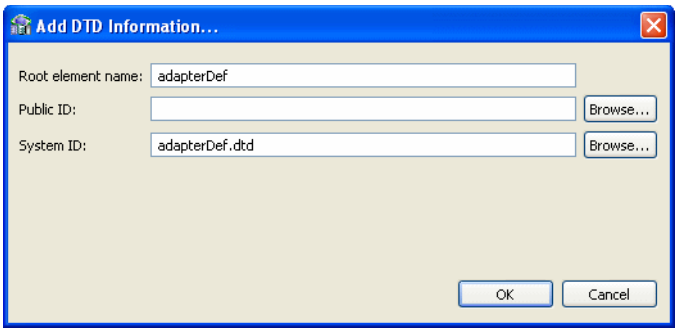
Add DTD Information

You can add DTD information to an element or attribute.

Do the following to add DTD Information

1. Right-click an element or attribute and select **Add DTD Information**. The Add DTD Information dialog box opens.

Figure 6–10 Add DTD Information Dialog Box



2. Enter the information requested in the dialog box. The following table describes the Add DTD Information dialog box.

Table 6–22 Add DTD Information

Field	Description
Root element name	The name of the XML root element.
Public ID	The value in this field is the Public Identifier. It is used to associate the XML file (using an XML catalog entry) with a DTD file by providing a hint to the XML processor. Click **Browse** to select an XML catalog entry from a list. An XML Catalog entry contains two parts, a Key (which represents a DTD or XML schema) and a URI (which contains information about a DTD or XML schema's location). Select the catalog entry you want to associate with your XML file.
System ID	The value in this field is the DTD the XML file is associated with. You can change the DTD the file is associated with by editing this field. The XML processor will try to use the Public ID to locate the DTD, and if this fails, it will use the System ID to find it. Click **Browse** to select a system ID. You can this in two ways: ■ Select the file from the workbench. In this case, update the with the import dialog box. ■ Select an XML catalog entry.

3. Save the file, then select it again in the Project Explorer and press F5 to refresh.

   The XML file is updated automatically.

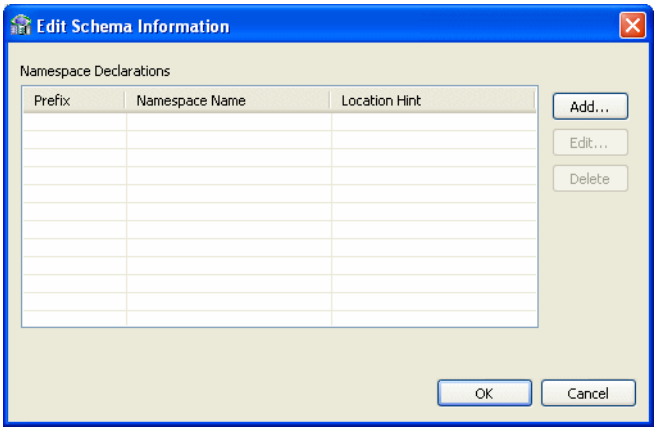
Edit Namespaces

You can make changes to the namespaces associated with an element or attribute.

Do the following to edit namespaces

1. Right-click an element or attribute and select **Edit namespaces**. The Edit Schema Information dialog box opens.

Figure 6–11 Edit Schema Information



2. Click on one of the buttons to make any changes to this information.

Do the following to add a new namespace

1. From the Schema Information dialog box, click **Add**.
2. The Add Namespace Definitions dialog box opens. Select one of the following:
   - **Select from registered namespaces**. This selection is available when the dialog box opens. Select from the list of registered namespaces and then click OK. If no registered namespaces are available, the list is empty.
   - **Specify new namespace**. Enter the information described in the following table:

Table 6–23 New Namespace

Field	Description
Prefix	The prefix is added to all qualified elements and attributes in the XML file.
Namespace Name	The namespace of the XML file.
Location Hint	The location of the XML schema of the XML file. An XML Catalog ID or a URI can be entered in this field. Click **Browse** to search for the schema you want You can this in two ways: ▪ Select the schema from the workbench. In this case, update the with the import dialog box. ▪ Select an XML catalog entry. The **Namespace Name** and **Prefix** fields are be filled with the appropriate values from the schema (you must leave the fields blank for this to occur). **Note**: If you are creating an XML file from an XML schema, you cannot change the Namespace Name or Location Hint values.

To edit a namespace

1. From the Schema Information dialog box, click **Edit**.
2. Enter the information in the fields.

Add Elements and Attributes

You can add additional elements and attributes to the XML file.

Do the following to add Elements and Attributes

1. Right-click an element.
2. Select one of the following:
   - **Add Attribute** to add an attribute under the selected element.
   - **Add Child** to add another element under the selected element
   - **Add Before** to add another element above the selected element
   - **Add After** to add another element below the selected element

   Note: The InFocus Design Studio XML editor is Context sensitive to InFocus schemas. This means that when adding elements and attributes to an XML file with an InFocus schema, you can select an element or attribute from a list of the possible values (depending on the schema definition). This list is available as a submenu.

3. Provide a name for the element or attribute if required. You may also be able to select the element from a submenu. The element or attribute will be added to the file.
4. Save the file, then select it again in the Project Explorer and press F5 to refresh.

   The XML file is updated automatically.

Replace an Element

You can replace an element with another legal element.

Do the following to replace an element

1. Right-click an element from the list in the editor.
2. Select **Replace with**.
3. Select an element from the submenu. Only legal elements are available.
4. The original element is replaced with the selected element.

Index

A

ACX trace parameter, 6-7
adapter inbound configuration, 4-1
add timestamp to traced events parameter, 6-7
Artithmetic fixed precision parameter, 6-12

B

binary XML log level parameter, 6-7
BPEL
  configuring inbound WDSL, 4-5
BPEL Process Manager, 4-1

C

cache
  maximum queries, 6-11
cache buffer size parameter, 6-9
code page parameter, 6-9
COM maximum XML size in memory
  parameter, 6-14
COM maximum XML size parameter, 6-15
COM XML transport buffer parameter, 6-15
commit on destroy, 6-13
continuous query
  prefix, 6-12
  retry interval, 6-11
  timeout, 6-11
continuous query prefix parameter, 6-12
continuous query retry interval parameter, 6-11
continuous query timeout parameter, 6-11

D

daemon
  logging, B-3
  security, B-6
  server modes, 6-2
  shutting down, 5-7
  starting, 5-7
  timeout, 5-11
data sources, disabling caching, 6-10
data types
  NLS string parameter, 6-9
debug parameters
  ACX trace, 6-7
  add timestamp to traced events, 6-7
  environment, 6-7
  GDB trace, 6-7
  general trace, 6-7
  lbinary XML log level, 6-7
  log file, 6-8
  optimizer trace, 6-8
  query processor trace, 6-7
  query warnings, 6-7
  trace directory, 6-8
  Transaction extended logging, 6-8
disable 2PC, 6-13
disable command reuse parameter, 6-10
disable compilation cache parameter, 6-11
disable DS property cache parameter, 6-10
disable insert parameterization parameter, 6-10
disable metadata caching parameter, 6-10
disable query parameterization parameter, 6-10
disable row mark fail fetch parameter, 6-10
disable SQS cache parameter, 6-12
disable threads parameter, 6-10
disabling
  parameterization of constants, 6-10
  parameterization of queries, 6-10
disk space requirements
  Windows, 2-2
dsm maximum buffer size parameter, 6-14
dsm middle buffer size parameter, 6-14
dsmMaxSortBufferSize parameter, 6-14

E

environment parameters
  ACX trace, 6-7
  add timestamp to traced events, 6-7
  Artithmetic fixed precision, 6-12
  cache buffer size, 6-9
  code page, 6-9
  COM maximum XML size, 6-15
  COM maximum XML size in memory, 6-14
  COM XML transport buffer, 6-15
  commit on destroy, 6-13
  continuous query prefix, 6-12
  continuous query retry interval, 6-11
  continuous query timeout, 6-11
  debug, 6-7

disable 2PC, 6-13
disable command reuse, 6-10
disable compilation cache, 6-11
disable DS property cache, 6-10
disable insert parameterization, 6-10
disable metadata caching, 6-10
disable query parameterization, 6-10
disable row mark fail fetch, 6-10
disable SQS cache, 6-12
disable threads, 6-10
dsm maximum buffer size, 6-14
dsm maximum sort buffer Size, 6-14
dsm middle buffer size, 6-14
expose XML fields, 6-13
file close on transaction, 6-14
file pool size, 6-14
file pool size per file, 6-14
first tree extensions, 6-11
GDB trace, 6-7
general trace, 6-7
hash buffer size, 6-14
hash enable RO, 6-14
hash max open files, 6-14
hash primary event size, 6-14
hash secondary event size, 6-14
ignore segment bind failure, 6-11
insert from select commit rate, 6-12
language, 6-9
log file, 6-8
maximum columns in parsing, 6-11
Maximum segmented database thread, 6-11
maximum SQL cache, 6-11
minimum number of parameters allocated, 6-11
miscellaneous, 6-8
nav utility editor, 6-8
NLS string, 6-9
odbc, 6-9
oledb, 6-9
optimistic for updates, 6-11
optimizer, 6-9
optimizer goal, 6-9
optimizer trace, 6-8
parser depth, 6-12
procedures cache size, 6-12
prompt database user pasword, 6-11
query processor trace, 6-7
query warnings, 6-7
queryProcessor, 6-10
recovery delay, 6-13
Replace invalid XML characters, 6-15
time limit, 6-13
token size, 6-12
trace directory, 6-8
transaction conversions parameters, 6-13
Transaction extended logging, 6-8
transaction extended logging, 6-13
transaction log file, 6-13
transactions, 6-13
ttemporary directory, 6-8
tuning, 6-13
use alternate qualifier, 6-11
use global file pool, 6-14
use table filter expression, 6-11
user commit confirm table, 6-13
write empty string as null, 6-11
XML, 6-14
XML date format parameter, 6-15
XML trim char column, 6-15
year 2000 policy, 6-8
error log, binary XML log level parameter, 6-7
error log, log file parameter, 6-8
exact arithmetic, 6-12
expose XML fields parameter, 6-13

F

file close on transaction, 6-14
file pool size, 6-14
file pool size per file, 6-14
first row optimization, optimizer goal parameter, 6-9
first tree extensions parameter, 6-11
floating point precision, 6-12

G

GDB trace parameter, 6-7
general parameters
cache buffer size, 6-9
nav utility editor, 6-8
temporary directory, 6-8
year 2000 policy, 6-8
general trace parameter, 6-7

H

hardware requirements
Windows disk space, 2-2
Windows memory, 2-2
Windows processor, 2-2
hash buffer size, 6-14
hash buffer size parameter, 6-14
hash enable RO parameter, 6-14
hash joins, 6-14
hash max open files parameter, 6-14
hash primary event size parameter, 6-14
hash secondary event size parameter, 6-14

I

ignore segment bind failure parameter, 6-11
index cache
dsm middle buffer size parameter, 6-14
insert from select commit rate parameter, 6-12
installing
Solaris Operating System (SPARC), 2-4

J

joins
hash buffer size parameter, 6-14

hash max open files parameter, 6-14

L

language parameter, 6-9
language parameters
  code page, 6-9
  environment, 6-9
  language, 6-9
  NLS string, 6-9
locking, optimistic, 6-11
log file parameter, 6-8
log files
  binary XML log level parameter, 6-7
  daemon options, B-3
  log file parameter, 6-8
logging
  daemon configurations, B-3
  optimizer strategy, 6-8

M

maximum columns in parsing parameter, 6-11
Maximum segmented database thread, 6-11
maximum SQL cache parameter, 6-11
memory
  dsm maximum buffer size parameter, 6-14
  dsm maximum sort buffer Size parameter, 6-14
  dsm middle buffer size parameter, 6-14
memory requirements
  Windows, 2-2
metadata
  disabling retrieval from cache, 6-10
Microsoft software requirements, 2-2
minimum number of parameters allocated
    parameter, 6-11
miscellaneous parameters
  environment, 6-8

N

nav utility editor parameter, 6-8
nav utility editor, text editor, 6-8
NLS string parameter, 6-9

O

odbc environment parameters, 6-9
oledb environment parameters, 6-9
operating system requirements
  Windows, 2-2
optimistic for updates parameter, 6-11
optimizer
  environment parameters, 6-9
  optimizer goal parameter, 6-9
  trace directory parameter, 6-8
optimizer goal parameter, 6-9
optimizer trace parameter, 6-8

P

parallel processing parameters
  disable threads, 6-10
parameterization of constants, disabling, 6-10
parameterization of queries, disabling, 6-10
parser depth parameter, 6-12
password, setting automatic prompt, 6-11
post-installation, Solaris Operating System
    (SPARC), 2-5
preinstallation, Solaris Operating System
    (SPARC), 2-3
procedures cache size parameter, 6-12
processor requirements
  Windows, 2-2
prompt database user pasword parameter, 6-11

Q

queries
  disabling reuse, 6-10
  disabling saved compilation, 6-11
  enabling retrieval after failure, 6-10
  maximum cached in memory, 6-11
  maximum length of string, 6-12
  maximum size after compilation, 6-11
query optimizer
  logging strategy, 6-8
  optimizer goal parameter, 6-9
  trace directory parameter, 6-8
query processor parameters
  Artithmetic fixed precision, 6-12
  continuous query prefix, 6-12
  continuous query retry interval, 6-11
  continuous query timeout, 6-11
  disable command reuse, 6-10
  disable compilation cache, 6-11
  disable DS property cache, 6-10
  disable insert parameterization, 6-10
  disable metadata caching, 6-10
  disable query parameterization, 6-10
  disable row mark fail fetch, 6-10
  disable SQS cache, 6-12
  emptyStringISNull parameter, 6-11
  expose XML fields, 6-13
  first tree extensions, 6-11
  ignore segment bind failure, 6-11
  insert from select commit rate, 6-12
  maximum columns in parsing, 6-11
  Maximum segmented database thread
      parameter, 6-11
  maximum SQL cache, 6-11
  minimum number of parameters allocated, 6-11
  optimistic for updates, 6-11
  parser depth, 6-12
  procedures cache size, 6-12
  prompt database user pasword, 6-11
  token size, 6-12
  use alternate qualifier, 6-11
  use table filter expression, 6-11
query processor trace parameter, 6-7

query warnings parameter, 6-7
queryProcessor environment parameters, 6-10

R

recovery delay, 6-13
Replace invalid XML characters parameter, 6-15
requirements
  Windows hardware requirements, 2-1
  Windows software requirements, 2-2
row optimization, optimizer goal parameter, 6-9

S

security
  daemon configurations, B-6
  setting automatic prompt, 6-11
segmented data sources, response upon failure, 6-11
servers
  configuring modes, 6-2
  reusable, 6-3
  Reuse limit daemon parameter, 6-3, B-13
software requirements
  Microsoft, 2-2
  Windows operating system, 2-2
Solaris Operating System (SPARC)
  installing, 2-4
  post-installation, 2-5
  preinstallation, 2-3
sort buffer, dsm maximum sort buffer size
    parameter, 6-14
staging area, 1-5
stored queries, maximum in cache memory, 6-12

T

temporary directory parameter, 6-8
temporary files, 6-8
threading model, 6-11
time limit parameter, 6-13
timeout
  client idle, 5-11
  daemon, 5-11
token size parameter, 6-12
trace directory parameter, 6-8
trace information, logging, 6-7
transaction conversions parameters, 6-13
transaction extended logging, 6-13
Transaction extended logging parameter, 6-8
transaction log file, 6-13
transactions environment parameters, 6-13
transactions parameters
  commit on destroy, 6-13
  disable 2PC, 6-13
  recovery delay, 6-13
  time limit, 6-13
  transaction conversions parameters, 6-13
  transaction extended logging, 6-13
  transaction log file, 6-13
  user commit confirm table, 6-13
tuning environment parameters, 6-13
tuning parameters
  dsm maximum buffer size, 6-14
  dsm maximum sort buffer Size, 6-14
  dsm middle buffer size, 6-14
  file close on transaction, 6-14
  file pool size, 6-14
  file pool size per file, 6-14
  hash buffer size, 6-14
  hash enable RO, 6-14
  hash max open files, 6-14
  hash primary event size, 6-14
  hash secondary event size, 6-14
  use global file pool, 6-14

U

use alternate qualifier parameter, 6-11
use global file pool, 6-14
use table filter expression parameter, 6-11
user commit confirm table, 6-13

W

Workspace server mode, B-13
write empty string as null parameter, 6-11

X

XML
  environment parameters, 6-14
XML date format parameter, 6-15
XML parameters
  COM maximum XML size, 6-15
  COM maximum XML size in memory, 6-14
  COM XML transport buffer, 6-15
  Replace invalid XML characters, 6-15
  XML date format parameter, 6-15
  XML trim char column, 6-15
XML trim char column parameter, 6-15

Y

Y2K
  See year 2000 policy parameter
year 2000 policy parameter, 6-8