INXIRE®

Installation Guide

Enterprise Content Management SDK

Version 3.0

Primary Author: Frank Closheim
Contributing Authors: Dave Long, Denis Scholler

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.
Oracle, SQL*Plus, PL/SQL, Solaris and Java are registered trademarks of Oracle and/ or its affiliates. Other names may be trademarks of their respective owners.
PostgreSQL and Postgres are registered trademarks of the PostgreSQL Community Association of Canada.

Table of Contents

1   Preface

Enterprise Content Management SDK (ECMSDK) is a robust and flexible runtime and development platform for building demanding content management applications. Prior to version 1.0, ECMSDK was known as Oracle Content Management SDK (Oracle CM SDK). The features and capabilities of ECMSDK are designed to help developers bring enterprise-scalable content management-based applications to market faster and better than with any other platform. ECMSDK provides a set of Java APIs that developers can use to customize or extend the product's core functionality in numerous ways to build comprehensive content-centric database solutions.

This document contains necessary information for the installation and configuration of ECMSDK Release 3.0

Note: This document is accurate at the time of publication. inxire will update the documentation periodically after the software release. You can access the latest information and additions to this document on the ECMSDK website at:

http://www.ecmsdk.com

1.1   Audience

The instructions in this ECMSDK Installation Guide are intended for anyone responsible for installing a new ECMSDK instance, upgrading an existing one, or migrate from Oracle CM SDK.

1.2   Related Documents

For more information, see these additional resources:

Oracle Database

- Oracle Database Installation Guide

- Oracle Database Backup and Recovery Basics

- Oracle Database SQL Language Reference

- Oracle Database JDBC Developer's Guide and Reference

- Oracle Text Reference

PostgreSQL Database

- Comprehensive Manual

Java

- Java EE Platform Installation Guide

- Java EE Platform API Specification

Oracle Content Management SDK

- Oracle CM SDK Data Sheet

2   Introduction to Installing ECMSDK

ECMSDK is written entirely in Java, with various components running in separate Java Virtual Machines (JVMs) on a middle-tier computer. One key function of the Java application code that runs on the middle tier is to map Java objects to relational database tables.

Starting with ECMSDK 3.0, ECMSDK uses either an Oracle Database or PostgreSQL for its content repository, and can run in a third party Application Server (optional).

ECMSDK can make use of a third party Lightweight Directory Access Protocol (LDAP), for user management and authentication. Because of these dependencies, installing ECMSDK involves the following:

- Installing a new Oracle Database or PostgreSQL, or adjusting the parameters of an existing instance

- Creating custom tablespaces for ECMSDK

- Setting up a JDK on the ECMSDK server

- Creating a ECMSDK repository

- Installing one or more ECMSDK middle tiers

This guide provides information about how to install an ECMSDK repository and ECMSDK middle tiers. It includes some of the information, but not all of the information, you need to install the other components. References to specific guides are provided in key places in this guide to help you through the installation process for all of these components.

2.1   Deployment Configuration

ECMSDK is typically deployed on multiple computers in a production environment. This configuration enables you to achieve better performance and throughput, and configure failover, load balancers, and other high availability options. With a multiple-computer deployment, you can also use computers with lower hardware requirements than required for a single-computer deployment.

ECMSDK can be installed on a single computer if the computer meets the recommended hardware and software requirements. If your computer does not meet the recommended requirements, the performance of this configuration might be less than satisfactory. A single-computer deployment does not allow you to use hardware-side load balancing or failover options and is primarily useful for demonstration and proof of concept applications.

See Also: *ECMSDK Administration Guide* for more information about planning for deployments.

3   Pre-Installation Requirements

This chapter describes required prerequisites and important procedures you should perform prior to making use of ECMSDK.

3.1   Software Requirements

ECMSDK depends on other software components. Their required minimum versions are listed below.

- Oracle Database 11g (11.2.0.2), or higher
  or
  Oracle Database Express Edition 11g Release 2, or higher
  or
  PostgreSQL relational database 10.6, or higher

- JDK 1.8 (version 1.8.0), or higher

3.2   Restrictions when using Oracle Database Express Edition

When installing ECMSDK against an Oracle Database Express Edition 11g (Oracle Database XE) or higher, the following restrictions apply:

1. Using External LOB (BFILEs) with ECMSDK is currently not supported with Oracle Database XE.

2. ECMSDK CredentialManager only supports CleartextCredential with Oracle Database XE. AcceptChallengeResponseCredential, AcceptHttpDigest-Credential and AcceptTokenCredential are currently not supported.

3.3   Restrictions when using PostgreSQL

When installing ECMSDK against a PostgreSQL 10.6 or higher, the following restrictions apply:

1. ECMSDK CredentialManager only supports CleartextCredential with Oracle Database XE. AcceptChallengeResponseCredential, AcceptHttpDigest-Credential and AcceptTokenCredential are currently not supported.

4   Installing the Content Repository

Starting with ECMSDK 3.0, the content repository can either be in an Oracle Database or PostgreSQL. The following chapters describe the installation of both databases in a nutshell. Further documentation should be consulted for complete installation instructions.

Only one type of database is required for ECMSDK. Restrictions might apply. See chapter 3.

4.1   Installing an Oracle Database

If you already have an Oracle Database installed, skip this step. Adjusting the parameters of the database will be covered in section 4.1.1.

For installation information about an Oracle Database, please refer to the *Oracle Database Installation Guide.* The installation procedure may be slightly different, depending on which option you select and which database version you are installing.

Note: When you refer to books in the Oracle Database documentation library, make sure that the version and platform of the book you are looking at matches the version and platform of the Oracle Database you are installing.

4.1.1   Adjusting the Parameters of an Existing Oracle Database

If you already have an Oracle Database installation available that can be used as the ECMSDK repository you must first set initialization parameters and the required database character set (ALUTF8) for the Oracle Database.

You can use the following SQL command to set individual parameters:

```
SQL> alter system set parameter_name=parameter_value
scope=both;
```

For example:

```
SQL> alter system set shared_pool_size=184549376
scope=both;
```

Configure your Oracle Database using the parameters shown in the Table 1 below.

Parameter	Minimum Value
java_pool_size	32 MB
job_queue_processes	4
open_cursors	300

Parameter	Minimum Value
`processes`	100
`session_max_open_files`	50

Table 1: Required Oracle Database Initialization Parameters

To check the value of the parameters, connect to the database using SQL*Plus and run the following query of the `v$parameter` view:

```
SQL> SELECT name, value FROM v$parameter WHERE name =
'processes';
```

To set the database character set for an existing database, you can use the Database Configuration Assistant (DBCA) to re-create the database with the correct character set, ALUTF8.

4.1.2   Creating ECMSDK Tablespaces

During the ECMSDK configuration, you must select in which tablespaces to store all schema objects. For a production environment, it is recommended to create custom tablespaces for the various tables and indexes that comprise the system before attempting to configure ECMSDK. The following section describes the requirements for creating custom tablespaces for ECMSDK.

To create custom tablespaces for ECMSDK, it is recommended that you create tablespaces as locally-managed tablespaces. Locally-managed tablespaces track all extent information in the tablespace itself, using bitmaps, resulting in simplified space allocation, ease of management, and performance benefits. The following is an example of the SQL syntax used to create custom tablespaces:

```
SQL> CREATE TABLESPACE "IFS_MAIN" DATAFILE
'/u01/app/oracle/product/oradata/orcl/ifsmain_01.dbf' SIZE
50M AUTOEXTEND ON LOGGING EXTENT MANAGEMENT LOCAL;
```

Note: You must specify the `AUTOEXTEND` parameter when associating datafiles with a custom tablespace. If you do not specify this parameter, ECMSDK might not fully work.

Table 2 lists the different types of data stored in ECMSDK and describes the purpose of each tablespace.

Storage Type	Example Name	Description
Metadata	`IFS_MAIN`	Stores metadata for documents, information about users and groups, and other ECMSDK object data.

Storage Type	Example Name	Description
Document Storage	`IFS_LOB`	Stores the Large Object (LOB) data for documents
Oracle Text (Token)	`IFS_CTX_I`	Stores words (tokens) extracted by Oracle Text from ECMSDK documents (the Oracle table `DR$IFS_TEXT$I`).
Oracle Text (Index)	`IFS_CTX_X`	Stores the Oracle B*tree index on the Oracle Text tokens (the Oracle index `DR$IFS_TEXT$X`).
Oracle Text (Other)	`IFS_CTX_K`	Stores miscellaneous Oracle Text tables (the Oracle tables `DR$IFS_TEXT$K`, `DR$IFS_TEXT$N`, `DR$IFS_TEXT$R`).
General Storage	`TEMP`	Temporary data during transactions

Table 2: Tablespace Definitions

4.2   Installing PostgreSQL Relational Database

If you already have a PostgreSQL installed, skip this step. The default parameters of the PostgreSQL relational database can be used in a ECMSDK test environment. Depending how much users, data and other load is necessary for your ECMSDK production environment, you might need to adjust them according to your needs.

For installation information about a PostgreSQL relational database, please refer to the *PostgreSQL Comprehensive Manual*. The installation procedure may be different, depending on which database version you are installing.

Note: When you refer to books in the PostgreSQL documentation library, make sure that the version and platform of the book you are looking at matches the version and platform of the PostgreSQL you are installing.

1.

5   Installing a Middleware

ECMSDK's only dependency on middleware is the requirement of Java 8 compliance. You can run the ECMSDK Nodes in any Java 8 compliant middleware, such as Apache Tomcat, Oracle GlassFish Server, Oracle WebLogic Server, or Red Hat JBoss.

Note: Refer to your specific middleware documentation for detailed installation instructions.

5.1   Installing a JDK

ECMSDK is written in pure Java and requires a Java Runtime Environment (JRE) available on the ECMSDK server. ECMSDK supports JDK version 1.8.0 or higher.

The following section refers to the Oracle JDK. Starting with ECMSDK 3.0 we also support **OpenJDK** as a JRE. Installing OpenJDK might differ from the Oracle JDK. Please refer to the corresponding documentation for more details.

See Also: See *JDK 8 and JRE 8 Installation Guide* for general information about installing JDK 8 and JRE 8 and *Oracle JDK 8 and JRE 8 Certified Systems Configurations* for information about supported platforms, operating systems, and browsers.

This section provides streamlined instructions for installing JDK 8 using archived binaries for 64-bit Linux. The following procedure may be slightly different from your actual installation, depending on which installation package you downloaded and which Java version you are installing. If you already have JDK installed on your ECMSDK server, you can skip this step.

To install the JDK 8 for 64-bit Linux, using an archive binary file (`.tar.gz`):

2. **Download the file.** The archive binary can be installed by anyone (not only root users), in any location that you can write to.

3. **Change directory** to the location where you would like the JDK to be installed. Move the `.tar.gz` archive binary to the current directory.

4. **Unpack** the tarball and **install** the JDK:

```
bash> tar zxvf jdk-8u<version>-linux-x64.tar.gz
```

   The Java Development Kit files are installed in a directory called `jdk1.8.0_<version>` in the current directory.

Delete the `.tar.gz` file if you want to save disk space.

6   Installing ECMSDK

This chapter guides you through the process of installing and configuring ECMSDK. To perform the non-interactive installation of ECMSDK you need to follow the steps below:

- Software Extraction

- Preparing the Installation

- Creating a New ECMSDK Schema

The examples in this chapter are specific to a Linux environment; use the appropriate commands for your operating environment.

6.1   Software Extraction

Once you have fulfilled the requirements listed in the previous chapter 3, perform the following steps.

Note: For any text in this section containing the following notation `<version>`, you must substitute the appropriate ECMSDK version number for the notation.

1. Copy the `ecmsdk-<version>-bin.zip` file to your target installation directory.

   For example:

   ```
   bash> cp ecmsdk-<version>-bin.zip /opt
   ```

2. Unzip the `ecmsdk-<version>-bin.zip` file to your target directory using the following commands:

   ```
   bash> cd /opt

   bash> unzip ecmsdk-<version>-bin.zip
   ```

   A directory called `ecmsdk` is created inside the target directory.

6.2   Preparing the Installation

Before you can start the non-interactive ECMSDK installation process you need to set the following environment variables using the appropriate operating system command (for example, `setenv`):

- `ECMSDK_HOME`: The absolute path to your `ecmsdk` directory (for example: `/opt/ecmsdk`)

- `JAVA_HOME`: The location where your JDK is installed

In addition, the `Environment.properties` file has to be configured prior installing ECMSDK to match your specific environment.

1. **Navigate** to the `$ECMSDK_HOME/lib` directory.

2. **Edit** the `Environment.properties` file to match your specific environment. Follow the instructions provided inside the file.

   Note: If you are installing ECMSDK against an Oracle Database Express Edition, make sure the parameter `IFS_CM_TYPE_XE_BOOL` is set to `true`.

   See Also: Chapter 3.2 for any restrictions that may apply.

3. **Save** the `Environment.properties` file back in the `$ECMSDK_HOME/lib` directory with the same filename.

Important Note: Do **NOT** store any copies of this file in the `$ECMSDK_HOME/lib` directory, even with modified names for backup purposes. This can corrupt the ECMSDK schema creation process.

6.3   Creating a New ECMSDK Schema

If you want to create a new empty ECMSDK schema, continue with this section 6.3. To patch an existing ECMSDK or Oracle CM SDK 9.0.4.6 installation, continue with chapter 7.
Direct upgrade of an existing Oracle CM SDK 9.0.4.0 schema is not supported; see details in chapter 8.

ECMSDK supports the separation of duty model for the installation of a new ECMSDK schema starting with ECMSDK version 2.1.

The two ways of installing ECMSDK are:

- Single-role installation

- Separation of duty installation

6.3.1   Single-role installation

If the Application Administrator can get the system credentials on the repository database, the following instructions apply.

To create a new ECMSDK schema:

1. Navigate to the `$ECMSDK_HOME/bin` directory

2. To make the file `ecmsdk` executable you might need to change the file permissions.

   For example with:

   ```
   bash> chmod u+x ecmsdk
   ```

3. Start a new ECMSDK installation by running
   `./ecmsdk install`

4. When the command completes, you have successfully created a new ECMSDK schema

Check `$ECMSDK_HOME/logs/CreateSchema.log` for any errors during installation.

You can now continue with chapter 9.

6.3.2   Separation of duty installation

There are cases where the Application Administrator don't get admin access on the repository database. For those cases the installation activities can be separated into DBA (Database Administrator) and Application Administrator (AA) work.

In short the steps are:

1. Drop any existing schema (as DBA)

2. Pre-installation steps (as DBA)

3. ECMDK schema installation (as AA)

4. Post-installation steps (as DBA)

The following instructions describe the details of the separation of duty installation process.

To create a new ECMSDK schema:

1. As the Application Administrator make sure the `Environments.properties` parameter `SKIP_SYSTEM_ACTIONS` is set to `true`.

   See Also: Chapter 6.2 for more details.

2. As a Database Administrator login to your repository database with a DBA account (typically SYSTEM or SYS). For example:

   ```
   bash> $ORACLE_HOME/bin/sqlplus /nolog

   SQL> CONNECT
   SYSTEM/your_system_password@connect_descriptor
   ```

3. Execute the SQL script `$ECMSDK_HOME /install/sql/SystemDropSchema.sql` against the repository database to drop any existing ECMSDK schemas using the following syntax:

   ```
   SQL> @SystemDropSchema.sql {ECMSDK_SCHEMA_NAME}
   ```

   For example:

   ```
   SQL> @SystemDropSchema.sql ecmsdk
   ```

Note: This script can log some expected errors if the schemas that it's trying to drop do not exist.

4. Execute the SQL script `$ECMSDK_HOME /install/sql/SystemPreInstall.sql` against the repository database to prepare the ECMSDK installation using the following syntax:

```
SQL> @SystemPreInstall.sql {ECMSDK_SCHEMA_NAME}
{ECMSDK_SCHEMA_PASSWORD} {ECMSDK_MAIN_TBS_NAME}
{ECMSDK_TEMP_TBS_NAME}
```

For example:

```
SQL> @SystemPreInstall.sql ecmsdk ecmsdk IFS_MAIN TEMP
```

Note: This script can log an expected error which can be ignored as it will be fixed in a later step.

```
Warning: Procedure created with compilation errors.

Errors for PROCEDURE CTXSYS.ECMSDK_WP:

LINE/COL ERROR

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

3/7      PL/SQL: Statement ignored

3/7      PLS-00201: identifier 'ECMSDK.ECMSDK_WP' must
be declared
```

5. As the Application Administrator navigate to the `$ECMSDK_HOME/bin` directory

6. To make the file `ecmsdk` executable you might need to change the file permissions.

   For example with:

```
bash> chmod u+x ecmsdk
```

7. Start a new ECMSDK installation by running `./ecmsdk install`

8. As a Database Administrator login to your repository database with a DBA account (typically SYSTEM or SYS). For example:

```
bash> $ORACLE_HOME/bin/sqlplus /nolog

SQL> CONNECT
SYSTEM/your_system_password@connect_descriptor
```

9. Execute the SQL script `$ECMSDK_HOME /install/sql/SystemPostInstall.sql` against the repository database to execute the post installation steps using the following syntax:

```
SQL> @SystemPostInstall.sql {ECMSDK_SCHEMA_NAME}
```

For example:

```
SQL> @SystemPostInstall.sql ecmsdk
```

10. When the command completes, you have successfully created a new ECMSDK schema

Check `$ECMSDK_HOME/logs/CreateSchema.log` for any errors during installation.

You can now continue with chapter 9.

7   Patch an Existing ECMSDK Installation

If you want to patch an existing ECMSDK installation, continue with this chapter 7.
To create a new empty ECMSDK schema, see section 6.3.
Direct upgrade of an existing Oracle CM SDK 9.0.4.0 schema is not supported; see details in chapter 8.

Important Note: The complete patch process should be tested carefully before considering patching a production environment.
If you make use of the patch feature of this software, you must take a full backup of your current ECMSDK environment and database repository, as this upgrade can **NOT** be de-installed once you patch from a previous ECMSDK schema.

In order to patch an existing ECMSDK installation you need to perform the following preparation steps:

1. Navigate to the `$ECMSDK_HOME/lib` directory

2. Open file `Environment.properties` for edit

3. In case you copied the file `Environment.properties` from a previous installation, make sure that all parameters of the `Environment.properties` file bundled with the latest patch are present in your current file.

4. Save the `Environment.properties` file back in the `$ECMSDK_HOME/lib` directory with the same filename.

5. Stop all running ECMSDK Nodes and custom ECMSDK applications which have access to the ECMSDK schema

6. Navigate to the `$ECMSDK_HOME/bin` directory

7. To make the file `ecmsdk` executable you might need to change the file permissions.

    For example with:

    ```
    bash> chmod u+x ecmsdk
    ```

8. Verify ECMSDK software and schema versions by running
    `./ecmsdk version`

9. Start the ECMSDK patch process by running
    `./ecmsdk install`

10. When the command completes, you have successfully upgraded an existing ECMSDK schema.

    Important Note: To fully patch an existing ECMSDK software you need to re-deploy **ALL** your ECMSDK Nodes into your middleware. Please see section 9 for detailed instructions.

Check `$ECMSDK_HOME/logs/PatchEcmsdk.log` for any errors during the upgrade.

You must now continue with chapter 9.

8   Upgrade From an Existing Oracle CM SDK Installation

If you want to create a new empty ECMSDK schema, see section 6.3.
To patch an existing ECMSDK or Oracle CM SDK 9.0.4.6 installation, follow
instructions in chapter 7.

This chapter describes the necessary steps to upgrade an existing Oracle CM SDK
9.0.4.0.0 schema to an ECMSDK schema.

Important Note: Starting with ECMSDK 2.0, direct upgrade from Oracle CM SDK
installations are not supported anymore. If you need to upgrade from an existing
Oracle CM SDK 9.0.4.0 schema, you first need to upgrade to ECMSDK 1.0.
For detailed upgrade instructions, please refer to the *ECMSDK Installation Guide
Version 1.0.*
After upgrading to ECMSDK 1.0 you can follow chapter 7 of this document for
patching an existing ECMSDK installation.

9   Post-Installation Steps

The instructions in this chapter assume that you have completed all of the pre-installation and configuration tasks described in previous chapters.

ECMSDK is written in pure Java with various components running in separate Java Virtual Machines (JVMs) on a middle-tier computer. ECMSDK has no dependencies on specific middleware. The only requirements for ECMSDK are a Java 8 compliant JVM or middleware.

There are two options of running ECMSDK Nodes in an environment, container managed or standalone. Both are described below.

Note: It is highly recommended to run ECMSDK Nodes, as these Nodes constitute the ECMSDK runtime environment, and execute and manage the ECMSDK threads, important background processes and protocol servers.

9.1   Deploying ECMSDK Nodes

One option of running ECMSDK Nodes is container managed on supported middleware.

See Also: Read section 5 for installation instructions and more information on supported middleware.

Once you have installed a Java 8 compliant middleware and created a new ECMSDK schema or finalized the upgrade of an existing ECMSDK schema, you are able to deploy and run new ECMSDK Nodes on your middle-tier computer.

The following steps only highlight the general principles, as the specific details are different for each middleware product. Please refer to the specific middleware documentation for detailed administration and configuration instructions.

1. Make sure your target Java container is not running before deploying a new ECMSDK Node.

2. To be able to connect to your new ECMSDK schema, you need to create a JDBC Data Source in your middleware, for example, `jdbc/OracleDS`, and set the database connection properties to point to your ECMSDK schema inside your Oracle Database.

3. Use command line, a supported script language, or the Administration Console of your middleware to deploy an ECMSDK Node. The Web Application Archive (WAR) file can be found at `$ECMSDK_HOME/lib/ecmsdk.war`.

4. If you named your Data Source created in step 2 something other than `jdbc/OracleDS`, you need to edit the `web.xml` of the deployed `ecmsdk` application and update the context parameter `IFS.NODE.DataSourceName` with the Data Source name you used in step 2.

5. Start your Java container. You have now successfully created a new ECMSDK Node in your middleware.

6. Repeat steps 1-5 for any additional ECMSDK Nodes you want to create.

9.2   Running ECMSDK Nodes standalone

Instead of running ECMSDK Nodes in a container managed environment, you can also run them standalone.

Once you have installed a new ECMSDK schema or finalized the upgrade of an existing ECMSDK schema, you are able to run new ECMSDK Nodes standalone on your middle-tier computer.

Important Note: The WebDAV protocol server will **not** be available when running ECMSDK Nodes standalone, as it requires a Servlet container. All other features are available when running ECMSDK Nodes either standalone or in a managed container.

Perform these steps to start a new ECMSDK Node standalone.

1. Navigate to the $ECMSDK_HOME/bin directory

2. To make the file ecmsdk executable you might need to change the file permissions.

    For example with:

    ```
    bash> chmod u+x ecmsdk
    ```

3. Verify ECMSDK software and schema versions by running
   ./ecmsdk version

4. Start a new ECMSDK Node by running
   ./ecmsdk node.start

5. The command will prompt you for additional information. You can use the default values for most of them, if you do not use custom configurations. However, if you are running multiple Nodes you need to ensure that the name of each node is unique across your ECMSDK installation.

6. Once the command completes, you have successfully created a new ECMSDK Node and are running it standalone.

7. Repeat steps 1-6 for any additional ECMSDK Nodes you want to create.

9.3   Update JAR dependency in custom application

After creating a new, or upgrading an existing, ECMSDK schema, you can include the new ecmsdk-api.jar in your custom application. The JAR is located in $ECMSDK_HOME/lib/ecmsdk-api-<version>.jar.

9.4   Verify ECMSDK API Changes

There have been a set of API changes made for ECMSDK since its predecessor Oracle CM SDK. To make it easier for developers who would like to make use of these changes, we provide a set of "Javadoc Summary" pages with ECMSDK. These pages include the API summaries of Oracle CM SDK software versions 9.0.4.2.x and the latest ECMSDK version. The "Javadoc Summary" pages can be found in:

- `$ECMSDK_HOME/doc/api/9042x.zip` for the Oracle CM SDK APIs

- `$ECMSDK_HOME/doc/api/930x.zip` for the ECMSDK APIs

Extract these ZIP files before opening the summary files. The summary files will give an overview of the APIs organized by package. Comparing a specific 9.0.4.2.x summary file with the ECMSDK version of the same file will help developers find changes that have been made to the APIs shipped with this ECMSDK software.

10 De-Installation Instructions

De-Installation of ECMSDK is **fully** supported if you have created a new ECMSDK schema.

If you have upgraded from an existing ECMSDK schema, please refer to the corresponding Release Notes bundled with each ECMSDK release, to see if the particular release supports a rollback of the upgrade. Undoing an ECMSDK upgrade is mostly depending on schema changes that might not be able to rollback.

In all other cases you need to restore from the backup you took in section 7.

Perform these steps to entirely de-install ECMSDK from your machines.

1. Stop all processes that might be still running against your ECMSDK schema.

2. Login to your Oracle Database repository with a DBA account (typically SYSTEM or SYS). For example:

```
bash> $ORACLE_HOME/bin/sqlplus /nolog

SQL> CONNECT
SYSTEM/your_system_password@connect_descriptor
```

3. Execute the SQL script `$ECMSDK_HOME /install/sql/SystemDropSchema.sql` against the repository database with the following syntax:

```
SQL> @SystemDropSchema.sql {ECMSDK_SCHEMA_NAME}
```

   For example:

```
SQL> @SystemDropSchema.sql ecmsdk
```

4. Remove your `$ECMSDK_HOME` directory from your ECMSDK server.

5. Unset the following environment variables using the appropriate operating system command (for example, `unsetenv`):

   - `ECMSDK_HOME`

   - `JAVA_HOME`

You have now successfully de-installed ECMSDK.

11   Troubleshooting ECMSDK Installations

This chapter describes common installation and configuration problems and solutions.

11.1   Verifying ECMSDK Requirements

Review the following information before performing any of the troubleshooting steps in this chapter:

- Ensure that the computer meets the installation requirements specified in chapter 3.

- Ensure that you have completed all of the pre-installation tasks specified in chapter 3.

- When installing ECMSDK, check that the Oracle Database with which you want to associate ECMSDK is running during installation.

11.2   Troubleshooting Installation Errors

If you encounter an error during the installation of ECMSDK due to incorrect information entered for any of the values in the `Environment.properties`

1. Note the error and review the installation logs for causes

   See Also: Section 11.2.1

2. Correct the values in `Environment.properties`

3. Remove the failed installation by following the steps in chapter De-Installation Instructions

4. Execute the scripts again.

See Also: Section 11.3

11.2.1   Location of Installation Log Files

Depending the type of installation or upgrade, ECMSDK writes the following log files during the installation/upgrade process:

- `$ECMSDK_HOME/logs/CreateSchema.log`

- `$ECMSDK_HOME/logs/PatchEcmsdk.log`

11.3   ECMSDK Installation and Configuration Problems and Solutions

Most installation and configuration errors involve failure to carefully follow pre-installation instructions. The following table describes some common installation and configuration problems, possible causes, and what you should do to correct the problem. Note that installation and configuration actions are captured in log files that you can examine to assist in troubleshooting efforts.

Problem	Probable Cause	Corrective Action
Invalid schema version	The version of the `ecmsdk-api.jar` doesn't match the ECMSDK schema version	Make sure that you always use the `ecmsdk-api.jar` version shipped with the ECMSDK release that was used to create or upgrade the ECMSDK schema.
BFile location not accessible (`IFS-32211`)	A BFILE location was specified that is not accessible by the database process user.	The path specified for `IFS_BFILE_BASE_PATH` in `Environment.properties` must be accessible by the database process user to be able to write content to this location.
Incorrect schema password for *$CM schema	Starting with Oracle Database 11g passwords become case-sensitive by default.	Starting with Oracle Database 11g schema passwords become case-sensitive by default. Make sure that you specify the correct case-sensitive schema password for all existing schemas in `Environment.properties`.
`ORA-00959: tablespace does not exist` occurs during schema creation	Specified tablespaces do not exist	Make sure that the tablespaces that have been specified in `Environment.properties` exist prior to the ECMSDK schema creation process. **See** chapter 4.1.2 for more details.
WebDAV Error under Tomcat	Tomcat uses a different mechanism for resources lookup. All configured entries and resources are placed in the java:comp/env portion of the JNDI namespace.	The web.xml need to have a different Data Source name entry for Tomcat: `<context-param> <description>The name of the DataSource for the ECMSDK repository</description> <param-name>IFS.NODE.DataSource Name</param-name> <param-value>java:/comp/env/jdb c/OracleDS</param-value> </context-param>`

Problem	Probable Cause	Corrective Action
`java.lang.ClassCastException` for JDBC objects occurs	JDBC object wrapping need to be disabled in your Java EE container	Disable the JDBC Data Type wrapping in your JDBC Connection Pool configuration of your Data Source. **Note:** Refer to your specific middleware documentation for detailed instructions.
`ORA-01920: user name 'ECMSDK' conflicts with another user or role name`	A schema user with that name already exists in the database	Either choose a different value for the `IFS_SCHEMA_NAME` in `Environment.properties` or set the `IFS_OVERWRITE_EXISTING_SCHEMA_BOOL` to `true` within the same file.
`Tomcat does not fully start`	`LogFilePath` is set to an invalid path or user has no permissions to write into path specified	Make sure that the web.xml of `ecmsdk.war` contains a valid path for context-parameter `LogFilePath`, and that the user you are starting Tomcat with has write permissions to the configured location. If no value is specified (default), the log file will be written to the standard runtime directory.

Table 3: Installation and Configuration Problems and Solutions

11.4   Need More Help?

For more help troubleshooting ECMSDK installation, go to http://www.ecmsdk.com

If you have a support contract with inxire, feel free to log a Support Ticket. The steps to download any available ECMSDK patches from ecmsdk.com are:

1. Go to http://www.ecmsdk.com.

2. Click Downloads.

3. View the latest Release Notes and go through the information before downloading the specific ECMSDK release.

If you do not find a solution for your problem, contact us:

inxire GmbH
Hanauer Landstr. 293 A

60314 Frankfurt, Germany

Telefon: +49 69 870000-10
Telefax: +49 69 870000-11

E-Mail: info@ecmsdk.com
Internet: http://www.ecmsdk.com