EMC® XDS Repository Connectors

Version 1.9

Installation Guide

EMC Corporation
Corporate Headquarters
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Legal Notice

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Documentation Feedback

Your opinion matters. We want to hear from you regarding our product documentation. If you have feedback about how we can make our documentation better or easier to use, please send us your feedback directly at ECD.Documentation.Feedback@emc.com

Table of Contents

Revision History

Revision Date	Description
February 2016	Initial publication.

Chapter 1

Before You Install

Before beginning installation, ensure that your system meets the requirements.

The *EMC XDS Repository Connectors Release Notes* contains information on the system requirements for your product. This documentation is available from EMC Online Support.

Chapter 2

Pre-installation Tasks

This chapter describes the pre-installation tasks and the steps to install the dependent libraries to successfully install the XDS Repository Connectors.

Pre-Installation Tasks for Connector for Documentum

You must complete the following tasks before installing the Connector for Documentum:

- Create a Documentum User Account for XDS Server
- Deploy the Documentum Object Types

Creating a Documentum User Account for XDS Server

The XDS Repository Server requires a user account to access the Documentum repository.

1. In the Documentum Content Server, create a Documentum user account named **`XdsServer`**.
2. Ensure that the `/Patients` cabinet exists in Documentum and the new user has unrestricted access and DELETE privileges on the `/Patients` cabinet in the newly created repository. The *EMC Documentum Administrator User Guide* provides the instructions for adding users.

   The *Configuring the HIM Documentum Metadata Properties* section in the *XDS Repository Connector for Documentum Administration Guide* provides details on cabinet configuration.
3. Note down the XDS Server user name and password for reference. You need these values for the next configuration step.

Deploying Object Types

The following DAR files contain the default object types used by HIP to install the repository server:

- `hip-dctm-1.9.0.dar`: Stores the Documentum Healthcare Information Model object types
- `hip-xds-1.9.0.dar`: Stores the HIP XDS object types

If you are deploying `hip-dctm-1.9.0.dar` as `dmadmin` on a Windows 2012 system, execute the DAR deployer with 'Run As Administrator' privilege to avoid receiving the following post-installation failure error '`Unable to create dmh indices`'.

1. Go to `<DCTM_install_dir>\product\7.2\install\composer\ComposerHeadless\`.
2. Double-click the `dardeployer.exe` file.
3. Define the following settings in the **DAR Details** and **Docbroker Details** section:
   - **DAR**: Click **Browse** and select the DAR file that you must deploy.
   - **Docbroker Host**: Select the Docbroker host.
   - **Docbroker Port**: Define the port number for the Docbroker host.
4. Click **Connect**.
5. Define the following settings in the **Repository Details** section:
   - **Repository**: Select the Documentum Repository from the list.
   - **User Name**: Type the Documentum Content Server **User Name** that you have already created.

     Creating a Documentum User Account for XDS Server, page 11 provides the instructions to create a user.
   - **Password**: Type the password for the Documentum Content Server user name.
6. Click **Install**.

   Deploy both the DAR files as described above.

Pre-Installation Tasks for Connector for xDB

You must install xDB before proceeding with the installation of XDS Repository Connector for xDB.

Setting up the Documentum xDB Healthcare Database

1. Install Documentum xDB on the system that hosts your Documentum xDB database.

   You may skip this step if you already have Documentum xDB in your environment. The *Documentum xDB Manual* provides more information about the installation instructions.
2. Create the Documentum xDB Healthcare database.

   Use the Documentum xDB Administrator tool to create a Healthcare database to hold registry data. Allocate enough resources to the default and temporary segments to match your performance requirements. Record the name of database for use in later configuration steps. This installation guide mentions **`Healthcare`** as a sample database name. The *Documentum xDB Manual* provides the instructions to create the database.
3. Establish user access for the Documentum xDB Healthcare database.

   Create a non-privileged user account in the Documentum xDB Healthcare database. The XDS Registry Server uses this account to access the Documentum xDB Healthcare database. Record the user name and password for use in later configuration steps. This installation guide mentions **`HealthcareServer`** as a sample user name.

Installing the Dependent Libraries

You must install the following third-party dependent libraries to successfully deploy the Repository Server WAR file for any XDS Repository Connector:

- Camel
- HL7 Application Programming Interface (HAPI)
- Integrating the Healthcare Enterprise (IHE)

Obtaining the Dependent Libraries

To obtain the Camel JAR files:

1. Create a folder in the local path to copy the Camel JAR files.

   For example:

   `C:\jarfiles\camel`
2. Go to www.camel.apache.org.
3. Download the following files:

   For Windows:

   `apache-camel-2.12.1.zip`

   For Linux:

   `apache-camel-2.12.1.tar.gz`
4. Extract the ZIP file to a local path.

   For example:

   `C:\apache-camel-2.12.1`
5. Go to the `<local path>\apache-camel-2.12.1` folder.
6. Copy the `lib` folder containing the JAR files to the `C:\jarfiles\camel` folder.

   For example:

   `C:\jarfiles\camel\lib`

To obtain the HAPI JAR files:

1. Create a folder in the local path to copy the HAPI JAR files.

   For example:

   `C:\jarfiles\hapi`
2. Go to www.sourceforge.net.
3. Download the `hapi-dist-2.0-all.zip` file.

4. Extract the ZIP file to a local path.

   For example,

   `C:\hapi-dist-2.0-all`

5. Go to the `<local path>\hapi-dist-2.0-all` folder.

6. Copy the `lib` folder containing the JAR files to the `C:\jarfiles\hapi` folder.

   For example:

   `C:\jarfiles\hapi\lib`

The HAPI JAR files are required only if you want to integrate repository with Epic using C4E.

Due to licensing restrictions, HIP products do not deliver the required HAPI library JAR files used when parsing HL7 feeds.

To obtain the IHE JAR files:

1. Create a folder in the local path to copy the IHE JAR files.

   For example:

   `C:\jarfiles\ihe`

2. Go to www.projects.openhealthtools.org.

3. Download the `org.openhealthtools.ihe_2.0.0.zip` file.

4. Extract the ZIP file to a local path.

   For example:

   `C:\openhealthtools\`

5. Go to the `C:\openhealthtools` folder.

6. Copy the following JAR files to the `C:\jarfiles\ihe` folder.

   - `org.openhealthtools.ihe.atna.context_2.0.0.jar`
   - `org.openhealthtools.ihe.atna.nodeauth_2.0.0.jar`
   - `org.openhealthtools.ihe.utils_2.0.0.jar`

7. Go to www.repo.openehealth.org.

8. Copy the following JAR file to the `C:\jarfiles\ihe` folder.

   `org.openhealthtools.ihe.atna.auditor-2.0.0-p4.jar`

To obtain the xDB JAR files (required only for XDS Repository Connector for xDB):

1. Create a folder in the local path to copy the xDB JAR files.

   For example:

   `C:\jarfiles\xdb`

2. Go to the xDB installation directory.

   For example:

   `C:\Program Files\xDB\`

3. Copy the `lib` folder containing the JAR files from `<xDB_install_dir>` to the `C:\jarfiles\xdb` folder.

For example:

```
C:\jarfiles\xdb\lib
```

Bundling the Dependent Libraries

1. From EMC Download Center, download the installation file for the connector that you want to install.

   For example:

   To install Connector for Documentum, download the `hip-cs-repository-1.9.0.zip` file.

   To install Connector for xDB, download the `hip-xdb-repository-1.9.0.zip` file.

2. Extract the ZIP file to a local folder.

   For example:

   For Connector for Documentum:

   ```
   C:\hip-cs-repository-1.9
   ```

   For Connector for xDB:

   ```
   C:\xdb-repository-1.9
   ```

   You can find the `build.xml` file in the extracted folder.

3. Go to the command prompt and navigate to the directory where `build.xml` is located.

   For example:

   For Connector for Documentum:

   ```
   C:\hip-cs-repository-1.9
   ```

   For Connector for xDB:

   ```
   C:\xdb-repository-1.9
   ```

4. Run `build.xml` using the following command:

   ```
   ant -f build.xml
   ```

   After you run the `ant -f build.xml` command, you get the `install` folder as follows:

   For Connector for Documentum:

   ```
   C:\hip-cs-repository-1.9\install
   ```

   For Connector for xDB:

   ```
   C:\xdb-repository-1.9\install
   ```

   You can find the `repository.war` file in the `install` folder of the respective connector.

5. When the script prompts you to enter the Camel home directory, type the complete path of the Camel home directory.

   For example:

   ```
   C:\jarfiles\camel
   ```

6. When the script prompts you to enter the HAPI home directory, type the complete path of the HAPI home directory.

   For example:

   ```
   C:\jarfiles\hapi
   ```

7. When the script prompts you to enter the IHE home directory, type the complete path of the IHE home directory .

   For example:

   ```
   C:\jarfiles\ihe
   ```

8. When the script prompts you to enter the xDB home directory, type the complete path of xDB home directory.

   For example:

   ```
   C:\jarfiles\xdb
   ```

   This step is required only for the installation of XDS Repository Connector for xDB.

Chapter 3

Configuring XDS Repository Server

The *XDS Repository Connectors Administration Guide* provides detailed information on the HIP configurations required for successful installation of XDS Repository Connector for Documentum, and xDB. Ensure that you complete the configurations before deploying the WAR files.

Chapter 4

Installing XDS Repository Connectors

This chapter describes the steps to install the XDS Repository Connectors.

Deploying the Connector WAR File Using Tomcat

1. Stop the J2EE Web Application container.
2. Go to the `install` folder obtained after running the build command.

   For example:

   For Connector for Documentum:

   ```
   C:\hip-cs-repository-1.9\install
   ```

   For Connector for xDB:

   ```
   C:\xdb-repository-1.9\install
   ```

3. Copy the WAR file to the following directory:

   ```
   <tomcat_install_dir>/webapps/
   ```

4. Start the J2EE Web Application container to expand the WAR file.

   The examples in this guide are provided with the assumption that the WAR file is deployed in the following location:

   ```
   <tomcat_install_dir>/webapps/
   ```

Deploying the Connector WAR File Using WebLogic

Before deploying the WAR file:

1. Set the following environment variable:

   ```
   Name: DOMAIN_HOME
   Value: ~\Oracle\Middleware\user_projects\domains\domain
   ```
   {domain used for deployment}

2. Set the following System Properties in the WebLogic startup script:

```
com.sun.xml.ws.spi.db.BindingContextFactory=com.sun.xml.ws.db.glassfish.JAXBRIContextFactory
javax.xml.bind.JAXBContext=com.sun.xml.bind.v2.ContextFactory
javax.wsdl.factory.WSDLFactory=com.ibm.wsdl.factory.WSDLFactoryImpl
```

3. Set the HIP home location in the startup script to obtain the log files generated in the `<user.home>/.hip/` folder.

   Example:

   ```
   set JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.emc.healthcare.home=C:\Users
   \<username>\.hip
   ```

1. Log in to the WebLogic Admin console.
2. Go to **base_domain** > **deployment**.
3. Click **Install**.
4. From **Install Application Assistant**, click the **upload your file** link in the Note.
5. From **Deployment Archive**, browse and select the WAR file.
6. Click **Next**.
7. Select **Install as application**.
8. Click **Next**.
9. Click **Finish**.
10. Check if the deployment state of server is Active.
    An Active state indicates a successful deployment.

Deploying the Connector for Documentum WAR File on Linux

1. Log in as **root** user.
2. Copy the Repository Connector for Documentum WAR file to the following location:

   ```
   $CATALINA_HOME/webapps
   ```

3. Run the following command to change the Tomcat installation owner to **dmadmin**.

   ```
   chown –R $CATALINA_HOME dmadmin:dmadmin
   ```

4. Set the **dmadmin** environment variables.
5. Set the HIP Java options in `$CATALINA_HOME/bin/setenv.sh`.

   ```
   JAVA_OPTS="-Xms512m -Xmx1g -XX:MaxPermSize=512m -Dcom.emc.healthcare.home=/home/dmadmin/.hip"
   ```

6. As **dmadmin**, copy the `.hip` folder to the following location:

   ```
   /home/dmadmin
   ```

7. As root user, create the Tomcat `startup/etc/init.d/tomcat` by setting the values appropriately.

   For example:

   ```
   #!/bin/bash
   ```

```
# description: Tomcat Start Stop Restart
# processname: tomcat
# chkconfig: - 90 10
# Source function library
. /etc/rc.d/init.d/functions
CATALINA_HOME=/app/apache-tomcat-7.0.42
TOMCAT_OWNER=dmadmin

#Set Startup Options for HIP
#See $CATALINA_HOME/bin/setenv.sh
#Check they have been used using ps-ef|grep tomcat
case $1 in
start)
        echo "Starting tomcat under dmadmin account..."
        echo "Note:xDB Must be Running or HIP Registry startup will fail..."
        su - $TOMCAT_OWNER -c "$CATALINA_HOME/bin/startup.sh"
        ;;
stop)
        echo "Stopping tomcat..."
        su - $TOMCAT_OWNER -c "$CATALINA_HOME/bin/shutdown.sh"
        ;;
restart)
        echo "Restarting tomcat under dmadmin account..."
        su - $TOMCAT_OWNER -c "$CATALINA_HOME/bin/shutdown.sh"
        sleep 2
        su - $TOMCAT_OWNER -c "$CATALINA_HOME/bin/startup.sh"
        sleep 2
        ;;
status)
        status tomcat
        ;;
*)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
        ;;
esac
exit 0
```

8. Change permissions and set Tomcat to auto-start on reboot.

```
chmod +x /etc/init.d/tomcat
chkconfig tomcat on
```

Chapter 5

Verifying the Installation

This chapter describes the steps to verify the installation of XDS Repository Connectors.

Verifying the Installation Using Tomcat

1. Ensure that the dependent libraries are installed.

   Installing the Dependent Libraries, page 13 provides details on installing dependent libraries.

2. Ensure that the `.hip` folder is available in the `C:\Users\<username>` folder.

   If you want to override the default location of the HIP configuration folder, override the `com.emc.healthcare.com` system property when you start the J2EE Web Application container.

   Use the following syntax:

   ```
   Dcom.emc.healthcare.home=<hip_config_directory>
   ```

3. Start the server using the normal start procedure for the J2EE web application container.

   For example, start the server on Tomcat using the following command:

   ```
   [root]# service Tomcat start
   ```

4. Check the log file for errors.

   For example:

   ```
   /usr/share/apache-Tomcat-7.0.42/logs/catalina.out
   ```

   Any error in the log file indicates an incorrect installation.

5. Open a web browser and type the URL for the server you installed:

   For example:

   ```
   http://<host:port>/repository/services
   ```

   The host must be the system where you installed the server and the port must be the secured HTTP port of Tomcat.

   The Web Service Definition Language (WSDL) page appears, which indicates that the server installation is successful.

Verifying the Installation Using WebLogic

1. Log in to the WebLogic Admin console.
2. Ensure that the dependent libraries are installed.

   Installing the Dependent Libraries, page 13 provides details on installing dependent libraries.
3. Ensure that the `.hip` folder is available in `C:\Users\<username>` folder.

   If the user does not have rights to access the `C:\Users\<username>` folder, perform the following steps:

   a. Create `.hip` folder in any other location.

   b. Go to `~\Oracle\Middleware\user_projects\domains\domain` {domain used for deployment}

   c. Add the following line to the `startWebLogic.cmd` file.

   ```
   set JAVA_OPTIONS=-Dcom.emc.healthcare.home=C:\.hip (".hip location")
   ```
4. Ensure that the configuration properties files are present in the `.hip\<server directory>` folder.
5. Restart the system for the above changes to take effect.
6. Start the WebLogic server.
7. Deploy the WAR file.

   Deploying the Connector WAR File Using WebLogic, page 19 provides the details.
8. Check the log file for errors.

   Any error in the log file indicates an incorrect installation.
9. Open a web browser and type the URL for the server you installed:

   For example:

   For Connector for Documentum:

   `http://<host:port>/cs-repository/services`

   For Connector for xDB:

   `http://<host:port>/xdb-repository/services/`

   The WSDL page appears, which indicates that the server installation is successful.

Chapter 6

Upgrading the Connectors

This chapter contains the instructions to upgrade the XDS Repository Connector for Documentum from version 1.8 to 1.9.

Upgrading XDS Repository Connector for Documentum from Version 1.8 to 1.9

1. To upgrade XDS Repository Connector for Documentum:
   a. Deploy the `hip-xds-1.9.0.dar` file.

   b. Deploy the `hip-dctm-1.9.0.dar` file.
2. Install the dependent libraries.

   The version of `org.openhealthtools.ihe.atna.auditor` jar file used by XDS Repository 1.9 is 2.0.0-p4. Ensure that you install the correct version.
3. Delete the previous version of Repository WAR file from the deployed location.
4. Build new repository WAR from the installation Zip file.

   Installation ZIP file for Connector for Documentum: `hip-cs-repository-1.9.0.zip` file.
5. Go to the `\repository\config\` folder in the WAR file.
6. Copy the folder containing the properties file to the HIP_HOME directory.
7. Configure the properties file in the HIP_HOME directory.
8. Deploy the `repository.war` file.
9. Verify the upgrade.

Chapter 7

Troubleshooting

This chapter describes the log settings, XDS Repository Connector for Documentum installation issues, and their resolutions.

Log Settings

A log is a chronological record of system activities that is sufficient to enable the reconstruction and examination of the sequence of environments and activities surrounding or leading to an operation, procedure, or event in a security-relevant transaction from inception to final results.

Log Description

The log file for XDS Repository Server is located in the `<hip_config_directory>\logs` folder.

Log Description for XDS Repository Connector for Documentum

For Apache Tomcat:

`C:\Users\<username>\.hip\logs\cs-repository.log`

For Oracle WebLogic:

`C:\Users\<username>\.hip\logs\cs-repository.log`

Log Description for XDS Repository Connector for xDB

The log file for the xDB XDS Repository Server is located in the `<hip_config_directory>\logs` folder.

For Apache Tomcat:

`C:\Users\<username>\.hip\logs\xdb-repository.log`

For Oracle WebLogic:

```
C:\Users\<username>\.hip\logs\xdb-repository.log
```

Log Management and Retrieval

Log Management and Retrieval for XDS Repository Connector for Documentum

XDS Repository Server uses the Simple Logging Facade for Java (SLF4J) combined with a `Log4j` logging provider implementation.

The default log level setting is INFO.

You can increase the trace messages by setting the log level to DEBUG in

`<tomcat installation directory>\webapps\repository\WEB-INF\classes\log4j.xml` file.

For example:

```
<--!Set to DEBUG to see detailed HIP message information -->
<logger name="com.emc.healthcare">
<level value="DEBUG"/>
</logger>
```

Log Management and Retrieval for XDS Repository Connector for xDB

XDS Repository Server uses the Simple Logging Facade for Java (SLF4J) combined with a `Log4j` logging provider implementation.

The default log level setting is INFO.

You can increase the trace messages by setting the log level to DEBUG in

`<tomcat installation directory>\webapps\repository\WEB-INF\classes\log4j.xml` file.

For example:

```
<--!Set to DEBUG to see detailed HIP message information -->
<logger name="com.emc.healthcare">
<level value="DEBUG"/>
</logger>
```

Issues and Resolutions

This section describes the XDS Repository Connector for Documentum installation issues and their resolutions.

ERROR Exception During ProvideDocumentSetProcessor Processing

Problem

The cs-repository webapp is successfully deployed but transaction fails with the following error message:

```
: 2014-05-16 16:13:26,906 INFO Object protocol version 2 2014-05-16
16:13:27,141 ERROR Exception during ProvideDocumentSetProcessor
processing com.documentum.fc.common.DfRuntimeException:  [DM
_SESSION_E_AUTH_FAIL]error:  "Authentication failed for user
Administrator with docbase Healthcare." at com.documentum.fc.common
.DfRuntimeException.convertToRuntimeException(DfRuntimeException.java:35)
at com.emc.healthcare.commons.cs.dfc.SimpleDfSessionManager.get
(SimpleDfSessionManager.java:83) at com.emc.healthcare.commons.cs
.dfc.SimpleDfSessionManager.get(SimpleDfSessionManager.java:33) at
com.emc.healthcare.commons.cs.dfc.ExchangeDfSessionManager.getNew
(ExchangeDfSessionManager.java:43)
```

Cause

The user name and password for authenticating the Content Server docbase is not correctly set.

Resolution

Check the following properties in the `cs-repository.properties` file:

```
# The user name for authenticating to the Content Server docbase.
# REQUIRED, NO DEFAULT
cs.userName=<user name>
# The password for authenticating to the Content Server docbase
# REQUIRED, NO DEFAULT
cs.password=<password>
```

Error Creating Bean with Name ’mllpSSLFilter’

Problem

You receive the following error message:

```
2014-05-16 14:58:30,143 ERROR Context initialization failed
org.apache.camel.RuntimeCamelException:  org.springframework.beans.factory
.BeanCreationException:  Error creating bean with name 'mllpSSLFilter'
```

```
defined in ServletContext resource [/WEB-INF/spring/context.xml]:
Cannot resolve reference to bean 'sslContext' while setting constructor
argument; nested exception is org.springframework.beans.factory
.BeanCreationException:  Error creating bean with name 'sslContextFactory'
defined in ServletContext resource [/WEB-INF/spring/context.xml]:
Cannot resolve reference to bean 'keyStore' while setting bean
property 'keyManagerFactoryKeyStore'; nested exception is org
.springframework.beans.factory.BeanCreationException:  Error creating
bean with name 'keyStoreFactory' defined in ServletContext resource
[/WEB-INF/spring/context.xml]:  Error setting property values; nested
exception is org.springframework.beans.PropertyBatchUpdateException;
nested PropertyAccessExceptions (1) are:  PropertyAccessException 1:
org.springframework.beans.MethodInvocationException:  Property 'dataFile'
threw exception; nested exception is java.lang.NullPointerException
at org.apache.camel.util.ObjectHelper.wrapRuntimeCamelException
(ObjectHelper.java:1344) at org.apache.camel.spring.SpringCamelContext
.onApplicationEvent(SpringCamelContext.java:120)
```

Cause

You are trying to use MLLP secure ports without specifying the HTTPS parameters.

Resolution

If you plan to use `hl7.inbound.mllp.securePort`, then you must specify the `https.server.privateKeyPassword` property.

You must also specify the following HTTPS properties in the `cs-repository.properties` file:

- https.keyStore
- https.keyStorePassword
- https.server.keyAlias
- https.server.privateKeyPassword
- https.trustStore
- https.trustStorePassword
- https.ciphersuites

For example:

```
https.keyStore=${com.emc.healthcare.home}/keystore.jks
https.keyStorePassword=changeit
https.trustStore=${com.emc.healthcare.home}/truststore.jks
https.trustStorePassword=changeit
https.ciphersuites=TLS_RSA_WITH_AES_128_CBC_SHA
https.server.privateKeyPassword=changeit
```

Context Initialization Failing when Deploying Server WAR Files

Issue with HIP Configuration

Problem

When you try to install the Repository WAR files, you receive the following error message:

```
o.s.web.context.ContextLoader - Context initialization failed
org.springframework.beans.factory.BeanInitializationException:
```

```
Could not load properties; nested exception is java.io
.FileNotFoundException:  C:\Users\Administrator\.hip\cs-repository\
cs-repository.properties (The system cannot find the path specified)at
org.springframework.beans.factory.config.PropertyResourceConfigurer
.postProcessBeanFactory (PropertyResourceConfigurer.java:87)
~[spring-beans-3.2.4.RELEASE.jar:3.2.4.RELEASE]
```

Cause

The `.hip` folder is not available in the `<user home>` folder.

Resolution

Ensure that the HIP configuration properties are available in the `%HIP HOME%` folder.

The *XDS Repository Connector for Documentum Administration Guide* provides detailed information on the HIP configurations required for successful installation of XDS Repository Connector for Documentum. Ensure that you complete the configurations before deploying the WAR files.

Issue with Camel Jar Files

Problem

When you try to install the Repository WAR files, you receive the following error message:

```
10:57:01.592 [localhost-startStop-1] INFO o.s.b.f.xml
.XmlBeanDefinitionReader - Loading XML bean definitions from
class path resource [META-INF/spring/xua-context.xml]10:57:01.895
[localhost-startStop-1] ERROR o.s.web.context.ContextLoader
- Context initialization failed org.springframework.beans
.factory.parsing.BeanDefinitionParsingException:  Configuration
problem:  Unable to locate Spring NamespaceHandler for XML schema
```

```
namespace [http://camel.apache.org/schema/spring] Offending
resource:  ServletContext resource [/WEB-INF/spring/context.xml] at
org.springframework.beans.factory.parsing.FailFastProblemReporter.error
(FailFastProblemReporter.java:68) ~[spring-beans-3.2.4.RELEASE.jar:3.2.4
.RELEASE]
```

Cause

Camel JAR files are not added to the Tomcat classpath.

Resolution

Install Camel Library Dependencies.

Installing the Dependent Libraries, page 13 describes the steps to install Camel dependent libraries.

Error while Installing HAPI Jar Files

Problem

Installation fails.

Cause

- You downloaded an incorrect version of HAPI.
- You copied the HAPI JAR files directly to the `cs-repository/WEB-INF/lib` folder.

Resolution

- Download the correct version of HAPI files and refer the files from a separate folder defined in the web applications server's context classpath.
- Check in Tomcat 7.0, the HAPI folder is referred from the `conf/context.xml` file.
- If you are using WebLogic, verify if the `HAPI 2.0` JAR files are successfully installed.

  Installing the Dependent Libraries, page 13 describes the steps for installing the JAR files.
- Check the log files in the logs folder (`~\Oracle\Middleware\user_projects\domains\domain{domain used for deployment}\servers\AdminServer\logs`) for errors.

Cannot Connect to the XDS Repository Server

Problem

You are unable to connect to the XDS Repository Server.

Cause

You are using an incorrect URL or the Repository server installation is incomplete.

Resolution

- Ensure that the endpoint (URL/port) used to connect to the XDS Repository Server is correct.

  You can also validate the URL by accessing the XDS Repository Server WSDL.

  `http://localhost:<port>/cs-repository/services`.
- Ensure that the server is up and running.

  Chapter 5, Verifying the Installation provides the steps for verification.

  Access the XDS Repository Server WSDL page at `http://localhost:<port>/cs-repository/services/`.

  If the WSDL loads then the XDS Repository Server is up.
- Check the `cs-repository.log` file for startup errors.

Java Errors at Startup

Problem

You receive the following error message during the startup:

```
java.lang.NoClassDefFoundError:  Lca/uhn/hl7v2/parser/Parser; at
java.lang.Class.getDeclaredFields0(Native Method) at java.lang
.Class.privateGetDeclaredFields(Unknown Source) at java.lang.Class
.getDeclaredFields(Unknown Source) at org.codehaus.groovy.vmplugin.v5
.Java5.configureClassNode(Java5.java:313)
```

Cause

The server cannot find the HAPI JAR files because they were not deployed or were deployed incorrectly.

Resolution

Install the HAPI JAR files.

Installing the Dependent Libraries, page 13 provides the steps.

DFC Error at Startup

Problem

You receive the following error:

```
2013-06-20 10:08:02,270 ERROR DfException::THREAD: Thread-8; MSG:
[DFC_API_W_ATTEMPT_TO_USE_DEPRECATED_CRYPTO_API] WARNING: Program attempts
to use deprecated non-FIPS compliant cryptography API. This is not
recommended.  Please consult the documentation for more detail.
```

Cause

NA

Resolution

You can ignore this message as this does not have impact on the product function.

Cannot Connect to the Documentum Repository

Problem

XDS Repository Server is unable to connect to the Documentum repository and you receive the following error message in the log file:

```
2011-04-27 21:17:47,134 WARN [DFC_DOCBROKER_EXCLUDED] Docbroker
"Healthcare/2001:0:4137:9e76:306f:1b44:3f57:cd9a:1489" excluded from
```

```
active docbroker list due to "3" connection failed attempts with exception
"Connection refused:  connect" java.net.ConnectException:  Connection
refused:  connect at java.net.PlainSocketImpl.socketConnect(Native Method)
at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:333) at
java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:195)
```

Cause

XDS Repository Server is unable to connect to the Documentum repository because:

- The Documentum repository is not currently running
- The Content Server database and log in information are incorrect

Resolution

- Ensure that the Documentum repository is running.
- Ensure that the `cs-repository.properties` file uses the correct database and log in information. The user names are case-sensitive.

Registering a Document More Than Once

Problem

If you submit a duplicate Provide and Register Document Set transaction after an initial successful transaction, the duplicate transaction fails.

Cause

The document already exists.

Resolution

You can re-register or replace the existing document, but you cannot replace or use version control on a document in the repository using the same unique ID.

XUA Policy File Error

Problem

You receive the following error message in the log file during initialization:

```
Context initialization failed org.apache.camel.RuntimeCamelException:
org.apache.cxf.ws.policy.PolicyException:  Policy reference
classpath:ws-policy.xml could not be resolved.
```

Cause

XDS Repository Server cannot find the `ws-policy.xml` file defined in the classpath.

Resolution

Ensure that you copy the sample `ws-policy.xml` file from

`/webapps/cs-repository/config/cs-repository/`

and place it in

`/webapps/cs-repository/WEB-INF/classes/`

Alternatively, you can place this file in a different folder and define the file location in the server classpath.

servicestore.jks File Not Found Error

Problem

You receive an error message as follows in the log file:

```
java.io.FileNotFoundException:  certificates\servicestore.jks (The system
cannot find the path specified)
```

Cause

XDS Repository Server is unable to find the keystore file specified in the `serviceKeystore.properties` file.

If you are using HTTPS to connect to the XDS Repository, the appropriate TLS certificates (keystore.jks, truststore.jks) must be located in the HIP_HOME directory.

Resolution

- Copy the keystore file to the location specified in `serviceKeystore.properties`.
- Verify if `tomcat\conf\server.xml` is configured for SSL as follows:

  ```
  <Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
  SSLEnabled="true"
  maxThreads="150" scheme="https" secure="true"
  clientAuth="false" sslProtocol="TLS"
  keystoreFile="C:/Users/Administrator/.hip/keystore.jks" keystorePass="changeit"
  truststoreFile="C:/Users/Administrator/.hip/truststore.jks" truststorePass="changeit"/>
  ```

Required Header Not Present Error

Problem

XDS Repository Server writes the following error to the log file:

```
org.apache.cxf.interceptor.Fault:  A required header representing a
Message Addressing Property is not present
```

Cause

An XUA-enabled XDS Repository Server received a request without a security header.

Resolution

With XUA enabled on the server, all requests must contain a security header. Either disable XUA on the server or instruct the sending application to send requests with security headers.

To disable XUA, open the `cs-repository.properties` file and set the `repository.xua.enabled` property to **`false`**.

Failure to Authenticate Security Token

Problem

Authentication of Security Token fails.

Cause

Incorrect configuration of XUA-related properties in the `cs-repository.properties` file.

Resolution

Ensure that the XUA-related properties in `cs-repository.properties` file are configured correctly.

Verify if the security options are enabled and configured correctly.

Check the following properties:

- role
- purpose of use
- authentication methods
- token issuer names
- audience restriction
- certificate configurations

java.lang.OutOfMemoryError: PermGen Space Error

Problem

You receive a `java.lang.OutOfMemoryError:PermGen space` error while verifying the deployment of HIP Repository.

Cause

The permanent generation heap is full.

Resolution

Increase the Permgen space.

For Tomcat:

```
Set JAVA_OPTS=-Xms256m -Xmx512m -XX:PermSize=256m -XX:MaxPermSize=512m
```

For WebLogic:

Replace the following lines in the `setDomainEnv.cmd` files located at `C:\Oracle\Middleware\user_projects\domains\base_domain\bin`

Replace:

```
set WLS_MEM_ARGS_64BIT="-Xms256m -Xmx512m"
set WLS_MEM_ARGS_32BIT="-Xms256m -Xmx512m"
```

With:

```
set WLS_MEM_ARGS_64BIT=-Xms256m -Xmx512m -XX:MaxPermSize=512m
set WLS_MEM_ARGS_32BIT=-Xms256m -Xmx512m -XX:MaxPermSize=512m
```

Errors Related to ADT Merge Patient Identities Requests

Problem

Retiring patient records are not moving to the Surviving Patients folder.

Cause

- Java Method Server is not running
- HipPatientMergeJob is inactive
- ADT Merge Identities related configuration is incorrect

Resolution

Verify if:

- The Documentum Java Method Server is running
- The HipPatientMergeJob is installed and active
- The `patientsCabinetPath` argument is set to valid Patients Records cabinet in HipPatientMergeJob custom arguments

The *XDS Repository Connector for Documentum Administration Guide* provides details on the configuration of Merge Job and Method Arguments. Ensure that the configurations are correct before deploying the WAR files.

Errors Related to HL7 MDM Message Processing

Problem

MDM T01 messages with the content object ID as unique document identifier are not sent.

Cause

Incorrect configuration of HL7 Outbound properties.

Resolution

Verify if:

- The `hl7.properties` file is located in the `<hip_home>/cs-repository` location.
- The `hl7.outbound.mllp.host` property is set to the appropriate HL7 External Engine.
- The `hl7.outbound.mllp.port` property is set to the correct port.

The *XDS Repository Connector for Documentum Administration Guide* provides detailed information about the configuration of HL7 Outbound properties.

Index