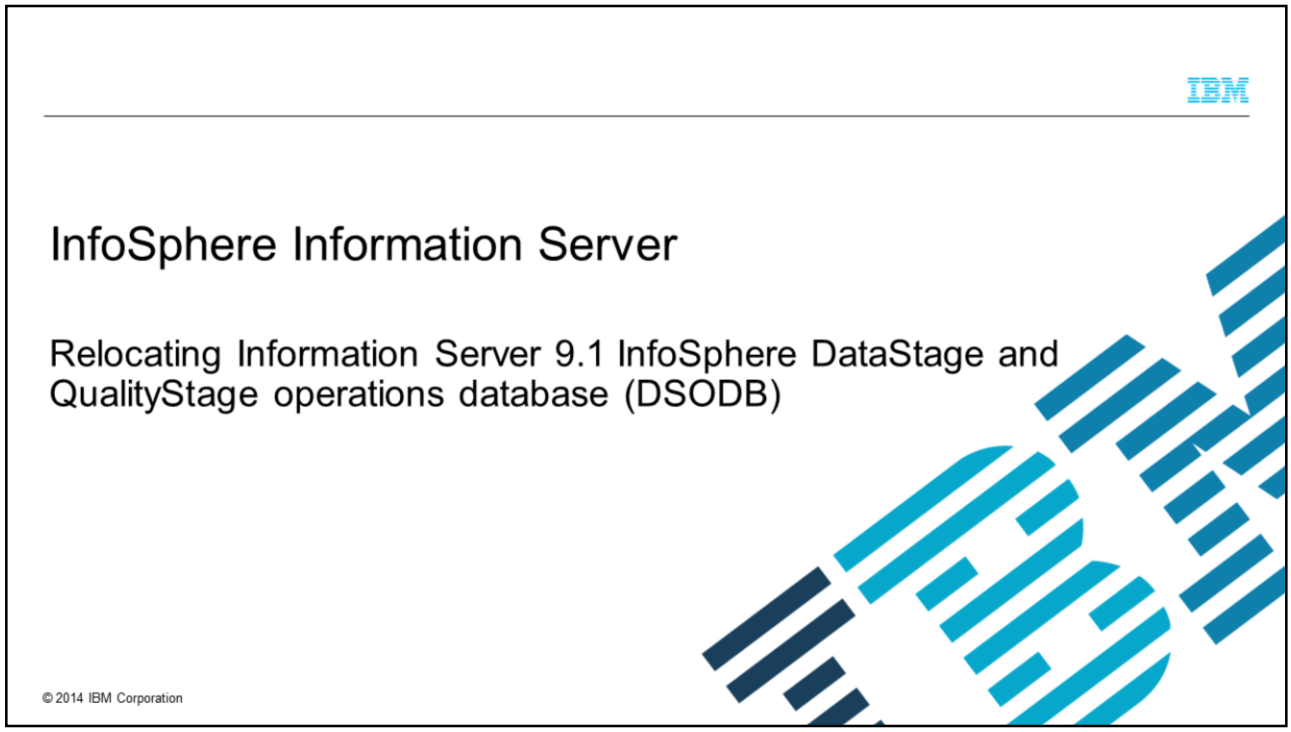


This presentation discusses how to relocate the InfoSphere® DataStage® and QualityStage operations database to another server. It assumes that you are still using the same services tier to manage repository registration in the same metadata repository. This presentation is valid for Information Server version 9.1

IBM

Objectives

- List repositories and databases
- Create repository properties file
- Reregister repository
- Update engine tier

The objectives of this presentation are to show how to list the repositories and databases, show how to create a new properties file for the repository, how to reregister the repository, update the engine tier, change the DSODB password if necessary, and how to test the new connection.

List repositories

- Log in to the Services tier computer
- List Repositories/Databases
  - UNIX® or Linux®
    cd <is_installPath>/ASBServer/bin
    ./RepositoryAdmin.sh -listRepositories
  - Windows®
    cd <is_installPath>\ASBServer\bin
    .\RepositoryAdmin –listRepositories
- Example:
  $ ./RepositoryAdmin.sh –listRepositories
  dsodb
  QSSRDDB

The first step is to login to the Information Server services tier computer. List out the databases and repositories using the RepositoryAdmin command to get the exact name of your operations database. The default name of the operations database is DSODB. Note the case of the database name as the commands in these slides are case-sensitive.

Edit repository properties file (1 of 3)

- Create repository properties file

```
./RepositoryAdmin.sh -displayRepository -rn dsodb -res dsodb.properties
DatabasePlatform.databaseType=DB2
DatabasePlatform.version=10.1
DatabaseServer.host=ipsvm00079.swg.usma.ibm.com
DatabaseServer.port=50000
Database.name=dsodb
Database.alias=null
Database.location=/opt/IS91/IBM/InformationServer/Repos/dsodb
Repository.name=dsodb
Repository.description=DSODB
Repository.tool=DataStage
Repository.context=
Repository.schema=dsodb
RepositoryConnection.name=dsodb
RepositoryConnection.userName=dsodb
RepositoryConnection.password={iisenc}N2RHakj6gLz7fCJ2yknhIg==
RepositoryConnection.connectionURL=jdbc:db2://ipsvm00079.swg.usma.ibm.com:50000/dsodb
RepositoryConnection.managedDataSourceName=Tablespace.name=DSODBSPACE
```

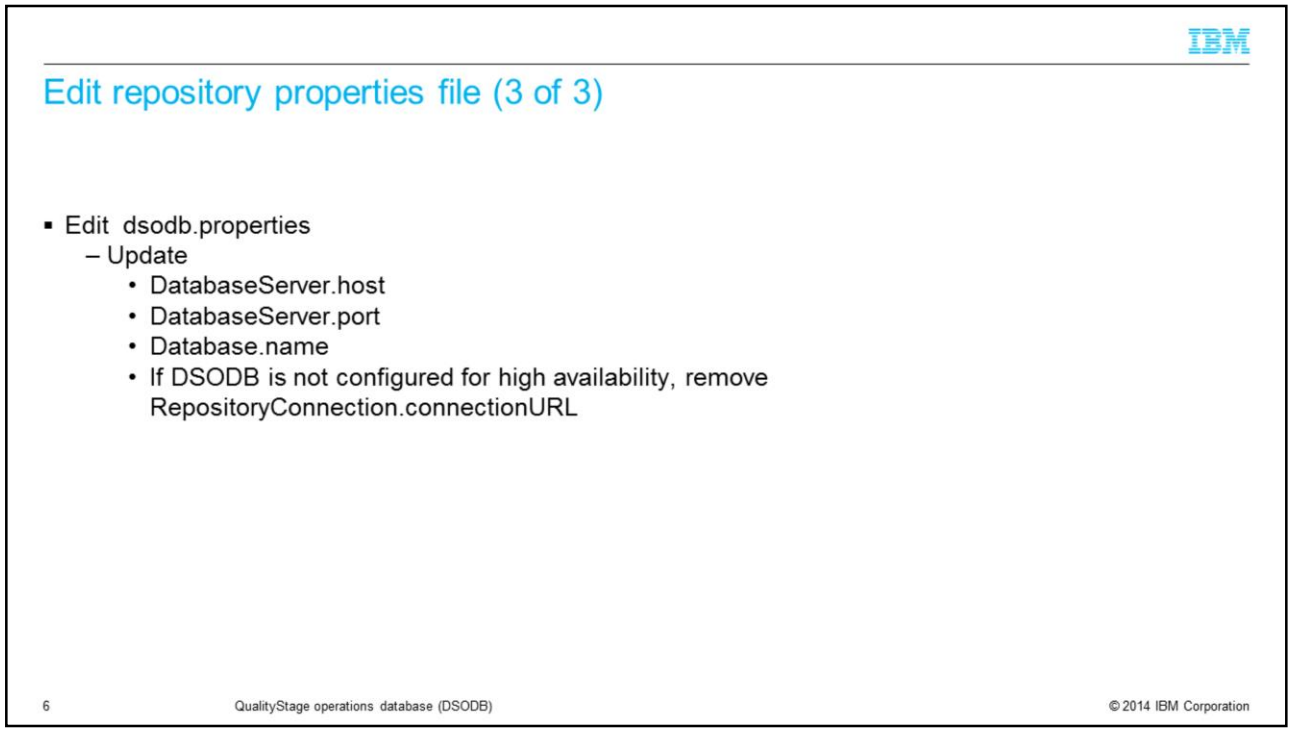

The next step is to create the repository properties file by using the RepositoryAdmin command. Use the repository name from the output of the last step. If you are unsure you have the correct database name for the operations database, look at the Repository.tool value in the output file that was created when you ran the RepositoryAdmin command. For the operations database, the value should be set to DataStage. If it is not, you are looking at the wrong database. Display the database properties values for the other databases that are listed in the previous step to find the correct database name for the operations database.

Edit repository properties file (2 of 3)

- Database.name is xmeta
  - Xmeta **must** be relocated first
  - If you already relocated xmeta, continue to next slide
- Database.name is not xmeta
  - Backup and restore the operations database
  - Database credentials and properties remain the same

If the Database.name property is set to xmeta, you must relocate the xmeta repository before you complete the steps in this module. If you already relocated xmeta, continue to the next slide. If not, follow the instructions in the IBM Education Assistant module on relocating the xmeta repository.

If the operations database is not in the xmeta database, backup and restore the operations database onto the new server by using the database backup and restore utilities. This presentation assumes that the database properties and the database credentials remain the same on the new database server.



Next, edit the dsodb.properties file that was created with the RepositoryAdmin command. Update the new values for host, port, and database name. If the DSODB database is not configured for high availability, you can remove the RepositoryConnection.connectionURL property from the file.

Unregister the repository

- Unregister
  RepositoryAdmin.bat -unregisterRepository -rn dsodb
- Use value set in Repository.name attribute
  - Value is case-sensitive

The next step is to unregister the repository. Because the new repository has the same name as the existing registered repository, you must first unregister the existing repository. Repository names must be unique. Be sure to use the value that is displayed in the Repository.name property that is obtained from the RepositoryAdmin command. This value is case-sensitive so be sure that it matches the RepositoryAdmin command output.

Register the repository

- Register the new server, database, and repository
  - Use the edited dsodb.properties file
  - UNIX/Linux
    RepositoryAdmin.sh -registerRepository -pf dsodb.properties
  - Windows
    RepositoryAdmin.bat -registerRepository -pf dsodb.properties

Next, register the new server, database, and repository by using the edited properties file created in the previous steps. For this step, you need to register the repository. When you register a repository with the RepositoryAdmin tool, if the server and database are not yet registered, they are registered during the same operation.

Update engine tier

- Update engine tier configuration file
  - Must be in the repository directory
  - User must have write permissions on DSODB
  - UNIX/Linux

    cd <IS_HOME>/Server/DSODB

    ../../ASBNode/bin/RegistrationCommand.sh -get_config -user isadmin -password pswd -rp dsodb -cf DSODBConnect.tmpl -res DSODBConnect.cfg
  - Windows

    cd <IS_HOME>\Server\DSODB

    ..\..\ASBNode\bin\RegistrationCommand.bat -get_config -user isadmin -password pswd -rp dsodb -cf DSODBConnect.tmpl -res DSODBConnect.cfg

The next step is to update the connection configuration file on each engine tier with the connection information for the new repository. Run the RegistrationCommand tool from the directory that corresponds to the repository, in this case, DSODB. Be sure that the user who is running the command has write access to the DSODB directory. The user that is specified in the RegistrationCommand –user argument must be an Information Server suite administrative user.

Change DSODB user password - Optional (1 of 3)

- Stop the AppWatcher process/service on Engine tier
  - UNIX or Linux
    cd <InformationServer_Home>/Server/DSODB/bin
    ./DSAppWatcher.sh –stop
  - Windows
    net stop "DataStage AppWatcher Service“
- Optional – Encrypt new password
  - Password that is saved in cfg file in clear text by default
  - Run encrypt command by using the full path to executable
    <InformationServer_Home>/ASBNode/bin/encrypt.sh
    Enter the text to encrypt:
    Enter the text again to confirm:
    {iisenc}PvqKLr7z3QOLJCQ4QhbrrA==

If there is a need to change the connection password for the DSODB database, follow the steps on the next three slides. If the password is the same, skip to the Testing the database connection slide.

To change the connection password for the DSODB database, the first step is to login to the engine tier and stop the AppWatcher process. On Windows, stop the AppWatcher service. For UNIX and Linux, run the DSAppWatcher –stop command from the DSODB/bin directory. After the AppWatcher is stopped, the next step is to set the new password. By default, the password is saved in the configuration file in plain text. If that is not acceptable, use the encrypt command that is displayed on this slide to first encrypt the password. Run the command with no parameters and it prompts for the text to encrypt. This text is not displayed on the screen. Confirm the text. The command then displays the encrypted value that can be used in the command to follow.

Change DSODB user password - Optional (2 of 3)

- Log in to Services Tier
  - UNIX, Linux, or Windows
    cd <InformationServer_Home>/ASBServer/bin
    ./RepositoryAdmin.<sh/bat> -updateRepositoryConnection -rn dsodb -cn dsodb -cw newvalue
  - -rn = Repository.name
    -cn = RepositoryConnection.name
    -cw = New password
  - Example with encrypted password:
    ./RepositoryAdmin.<sh/bat> -updateRepositoryConnection -rn dsodb -cn dsodb –cw
    "{iisenc}PvqKLr7z3QOLJCQ4QhbrrA=="

Next, login to the services tier and run the RepositoryAdmin command that is displayed on this slide to update the repository connection information with the new connection password. Use the repository output information that is obtained on slide 4 to get the proper values for this command. The syntax of this command is the same for Windows, other than the command has a .bat extension. If you are using an encrypted password, put the results of the encryption program in for the –cw option. See the example displayed on this slide.

Change DSODB user password – Optional (3 of 3)

- Log in to Engine tier
- Ensure that user has permissions to write to <InformationServer_Home>/Server/DSODB
- Generate a new repository connection file
  cd <InformationServer_Home>/Server/DSODB
  ../../ASBNode/bin/RegistrationCommand.<sh/bat> -user <Suite_admin> -password pswd -gcf -rp dsodb -cf DSODBConnect.tmpl -res DSODBConnect.cfg
- -rp = Repository.name

Finally, login to the engine tier and change directories to the InformationServer/Server/DSODB directory. Run the RegistrationCommand script by using the syntax that is displayed on this slide. Be sure that the user who is running the script has permissions to write to the DSODB directory. The value for the –rp argument is the value that is obtained for Repository.name from slide 4.

Test connection

- Connect to the services tier
  - UNIX/Linux
    cd <IS_HOME>/ASBServer/bin
    ./RepositoryAdmin.sh -testRepositoryConnection -rn dsodb -cn dsodb
  - Windows
    cd <IS_HOME>\ASBServer\bin
    .\RepositoryAdmin.bat -testRepositoryConnection -rn dsodb -cn dsodb
  - "-rn" = Repository.Name
  - "-cn" = RepositoryConnection.Name
  - Example:
    $ ./RepositoryAdmin.sh -testRepositoryConnection -rn dsodb -cn dsodb

    Connection to repository dsodb was successful.

The last step is to test the connection to DSODB by using the RepositoryAdmin command that is in the ASBServer/bin directory of the Information Server installation directory. The –rn argument is the value of Repository.Name and –cn is the value of RepositoryConnection.Name from the output from the RepositoryAdmin –displayRepositories command on slide 4. The command on this slide returns a message that the connection was successful. If it does not, verify the values for –rn and –cn are correct and in the same case as the output obtained from slide 4. If it is correct, go back and verify the changes that are made and test again.

Trademarks, disclaimer, and copyright information