Third-Party Software Support

Converting from SAS® Table Server to a PostgreSQL Database

Table of Contents

This document provides instructions for updating an existing deployment of the third maintenance release for SAS® 9.2 (TS2M3) that is configured to use SAS® Table Server. These instructions explain how to migrate from SAS Table Server to a PostgreSQL database for use with the SAS® Web Infrastructure Platform, SAS® Shared Services, and the SAS® Content Server.

Prerequisite Steps

Before you attempt the migration steps that are provided in this document, complete the following tasks:

1. Download the SAS WIP Database Importer (ftp://ftp.sas.com/techsup/download/blind/DbImporter92m3.zip).
2. Install a Java 6 Java Runtime Environment (JRE).
3. Add the JAVA_HOME environment variable to your system environment properties and set it to the value of the installation home directory of the Java 6 JRE.
4. Download Apache Ant 1.6.3 or later. In your command window or in your system's environment properties, set the ANT_HOME environment variable to the value of its installation home directory.

   Note: Apache Ant 1.8.4 is included in the download for the SAS WIP Database Importer that is mentioned above in Step 1.
5. Download the PostgreSQL Java Database Connectivity (JDBC) driver Java Archive (JAR) file and add it to the **`jars/PostgreSQL/`** directory that is defined in the build.properties file within the download for the SAS WIP Database Importer.

Database Migration Instructions for the WebSphere Application Server

Step 1: Extract the Web Infrastructure Platform and Shared Services data from SAS Table Server.

Extract the data that is related to the Web Infrastructure Platform and Shared Services from SAS Table Server. To do this, use the SAS® Migration Utility and package the data in a ZIP file. You use this file later to load the data into the new database.

1. Download the SAS Migration Utility (support.sas.com/demosdownloads/setupcat.jsp?cat=SAS+Migration+Utility) and save it to a folder on your file system, such as **`C:\SMU`**.
2. Copy the contents of the smu.properties file that is located in the Appendix of this document and save it to a file on your system.
3. In the smu.properties file, update the values for the host name, port number, file path, user name, and password to match your environment.
4. Create the folder **`C:\SMU_packages`** on your file system. This is the location where the migration package will be created. The folder value should match the value for the **SMU.Output.Dir** property from the smu.properties file.

5. From the command prompt, change to the directory where the downloaded SAS Migration Utility is stored.

6. Submit the following command to create the migration package:

   ```
   smu92_32 -properties "path-to-your-smu.properties-file" -only biservmid
   ```

   Here is an example:

   ```
   smu92_32 -properties "C:\smu.properties" -only biservmid
   ```

7. Go to the **`C:\SMU_packages\`*`your-environment`*`\biservmid`** directory and locate the WIP_database.zip file, which contains the data from SAS Table Server.

Step 2. Create the new target database.

Install and configure the new target database if you do not already have one running. If needed, consult the database administrator at your site for assistance.

Step 3. Populate the database with the tables that are needed for the Web Infrastructure Platform and Shared Services.

Create the Web Infrastructure Platform and Shared Services tables in PostgreSQL.

1. Open the database_postgres.properties file from the SAS WIP Database Importer that you previously downloaded. Update the values for the following properties to match the values for your environment:

   - **database.host**
   - **database.port**
   - **database.name**
   - **database.user.id**
   - **database.user.password**
   - **database.schema.pattern**

   Note: The PostgreSQL database schema is the uppercased value of the log on ID (for example, ${*DATABASE.USER.ID*}).

2. Open the build.properties file from the SAS WIP Database Importer. Remove the comment delimiters from the following line:

   ```
   ##database.type=postgres
   ```

3. Update the values for the following properties to match the values for your environment:

   a. **database.type**—specifies the type of the target database (for example, `database.type=postgres`).

   b. **sql.scripts.data.dir**—specifies the absolute path to the data directory in the SASHOME location, which contains the SQL scripts for the Web Infrastructure Platform that are used to create its database tables, load

its out-of-the-box data, and drop its database tables (for example, **`sql.scripts.data.dir=C:\\Program Files\\SAS\\SASSharedServices\\9.2\\Config\\Deployment\\Data\\`**).

c. **database.zip.file**—specifies the absolute path to the v920m3 WIP_database.zip file from which records will be read and inserted into the target v920m3 database. The WIP_database.zip file is created by the SASSMU2 (for example, **`database.zip.file=C:\\Public\\v920m3\DbImporter92m3\\test\\SASSMU2\\WIP_database.zip`**).

d. **log4j.config.file**—specifies the Log4J configuration file which is used by the database importer for the Web Infrastructure Platform. Edit the log4j.properties file to specify the location of the log file (for example, **`log4j.config.file=file:///C:/Public/v920m3/DbImporter92m3/log4j.properties`**).

e. **java.maxmemory**—specifies the maximum amount of memory that can be used by the database importer for the Web Infrastructure Platform (Java application) (for example, `java.maxmemory=2048m`).

4. Open a command window and change to the SAS WIP Database Importer directory. Submit the **`createDatabaseTables`** command to create the database tables for the Web Infrastructure Platform and Shared Services (for example, `C:\DbImporter92m3> ant -f build.xml createDatabaseTables`).

Step 4. Update the Web Infrastructure Platform and Shared Services data source that is defined in WebSphere.

The initial installation created a data source named SharedServices that references SAS Table Server. This step redirects that reference to a new data source that relies on PostgreSQL.

Before you create the JDBC provider (named SharedServices JDBC Provider) and the JDBC data source (named SharedServices), enter the value **`PostgreSQLJAASAlias`** in the **`JAAS - J2C Authentication Data Alias`** text box. This will contain the user ID and password that are used to authenticate the connection to the SharedServices data source. This alias will be referenced when you are creating the new JDBC data source SharedServices.

1. Log on to WebSphere (for example, http://*server-name*:9060/ibm/console/login.do).

2. Select **Security ► Secure administration, applications, and infrastructure ► Java Authentication and Authorization Service ► J2C authentication data link**.

3. Enter the value **`PostgreSQLJAASAlias`** in the **`JAAS - J2C Authentication Data Alias`** text box.

4. On the JAAS - J2C Authentication Data New page, enter the following information for these required fields:

   - **`Alias`**: **`PostgreSQLJAASAlias`**
   - **`User ID`**: ***`user-ID-for-connection-to-the-database`***
   - **`Password`**: ***`password-for-connection-to-the-database`***

   Note: WebSphere stores this password using its own encoding mechanism.

You must remove the existing SharedServices data source definition that is based on SAS Table Server information using the WebSphere Administration Console:

1. In the left pane, select **Resources ► JDBC ► Data sources**.
2. Select **SharedServices** and click **Delete**.
3. Click **Save** to store the change to the master configuration.

Next, create a new SharedServices data source definition that is based on PostgreSQL.

1. Select the scope as follows:

   Node: *node-name*
   Server: SASServer1

2. Click **New**.
3. From the Create a New Data Source page, specify the following information:

   a. Step 1: Enter basic datasource information

      Datasource name: SharedServices
      JNDI name: sas/jdbc/SharedServices

      Click **Next**.

   b. Step 2: Select JDBC provider

      Select **Create a new JDBC provider**.

      Click **Next**.

   c. Step 2.1: Create new JDBC provider

      Database type: *user-defined*
      Implementation class name= org.postgresql.ds.PGConnectionPoolDataSource
      Name=SharedServices PostgreSQL JDBC Provider
      Description=PostgreSQL JDBC Driver for SharedServices

      Click **Next**.

   d. Step 2.2: Enter database class path information

      Specify the directory location of the PostgreSQL JDBC JAR file.

      Click **Next**.

   e. Step 3: Enter database-specific properties

      Data store helper class name= com.ibm.websphere.rsadapter.
      GenericDataStoreHelper

      Deselect **Use this data source in container managed persistence (CMP)**.

      Click **Next**.

f. Step 4: Set up security aliases

Component-managed authentication alias: *cell-name*/PostgreSQLJAASAlias
Mapping-configuration alias: (none)
Container-managed authentication alias: *cell-name*/PostgreSQLJAASAlias

Click **Next**.

g. Step 5: Summary

Review the following items:

Scope	cells:*cell-name*:nodes:*node-name*:servers:SASServer1
Data source name	SharedServices
JNDI name	sas/jdbc/SharedServices
JDBC provider name	PostgreSQL SharedServices JDBC Driver
Description	PostgreSQL JDBC Driver for SharedServices
Class path	${PostgreSQL_JDBC_DRIVER_PATH}/*jar-name*.jar
${PostgreSQL_JDBC_DRIVER_PATH}	*your-path-here/jar-name*.jar
Implementation class name	org.postgresql.ds.PGConnectionPoolDataSource
Data store helper class name	com.ibm.websphere.rsadapter.GenericDataStoreHelper
Use this data source in container managed persistence (CMP)	false
Component-managed authentication alias	*cell-name*/PostgreSQLJAASAlias
Mapping-configuration alias	(none)
Container-managed authentication alias	*cell-name*/PostgreSQLJAASAlias

4. Click **Finish** to complete the new definition.

Be sure to save the changes to the master configuration.

Step 5. Update the configuration properties for the Web Infrastructure Platform and Shared Services.

The SAS installation and configuration process stores metadata about the properties for Shared Services. Updating the metadata is a manual change to the configuration. Therefore, the properties must be updated to accommodate possible future migrations. Here are the steps to update the metadata for the configuration properties:

1. Start SAS® Management Console and log on as **`sasadm`** (or another fully privileged user ID).
2. Click the **Folders** tab and select **System ► Applications ► SAS Shared Services**.
3. Select the **`SharedServices9.2`** folder.
4. In the right pane, right-click **SharedServices9.2** and select **Properties**.
5. Click the **Configuration** tab.
6. Specify the following values for the SharedServices properties:
   a. **data.dbms.type**: **`postgres`**
   b. **dbms.biservmid.host: *`PostgreSQL-host-name`***
   c. **dbms.biservmid.jdbc.dir: *`directory-path-containing-the-PostgreSQL-JDBC-driver-JAR`***

d. **dbms.biservmid.name:** ***PostgreSQL-database-name***

e. **dbms.biservmid.port:** ***PostgreSQL-port-number***

f. **dbms.biservmid.userid:** ***PostgreSQL-userID***

g. **dbms.biservmid.validation.query:** **`select 1 from dual`**

7. Click **OK** to exit the dialog box, and then exit SAS Management Console.

Step 6. Migrate the Web Infrastructure Platform and Shared Services from SAS Table Server to the new database.

Migrate the data that was exported from SAS Table Server to the new database.

1. Open a command window and navigate to the directory for the SAS WIP Database Importer.
2. Submit the **`importDatabase`** command to create the Web Infrastructure Platform and Shared Services database tables (for example, `C:\DbImporter92m3> ant -f build.xml importDatabase`).

Step 7. Update the workflow to use the proper dialect.

Update the workflow in order to use the new database.

1. Open the sas.shared9.2.ear file. Within this EAR file, open the workflow.properties file, which resides in the **`\sas.workflow.war\WEB-INF\`** directory.
2. Add a comment delimiter to the following line:

```
workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
TSFirebirdDialect
```

   Here is an example:

```
#workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
TSFirebirdDialect
```

3. Remove the comment delimiter from the following line:

```
#workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

   Here is an example:

```
workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

4. Redeploy the sas.shared9.2.ear file.

Step 8. Populate the database with the tables that are needed for the SAS Content Server.

The scripts that were needed to create the tables for the SAS Content Server were executed by the **createDatabaseTables** command in the previous steps.

Step 9. Configure the SAS Content Server to use the new database as its repository.

This section discusses the steps that are needed to migrate the stored SAS Content Server data from SAS Table Server to a PostgreSQL database. It also details the steps that are needed to configure the SAS Content Server for the new PostgreSQL database. In order to migrate the data, you need to convert the SAS Content Server from using the SAS Table Server to using the file system and then to using PostgreSQL.

Revert to the file system after you have configured the SAS Content Server for SAS Table Server.

1. Stop WebSphere.
2. Back up the following directory:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer**
3. Create a new directory as follows:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp**
4. Copy the repository.xml file that resides in the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates/** directory to the ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp** directory.
5. Change directory to ***SAS-configuration-directory*/Lev1/Web/Utilities**. Verify that your JCRCopyRepository.sh script has only one set of Java Naming and Directory Interface (JNDI) arguments for SAS Table Server. It should look similar to the following:

```
#!/bin/sh -p
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
picklist
DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/
Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/
config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/
```

```
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.
nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc:sastkts://your-host-name:
 2172?constring=\(DSN=SharedServices\) \
-Dscs.jndi.driver=com.sas.tkts.TKTSDriver \
-Dscs.jndi.user=sastrust@saspw \
-Dscs.jndi.pwd=your-password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

6. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/Lev1/AppData
/SASContentServer/Repository /SAS-configuration-directory/Lev1/Lev1
/AppData/SASContentServer/temp
```

You are now using the file system as the temporary back end for the SAS Content Server. Next, configure the SAS Content Server to use PostgreSQL as the back-end database.

7. Change the name of the SAS Content Server repository from **Repository** to **RepositoryTS**.

   - For a Windows operating environment, use the following command:

     ```
     move C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
     C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\RepositoryTS
     ```

   - For UNIX and z/OS operating environments, use the following command:

     ```
     mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
     SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryTS
     ```

8. Create a new directory called **Repository** in the same location.

   - Under Windows, use the following command:

     ```
     mkdir C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
     ```

   - Under UNIX and z/OS, use the following command:

     ```
     mkdir SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
     ```

9. Copy the repository.postgres.xml file from the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates** directory to the */SAS-configuration-directory*/Lev1/AppData/SASContentServer/Repository** directory that was created in the

preceding step. Rename the file to repository.xml. The following example shows syntax for a UNIX environment:

```
cp /SASHOME/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-
INF/templates/repository.postgres.xml Repository/repository.xml
```

10. Obtain the values for the database name, host name, port number, user ID, and password from the web application server.

    On the IBM WebSphere application server, the values are available from the WebSphere Administration Console. In the console, select **Resources ► JDBC ► Data Sources ► Custom Properties**.

11. Change directory to **/*SAS-configuration-directory*/Lev1/Web/Utilities**. Ensure that the PostgreSQL driver is included in the DRIVER list in your JCRCopyRepository.sh script. Then update the JNDI values to the values that are needed to connect to the PostgreSQL database. Your script should look similar the following:

```
#!/bin/sh -p
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
picklist
DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/SASShared
Services9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/Web/
Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/config/
Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/sas/
config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar:/your-directory/your-
driver-name.jar
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc:postgresql://host:5432/SharedServices\
-Dscs.jndi.driver=org.postgresql.Driver \
-Dscs.jndi.user=your-id  \
-Dscs.jndi.pwd=your-password  \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

12. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/AppData
/SASContentServer/temp /SAS-configuration-directory/Lev1/AppData
/SASContentServer/Repository
```

13. Restart WebSphere.

Step 10. Verify that the new database is the new back-end data store.

To verify that the SAS Content Server is using PostgreSQL, save a new report in SAS® Web Report Studio or add content from SAS Management Console to the SAS Content Server. The size of your tables will increase if the SAS Content Server is properly configured to use PostgreSQL.

To verify that the Web Infrastructure Platform and Shared Services are using PostgreSQL, save a new alert in the SAS® BI Dashboard. The size of your tables will increase if the Web Infrastructure Platform and Shared Services are properly configured to use PostgreSQL.

Database Migration Instructions for the WebLogic Application Server

Step 1: Extract the Web Infrastructure Platform and Shared Services data from SAS Table Server.

Extract the data that is related to the Web Infrastructure Platform and Shared Services from SAS Table Server. To do this, use the SAS Migration Utility and package the data in a ZIP file. You use this file later to load the data into the new database.

1. Download the SAS Migration Utility (support.sas.com/demosdownloads/setupcat.jsp?cat=SAS+Migration+Utility) and save it to a folder on your file system, such as **C:\SMU**.

2. Copy the contents of the smu.properties file that is located in the Appendix of this document and save it to a file on your system.

3. In the smu.properties file, update the values for the host name, port number, file path, user name, and password to match your environment.

4. Create the folder **C:\SMU_packages** on your file system. This is the location where the migration package will be created. The folder value should match the value for the **SMU.Output.Dir** property in the smu.properties file.

5. From the command prompt, change to the directory where the downloaded SAS Migration Utility is stored.

6. Submit the following command to create the migration package:

```
smu92_32 -properties "path-to-your-smu.properties-file" -only biservmid
```

   Here is an example:

```
smu92_32 -properties "C:\smu.properties" -only biservmid
```

7. Go to the **C:\SMU_packages*your-environment*\biservmid** directory and locate the WIP_database.zip file, which contains the data from SAS Table Server.

Step 2. Create the new target database.

Install and configure the new target database if you do not already have one running. If needed, consult the database administrator at your site for assistance.

Step 3. Populate the database with the tables that are needed for the Web Infrastructure Platform and Shared Services.

1. Open the database_postgres.properties file from the SAS WIP Database Importer that you previously downloaded. Update the values for the following properties to match the values for your environment:

   - **database.host**
   - **database.port**
   - **database.name**
   - **database.user.id**
   - **database.user.password**
   - **database.schema.pattern**

   Note: The PostgreSQL database schema is the uppercased value of the log on ID (for example, ${*DATABASE.USER.ID*}).

2. Open the build.properties file from the SAS WIP Database Importer. Remove the comment delimiters from the following line:

   ```
   ##database.type=postgres
   ```

3. Update the values for the following properties to match the values for your environment:

   a. **database.type**—specifies the type of the target database (for example, `database.type=postgres`.

   b. **sql.scripts.data.dir**—specifies the absolute path to the data directory in the SASHOME location, which contains the SQL scripts for the Web Infrastructure Platform that are used to create its database tables, load its out-of-the-box data, and drop its database tables (for example, **sql.scripts.data.dir=C:\\ Program Files\\SAS\\SASSharedServices\\9.2\\Config\\Deployment\\Data**).

   c. **database.zip.file**—specifies the absolute path to the v920m3 WIP_database.zip file from which records will be read and inserted into the target v920m3 database. The WIP_database.zip file is created by the SASSMU2 (for example, **database.zip.file=C:\\Public\\v920m3\DbImporter92m3 \\test\\SASSMU2\\WIP_database.zip**).

d. **log4j.config.file**—specifies the Log4J configuration file which is used by the database importer for the Web Infrastructure Platform. Edit the log4j.properties file to specify the location of the log file (for example, **`log4j.config.file=file:///C:/Public/v920m3/DbImporter92m3/log4j.properties`**).

e. **java.maxmemory**—specifies the maximum amount of memory that can be used by the database importer for the Web Infrastructure Platform (Java application) (for example, `java.maxmemory=2048m`).

4. Open a command window and change to the SAS WIP Database Importer directory. Submit the **`createDatabaseTables`** command to create the database tables for the Web Infrastructure Platform and Shared Services (for example, `C:\DbImporter92m3> ant -f build.xml createDatabaseTables`).

Step 4. Update the Web Infrastructure Platform and Shared Services data source that is defined in WebLogic.

The initial installation created a data source named SharedServices that references SAS Table Server. This step redirects that reference to a new data source that relies on PostgreSQL.

Update the values for the database name, host name, port number, user ID, and password for the PostgreSQL database Connection Pool in the WebLogic administrative console.

On the PostgreSQL WebLogic application server, the values are available from the WebLogic Administration Console. Select **SASDomain ► Services ► JDBC ► Data Sources ► SharedServices ► Configuration** and then click the **Connection Pool** tab. Update the values for the new PostgreSQL database.

Step 5. Update the configuration properties for the Web Infrastructure Platform and Shared Services.

The SAS installation and configuration process stores metadata about the properties for Shared Services. Updating the metadata is a manual change to the configuration. Therefore, the properties must be updated to accommodate possible future migrations. Here are the steps to update the metadata for the configuration properties:

1. Start SAS Management Console and log on as **`sasadm`** (or another fully privileged user ID).

2. Click the **Folders** tab and select **System ► Applications ► SAS Shared Services**.

3. Select the **`SharedServices9.2`** folder.

4. In the right pane, right-click **SharedServices9.2** and select **Properties**.

5. Click the **Configuration** tab.

6. Specify the following values for the SharedServices properties:

   a. **data.dbms.type**: **`postgres`**

   b. **dbms.biservmid.host:** ***`PostgreSQL-host-name`***

   c. **dbms.biservmid.jdbc.dir:** ***`directory-path-containing-the-PostgreSQL-JDBC-driver-JAR`***

d. **dbms.biservmid.name:** ***PostgreSQL-database-name***

e. **dbms.biservmid.port:** ***PostgreSQL-port-number***

f. **dbms.biservmid.userid:** ***PostgreSQL-userID***

g. **dbms.biservmid.validation.query:** **`select 1 from dual`**

7. Click **OK** to exit the dialog box, and then exit SAS Management Console.

Step 6. Migrate the Web Infrastructure Platform and Shared Services from SAS Table Server to the new database.

Migrate the data that was exported from SAS Table Server to the new database.

1. Open a command window and navigate to the directory for the SAS WIP Database Importer.
2. Submit the **`importDatabase`** command to create the Web Infrastructure Platform and Shared Services database tables (for example, `C:\DbImporter92m3> ant -f build.xml importDatabase`).

Step 7. Update the workflow to use the proper dialect.

Using the following steps, update the workflow to use the new database.

1. Open the sas.shared9.2.ear file. Then, within this EAR file, open the workflow.properties file, which resides in the **`\sas.workflow.war\WEB-INF\`** directory.
2. Add a comment delimiter to the following line:

   ```
   workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
   TSFirebirdDialect
   ```

   Here is an example:

   ```
   #workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
   TSFirebirdDialect
   ```

3. Remove the comment delimiter from the following line:

   ```
   #workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
   ```

   Here is an example:

   ```
   workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
   ```

4. Redeploy the sas.shared9.2.ear file.

Step 8. Populate the database with the tables that are needed for the SAS Content Server.

The scripts that were needed to create the tables for the SAS Content Server were executed by the **`createDatabaseTables`** command in the previous steps.

Step 9. Configure the SAS Content Server to use the new database as its repository.

This section discusses the steps that are needed to migrate the stored SAS Content Server data from SAS Table Server to a PostgreSQL database. It also details the steps that are needed to configure the SAS Content Server for the new PostgreSQL database. In order to migrate the data, you need to convert the SAS Content Server from using SAS Table Server to using the file system and then to using PostgreSQL.

Revert to the file system after you have configured the SAS Content Server for SAS Table Server.

1. Stop the WebLogic process.
2. Back up the following directory:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer**
3. Create a new directory as follows:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp**
4. Copy the repository.xml file that resides in the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates/** directory to the ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp** directory.
5. Change directory to ***SAS-configuration-directory*/Lev1/Web/Utilities**. Verify that your JCRCopyRepository.sh script has only one set of JNDI arguments for SAS Table Server. It should look similar to the following:

```
#!/bin/sh -p
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
picklist
DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/
Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/
config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.
nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
```

```
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc:sastkts://your-host-name:
 2172?constring=\(DSN=SharedServices\) \
-Dscs.jndi.driver=com.sas.tkts.TKTSDriver \
-Dscs.jndi.user=sastrust@saspw \
-Dscs.jndi.pwd=your-password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

6. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/Lev1/AppData
/SASContentServer/Repository /SAS-configuration-directory/Lev1/Lev1
/AppData/SASContentServer/temp
```

You are now using the file system as the temporary back end for the SAS Content Server. Next, configure the SAS Content Server to use PostgreSQL as the back-end database.

7. Change the name of the SAS Content Server repository from **Repository** to **RepositoryTS**.

   - For a Windows operating environment, use the following command:

```
move C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\RepositoryTS
```

   - For UNIX and z/OS operating environments, use the following command:

```
mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryTS
```

8. Create a new directory called **Repository** in the same location.

   - Under Windows, use the following command:

```
mkdir C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
```

   - Under UNIX and z/OS, use the following command:

```
mkdir SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
```

9. Copy the repository.postgres.xml file from the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates** directory to the ***/SAS-configuration-directory*/Lev1/AppData/SASContentServer/Repository** directory that was created in the previous step. Rename the file to repository.xml. The following example shows syntax for a UNIX environment:

```
cp /SASHOME/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-
INF/templates/repository.postgres.xml Repository/repository.xml
```

10. Obtain the values for the database name, host name, port number, user ID, and password from the web application server.

On the PostgreSQL WebLogic application server, the values are available from the WebLogic Administration Console. Select **SASDomain ► Services ► JDBC ► Data Sources ► SharedServices ► Configuration** and then click the **Connection Pool** tab access the values.

11. Change directory to **/*SAS-configuration-directory*/Lev1/Web/Utilities**. Ensure that the PostgreSQL driver is included in the DRIVER list in your JCRCopyRepository.sh script. Then update the JNDI values to the values that are needed to connect to the PostgreSQL database. Your script should look similar the following:

```
#!/bin/sh -p
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
picklist
DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/
Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/
config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.
nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar:/your-directory/your-
driver-name.jar
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jdbcUrl=jdbc:postgresql://host:5432/SharedServices\
-Dscs.jndi.driver=com.postgresql.jdbc.Driver \
-Dscs.jndi.user=your-id  \
-Dscs.jndi.pwd=your-password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

12. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/AppData
/SASContentServer/temp /SAS-configuration-directory/Lev1/AppData
/SASContentServer/Repository
```

13. Restart WebLogic.

Step 10. Verify that the new database is the new back-end data store.

To verify that the SAS Content Server is using PostgreSQL, save a new report in SAS Web Report Studio or add content from SAS Management Console to the SAS Content Server. The size of your tables will increase if the SAS Content Server is properly configured to use PostgreSQL.

To verify that the Web Infrastructure Platform and Shared Services are using PostgreSQL, save a new alert in the SAS BI Dashboard. The size of your tables will increase if the Web Infrastructure Platform and Shared Services are properly configured to use PostgreSQL.

Database Migration Instructions for the JBoss Application Server

Step 1. Extract the Web Infrastructure Platform and Shared Services data from SAS Table Server.

Extract the data that is related to the Web Infrastructure Platform and Shared Services from SAS Table Server. To do this, use the SAS Migration Utility and package the data in a ZIP file. You use this file later to load the data into the new database.

1. Download the SAS Migration Utility (support.sas.com/demosdownloads/setupcat.jsp?cat=SAS+Migration+Utility) and save it to a folder on your file system, such as **`C:\SMU`**.
2. Copy the contents of the smu.properties file that is located in the Appendix of this document and save it to a file on your system.
3. In the smu.properties file, update the values for the host name, port number, file path, user name, and password to match your environment.
4. Create the folder **`C:\SMU_packages`** on your file system. This is the location where the migration package will be created. The folder value should match the value for the **SMU.Output.Dir** property from the smu.properties file.
5. From the command prompt, change to the directory where the downloaded SAS Migration Utility is stored.
6. Submit the following command to create the migration package:

   ```
   smu92_32 -properties "path-to-your-smu.properties-file" -only biservmid
   ```

   Here is an example:

   ```
   smu92_32 -properties "C:\smu.properties" -only biservmid
   ```

7. Go to the **`C:\SMU_packages\`*`your-environment`*`\biservmid`** directory and locate the WIP_database.zip file, which contains the data from SAS Table Server.

Step 2. Create the new target database.

Install and configure the new target database if you do not already have one running. If needed, consult the database administrator at your site for assistance.

Step 3. Populate the database with the tables that are needed for the Web Infrastructure Platform and Shared Services.

Create the Web Infrastructure Platform and Shared Services tables in PostgreSQL.

1. Open the database_postgres.properties file from the SAS WIP Database Importer that you previously downloaded. Update the values for the following properties to match the values for your environment:

   - **database.host**
   - **database.port**
   - **database.name**
   - **database.user.id**
   - **database.user.password**
   - **database.schema.pattern**

   Note: The PostgreSQL database schema is the uppercased value of the log on ID (for example, ${*DATABASE.USER.ID*}).

2. Open the build.properties file from the SAS WIP Database Importer. Remove the comment delimiters from the following line:

   ```
   ##database.type=postgres
   ```

3. Update the values for the following properties to match the values for your environment:

   a. **database.type**—specifies the type of the target database (for example, `database.type=postgres`).

   b. **sql.scripts.data.dir**—specifies the absolute path to the data directory in the SASHOME location, which contains the SQL scripts for the Web Infrastructure Platform that are used to create its database tables, load its out-of-the-box data, and drop its database tables (for example, **`sql.scripts.data.dir=C:\\Program Files\\SAS\\SASSharedServices\\9.2\\Config\\Deployment\\Data\\`**).

   c. **database.zip.file**—specifies the absolute path to the v920m3 WIP_database.zip file from which records will be read and inserted into the target v920m3 database. The WIP_database.zip file is created by the SASSMU2 (for example, **`database.zip.file=C:\\Public\\v920m3\DbImporter92m3\\test\\SASSMU2\\WIP_database.zip`**).

   d. **log4j.config.file**—specifies the Log4J configuration file which is used by the database importer for the Web Infrastructure Platform. Edit the log4j.properties file to specify the location of the log file (for example, **`log4j.config.file=file:///C:/Public/v920m3/DbImporter92m3/log4j.properties`**).

e. **java.maxmemory**—specifies the maximum amount of memory that can be used by the database importer for the Web Infrastructure Platform (Java application) (for example, `java.maxmemory=2048m`).

4. Open a command window and change to the SAS WIP Database Importer directory. Submit the **`createDatabaseTables`** command to create the database tables for SAS the Web Infrastructure Platform and Shared Services (for example, `C:\DbImporter92m3> ant -f build.xml createDatabaseTables`).

Step 4. Update the Web Infrastructure Platform and Shared Services data source that is defined in JBoss.

The initial installation created a data source named SharedServices that referenced SAS Table Server. This step redirects that reference to a new data source that relies on PostgreSQL.

1. Copy the PostgreSQL JDBC driver JAR file ojdbc6.jar to the ***/JBoss-home*/server/SASServer1/lib** directory.

2. Open the SharedServices–ds.xml file that is located in the ***/JBoss-home*/server/SASServer1 /deploy/** directory. Update the values in the file to point to the PostgreSQL database, as shown in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasource>
<local-tx-datasource>
<driver-class>org.postgresql.Driver</driver-class>
<jndi-name>sas/jdbc/SharedServices</jndi-name>
<connection-property name="stmtPooling">0</connection-property>
<connection-property name="constring">(DSN=SharedServices)</connection-property>
<connection-url>jdbc:postgresql://host:5432/SharedServices</connection-url>
<user-name>your-user-name</user-name>
<password>your-password</password>
</local-tx-datasource>
</datasource>
```

Step 5. Update the configuration properties for the Web Infrastructure Platform and Shared Services.

The SAS installation and configuration process stores metadata about the properties for Shared Services. Updating the metadata is a manual change to the configuration. Therefore, the properties must be updated to accommodate for possible future migrations. Here are the steps to update the metadata for the configuration properties:

1. Start SAS Management Console and log on as **`sasadm`** (or another fully privileged user ID).

2. Click the **Folders** tab and select **System ► Applications ► SAS Shared Services**.

3. Select the **`SharedServices9.2`** folder.

4. In the right pane, right-click **SharedServices9.2** and select **Properties**.

5. Click the **Configuration** tab.

6. Specify the following values for the SharedServices properties:

a. **data.dbms.type**: **`postgres`**

b. **dbms.biservmid.host:** ***`PostgreSQL-host-name`***

c. **dbms.biservmid.jdbc.dir:** ***`directory-path-containing-the-PostgreSQL-JDBC-driver-JAR`***

d. **dbms.biservmid.name:** ***`PostgreSQL-database-name`***

e. **dbms.biservmid.port:** ***`PostgreSQL-port-number`***

f. **dbms.biservmid.userid:** ***`PostgreSQL-userID`***

g. **dbms.biservmid.validation.query:** **`select 1 from dual`**

7. Click **OK** to exit the dialog box, and then exit SAS Management Console.

Step 6. Migrate the Web Infrastructure Platform and Shared Services from SAS Table Server to the new database.

Migrate the data that was exported from SAS Table Server to the new database.

1. Open a command window and navigate to the directory for the SAS WIP Database Importer.

2. Submit the **`importDatabase`** command to create the Web Infrastructure Platform and Shared Services database tables (for example, `C:\DbImporter92m3> ant -f build.xml importDatabase`).

Step 7. Update the workflow to use the proper dialect.

Update the workflow in order to use the new database.

1. Open the sas.shared9.2.ear file. Within this EAR file open workflow.properties, which resides in the **`\sas.workflow.war\WEB-INF\`** directory.

2. Add a comment delimiter to the following line:

```
workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
TSFirebirdDialect
```

Here is an example:

```
#workflow.hibernate.dialect=com.sas.workflow.engine.services.dao.
TSFirebirdDialect
```

3. Remove the comment delimiter from the following line:

```
#workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

Here is an example:

```
workflow.hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

4. Redeploy the sas.shared9.2.ear file.

Step 8. Populate the database with the tables that are needed for the SAS Content Server.

The scripts that were needed to create the tables for the SAS Content Server were executed by the **createDatabaseTables** command in the previous steps.

Step 9. Configure the SAS Content Server to use the new database as its repository.

This section discusses the steps that are needed to migrate the stored SAS Content Server data from SAS Table Server to a PostgreSQL database. It also details the steps that are needed to configure the SAS Content Server for the new PostgreSQL database. In order to migrate the data, you need to convert the SAS Content Server from using SAS Table Server to using the file system and then to using PostgreSQL.

Revert to the file system after you have configured the SAS Content Server for SAS Table Server.

1. Stop the JBoss process.
2. Back up the following directory:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer**
3. Create a new directory as follows:

   ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp**
4. Copy the repository.xml file that resides in the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates/** directory to the ***SAS-configuration-directory*/Lev1/AppData/SASContentServer/temp** directory.
5. Change directory to ***SAS-configuration-directory*/Lev1/Web/Utilities**. Verify that your JCRCopyRepository.sh script has only one set of JNDI arguments for SAS Table Server. It should look similar to the following:

```
#!/bin/sh -p
#
# JCRCopyRepository.sh
#
. `dirname $0`/../../level_env.sh
LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
picklist
DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/
Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/
config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/
```

```
sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.
nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar
CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc:sastkts://your-host-name:
 2172?constring=\(DSN=SharedServices\) \
-Dscs.jndi.driver=com.sas.tkts.TKTSDriver \
-Dscs.jndi.user=sastrust@saspw \
-Dscs.jndi.pwd=your-password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

6. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/Lev1/AppData
/SASContentServer/Repository /SAS-configuration-directory/Lev1/Lev1
/AppData/SASContentServer/temp
```

You are now using the file system as the temporary back end for the SAS Content Server. Next, configure the SAS Content Server to use PostgreSQL as the back-end database.

7. Change the name of the SAS Content Server repository from **Repository** to **RepositoryTS**.

   - For a Windows operating environment, use the following command:

     ```
     move C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
     C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\RepositoryTS
     ```

   - For UNIX and z/OS operating environments, use the following command:

     ```
     mv SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
     SAS-configuration-directory/Lev1/AppData/SASContentServer/RepositoryTS
     ```

8. Create a new directory called **Repository** in the same location.

   - Under Windows, use the following command:

     ```
     mkdir C:\SAS-configuration-directory\Lev1\AppData\SASContentServer\Repository
     ```

   - Under UNIX and z/OS, use the following command:

     ```
     mkdir SAS-configuration-directory/Lev1/AppData/SASContentServer/Repository
     ```

9. Copy the repository.postgres.xml file from the ***SASHOME*/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-INF/templates** directory to the ***/SAS-configuration-directory*/Lev1/AppData/SASContentServer/Repository** directory that was created in the preceding step. Rename the file to repository.xml. The following example shows syntax for a UNIX environment:

   ```
   cp /SASHOME/SASWebInfrastructurePlatform/9.2/Static/wars/sas.svcs.scs/WEB-
   INF/templates/repository.postgres.xml Repository/repository.xml
   ```

10. Open the repository.xml file and search for the following parameter:

    ```
    <param name="url" value="sas/jdbc/SharedServices"/>
    ```

    Modify that parameter as follows:

    ```
    <param name="url" value="java:sas/jdbc/SharedServices"/>
    ```

    This value appears six times within the repository.xml file. You must modify all six instances of the parameter.

    Note: If a workspace.xml file exists in the ***/SAS-configuration-directory*/Lev1/AppData/SASContentServer/Repository/workspaces/default** directory on the middle-tier server, it will contain two instances of the same parameter. You must also modify those two instances with the value `<param name="url" value="java:sas/jdbc/SharedServices"/>`.

11. Obtain the values for the database name, host name, port number, user ID, and password from the web application server.

    On the JBoss web application server, these values are located in the SharedServices–ds.xml file that resides in the ***SAS-configuration-directory*/Lev1/Web/Common/jboss** directory.

12. Change directory to ***/SAS-configuration-directory*/Lev1/Web/Utilities**. Ensure that the PostgreSQL driver is included in the DRIVER list in your JCRCopyRepository.sh script. Then update the JNDI values to the values that are needed to connect to the PostgreSQL database. Your script should look similar the following:

    ```
    #!/bin/sh -p
    #
    # JCRCopyRepository.sh
    #
    . `dirname $0`/../../level_env.sh
    LAUNCHERJAR=$SASVJR_HOME/eclipse/plugins/sas.launcher.jar
    UTILITIESDIR=$LEVEL_ROOT/Web/Utilities
    PICKLISTS=/sas/SASWebInfrastructurePlatform/9.2/Picklists/wars/sas.svcs.scs/
    picklist
    DRIVER=/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/icu4j.
    jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/log4j.jar:
    /sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.jar:/
    sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.core.nls.
    jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.icons.
    contents.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
    sas.icons.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/
    sas.icons.nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
    JDBCDrivers/sas.intrnet.javatools.jar:/sas/config/Lev1/Web/Applications/
    SASSharedServices9.2/JDBCDrivers/sas.intrnet.javatools.nls.jar:/sas/config/Lev1/
    Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.nls.collator.jar:/sas/
    config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.jar:/
    sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.oda.tkts.
    ```

```
nls.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/JDBCDrivers/sas.
security.sspi.jar:/sas/config/Lev1/Web/Applications/SASSharedServices9.2/
JDBCDrivers/sas.svc.connection.jar:/sas/config/Lev1/Web/Applications/
SASSharedServices9.2/JDBCDrivers/sas.svc.connection.nls.jar:/your-directory/your-
driver-name.jar

CLASSPATH=$UTILITIESDIR:$LAUNCHERJAR
"$JAVA_JRE_COMMAND" \
-classpath "$CLASSPATH" \
-Djava.system.class.loader=com.sas.app.AppClassLoader \
-Dsas.app.launch.config="$PICKLISTS" \
-Dsas.app.repository.path="$SASVJR_REPOSITORYPATH" \
-Dsas.app.class.path="$UTILITIESDIR:$DRIVER" \
-Djava.security.auth.login.config=../Common/login.config \
-Xmx256m \
-Dscs.jndi.jndiName=sas/jdbc/SharedServices \
-Dscs.jndi.jdbcUrl=jdbc:postgresql://host:5432/SharedServices\
-Dscs.jndi.driver=org.postgresql.Driver \
-Dscs.jndi.user=your-id   \
-Dscs.jndi.pwd=your-password \
org.apache.jackrabbit.core.JCRCopyRepository $1 $2
exit 0
```

13. Execute the following command:

```
./JCRCopyRepository.sh /SAS-configuration-directory/Lev1/AppData
/SASContentServer/temp /SAS-configuration-directory/Lev1/AppData
/SASContentServer/Repository
```

14. Restart JBoss.

Step 10. Verify that the new database is the new back-end data store.

To verify that the SAS Content Server is using PostgreSQL, save a new report in SAS Web Report Studio or add content to the SAS Content Server from SAS Management Console. The size of your tables will increase if the SAS Content Server is properly configured to use PostgreSQL.

To verify that the Web Infrastructure Platform and Shared Services are using PostgreSQL, save a new alert in the SAS BI Dashboard. The size of your tables will increase if the Web Infrastructure Platform and Shared Services are properly configured to use PostgreSQL.

Appendix

This appendix contains the contents of the smu.properties file.

```
   #
   # The configuration directory from which the migration
   # utility will read.  This should typically be the "Lev"
   # directory.  It will usually contain SASMain, Data,
   # and Product directories.
   #

SMU.config.dir=C:\\SAS\\EntBIServer\\Lev1

   #
   # The directory that contains the SAS executable.
   #

SMU.SASROOT=C:\\Program Files\\SAS\\SASFoundation\\9.2

   #
   # The directory that is the installation base of
   # the SAS product set.
   #

SMU.SASHOME=C:\\Program Files\\SAS

   #
   # The metadata server host. This must be defined for
   # the migration utility to run successfully.  The
   # port defaults to 8561.  If a different metadata
   # port was used, this must be set to the correct port.
   #

SMU.host.metadata=your-metadata-server-host-name
   #SMU.port.metadata=8561

   #
   # The metadata server administrative user and password.
   # This must be an unrestricted user so that all data can be
   # read for metadata extraction.  The password should be
   # encoded using the {sas001} method.  Use PROC PWENCODE
   # to get the encoded password.
   #

SMU.user=sasadm@saspw
SMU.password=your-password

   #
   # A workspace profile is acceptable as an alternative
   # to providing the host, port, user, and password
   # (if the password is in the profile).  This can be
   # just the profile name, which will look in the default
   # location and the current working directory, or a full
   # path to the profile.
```

```
#
#SMU.profile=my-server

#
# The migration package output directory.  It will be
# created if this is being run against the metadata server
# tier.  Otherwise, it should already contain the results
# from your metadata server tier run and any other upstream
# tiers.  The results from every tier of your deployment
# should be included in the same migration package.
#

SMU.Output.Dir=C:\\SMU_packages

#
# If the SMU is being run on a system with multiple network
# interfaces, or a dynamic host name, this property might
# need to be set to get the "right" name that is used for
# directory naming, and so on.
#
#SMU.localhost=my.localhost.com

#
# Specify the user ID and password which will be used to
# open a connection to the Shared Services database.
# If the database for shared services is the SASTable Server,
#supply the credentials for the SAS Trusted User.
# If the database for shared services is not a SAS database,
# supply the credentials that are appropriate for that database.
#

SMU.webinfpltfm.dbms.userid=sastrust@saspw
SMU.webinfpltfm.dbms.password=your-password

#
# In order for the SAS Content Server content repository
# to be copied, the content server web application has
# to be stopped.  This property allows the migration
# utility to pause and prompt the user to stop the
# application, copy the repository, and then pause and
# prompt the user to restart the application.  If this
# is false, and a lock on the repository is detected by
# the analysis, it will put an error message in the
# report and not attempt the copy.
#
#SMU.scs.allow.sync=true

#
# These properties are used for SAS Content Servers that
# have had their repositories customized by the end user.
# If you don't understand these, chances are you don't
# need them.
#
#scs.jndi.jndiName=
#scs.jndi.driver=
#scs.jndi.jdbcUrl=
#scs.jndi.user=
#scs.jndi.pwd=
```

```
   #scs.jndi.jdbcdir=

   # Specifies non-standard locations for SAS application data sets and
   # catalogs for the migration utility to move.
   #
   # The utility packages these directories in the levconfig output
   # folder in the "userdirs" subdirectory.
   #
   # List absolute paths or paths relative to the SAS configuration
   # directory. Separate multiple paths with a comma.
   #
   # levconfig.user.dirs=my_SAS_solution_data_sets,my_SAS_solution_catalogs,
   # C:\my_data\my_SAS_solution_misc

   # This property should be set to true, if SMU will be run multiple
   # times on the same machine because multiple tiers of SAS 9.2
   # are deployed on the same machine.
   #SMU.isMultipleTierMachine=true

SMU.SAS.version=9.2
```