Siebel Deployment or Releases are done to **prorogate the developed changes from one environment to another environment** like Dev to test, test to UAT, UAT to prod. Migrating repositories between databases is a common requirement when going live to a new development, testing, or production environment.

Whenever there is release or migration between environments the following Type of data needs to be moved.

1. SRF
2. Siebel Repository
3. Master Data

**Siebel Repository Migration from Source to Target:**

**Siebel Repository Migration** is required whenever **there is change in workflows or database layer objects** (**Table, columns or indexes etc…)** Siebel Repository. Sometimes we will move only the SRF without Siebel Repository for small bug fix releases. It is Best practice to move the Siebel repository for all the Migrations or Builds from one Environment to another environment regardless of the type of release**(may be bug fix or big Release**).

There are two ways of doing the Repository migration

1. Though the Database configuration Wizard provided by the Siebel which provides user interface
2. Though the Scripts – Directly using the scripts which are intern created and executed by the Siebel Database Configuration Wizard.

Even if you do the Repository migration though the Database configuration wizard it will create same scripts and execute in the back ground.

**Please refer the Going live with the Siebel Business Application PDF from Siebel bookshelf for doing the repository migration through Siebel Database Configuration Wizard.**

**Steps for Repository migration through scripts (Windows Environment):**

1. Export the Repository from Source Environment(DEV):
   * Login to one of the Application server in the Source Environment (Dev)
   * Open the Command prompt and Execute the Following Command with the Parameters of the Environment.  
     $SIEBEL\_HOME\siebsrvr\bin\repimexp.exe /a E /G ALL /u SADMIN /p password /c devsiebent\_DSN /d siebel /r ‘Siebel Repository’ /f D:\export.dat /l D:\export.log
   * Where $SIEBEL\_HOME\siebsrvr\bin\repimexp.exeis the Location of the **repimexp.exe,** which is Executable file to Export Siebel repository. Same Executable will be used for the Repository Export and Import but with Different Action (/a) option.
   * After successful run of the above script Export Dump file (export.dat) and Log file will be created in the specified directories in the script.
2. Move the export file to the Application server in Target environment.
3. Stop the Siebel Application servers in target environment
4. Import the Repository in the Destination Environment using the below Command
   * $SIEBEL\_HOME\siebsrvr\bin\repimexp /a I /G ALL /u SADMIN /p \*\*\*\*\*\*\* /c devsiebent\_DSN /d siebel /r **NewSiebelRepository** /f D:/export.dat /l D:\import.log
5. Log-in to tools in the target environment and rename the Old repository appropriately and rename the newly imported repository (**NewSiebelRepository)** to ‘Siebel Repository’.
6. Run the DDL SYNC if there is any Table Level Changes in the New Siebel Repository using below Scripts.
   * Create the Schema file from the Repository using below command:  
     $SIEBEL\_HOME\siebsrvr\bin**\**ddldict /u sadmin /p SADMIN\_PASSWORD /c devsiebent\_DSN /d Siebel /f ‘D:\schema.ddl’ /e y /a y /l ‘/D:\dictionayschema.log’ /n ‘Siebel Repository’ /T DCIR
   * Apply the Schema changes to database from schema file:  
     $SIEBEL\_HOME\siebsrvr\bin**\**ddlimp /u siebel /p SIEBEL\_PASSWORD /c devsiebent\_DSN /g SSE\_ROLE /f ‘D:\schema.ddl’ /e n /B TS\_PRDSBL\_DATA /X TS\_PRDSBL\_IDX /R Y /W Y /s N /l ‘/D:\Applyschema.log’
7. Move the SRF to the Target Environment servers
8. Run the **genbscript** to generate browser script using following command and move it to Web server’s Public Folder
   * $SIEBEL\_HOME\siebsrvr\bin\genbscript $SIEBEL\_HOME\siebsrvr\bin\enu\publicsector.cfg TARGET\_LOCATION\_PATH ENU
   * You need to run this command once for each language your Application supports
9. Start the Application servers

**Master Data Migration:**

To move Master data such as List of Values, responsibilities, Views, PDQs etc we can use one of the below given approaches

1. [ADM (Application Deployment Manager)](http://siebelunleashed.com/application-deployment-manager-adm-introduction/): Please Refer the Siebel Application Deployment manager Guide for migrations using ADM.
2. [EIM (Enterprise Integration Manager):](http://siebelunleashed.com/siebel-enterprise-integration-manager-eim-an-overview/) can also be used to achieve the same.

**SRF Movement from Source (DEV) to Target (Test) Environment:**

SRF can be moved in following ways from source to target.

1. Compile SRF in Source Environment (DEV) and move it to target Environment. This approach is followed Do this whenever there in no repository movement (in minor bug fixes release).
2. Move the Siebel Repository to the Target Environment (Test) and Compile SRF in target environment. This is followed normal releases.

Using the **Command Line Scripts** is very easy if you understand importance and Parameters of Each Command.  Below is the explanation of some of the switches used in different commands in this article:

**/a**: Represents the Action to be taken, **E for Export** the Siebel Repository. **I for import** the Siebel repository.  
**/G:** Rrepresents the languages to be exported or imported. **Option AL**L will export or Import all the languages in the Repository or DAT file  
**/C:** Represents the **Server Data source** name to connect to Database**.** Check the Application cfg file or ODBC configuration Tool for the exact server Data source name  
**/d:** Represents the table owner name, normally it is SIEBEL  
**/r:** Represents the **Repository name** to be Exported .Most of the Environments it will be ‘Siebel Repository’. Since a Siebel Environment can have multiple Repositories but only one with the Name ‘Siebel Repository’ **(repository name to be used by Application is specified in the Application cfgs)**.  
**/f:** Represents Export file name and Location, it will be in .dat format.  
**/l:** Specifies is the Location of the Log File.

The above commands Paths and Directory structure will change between Windows and UNIX environments. You have to 4 Commands to do the entire Repository migration task. Prepare the above commands and save them as Batch scripts in Windows and Shell scripts in UNIX for future release. You can do simple modifications to these scripts for every release and execute them.

***Application release automation for Siebel:***

Siebel releases are complex. Executing one well is difficult. Extracting databases, migrating SRF repositories, deploying non-repository items and managing all the planned and unplanned outages is no small feat.

how they used Application Release Automation to shorten downtime of Siebel and increase quality and stability at a huge telecom provider in the Netherlands.

* Enable agile releases and deploy Siebel updates in just minutes
* Reduce planned and unplanned downtime
* Deploy the right artifacts and configuration settings for even the most complex environments
* Prevent unauthorized access to production environments
* Free up Siebel administrators and developers from release and deployment tasks

**Automic Siebel Deployment Automation:**

**Problem**

Siebel is just not as agile as the many other applications at the front end of your business today, making it a bottleneck for innovation and potentially impacting customer experience. Deployments are manual, require a huge amount of co-ordination between different teams and take forever to complete. Overruns cause unplanned downtime, and with call centers unable to take calls, hundreds of agents sit idle while customers become dissatisfied.

**Solution**

Automic Siebel Deployment Automation increases the agility in Siebel environments.

Deep integration with Siebel Tools coupled with preconfigured workflow templates for full Siebel Export and Import provide fast, consistent and reliable Siebel deployments every time – reducing downtime and leaving new application development and integration initiatives uninterrupted.

**Features:**

* Automate the entire Siebel deployment process for multiple environments from a single point of control
* Enable deep integration with Siebel through the Siebel Data Bean API
* Leverage preconfigured workflow templates that offer full Siebel Export and packages SRF’s, browser scripts, repository files, schema files into a single package ready for deployment
* Ensure graceful shutting down of servers in the correct order and importing SRFs and repositories in parallel, before verification, utilizing preconfigured workflows for full Siebel Import
* Establish process controls to ensure only authorized staff can promote updates to Siebel production environments
* Gain visibility across every step of the Siebel deployment process with full audit trails
* Safely roll back to the previous application state should a deployment failure occur

Problem

Siebel is just not as agile as the many other applications at the front end of your business today, making it a bottleneck for innovation and potentially impacting customer experience. Deployments are manual, require a huge amount of co-ordination between different teams and take forever to complete. Overruns cause unplanned downtime, and with call centers unable to take calls, hundreds of agents sit idle while customers become dissatisfied.

Solution

Automic Siebel Deployment Automation increases the agility in Siebel environments. Deep integration with Siebel Tools coupled with preconfigured workflow templates for full Siebel Export and Import provide fast, consistent and reliable Siebel deployments every time – reducing downtime and leaving new application development and integration initiatives uninterrupted.

Problem

Siebel is just not as agile as the many other applications at the front end of your business today, making it a bottleneck for innovation and potentially impacting customer experience. Deployments are manual, require a huge amount of co-ordination between different teams and take forever to complete. Overruns cause unplanned downtime, and with call centers unable to take calls, hundreds of agents sit idle while customers become dissatisfied.

Solution

Automic Siebel Deployment Automation increases the agility in Siebel environments. Deep integration with Siebel Tools coupled with preconfigured workflow templates for full Siebel Export and Import provide fast, consistent and reliable Siebel deployments every time – reducing downtime and leaving new application development and integration initiatives uninterrupted.

Problem

Siebel is just not as agile as the many other applications at the front end of your business today, making it a bottleneck for innovation and potentially impacting customer experience. Deployments are manual, require a huge amount of co-ordination between different teams and take forever to complete. Overruns cause unplanned downtime, and with call centers unable to take calls, hundreds of agents sit idle while customers become dissatisfied.

Solution

Automic Siebel Deployment Automation increases the agility in Siebel environments. Deep integration with Siebel Tools coupled with preconfigured workflow templates for full Siebel Export and Import provide fast, consistent and reliable Siebel deployments every time – reducing downtime and leaving new application development and integration initiatives uninterrupted.

Problem

Siebel is just not as agile as the many other applications at the front end of your business today, making it a bottleneck for innovation and potentially impacting customer experience. Deployments are manual, require a huge amount of co-ordination between different teams and take forever to complete. Overruns cause unplanned downtime, and with call centers unable to take calls, hundreds of agents sit idle while customers become dissatisfied.

Solution

Automic Siebel Deployment Automation increases the agility in Siebel environments. Deep integration with Siebel Tools coupled with preconfigured workflow templates for full Siebel Export and Import provide fast, consistent and reliable Siebel deployments every time – reducing downtime and leaving new application development and integration initiatives uninterrupted.

Automate the entire Siebel deployment process for multiple environments from a single point of control

•Enable deep integration with Siebel through the Siebel Data Bean API

•Leverage preconfigured workflow templates that offer full Siebel Export and packages SRF’s, browser scripts, repository files, schema files into a single package ready for deployment

•Ensure graceful shutting down of servers in the correct order and importing SRFs and repositories in parallel, before verification, utilizing preconfigured workflows for full Siebel Import

•Establish process controls to ensure only authorized staff can promote updates to Siebel production environments

•Gain visibility across every step of the Siebel deployment process with full audit trails

•Safely roll back to the previous application state should a deployment failure occur

**Build plugin on your own:**

If you want to build the plugin on your own, set up your eclipse environment. In order to do this, you need to install the UrbanCode Plugin development environment for eclipse The project for the plugin is located here: <https://hub.jazz.net/project/ucplugin/UrbanCode%20Deploy%20Plug-in%20For%20Eclipse%20Development>

If you want to check out the code directly from this project, you will also need the eclipse Jazz plugin. The Jazz eclipse plugin update site is (at time of writing) here <https://jazz.net/downloads/rational-team-concert/5.0.2/5.0.2/p2> Once you've installed the Jazz plugin just click on "Configure eclipse client" and paste the proviced xml into the popup that will appear in eclipse when you select File->Accept JazzHum client configuration...

http://www.ibm.com/support/knowledgecenter/SSEP7J\_10.2.0/com.ibm.swg.ba.cognos.vvm\_user\_guide.10.2.0.doc/t\_vvm\_user\_siebel\_jars\_install.html

**How to build the plugin from eclipse client:**

1. Expand the Groovy project that you checked-out from example template.
2. Open build.xml file and execute it as an Ant Build operation (Run As -> Ant Build)
3. The built plugin is located at releases/UCDSiebelPlugin\_v.zip

**How to build the plugin from command line:**

1. Navigate to the base folder of the project through command line.
2. Make sure that there is build.xml file there, and then execute 'ant' command.
3. The built plugin is located at releases/UCDSiebelPlugin\_vdev.zip Note: Edit the Build.xml to change the version 'dev' to a release number.

**Compatibility:**

1. The IBM UrbanCode Deploy automation plugin works with Siebel version 8.X.
2. This plugin requires version 6.1.1 or later of IBM UrbanCode Deploy.
3. <https://hub.jazz.net/project/dennisstav/UrbanCode%20Deploy%20Siebel%20Plugin/overview>

**Installation:**

1. https://github.com/IBM-UrbanCode/Siebel-UCD?cm\_mc\_uid=77921233249114478292269&cm\_mc\_sid\_50200000=1477643948
2. Note: Two Siebel Data Bean specific jars are required to run this plugin: Siebel.jar and SiebelJI\_enu.jar.
3. These can be downloaded from Oracle's website. To install the jars, follow these steps:
4. first extract the plugin, then copy the jars into the /lib directory, re-zip the extracted plugin files,
5. and install as normal into UCD. These jars must be named as stated above. Directions to find the
6. jars on your local Siebel instance:
7. <http://www.ibm.com/support/knowledgecenter/SSEP7J_10.2.0/com.ibm.swg.ba.cognos.vvm_user_guide.10.2.0.doc/t_vvm_user_siebel_jars_install.html>

**Installing the Siebel Data Bean JAR files**

****

You must install the Siebel Data Bean JAR files to enable connectivity to Siebel.

**Before you begin**

If you do not have access to the Siebel Data Bean Jar files see your Siebel System Administrator.

**About this task**

The following instructions describe how to install the Siebel Data Bean JAR files for Siebel versions 8.0 and 8.1.

**Procedure**

1. Create a directory for the Siebel Data Bean JAR files at the following location <VVM>/apps/dlm/app\_ds\_siebel/lib/<version>, where *<VVM>* is the root directory of your Virtual View Manager Server and *<version>* is the Siebel version (8.0 or 8.1).
2. Copy the Siebel.jar and SiebelJI\_enu.jar files obtained from your Siebel Administrator to the directory that you created in Step 1.