Banking Application Build Process Guide

Version 2005

May 2005

Siebel Systems, Inc., 2207 Bridgepointe Parkway, San Mateo, CA 94404

Contents

1 What's New in This Release

What's New in Banking Application Build Process Guide, Version 2005

Table 1 lists changes in this version of the documentation to support release 2005 of the software.

Table 1. What's New in Banking Application Build Process Guide, Version 2005

Topic	Description
Banking Application Components	Updated because Entitlements is no longer a Web application, but is part of the Administration Console.
Module Layer Classes JAR Files	References to Solutionset Layer were changed to Module Layer.
General Support Classes JAR Files	Updated to remove the information about the entitlement-war-support.jar file.
Signed JAR Files in the Release	References to Solutionset Layer were changed to Module Layer. The entry for entitlement-war-support.jar file was removed.
Teller, Internet Banking, and Supports	Updated to remove the information about Entitlements from the heading and body of the topic.
Prerequisites for the WebLogic Build Process	Updated to reflect the change in the settings for the build properties file.
Running the Automated WebLogic Build Process	Updated to reflect the change in the build commands.
Building WebLogic Projects Individually	Updated to reflect the change in the build commands.
Chapter 5 WebSphere 5.1.1 Build Process	References to WebSphere 5.1 were changed to WebSphere 5.1.1.
Prerequisites for the WebSphere Build Process	Updated for the support for Oracle and DB2 databases.
Running the Automated WebSphere Build Process	Updated with the changed build commands for support of Oracle and DB2 commands.
Building WebSphere Projects Individually	Updated with the changed build commands for support of Oracle and DB2 commands.

2 Banking Application Components

This chapter introduces the Siebel Retail Finance Banking Application components and their build process. It contains the following topics:

- Banking Application Components
- About the Build Process

Banking Application Components

The Siebel Retail Finance Banking Application consists of the following main components:

- A single server-side business logic, shared between two products: Branch Teller and Internet Banking. This business logic is contained within the Enterprise Archive (EAR) file that deploys the Banking Application as a set of enterprise bean JAR files.
- A Java Swing-based front end for the Branch Teller Client and the Branch Teller Administrator Client, deployed within the EAR file as Java WebStart-enabled applications.
- A Java Server Pages (JSP)-based front end for the Internet Banking product. This is deployed in the same EAR file as the server-side business logic.
- A Branch Offline Server.
- A Web application for MCA Services administration functionality.

About the Build Process

There is a single build process for building both Branch Teller and Internet Banking because they share a common server side; the front end is the only difference between the two products from a build perspective.

The process for building the EAR file is the same regardless of whether a license is held for Branch Teller, Internet Banking, or both. For example, if just Branch Teller has been licensed, the Internet Banking collateral is not available; the build process recognizes this, and skips that stage of the process.

Any build process steps that are specific to a particular product are noted in this guide.

3 Build Process Prerequisites

You must extract the Banking Application for the appropriate target application server and database platform from the appropriate Banking Application pack so that required directory structures are created. See the *Siebel Retail Finance Banking Application Installation Guide* for details about the Banking Application packs available for the supported platforms in the current release.

The build process is currently supported on the Windows platform only. As part of the installation process, the Banking Application pack is extracted to D:\siebel.

This chapter contains the following sections:

- Extracting the Source Code
- Banking Application Resources

Extracting the Source Code

Before you perform the build process, you must prepare the build environment. This preparation involves extracting the source code, support files, and build files from the versions of the Siebel Retail Finance Common Software Resources and product Build Pack CDs that are appropriate to the product or platform for which they are being built.

Extracting Resources for Building Branch Teller on WebLogic

Extract the following resources to the D:\ path:

- The SRFBankingApplicationCommonSoftwareResources.jar file from the Siebel Retail Finance Common Software Resources CD
- The SRFBranchTellerBuildPack.jar file from the Siebel Retail Finance Branch Teller Extension Pack CD

Extracting Resources for Building Branch Teller on WebSphere

Extract the following resources to the D:\ path:

- The JAR SRFBankingApplicationCommonSoftwareResources.jar file from the Siebel Retail Finance Common Software Resources CD
- The SRFBranchTellerBuildPack.jar file from the Siebel Retail Finance Branch Teller Extension Pack (WebSphere)

Banking Application Resources

The following sections describe the resources that you extract:

- The Standard Directory Structure
- Classes JAR files in the Standard Directory Structure
- Signed JAR Files in the Release
- Source Code in the Release
- Teller, Internet Banking, and Supports

The Standard Directory Structure

After you extract the files contained in the Common Software Resources and Build Packs, the Standard Directory structure exists for the Banking Application, as shown in Table 2.

Table 2. Standard Directory Structure

Subdirectory of \siebel	Description
\3rdParty	Contains third-party libraries required for supporting the Banking Application. You must install additional third-party JAR files that are not included with this distribution, see *Siebel Retail Finance Banking Application Installation Guide* for further details.
\Branch	Contains all server-side implementation layer source code, deployment descriptors, resources for the Banking Application, and ANT scripts to compile the Banking Application.
\branch-common	Contains common branch code.
\BranchAdministrationClient	Contains all the front-end code to run the Branch Teller administration application, and an ANT script to compile the Web Archive (WAR) file. [1]
\Branch Client	Contains all the front-end code to run the Branch Teller client, and an ANT script to compile the WAR file. [2]
\Build	Contains the ANT scripts and other required resources for assembling and compiling the Banking Application and for generating a deployable EAR file.
\Common	Contains the common JAR files and resource files that are shared between Banking Application products.
\deploy	Contains the prebuilt deployable Banking Application EAR file siebel.ear for deployment, and data to

[1] Only present if Branch Teller is licensed.
[2] Only present if Branch Teller is licensed.

Subdirectory of \siebel	Description
	populate the application database.
\Entitlements	Contains the Entitlements source code, deployment descriptors, and ANT scripts to build Entitlements.
\eBanking	Contains the Internet Banking source code, JSPs, and ANT scripts to build Internet Banking[3].
\Offline	Contains the Offline source code, and ANT scripts to compile and build the Offline Server.
\database	Contains the database dump file, and reset scripts for the database.

A third-party library that is not in the release distribution, comm.jar is required to compile the Banking Application and enable some peripheral devices. In addition, the jFdfTk.jar library is required to support the creation of a PDF form of a currency transaction report. See the *Siebel Retail Finance Banking Application Installation Guide* for instructions on how to obtain these libraries. When you have downloaded these files, copy them to the following locations:

- \siebel\3rdParty\lib\comm\Win32\comm.jar
- \siebel\3rdParty\lib\fdf\jFdfTk.jar

You must create the folders if they are not already present.

All the source code and supports required for a Banking Application build should now be in place.

Classes JAR files in the Standard Directory Structure

The Siebel Retail Finance Common Software Resources pack contains several important classes JAR files contained in the \siebel\Common\lib folder. The JAR files in this folder are described in the following sections.

General Support Classes JAR Files

The following JAR files contain compiled classes only (no source code). For ease of future support, you must add these JAR files to the classpath of the extended Banking Application when it is being extended and deployed.

Table 3. General Support Classes JAR Files

Support classes JAR File	Description
mca.jar	Contains MCA Services classes.
core-bos.jar	Contains the Core and Sector Layer classes for all Banking Application entity beans.

[3] Internet Banking front-end code is only present if the Internet Banking product has been licensed.

Support classes JAR File	Description
bfa-utils.jar	Contains utility classes.
statemachine-ext.jar	Contains extension classes for the StateSoft Statemachine framework supporting the Screen Orchestrator tool in the Financial Transactions WorkBench.
mca-eabmq.jar	Contains Enterprise Access Bean (EAB) and MQSeries connector helpers.

Module Layer Classes JAR Files

The following JAR files contain compiled classes only (no source code). For ease of future support, you must add these JAR files to the classpath of the extended product when it is being deployed.

Table 4. Module Layer Classes Jar Files

Module Layer classes JAR file	Description
branch.jar	Contains all server-side classes, including EJBs, Parameter Objects, Constants and utility classes, for the Module Layer of the Banking Application codebase.
entitlements.jar	Contains module level classes from the Entitlements submodule.
branch-common-fe.jar	Contains all common front end classes used between the Branch Teller client and the Branch Administration Client.
branch-common.jar	Contains all common classes used between the Branch Teller client and the Branch Server side.
offline.jar	Contains module level classes from the Offline sub-module.

Domain Layer Classes JAR Files

There are only compiled classes in these JAR files (no source code). As these are domain-level JAR files, the classes in these JAR files are altered when extensions are made.

Table 5. Domain Layer Classes JAR Files

Domain Layer classes JAR file	Description
branch-impl.jar	Contains the Domain Layer classes for all server-side Banking Application classes: Entity, Session, Parameter/Value/Factory objects, and non-BFA utility classes. Some or all of the classes in this JAR file might

Domain Layer classes JAR file	Description
	be required for deployment of the generic or extended Banking Application.
entitlements-impl.jar	Contains domain-level classes from the Entitlements submodule.
branch-common-impl.jar	Contains all common Domain Layer classes, shared between the Branch Teller client and the Branch Server side.

Signed JAR Files in the Release

A number of JAR files supplied with this release are signed. Signing is carried out for JAR files that contain compiled code pertaining to the Core, Sector, and Module Layers of the four-tier structure (the layers that must not be altered on a customer installation). The purpose of siging is to aid the verification of JAR versions. Signing a JAR file prevents any changes being made to the file. Thus, Siebel Systems can be sure that the version of a JAR file being used on a customer site corresponds exactly to the version originally supplied.

The following JAR files are signed as they must not be modified in customer implementations:

- \siebel\Common\lib\bfa-utils.jar
- \siebel\Common\lib\branch.jar
- \siebel\Common\lib\core-bos.jar
- \siebel\Common\lib\entitlements.jar
- \siebel\Common\lib\mca.jar
- \siebel\Common\lib\mca-eabmq.jar
- \siebel\Common\lib\statemachine-ext.jar
- \siebel\Common\lib\offline.jar
- \siebel\Common\lib\branch-common-fe.jar
- \siebel\Common\lib\branch-common.jar

The following JAR files are not signed as you can modify them in customer implementations:

- \siebel\Common\lib\branch-impl.jar
- \siebel\Common\lib\entitlements-impl.jar
- \siebel\Common\lib\branch-common-impl.jar

Source Code in the Release

The source code for the Domain Layer is provided as Java files in the various source folders in the standard directory structure. The locations used are the locations expected by the build files that are part of this release.

The following .jar files are provided:

- offline-src.jar
- offline-impl-src.jar
- branch-impl-src.jar
- branch-common-impl-src.jar
- branch-common-fe-src.jar
- entitlements-impl-src.jar

Other source files can be provided depending on the terms of your contract.

The following .jar files can be provided:

- entitlements-src.jar
- core-bos-src.jar
- branch-src.jar
- branch-common-src.jar

Teller, Internet Banking, and Supports

The Branch Teller client application is implemented as a Swing application using the Screen Orchestrator tool. This tool uses a product statechart that can be extended in client customizations that use the Screen Orchestrator.

The statechart XML files and supports for the Banking Application components are located in subfolders in the Standard Directory Structure as shown in Table 6.

Table 6. Statechart XML files and Supports for the Banking Application Components

Component	Subfolders
Branch Teller Client	`\siebel\BranchClient\statechart`
Branch Teller Administration Client	`\siebel\BranchAdministrationClient\statechart`
Internet Banking	`\siebel\eBanking\Source\JSPs`

4 WebLogic 8.1 Build Process

You can run an automated build process, or you can build projects individually.

This chapter contains the following topics:

- Prerequisites for the WebLogic Build Process
- Running the Automated WebLogic Build Process
- Building WebLogic Projects Individually

Prerequisites for the WebLogic Build Process

To run the WebLogic build process, WebLogic 8.1.0 must be installed on the build machine.

To build an EAR file, you must configure the \siebel\Common\build.properties file.

Configure the following settings in the build.properties file. These settings must not contain empty strings and the storepass value must be at least six characters in length.

```
alias=your keystore alias

storepass=your keystore password

companyName=your section

operatingUnit=your unit

organisation=your company

country=your country
```

This signs the JAR files and WAR files with the credentials that you supply (signing is needed for the front-end WAR files that are deployed using WebStart).

Running the Automated WebLogic Build Process

To build the Retail Finance EAR file you must build the projects in the following order:

1. Branch (includes internetBanking and branch-common projects)
2. BranchAdministratorClient
3. BranchClient
4. Build

The build scripts supplied in the product extension packs automatically build the projects in this order.

The \siebel\Build folder contains:

- A Windows script, JavaPrompt.vbs, which creates a command prompt that has the classpath and path variables configured to invoke the Banking Application build process.
- A build script, build-all.xml, which builds all the projects in the correct order and creates the EAR file in the \siebel\Build\release folder.

To run the automated build process

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

```
cd Build

buildall weblogic
```

Building WebLogic Projects Individually

If required, you can build each project individually.

To build the Entitlements project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

```
cd Entitlements

build weblogic
```

NOTE: The MCA and Entitlements projects are delivered precompiled and are not rebuilt during the build. The precompiled EJB file is located in the subfolders at Siebel\Build\resource.

To build the Branch project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

```
cd Branch

build weblogic
```

To build the BranchAdministratorClient project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   ```
   cd BranchAdministratorClient
   ```

   ```
   build
   ```

   The ANT build process creates a branchadmin.war WAR file, and this Web application is stored in D:\siebel\Build\staging\war.

To build the BranchClient project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   ```
   cd BranchClient
   ```

   ```
   build
   ```

   The ANT build process creates a branchteller.war WAR file, and this Web application is stored in D:\siebel\Build\staging\war.

To build the EAR file

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   ```
   cd Build
   ```

   ```
   build weblogic
   ```

After you rebuild the EAR, you can redeploy it on your application server. See the *Siebel Retail Finance Banking Application Installation Guide* for instructions.

5 WebSphere 5.1.1 Build Process

You import and build the Banking Application build process for WebSphere 5.1.1 by using the EJBDeploy.bat file that is located in the *WebSphereHome*\bin folder. This chapter provides an example build process; it is not an exhaustive guide to building the Banking Application for WebSphere.

This chapter contains the following topics:

- Prerequisites for the WebSphere Build Process
- Running the Automated WebSphere Build Process

Prerequisites for the WebSphere Build Process

You must have WebSphere 5.1.1 on your build machine. WebSphere supports deployment to both Oracle and DB2 databases. You must specify which database platform to target when executing the build script.

To build an EAR file, you must configure the \siebel\Common\build.properties file.

Configure the following settings in the build,properties file. These settings must not contain empty strings and the storepass value must be at least six characters in length.

```
alias=your keystore alias

storepass=your keystore password

companyName=your section

operatingUnit=your unit

organisation=your company

country=your country
```

These settings sign the JAR files and WAR files with the credentials that you supply (signing is needed for the front-end WAR files that are deployed using WebStart).

Running the Automated WebSphere Build Process

The \siebel\Build folder contains:

- A Windows script, JavaPrompt.vbs, which creates a command prompt that has the CLASSPATH and PATH variables configured to invoke the Banking Application build process.

- A build script, build-all.xml, which builds all the projects in the correct order and creates the EAR file in the \siebel\Build\release folder.

To create an EAR targeted to an Oracle database

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   ```
   cd Build

   buildall websphere-orcl
   ```

To create an EAR targeted to a DB2 database

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   ```
   cd Build

   buildall websphere-db2
   ```

Building WebSphere Projects Individually

If required, you can build each project individually as follows:

To build the Entitlements project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. If you have an Oracle database, enter the following at the command prompt:

   ```
   cd Entitlements

   build websphere-orcl
   ```

4. If you have a DB2 database, enter the following at the command prompt:

   ```
   cd Entitlements

   build websphere-db2
   ```

To build the Branch project

1. Navigate to the \siebel\Build directory.

2. Double-click on the JavaPrompt.vbs file.
3. If you have an Oracle database, enter the following at the command prompt:

   ```
   cd Branch

   build websphere-orcl
   ```

4. If you have a DB2 database, enter the following at the command prompt:

   ```
   cd Branch

   build websphere-db2
   ```

To build the BranchAdministratorClient project

1. Navigate to the \siebel\Build directory.
2. Double-click on the JavaPrompt.vbs file.
3. Enter the following at the command prompt:

   cd BranchAdministratorClient

   build

   The ANT build process creates a branchadmin.war WAR file, and this Web application is stored in D:\siebel\Build\staging\war.

To build the BranchClient project

1. Navigate to the \siebel\Build directory.
1. Double-click on the JavaPrompt.vbs file.
2. Enter the following at the command prompt:

   ```
   cd BranchClient

   build
   ```

   The ANT build process creates a branchteller.war WAR file, and this Web application is stored in D:\siebel\Build\staging\war.

To build the EAR file

1. Navigate to the \siebel\Build directory.
1. Double-click on the JavaPrompt.vbs file.
2. If you have an Oracle database, enter the following at the command prompt:

   ```
   cd Build

   build websphere-orcl
   ```

3. If you have a DB2 database, enter the following at the command prompt:

   ```
   cd Build
   ```

```
build websphere-db2
```

Now that you have rebuilt the EAR, you can redeploy it on your application server. See the *Siebel Retail Finance Banking Application Installation Guide* for instructions.

NOTE: The MCA and Entitlements projects are delivered precompiled and are not rebuilt during the build. The precompiled EJB file is located in the subfolders at Siebel\Build\resource.

NOTE: A large number of reflection errors are thrown during the WebSphere build process; these errors should be ignored. A full build can run for a considerable length of time—typically 30-40 minutes on a 2GHz or faster PC with 512MB RAM.