



Siebel Retail Finance Banking Application Build Process Guide

Version 2007

April 2007

Copyright © 2005, 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

1 What's New in This Release

2 Banking Application Components

Banking Application Components 7

About the Build Process 7

3 Preparing the Build Environment

Extracting the Source Code 9

 Extracting Resources for Building Branch Teller 9

 Extracting Resources for Building Internet Banking 9

Adding Java Communications API Components to the Build Process 10

Adding CTR PDF Form Generation Components to the Build Process 10

Banking Application Build Resources 10

 Build Directory Structure 11

 Class JAR files in the Build Directory 12

 Domain Layer Source Code 14

 Branch Teller Statechart Files 14

4 WebLogic Build Process

WebLogic Build Process Prerequisites 17

Signing Customized JAR and WAR Files 17

Building the Siebel Retail Finance EAR for Deployment on WebLogic 18

Building Individual Siebel Retail Finance Projects for Deployment on WebLogic 18

5 WebSphere Build Process

WebSphere Build Process Prerequisites 21

Signing Customized JAR and WAR Files 21

Building the Siebel Retail Finance EAR for Deployment on WebSphere 22

Building Individual Siebel Retail Finance Projects for Deployment on
WebSphere 23

1

What's New in This Release

What's New in Siebel Retail Finance Banking Application Build Process Guide, Version 2007

Table 1 lists changes in this version of the documentation to support version 2007 of the software.

Table 1. What's New in Siebel Retail Finance Banking Application Build Process Guide, Version 2007

Topic	Description
Banking Application Components	This topic has been updated to include Internet Banking and Web Services components.
Extracting Resources for Building Internet Banking	This topic has been added to support the Internet Banking build process.
Adding Java Communications API Components to the Build Process	This topic has been added to enable support for RS232 and IEEE 1284 peripheral devices.
Adding CTR PDF Form Generation Components to the Build Process	This topic has been added to enable support for currency transaction report (CTR) PDF form generation.
Signed JAR Files	This topic has been removed. All JAR and WAR files in the build folder must be signed after customization.
WebSphere Build Process Prerequisites	This topic has been updated.
Signing Customized JAR and WAR Files	This topic has been updated.
Building the Siebel Retail Finance EAR for Deployment on WebSphere	This topic has been updated.

2

Banking Application Components

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

- Banking Application Components
- About the Build Process

Banking Application Components

The Siebel Retail Finance Banking Application includes the following 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 Applications as a set of enterprise 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 Faces (JSF) -based front end, built using the Oracle Application Development Framework (ADF) Faces component library, for the Internet Banking product. This is deployed in the same EAR file as the server-side business logic.
- A Branch Offline Server to support a subset of transactions should the Central Server go offline. The Branch Offline Server is comprised of a lightweight RMI application and a data store.
- A Web application for MCA Services administration functionality.
- Siebel Retail Finance Web Services.

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 only 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

Preparing the Build Environment

This chapter covers preparing the build environment and includes the following topics:

- Extracting the Source Code
- Banking Application Build Resources

NOTE: Refer to the *Siebel Retail Finance System Requirements and Supported Platforms* document on SupportWeb (<http://supportweb.siebel.com>) for information regarding the supported environments, including the supported database and application server versions.

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. 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.

Extracting Resources for Building Branch Teller

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 Internet Banking

Extract the following resources to the D:\ path:

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

Adding Java Communications API Components to the Build Process

Support for peripheral devices connected to RS232 serial ports and IEEE 1284 parallel ports, for example, check readers, is provided through the Java Communications API. You must download this third party software, if peripheral devices are to be used in your Banking Application installation. This task includes the comm.jar file in the banking application when the EAR file is rebuilt. You need to install the comm.jar file on every Branch Teller client machine if you need to enable peripheral devices support but do not rebuild the EAR file to include the comm.jar file.

To add Java Communications API components to the build process

- 1 Download the Java Communications API 2.0 from <http://javashoplm.sun.com/ECom/docs/Welcome.jsp?StoreId=22&PartDetailId=7235-javacomm-2.0-spec-oth-JSpec&SiteId=JSC&TransactionId=noreg>
- 2 Extract the comm.jar file from the Java Communications API zip file.
- 3 Copy the comm.jar file to the \siebel\3rdParty\lib\comm\Win32 folder.

The comm.jar file is added to the banking application on the central server when the EAR file is re-built.

Adding CTR PDF Form Generation Components to the Build Process

The currency transaction report (CTR) functionality supports the creation of CTR PDF forms. To enable this functionality, you must download the Adobe Forms Data Format (FDF) Toolkit for Java and rebuild the *BranchClient* and *BranchAdministrator* projects.

To add CTR PDF form generation components to the build process

- 1 Download the FDF Toolkit for Java from <http://partners.adobe.com/public/developer/acrobat/fdf/topic.html>.
- 2 Extract the jFdfTk.jar file from the FDF Toolkit zip file.
- 3 When your application is accepted, follow the instructions on the Web site for downloading and installing the software.
- 4 Copy the jFdfTk.jar file to the \siebel\3rdParty\lib\fdf folder.

The jFdfTk.jar file is added to the banking application central server when the EAR file is re-built.

Banking Application Build Resources

The following topics describe the Siebel Retail Finance banking application build resources:

- Build Directory Structure
- Class JAR files in the Build Directory
- **Error! Reference source not found.**
- Domain Layer Source Code
- Branch Teller Statechart Files

Build Directory Structure

Table 2 outlines the build directory that is created when you extract the files contained in the Common Software Resources and Build Packs.

Table 2. Standard Directory Structure

Build Directory	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 <i>Siebel Retail Finance Banking Application Installation Guide</i> 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.
\BranchAdministratorClient	Contains all the front-end code to run the Branch Teller administration application, and an ANT script to compile the Web Archive (WAR) file. NOTE: Branch Teller Administrator front-end code is only shipped if the Branch Teller product has been licensed.
\BranchClient	Contains all the front-end code to run the Branch Teller client, and an ANT script to compile the WAR file. NOTE: Branch Teller front-end code is only shipped if the Branch Teller product has been licensed.
\InternetBankingClient	Contains the Internet Banking source code, JSPs, and ANT scripts to build Internet Banking. NOTE: Internet Banking front-end code is only shipped if the Internet Banking product has been licensed.

Build Directory	Description
\IBWebServiceClient	Contains the web service interface to a subset of the Internet Banking functionality. NOTE: Web Services front-end code is only shipped if the Internet Banking product has been licensed.
\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 pre-built deployable Banking Application EAR file siebel.ear for deployment, and data to populate the application database.
\Entitlements	Contains the Entitlements source code, deployment descriptors, and ANT scripts to build Entitlements.
\Offline	Contains the Offline source code, and ANT scripts to compile and build the Branch Offline Server.
\database	Contains the database dump file, and reset scripts for the database.

Class JAR files in the Build Directory

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

Module Layer Class 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 3. 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

Module Layer classes JAR file	Description
	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 Class JAR Files

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

Table 4. 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 utility classes. Some or all of the classes in this JAR file might 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.

Support Class 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 5. 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.
bfa-utils.jar	Contains utility classes.
statemachine-ext.jar	Contains extension classes for the StateSoft Statemachine framework supporting the Screen

Support classes JAR File	Description
	Orchestrator tool in the Financial Transactions WorkBench.
mca-eabmq.jar	Contains Enterprise Access Bean (EAB) and MQSeries connector helper classes.

Domain Layer Source Code

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 source code 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

The following source code JAR files may be provided, depending on the terms of your contract:

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

Branch Teller Statechart Files

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 for the Branch Teller 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

Component	Subfolders
Branch Teller Administration Client	\siebel\BranchAdministratorClient\statechart

4

WebLogic Build Process

This chapter covers building Siebel Retail Finance Modules for the WebLogic application server, with the build scripts provided.

This chapter contains the following topics:

- WebLogic Build Process Prerequisites
- Signing Customized JAR and WAR Files
- Building the Siebel Retail Finance EAR for Deployment on WebLogic
- Building Individual Siebel Retail Finance Projects for Deployment on WebLogic

WebLogic Build Process Prerequisites

The following prerequisites apply when building an EAR file for deployment on the WebLogic application server:

- The WebLogic application server must be installed on the build machine.
- The `\siebel\Common\build.properties` file must be correctly configured; ensure that the `bea.home` property points to the WebLogic installation root.

Signing Customized JAR and WAR Files

All customized JAR and WAR files must be signed before re-deployment; if the files have not been signed a WebStart error will be thrown and deployment will fail. The following settings in the `build.properties` file must be configured to enable JAR and WAR file signage. These settings must not contain empty strings and the value for the `storepass` setting must be at least six characters in length.

`alias=<keystore alias>`

`storepass=<keystore password>`

`companyName=<company>`

`operatingUnit=<operating unit>`

`organization=<organization>`

`country=<country>`

Building the Siebel Retail Finance EAR for Deployment on WebLogic

This topic covers building the Siebel Retail Finance banking application EAR file for deployment on the WebLogic application server. The following build resources are provided in the \siebel\Build folder:

- 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.

The file build-all.xml builds the projects in the following order:

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

NOTE: The precompiled Entitlements project is located in the Siebel\Build\resource folder and the build-all script does not rebuild this project. Refer to the topic Building Individual Siebel Retail Finance Projects for Deployment on WebSphere to rebuild the Entitlements project.

To build the Siebel Retail Finance EAR for deployment on WebLogic

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

```
cd Build
buildall weblogic
```

Building Individual Siebel Retail Finance Projects for Deployment on WebLogic

This topic covers building Siebel Retail Finance projects individually for deployment on the WebLogic application server.

To build the Entitlements project for deployment on WebLogic

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

```
cd Entitlements
```

```
bui l d webl ogi c
```

To build the Branch project for deployment on WebLogic

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

```
cd Branch  
bui l d webl ogi c
```

To build the BranchAdministratorClient project for deployment on WebLogic

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

```
cd BranchAdmi ni stratorCl i ent  
bui l d
```

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 for deployment on WebLogic

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

```
cd BranchCl i ent  
bui l d
```

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 Siebel Retail Finance EAR file for deployment on WebLogic

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

```
cd Bui l d  
bui l d webl ogi c
```

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 Build Process

This chapter covers building Siebel Retail Finance Modules for the WebSphere application server, with the build scripts provided.

This chapter contains the following topics:

- WebSphere Build Process Prerequisites
- Signing Customized JAR and WAR Files
- Building the Siebel Retail Finance EAR for Deployment on WebSphere
- Building Individual Siebel Retail Finance Projects for Deployment on WebSphere

WebSphere Build Process Prerequisites

The following prerequisites apply when building an EAR file for deployment on the WebSphere application server:

- The WebSphere application server must be installed on the build machine.
- The `\siebel\Common\build.properties` file must be correctly configured; ensure that the `websphere.home` property points to the WebSphere installation root. The build scripts assume that the WebSphere installation root is `D:\WAS\AppServer`.

Signing Customized JAR and WAR Files

All customized JAR and WAR files must be signed before re-deployment; if the files have not been signed a WebStart error will be thrown and deployment will fail. The following settings in the `build.properties` file must be configured to enable JAR and WAR file signage. These settings must not contain empty strings and the value for the `storepass` setting must be at least six characters in length.

`alias=<keystore alias>`

`storepass=<keystore password>`

`companyName=<company>`

`operatingUnit=<operating unit>`

`organization=<organization>`

`country=<country>`

Building the Siebel Retail Finance EAR for Deployment on WebSphere

This topic covers building the Siebel Retail Finance banking application EAR file for deployment on the WebSphere application server. The following build resources are provided in the \siebel\Build folder:

- 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.

The file `build-all.xml` builds the projects in the following order:

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

NOTE: The precompiled Entitlements project is located in the `Siebel\Build\resource` folder and the `build-all` script does not rebuild this project. Refer to the topic *Building Individual Siebel Retail Finance Projects for Deployment on WebSphere* to rebuild the Entitlements project.

To build the Siebel Retail Finance EAR for deployment on WebSphere and Oracle database

- 1 Navigate to the \siebel\Build directory.
- 2 Double-click the `JavaPrompt.vbs` file.
- 3 Enter the following at the command prompt:
`cd Build`
`buildall websphere-orcl`

To build the Siebel Retail Finance EAR for deployment on WebSphere and DB2 database

- 1 Navigate to the \siebel\Build directory.
- 2 Double-click the `JavaPrompt.vbs` file.
- 3 Enter the following at the command prompt:
`cd Build`
`buildall websphere-db2`

Building Individual Siebel Retail Finance Projects for Deployment on WebSphere

This topic covers building Oracle's Siebel Retail Finance projects individually for deployment on the WebSphere application server.

To build the Entitlements project for deployment on WebSphere

- 1 Navigate to the \siebel\Build directory.
- 2 Double-click 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 for deployment on WebSphere

- 1 Navigate to the \siebel\Build directory.
- 2 Double-click 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 for deployment on WebSphere

- 1 Navigate to the \siebel\Build directory.
- 2 Double-click 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 for deployment on WebSphere

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

```
cd BranchClient  
build
```

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

To build the EAR file for deployment on WebSphere

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

```
cd Build  
build websphere-orcl
```

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

```
cd Build  
build websphere-db2
```

This process will deploy the application to the local machine. If you need to deploy to another machine, you can follow the instructions in the *Siebel Retail Finance Banking Application Installation Guide*.

NOTE: A full build can run for a considerable length of time—typically 30-40 minutes on a 2 gigahertz (GHz) or faster PC with 512 megabyte (MB) RAM.