# Integration Guide for Oracle Billing Insight

Version 7.0

March 2016

ORACLE

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

# Contents

## Chapter 1: What's New in This Release

## Chapter 2: Using an External Authentication System

## Chapter 3: Integrating Oracle Billing Insight With Oracle BRM

# 1 What's New in This Release

## What's New in Integration Guide for Oracle Billing Insight, Version 7.0

*Integration Guide for Oracle Billing Insight* is a new guide with this release.

Table 1 lists the chapters in this guide.

Table 1. What's New in Integration Guide for Oracle Billing Insight, Version 7.0

| Topic | Description |
|---|---|
| Chapter 1, "What's New in This Release" | This chapter lists the updates to the guide for the current release of the software. |
| Chapter 2, "Using an External Authentication System" | This chapter has been moved here from *Implementation Guide for Oracle Billing Insight*. It describes how to integrate SSO, LDAP, and other authentication systems with Oracle Billing Insight. |
| Chapter 3, "Integrating Oracle Billing Insight With Oracle BRM" | This chapter describes how to integrate Oracle Billing Insight with Oracle BRM (Communications Billing and Revenue Management). |
| Chapter 4, "Configuring a Content Management Server" | This chapter has been moved here from *Implementation Guide for Oracle Billing Insight*. It describes how to configure a content management server for use with Oracle Billing Insight. |

# 2 Using an External Authentication System

This chapter covers how to configure Oracle Billing Insight to use an external authentication system, such as SSO or LDAP. It includes the following topics:

- Configuring Oracle Billing Insight to use a Single Sign-on System on page 7
- Configuring Oracle Billing Insight to use an LDAP System on page 17
- Synchronizing Single Sign-On or LDAP User Profile Information with Oracle Billing Insight on page 19
- Avoiding Clickjacking Using X-Frame-Options Security Settings on page 22
- Synchronizing User Profile Information From Oracle Billing Insight With an External Customer Relationship Management System on page 20

## Configuring Oracle Billing Insight to use a Single Sign-on System

Spring Security provides hooks for single sign-on implementation, such as Central Authentication Service (CAS), Oracle Identity and Access Manager (OIM), or OpenID. Spring Security also supports LDAP and Pre-Authentication. You can customize the Spring Security implementation for your system requirements. For more information on how to implement customized hooks, see the Spring Security documentation at

http://static.springsource.org/spring-security

Also consult your single sign-on system vender for information on integration with Spring Security.

See the following examples for integrating Oracle Billing Insight with CAS and OIM:

- "Example of Integrating Oracle Billing Insight with CAS" on page 7
- "Example of Integrating Oracle Billing Insight with OIM" on page 12

For more information about using CAS, see the documentation at

http://www.jasig.org/cas

For information about OIM, see *Oracle® Fusion Middleware Installation Guide for Oracle Identity and Access Management.*

### Example of Integrating Oracle Billing Insight with CAS

This topic gives one example of integrating Oracle Billing Insight with CAS on Oracle WebLogic. The specific steps and files required for your SSO implementation will vary. For the Assisted Service and Web Services applications, additional examples of how to integrate Oracle Billing Insight with CAS are located in the following directory:

- **UNIX.** *EDX_HOME*/samples/sso/cas
- **Windows.** *EDX_HOME*\samples\sso\cas

### *To integrate Oracle Billing Insight with CAS on Oracle WebLogic*

1. Verify that the following requirements have been met:
   - The CAS SSO system is set up.
   - The Oracle Billing Insight installation and configuration is complete.
2. Import the SSO user in the Oracle Billing Insight database.
3. Access your LDAP server, and add a user and group for the Self-Service and Assisted Service applications.
4. Download the libraries required to support integrating your SSO implementation with the CAS Server 3.4.11 to the *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0.ear\lib directory. These are open source resources and can be accessed from any public maven repository, CAS Web site, and so on. These can include files such as:
   - cas-client-core-3.1.10.jar
   - opensaml-1.1b.jar
   - spring-security-cas-3.1.0.RELEASE.jar
   - xmlsec-1.3.0.jar
   - httpcore-4.1.jar (For the Assisted Service application only)
   - Add the JAR file paths to the MANIFEST.MF file found in the *EDX_HOME*\J2EEApps\selfservice\webcear\APP-INF\lib\xma-classpath.jar\META-INF\ directory, such as:

     ```
     ../../lib/xmlsec-1.3.0.jar ../../lib/opensaml-1.1b.jar ../../lib/spring-
     security-cas-3.1.0.RELEASE.jar ../../lib/cas-client-core-3.1.10.jar
     ```
5. If there are any conflicts among third party library versions, then add the following statement to the weblogic-application.xml file found in the *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0.ear\META-INF\ directory:

   ```
   <package-name>{jar path}</package-name>
   ```

   For example:

   ```
   <package-name>org.opensaml.*</package-name>
   ```
6. Add the following content to the web.xml file found in the *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0.ear\selfservice-web-1.0-SNAPSHOT.war\WEB-INF directory:

   ```
   <filter><filter-name>characterEncodingFilter</filter-name>
   <filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-
   class><init-param><param-name>encoding</param-name>
   <param-value>UTF-8</param-value></init-param>
   ```

```
</filter><filter-mapping><filter-name>characterEncodingFilter</filter-
name><url-pattern>/*</url-pattern></filter-mapping>
<listener><listener-
class>org.jasig.cas.client.session.SingleSignOutHttpSessionListener</listener-
class></listener>
```

7 Edit the session time-out filter in the web.xml file to specify j_spring_cas_security_logout as the parameter value:

```
<param-value>(/j_spring_security_check|/nonAuth/logout.action|/
j_spring_cas_security_logout).*</param-value>
```

8 Edit the spring-security.xml file found in the *EDX_HOME*\config\security\selfservice\. Update the following:

- Add a CAS entry point to the security:http statement:

```
<security:http entry-point-ref="casEntryPoint" access-decision-manager-
ref="accessDecisionManager">
```

- Add a CAS custom filter to the security:http statement:

```
<security:custom-filter position="CAS_FILTER" ref="casFilter" />
```

- Add beans serviceProperties, casFilter, and casEntryPoint, specifying the server and ports for your implementation:

```
<bean id="serviceProperties"
   class="org.springframework.security.cas.ServiceProperties">
   <property name="service"> <value>https://
   your_server_name:your_application_port/SelfService/
   j_spring_cas_security_check </value> </property>
   <property name="sendRenew" value="false"/>
</bean>
<bean id="casFilter"
   class="org.springframework.security.cas.web.CasAuthenticationFilter">
   <property name="authenticationManager" ref="authenticationManager"/>
</bean>
<bean id="casEntryPoint"
   class="org.springframework.security.cas.web.CasAuthenticationEntryPoint">
   <property name="loginUrl" value="https://your_server_name:8443/cas-
   server-webapp-3.4.11/login"/> <property name="serviceProperties"
   ref="serviceProperties"/>
</bean>
```

- Set the service value to the URL of your Oracle Billing Insight application. Set the login URL value to the URL of your CAS server login.
- Update the authentication provider:

```
<security:authentication-provider ref="casAuthenticationProvider" />
```

- Add the casAuthenticationProvider bean:

```
<bean id="casAuthenticationProvider"
   class="org.springframework.security.cas.authentication.CasAuthentication
   Provider"><property name="authenticationUserDetailsService">
```

```
    <bean class="org.springframework.security.core.userdetails.UserDetails
    ByNameServiceWrapper"><constructor-arg ref="userDetailsService" />
    </bean></property><property name="serviceProperties"
    ref="serviceProperties" /><property name="ticketValidator">
    <bean class="org.jasig.cas.client.validation.Saml11TicketValidator">
    <constructor-arg index="0" value="https://your_server_name:8443/
    cas-server-webapp-3.4.11" /></bean></property><property
    name="key"value="an_id_for_this_auth_provider_only"/>
</bean>
```

- Add a custom filter for single logout:

```
<security:custom-filter ref="requestSingleLogoutFilter"
    before="LOGOUT_FILTER"/>
<security:custom-filter ref="singleLogoutFilter" before="CAS_FILTER"/>
```

- Remove the following:

```
<security:logout logout-success-url="/nonAuth/logout.action"/>
<security:form-login login-page="/nonAuth/login.action" default-target-
url="/reporting/reportStart.action" authentication-failure-url="/nonAuth/
login.action?login_error=1"/>
```

- Add the filter definitions:

```
<!-- This filter handles a Single Logout Request from the CAS Server -->
<bean id="singleLogoutFilter"
class="org.jasig.cas.client.session.SingleSignOutFilter"/>
<bean id="requestSingleLogoutFilter"
class="org.springframework.security.web.authentication.logout.LogoutFilter">
<constructor-arg value="https://your_server_name:8443/cas-server-webapp-
3.4.11/logout?service=https://your_server_name:your_server_port/
selfservice"/><constructor-arg>
<bean class="org.springframework.security.web.authentication.logout.Security
ContextLogoutHandler"/></constructor-arg>
<property name="filterProcessesUrl" value="/j_spring_cas_security_logout"/
></bean>
```

- Set the URL of your CAS server.

9 Install the certification file for both the SSO system and Oracle Billing Insight if the SSL connection is enabled.

10 Configure Oracle Billing Insight to work in SSO enabled mode, which turns on and off the appropriate UI features. Make the following edits in the globalConfig.xma.xml file, found in the *EDX_HOME*/xma/config/modules folder:

- Turn on the global configuration flags for SSO. Change the following property values to true:

```
<property name="ebillingSingleSignOnEnabled"><value>false</value></property>
<property name="csrSingleSignOnEnabled"><value>false</value></property>
```

- If you use Web Services, then set the following SSO flag to true:

```
<property name="ebillingWebServiceSingleSignOnEnabled"><value>false</
value></property>
```

- Add a URL for single sign-out. The default value for CAS is j_spring_cas_security_logout:

  ```
  <property name="singleSignOutUrl"><value>j_spring_cas_security_logout</
  value></property>
  ```

- Add home page links for the Self-Service and Assisted Service applications:

  ```
  <property name="ebillingHomePageUrl"><value>http://localhost:7001/portlet</
  value></property>
  <property name="csrHomePageUrl"><value>http://localhost:7001/portlet</
  value></property>
  ```

11 Run the automated configuration script:

a Go to the following directory:

- **UNIX.** *EDX_HOME*/samples/sso/CAS
- **Windows.** *EDX_HOME*\samples\sso\CAS

In the directory, *EDX_HOME* is the location where you installed Oracle Billing Insight.

b Update the spring-security-sso.properties files found in each of the following directories:

UNIX:

- *EDX_HOME*/samples/sso/CAS/conf/selfservice
- *EDX_HOME*/samples/sso/CAS/conf/csr
- *EDX_HOME*/samples/sso/CAS/conf/rs

Windows: *EDX_HOME*\samples\sso\CAS

- *EDX_HOME*\samples\sso\CAS\conf\selfservice
- *EDX_HOME*\samples\sso\CAS\conf\csr
- *EDX_HOME*\samples\sso\CAS\conf\rs

In the property files, set the following parameters for your implementation.

f

| Property | Description |
|---|---|
| protocol | The connection protocol, such as HTTPS. |
| hostname | The name of the server host where the Self-Service application resides. |
| application_name | The name of your Self-Service application. |
| port | The port number for accessing the Self-Service application. |
| sso_protocol | The name of the single sign-on protocol, such as HTTPS. |
| sso_server_hostname | The name of the server host of the single sign-on system. |
| sso_application_name | The name of the single sign-on application. |
| sso_application_port | The single sign-on port number. |

| Property | Description |
|---|---|
| sso_login_url | The login URL of the single sign-on application. |
| sso_logout_url | The logout URL of the single sign-on application. |

c. Enter Ant. The script automatically runs Options 5 - 10.

12. Copy the following EAR files from the *EDX_HOME*/`samples/sso/CAS/target` directory to the *EDX_HOME*/J2EEApp folder, and then redeploy the files:
    - selfservice-weblogic-7.0.ear
    - rs-weblogic-7.0.ear
    - csr-app-7.0.ear

## Example of Integrating Oracle Billing Insight with OIM

This topic provides one example of integrating Oracle Billing Insight with Oracle Identity and Access Manager (OIM) on Oracle WebLogic. The specific steps and files required for your SSO implementation will vary.

### *To integrate Oracle Billing Insight with OIM on Oracle WebLogic*

1. Verify that the following requirements have been met:
    - The OIM SSO system is set up.
    - The Oracle Billing Insight installation and configuration is complete.
2. Open the globalConfig.xma.xml file, located in the following directory:
    - **UNIX.** *EDX_HOME*/`xma/config/modules`
    - **Windows.** *EDX_HOME*`\xma\config\modules`

    Set the Set ebillingSingleSignOnEnabled property to True. Set the singleSignOutUrl property to j_spring_security_logout.
3. Import the SSO user in the Oracle Billing Insight database.
4. Access Oracle Directory Services Manager (ODSM), and add a user group for the Self-Service and Assisted Service applications, and assign users to groups.
5. Edit the web.xml and weblogic.xml files in the Oracle Billing Insight EAR file, located in the following directory:
    - **UNIX.** *EDX_HOME*/`J2EEApps/selfservice/weblogic/selfservice-weblogic-7.0.ear/selfservice-1.0-SNAPSHOT.war/WEB-INF`
    - **Windows.** *EDX_HOME*`\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.0.ear\selfservice-1.0-SNAPSHOT.war\WEB-INF`

    a. Add the following content to the web.xml file:

```
<login-config><auth-method>CLIENT-CERT</auth-method><realm-name>myRealm</
realm-name></login-config>
<security-role><role-name>AUTH_USER</role-name></security-role>
<security-constraint>
<web-resource-collection>
<web-resource-name>All areas</web-resource-name>
<url-pattern>/*</url-pattern>
</web-resource-collection>
<auth-constraint><role-name>AUTH_USER</role-name></auth-constraint>
</security-constraint>
```

b. Add the following content to the weblogic.xml file, where the value of principal-name is the group to be created in your LDAP server:

```
<security-role-assignment>
<principal-name>EBilling_USERS</principal-name>
<principal-name>CSR_USERS</principal-name>
<role-name>AUTH_USER</role-name>
</security-role-assignment>
```

6. Edit the Spring Security configuration file, spring-security.xml, located in the following directory:

   - **UNIX.** *EDX_HOME*/config/security/selfservice
   - **Windows.** *EDX_HOME*\config\security\selfservice

   Sample configuration files can be found in the following directories:

   - **UNIX.** *EDX_HOME*/samples/sso/OAM/conf/selfservice and *EDX_HOME*/samples/sso/OAM/conf/csr
   - **Windows.** *EDX_HOME*\samples\sso\OAM\conf\selfservice and *EDX_HOME*\samples\sso\OAM\conf\csr

7. A samples JAR file, ebilling-sso-oam-1.0-SNAPSHOT.jar, is located in the following directory:

   - **UNIX.** EDX_HOME/samples/sso/OAM/xma
   - **Windows.** EDX_HOME\samples\sso\OAM\xma

   Move the file into the following directory:

   - **UNIX.** EDX_HOME/J2EEApps/selfsevice/weblogic/selfservice-weblogic-7.0.ear/xma
   - **Windows.** EDX_HOME\J2EEApps\selfsevice\weblogic\selfservice-weblogic-7.0.ear\xma

   Add the jar file path to the MANIFEST.MF file, located in the following directory:

   - **UNIX.** EDX_HOME/J2EEApps/selfsevice/weblogic/selfservice-weblogic-7.0.ear/APP-INF/lib/xma-classpath.jar/META-INF
   - **Windows.** EDX_HOME\J2EEApps\selfsevice\weblogic\selfservice-weblogic-7.0.ear\APP-INF\lib\xma-classpath.jar\META-INF

   Samples provide minimal functionality only. Additional requirements can be implemented by extending Spring Security.

8. Change the following beans:

- Replace "FORM_LOGIN_FILTER" with "PRE_AUTH_FILTER" in the following section:

```
<security:http entry-point-ref="EBillingEntryPoint"
   access-decision-manager-ref="accessDecisionManager">
   <security:custom-filter position="PRE_AUTH_FILTER"
   ref="j2eePreAuthFilter" />
</security:http>
```

- Define the "j2eePreAuthFilter" bean as follows:

```
<bean id="j2eePreAuthFilter"
class="com.edocs.common.security.authenticate.sso.EBillingPreAuthenticatedPr
ocessingFilter">
<property name="authenticationManager" ref="authenticationManager"/>
<property name="authenticationDetailsSource">
<bean
class="org.springframework.security.web.authentication.preauth.j2ee.J2eeBase
dPreAuthenticatedWebAuthenticationDetailsSource">
<property name="mappableRolesRetriever"
><bean
class="org.springframework.security.web.authentication.preauth.j2ee.WebXmlMa
ppableAttributesRetriever" />
</property> <property name="userRoles2GrantedAuthoritiesMapper">
bean
class="org.springframework.security.core.authority.mapping.SimpleAttributes2
GrantedAuthoritiesMapper"><property name="convertAttributeToUpperCase"
value="true"/></bean></property> </bean> </property> <property
name="ebllingAuthenticationDetailsSource"><bean
class="com.edocs.common.security.authenticate.EBillingWebAuthenticationDetai
lsSource" /> </property></bean>
```

- Replace preAuthenticatedAuthenticationProvider with daoAuthenticationProvider at each occurrence in the following section:

```
<security:authentication-manager alias="authenticationManager">
<security:authentication-provider ref="impersonateAuthenticationProvider"/>
<security:authentication-provider
ref='preAuthenticatedAuthenticationProvider'/>
</security:authentication-manager>
<bean id="preAuthenticatedAuthenticationProvider"
class="org.springframework.security.web.authentication.preauth.PreAuthentica
tedAuthenticationProvider">
<property name="preAuthenticatedUserDetailsService"ref="userDetailsService"/
></bean>
```

- Change the bean ID userDetailsService to "com.edocs.common.security.authenticate.EBillingPreAuthUserDetailsService” in the following section:

```
<bean id="userDetailsService"
class="com.edocs.common.security.authenticate.EBillingUserDetailsService" />
```

- Define the logout URL in <security:logout>:

```
<security:http entry-point-ref="EBillingEntryPoint"
   access-decision-manager-ref="accessDecisionManager">
   security:logout logout-url="/j_spring_security_logout"
   logout-success-url="${logout_link}" invalidate-session="true" />
</security:http>
```

9 Configure the Oracle WebLogic domain for the Self-Service application to support OIM single sign-on:

a Add the OIM dependency into the application domain. Copy the oamAuthnProvider.jar file from the *OAM_HOME* directory to the following directory:

- **UNIX.** `*WL_HOME*/server/lib/mbeantypes`
- **Windows.** `*WL_HOME*\server\lib\mbeantypes`

b Add providers to the security realms in the domain. In the Oracle Weblogic console, click Home, Summary of Security Realms, myrealm, and then Providers. Add OAMIdentityAsserter, OIDAuthenticator in Configure OAMIdentityAsserter and OIDAuthenticator.

c Reorder the providers. On the console, click Home, Summary of Security Realms, myrealm, and then Providers. Reorder the providers as follows:

- OAMIdentityAsserter
- OIDAuthenticator
- DefaultAuthenticator
- DefaultIdentityAsserter

10 Install and configure Oracle HTTP Server and a WebGate:

a Install and configure Oracle HTTP Server. For information about implementing Oracle HTTP Server (OHS), see *Oracle® Fusion Middleware Installing and Configuring Oracle HTTP Server.*

b In the ${INSTANCE_HOME}/config/OHS/${YOUR OHS INSTANCE NAME}/moduleconf, create a file called ebilling.conf, with the following content:

```
NameVirtualHost *:${your port}
<VirtualHost *:${your port}>
   #  ServerName admin.mycompany.com:80
   #  ServerAdmin you@your.address
   RewriteEngine On
   RewriteOptions inherit
   #  RewriteRule ^/console/jsp/common/logout.jsp /oamsso/logout.html [PT]
   #  RewriteRule ^/em/targetauth/emaslogout.jsp /oamsso/logout.html [PT]
   # Admin Server and EM
   <Location/selfservice>
     SetHandler weblogic-handler
     WebLogicHost ${your SelfService host name}
     WeblogicPort ${your SelfService port}
   </Location>
</VirtualHost>
```

c Restart OHS.

11 Run the automated configuration script:

a Go to the following directory:

- **UNIX.** *EDX_HOME*/samples/sso/OAM/build
- **Windows.** *EDX_HOME*\samples\sso\OAM\build

b In the build.properties file, set the following parameters for your implementation.

| Property | Description |
|---|---|
| edx_home | The location where you installed Oracle Billing Insight. |
| wls_home | The location where Oracle WebLogic is installed. |
| wls_user | The name of the Oracle WebLogic user. |
| wls_password | The password of the Oracle WebLogic user. |
| app_host | The name of your application host server. |
| ebilling_domain_name | Your Self-Service application domain name. |
| csr_domain_name | Your Assisted Service application domain name. |
| ebilling_port | The port number of your Self-Service application. |
| csr_port | The port number of your Assisted Service application. |
| rs_port | The port number of your RESTful Web services. |

c Enter Ant. The script automatically runs Options 2 - 9.

12 Update the spring-security-sso.properties file, found in each of the following folders:

- *EDX_HOME*/samples/sso/OAM/conf/selfservice
- *EDX_HOME*/samples/sso/OAM/conf/csr
- *EDX_HOME*/samples/sso/OAM/conf/rs

In each file, set the logout_link property as follows, specifying the OIM and OHS server and ports for your implementation:

```
logout_link=http://your_OAM_server_name:your_OAM_server_port/oam/server/
logout?end_url=http://your_OHS_server_name:your_OHS_server_port/csr
```

13 Copy the following EAR files from the *EDX_HOME*/samples/sso/OAM/build/target directory to the *EDX_HOME*/J2EEApp folder, and then redeploy the files:

- selfservice-weblogic-7.0.ear
- rs-weblogic-7.0.ear
- csr-app-7.0.ear

# Configuring Oracle Billing Insight to use an LDAP System

You can configure Oracle Billing Insight for access using an LDAP user identity store. Spring Security supports LDAP authentication. Spring Security provides hooks for LDAP. You can customize the Spring Security implementation for your system requirements. For more information on how to implement customized hooks, see the Spring Security documentation at

http://static.springsource.org/spring-security

Also consult your LDAP system vender for information on integration with Spring Security.

Using an LDAP system replaces the Oracle Billing Insight user management functionality with the external system's functionality. You must customize your LDAP system to implement any Oracle Billing Insight user management features you require.

## Example of Integrating Oracle Billing Insight with an LDAP System

This topic gives one example of integrating Oracle Billing Insight with an LDAP system on Oracle WebLogic. The specific steps and files required for your LDAP implementation will vary.

### *To configure Oracle Billing Insight to use an LDAP system*

1 Add the LDAP server URL to the spring-security.xml file, found in the *EDX_HOME/* \config\security\selfservice directory, where *EDX_HOME* is the directory where you installed Oracle Billing Insight:

```
<security:ldap-server url="ldap://your_server_name:3060" />
```

2 In the same file, change the authentication-provider to ldapAuthProvider:

```
<security:authentication-manager alias="authenticationManager">
<!--<security:authentication-provider ref="daoAuthenticationProvider" />-->
<security:authentication-provider ref="ldapAuthProvider"/>
</security:authentication-manager>
```

3 Add the contextSource bean and set the LDAP URL, user dn, and password:

```
<bean id="contextSource"
class="org.springframework.security.ldap.DefaultSpringSecurityContextSource">
<constructor-arg value="ldap://your_server_name:3060"/>
<property name="userDn"
value="cn=orcladmin,cn=Users,dc=us,dc=oracle,dc=com"/><property name="password"
value="Welcome1"/></bean>
```

4 Add the ldapAuthProvider bean, and configure the BindAuthenticator and DefaultLdapAuthoritiesPopulator. Implement the userDetailsContextMapper bean with your own class, for example:

```
<bean id="ldapAuthProvider"
class="org.springframework.security.ldap.authentication.LdapAuthenticationProvi
der">
<constructor-arg>
```

```
<bean
class="org.springframework.security.ldap.authentication.BindAuthenticator">
<constructor-arg ref="contextSource" /><property name="userSearch"><bean
id="userSearch"
class="org.springframework.security.ldap.search.FilterBasedLdapUserSearch">
<constructor-arg index="0" value="cn=Users, dc=us,dc=oracle,dc=com"/>
<constructor-arg index="1" value="(cn={0})"/>
<constructor-arg index="2" ref="contextSource" />
</bean></property></bean></constructor-arg><constructor-arg>
<bean
class="org.springframework.security.ldap.userdetails.DefaultLdapAuthoritiesPopu
lator">
<constructor-arg ref="contextSource" />
<constructor-arg value="cn=Groups, dc=us,dc=oracle,dc=com" />
<property name="groupSearchFilter" value="(uniquemember={0})"/>
<property name="rolePrefix" value=""/>
<property name="searchSubtree" value="true"/>
<property name="convertToUpperCase" value="true"/>
</bean>
</constructor-arg>
<property name="userDetailsContextMapper" ref="EBillingLdapUserDetailsMapper"/>
</bean>
<bean id="EBillingLdapUserDetailsMapper"
class="com.edocs.common.security.authenticate.ldap.EBillingLdapUserDetailsMappe
r">
<property name="userDetailsService"><ref bean="userDetailsService" />
</property>
</bean>
```

5. Implement the UserDetailsContextMapper interface to map the LDAP context to Oracle Billing Insight user objects. The following example code represents a portion of such an implementation:

```
public class EBillingLdapUserDetailsMapper extends LdapUserDetailsMapper {
private EBillingUserDetailsService userDetailsService;public
EBillingUserDetailsService getUserDetailsService() {return userDetailsService;}

public void setUserDetailsService(EBillingUserDetailsService userDetailsService)
{this.userDetailsService = userDetailsService;}

public UserDetails mapUserFromContext(DirContextOperations ctx, String username,
Collection<? extends GrantedAuthority> authorities) {return
userDetailsService.loadUserByUsername(username);}}
```

The creation of the UserDetails object is controlled by the provider's UserDetailsContextMapper implementation, which is responsible for mapping user objects to and from LDAP context data:

```
public interface UserDetailsContextMapper {UserDetails
mapUserFromContext(DirContextOperations ctx, String username, Collection<?
extends GrantedAuthority> authorities);void mapUserToContext(UserDetails user,
DirContextAdapter ctx);}
```

# Synchronizing Single Sign-On or LDAP User Profile Information with Oracle Billing Insight

You must synchronize key user information between the external identity management system (the system of record for users) and Oracle Billing Insight. You can synchronize in real-time using a set of Web Services or in batch using Command Center jobs.

The LoadExternalB2B, LoadExternalB2C, and LoadExternalCSR Command Center jobs synchronize users into Oracle Billing Insight using an input file. For details on configuring and running the load jobs appropriate for your implementation, see *Administration Guide for Oracle Billing Insight*.

Table 2 shows the required and optional user information when synchronizing an external identity system with Oracle Billing Insight.

Table 2. Required and Optional User Information When Synchronizing From an SSO or LDAP

| User Type | Required Information | Optional Information |
|---|---|---|
| Business | Required information for business user updates:<br>■ User ID<br>■ First Name<br>■ Last Name<br>■ Role<br>■ Email Address<br>■ Company ID | Optional information for business user updates:<br>■ Preferred Language<br>■ Mobile Phone Number for SMS<br>■ Mobile Carrier for SMS<br>■ Brand (To determine which CSS file to use for the UI look and feel) |

Table 2. Required and Optional User Information When Synchronizing From an SSO or LDAP

| User Type | Required Information | Optional Information |
|---|---|---|
| Consumer | Required information for consumer user updates:<br>■ User ID<br>■ First Name Last Name<br>■ Role<br>■ Email Address<br>■ Biller ID<br>■ Billing Account Number | Optional information for consumer user updates:<br>■ Preferred Language<br>■ Brand (To determine which CSS file to use for the UI look and feel)<br>■ Mobile Phone Number for SMS<br>■ Mobile Carrier for SMS |
| Assisted Service | Required information for Assisted Service user updates:<br>■ User ID<br>■ First Name Last Name<br>■ Role<br>■ Email Address | Optional information for Assisted Service user updates:<br>■ Preferred Language |

# Synchronizing User Profile Information From Oracle Billing Insight With an External Customer Relationship Management System

You can customize your implementation of Oracle Billing Insight to automatically update an external Customer Relationship Management (CRM) system when a user modifies profile information.

### *To update an external CRM system with user profile information*

1. Customize the IUserProfileWSConnector API to update your external CRM when a user updates user profile data in Oracle Billing Insight.
2. Send a confirmation to Oracle Billing Insight when the data has been received. Also mark the request complete in Oracle Billing Insight, returning an external ID and status.
3. Create a pipe-delimited TXT data input file in the following format.

| Name | Required | Description |
|---|---|---|
| User ID | Yes | The user’s identifier |
| First Name | Yes | The user’s first name |

| Name | Required | Description |
|---|---|---|
| Middle Name | No | The user's middle name |
| Last Name | Yes | The user's last name |
| Mailing Address Line 1 | No | The user's mailing address line 1 |
| Mailing Address Line 2 | No | The user's mailing address line 2 |
| Mailing Address Line 3 | No | The user's mailing address line 3 |
| City | No | The user's city |
| State | No | The user's state |
| Zip Code | No | The user's Zip Code |
| Home Number | No | The user's home phone number |
| Mobile Number | No | The user's mobile phone number |
| Mobile Carrier | No | The user's mobile service provider |
| Email Address | Yes | The user's email address |
| Paper On Off Selection | No | The user selects to have paper delivery turned on or off |
| Language Choice | No | The user selects a default language choice |
| External User ID | No | For use with a single sign-on system that uniquely identifies the user |

## Implementing the IUserProfileWSConnector API

The com.edocs.common.api.webservice.connector.IUserProfileWSConnector API lets you update a customer relationship management (CRM) application with updates to user profiles.

When integrating Oracle Billing Insight with other applications, such as a CRM or single sign-on where user profile information is owned by the external application, but the user updates their profile in the Self-Service application, the data must be synchronized. When a user updates his or her user profile, the Self-Service application calls method IUserProfileWSConnector.updateUserProfile(UserProfileWSBean userProfile) method, which you can implement to send the information back the the external system. For details about how user authorization works, see

### *To implement the com.edocs.common.api.webservice.connector.IuserProfileWSConnector API*

- Update the webservice.xma.xml file, located in the following directory:
  - **UNIX.** *EDX_HOME*/xma/config/modules/webservice
  - **Windows.** *EDX_HOME*/xma/config/modules/webservice

  For example, if the implementation class is customerpath.CustomerUserProfileWSConnector, then update the IUserProfileWSConnector bean in the webservice.xma.xml file:

```
<bean id="IUserProfileWSConnector"
    class="com.edocs.common.webservice.ws.connector.
    DefaultUserProfileWSConnector">
</bean>
```

Change the XML file as follows:

```
<bean id="IUserProfileWSConnector"
    class="customerpath.CustermertUserProfileWSConnector">
</bean>
```

# Avoiding Clickjacking Using X-Frame-Options Security Settings

Oracle Billing Insight uses X-Frame-Options HTTP response headers to avoid UI redress attacks, called clickjacking, by preventing application content from being embedded into other sites. On a clickjacked page, attackers can load and embedded any Oracle Billing Insight page over the external site page in a transparent frame, tricking Oracle Billing Insight users into performing actions that could enable the attacker to steal payment account information, for example.

Oracle Billing Insight uses the SAMEORIGIN X-Frame-Option value, which displays the page in a frame as long as the site including it in a frame is the same as the one serving the page. This prevents Oracle Billing Insight content from being embedded into other sites, and is recommended by OWASP.

If you have implemented your own integration from an external system, you can follow the steps in this topic to configure the Self-Service and Assisted Service applications to allow specific origin using the ALLOW-FROM URI response header value, and prevent Oracle Billing Insight from being embedded by any outside pages.

**CAUTION:** If you attempt to access and embed the Self-Service or Assisted Service applications from an external system (which is not same origin), you risk clickjacking from cross-site scripting.

The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page in a <frame>, <iframe> or <object>. There are three possible values for X-Frame-Options:

- **SAMEORIGIN.** The page can be displayed in a frame on the same origin only.
- **ALLOW-FROM *URI*.** The page can only be displayed in a frame on an origin specified as the URI.
- **DENY.** The page cannot be displayed in a frame from any site attempts (from other sites or the same site).

There can also be limitations on browser compatibilities. If you are using an old browser, you may also need to configure the Assisted Service application to avoid clickjacking. For more information about browser issues, see

https://developer.mozilla.org/en-US/docs/Web/HTTP/X-Frame-Options

### *To configure Oracle Billing Insight applications to avoid clickjacking when accessing from external sites*

1 For the Self-Service and Assisted Service applications:

a Open the globalConfig.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules
- **Windows.** *EDX_HOME*\xma\config\modules

b In the following code, change the X-Frame-Options value from SAMEORIGIN to ALLOW-FROM $URL:

```
-<property name="additionalHttpResponseHeaders">
   -<props>
      <prop key="X-Frame-Options">SAMEORIGIN</prop>
   </props>
</property>
-<property name="csrAdditionalHttpResponseHeaders">
   -<props>
      <prop key="X-Frame-Options">SAMEORIGIN</prop>
   </props>
</property>
```

2 For accessing the Assisted Service application in old browsers that do not support X-Frame Options:

a Open the the _assets\templates\security.jsp file found in the csr-web-1.0-SNAPSHOT.war file.

b Edit the if (self === top) line, specifying the external site, as in:

```
if (self === top || top.location.href === 'www.example.com')
```

This line is found in the following section within the JSP file:

```
<style id="antiClickjack">
body {
 display: none !important;
}
</style>
<script type="text/javascript">
   if (self === top) {
      var antiClickjack = document.getElementById("antiClickjack");
       antiClickjack.parentNode.removeChild(antiClickjack);
    else {
      top.location = self.location;
}
```

# 3 Integrating Oracle Billing Insight With Oracle BRM

This chapter describes how to integrate Oracle Billing Insight with Oracle BRM (Communications Billing and Revenue Management). It includes the following topics:

## Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight

The process of setting up and integrating Oracle Billing Insight with Oracle BRM (Communications Billing and Revenue Management) requires that you install and configure required software, configure mapping tables, and load existing data from Oracle BRM into Oracle Billing Insight.

To set up and integrate Oracle BRM with Oracle Billing Insight, perform the following tasks:

1. Verify that your Oracle BRM data conforms to the requirements of Oracle Billing Insight described in “Oracle BRM Data Assumptions” on page 27.
2. Installing and Configuring BRM Provider on page 29. For more information about the integration tools, see About Using BRM Provider and BRM AQAdapter to Load Oracle BRM Data on page 27
3. Installing and Configuring BRM AQAdapter on page 35

4. Configuring Mapping Tables on page 39
5. Loading Oracle BRM Master Data to Oracle Billing Insight on page 41
6. Provisioning Oracle BRM Account Data in Batch to Oracle Billing Insight on page 42
7. Loading the Oracle BRM Postpaid Data to Oracle Billing Insight on page 44
8. Loading the Oracle BRM Unbilled (CDR) Data to Oracle Billing Insight on page 45
9. Loading the Oracle BRM Prepaid Data to Oracle Billing Insight on page 46

# Process of Loading Data from Oracle BRM in a Live Production Environment

Once you have completed the initial process of integrating Oracle BRM with Oracle Billing Insight, you perform the following tasks to continuously load and synchronize data from Oracle BRM to Oracle Billing Insight in a live production environment:

- Synchronize new or updated Oracle BRM account data in real-time using RESTful Web services with BRM AQAdapter. You can provision account contact information, payment accounts for credit card cards, and payment transactions in real-time using RESTful Web services. For more information, see Provisioning Oracle BRM Account Data in Real-Time to Oracle Billing Insight Using BRM AQAdapter on page 43.
- Run the HierarchyCopy job at beginning of the each new billing period. For details, see *Administration Guide for Oracle Billing Insight*.
- Schedule BRM Provider to initiate loading postpaid data from Oracle BRM to Oracle Billing Insight whenever the bill utility is run in Oracle BRM. Once configured, the pin_bill_day utility triggers the custom billRun event, and BRM AQAdapter processes the loading of postpay billing data from Oracle BRM to Oracle Billing Insight. (You can schedule BRM Provider to automatically initiate the postpaid data load while configuring BRM Provider. For details, see "Installing and Configuring BRM Provider" on page 29.)
- Schedule unbilled (CDR) and prepay data (including last top up, current balance, expiry data, CDR data, and so on) to be loaded by BRM Provider in configurable time periods. The default time interval is every 15 minutes. For details, see "Scheduling Unbilled and Prepaid Data Loading" on page 46.
- If there are changes to the Oracle BRM bill item configuration, then you must reconfigure the mapping tables in the pre-staging schema to reflect the changes. For details, see"Configuring Mapping Tables" on page 39.
- If there are changes to master data in Oracle BRM, then you must load the new data using the master data load ODI process. Master data includes payment types, adjustment types, charge type and sub charge type, product and sub-product information, tariff information, unit type and currency type. For details, see "Loading Oracle BRM Master Data to Oracle Billing Insight" on page 41.

# Oracle BRM Data Assumptions

The following data and data relationships in Oracle BRM are required for integrating Oracle BRM with Oracle Billing Insight. Due to the flexible nature of Oracle BRM, it is not possible to list all the possible requirements for integration.

- Billing accounts for the same company must have the same company value. Additionally:
  - The top level account of hierarchy must have the account contact information
  - The company contact is required information for extracting the company object.
  - The integration uses the canon_company column of the Oracle BRM account_nameinfo_t table for the company identity, so all accounts for the one company must have the same value in this field.
- One service can be associated with one number device only. For example, if a user purchases telephone service, only one number-device can be associated to this service at a time. This allows Oracle Billing Insight to retrieve the subtotal of a device at the item level.
- The integration uses items to differentiate charge types. One item must contain only one type of usage charge. Different types of charges must be broken down into different items, if they are currently aggregated in Oracle BRM. For example, telephone, SMS, and data usage charges must be separate items.
- An account can have multiple services, and each service or combination of services can have one billinfo (bill unit) value. Oracle Billing Insight supports one billinfo for one account, and no billinfo at the service level. The billinfo value can be prepay or postpay. The billinfo value can associate with a business profile, and Oracle Billing Insight assumes billinfo is prepay if the associated business profile has a key/value pair of prepaid/yes.
- Add a description for the device when associating it to a service. Oracle Billing Insight uses the the DESCR column of the DEVICE_T Oracle BRM table as the contact name for the device.
- Make sure the billing contact information, address, city, state, canon_countr, and zip code values are not null in the account_nameinfo_t table. This is required information in Oracle Billing Insight.

# About Using BRM Provider and BRM AQAdapter to Load Oracle BRM Data

You use the BRM Provider and BRM AQAdapter tools to integrate Oracle BRM with Oracle Billing Insight and to load and synchronize data in a live production environment.

## About BRM Provider

BRM Provider consists of a set of Oracle Data Integrator (ODI) processes and a pre-staging schema. The set of packages, load plans, and scenarios load billing data from the Oracle BRM server into the Oracle Billing Insight database. The processes first move Oracle BRM data to a pre-staging area, then transforms and loads the data into Oracle Billing Insight staging, and then moves it into the Oracle Billing Insight production tables.

The pre-staging schema contains over thirty BRM tables, a database link to connect to the Oracle BRM database, and data loading control tables. For details, see "Integration Extension Points" on page 48.

There are four types of billing-related data in Oracle Billing Insight, and each type of data has a dedicated extract and load process for loading Oracle BRM data:

- Master
- Postpaid
- Unbilled
- Prepaid

Each extract and load process from Oracle BRM to Oracle Billing Insight is based on a specific start timestamp (inclusive) and end timestamp (exclusive). Only one data load process is allowed at a time.

You can use one of the following methods to extract the billing data from Oracle BRM using BRM Provider:

- Directly from the Oracle BRM server, recommended for a low data volume server.
- From a backup server replicated using Oracle GoldenGate, recommended for a high data volume and performance-bound Oracle BRM server.

## BRM AQAdapter

BRM AQAdapter processes the following real-time events generated by Oracle BRM, synchronizing or loading the data to Oracle Billing Insight using RESTful Web services:

- **Account Provisioning.** Companies, billing accounts, or service agreements are created or updated in Oracle BRM, and BRM AQAdapter provisions the data to Oracle Billing Insight. (BRM AQAdapter also creates or updates the corresponding billing hierarchies as needed.)
- **Payment Synchronization.** Payment transactions, new credit card accounts, and payment contacts created or updated in Oracle BRM are synchronized by BRM AQAdapter to Oracle Billing Insight. (Payment transactions for both postpaid and prepaid update as external payments in Oracle Billing Insight.)
- **Monthly Billing.** You can schedule BRM Provider to initiate loading postpaid data from Oracle BRM to Oracle Billing Insight whenever the bill utility is run in Oracle BRM. The pin_bill_day utility triggers the custom billRun event, and BRM AQAdapter processes the loading of postpay billing data from Oracle BRM to Oracle Billing Insight.

BRM AQAdapter runs as a Java client outside of the Oracle WebLogic container.

Oracle BRM publishes events using the Synchronization Queue DM and the Enterprise Applications Integration (EAI) framework, which consists of the event notification system and the Payload Generator External Module (EM).

The Synchronization Queue DM and EAI framework work together to publish changes to a central Oracle Advanced Queuing (AQ) database queue. Oracle Billing Insight retrieves interested events from the specified Oracle AQ.

A dedicated event queue for integration is created in the Oracle BRM database. Only subscribed events publish into the event queue by Oracle BRM. BRM AQAdapter uses Spring JMS and Spring Data JDBC Extensions to retrieve events from AQ. The BRM AQAdapter event handler processes the events using Oracle Billing Insight Web service APIs. For more information, see *Web Services Reference for Oracle Billing Insight*.

## About Load Management Control Tables

Oracle Billing Insight uses the following load management tables with the Oracle BRM data load processes:

- **EDX_INTG_PROVIDER_LOAD_CTRL.** This table contains one entry for each type of data to track the start timestamp of the last load run.
- **EDX_INTG_PROVIDER_LOAD.** This table contains one entry for each load run to track the type of data load, the start and end timestamp, and the load status.
- **EDX_INTG_PROVIDER_LOG.** This table stores detailed log information for each load run, including load ID, event type (info, warning, or error), timestamp, affected table, row inserted, and log messages.

# Installing and Configuring BRM Provider

Follow these steps to install and configure BRM Provider, used to load billing data from Oracle BRM to Oracle Billing Insight.

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To install and configure BRM Provider*

1. Verify that the ETL environment is installed and configured. For details see *Installation Guide for Oracle Billing Insight*.
2. Configure the TNS alias:

   a. On the Oracle Billing Insight database server, open the tnsnames.ora file, configure a TNS alias connecting to the BRM database for your implementation, and then save the file.

   ```
   PINDB =
   (DESCRIPTION =
     (ADDRESS = (PROTOCOL = TCP)(HOST = Host_Name.Example.com)(PORT = Port_Num))
       (CONNECT_DATA =
        (SERVER = DEDICATED)
        (SERVICE_NAME = PINDB)
      )
   )
   ```

   where:

   - *Host_Name.Example.com* is the BRM database host name.
   - *Port_Num* is the BRM database listener port.

- *PINDB* is the BRM database service name.

BRM Provider creates a new schema on the Oracle Billing Insight database for pre-staging purposes, which stores simplified versions of Oracle BRM tables.

**b** Do one of the following:

- If you are installing BRM Provider directly on the Oracle Billing Insight database server, make sure that your tnsnames.ora file contains following alias connecting your Oracle Billing Insight database service.
- If you are installing BRM Provider remotely (not from Oracle Billing Insight database server), make sure that you have Oracle Client installed on your remote computer, and configure two TNS aliases on your Oracle Client; one must connect to the Oracle BRM database and the other must connect to the Oracle Billing Insight database. For example:

```
EBILL =
(DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = Host_Name.Example.com)(PORT = Port_Num))
   (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = EBILL)
 )
 )
```

where:

- *Host_Name.Example.com* is the EBILL database host name.
- *Port_Num* is the EBILL database listen port.
- *EBILL* is the EBILL database service name.

**3** Complete this step if your Oracle BRM implementation does not meet the following requirements:

- The IFW_IMPACT_CAT data in the pipeline schema is loaded into the CONFIG_IMPACT_CATEGORIES_T table in the Oracle BRM schema.
- The IFW_RUM data in the pipeline schema is loaded into the CONFIG_CANDIDATE_RUMS_T table in the Oracle BRM schema.

**a** (Skip this step if your pipeline tables and Oracle BRM tables reside in the same database schema in your deployment.) Go to the *EDX_HOME*/integration/db/provider/brm/Scripts directory. Connect to the pipeline schema and run the following script to grant the table access privileges to the Oracle BRM schema:

```
sqlplus pipeline_schema_name/pipeline_schema_password@DB_TNS_alias
@ifw_grant_priv.sql
```

where:

- *pipeline_schema_name* is the pipeline schema name.
- *pipeline_schema_password* is the pipeline schema password
- *DB_TNS_alias* is the Oracle BRM database TNS alias.

When prompted for the Oracle BRM database user, specify a value for *pin_1*, where *pin_1* is the Oracle BRM schema name.

**b** Connect to the Oracle BRM schema and run the following script:

```
sqlplus BRM_schema_name/BRM_schema_password@DB_TNS_alias@prerequisites.sql
```

where:

- *BRM_schema_name* is the Oracle BRM schema name.
- *BRM_schema_password* is the Oracle BRM schema password.
- *DB_TNS_alias* is the Oracle BRM database TNS alias.

The script prompts you for the Oracle BRM pipeline schema owner name.

When prompted, enter a value for the pipeline database user, specify a value for *pre_1*, where, *pre_1* is the pipeline schema name.

This step verifies that all data required by the integration solution is located in the Oracle BRM schema. Oracle BRM Provider extracts data from the Oracle BRM schema only.

**4** Configure the parameters in the BRMProvider.properties file, shown here, for your organization. (Verify that the BRM Provider Oracle Data Integrator (ODI) project is in the same working repository where the ETL project was imported.) This file is located in the *EDX_HOME/*integration/db/provider directory.

```
/*
 * Name: BRMProvider.properties
 *
 * $Header: $
 * Copyright (c) 1996, 2015 Oracle. All rights reserved.
 */
###############################################
###   Parameters of Pre-staging area        ###
###############################################
--EBILL db host name
EBILL_DB_HOST=
--EBILL db listen port
EBILL_DB_PORT=1521
--EBILL db service name
EBILL_DB_SERVICE=
--Pre-staging user name
PRESTG_USER=
--Pre-staging password
PRESTG_PASSWD=
--EBILL TNS (Transparent Network Substrate) NAME
EBILL_TNS_NAME=
--EBILL SYS PASSWORD
SYS_PASSWD=
--EBILL olap user name
OLAP_USER=
--EBILL oltp user name
OLTP_USER=
###############################################
###  Parameters of BRM DB or BRM staging DB ###
###############################################
--BRM Billing System ID(assigned value)
```

```
--Should be the same as the value in adapter configuration file.
BRM_BILLER_ID=BRM
--BRM pin user name
BRM_USER=
--BRM pin user password
BRM_PASSWD=
--BRM PINDB TNS alias
PINDB_TNS_NAME=
###############################################
###  Parameters of pre-staging tablespaces  ###
###############################################
--data tablespace file location
L_DB_PRESTG_DATA_FILE_LOC=/export/home/oracle/oradata
--index tablespace file location
L_DB_PRESTG_INDX_FILE_LOC=/export/home/oracle/oradata
###############################################
###   Parameters for importing ODI objects  ###
###############################################
-- ODI home directory
ODI_HOME_DIR=/export/home/oracle/Middleware/ODI1213
-- ODI user name
ODI_USER=SUPERVISOR
-- ODI password
ODI_PASSWD=
-- Master repository db host
MASTERREP_HOST=
-- Master repository db port
MASTERREP_PORT=1521
-- Master repository db service name
MASTERREP_SERVICE=
-- Master repository schema owner name
MASTERREP_USER=EBILL_ODI_REPO
-- Master repository schema password
MASTERREP_PASSWD=
-- Bin directory where the standalong agent is located
AGENT_BIN_DIR=/export/home/oracle/Middleware/ODI_Alone/user_projects/domains/
ODI_domain/bin
-- Agent name
AGENT_NAME=OracleDIAgent1
-- location where the Billing Insight XML files are located
XMLFILES_DIR=/Oracle/BillingInsight/integration/db/provider/odi
-- Work repository name
WORKREP_NAME=WORKREP
-- Billing data will be extracted since this start date (in MM/DD/YYYY format)
BILLING_DATA_START_DT=01/01/2015
```

5. Set up the pre-staging area:

   a. Go to the *EDX_HOME*/integration/db/provider directory and run the ant script.

   b. On the Oracle Billing Insight BRM Provider Install Menu, run Options 1, 2, and 3 individually or select Option 4 (Run steps 1-3) to install.

6. Specify the required parameters in the ODI studio to enable load notification email to be sent when loading completes:

a  Log into the ODI studio, expand Global Objects and then Global Variables, and set the following global variables values for notification:

| Global Variable | Description |
|---|---|
| 08_Email_ToAddress | The email address to receive processing notifications from ODI. |
| 09_Email_FromAddress | The email address to appear in the From field on email notifications generated by ODI. |
| 10_Email_CCAddress | The email address to copy on any notifications sent by ODI. |
| 11_Email_SMTPPort | The number of the authenticated SMTP listener port to use for sending email notifications from ODI. The default is 465. |
| 12_Email_SMTPServer | The name of the SMTP server to use for sending email notifications from ODI. |

b  Expand Projects, BRMProvider, DATA_LOAD, and Packages, and then double-click Notification package. On the bottom right panel, set the Password for Authentication parameter.

c  Regenerate the Notification Version 001 scenario. Expand Load Plans and Scenarios, BRMProvider, and DATA_LOAD. Right-click Notification Version 001 and select Regenerate.

7  If you want BRM AQAdapter to automatically start the postpay data extraction and load into Oracle Billing Insight using the ODI Webservice when Oracle BRM finishes a bill run, then follow these steps to modify the pin_bill_day script, adding the bill_run_event.sh script into the existing logic to publish a billRun event into the Advanced Queue:

a  To use encryption, make OpenSSL available on your Oracle BRM server.

The enq_bill_run_event.sh script uses OpenSSL to perform encryption.

b  Set ORACLE_HOME and PIN_HOME for the enq_bill_run_event.sh script.

c  Upload the enq_bill_run_event.sh and enq_bill_run_event.txt files located in the *EDX_HOME/*`integration/aqadaptor/brm/` directory to the `$PIN_HOME/bin` directory, and add execution privilege to the enq_bill_run_event.sh script.

d  Open the enq_bill_run_event.txt file and set the Oracle BRM schema password *as pin_db_password* in the following statement:

```
sm_pw pin_db_password
```

e  Run the enq_bill_run_event.sh script once to encrypt the password. This generates the enq_bill_run_event.enc file which contains the encrypted password. (The enq_bill_run_event.txt file is deleted after the script runs.)

f  Open the pin_bill_day file and add the last two lines as shown below after the execution of pin_bill_accts:

```
#
#       IMPORTANT: Must finish before next step.
#
```

```
#------------------------------------------------------------------
pin_bill_accts -pay_type 10007 $1 $2 $3
err=$?
if [ $err != 0 ]; then
   exit $err;
fi
pin_bill_accts -discount $1 $2 $3
err=$?
if [ $err != 0 ]; then
   exit $err;
fi
pin_bill_accts -sponsorship $1 $2 $3
err=$?
if [ $err != 0 ]; then
   exit $err;
fi
pin_bill_accts $1 $2 $3
err=$?
if [ $err != 0 ]; then
   exit $err;
fi
wait
echo "Publish billRun event"
enq_bill_run_event.sh > /dev/null
```

8 Comment out the pin_collect feature in the pin_bill_day script to enable customers to make payments using Oracle Billing Insight and suppress the automatic payment feature in Oracle BRM. Comment all of the pin_collect lines for automated credit card collection in the script as shown here:

```
#------------------------------------------------------------------
# Credit Card Collection        (pin_collect -pay_type 10003 -vendor fusa)
#
#       Collect the balance due from credit card accounts.
#
#------------------------------------------------------------------
#pin_collect -inactive -pay_type 10003 -vendor fusa
#pin_collect -active -pay_type 10003 -vendor fusa&
#pin_collect -close -pay_type 10003 -vendor fusa&
#------------------------------------------------------------------
# Wait for credit card collection to finish up.
#------------------------------------------------------------------
#wait
#------------------------------------------------------------------
# Direct Debit Collection       (pin_collect -pay_type 10005 -vendor fusa)
#
#       Collect the balance due from direct debit accounts.
#
#------------------------------------------------------------------
#pin_collect -inactive -pay_type 10005 -vendor fusa
#pin_collect -active -pay_type 10005 -vendor fusa&
#pin_collect -close -pay_type 10005 -vendor fusa&
#------------------------------------------------------------------
# Wait for direct debit collection to finish up.
```

```
#--------------------------------------------------------------------
#wait
#--------------------------------------------------------------------
# SEPA Direct Debit Collection       (pin_collect -pay_type 10018)
#
#       Collect the balance due from sepa accounts.
#
#--------------------------------------------------------------------
#pin_collect -inactive -pay_type 10018&
#pin_collect -active -pay_type 10018&
#pin_collect -close -pay_type 10018&
#--------------------------------------------------------------------
# Wait for sepa direct debit collection to finish up.
#--------------------------------------------------------------------
#wait
```

# Installing and Configuring BRM AQAdapter

BRM AQAdapter is provided with Oracle Billing Insight for use provisioning real-time data events from Oracle BRM to Oracle Billing Insight. You can install and run BRM AQAdapter on any computer that can access both the Oracle Billing Insight Web Services application server and the Oracle BRM database. However, you must create the event queue dedicated to Oracle Billing Insight on the Oracle BRM database server.

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To install and configure BRM AQAdapter on the Oracle BRM server*

1. Verify that the following products have been installed:
   - Oracle Communications Billing and Revenue Management (BRM) Synchronization Queue Data Manager (DM)
   - Oracle Communications Billing and Revenue Management (BRM) Enterprise Application Integration (EAI) Manager
2. Copy the following folder from Oracle Billing Insight to the target Oracle BRM server where you want to run BRM AQAdapter (to be able to run the database script on the Oracle BRM server). BRM AQAdapter can run as standalone application on any computer that can access the Oracle Billing Insight Web Services application server and the Oracle BRM database server.)
   - **UNIX.** *EDX_HOME*/integration/adapter
   - **Windows.** *EDX_HOME*\integration\adapter
3. Copy the following folder to a temporary folder on the BRM database server, such as /opt/portal/adapter:
   - ❑ **UNIX.** *EDX_HOME*/integration/db/adapter

- ❑ **Windows.** *EDX_HOME*`\integration\db\adapter`

4 On the Oracle BRM database server, connect to the Oracle BRM database as sysdba, and grant AQ privilege to the Oracle BRM schema:

a Go to the temporary folder where the scripts are copied, such as `/opt/portal/adapter`.

b Connect to the Oracle BRM db instance as sysdba, for example:

```
sqlplus sys/password@TNS_Alias as sysdba
```

where:

- ❑ *password* is sys account password of the Oracle BRM database.
- ❑ *TNS_Alias* is the Oracle BRM database TNS alias.

c Grant privilege to the schema, for example:

```
SQL>@grant_priv.sql Schema_Name
```

In this command, *Schema_Name* is the BRM schema name.

5 Create the event queue:

a On the Oracle BRM database server, go to the temporary folder where you copied the database scripts, such as `/opt/portal/adapter`, and connect to the BRM database as the BRM schema owner, for example:

```
sqlplus Schema_Name/Password@TNS_Alias
```

where:

- ❑ *Schema_Name* is BRM schema name.
- ❑ *Password* is the BRM schema password.
- ❑ *TNS_Alias* is the BRM database TNS alias.

b Create the event queue:

```
SQL>@create_queue.sql
```

6 You must populate the supported BRM payment types and status information, as well as how these definitions map to the Oracle Billing Insight payment type and status.

Oracle Billing Insight payment type and status definitions can be found in following tables:

- EDX_PMT_PAYMENT_STATUS_DEF
- EDX_PMT_PAYMENT_TYPE_DEF

a Go to the following directory on the Oracle Billing Insight database server:

- ❑ **UNIX.** *EDX_HOME/*`integration/db/adapter`
- ❑ **Windows.** *EDX_HOME*`\integration\db\adapter`

b In the following file, add the BRM payment types and status, plus the mappings to the Oracle Billing Insight definitions:

```
SQL>@seed_data.sql
```

c. Connect to the Oracle Billing Insight OLTP database as the schema owner, for example:

```
sqlplus OLTP_Schema_Name/Password@BillingInsight_TNA_Name
```

7. Configure the AQAdapter properties required for integration. On the server where you plan to run AQAdapter, update the parameters in the *EDX_HOME/*integration/aqadaptor/config/adaptor.properites file for your own environment, for example:

```
#Configurations for AQ database
#BRM database url
jdbc.url=jdbc:oracle:thin:@server.example.com:1521:TNS_Alias
#BRM schema name
jdbc.username=Username
#BRM schema password
jdbc.password=Password
#Queue name for BRM Billing Insight integration
queue.destination=BRM_Schema_Name.EBILL_SYNC_QUEUE
#Billing Insight web service application URL
webservice.uri=http://server.example.com:7001/rs/api/v2
#The date pattern for event date, it should be same as EAI Manager date format
datePattern=yyyy-MM-dd'T'HH:mm:ss
#Biller ID for BRM system, should match with what you configured for BRM
provider
biller.id=BRM
#ODI User
odi.user=SUPERVISOR
#ODI Password
odi.password=Password
#ODI Work Repository Name
odi.work.repository=WORKREP
#ETL auto reject flag, value should be 'Y' or 'N' with single quotes
etl.auto.reject='Y'
#ODI Web Service URI
odi.webservice.uri=http://server.example.com:20910/oraclediagent/
OdiInvoke?wsdl
#BRM OPCODE connection URI
infranet.connection=pcp://
root.0.0.0.1:&aes|08|0D5E11BFDD97D2769D9B0DBFBD1BBF7EE301373B06419EA64E70B27
65C7E3DE74F@slc00fmw:11960/0.0.0.1 /service/admin_client 1
```

8. Install and configure BRM DM AQ:

a. Upload the payloadconfig_ebilling_sync.xml file found in the *EDX_HOME/*integration/aqadaptor/brm/ directory to the $PIN_HOME/sys/eai_js on the BRM server.

The payloadconfi_ebilling_sysnc.xml file contains a list of business events and the content of each event that AQAdapter needs to process for the integration.

b. Log into the BRM server and edit the Infranet.properties, file, found in the $PIN_HOME/sys/eai_js/ directory to specify the name of your payload configuration file and set the property infranet.eai.xml_zero_epoch_as_null = false. Change infranet.eai.configFile to the payloadconfig_ebilling_sync.xml file, for example:

```
infranet.eai.configFile=/scratch/ra-user-1/opt/portal/7.5/sys/eai_js/
payloadconfig_ebilling_sync.xml
```

If your Oracle BRM implementation already has an EAI publisher, the Synchronization Queue DM installation program merges the Synchronization Queue DM payload configuration file with the existing payload configuration file that is referenced in the Infranet.properties file, located in the $PIN_HOME/sys/eai_js/ directory. For more information on configuring the EAI payload for the Synchronization Queue DM, see *Oracle Communications Billing and Revenue Management Synchronization Queue Manager*.

c. Open the aq_queuenames file in a text editor. This file is located in the $PIN_HOME/sys/dm_aq/ directory. Specify the Oracle Billing Insight queue name and subscribe the events that Oracle Billing Insight requires. Add the same content to the end of the aq_queuenames file located in the *EDX_HOME/*integration/aqadaptor/brm directory.

   Change $QUEUE_NAME to EBILL_SYNC_QUEUE. If you customized the create_queue.sql script, then use the queue name you created. For more information about configuring the Synchronization Queue DM, see *Oracle Communications Billing and Revenue Management Synchronization Queue Manager*.

d. Specify the list of BRM events that Oracle Billing Insight requires. Open the pin_notify_ebilling file located in the *EDX_HOME/*integration/aqadaptor/brm directory. Add (merge) the contents into the pin_notify file, located in the *PIN_HOME/*sys/data/config/ directory. Save the pin_notify file.

   Oracle Billing Insight requires the following events:

   - event/audit/customer/payinfo/cc
   - event/audit/customer/payinfo/dd
   - event/billing/payment/cc
   - event/billing/product/action/purchase
   - event/customer/billinfo/create
   - event/customer/billinfo/modify
   - event/customer/nameinfo
   - event/customer/status
   - event/device/associate
   - event/notification/account/create
   - event/notification/service/create
   - event/notification/service/modify
   - event/notification/service/pre_create

e. Under the $PIN_HOME/sys/data/config directory, run the following command to load event list into BRM database:

```
load_pin_notify pin_notify
```

   If you are not using the default configuration files, see details about loading the event notification list in *Communications Billing and Revenue Management Developer's Guide*.

f. Restart BRM eai_js and dm_aq. Under the `$PIN_HOME/bin` directory, run the following:

```
stop_eai_js
stop_dm_aq
start_eai_js
start_dm_aq
```

9. Update the token expiration default value in the Web Services application to prevent it from expiring:

   a. Edit the webservice.xma.xml file, located in the *EDX_HOME*`/xma/config/modules/webservice/` directory. In the IWebserviceAuthTokenProvider bean, change the expiredTimeLength value from 20 (the default) to -1.

   b. Restart your Web Services application.

10. Start AQAdapter:

    a. Verify that the Oracle Billing Insight RESTful Web Service server is running.

    b. Go to the folder where you installed aqadaptor and run the following script:

    ```
    ./start.sh
    ```

    c. Enter the CSR admin user name and password.

    d. Check the ./AQAdaptor.log for the following message: *BillingInsight AQ Adaptor Started Successfully*.

11. (Linux Only) Stop AQAdaptor. Go to the folder where you installed AQAdaptor and run the following script:

    ```
    ./stop.sh
    ```

# Configuring Mapping Tables

You must configure the following mapping tables before loading data from Oracle BRM to Oracle Billing Insight:

- PRESTG_BRM_ITEM_MASTER_MAP (Item mapping)
- EDX_INTG_PROVIDER_RESOURCE (Resource mapping)

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To configure the mapping tables for data loading*

1. Populate the PRESTG_BRM_ITEM_MASTER_MAP table with the values to map the Oracle BRM bill items for your implementation to the usage types and charge types in Oracle Billing Insight. For details, see "About the Item Mapping Table" on page 40.
2. Populate the EDX_INTG_PROVIDER_RESOURCE table with the code-name pairs for master data in Oracle Billing Insight. For details, see "About the Resource Mapping Table" on page 41.

## About the Item Mapping Table

The item mapping table, PRESTG_BRM_ITEM_MASTER_MAP maps Oracle BRM bill items to the usage types and charge types in Oracle Billing Insight. Bill items in Oracle BRM are customizable and vary in different BRM implementations. You populate this table when configuring the pre-staging schema.

Table 3 shows an example of a configured item mapping table, PRESTG_BRM_ITEM_MASTER_MAP.

Table 3. Sample of a Configured Item Mapping Table, PRESTG_BRM_ITEM_MASTER_MAP

| BRM_ITEM_TYPE | BRM_ITEM_TAG | CHARGE _TYPE | SUB_ CHARGE _TYPE | USAGE _TYPE | USAGE_ TYPE_ CD | STMT_ MONTHLY | STMT_ USAGE | STMT_ CREDIT | STMT _TAX | STMT _OTHER |
|---|---|---|---|---|---|---|---|---|---|---|
| /item/cycle_arrear | cycle_arrear | Null | Null | Null | Null | Y | Null | Null | Null | Null |
| /item/ cycle_forward | cycle_forward | Null | Null | Null | Null | Y | Null | Null | Null | Null |
| /item/ cycle_forward _arrear | cycle_forward _arrear | Null | Null | Null | Null | Y | Null | Null | Null | Null |
| /item/onetime | purchase | Y | Y | Null | Null | Null | Null | Null | Null | Y |
| /item/onetime | purchase_account | Y | Y | Null | Null | Null | Null | Null | Null | Y |
| /item/cycle_tax | cycle_tax | Null | Null | Null | Null | Null | Null | Null | Y | Null |
| /item/settlement/ roaming | roaming_ settlement | Y | Null | Null | Null | Null | Null | Null | Null | Y |
| /item/late_fee | LateFee | Y | Null | Null | Null | Null | Null | Null | Null | Y |
| /item/remittance | remittance | Null | Null | Null | Null | Null | Null | Null | Null | Null |
| /item/incentive | incentive | Null | Null | Null | Null | Null | Null | Null | Null | Null |
| /item/usage | RELUsage | Null | Null | Y | VOICE | Null | Y | Null | Null | Null |
| /item/usage | Tokai | Null | Null | Null | Null | Null | Null | Null | Null | Null |
| /item/misc | miscGPRS | Null | Null | Y | DATA | Null | Y | Null | Null | Null |
| /item/misc | miscWAP | Null | Null | Y | DATA | Null | Y | Null | Null | Null |
| /item/misc | miscGSMTEL | Null | Null | Y | VOICE | Null | Y | Null | Null | Null |
| /item/misc | miscGSMSMS | Null | Null | Y | MESSAG E | Null | Y | Null | Null | Null |
| /item/misc | miscGSMFAX | Null | Null | Y | VOICE | Null | Y | Null | Null | Null |
| /item/misc | miscGPRS | Null | Null | Y | DATA | Null | Y | Null | Null | Null |
| /item/misc | miscGSMDATA | Null | Null | Y | DATA | Null | Y | Null | Null | Null |

The STMT_MONTHLY, STMT_USAGE, STMT_CREDIT, STMT_TAX, and STMT_OTHER columns are used to classify the charge, and only one of these columns can be set to Y for an item type.

If the USAGE_TYPE column is set to Y, then the USAGE_TYPE_CD column must be set. Valid values for the USAGE_TYPE_CD column are VOICE, DATA, and MESSAGE.

You can map a bill item to CHARGE_TYPE and SUB_CHARGE_TYPE, if the bill item is not the usage charge or the monthly charge.

If the bill items change in Oracle BRM, you must update the item mapping table and load master data again.

## About the Resource Mapping Table

The resource table, EDX_INTG_PROVIDER_RESOURCE, stores the code-name pairs for Oracle Billing Insight master data. You can specify the interface display name for each code. You populate this table as part of configuring the pre-staging schema.

If any master data is changed in Oracle BRM, you must update the code-name pairs in the resource mapping table and load master data again.

# Loading Oracle BRM Master Data to Oracle Billing Insight

You use a dedicated ODI extract and load process to load master data from Oracle BRM to the Oracle Billing Insight database. You load master data when initially integrating Oracle BRM and Oracle Billing Insight, and again as needed if your master data changes.

Master data includes payment types, adjustment types, charge type and sub-charge type, product and sub-product info, tariff info, unit type and currency type. Master data contains DIM code and corresponding name information.

Master data cannot be rejected once it is loaded. Only one data load process is allowed at a time.

You must load Oracle BRM master data before provisioning account information and loading billing data.

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To load master data from Oracle BRM into Oracle Billing Insight*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.
3. Right click LOADMASTERDATA Version 001 and click Run.
4. Select ODIAgent from the Logical Agent drop-down list.
5. Specify 'NULL' in the Value field for the following variables in the Variable Values dialog:
   - **BRMPROVIDER.pStartDate**. When set to 'NULL' (in single quotation marks), the FROM_DATE value of EDX_INTG_PROVIDER_LOAD_CTRL table is used. The date format is YYYYMMDDHHMISS.

- **BRMPROVIDER.pEndDate** - When set to 'NULL' (in single quotation marks), the sysdate of the Oracle BRM database is used. The date format is YYYYMMDDHHMISS

  To change a value, uncheck the Latest Value check box, and enter the new value. Enclose all string values in single quotation marks.

  Master data created or modified between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) loads into Oracle Billing Insight.

6. If you want to automatically reject a load when an error occurs, set the global variable 06_Auto_Reject to ‘Y’ (in single quotation marks).

   If a load fails, automatic rejection clears the data from the associated staging tables and the data load log will show the discrepancy between the number of records inserted in the staging table and production fact tables. If a failure occurs when using this option, then you will no longer be able to reference the data in the staging tables.

   If you specify N to suppress this feature, you can reference the staging table data if a failure occurs. After successfully determining the cause of the failure, you must manually reject the data load. No other master data can be loaded from Oracle BRM until the current run master data is either published or rejected.

# Provisioning Oracle BRM Account Data in Batch to Oracle Billing Insight

This task describes how to load all existing company, account, and service-device data from the Oracle BRM database for initial integration into Oracle Billing Insight using a provided PL/SQL script.

**NOTE:** In a live production environment, you use BRM AQAdapter to provision new company, account, and service data in real time. For details, see “Provisioning Oracle BRM Account Data in Real-Time to Oracle Billing Insight Using BRM AQAdapter” on page 43.

**CAUTION:** The combination of the "CORPACCNO", "CORPTAXID", "STREET", "CITY", "STATE", and "ZIPCODE" fields must be unique, or the load will fail due to a unique constraints violation. Oracle BRM does not have this restriction.

This task is a step in “Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight” on page 25.

### *To export a batch provisioning file from Oracle BRM database to Oracle Billing Insight*

1. Connect to the Oracle BRM database.
2. Run the ExportProvisioningFile.sql script, located in the *EDX_HOME*/integration/db/provider/brm/Scripts/ directory.
3. For the biller ID input parameter, specify the value you configured for the BRM_BILLER_ID parameter in the BRMProvider.properties file, such as BRM_BILLER_ID=BRM. For example:

   ```
   sqlplus pin_1/BRM_schema_password@PINDB @ExportProvisioningFile BRM
   ```

4 The script generates a file named PROV_BILLING_XXXXXXXXXXXXXX.DAT in the same directory. Change XXXXXXXXXXXXXX in the file name to the timestamp, such as 20150610080000.

5 Process this DAT file using the ProvisioningData job in the Oracle Billing Insight Command Center to load accounts and services into Oracle Billing Insight. For details, see *Administration Guide for Oracle Billing Insight*.

# Provisioning Oracle BRM Account Data in Real-Time to Oracle Billing Insight Using BRM AQAdapter

In a live production environment, you use BRM AQAdapter with RESTful Web services to synchronize new company, account, and service or device data to the Oracle Billing Insight database in real-time. In addition, you can synchronize account contact information, payment accounts for credit cards, and payment transactions in real-time using RESTful Web services. For information on using BRM AQAdapter to provision account data to Oracle Billing Insight in real-time, see *Web Services Reference for Oracle Billing Insight*.

**NOTE:** Only Chase Paymentech Orbital gateway is supported in Oracle Billing Insight version 7.0.

This task is a step in "Integration Extension Points" on page 48.

### *To provision account data in real-time using BRM AQAdapter*

1 Use Web services to synchronize company, account, and service or device data from Oracle BRM to Oracle Billing Insight.

2 Run BRM AQAdapter continuously to process real-time events. If you stop AQAdapter, such as for maintenance, the events in the queue will process the next time you start AQAdapter.

3 Monitor provisioning exceptions. For details, see Monitoring Exceptions in BRM AQAdapter on page 43.

## Monitoring Exceptions in BRM AQAdapter

Follow these steps to monitor and resolve exceptions that occur when using BRM AQAdapter

### *To monitor exceptions in AQAdapter*

1 If an exception occurs, check the queue table. When an error is generated, the extension _E is appended to the queue table name. For example, EBILL_SYNC_QUEUE changes to EBILL_SYNC_QUEUE_E.

2 Review the error log in the EBILL_SYNC_QUEUE_E and correct the issue.

3 Remove the _E extension from the queue name.

4 In the queue table, update the STATE to 0.

AQAdapter resumes and re-executes the action.

# Loading the Oracle BRM Postpaid Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle BRM postpaid data to Oracle Billing Insight. You can use this process to load postpaid billing data in bulk when performing the initial integration of Oracle BRM and Oracle Billing Insight. (You can configure BRM Provider to trigger the postpaid load process using BRM AQAdapter when the bill utility is run in Oracle BRM. For details, see "Installing and Configuring BRM Provider" on page 29.)

If BRM AQAdapter and the ODI agent are deployed on different hosts, make sure that the ODI Agent listens on the hostname instead of the localhost.

Only one data load process is allowed at a time. You must load master data and provision account data before loading postpaid billing data.

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To load Oracle BRM postpaid data into Oracle Billing Insight*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.
3. Right click LOADPOSTPAIDDATA Version 001 and click Run.
4. Select ODIAgent from the Logical Agent drop-down list.
5. Specify ‘NULL’ in the Value field for the following variables in the Variable Values dialog:
   - **BRMPROVIDER.pStartDate**. When set to 'NULL' (in single quotation marks), the FROM_DATE value of EDX_INTG_PROVIDER_LOAD_CTRL table is used. The date format is YYYYMMDDHHMISS.
   - **BRMPROVIDER.pEndDate**. When set to 'NULL' (in single quotation marks), the sysdate of the Oracle BRM database is used. The date format is YYYYMMDDHHMISS.

   To change a value, uncheck the Latest Value check box, and enter the new value. Enclose all string values in single quotation marks.

   Bills generated between the BRMPROVIDER.pStartDate variable (inclusive)  and BRMPROVIDER.pEndDate variable (exclusive) load into Oracle Billing Insight.

6. If you want to automatically reject a load when an error occurs, set the global variable 06_Auto_Reject to ‘Y’ (in single quotation marks).

   If a load fails, automatic rejection clears the data from the pre-staging tables and the associated production tables. The data load log will show the discrepancy between the number of records inserted in the staging table and production fact tables. If a failure occurs when using this option, then you will no longer be able to reference the data in the staging tables.

   If you specify N to suppress this feature, you can reference the staging table data if a failure occurs. After successfully determining the cause of the failure, you must manually reject the data load before processing the next file. No other billing data can be loaded from Oracle BRM until the current run data is either published or rejected.

# Loading the Oracle BRM Unbilled (CDR) Data to Oracle Billing Insight

You use a dedicated extract and load process to load unbilled data from Oracle BRM to the Oracle Billing Insight database.

You can schedule unbilled data loading in configurable time periods (the default is every 15 minutes). For details, see Scheduling Unbilled and Prepaid Data Loading on page 46.

Only one data load process is allowed at a time.

This task is a step in “Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight” on page 25.

### *To load Oracle BRM unbilled data to Oracle Billing Insight*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.
3. Right click LOADUNBILLEDDATA Version 001 and click Run.
4. Select ODIAgent from the Logical Agent drop-down list.
5. Specify ‘NULL’ in the Value field for the following variables in the Variable Values dialog:

   - **BRMPROVIDER.pStartDate**. When set to 'NULL' (in single quotation marks), the FROM_DATE value of EDX_INTG_PROVIDER_LOAD_CTRL table is used. The date format is YYYYMMDDHHMISS.
   - **BRMPROVIDER.pEndDate**. When set to 'NULL' (in single quotation marks), the sysdate of the Oracle BRM database is used. The date format is YYYYMMDDHHMISS.

   To change a value, uncheck the Latest Value check box, and enter the new value. Enclose all string values in single quotation marks.

   Unbilled data generated between the BRMPROVIDER.pStartDate variable (inclusive)  and BRMPROVIDER.pEndDate variable (exclusive) load into Oracle Billing Insight.

6. If you want to automatically reject an unbill load when an error occurs, set the global variable 06_Auto_Reject to 'Y' (in single quotation marks).

   If a load fails, automatic rejection clears the data from the pre-staging tables and the associated staging tables. The data load log will show the discrepancy between the number of records inserted in the staging table and production fact tables. If a failure occurs when using this option, then you will no longer be able to reference the data in the staging tables.

   If you specify N to suppress this feature, you can reference the staging table data if a failure occurs. After successfully determining the cause of the failure, you must manually reject the data load before processing the next file. No other unbilled data can be loaded until the current run data is either published or rejected.

## Scheduling Unbilled and Prepaid Data Loading

You can schedule Oracle BRM to load unbilled and prepaid data on a regular basis. The default time interval for loading unbilled and prepaid data is every 15 minutes. you can change this value based on your business needs.

### *To schedule unbilled and prepaid data loading*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.
3. Right-click SCHEDULE_UB_PREPAY Version 001, click Scheduling, and double-click GLOBAL/ODIAgent.
4. Click Execution Cycle and change the value of the Interval Between Repetitions field to the time interval for your business scheduling needs.
5. Click Definition and change the status to Active.
6. Click the Topology tab, expand Physical Architecture and Agents, right-click OracleDIAgent1, and click Update Schedule.

# Loading the Oracle BRM Prepaid Data to Oracle Billing Insight

You use a dedicated ODI extract and load process to load prepaid data from Oracle BRM to the Oracle Billing Insight database.

You can schedule prepaid data loading in configurable time periods (the default is every 15 minutes). For details, see Scheduling Unbilled and Prepaid Data Loading on page 46.

Only one data load process is allowed at a time.

This task is a step in "Process of Setting Up and Integrating Oracle BRM With Oracle Billing Insight" on page 25.

### *To load Oracle BRM prepaid data to Oracle Billing Insight*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.
3. Right click LOADPREPAIDDATA Version 001 and click Run.
4. Select ODIAgent from the Logical Agent drop-down list.
5. Specify ‘NULL’ in the Value field for the following variables in the Variable Values dialog:

   - **BRMPROVIDER.pStartDate**. When set to 'NULL' (in single quotation marks), the FROM_DATE value of EDX_INTG_PROVIDER_LOAD_CTRL table is used. The date format is YYYYMMDDHHMISS.
   - **BRMPROVIDER.pEndDate**. When set to 'NULL' (in single quotation marks), the sysdate of the Oracle BRM database is used. The date format is YYYYMMDDHHMISS

   To change a value, uncheck the Latest Value check box, and enter the new value. Enclose all string values in single quotation marks.

   Prepaid data generated between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) load into Oracle Billing Insight.

6. If you want to automatically reject a prepaid load when an error occurs, set the global variable 06_Auto_Reject to ‘Y’ (in single quotation marks).

   If a load fails, automatic rejection clears the data from the pre-staging tables and the associated staging tables. The data load log will show the discrepancy between the number of records inserted in the staging table and production fact tables. If a failure occurs when using this option, then you will no longer be able to reference the data in the staging tables.

   If you specify N to suppress this feature, you can reference the staging table data if a failure occurs. After successfully determining the cause of the failure, you must manually reject the data load before processing the next file. No other prepaid data can be loaded until the current run data is either published or rejected.

# Rejecting a Data Load

If you set the value of the GLOBAL.06_Auto_Reject variable to N when loading master, postpaid, unbilled, or prepaid data using the scenarios for Oracle BRM integration, then if an error occurs during loading you must reject the load manually before you can reload the data or start another load.

(If you set the value of the GLOBAL.06_Auto_Reject variable to Y, then data is rejected automatically if an error occurs during loading.)

### *To manually reject data*

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand BRMProvider, and expand DATA_LOAD.

3. Right click REJECT_DATA Version 001, select Run.
4. Select ODIAgent from the Logical Agent drop-down list.
5. Specify the Value field for BRMPROVIDER.pEltKey variable in the Variable Values dialog. Use the value from the PRESTG_ETL_KEY field of the prestg.EDX_INTG_PROVIDER_LOAD table.

# Integration Extension Points

Within the limitations and assumptions described in this topic, the following fields can be extended in some integrations of Oracle BRM and Oracle Billing Insight.

### Service Number

The BRM_SERVICE_T table contains an extended column to store service agreement number. The service agreement number is shown for the billing account in the Oracle Billing Insight interface. The default rule for setting the service agreement number for an Oracle BRM service is:

- Use the phone device ID (phone number) as the service agreement number.
- If no phone device is associated with the service, then use the SIM device ID as the service agreement number.
- If no phone device and no SIM device are associated with the service, then use the service ID as the service agreement number.

The same logic is used to set the service number in BRM Provider, BRM Adapter, and in the batch provisioning script. If you customize the logic for setting the service agreement number in one place, then change the logic in all three places.

In BRM Provider, customize the pkg_intg_load_prestg_billing. pr_load_service procedure to change the default behavior.

The batch provisioning script, ExportProvisioningFile.sql, is located in the *EDX_HOME*/`integration/db/provider/brm/Scripts/` directory.

### Service Owner Name

If service is associated with a device in your implementation, then Oracle Billing Insight uses DEVICE_T.DECSR as the service owner name by default. You must populate DEVICE_T.DECSR if it is you want it to display in the Oracle Billing Insight interface.  If you do not want to use DEVICE_T.DECSR as the source of the service owner name, then customize the adapter handler.

### Area Code

Oracle Billing Insight uses the area code to store the calling city, calling country and called city, and the called country. Since this data is not available in Oracle BRM, four extended columns are included in the following pre-staging tables:

- BRM_EVT_DL_SESS_TL_GSM_T
- BRM_EVT_SESS_TL_GSM_T

If the information is available in your deployment, you can populate the column data as shown in Table 4:

Table 4. Extended Columns for Called and Calling City and Country

| Extended Column Name | Type | Length | Required? | Sample Data |
|---|---|---|---|---|
| CALLING_AREA | VARCHAR2 | 255 | N | Anytown, CA |
| CALLING_COUNTRY | VARCHAR2 | 255 | N | USA |
| CALLED_AREA | VARCHAR2 | 255 | N | Anytown, CA |
| CALLED_COUNTRY | VARCHAR2 | 255 | N | USA |

Customize the following procedures to set values for these extended columns:

- pkg_intg_load_prestg_billing.pr_load_evt_sess_tlc_gsm
- pkg_intg_load_prestg_billing. pr_load_evt_dl_sess_tlc_gsm

### Tariff

In Oracle BRM, a single transaction can be associated with multiple tariffs, however only one tariff can be shown with the CDR record in Oracle Billing Insight.

By default, BRM Provider gets the first tariff and extracts it with the CDR record to Oracle Billing Insight. You can change the default behavior by customizing the pkg_intg_load_stage_billing package implementation. The following procedures in the the pkg_intg_load_stage_billing package contain the default logic to get the tariff code:

- pr_load_4000_cdr
- pr_load_4900_cdr
- pr_load_9500_pp_cdr

### Tax Sub-charge Code

You can populate the tax sub-charge code using the SeedData.sql script, located in the *EDX_HOME/*`integration/db/provider/brm/Taxes/` directory.

There is one top-level charge type with code 'Taxes' and several sub-charge type codes, such as 0 (Federal tax) and 1 (State tax).

You can change the tax charge type codes for your business needs. Change the following procedure, used to extract tax-related charges, to reflect the code changes:

```
pkg_intg_load_stage_billing.pr_load_3100_servcharge
```

A charge note can be extracted to describe the charges. By default, the fn_get_tax_charge_note field is used to generate the charge note. You can customize this behavior for your business needs.

The charge note does not appear in the Oracle Billing Insight interface, however you can customize your report to show the charge note in your interface if necessary.

# 4 Configuring a Content Management Server

This chapter describes how to configure Oracle Billing Insight to store and access content in Oracle WebCenter, or integrate other content management systems. It includes the following topics:

- Configuring Oracle Billing Insight With Oracle WebCenter Content on page 51
- Integrating Oracle Billing Insight With Other Content Management Systems on page 52

## Configuring Oracle Billing Insight With Oracle WebCenter Content

You can configure Oracle Billing Insight to store batch report and statement files in Oracle WebCenter. By default, Command Center jobs store the generated batch report files, and the billing statement PDF and DAISY (Digital Accessible Information System) audio files to the local file system.

You can load statement PDF files from your billing system into Oracle WebCenter and provide the file location to the Self-Service application.

### *To configure Oracle Billing Insight with Oracle WebCenter content*

1. Open the integration.xma.xml file, located in the following directory:
   - **UNIX.** *EDX_HOME*/xma/config/modules/integration/
   - **Windows.** *EDX_HOME*\xma\config\modules\integration\
2. Locate the contentManager bean in the integration.xma.xml file:

   ```
   <bean id="contentManager"
   class="com.edocs.common.integration.contmgmt.FileContentManager"
   scope="singleton"/>
   ```

   Replace the content with the following code:

   ```
   <bean id="contentManager"
   class="com.edocs.common.integration.contmgmt.UCMContentManager"
   scope="singleton">
   <property name="contentServerProperties">
   <props> <prop key="contentServerURL">http://yourcontentserver:portnumber/cs/
   idcplg</prop>
   <prop key="contentServerUser">username</prop>
   <prop key="contentServerPassword">serverpassword</prop></props> </property>
   </bean>
   ```

3 If you want to use a secure HTTPS connection, then follow these steps:

a In the integration.xma.xml file, update the contentServerURL property in the contentServerConfig bean, for example:

https://yourcontentserver:portnumber/cs/idcplg

**NOTE:** The default HTTPS port for Oracle WebCenter is 16201.

b Download the HTTPS certification file from your Oracle WebCenter content server and import it into the Self-Service server. In the Oracle WebLogic console, click Environment, Servers, AdminServer, and then click the Keystores tab. Find the Java Standard Trust Keystore file location for your Oracle WebLogic server, for example:

```
/export/home/edocs/Softwares/bea/jdk160_29/jre/lib/security/cacerts
```

c Go to your $JAVA_HOME/bin directory, and import the certification file from the Oracle WebCenter content server to your Oracle WebLogic keystore, for example:

```
keytool -import -alias ucm -keystore /export/home/edocs/Softwares/bea/
jdk160_29/jre/lib/security/cacerts -file /export/home/edocs/example.crt -
trustcacerts
```

In this example, `/export/home/edocs/example.crt` is the certification file from Oracle WebCenter.

d Enter the keystore password.

4 Restart the Command Center server.

# Integrating Oracle Billing Insight With Other Content Management Systems

You can integrate other content management systems with Oracle Billing Insight to store batch report and statement files. By default, Command Center jobs store the generated batch report files and billing statement PDF and DAISY (Digital Accessible Information System) audio files to the local file system. If you want to use Oracle WebCenter for content management, see “Configuring Oracle Billing Insight With Oracle WebCenter Content” on page 51.

Follow these steps to integrate Oracle Billing Insight with other content management systems.

### *To integrate Oracle Billing Insight with a content management system*

1 Open the integration.xma.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/xma/config/modules/integration/
- **Windows.** *EDX_HOME*\xma\config\modules\integration\

2 Locate the contentManager bean in the integration.xma.xml file:

```
<bean id="contentManager"
class="com.edocs.common.integration.contmgmt.FileContentManager"
scope="singleton"/>
```

3 Replace the contentManager class with the name of your own, custom contentManagerclass. In the <props> </props> section of the contentServerProperties element, define as many properties as necessary to connect to your content server, for example:

```
<bean id="contentManager" class="com.yourcompany.customname.customCMS"
scope="singleton">
<property name="contentServerProperties"><props>
<prop key="contentServerURL">http://yourcontentserver:portnumber/cs/idcplg</
prop>
<prop key="contentServerUser">username</prop>
<prop key="contentServerPassword">userpassword</prop>
</props></property></bean>
```

4 Restart the Command Center server.

# Index