Integration Guide for Oracle Billing Insight

Version 7.1, Rev. A

March 2017

ORACLE

Contents

Chapter 6: Integrating Oracle Billing Insight With CRM Applications

1 What's New in This Release

What's New in Integration Guide for Oracle Billing Insight, Version 7.1, Rev. A

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

Table 1. What's New in Integration Guide for Oracle Billing Insight, Version 7.1, Rev. A

Topic	Description
"Configuring Mapping Tables" on page 46	Modified topic. Modified the steps for configuring mapping tables.
Chapter 5, "Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing,"	New chapter. Describes how to integrate Oracle Utilities Customer Care and Billing (CC&B) with Oracle Billing Insight.
"Integrating the Assisted Service Application With CRM or Other Back-Office Systems" on page 87	Modified topic. Added information about configuring a single-sign on system with the Assisted Service application.

What's New in Integration Guide for Oracle Billing Insight, Version 7.1

Table 2 lists the changes in this version of the documentation to support this release of the software.

Table 2. What's New in Integration Guide for Oracle Billing Insight, Version 7.1

Topic	Description
"Example of Integrating Oracle Billing Insight with CAS" on page 9	Modified topic. Added a step to set a property in the setDomainEnv.sh script for a CAS SSO-enabled domain.
Chapter 4, "Integrating Oracle Billing Insight With Oracle BRM"	Modified all topics. Added the option of loading Oracle BRM data to flat files, which can then be loaded into Oracle Billing Insight using the standard Oracle Billing Insight ODI load processes. This option makes it possible to integrate Oracle BRM in implementations where the Oracle Billing Insight database does not have direct access to the Oracle BRM staging database, such as an On-Demand deployment of Oracle Billing Insight with an On-Premise deployment of Oracle BRM.

Table 2. What's New in Integration Guide for Oracle Billing Insight, Version 7.1

Topic	Description
"Process of Installing and Configuring Oracle BRM Integration Components" on page 33 "Installing Oracle BRM Integration Components Using InstallAnywhere" on page 34	New topics. Descriptions of the process of installing the components required for Oracle BRM integration and how to use InstallAnywhere have been added.
"Integrating the Assisted Service Application With CRM or Other Back-Office Systems" on page 87	Modified topic. This topic was moved to this guide from *Implementation Guide for Oracle Billing Insight*. Minor updates have also been made.
"Avoiding Clickjacking Using X-Frame-Options Security Settings" on page 92	Modified topic. Updated the instructions for configuring the integrated Assisted Service application to avoid clickjacking.
"Configuring the Self-Service Application to Connect to Oracle Service Cloud" on page 93	New topic. Describes how to connect Oracle Billing Insight with Oracle Service Cloud.

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

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 9
- "Example of Integrating Oracle Billing Insight with OIM" on page 14

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. Edit the setDomainEnv.sh script (setDomainEnv.cmd for Windows) for the CAS SSO-enabled Oracle WebLogic domain. Add the following property into the property into the JAVA_VM definition:

   ```
   -Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.
   SAXParserFactoryImpl

   -Djavax.xml.parsers.DocumentBuilderFactory=org.apache.xerces.jaxp.
   DocumentBuilderFactoryImpl
   ```

   For example:

   ```
   JAVA_VM="${JAVA_VM} -Dedx.home=${EDX_HOME} -
   Dlog4j.configuration=file:${EDX_HOME}/config/log4j_csr.xml -
   Dorg.owasp.esapi.resources=${EDX_HOME}/config -
   DPAYMENTECH_HOME=$PAYMENTECH_HOME -Daxis.ClientConfigFile=$CYBERSOURCE_HOME/
   CyberSourceClientDeploy.wsdd -Dweblogic.security.SSL.protocolVersion=TLS1 -
   Dweblogic.security.SSL.minimumProtocolVersion=TLSv1.2 -
   Djavax.xml.parsers.SAXParserFactory=org.apache.xerces.jaxp.SAXParserFactoryImpl
   "
   export JAVA_VM
   ```

3. Import the SSO user in the Oracle Billing Insight database.
4. Access your LDAP server, and add a user and group for the Self-Service and Assisted Service applications.
5. 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-7.1.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
     ```

6 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-7.1.ear\META-INF\ directory:

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

For example:

```
<package-name>org.opensaml.*</package-name>
```

7 Add the following content to the web.xml file found in the *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.1.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>
```

8 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>
```

9 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.

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

11 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>
```

12 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.

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.
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.

13. 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-7.1.ear
    - rs-weblogic-7.1.ear
    - csr-app-7.1.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-7.1.ear/selfservice-1.0-SNAPSHOT.war/WEB-INF
- **Windows.** *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.1.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-7.1.ear/xma
- **Windows.** EDX_HOME\J2EEApps\selfsevice\weblogic\selfservice-weblogic-7.1.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-7.1.ear/APP-INF/lib/xma-classpath.jar/META-INF
- **Windows.** EDX_HOME\J2EEApps\selfsevice\weblogic\selfservice-weblogic-7.1.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-providerref="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-7.1.ear
- rs-weblogic-7.1.ear
- csr-app-7.1.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 3 shows the required and optional user information when synchronizing an external identity system with Oracle Billing Insight.

Table 3. 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: ■ User ID ■ First Name ■ Last Name ■ Role ■ Email Address ■ Company ID	Optional information for business user updates: ■ Preferred Language ■ Mobile Phone Number for SMS ■ Mobile Carrier for SMS ■ Brand (To determine which CSS file to use for the UI look and feel)
Consumer	Required information for consumer user updates: ■ User ID ■ First Name Last Name ■ Role ■ Email Address ■ Biller ID ■ Billing Account Number	Optional information for consumer user updates: ■ Preferred Language ■ Brand (To determine which CSS file to use for the UI look and feel) ■ Mobile Phone Number for SMS ■ Mobile Carrier for SMS
Assisted Service	Required information for Assisted Service user updates: ■ User ID ■ First Name Last Name ■ Role ■ Email Address	Optional information for Assisted Service user updates: ■ 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
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 to the external system.

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.CustomertUserProfileWSConnector">
</bean>
```

3 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

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 25.

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.

4 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 Integrating Oracle BRM With Oracle Billing Insight

The process of initial integration of Oracle Billing Insight with Oracle Communications Billing and Revenue Management (BRM) requires that you install and configure required software and load data in bulk.

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

1. Identify the integration solution you need to integrate Oracle BRM data with Oracle Billing Insight for your deployment:
   - **Loading Oracle BRM data directly to Oracle Billing Insight database tables.** Use this option if your Oracle Billing Insight database can access the Oracle BRM database directly. This method uses BRM Provider, which you configure to output Oracle BRM billing data to a pre-staging schema installed on the Oracle Billing Insight database, and load the data directly to the Oracle Billing Insight database production tables. BRM Provider makes use of specialized ODI processes described in this chapter.

- **Extracting Oracle BRM data to flat text files.** Use this option if your Oracle Billing Insight database does not have direct access to the Oracle BRM staging database, such as if you have an On-Demand (cloud service) deployment of Oracle Billing Insight and Oracle BRM is On-Premise. This integration solution uses BRM Provider, which you configure to extract Oracle BRM billing data into billing text files. BRM Provider makes use of specialized ODI processes described in this chapter.

  You must then FTP the files to Oracle where the hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. The standard Oracle Billing Insight ODI processes are described in *Administration Guide for Oracle Billing Insight*.

2. Verify that your Oracle BRM data conforms to the requirements of Oracle Billing Insight described in "Assumptions About Oracle BRM Data" on page 32. For information about how Oracle BRM data maps and synchronizes with Oracle Billing Insight, see "Reference for Mapping of Oracle BRM and Oracle Billing Insight Data Objects" on page 60.
3. Follow the steps in "Process of Installing and Configuring Oracle BRM Integration Components" on page 33. (It is in the configuration of BRM Provider that you specify Oracle BRM data loading output to either files or tables.)

   For more information about BRM Provider and BRM AQAdapter, see "About BRM Provider" on page 31 and "About BRM AQAdapter" on page 32.
4. "Configuring Mapping Tables" on page 46.
5. "Loading Oracle BRM Master Data to Oracle Billing Insight" on page 48

   If you output Oracle BRM master data to a flat file (you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider), then also copy the generated master text files to the appropriate load input directory on your Oracle Billing Insight installation, and load the master data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.
6. "Provisioning Oracle BRM Account Data in Batch to Oracle Billing Insight" on page 49
7. "Loading Oracle BRM Data (Postpay, Unbilled, and Prepay) to Oracle Billing Insight" on page 51

   If you output Oracle BRM billing data to a flat file (you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider), then FTP the generated billing files to the appropriate load input directory on your Oracle Billing Insight installation. The Oracle Service Cloud hosting team will load the billing data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.

About BRM Provider

You use BRM Provider to output data from Oracle BRM for integration with Oracle Billing Insight. You can configure BRM Provider to output Oracle BRM data in one of two ways, depending on your deployment:

- **Directly to the Oracle Billing Insight database.** You can configure BRM Provider to first move Oracle BRM data to a pre-staging area, transform and load the data into Oracle Billing Insight staging, and then move it into the Oracle Billing Insight production tables. Your deployment must be able to access the Oracle BRM staging database directly from Oracle Billing Insight. This method requires that your implementation of Oracle Billing Insight be able to access the Oracle BRM staging database directly.

  (To use this method, when you configure BRM Provider, set the OUTPUT_TYPE property in the BRMProvider.properties file to TABLE.)

- **To a text billing file.** BRM Provider can generate text billing files to a location that you specify. You then upload (FTP) the text billing files to the Oracle Billing Insight database, and load the billing data using the standard Oracle Billing Insight data loading processes described in *Administration Guide for Oracle Billing Insight*. This method provides an integration solution for deployments that can not access the Oracle BRM staging database directly from Oracle Billing Insight, such as where Oracle Billing Insight is an On-Demand (service cloud) implementation and Oracle BRM is On-Premise.

  (To use this method, when you configure BRM Provider, set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE.)

For information about how Oracle BRM data maps and synchronizes with Oracle Billing Insight, see "Reference for Mapping of Oracle BRM and Oracle Billing Insight Data Objects" on page 60.

Depending on the data volume, use the server configuration appropriate for unloading 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.

About the BRM Provider Load Processes

BRM Provider consists of a set of Oracle Data Integrator (ODI) processes and a pre-staging schema. The BRM Provider set of packages, load plans, and scenarios can load billing data from the Oracle BRM server into the Oracle Billing Insight database directly or unload the data into text billing files.

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 58.

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 directly to Oracle Billing Insight or to text files:

- Master

- Postpay
- Unbilled
- Prepay

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.

About BRM AQAdapter

In a live production environment, you use BRM AQAdapter to process 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 and payment contacts created or updated in Oracle BRM are synchronized by BRM AQAdapter to Oracle Billing Insight. (Payment transactions for both postpay and prepay update as external payments in Oracle Billing Insight.)
- **Monthly Billing.** You can schedule BRM Provider to initiate loading postpay 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*.

Assumptions About Oracle BRM Data

Certain 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. For additional information about how Oracle BRM data maps and synchronizes with Oracle Billing Insight, see "Reference for Mapping of Oracle BRM and Oracle Billing Insight Data Objects" on page 60.

Integration of Oracle Billing Insight with Oracle BRM is based on the following data assumptions:

- 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.

Process of Installing and Configuring Oracle BRM Integration Components

To integrate Oracle BRM with Oracle Billing Insight, perform the following tasks to install and configure the required integration components for your environment:

1. "Installing Oracle BRM Integration Components Using InstallAnywhere" on page 34
2. "Configuring BRM Provider" on page 35. Follow these steps to configure either direct load from Oracle BRM to Oracle Billing Insight database tables (table output) or loading of Oracle BRM billing data to a flat text file (file output).
3. "Configuring BRM AQAdapter" on page 42 to configure the synchronization and load of Oracle BRM data in a live production environment.

Installing Oracle BRM Integration Components Using InstallAnywhere

You can install the components required for integrating Oracle BRM with Oracle Billing Insight using InstallAnywhere.

To install Oracle BRM integration components using InstallAnywhere

1. Start InstallAnywhere in UI mode:
   - **UNIX.** Log in using the user and group name that you want to run BRM AQAdapter. Make sure DISPLAY is set, then enter the following command:

     ```
     ./BillingInsight.bin
     ```

     To start InstallAnywhere in Console Mode, enter the following command then follow the on-screen instructions:

     ```
     ./BillingInsight.bin -i console
     ```

   - **Windows.** Double-click the BillingInsight.exe file, and follow the on-screen instructions.
2. On the Introduction screen, read the Oracle Billing Insight introductory information. Click Next to continue.
3. On the License Agreement screen, read the licensing agreement carefully, select the terms acceptance, then click Next.
4. On the Enter Serial Number screen, enter your product serial number, then click Next.
5. (UNIX Only) On the Owner of Web Application Server Directories screen, enter the name of the owner. If you have installed other Oracle Billing Insight products, then use the same owner at this screen that you used for those product installations. Then click Next.
6. (UNIX Only) On the Group of Web Application Server Directories screen, enter the name of the group. If you have installed other Oracle Billing Insight products, then use the same group at this screen that you used for those product installations. Then click Next.
7. On the Choose Install Folder screen, accept the default installation folder or click Choose and enter the directory where you want to install the Oracle Billing Insight integration components files and directories.

   The directory where you install Oracle Billing Insight integration components is referred in this guide as *INTG_HOME*. Click Next to continue.
8. On the Choose Install Set screen, select Option 3, Integration Components (BRM), and click Next.
9. (Windows Only) On the Choose Shortcut Folder screen, choose the New Program Group, then click Next.

10 On the Preinstallation Summary screen, verify that the information is correct, then click Install.

To correct any entries, click Previous. The installer copies the Oracle Billing Insight software components to the designated installation folder. A status bar on the bottom of the screen shows each component being installed.

If the installation is successful, then a congratulatory message appears with the directory that contains the Oracle Billing Insight integration components. Click Next.

11 Click Done to exit the installer.

If the installation fails, then determine the cause of the problem, and run InstallAnywhere again to reinstall Oracle Billing Insight integration components.

Configuring BRM Provider

Follow these steps to configure BRM Provider. BRM Provider can be installed on the Oracle BRM server or a target server, which can be the Oracle Billing Insight database server or another database server.

This task is a step in "Process of Integrating Oracle BRM With Oracle Billing Insight" on page 29.

To configure BRM Provider

1 Verify your environment configuration:

- If you plan to extract and load billing data from Oracle BRM directly to the Oracle Billing Insight database tables, then verify that the ETL environment is installed and configured. For details see *Installation Guide for Oracle Billing Insight*.
- If you plan to extract billing data from Oracle BRM to a billing text file (for an On Demand implementation of Oracle Billing Insight), verify that Oracle Data Integrator (ODI) is installed, and the ODI repository and ODI agent are ready on the target database server where you plan to install the BRM Provider pre-staging schema.

2 Configure the TNS alias:

a On the target database server where you plan to install the pre-staging schema, 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 target database for pre-staging purposes, which stores simplified versions of Oracle BRM tables. When you plan to output Oracle BRM billing data to database tables (and specify the OUTPUT_TYPE property value as TABLE in the BRMProvider.properties file), the target database must be the Oracle Billing Insight database.

b If you are installing BRM Provider directly on the target database server, then make sure that your tnsnames.ora file contains the following alias connecting your target database service.

If you are installing BRM Provider remotely (not from the target database server), then 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 target database.

For example:

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

where:

- *Host_Name.Example.com* is the target database host name.
- *Port_Num* is the target database listen port.
- *TARGET_SERVICE* is the target 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 *INTG_HOME*/integration/brm/db/provider/brm/Scripts directory, where INTG_HOME is the directory where you installed Oracle Billing Insight integration components. 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 schema owner name, enter a value for *BRM_schema_owner_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 schema owner name, such as for *pre_1*.

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. This file is located in the *INTG_HOME*/integration/brm/db/provider directory. If you specify OUTPUT_TYPE as TABLE, verify that the BRM Provider ODI project is in the same working repository where the ETL project was imported.

Property Name	Description
OUTPUT_TYPE	The BRM Provider output type for billing data (postpay, prepay, and unbilled) and master data: ■ **FILE.** Outputs Oracle BRM data to a text file. The file will be located at the directory specified in the BRMPROV_BILLFILE_LOCATION property. ■ **TABLE.** Inserts Oracle BRM into Oracle Billing Insight staging tables. (Requires that the pre-staging schema be installed on the EBILL database.)
PRESTG_WITH_BRM	Indicates whether to install the pre-staging schema on the Oracle BRM staging database: ■ **Y.** Yes - Install with Oracle BRM. (If this property is set to Y, make sure ODI is available on the Oracle BRM database server, and the ODI repository and Agent have been created before installing BRM Provider.) ■ **N.** No - Do not install with Oracle BRM. (Set to N if OUTPUT_TYPE is TABLE.)
BRMPROV_BILLFILE_LOCATION	The directory for storing generated billing files. This property is required only if OUTPUT_TYPE is set to FILE.

Property Name	Description
BRMPROV_TMPFILE_LOCATION	The directory for storing temporary files. This property is required only if OUTPUT_TYPE is set to FILE.
PRESTG_HOST	The pre-staging database host name.
PRESTG_PORT	The pre-staging database port.
PRESTG_SERVICE	The pre-staging database service.
PRESTG_TNS_NAME	The pre-staging TNS (Transparent Network Substrate) name. This property must be the EBILL TNS name if OUTPUT_TYPE is set to TABLE.
PRESTG_USER	The pre-staging user name.
PRESTG_PASSWD	The pre-staging password.
PRESTG_SYS_PASSWD	The pre-staging SYS PASSWORD. This property must be the EBILL database sys password if OUTPUT_TYPE is set to TABLE.
OLAP_USER	The EBILL OLAP user name. This property is required only if OUTPUT_TYPE is set to TABLE.
BRM_BILLER_ID	The Oracle BRM Billing System ID (assigned value). The value must be the same as the value in BRM AQAdapter configuration file adaptor.properites.
BRM_USER	The Oracle BRM pin user name.
BRM_PASSWD	The Oracle BRM pin user password.
PINDB_TNS_NAME	The Oracle BRM PINDB TNS alias.
PINDB_SYS_PASSWD	The Oracle BRM PINDB SYS PASSWORD. This property is required if PRESTG_WITH_BRM is set to Y.
L_DB_PRESTG_DATA_FILE_LOC	The data tablespace file location.
L_DB_PRESTG_INDX_FILE_LOC	The index tablespace file location.
ODI_HOME_DIR	The ODI home directory, or ODI client home directory if installed remotely.
ODI_USER	The ODI user name (SUPERVISOR).
ODI_PASSWD	The ODI password.
MASTERREP_HOST	The master repository database host.
MASTERREP_PORT	The master repository database port.
MASTERREP_SERVICE	The master repository database service name.
MASTERREP_USER	The master repository schema owner name.
MASTERREP_PASSWD	The master repository schema password.

Property Name	Description
WORKREP_NAME	The work repository name.
AGENT_BIN_DIR	The bin directory where the standalone agent is located.
AGENT_HOST	The agent host. Specify the hostname instead of the localhost. The agent must be run on the database server where the pre-staging schema is installed.
AGENT_HOST_PORT	The agent port (the default is 20910).
AGENT_PROTOCOL	The agent protocol (the default is http).
AGENT_NAME	The physical agent name (the default is OracleDIAgent1).
LOG_AGENT_NAME	The logical agent name (the default ODIAgent).
XMLFILES_DIR	The directory where the BRM Provider ODI XML files are located.
BILLING_DATA_START_DT	The start date for extracting billing data (in MM/DD/YYYY format).

5 Set up the pre-staging area:

a Go to the *INTG_HOME*/integration/brm/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.

If you receive the following error when running Option 3, Configuring working repository, you can ignore it:

```
com.sunopsis.core.SecurityAccessException:
javax.crypto.IllegalBlockSizeException: Input length must be multiple of 16
when decrypting with padded cipher

ODI cannot decrypt the password cipher text which was not encrypted previously
by the current master repository cipher.
```

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.

Global Variable	Description
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 Web service when Oracle BRM finishes a bill run, then follow these steps to modify the pin_bill_day script, adding the eng_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 *INTG_HOME/*`integration/brm/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
```

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 Integrating Oracle BRM With Oracle Billing Insight" on page 29.

To 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.** *INTG_HOME*/integration/brm/adapter
   - **Windows.** *INTG_HOME*\integration\brm\adapter
3. Copy the following folder to a temporary folder on the BRM database server, such as /opt/portal/adapter:
   - **UNIX.** *INTG_HOME*/integration/brm/db/adapter
   - **Windows.** *INTG_HOME*\integration\brm\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:

- **UNIX.** *INTG_HOME*`/integration/brm/db/adapter`
- **Windows.** *INTG_HOME*`\integration\brm\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. Execute the seed_data.sql file, 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 *INTG_HOME*`/integration/brm/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/oracledіagent/OdiInvoke?wsdl
#BRM OPCODE connection URI
infranet.connection=pcp://
root.0.0.0.1:&aes|08|0D5E11BFDD97D2769D9B0DBFBD1BBF7EE301373B06419EA64E70B2765C
7E3DE74F@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 *INTG_HOME*/integration/brm/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 *INTG_HOME*`/integration/brm/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 *INTG_HOME*`/integration/brm/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. 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.*

10 (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 you can load data from Oracle BRM to Oracle Billing Insight:

- PRESTG_BRM_ITEM_MASTER_MAP (Item mapping)
- EDX_INTG_PROVIDER_RESOURCE (Resource mapping)
- EDX_INTG_PARTN_COMP_MAP (Company mapping)
- EDX_INTG_PARTN_ACCT_MAP (Account mapping)
- EDX_INTG_PERIOD_DIM (Billing period mapping)

This task is a step in "Process of Integrating Oracle BRM With Oracle Billing Insight" on page 29.

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 47.

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 48.

3 If you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider, synchronize the data from the following Oracle Billing Insight tables to the specified mapping table.

Oracle Billing Insight OLAP Database Table	Mapping Table
EDX_RPT_PARTN_COMP_MAP	EDX_INTG_PARTN_COMP_MAP
EDX_RPT_PARTN_ACCT_MAP	EDX_INTG_PARTN_ACCT_MAP
EDX_RPT_PERIOD_DIM	EDX_INTG_PERIOD_DIM

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

Table 4. 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	MESSAGE	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 extract and load process to load Oracle BRM master data directly to Oracle Billing Insight or to load into a master data file, depending on the output type you chose when installing BRM Provider (TABLE or FILE). You use this process to load master data in bulk when performing the initial integration of Oracle BRM and Oracle Billing Insight.

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.

You load master data when initially integrating Oracle BRM and Oracle Billing Insight and again as needed if your master data changes.

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 Integrating Oracle BRM With Oracle Billing Insight” on page 29.

To load master data from Oracle BRM to Oracle Billing Insight (or to a file)

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

- **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.

Master data created or modified between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) will be included in the output, generated in the format specified in your BRM Provider configuration:

- **To a Table.** When the output type configured in BRM Provider is TABLE, master data loads directly into a Oracle Billing Insight DIM table.
- **To a File.** When the output type configured in BRM Provider is FILE, master data is generated to a text file in the location you specify in the BRMPROV_BILLFILE_LOCATION parameter. You must upload (FTP) this file to the Oracle Billing Insight database server, where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. For details about standard Oracle Billing Insight ODI processes, see *Administration Guide for 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 are generating Oracle BRM output to files, automatic rejection removes files from the BRMPROV_BILLFILE_LOCATION and BRMPROV_TMPFILE_LOCATION directories if files exist.

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 or 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 50.

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 Integrating Oracle BRM With Oracle Billing Insight" on page 29.

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 *INTG_HOME*/integration/brm/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 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 for Oracle BRM integration in this version of Oracle Billing Insight.

This task is a step in "Integration Extension Points" on page 58.

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 51.

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 Oracle BRM Data (Postpay, Unbilled, and Prepay) to Oracle Billing Insight

BRM Provider comes with dedicated ODI processes for loading different types of Oracle BRM billing data directly to the Oracle Billing Insight database (or to a file) as described in the following topics:

Loading Oracle BRM Postpay Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle BRM postpay data directly to Oracle Billing Insight or to load into billing data files, depending on the output type you chose when installing BRM Provider (TABLE or FILE). This process lets you load postpay billing data in bulk when performing the initial integration of Oracle BRM and Oracle Billing Insight.

You can configure BRM Provider to trigger the postpay load process using BRM AQAdapter when the bill when the bill utility is run in Oracle BRM. For details, see "Configuring BRM Provider" on page 35.

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 postpay billing data.

This task is a step in "Process of Integrating Oracle BRM With Oracle Billing Insight" on page 29.

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

   - **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.

   Bills generated between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) will be included in the output, generated in the format specified in your BRM Provider configuration:

   - **To a Table.** When the output type configured in BRM Provider is TABLE, postpay billing data loads directly into a Oracle Billing Insight DIM table.
   - **To a File.** When the output type configured in BRM Provider is FILE, postpay billing data is generated to a text file in the location you specify in the BRMPROV_BILLFILE_LOCATION parameter. You must upload this file to the Oracle Billing Insight database server where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see *Administration Guide for 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 are generating Oracle BRM output to files, automatic rejection removes files from the BRMPROV_BILLFILE_LOCATION and BRMPROV_TMPFILE_LOCATION directories if files exist.

   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 Oracle BRM Unbilled (CDR) Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle BRM unbilled data directly to Oracle Billing Insight or to load into billing data files, depending on the output type you chose when installing BRM Provider (TABLE or FILE). This process lets you load unbilled data in bulk when performing the initial integration of Oracle BRM and Oracle Billing Insight.

You can schedule unbilled data loading in configurable time periods (the default is every 15 minutes). For details, see Loading the Oracle BRM Prepay Data to Oracle Billing Insight on page 54.

Only one data load process is allowed at a time.

This task is a step in "Process of Integrating Oracle BRM With Oracle Billing Insight" on page 29.

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

   - **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.

   Unbilled data generated between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) will be included in the output, generated in the format specified in your BRM Provider configuration:

   - **To a Table.** When the output type configured in BRM Provider is TABLE, unbilled data loads directly into a Oracle Billing Insight DIM table.
   - **To a File.** When the output type configured in BRM Provider is FILE, unbilled data is generated to a text file in the location you specify in the BRMPROV_BILLFILE_LOCATION parameter. You must upload this file to the Oracle Billing Insight database server where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see *Administration Guide for 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 are generating Oracle BRM output to files, automatic rejection removes files from the BRMPROV_BILLFILE_LOCATION and BRMPROV_TMPFILE_LOCATION directories if files exist.

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.

Loading the Oracle BRM Prepay Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle BRM prepay data directly to Oracle Billing Insight or to load into billing data files, depending on the output type you chose when installing BRM Provider (TABLE or FILE). This process lets you load prepay data in bulk when performing the initial integration of Oracle BRM and Oracle Billing Insight.

You can schedule prepay data loading in configurable time periods (the default is every 15 minutes). For details, see Loading the Oracle BRM Prepay Data to Oracle Billing Insight on page 54.

Only one data load process is allowed at a time.

This task is a step in “Process of Integrating Oracle BRM With Oracle Billing Insight” on page 29.

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

- **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.

Prepay data generated between the BRMPROVIDER.pStartDate variable (inclusive) and BRMPROVIDER.pEndDate variable (exclusive) will be included in the output, generated in the format specified in your BRM Provider configuration:

- **To a Table.** When the output type configured in BRM Provider is TABLE, prepay data loads directly into a Oracle Billing Insight DIM table.
- **To a File.** When the output type configured in BRM Provider is FILE, prepay data is generated to a text file in the location you specify in the BRMPROV_BILLFILE_LOCATION parameter. You must upload this file to the Oracle Billing Insight database server where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the prepay billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see *Administration Guide for Oracle Billing Insight*.

NOTE: The prepay load process loads data 60 days back from the pEndDate. If pStartDate is less than pEndDate - 60, then the process sets to EndDate - 60 automatically.

6 If you want to automatically reject a prepay 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 are generating Oracle BRM output to files, automatic rejection removes files from the BRMPROV_BILLFILE_LOCATION and BRMPROV_TMPFILE_LOCATION directories if files exist.

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 prepay data can be loaded until the current run data is either published or rejected.

Scheduling Unbilled and Prepay Data Loading

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

To schedule unbilled and prepay 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.

Rejecting a Data Load

If you set the value of the GLOBAL.06_Auto_Reject variable to 'N' (in single quotation marks) when loading master, postpay, unbilled, or prepay 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.

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.

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 as needed to continuously load and synchronize data from Oracle BRM to Oracle Billing Insight in a live production environment:

1. 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 46.

2. 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 48.

   If you output Oracle BRM master data to a flat file (you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider), then also copy the generated master text files to the appropriate load input directory on your Oracle Billing Insight installation, where the Oracle Cloud Service hosting team will load the master data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.

3. Run the HierarchyCopy job at beginning of the each new billing period. For details, see *Administration Guide for Oracle Billing Insight*.

4. Synchronize new or updated Oracle BRM account data in real-time using RESTful Web services with BRM AQAdapter. You can provision account contact information 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 50.

5. Schedule BRM Provider to initiate loading postpay 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 postpay data load while configuring BRM Provider. For details, see "Configuring BRM Provider" on page 35.)

   If you output Oracle BRM postpay data to a flat file (you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider), then also copy the generated postpay billing files to the appropriate load input directory on your Oracle Billing Insight installation, where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the postpay billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see *Administration Guide for Oracle Billing Insight*.

6 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 Prepay Data Loading" on page 55.

If you chose to output Oracle BRM data to a flat file (you set the OUTPUT_TYPE property in the BRMProvider.properties file to FILE when configuring BRM Provider), then also copy the generated billing files to the appropriate load input directory on your Oracle Billing Insight installation, where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the unbilled and prepay billing text files into the Oracle Billing Insight database production tables.

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 *INTG_HOME*/integration/brm/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 5.

Table 5. 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 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 by customizing following procedure:

```
pkg_intg_load_stage_master.pr_populate_master_records
```

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 variable and procedure, used to extract tax-related charges, to reflect the code changes:

```
pkg_intg_load_stage_billing.gTaxChargeTypeCode
```

```
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.

Reference for Mapping of Oracle BRM and Oracle Billing Insight Data Objects

The following topics contain reference information about how Oracle BRM data objects map to major Oracle Billing Insight data objects, and how the data is handled during the extraction and synchronization:

Mapping of Oracle BRM Billing Data Objects With Oracle Billing Insight

The process of integrating Oracle BRM data requires provisioning of billing objects, such as company, account, service or device, and payment account, into Oracle Billing Insight. You can provision this data in either real-time or in batch.

In real-time provisioning, BRM Adapter processes the AQ events and creates objects in Oracle Billing Insight using RESTful Web Services. Table 6 lists the Oracle Billing Insight objects and the Oracle AQ events that BRM Adapter processes, the RESTful Web Service API, and the Oracle Billing Insight production tables that store these provisioning objects.

Table 6. Synchronization of Oracle Billing Insight Provisioning Objects

Oracle Billing Insight Object	Oracle AQ Event Name	Oracle Billing Insight RESTful API	Oracle Billing Insight Production Table Name	Notes
Company	/event/notification /account/create	/accounts	EDX_OMF_COMPANY	CANON_ COMPANY maps to company ID
Account	/event/notification/account/create /event/customer/billinfo/create	/accounts	EDX_OMF_ACCOUNT	A/R responsive accounts only ACCOUNT_NO maps to account number
Service Agreement	/event/notification/service/create /event/notification/service/modify /event/device/associate /event/billing/product/action/purchase	/serviceAgreements	EDX_OMF_SERVICEAGREEMENT	DEVICE_ID of number device maps to service agreement number
Billing Contact	/event/customer/nameinfo	/accounts	EDX_OMF_CONTACT_INFO	None
Payment Account	/event/audit/customer/payinfo/cc	/payment/pmtAccounts	PAYMENT_ACCOUNTS	Credit card only
Payment Transaction	/event/billing/payment/cc	/payment/ transactions/external	EDX_PMT_EXT_PAYMENT	External payments to Oracle Billing Insight

Mapping of Oracle BRM Database Extraction Tables With Oracle Billing Insight

A subset of Oracle BRM data in 40 database tables are read, extracted, and mapped to Oracle Billing Insight. BRM Provider creates 37 tables in its schema as pre-staging tables. These pre-staging tables contain data with mirrored rows but columns that are simplified relative to the corresponding BRM tables. All 40 Oracle BRM tables, 37 Oracle Billing Insight pre-staging tables created by BRM Provider, and their mapping relationships are shown in Table 7. (It is recommended that you add indexes to existing Oracle BRM tables to improve extraction performance. For details on recommended indexes, see Table 8 on page 63.)

Table 7. Mapping of Oracle BRM Database Tables With Pre-Staging Tables

Number	Oracle BRM Table	Oracle Billing Insight Pre-Staging Table	BRM Provider Procedure
1	ACCOUNT_NAMEINFO_T	BRM_ACCOUNT_NAMEINFO_T	pkg_intg_load_prestg_billing.pr_load_account_nameinfo
2	ACCOUNT_T	BRM_ACCOUNT_T	pkg_intg_load_prestg_billing.pr_load_account
3	BAL_GRP_SUB_BALS_T	BRM_BAL_GRP_SUB_BALS_T	pkg_intg_load_prestg_billing.pr_load_bal_grp_sub_bals
4	BAL_GRP_T	BRM_BAL_GRP_T	pkg_intg_load_prestg_billing.pr_load_bal_grp

Table 7. Mapping of Oracle BRM Database Tables With Pre-Staging Tables

Number	Oracle BRM Table	Oracle Billing Insight Pre-Staging Table	BRM Provider Procedure
5	BILL_T	BRM_BILL_T	pkg_intg_load_prestg_billing.pr_load_bill
6	BILLINFO_T	BRM_BILLINFO_T	pkg_intg_load_prestg_billing.pr_load_bill_info
7	CONFIG_BEID_BALANCES_T	BRM_CFG_BEID_BAL_T	pkg_intg_load_prestg_master.pr_load_cfg_beid_bal
8	CONFIG_BUS_PROFILE _KEY_VALUE_T	BRM_CFG_BUS_PROF_KEY_VALUE_T	pkg_intg_load_prestg_master.pr_load_cfg_bus_prof
9	CONFIG_CANDIDATE_RUMS_T	BRM_CFG_CANDI_RUMS_T	pkg_intg_load_prestg_master.pr_load_cfg_rums
10	CONFIG_IMPACT_CATEGORIES_T	BRM_CFG_IMPACT_CAT_T	pkg_intg_load_prestg_master.pr_load_cfg_impact_ca
11	CONFIG_ITEM_TAG_T	BRM_CFG_ITEM_TAGS_T	pkg_intg_load_prestg_master.pr_load_cfg_item_tag
12	CONFIG_ITEM_TYPES_T	BRM_CFG_ITEM_TYPES_T	pkg_intg_load_prestg_master.pr_load_cfg_item_types
13	CONFIG_RUM_MAP_T	BRM_CFG_CANDI_RUMS_T	pkg_intg_load_prestg_master.pr_load_cfg_rums
14	CONFIG_T	BRM_CFG_T	pkg_intg_load_prestg_master.pr_load_cfg
15	DD_OBJECTS_T	BRM_DD_OBJECTS_T	pkg_intg_load_prestg_master.pr_load_dd_objects
16	DEAL_PRODUCTS_T	BRM_DEAL_PROD_T	pkg_intg_load_prestg_master.pr_load_deal_prod
17	DEAL_T	BRM_DEAL_T	pkg_intg_load_prestg_master.pr_load_deal
18	DEVICE_SERVICES_T	BRM_SERVICE_T	pkg_intg_load_prestg_billing.pr_load_service
19	DEVICE_T	BRM_SERVICE_T	pkg_intg_load_prestg_billing.pr_load_service
20	DISCOUNT_T	BRM_DISCOUNT_T	pkg_intg_load_prestg_master.pr_load_discount
21	DISCOUNT_USAGE_MAP_T	BRM_DISCOUNT_UGE_MAP_T	pkg_intg_load_prestg_master.pr_load_discount_usage _map
22	EVENT_BAL_IMPACTS_T	BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_prestg_billing.pr_load_event_bal_impact
23	EVENT_BILLING_PAYMENT_T	BRM_EVT_BILLING_PMT_T	pkg_intg_load_prestg_billing.pr_load_event_bill_pmt
24	EVENT_DLAY_SESS_TLCS_T	BRM_EVT_DL_SESS_TL_T	pkg_intg_load_prestg_billing.pr_load_evt_dl_sess_tlc
25	EVENT_DLYD_SESSION_TLCO _GSM_T	BRM_EVT_DL_SESS_TL_GSM_T	pkg_intg_load_prestg_billing.pr_load_evt_dl_sess_tlc _gsm
26	EVENT_PRODUCT_FEE_CYCLE_T	BRM_EVT_T	pkg_intg_load_prestg_billing.pr_load_event
27	EVENT_SESSION_TLCO_GSM_T	BRM_EVT_SESS_TL_GSM_T	pkg_intg_load_prestg_billing.pr_load_evt_sess_tlc_gsm
28	EVENT_SESSION_TLCS_T	BRM_EVT_SESS_TL_T	pkg_intg_load_prestg_billing.pr_load_evt_sess_tlc
29	EVENT_T	BRM_EVT_T	pkg_intg_load_prestg_billing.pr_load_event
30	EVENT_TAX_JURISDICTIONS_T	BRM_EVT_TAX_JURI_T	pkg_intg_load_prestg_billing.pr_load_event_tax_juris
31	GROUP_PRICELIST_MEMBERS_T	BRM_GROUP_PLANLIST_MEM_T	pkg_intg_load_prestg_master.pr_load_group_plan_mem
32	GROUP_T	BRM_GROUP_T	pkg_intg_load_prestg_master.pr_load_groups
33	ITEM_T	BRM_ITEM_T	pkg_intg_load_prestg_billing.pr_load_bill_item
34	PLAN_SERVICES_DETAILS_T	BRM_PLAN_SERV_DEALS_T	pkg_intg_load_prestg_master.pr_load_plan_serv_deal
35	PLAN_SERVICES_T	BRM_PLAN_SERV_T	pkg_intg_load_prestg_master.pr_load_plan_serv
36	PLAN_T	BRM_PLAN_T	pkg_intg_load_prestg_master.pr_load_plan
37	PRODUCT_T	BRM_PRODUCT_T	pkg_intg_load_prestg_master.pr_load_product
38	PRODUCT_USAGE_MAP_T	BRM_PRODUCT_UGE_MAP_T	pkg_intg_load_prestg_master.pr_load_product_usage _map

Table 7. Mapping of Oracle BRM Database Tables With Pre-Staging Tables

Number	Oracle BRM Table	Oracle Billing Insight Pre-Staging Table	BRM Provider Procedure
39	PURCHASED_PRODUCT_T	BRM_PURCH_PRODUCT_T	pkg_intg_load_prestg_billing.pr_load_purchased_product
40	SERVICE_T	BRM_SERVICE_T	pkg_intg_load_prestg_billing.pr_load_service

Recommended Indexes on Oracle BRM Database Tables

When BRM Provider extracts Oracle BRM data, the query uses a date and time range parameter. To improve the performance of the query, it is recommended that you create indexes on the Oracle BRM database tables listed in Table 8.

Table 8. Indexes Recommended on Oracle BRM Database Tables

Number	Oracle BRM Table Name	Index Column Name
1	ACCOUNT_T	CREATED_T
2	SERVICE_T	CREATED_T
3	SERVICE_T	MOD_T
4	PURCHASED_PRODUCT_T	CREATED_T
5	PURCHASED_PRODUCT_T	MOD_T
6	BILLINFO_T	CREATED_T
7	BILLINFO_T	MOD_T
8	BILLINFO_T	POID_ID0, BUSINESS_PROFILE_OBJ_ID0
9	BAL_GRP_T	CREATED_T
10	BAL_GRP_T	MOD_T
11	BILL_T	CREATED_T, MOD_T, END_T
12	BILL_T	END_T, BILLINFO_OBJ_ID0, ACCOUNT_OBJ_ID0, POID_ID0
13	ITEM_T	CREATED_T, MOD_T
14	ITEM_T	POID_ID0, BILL_OBJ_ID0
15	ITEM_T	POID_ID0, POID_TYPE, CREATED_T, ACCOUNT_OBJ_ID0
16	EVENT_T (Local partitioned index)	END_T
17	EVENT_PRODUCT_FEE_CYCLE_T (Local partitioned index)	OBJ_ID0, CYCLE_START_T, CYCLE_END_

Mapping of Pre-Staged Oracle BRM Master Data With Oracle Billing Insight

The Oracle Billing Insight data loading process accepts two types of input data: master data and billing data. BRM Provider loads data from Oracle BRM tables into pre-staging tables, and then transforms this data and loads it into flat files or staging tables for Oracle Billing Insight data loading. Table 9 shows how master data from Oracle Billing Insight pre-staging tables maps to production tables and to file record types. (Mappings for billing transaction data are shown in Table 10 on page 65.)

Table 9. Mapping of Pre-Staged Master Data With Oracle Billing Insight

BRM Provider Pre-Staging Table	Transformation Procedure	Oracle Billing Insight Production Table	Oracle Billing Insight File Input Record Type
BRM_DD_OBJECTS_T	pkg_intg_load_stage_master.pr_load_100_pmt_type	EDX_RPT_PAYMENT_TYPE_DIM	100 - Payment type
BRM_DD_OBJECTS_T	pkg_intg_load_stage_master.pr_load_110_adjust_type	EDX_RPT_ADJUSTMENT_TYPE_DIM	110 - Adjustment type
BRM_CFG_ITEM_TYPES_T BRM_DISCOUNT_T Tax	pkg_intg_load_stage_master.pr_load_120_charge_type pr_populate_master_records	EDX_RPT_CHARGE_TYPE_DIM	120 - Charge type
BRM_CFG_ITEM_TYPES_T Tax jurisdiction	pkg_intg_load_stage_master.pr_load_130_subcharge_type pr_populate_master_records	EDX_RPT_SUB_CHARGE_TYPE_DIM	130 - Sub charge type
BRM_PLAN_T BRM_PLAN_SERV_T BRM_PLAN_SERV_DEALS_T BRM_DEAL_T BRM_DEAL_PROD_T BRM_PRODUCT_T BRM_GROUP_T BRM_GROUP_PLANLIST_MEM_T	pkg_intg_load_stage_master.pr_load_150_prod_subprod	EDX_RPT_PRODUCT_DIM EDX_RPT_SUB_PRODUCT_DIM	150 - Product, sub product
BRM_CFG_IMPACT_CAT_T	pkg_intg_load_stage_master.pr_load_170_tariff_code	EDX_RPT_TARIFF_DIM	170 - Tariff
BRM_CFG_CANDI_RUMS_T	pkg_intg_load_stage_master.pr_load_220_unit_type	EDX_RPT_UNIT_DIM	220 - Unit
Direction	pkg_intg_load_stage_master.pr_populate_master_records	EDX_RPT_DIRECTION_DIM	240 - Direction
BRM_CFG_BEID_BAL_T	pkg_intg_load_stage_master.pr_load_270_currency	EDX_RPT_CURRENCY_DIM	270 - Currency

Mapping of Pre-Staged Oracle BRM Billing Transaction Data With Oracle Billing Insight

Table 10 shows how billing transaction data from Oracle Billing Insight pre-staging tables maps to production tables and to file record types. Only critical pre-staging tables are listed for each record type.

Table 10. Mapping of Pre-Staged Billing Transaction Data With Oracle Billing Insight

BRM Provider Pre-Staging Table	Transformation Procedure	Oracle Billing Insight Table	Oracle Billing Insight File Input Record Type
BRM_BILL_T BRM_EVT_BAL_IMPACTS_T BRM_EVT_BILLING_PMT_T	pkg_intg_load_stage_billing. pr_load_1000_stmt	EDX_RPT_STATEMENT_FACT	1000 - Statement
BRM_BILL_T BRM_ITEM_T BRM_EVT_BILLING_PMT_T	pkg_intg_load_stage_billing. pr_load_1100_stmtpmt	EDX_RPT_STATEMENT_PAYMENT_FACT	1100 - Statement Payment
BRM_BILL_T BRM_ITEM_T BRM_EVT_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_1200_stmtadj	EDX_RPT_STATEMENT_ADJUSTMENT_FACT	1200 - Statement Adjustment
BRM_BILL_T BRM_ACCOUNT_NAMEINFO_T	pkg_intg_load_stage_billing. pr_load_1300_address	EDX_RPT_ADDRESS_DIM	1300 - Mail-to Address
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_2000_account_a pr_load_2000_account_s	EDX_RPT_ACCOUNT_FACT	2000 - Account charges
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_2100_acctcharge	EDX_RPT_ACCOUNT_CHARGE_FACT	2100 - Account charges at charge type level
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_3000_service	EDX_RPT_SERVICE_FACT	3000 - Service charges
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_3100_servcharge	EDX_RPT_SERVICE_CHARGE_FACT	3100 - Service charge at charge type level
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T BRM_PURCH_PRODUCT_T	pkg_intg_load_stage_billing. pr_load_3200_servprod	EDX_RPT_SERVICE_PRODUCT_FACT	3200 - Service charge at product level
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_3300_servusage	EDX_RPT_SERVICE_USAGE_FACT	3300 - Service charge at usage level
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T	pkg_intg_load_stage_billing. pr_load_3400_servtariff	EDX_RPT_SERVICE_TARIFF_FACT	3400 - Service charge at tariff level
BRM_BILL_T BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T BRM_EVT_DL_SESS_TL_T BRM_EVT_DL_SESS_TL_GSM_T	pkg_intg_load_stage_billing. pr_load_4000_cdr	EDX_RPT_SERVICE_DETAIL_FACT	4000 - Service detail
BRM_EVT_BAL_IMPACTS_T BRM_EVT_DL_SESS_TL_T BRM_EVT_DL_SESS_TL_GSM_T	pkg_intg_load_stage_billing. pr_load_4000_cdr	EDX_RPT_UNBILLED_DETAIL_FACT	4900 - Unbilled detail
BRM_EVT_BAL_IMPACTS_T BRM_EVT_BILLING_PMT_T BRM_BAL_GRP_T BRM_BAL_GRP_SUB_BALS_T	pkg_intg_load_stage_billing. pr_load_9000_pp_account	EDX_RPT_PREPAY_ACCOUNT_FACT	9000- Prepaid charges

Table 10. Mapping of Pre-Staged Billing Transaction Data With Oracle Billing Insight

BRM Provider Pre-Staging Table	Transformation Procedure	Oracle Billing Insight Table	Oracle Billing Insight File Input Record Type
BRM_ITEM_T BRM_EVT_BAL_IMPACTS_T BRM_PURCH_PRODUCT_T	pkg_intg_load_stage_billing. pr_load_9200_pp_product	EDX_RPT_PREPAY_PRODUCT_FACT	9200 - Prepaid charges at product level
BRM_ITEM_T BRM_EVT_T BRM_EVT_ BAL_IMPACTS_T BRM_EVT_SESS_TL_T BRM_EVT_SESS_TL_GSM_T	pkg_intg_load_stage_billing. pr_load_9500_pp_cdr	EDX_RPT_PREPAY_DETAIL_FACT	9500 - Prepaid details

5 Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing

This chapter describes how to integrate Oracle Billing Insight with Oracle Utilities Customer Care And Billing (CC&B). It includes the following topics:

Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing

The process of initial integration of Oracle Billing Insight with Oracle Utilities CC&B requires that you install and configure required software and load data in bulk.

To integrate Oracle Utilities CC&B with Oracle Billing Insight, perform the following tasks:

1. Identify the integration solution you need to integrate Oracle Utilities CC&B data with Oracle Billing Insight for your deployment:
   - **Loading Oracle Utilities CC&B data directly to Oracle Billing Insight database tables.** Use this option if your Oracle Billing Insight database can access the Oracle Utilities CC&B database directly. This method uses CC&B Provider, which you configure to output Oracle Utilities CC&B billing data to a pre-staging schema installed on the Oracle Billing Insight database, and load the data directly to the Oracle Billing Insight database production tables. CC&B Provider makes use of specialized ODI processes described in this chapter.

- **Extracting Oracle Utilities CC&B data to flat text files.** Use this option if your Oracle Billing Insight database does not have direct access to the Oracle Utilities CC&B database, such as if you have an On-Demand (cloud service) deployment of Oracle Billing Insight and Oracle Utilities CC&B is On-Premise. This integration solution uses CC&B Provider, which you configure to extract Oracle Utilities CC&B billing data into billing text files. CC&B Provider makes use of specialized ODI processes described in this chapter.

  You must then FTP the files to Oracle where the hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. The standard Oracle Billing Insight ODI processes are described in *Administration Guide for Oracle Billing Insight*.

2. Verify that your Oracle Utilities CC&B data conforms to the requirements of Oracle Billing Insight described in "Assumptions About Oracle Utilities CC&B Data" on page 70. For information about how Oracle Utilities CC&B data maps with Oracle Billing Insight, see "Reference for Mapping of Oracle Utilities CC&B and Oracle Billing Insight Data Objects" on page 83.
3. Follow the steps in "Process of Installing and Configuring Oracle Utilities CC&B Integration Components" on page 70. (It is in the configuration of CC&B Provider that you specify Oracle Utilities CC&B data loading output to either files or tables.)

   For more information about CC&B Provider, see "About CC&B Provider" on page 69.
4. "Configuring Mapping Tables" on page 76
5. "Loading CC&B Master Data to Oracle Billing Insight" on page 77

   If you output Oracle Utilities CC&B master data to a flat file (you set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE when configuring CC&B Provider), then also copy the generated master text files to the appropriate load input directory on your Oracle Billing Insight installation, and load the master data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.
6. "Provisioning Oracle Utilities CC&B Account Data in Batch to Oracle Billing Insight" on page 79
7. "Loading Oracle Utilities CC&B Postpay Data to Oracle Billing Insight" on page 80

   If you output Oracle Utilities CC&B billing data to a flat file (you set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE when configuring CC&B Provider), then FTP the generated billing files to the appropriate load input directory on your Oracle Billing Insight installation. The Oracle Service Cloud hosting team will load the billing data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.

About CC&B Provider

You use CC&B Provider to output data from Oracle Utilities CC&B for integration with Oracle Billing Insight. You can configure CC&B Provider to output Oracle Utilities CC&B data in one of two ways, depending on your deployment:

- **Directly to the Oracle Billing Insight database.** You can configure CC&B Provider to first move Oracle Utilities CC&B data to a pre-staging area, transform and load the data into Oracle Billing Insight staging, and then move it into the Oracle Billing Insight production tables. Your deployment must be able to access the Oracle Utilities CC&B database directly from Oracle Billing Insight. This method requires that your implementation of Oracle Billing Insight be able to access the Oracle Utilities CC&B staging database directly.

  To use this method, when you configure CC&B Provider, set the OUTPUT_TYPE property in the CCBProvider.properties file to TABLE.

- **To a text billing file.** CC&B Provider can generate text billing files to a location that you specify. You then upload (FTP) the text billing files to the Oracle Billing Insight database, and load the billing data using the standard Oracle Billing Insight data loading processes described in *Administration Guide for Oracle Billing Insight*. This method provides an integration solution for deployments that cannot access the Oracle Utilities CC&B database directly from Oracle Billing Insight, such as where Oracle Billing Insight is an On-Demand (service cloud) implementation and Oracle Utilities CC&B is On-Premise.

  To use this method, when you configure CC&B Provider, set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE.

For information about how Oracle Utilities CC&B data maps with Oracle Billing Insight, see "Reference for Mapping of Oracle Utilities CC&B and Oracle Billing Insight Data Objects" on page 83.

Depending on the data volume, use the server configuration appropriate for unloading billing data from Oracle Utilities CC&B using CC&B Provider:

- Directly from the Oracle Utilities CC&B 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 Utilities CC&B server.

About the CC&B Provider Load Processes

CC&B Provider consists of a set of Oracle Data Integrator (ODI) processes and a pre-staging schema. The CC&B Provider set of packages, load plans, and scenarios can load billing data from the Oracle Utilities CC&B server into the Oracle Billing Insight database directly or unload the data into text billing files.

The pre-staging schema contains over thirty CC&B tables, a database link to connect to the Oracle Utilities CC&B database, and data loading control tables.

There are two types of billing-related data in Oracle Billing Insight, and each type of data has a dedicated extract and load process for loading Oracle Utilities CC&B data directly to Oracle Billing Insight or to text files:

- **Master.** For master data, an entire dataset is extracted with each load.

TechPubs Draft **Modified**: March 2, 2017 11:54 am

- **Postpay.** The postpay extract and load process from Oracle Utilities CC&B 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.

Assumptions About Oracle Utilities CC&B Data

Certain data and data relationships in Oracle Utilities CC&B are required for integrating Oracle Utilities CC&B with Oracle Billing Insight.

Integration of Oracle Billing Insight with Oracle Utilities CC&B is based on the following data assumptions:

- Balance forward billing is used.
- The top level Person ID of a Person Hierarchy is extracted as the company identity ID.
- A service agreement type (CI_SA_TYPE) with one-off type of charge will map to charge type (the 120 record) in Oracle Billing Insight. A service agreement type with regular recurring type charge will map to service type (210 record) in Oracle Billing Insight.
- Same service agreement type codes (CI_SA_TYPE_L.SA_TYPE_CD) must have the same descriptions (CI_SA_TYPE_L.DFLT_DESCR_ON_BILL and CI_SA_TYPE_L.DESCR).
- A meter could be exchanged for some reason, in which case there will be multiple meters associated with same service at specific time period. This is not supported well. For the new meter you must provision a new service into Oracle Billing Insight and cancel the old meter by provisioning. Also, for the period during which a meter is exchanged, billing data will use the new meter as the service number.
- The meter ID field, BADGE_NBR, is used as the service number in Oracle Billing Insight. For the service agreement (SA) without an associated meter, service agreement ID (SA_ID) will be used as the service number in Oracle Billing Insight. A meter can have multiple unique service types in Oracle Billing Insight. For example, the same meter could be used for both water and waste water charges. For the sub SA's that have the same service type code and are implicitly associated with the same meter by their master SA, these sub SA's will be mapped to one SA in Oracle Billing Insight. There is no master-sub SA concept in Oracle Billing Insight.

For additional information about how Oracle Utilities CC&B data maps with Oracle Billing Insight, see "Reference for Mapping of Oracle Utilities CC&B and Oracle Billing Insight Data Objects" on page 83.

Process of Installing and Configuring Oracle Utilities CC&B Integration Components

To integrate Oracle Utilities CC&B with Oracle Billing Insight, perform the following tasks to install and configure the required integration components for your environment:

1 "Installing Oracle Utilities CC&B Integration Components from Patch" on page 71

2 "Configuring CC&B Provider" on page 71. Follow these steps to configure either direct loading from Oracle Utilities CC&B to Oracle Billing Insight database tables (table output) or loading of Oracle Utilities CC&B billing data to a flat text file (file output).

Installing Oracle Utilities CC&B Integration Components from Patch

You must install Oracle Billing Insight patch 7.1.0.1 to receive the Oracle Utilities CC&B integration components. For the patch installation instructions, see the Oracle Billing Insight 7.1 Patch 1 Release Notes.

Configuring CC&B Provider

Follow these steps to configure CC&B Provider. You can install CC&B Provider on the Oracle Utilities CC&B server or a target server, which can be the Oracle Billing Insight database server or another database server.

This task is a step in "Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing" on page 67.

To configure CC&B Provider

1 Verify your environment configuration:

- If you plan to extract and load billing data from Oracle Utilities CC&B directly to the Oracle Billing Insight database tables, then verify that the ETL environment is installed and configured. For details see *Installation Guide for Oracle Billing Insight*.
- If you plan to extract billing data from Oracle Utilities CC&B to a billing text file (for an On Demand implementation of Oracle Billing Insight), verify that Oracle Data Integrator (ODI) is installed, and the ODI repository and ODI agent are ready on the target database server where you plan to install the CC&B Provider pre-staging schema.

2 Configure the TNS alias:

a On the target database server where you plan to install the pre-staging schema, open the tnsnames.ora file, configure a TNS alias connecting to the CC&B database for your implementation, and then save the file.

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

where:

TechPubs Draft **Modified**: March 2, 2017 11:54 am

- *Host_Name.Example.com* is the CC&B database host name.
- *Port_Num* is the CC&B database listener port.
- *CCBDB* is the CC&B database service name.

CC&B Provider creates a new schema on the target database for pre-staging purposes, which stores simplified versions of Oracle Utilities CC&B tables. When you plan to output Oracle Utilities CC&B billing data to database tables (and specify the OUTPUT_TYPE property value as TABLE in the CCBProvider.properties file), the target database must be the Oracle Billing Insight database.

b If you are installing CC&B Provider directly on the target database server, then make sure that your tnsnames.ora file contains the following alias connecting your target database service.

If you are installing CC&B Provider remotely (not from the target database server), then 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 Utilities CC&B database and the other must connect to the target database.

For example:

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

where:

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

3 Configure the following parameters in the CCBProvider.properties file for your organization. This file is located in the *INTG_HOME*/integration/ccb/db/provider directory. If you specify the OUTPUT_TYPE property as TABLE, then verify that the CC&B Provider ODI project is in the same working repository where the ETL project was imported.

Parameter	Description
OUTPUT_TYPE	The CC&B Provider output type for billing data (postpay) and master data: ■ **FILE.** Outputs Oracle Utilities CC&B data to a text file. The file will be located at the directory specified in the PROV_BILLFILE_LOCATION property. ■ **TABLE.** Inserts Oracle Utilities CC&B into Oracle Billing Insight staging tables. (Requires that the pre-staging schema be installed on the EBILL database.)
PRESTG_WITH_BILLER	Indicates whether to install the pre-staging schema on the Oracle Utilities CC&B database: ■ **Y.** Yes - Install with Oracle Utilities CC&B. (If this property is set to Y, make sure ODI is available on the Oracle Utilities CC&B database server, and the ODI repository and Agent have been created before installing CC&B Provider.) ■ **N.** No - Do not install with Oracle Utilities CC&B. (Set to N if OUTPUT_TYPE is TABLE.)
PROV_BILLFILE_LOCATION	The directory for storing generated billing files. This property is required only if OUTPUT_TYPE is set to FILE.
PROV_TMPFILE_LOCATION	The directory for storing temporary files. This property is required only if OUTPUT_TYPE is set to FILE.
PRESTG_HOST	The pre-staging database host name.
PRESTG_PORT	The pre-staging database port.
PRESTG_SERVICE	The pre-staging database service.
PRESTG_TNS_NAME	The pre-staging TNS (Transparent Network Substrate) name. This property must be the EBILL TNS name if OUTPUT_TYPE is set to TABLE.
PRESTG_USER	The pre-staging user name.
PRESTG_PASSWD	The pre-staging password.
PRESTG_SYS_PASSWD	The pre-staging SYS PASSWORD. This property must be the EBILL database sys password if OUTPUT_TYPE is set to TABLE.
OLAP_USER	The EBILL OLAP user name. This property is required only if OUTPUT_TYPE is set to TABLE.

Parameter	Description
BILLER_ID	The Oracle Utilities CC&B Billing System ID (assigned value).
BILLER_DB_USER	The Oracle Utilities CC&B database user name.
BILLER_DB_PASSWD	The Oracle Utilities CC&B database user password.
BILLER_DB_TNS_NAME	The Oracle Utilities CC&B database TNS alias.
BILLER_DB_SYS_PASSWD	The Oracle Utilities CC&B database SYS account password. This property is required if PRESTG_WITH_BILLER is set to Y.
L_DB_PRESTG_DATA_FILE_LOC	The data tablespace file location.
L_DB_PRESTG_INDX_FILE_LOC	The index tablespace file location.
ODI_HOME_DIR	The ODI home directory, or ODI client home directory if installed remotely.
ODI_USER	The ODI user name (SUPERVISOR).
ODI_PASSWD	The ODI password.
MASTERREP_HOST	The master repository database host.
MASTERREP_PORT	The master repository database port.
MASTERREP_SERVICE	The master repository database service name.
MASTERREP_USER	The master repository schema owner name.
MASTERREP_PASSWD	The master repository schema password.
WORKREP_NAME	The work repository name.
AGENT_BIN_DIR	The bin directory where the standalone agent is located.
AGENT_HOST	The agent host. Specify the hostname instead of the localhost. The agent must be run on the database server where the pre-staging schema is installed.
AGENT_HOST_PORT	The agent port (the default is 20910).
AGENT_PROTOCOL	The agent protocol (the default is http).
AGENT_NAME	The physical agent name (the default is OracleDIAgent1).
LOG_AGENT_NAME	The logical agent name (the default ODIAgent).
XMLFILES_DIR	The directory where the CC&B Provider ODI XML files are located.
BILLING_DATA_START_DT	The start date for extracting billing data (in MM/DD/YYYY format).

4 Set up the pre-staging area:

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

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

If you receive the following error when running Option 3, Configuring working repository, you can ignore it:

```
com.sunopsis.core.SecurityAccessException:
javax.crypto.IllegalBlockSizeException: Input length must be multiple of 16 when
decrypting with padded cipher

ODI can not decrypt the password cipher text which was not encrypted previously
by the current master repository cipher.
```

5 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, CCBProvider, 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, CCBProvider, and DATA_LOAD. Right-click Notification Version 001 and select Regenerate.

Configuring Mapping Tables

You must configure the following mapping tables before you can load data from Oracle Utilities CC&B to Oracle Billing Insight:

- EDX_INTG_SVS_MAP (Service type mapping)
- EDX_INTG_PARTN_COMP_MAP (Company mapping)
- EDX_INTG_PARTN_ACCT_MAP (Account mapping)
- EDX_INTG_PERIOD_DIM (Billing period mapping)

This task is a step in "Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing" on page 67.

To configure the mapping tables for data loading

1. Populate the EDX_INTG_SVS_MAP table with values for service type, either:
   - One-off charge service (O), or
   - Recurring charge service (R).

   For details, see "About the Service Type Mapping Table" on page 76.
2. If you set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE when configuring CC&B Provider, then synchronize the data from the following Oracle Billing Insight tables to the specified mapping table.

Oracle Billing Insight OLAP Database Table	Mapping Table
EDX_RPT_PARTN_COMP_MAP	EDX_INTG_PARTN_COMP_MAP
EDX_RPT_PARTN_ACCT_MAP	EDX_INTG_PARTN_ACCT_MAP
EDX_RPT_PERIOD_DIM	EDX_INTG_PERIOD_DIM

About the Service Type Mapping Table

The service type mapping table, EDX_INTG_SVS_MAP, maps Oracle Utilities CC&B service types to the Oracle Billing Insight one-off charge service types and recurring charge service types. Only service agreements with recurring charge service types (such as Water, Gas, Electric, and so on) will be mapped to a service agreement in Oracle Billing Insight. You populate this table when configuring the pre-staging schema.

Table 11 shows an example of a configured service type mapping table, EDX_INTG_SVS_MAP.

Table 11. Example of a Configured Service Type Mapping Table, EDX_INTG_SVS_MAP

SVC_TYPE_CD Column	SVR_TYPE Column
W	R
O	O

Table 11. Example of a Configured Service Type Mapping Table, EDX_INTG_SVS_MAP

SVC_TYPE_CD Column	SVR_TYPE Column
E	R
G	R
C	R
WW	R
RF	R
M	O

The SVC_TYPE_CD column data comes from the CI_SVC_TYPE table. The SVR_TYPE column is used to flag the service type, either R (Recurring charge) or O (One-off charge).

If the service type changes in Oracle Utilities CC&B, you must update the service type mapping table and load master data again.

Loading CC&B Master Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle Utilities CC&B master data directly to Oracle Billing Insight or to load into a master data file, depending on the output type you chose when installing CC&B Provider (TABLE or FILE). You use this process to load master data in bulk when performing the initial integration of Oracle Utilities CC&B and Oracle Billing Insight.

Master data includes payment types, adjustment types, charge type, usage type, tariff info, service type, unit type, country code and currency type. Master data contains DIM code and corresponding name information.

You load master data when initially integrating Oracle Utilities CC&B and Oracle Billing Insight and again as needed if your master data changes.

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

You must load Oracle Utilities CC&B master data before provisioning account information and loading billing data.

This task is a step in "Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing" on page 67.

To load master data from Oracle Utilities CC&B to Oracle Billing Insight (or to a file)

1. Log in to the Oracle Data Integrator as the repository owner.
2. On the Designer tab, expand Load Plans and Scenarios, expand CCBProvider, 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. Enclose all string values in single quotation marks.
   - **CCBPROVIDER.pStartDate**. Always set to 'NULL'.
   - **CCBPROVIDER.pEndDate.** Always set to 'NULL'.

   Master data will be included in the output, generated in the format specified in your CC&B Provider configuration:
   - **To a Table.** When the output type configured in CC&B Provider is TABLE, master data loads directly into an Oracle Billing Insight DIM table.
   - **To a File.** When the output type configured in CC&B Provider is FILE, master data is generated to a text file in the location you specify in the PROV_BILLFILE_LOCATION parameter. You must upload (FTP) this file to the Oracle Billing Insight database server, where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. For details about standard Oracle Billing Insight ODI processes, see *Administration Guide for 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 are generating Oracle Utilities CC&B output to files, automatic rejection removes files from the PROV_BILLFILE_LOCATION and PROV_TMPFILE_LOCATION directories if files exist.

   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 Utilities CC&B until the current run master data is either published or rejected.

   In the Oracle Billing Insight OLAP.EDX_RPT_CURRENCY_TYPE_DIM table, be sure to select your default currency and set the CURRENCY_IS_DEFAULT field to Y for the selected default currency code.

Provisioning Oracle Utilities CC&B Account Data in Batch to Oracle Billing Insight

This task describes how to load all existing company, account, and service or meter data from the Oracle Utilities CC&B database for initial integration into Oracle Billing Insight using a provided PL/SQL script.

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 Utilities CC&B does not have this restriction.

This task is a step in "Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing" on page 67.

To export a batch provisioning file from Oracle Utilities CC&B database to Oracle Billing Insight

1. Connect to the Oracle Utilities CC&B database.
2. Edit the ExportProvisioningFile.sql script, located in the *INTG_HOME*/integration/ccb/db/provider/ccb/Scripts/ directory. Edit the script to extract service agreements with recurring charges only (as indicated in the service mapping table EDX_INTG_SVS_MAP). Change line 166 to exclude the service type codes where the mapped value in the SVR_CODE field is set to 'O':

   ```
   where svc_type_cd not in ('O', 'M')
   ```

   For details, see "About the Service Type Mapping Table" on page 76.
3. Run the ExportProvisioningFile.sql script. Specify the following parameters:
   a. For the biller ID input parameter, specify the value you configured for the BILLER_ID parameter in the CCBProvider.properties file, such as BILLER_ID=CCB.
   b. For the billing data start date, specify the value you configured for the BILLING_DATA_START_DT parameter in the CCBProvider.properties file.

   For example:

   ```
   sqlplus CCB_schema_owner/CCB_schema_password@CCBDB @ExportProvisioningFile CCB 01/01/2015
   ```
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*.

Loading Oracle Utilities CC&B Postpay Data to Oracle Billing Insight

You use a dedicated extract and load process to load Oracle Utilities CC&B postpay data directly to Oracle Billing Insight or to load into billing data files, depending on the output type you chose when installing CC&B Provider (TABLE or FILE). This process lets you load postpay billing data in bulk when performing the initial integration of Oracle Utilities CC&B and Oracle Billing Insight.

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

This task is a step in "Process of Integrating Oracle Billing Insight With Oracle Utilities Customer Care And Billing" on page 67.

To load Oracle Utilities CC&B postpay 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 CCBProvider, 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. To change a value, uncheck the Latest Value check box, and enter the new value. Enclose all string values in single quotation marks.

   - **CCBPROVIDER.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.
   - **CCBPROVIDER.pEndDate.** When set to 'NULL' (in single quotation marks), the sysdate of the Oracle Utilities CC&B database is used. The date format is YYYYMMDDHHMISS.

   Bills generated between the CCBPROVIDER.pStartDate variable (inclusive) and CCBPROVIDER.pEndDate variable (exclusive) will be included in the output, generated in the format specified in your CC&B Provider configuration:

   - **To a Table.** When the output type configured in CC&B Provider is TABLE, postpay billing data loads directly into a Oracle Billing Insight production tables.
   - **To a File.** When the output type configured in CC&B Provider is FILE, postpay billing data is generated to a text file in the location you specify in the PROV_BILLFILE_LOCATION parameter. You must upload this file to the Oracle Billing Insight database server where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see A*Administration Guide for 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 are generating Oracle Utilities CC&B output to files, automatic rejection removes files from the PROV_BILLFILE_LOCATION and PROV_TMPFILE_LOCATION directories if files exist.

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 Utilities CC&B 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' (in single quotation marks) when loading master or postpay data using the scenarios for Oracle Utilities CC&B 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 CCBProvider, 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 CCBPROVIDER.pEltKey variable in the Variable Values dialog. Use the value from the PRESTG_ETL_KEY field of the prestg.EDX_INTG_PROVIDER_LOAD table.

About Load Management Control Tables

Oracle Billing Insight uses the following load management tables with the Oracle Utilities CC&B 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.

Process of Loading Data from Oracle Utilities CC&B in a Live Production Environment

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

1. If there are changes to the Oracle Utilities CC&B service type 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 76.

2. If there are changes to master data in Oracle Utilities CC&B, then you must load the new data using the master data load ODI process. Master data includes payment types, adjustment types, charge type, usage type, tariff information, service type, unit type, country code, and currency type. For details, see "Loading CC&B Master Data to Oracle Billing Insight" on page 77.

   If you output Oracle Utilities CC&B master data to a flat file (you set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE when configuring CC&B Provider), then also copy the generated master text files to the appropriate load input directory on your Oracle Billing Insight installation, where the Oracle Cloud Service hosting team will load the master data files using the standard ODI load processes described in *Administration Guide for Oracle Billing Insight*.

3. Run the HierarchyCopy job at beginning of the each new billing period. For details, see *Administration Guide for Oracle Billing Insight*.

4. Synchronize new or updated Oracle Utilities CC&B account data in batch using the ProvisioningData job in the Oracle Billing Insight Command Center or by calling the RESTful WebService API. For details, see *Administration Guide for Oracle Billing Insight*.

5. Schedule CC&B Provider to initiate loading postpay data from Oracle Utilities CC&B to Oracle Billing Insight whenever the bill is generated in Oracle Utilities CC&B.

   If you output Oracle Utilities CC&B postpay data to a flat file (you set the OUTPUT_TYPE property in the CCBProvider.properties file to FILE when configuring CC&B Provider), then also copy the generated postpay billing files to the appropriate load input directory on your Oracle Billing Insight installation, where the Oracle Service Cloud hosting team will use the standard Oracle Billing Insight ODI processes to load the postpay billing text file into the Oracle Billing Insight database production tables. For details about the standard Oracle Billing Insight ODI processes, see *Administration Guide for Oracle Billing Insight*.

Reference for Mapping of Oracle Utilities CC&B and Oracle Billing Insight Data Objects

The following topics contain reference information about how Oracle Utilities CC&B data objects map to major Oracle Billing Insight data objects, and how the data is handled during the extraction:

Mapping of Oracle Utilities CC&B Data Objects With Oracle Billing Insight

The process of integrating Oracle Utilities CC&B data requires provisioning of billing objects, such as company, account, and service or meter into Oracle Billing Insight. You can provision this data in either real-time using RESTful WebService APIs or in batch.

Table 12 shows how the required Oracle Billing Insight provisioning objects map to Oracle Utilities CC&B tables, and which Oracle Billing Insight production tables store these provisioning objects, and the RESTful Web Service APIs used to provision the data.

Table 12. Mapping of Oracle Utilities CC&B Data Data With Oracle Billing Insight Provisioning Objects

Oracle Billing Insight Object	Oracle Utilities CC&B Table	Oracle Billing Insight RESTful API	Oracle Billing Insight Production Table Name	Notes
Company	CI_PER CI_PER_NAME CI_PER_PER	/accounts /companies	EDX_OMF_COMPANY	The top level PER_ID of the person hierarchy maps to the company ID. The ENTITY_NAME column maps to the company name.
Account	CI_ACCT	/accounts	EDX_OMF_ACCOUNT	None
Service Agreement	CI_SA CI_MTR	/serviceAgreement	EDX_OMF_SERVICEAGREEMENT	The BADGE_NBR or SA_ID maps to the service agreement number.
Billing Contact	CI_PREM	/accounts	EDX_OMF_CONTACT_INFO	None

TechPubs Draft **Modified**: March 2, 2017 11:54 am

Mapping of Oracle Utilities CC&B Extraction Database Tables With Oracle Billing Insight

Table 13 shows how the Oracle Utilities CC&B database tables used for extraction map to Oracle Billing Insight pre-staging tables used by CC&B Provider, and the CC&B Provider procedures used for each extraction.

Table 13. Mapping of Oracle Utilities CC&B Database Tables With Pre-Staging Tables

Number	Oracle Utilities CC&B Table	Oracle Billing Insight Pre-Staging Table for CC&B Provider	CC&B Provider Procedure
1	CI_TENDER_TYPE_L	CCB_TENDER_TYPE_L	pkg_intg_load_prestg_master.pr_load_tender_type
2	CI_ADJ_TYPE_L	CCB_ADJ_TYPE_L	pkg_intg_load_prestg_master.pr_load_adj_type
3	CI_SA_TYPE	CCB_SA_TYPE	pkg_intg_load_prestg_master.pr_load_sa_type
4	CI_SA_TYPE_L	CCB_SA_TYPE	pkg_intg_load_prestg_master.pr_load_sa_type
5	CI_UOM_L	CCB_UOM_L	pkg_intg_load_prestg_master.pr_load_uom
6	CI_TOU_L	CCB_TOU_L	pkg_intg_load_prestg_master.pr_load_tou
7	CI_COUNTRY_L	CCB_COUNTRY_L	pkg_intg_load_prestg_master.pr_load_country
8	CI_CURRENCY_CD	CCB_CURRENCY_CD	pkg_intg_load_prestg_master.pr_load_currency
9	CI_CURRENCY_CD_L	CCB_CURRENCY_CD	pkg_intg_load_prestg_master.pr_load_currency
10	CI_BILL_MSG_L	CCB_BILL_MSG_L	pkg_intg_load_prestg_master.pr_load_bill_msg
11	CI_RC	CCB_RC	pkg_intg_load_prestg_master.pr_load_rc
12	CI_RC_L	CCB_RC	pkg_intg_load_prestg_master.pr_load_rc
13	C1_CALC_RULE	CCB_CALC_RULE	pkg_intg_load_prestg_master.pr_load_calc_rule
14	CI_ACCT	CCB_ACCT	pkg_intg_load_prestg_billing.pr_load_acct
15	CI_ACCT_PER	CCB_ACCT	pkg_intg_load_prestg_billing.pr_load_acct
16	CI_SA	CCB_SA	pkg_intg_load_prestg_billing.pr_load_sa
17	CI_PER_NAME	CCB_PER	pkg_intg_load_prestg_billing.pr_load_per
18	CI_PER	CCB_PER	pkg_intg_load_prestg_billing.pr_load_per
19	CI_PER_PER	CCB_PER_PER	pkg_intg_load_prestg_billing.pr_load_per_per
20	CI_PREM	CCB_PREM	pkg_intg_load_prestg_master.pr_load_prem
21	CI_MTR_CONFIG	CCB_REG_READ	pkg_intg_load_prestg_billing.pr_load_reg_read
22	CI_MTR	CCB_REG_READ	pkg_intg_load_prestg_billing.pr_load_reg_read
23	CI_BILL	CCB_BILL	pkg_intg_load_prestg_billing.pr_load_bill
24	CI_BILL_CYC_SCH	CCB_BILL	pkg_intg_load_prestg_billing.pr_load_bill
25	CI_BILL_MSGS	CCB_BILL_MSGS	pkg_intg_load_prestg_billing.pr_load_bill_msgs
26	CI_BILL_SA	CCB_BILL_SA	pkg_intg_load_prestg_billing.pr_load_bill_sa
27	CI_FT	CCB_FT	pkg_intg_load_prestg_billing.pr_load_ft
28	CI_PAY	CCB_PAY	pkg_intg_load_prestg_billing.pr_load_pay
29	CI_PAY_EVENT	CCB_PAY	pkg_intg_load_prestg_billing.pr_load_pay
30	CI_PAY_TNDR	CCB_PAY	pkg_intg_load_prestg_billing.pr_load_pay
31	CI_ADJ	CCB_ADJ	pkg_intg_load_prestg_billing.pr_load_adj

Table 13. Mapping of Oracle Utilities CC&B Database Tables With Pre-Staging Tables

Number	Oracle Utilities CC&B Table	Oracle Billing Insight Pre-Staging Table for CC&B Provider	CC&B Provider Procedure
32	CI_BILL_ROUTING	CCB_BILL_ROUTING	pkg_intg_load_prestg_billing.pr_load_bill_routing
33	CI_BILL_RT_TYPE	CCB_BILL_ROUTING	pkg_intg_load_prestg_billing.pr_load_bill_routing
34	CI_BSEG	CCB_BSEG	pkg_intg_load_prestg_billing.pr_load_bseg
35	CI_BSEG_CALC	CCB_BSEG_CALC	pkg_intg_load_prestg_billing.pr_load_bseg_calc
36	CI_BSEG_CALC_LN	CCB_BSEG_CALC_LN	pkg_intg_load_prestg_billing.pr_load_bseg_calc_ln
37	CI_BSEG_READ	CCB_BSEG_READ	pkg_intg_load_prestg_billing.pr_load_bseg_calc_read
38	CI_REG_READ	CCB_REG_READ	pkg_intg_load_prestg_billing.pr_load_reg_read
39	CI_MR	CCB_REG_READ	pkg_intg_load_prestg_billing.pr_load_reg_read
40	CI_BSEG_SQ	CCB_BSEG_SQ	pkg_intg_load_prestg_billing.pr_load_bseg_sq

Mapping of Pre-Staged Oracle Utilities CC&B Master Data With Oracle Billing Insight

Table 14 shows how master data from the Oracle Billing Insight pre-staging tables map to production tables and to file record types.

Table 14. Mapping of Pre-Staged Master Data With Oracle Billing Insight

CC&BProvider Pre-Staging Table	Transformation Procedure	Oracle Billing Insight Production Table	Oracle Billing Insight File Input Record Type
CI_TENDER_TYPE_L	pkg_intg_load_stage_master.pr_load_100_pmt_type	EDX_RPT_PAYMENT_TYPE_DIM	100 - Payment type
CI_ADJ_TYPE_L	pkg_intg_load_stage_master.pr_load_110_adjust_type	EDX_RPT_ADJUSTMENT_TYPE_DIM	110 - Adjustment type
CI_SA_TYPE_L	pkg_intg_load_stage_master.pr_load_120_charge_type	EDX_RPT_CHARGE_TYPE_DIM	120 - Charge type
CI_UOM_L	pkg_intg_load_stage_master.pr_load_160_usage_type	EDX_RPT_USAGE_TYPE_DIM	160 - Usage type
CI_TOU_L	pkg_intg_load_stage_master.pr_load_170_tariff_code	EDX_RPT_TARIFF_DIM	170 - Tariff
CI_SA_TYPE_L	pkg_intg_load_stage_master.pr_load_210_service_type	EDX_RPT_SERVICE_TYPE_DIM	210 - Service type
CI_UOM_L	pkg_intg_load_stage_master.pr_load_220_unit_type	EDX_RPT_UNIT_DIM	220 - Unit
CI_COUNTRY_L	pkg_intg_load_stage_master.pr_load_260_country	EDX_RPT_COUNTRY_DIM	260 - Country
CI_CURRENCY_CD CI_CURRENCY_CD_L	pkg_intg_load_stage_master.pr_load_270_currency	EDX_RPT_CURRENCY_DIM	270 - Currency

Mapping of Pre-Staged Billing Transaction Data from CC&B Provider With Oracle Billing Insight

Table 15 shows how billing transaction data from the CC&B Provider pre-staging tables map to Oracle Billing Insight production tables and file record types. Only critical pre-staging tables are listed for each record type.

Table 15. Mapping of Pre-Staged Billing Transaction Data from CC&B Provider Pre-Staging Tables With Oracle Billing Insight

CC&BProvider Pre-Staging Table	Transformation Procedure	Oracle Billing Insight Table	Oracle Billing Insight File Input Record Type
CCB_BILL CCB_BILL_SA CCB_FT CCB_BILL_MSGS	pkg_intg_load_stage_billing.pr_load_1000_stmt	EDX_RPT_STATEMENT_FACT	1000 - Statement
CCB_BILL CCB_FT CCB_PAY	pkg_intg_load_stage_billing.pr_load_1100_stmtpmt	EDX_RPT_STATEMENT_PAYMENT_FACT	1100 - Statement Payment
CCB_BILL CCB_FT CCB_ADJ CCB_ADJ_TYPE_L	pkg_intg_load_stage_billing.pr_load_1200_stmtadj	EDX_RPT_STATEMENT_ADJUSTMENT_FACT	1200 - Statement Adjustment
CCB_BILL CCB_BILL_ROUTING	pkg_intg_load_stage_billing.pr_load_1300_address	EDX_RPT_ADDRESS_DIM	1300 - Mail-to Address
CCB_BILL CCB_BILL_ROUTING CCB_FT	pkg_intg_load_stage_billing.pr_load_2000_account_a pkg_intg_load_stage_billing.pr_load_2000_account_s	EDX_RPT_ACCOUNT_FACT	2000 - Account charges
CCB_BILL CCB_BSEG CCB_BSEG_CALC CCB_SA CCB_SA_TYPE	pkg_intg_load_stage_billing.pr_load_2100_acctcharge	EDX_RPT_ACCOUNT_CHARGE_FACT	2100 - Account charges at charge type level
CCB_BILL CCB_BSEG CCB_FT CCB_SA CCB_SA_TYPE	pkg_intg_load_stage_billing.pr_load_3000_service	EDX_RPT_SERVICE_FACT	3000 - Service charges
CCB_BILL CCB_BSEG CCB_BSEG_CALC_LN CCB_BSEG_SQ CCB_SA	pkg_intg_load_stage_billing.pr_load_3300_servusage	EDX_RPT_SERVICE_USAGE_FACT	3300 - Service charge at usage level
CCB_BILL CCB_BSEG CCB_BSEG_CALC_LN CCB_SA	pkg_intg_load_stage_billing.pr_load_3400_servtariff	EDX_RPT_SERVICE_TARIFF_FACT	3400 - Service charge at tariff level
CCB_BILL CCB_BSEG CCB_BSEG_CALC CCB_BSEG_CALC_LN CCB_SA	pkg_intg_load_stage_billing.pr_load_3500_servmisc	EDX_RPT_SERVICE_MISC_FACT	3500 - Service misc charge
CCB_BILL CCB_BSEG CCB_PREM	pkg_intg_load_stage_billing.pr_load_5000_prem	EDX_RPT_PREMISE_DIM	5000 - Premise
CCB_BILL CCB_BSEG CCB_BSEG_READ CCB_REG_READ	pkg_intg_load_stage_billing.pr_load_5100_consum	EDX_RPT_CONSUM_FACT	5100 - Consumption

6 Integrating Oracle Billing Insight With CRM Applications

This chapter covers how to configure Oracle Billing Insight to connect to CRM applications. It includes the following topics:

Integrating the Assisted Service Application With CRM or Other Back-Office Systems

You can integrate the Assisted Service application with CRM or other back-office applications to provide customer billing analytics and related data, make payments on behalf of customers, and perform other customer care tasks.

External CRM applications must allocate an HTML iFrame in their Web page to reference the Oracle Billing Insight Assisted Service application and pass the required parameters using a secure connection. The required parameters include the credentials of the agent user created specifically for authenticating an agent session, the identity of the agent user on whose behalf the session is created, and the account number. Once the credentials are authenticated in Oracle Billing Insight, the session to the Assisted Service application is opened for access.

To integrate the Assisted Service application with a CRM application

1. Install the Assisted Service application. For details, see *Installation Guide for Oracle Billing Insight*.
2. Create an agent user for the Assisted Service application to use to log into the Oracle Billing Insight Assisted Service application. You can skip this step if the Assisted Service application is configured as a Single-Sign On with the CRM application.
    a. Deploy the Assisted Service application. For details on deploying the Assisted Service application, see *Installation Guide for Oracle Billing Insight*.
    b. Use the agent boot user, created when you installed the Oracle Billing Insight database, to log into the Assisted Service application and create an agent user. This username and password will be used for the username and password parameters required by the Assisted Service application for logging in and passing data to the Assisted Service application.
3. Replace the following default files with files required for integration:
    - Spring security configuration file:

- **UNIX.** Replace the spring-security.xml file found in the *EDX_HOME*/config/security/csr/ directory with the one found in the *EDX_HOME*/config/security/csr/integration directory.
- **Windows.** Replace the spring-security.xml file found in the *EDX_HOME*\config\security\csr\ directory with the one found in the *EDX_HOME*\config\security\csr\integration directory.

■ Template files:

- **UNIX.** Replace all of the template files found in the *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war_assets\templates\ directory with the files found in the *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war_assets\templates\integration directory.
- **Windows.** Replace all of the template files found in the *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/_assets/templates/ directory with the files found in the *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/_assets/templates/integration directory.

■ Agent login file:

- **UNIX.** Replace the CSR_login.jsp file found in the *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war\ directory with the CSR_login.jsp file found in the *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war\integration directory.
- **Windows.** Replace the CSR_login.jsp file found in the *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/ directory with the CSR_login.jsp file found in the *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/integration directory.

4 Configure the Assisted Service application session to never time out, which turns off the default behavior of sending email reminders for password expiration:

a Open the web.xml file in the following directory:

- **UNIX.** *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war\WEB-INF\
- **Windows.** *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/WEB-INF/

b Change the value of the session-timeout parameter to -1:

```
<session-config><session-timeout>-1</session-timeout></session-config>
```

5 Configure the Assisted Service application user password to never expire:

a Open the security.xma.xml file in the following directory:

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

b Change the daysBeforeExpiration property value to -1:

```
<property name="daysBeforeExpiration">
<value>-1</value> <!--User password will be expired in the given days after
created-->
</property>
```

6 Deploy, or redeploy, the Assisted Service application. For details on deploying the Assisted Service application, see *Installation Guide for Oracle Billing Insight*.

7 On a Web page of the CRM application, create an iFrame large enough to display the content provided by Oracle Billing Insight. Use a secure HTTPS connection to access the Assisted Service application with the following parameters.

Parameter	Required?	Description
username	Y	The name of the user created for logging into the Assisted Service application.
password	Y	The password of the user created for logging into the Assisted Service application.
extCsrId	Y	The name of the third-party user who is sending the request to access the Assisted Service application
accountNumber	Y	The account number. The Assisted Service application retrieves the unique account information from the Oracle Billing Insight database.
billerId	N	The biller Id. If multiple billing systems are involved in this deployment, you must use the billerId parameter with the accountNumber to uniquely identify an account.
localeString	N	The locale string for localization, such as es_ES.
targetUser	Y	The UserID of the end user to view in the integration environment, such as ftown.
targetCompany	Y	The IdentityId of the company to view in the integration environment, such as CUELLE.
brandId	N	The valueCode from the EDX_SYS_BRAND table, such as Take1, for referencing a customized CSS.

It is recommended that you pass the parameters using a POST request, however, you can also use a single, secure URL connection, for example:

https://*hostname*:*port*/*selfservicecsr*/
login?username_username=*username*&password=*password*& extCsrId=*CSRID*&
accountNumber=*accountnumber*

where:

- *hostname* is the name of the server where you installed the Assisted Service application.
- *port* is the port number where you installed the Assisted Service application.
- *selfservicecsr* is the name of the Assisted Service application.

- *username* is the value of the username parameter.
- *password* is the value of the password parameter.
- *CSRID* is the value of the extCsrId parameter.
- *accountnumber* value of the accountNumber parameter.

NOTE: This base URL is different from the one for Oracle Billing Insight version 7.0.

8 (Optional) If you are configuring the Assisted Service application integration for a demo, where the SSL Digital Certificate is not installed on the Assisted Service application server, you can use a non-HTTP connection access URL:

a Open the updated spring-security.xml file, located in the following directory:

- **UNIX.** *EDX_HOME*/config/security/csr/integration
- **Windows.***EDX_HOME*\config\security\csr\integration

b In the channelProcessingFilter bean, change

```
<security:intercept-url pattern="\A/.*\Z" access="REQUIRES_SECURE_CHANNEL"/>
```

to

```
<security:intercept-url pattern="\A/.*\Z" access="ANY_CHANNEL"/>
```

c Specify the following iFrame URL:

https://*hostname*:*port*/*selfservicecsr*/login?username_username=*username*&password=*password*& extCsrId=*CSRID*& accountNumber=*accountnumber*

9 (Optional) To alter the look and feel of the Assisted Service application to match the CRM application, edit the swan_integrate.css file, found in the following location:

- **UNIX.** *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war_assets\swan
- **Windows.** *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/_assets/swan

10 (Optional) To add or remove reports on the Assisted Service application dashboard, edit the dashBoardLayout.jsp file, found in the following location:

- **UNIX.** *EDX_HOME*\J2EEApps\csr\weblogic\csr-app-7.1.ear\csr-web-1.0-SNAPSHOT.war_assets\templates
- **Windows.** *EDX_HOME*/J2EEApps/csr/weblogic/csr-app-7.1.ear/csr-web-1.0-SNAPSHOT.war/_assets/templates

11 (Optional) You can configure a single-sign on (SSO) system for the Assisted Service application. Use the basic methods described in "Configuring Oracle Billing Insight to use a Single Sign-on System" on page 9 applied to the Assisted Service application only, with the following differences:

a Copy the spring-security.xml file, located in the *EDX_HOME*\config\security\csr\ssoIntegration directory, to the following location:

EDX_HOME\config\security\csr

b Synchronize users enrolled in CRM Self-Service to Oracle Billing Insight. You do not need to synchronize the CRM customer service representative users with Oracle Billing Insight. You do not need to run the LoadExternalCSR job.

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 embed 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.

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 the integrated Assisted Service application to avoid clickjacking when accessing from external sites

1 Open the spring-security.xml file, located in the following directory:

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

2 In the following code, change the allowFrom value to your domain name or hostname where the iFrame referencing application is hosted:

```
<bean id="EBillAllowFromStrategy"
class="com.edocs.common.security.intg.EBillAllowFromStrategy">
      <!-- which permits the specified 'uri' to frame integrated csr application.
Support wild card *. Support multiple 'uri' delimited by "|" -->
      <!-- examples: -->
      <!-- www.crm.company.com -->            <!-- for specific host name -->
      <!-- *.company.com -->  <!-- for all hosts of oracle.com domain -->
      <!-- *.company.com|*.group.com --> <!-- for all hosts of oracle.com
      and oracledemos.com domain -->
      <property name="allowFrom" value="*.oracle.com" />
</bean>
```

Configuring the Self-Service Application to Connect to Oracle Service Cloud

You can configure the Self-Service application to integrate with Oracle Service Cloud, displaying FAQ, Ask Question, and Chat features in the user interface, for example.

To integrate Oracle Billing Insight with Oracle Service Cloud

1 Open the insert_servicecloud.jsp file, located in the following directory:

- **UNIX.** *EDX_HOME*/J2EEApps/selfservice/weblogic/selfservice-weblogic-7.1.ear/selfservice-web-1.0-SNAPSHOT.war/_includes
- **Windows.** *EDX_HOME*\J2EEApps\selfservice\weblogic\selfservice-weblogic-7.1.ear\selfservice-web-1.0-SNAPSHOT.war_includes

2 To change the branding or service access URL, update the following lines in the insert_servicecloud.jsp file with your links:

```
src="//www.demo.com/Client.js"

"//www.demo.com/ci/ws/get"
```

The contents of the insert_servicecloud.jsp file are shown here:

```
<%@ include file="/_includes/taglibraries.jsp" %>
<%@ taglib uri="/WEB-INF/edocs-common.tld" prefix="edocs" %>
<script type="text/javascript" src="//www.demo.com/Client.js">
//please replace with valid widget js link in src
</script>
<script type="text/javascript">
Client.Controller.addComponent(
    {
       c: "267;283",
       description: false,
       div_id: "myDiv",
       target: "_blank",
       instance_id: "skw_0",
       module: "KnowledgeSyndication",
```

```
        type: 3
    },
          //please replace with valid widget link below
  "//www.demo.com/ci/ws/get"
  );
</script>

<div class="quicklinks">
       <h3>FAQ</h3>
       <div class="content">
       <div id="myDiv"></div>
       </div>
</div>

<div class="quicklinks">
       <h3><s:text name="serviceCloud.title.moreSupportOptions"/></h3>
       <div class="content">
       <h4><img src="${ctx}/_assets/images/aaq.png" width="32" height="32"/>
         <a href="#"
       onClick="MyWindow=window.open('','MyWindow','width=700,height=800');
       returnfalse;"><s:text
name="serviceCloud.title.askAQuestion"/></a></h4>
       <p class="support"><s:text name="serviceCloud.message.submitAQuestion"/
       ></p>
       <h4><img src="${ctx}/_assets/images/chat.png" width="32" height="32"/>
         <a href="#"
       onClick="MyWindow=window.open('','MyWindow','width=700,height=800');
       returnfalse;"><s:text
name="serviceCloud.title.liveChat"/></a></h4>
       <p class="support"><s:text name="serviceCloud.message.chatDirectly"/></p>
       <h4><img src="${ctx}/_assets/images/community.png" width="32"
       height="32"/>   <a href="#"
       onClick="MyWindow=window.open('','MyWindow','width=700,height=800');
       returnfalse;"><s:text
name="serviceCloud.title.askTheCommunity"/></a></h4>
       <p class="support"><s:text
       name="serviceCloud.message.submitToCommunity"/></p>
  </div>
</div>
```

3 Refresh your browser to view the changes to the sidebar_servicecloud_insert.jsp file. When the EAR file is not deployed in exploded mode, it is necessary to clear the cache and restart the Oracle WebLogic server for the Self-Service domain.

a To clear the cache, go to the Self-Service domain folder and run the following command to stop the Oracle WebLogic server:

- **UNIX.** bin/stopWebLogic.sh
- **Windows.** bin\stopWebLogic.cmd

b Remove files in the following directory:

- **UNIX.** $selfservice_domain/servers/AdminServer/tmp
- **Windows.** $selfservice_domain\servers\AdminServer\tmp

c To restart the Oracle WebLogic server, run the following command:

- **UNIX.** startWebLogic.sh
- **Windows.** startWebLogic.cmd

Index

E

F

H

I

J

L

M

O

P

R

S

T

U

W

X