Oracle® Identity Manager

Connector Guide for IBM RACF Advanced

Release 9.1.0.22

F21285-20

Oracle Identity Manager Connector Guide for IBM RACF Advanced, Release 9.1.0.22

F21285-20

Primary Author: Oracle Corporation

Contents

Preface

What's New in the Oracle Identity Manager Advanced Connector for IBM RACF?

1 About the IBM RACF Advanced Connector

2 Installing and Configuring the LDAP Gateway

3 IBM RACF Connector Deployment on Oracle Identity Manager

List of Figures

List of Tables

Preface

This guide describes the connector that is used to integrate Oracle Identity Manager with IBM RACF.

Audience

This guide is intended for resource administrators and target system integration teams. Installation of the connector components on the mainframe requires experience with IBM RACF and various z/OS technologies and components, including TCP/IP, QSAM (flat files), and z/OS libraries.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For information about installing and using Oracle Identity Governance 12.2.1.4.0, visit the following Oracle Help Center page:

`https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.4/index.html`

For information about installing and using Oracle Identity Governance 12.2.1.3.0, visit the following Oracle Help Center page:

https://docs.oracle.com/en/middleware/idm/identity-governance/12.2.1.3/index.html

For information about Oracle Identity Governance Connectors 12.2.1.3.0 documentation, visit the following Oracle Help Center page:

`https://docs.oracle.com/en/middleware/idm/identity-governance-connectors/12.2.1.3/index.html`

For information about Oracle Identity Manager Connectors 11.1.1 documentation, visit the following Oracle Help Center page:

`http://docs.oracle.com/cd/E22999_01/index.htm`

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in the Oracle Identity Manager Advanced Connector for IBM RACF?

This chapter details updates made to the software and documentation for the Oracle Identity Manager Advanced Connector for IBM RACF.

The updates discussed in this chapter are divided into the following categories:

- Software Updates

  This section describes updates made to the connector software.

- Documentation-Specific Updates

  These include major changes made to the connector documentation. These changes are not related to software updates.

Software Updates

These are the updates made to the connector software.

- Software Updates in Release 9.1.0.22.0
- Software Updates in Release 9.1.0.21.0
- Software Updates in Release 9.1.0.20.0
- Software Updates in Release 9.1.0.19.0
- Software Updates in Release 9.1.0.18.0
- Software Updates in Release 9.1.0.17.0
- Software Updates in Release 9.1.0.16.0
- Software Updates in Release 9.1.0.15.0
- Software Updates in Release 9.1.0.14.0
- Software Updates in Release 9.1.0.13.0
- Software Updates in Release 9.1.0.12.0
- Software Updates in Release 9.1.0.11.0
- Software Updates in Release 9.1.0.11.0
- Software Updates in Release 9.1.0.10.0
- Software Updates in Release 9.1.0.9.0
- Software Updates in Release 9.1.0.8.0
- Software Updates in Release 9.1.0.7.0
- Software Updates in Release 9.1.0.6.0
- Software Updates in Release 9.1.0.5.1

- Software Updates in Release 9.1.0.5.0
- Software Updates in Release 9.1.0.4.0
- Software Updates in Release 9.1.0.3.0
- Software Updates in Release 9.1.0.2.0
- Software Updates in Release 9.1.0.1.0
- Software Updates in Release 9.1.0.0.0

Resolved Issues in Release 9.1.0.22

The following table lists the issues resolved in release 9.1.0.21.0:

Bug Number	Issue	Resolution
33479596	This update provides an ability to check the OIM data to be able to do some custom checks. This gives the customers an ability to write custom transformation logic inside the transform method. This custom transformation is called before the reconciliation event is being triggered.	This issue has been resolved.
35267151	Currently only user’s connected groups are displayed without connect-owner details. By checking the connected groups and respective connect-owners, customers want the ability to model such group connections using OIM child forms. Since this form currently does not have a connect-owner field, the group connection gets provisioned with secure-id defined for Pioneer STC. Hence the customer is looking for a new field (connect-owner) on the child form when adding a user to a group.	This issue has been resolved.

Resolved Issues in Release 9.1.0.21

The following table lists the issues resolved in release 9.1.0.21.0:

Bug Number	Issue	Resolution
35276005	Patch for Spring Framework Vulnerability for all mainframe connectors. Spring Framework version updated is 5.3.27	This issue has been resolved.

Resolved Issues in Release 9.1.0.20

The following table lists the issues resolved in release 9.1.0.20.0:

Bug Number	Issue	Resolution
35146143	FIND ALL GROUPS PARSING IS "COMBINING" TWO GROUPS	This issue has been resolved.

Resolved Issues in Release 9.1.0.19

The following table lists the issues resolved in release 9.1.0.19.0:

Bug Number	Issue	Resolution
35012329	INCORRECT OU IS UPDATED WHEN DELETE USER FOLLOWED BY FULL BATCH RECON FOR USERS	This issue has been resolved.
35002174	Placeholder for voyager looping	This issue has been resolved.

Resolved Issues in Release 9.1.0.18

The following table lists the issues resolved in release 9.1.0.18.0:

Bug Number	Issue	Resolution
34574819	RACF CONN LOOKUP RECON JOB RUNS WIPE OUT CATALOG DISPLAY NAME OF ENTITLEMENT ON EVERY RUN	This issue has been resolved.

Resolved Issues in Release 9.1.0.17

The following table lists the issues resolved in release 9.1.0.17.0:

Bug Number	Issue	Resolution
34395959	LDAP GW ENFORCES REQUIRED ON ACCESS LEVEL FOR RESOURCE PROFILE PROVISIONING	This issue has been resolved.
34419876	SOMETIMES, WHEN WE REVOKE DATASETS FROM SOME USERS, THE REVOKE TASK FAILS	This issue has been resolved.
34319722	SUPPORT for RACF Resources more than 4096 records	This issue has been resolved.
34362872	Patch for Spring Framework Vulnerability for all mainframe connectors. Spring Framework version updated is 5.3.21	This issue has been resolved.

Resolved Issues in Release 9.1.0.16

The following table lists the issues resolved in release 9.1.0.16.0:

Bug Number	Issue	Resolution
34117288	CERTIFICATION OF Z/OS 2.5 FOR RACF CONNECTOR	This issue has been resolved.
34082221	Patch for Spring Framework Vulnerability for all mainframe connectors. Spring Framework version updated is 5.3.19	This issue has been resolved.

Resolved Issues in Release 9.1.0.15

The following table lists the issues resolved in release 9.1.0.15.0:

Bug Number	Issue	Resolution
33762490	ldap gateway fix for "account already exists" error condition	This issue has been resolved.
33668629	PATCH FOR LOG4J ISSUE FOR ALL MAINFRAME CONNECTORS. LOG4J VERSION UPDATED IS 2.17.1	This issue has been resolved.

Resolved Issues in Release 9.1.0.14

The following table lists the issues resolved in release 9.1.0.14.0:

Bug Number	Issue	Resolution
33668629	PATCH FOR LOG4J ISSUE FOR ALL MAINFRAME CONNECTORS. LOG4J VERSION UPDATED IS 2.17.0	This issue has been resolved.

Resolved Issues in Release 9.1.0.13

The following table lists the issues resolved in release 9.1.0.13.0:

Bug Number	Issue	Resolution
33668629	PATCH FOR LOG4J ISSUE FOR ALL MAINFRAME CONNECTORS. LOG4J VERSION UPDATED IS 2.16.0	This issue has been resolved.

Resolved Issues in Release 9.1.0.12.0

The following table lists the issues resolved in release 9.1.0.12.0:

Bug Number	Issue	Resolution
33087587	RECONCILING RACF USER'S DATASET AND PROFILES INTO OIM USING EXTRACTS IN Z/OS V2.4	This issue has been resolved.

Software Updates in Release 9.1.0.11.0

The following are software updates in release 9.1.0.11.0:

- Resolved Issues in Release 9.1.0.11.0

Resolved Issues in Release 9.1.0.11.0

The following table lists the issues resolved in release 9.1.0.11.0:

Bug Number	Issue	Resolution
33033255	RECONCILING RACF USER'S DATASET AND PROFILES INTO OIM	This issue has been resolved.
33350939	RACF Command to remove instdata value is not generated	This issue has been resolved.
33350541	RACF with Default Group Update also triggering a Group Connect command	This issue has been resolved.

Software Updates in Release 9.1.0.10.0

The following are software updates in release 9.1.0.10.0:

- Resolved Issues in Release 9.1.0.10.0

Resolved Issues in Release 9.1.0.10.0

The following table lists the issues resolved in release 9.1.0.10.0:

Bug Number	Issue	Resolution
33203187	RACF - CSDATA attribute cannot be set to null	This issue has been resolved.

Software Updates in Release 9.1.0.9.0

The following are software updates in release 9.1.0.9.0:

- Resolved Issues in Release 9.1.0.9.0

Resolved Issues in Release 9.1.0.9.0

The following table lists the issues resolved in release 9.1.0.9.0:

Bug Number	Issue	Resolution
32577059	Enhanced Alias processing to be able to use IDCAMS JCL	This issue has been resolved.

Software Updates in Release 9.1.0.8.0

The following are software updates in release 9.1.0.8.0:

- Resolved Issues in Release 9.1.0.8.0

Resolved Issues in Release 9.1.0.8.0

The following table lists the issues resolved in release 9.1.0.8.0:

Bug Number	Issue	Resolution
Internal	Memory leak issue in the gateway	This issue has been resolved.
Internal	Race condition issue with RACF batch reconciliations	This issue has been resolved.

Software Updates in Release 9.1.0.7.0

The following are software updates in release 9.1.0.7.0:

- Resolved Issues in Release 9.1.0.7.0

Resolved Issues in Release 9.1.0.7.0

The following table lists the issues resolved in release 9.1.0.7.0:

Bug Number	Issue	Resolution
32498921	CVE-2021-26117: APACHE ACTIVEMQ UPDATE TO AT LEAST 5.16.1 OR 5.15.14	This issue has been resolved.
32054805	CVE-2019-10086: APACHE COMMONS BEANUTILS UPDATE TO AT LEAST 1.9.4	This issue has been resolved.
31974483	CVE-2020-5421: SPRING FRAMEWORK UPDATE TO AT LEAST 5.2.9, 5.1.18, 5.0.19, OR 4.3.29	This issue has been resolved.

Software Updates in Release 9.1.0.6.0

The following are software updates in release 9.1.0.6.0:

- Resolved Issues in 9.1.0.6.0

Resolved Issues in 9.1.0.6.0

The following table lists the issues resolved in release 9.1.0.6.0:

Bug Number	Issue	Resolution
31046304	IPV6 support for RACF.	This issue has been resolved.
31046245	OIM RACF CONNECTOR SUPPORT FOR IPV6.	This issue has been resolved.
32491842	RACF 9.1.0.5 ADDUSER command failed	This issue has been resolved.
32613512	RACF 9.1.0.X None of the RACF delete Events are getting processed completely. All are getting stuck in 'Event Received'	This issue has been resolved.

Software Updates in Release 9.1.0.5.1

The following are software updates in release 9.1.0.5.1:

- Resolved Issues in 9.1.0.5.1

Resolved Issues in 9.1.0.5.1

The following table lists the issues resolved in release 9.1.0.5.1:

Bug Number	Issue	Resolution
32430567	RACF 9.1.0.5 LMTS updated in alternate IT Resource when Scheduled Task is run.	This issue has been resolved.
31829404	RACF 9.1.x - Recon Timezone Issue On OIM scheduled job page , in 'LDAP Time Zone' field enter the Timezone database name value instead of the abbreviated timezone. To find out TimeZone database name value refer to List of tz database time zones . Sample value: America/New_York instead of EST	This issue has been resolved.

Software Updates in Release 9.1.0.5.0

The following are software updates in release 9.1.0.5.0:

- Support for Filtering
- Secondary IT Resource Parameter Added
- Additional Jobs
- Resolved Issues in 9.0.1.5.0

Support for Filtering

Support for filtering has been added for the following jobs:

- RACF Reconcile All Users
- RACF Reconcile All LDAP Users

Secondary IT Resource Parameter Added

Secondary IT Resource Parameter has been added for the following job:

- RACF Reconcile All LDAP Users

Additional Jobs

The following jobs have been added to fetch groups from the mainframe:

- RACF Reconcile Groups to Internal LDAP
- RACF Find All LDAP Groups

Resolved Issues in 9.1.0.5.0

The following table lists the issues resolved in release 9.1.0.5.0:

Bug Number	Issue	Resolution
31598874	RACF 9.1.0.3 - Doesn't have "Support for Filtering" capability on Reconciliation	This issue has been resolved.
29998398	Provided a new job on OIM to fetch Groups from Mainframe to Internal LDAP and another job on OIM to get the Groups from LDAP and load the lookup in OIM.	This issue has been resolved.
30788999	'RACF Reconcile All LDAP Users' doesn't have Secondary IT Resource Parameter.	This issue has been resolved.

Software Updates in Release 9.1.0.4.0

The following are software updates in release 9.1.0.4.0:

- Addition of a New Property in the racf.properties File
- Resolved Issues

Addition of a New Property in the racf.properties File

A new property, `sendAltGrpWithMembershipUpdate`, has been added to the racf.properties file. Use this property to determine if other group attributes can be modified along with the membership update.

See Setting Connection Properties for more information about this property.

Resolved Issues in Release 9.1.0.4.0

The following table lists the issues resolved in release 9.1.0.4.0:

Bug Number	Issue	Resolution
31941015	When voyager tried to write back information to the LDAP gateway, it would fail with the following error: `cn=XXXX,ou=racf,ou=Groups, dc=system,dc=backend cannot be parsed as a valid DN: The provided value "UID" could not be parsed as a valid distinguished name because the last non-space character was part of the attribute name 'UID'. It will be excluded from the set of group members`	This issue has been resolved.

Bug Number	Issue	Resolution
31940817	Duration required to run the RACF Reconcile Users To Internal LDAP schedule job was long resulting in fewer number of users being reconciled.	This issue has been resolved.
31829404	Due to unsuccessful timezone conversion upon a reconciliaton operation, logs displayed timezone in the EST -5 hrs format.	This issue has been resolved.
31753123	Unable to search the key `VOYSDV54` when the logger was present in the `INFO` mode.	This issue has been resolved.
31910630	Error resulted in logs due to prompt for update user/account operation even before completion of create user/account operation.	This issue has been resolved.

32121259	Below lines from the config.ldif file of the LDAP Gateway version 6.8.0 have been removed to increse performance:
	`ds-cfg-index-type: presence -- removed from settings under dn:ds-cfg-attribute=cn,cn=Index,ds-cfg-backend-id=userRoot,cn=Backends,cn=config`
	`ds-cfg-index-type: substring -- removed from settings under dn:ds-cfg-attribute=objectClass,cn=Index,ds-cfg-backend-id=userRoot,cn=Backends,cn=config`
	`ds-cfg-index-type: presence -- removed from settings under dn:ds-cfg-attribute=uid,cn=Index,ds-cfg-backend-id=userRoot,cn=Backends,cn=config`
		This issue has been resolved.

Software Updates in Release 9.1.0.3.0

The following are software updates in release 9.1.0.3.0:

- Support for New Oracle Identity Governance Release
- Addition of a New Parameter in the Pioneer Control File

- Addition of New Informational Messages
- Resolved Issues

Support for New Oracle Identity Governance Release

From this release onward, you can install and use the connector with Oracle Identity Governance 12c PS4 (12.2.1.4.0).

See Table 1-1 for the full list of certified Oracle Identity Governance releases.

Addition of a New Parameter in the Pioneer Control File

A new parameter, `EXPORT_MON`, has been added to the *HLQ*.PIONEER.CONTROL.FILE file. Use this parameter to specify whether you want to monitor user or group imports with messages displayed for every specified number of records. By default, the value of this parameter is set to `NO`.

See Configuring the Provisioning Agent for more information about this parameter and the permitted values.

Addition of New Informational Messages

The IDFRPI066 and IDFRPI067 informational message IDs have been added as a result of introduction of the EXPORT_MON parameter in the *HLQ*.PIONEER.CONTROL.FILE file.

See Pioneer Messages for the message IDs and its corresponding text.

Resolved Issues in Release 9.1.0.3.0

The following table lists the issues resolved in release 9.1.0.3.0:

Bug Number	Issue	Resolution
30955398	The number of records processed by the RACFROU batch job was logged incorrectly in the *HLQ*`.PIONEER.IMPORTU.FILE` dataset. The count in the *HLQ*`.PIONEER.IMPORTU.FILE` dataset was double the number of records processed.	This issue has been resolved.
31009468	When you updated the display name of an account in the target system, only the value of the sn attribute in LDAP was updated. The cn value was not updated.	This issue has been resolved.
31046369	The IT Resource field was not configured as a key field for reconciliation matching.	This issue has been resolved.

Software Updates in Release 9.1.0.2.0

The following is a software update in release 9.1.0.2.0:

Customizing the IRREVX01 RACF Command Exit

From this release onward, you can integrate any custom version of the RACF command exit (IRREVX01) in your environment with the connector-specific version of the IRREVX01 exit (module name: IDFINSTX). The connector installation package includes sample files that let you add your modifications and then integrate different versions of the IRREVX01 exit.

See Customizing the Reconciliation Exit for more information about working with custom reconciliation exit routines.

Software Updates in Release 9.1.0.1.0

The following are the software updates in release 9.1.0.1.0:

- Transformation of LDAP Gateway Attributes
- Running Multiple Instances of the LDAP Gateway on the Same Host
- CRUD Operations on RACLIST Resource Classes

Transformation of LDAP Gateway Attributes

By including transformation rules within the `*LDAP_INSTALL_DIR*/conf/customer-configuration.properties` file, you can configure the LDAP gateway to transform the gateway attributes in search results.

See Configuring Transformation of the LDAP Gateway Attributes for more information on the transformation rules to include and its format.

Running Multiple Instances of the LDAP Gateway on the Same Host

From this release onward, you can run multiple instances of the LDAP Gateway on the same host.

See Configuring Multiple Instances of the LDAP Gateway for more information on configuring and running multiple gateway instances in your environment.

CRUD Operations on RACLIST Resource Classes

The connector provides support for performing CRUD operations on RACLIST resource classes. To support this feature, the "supportedResourceClasses" property has been added to racf.properties file that is located in the `*LDAP_INSTALL_DIR*/conf` directory.

See the "supportedResourceClasses" property in Table 2-2 for more information on configuring the connector for this feature.

Software Updates in Release 9.1.0.0.0

The following are the software updates in release 9.1.0.0.0:

- Support for New Oracle Identity Governance Release
- Support for New Target System Version
- Detailed Audit Logs
- Support for High Availability and Disaster Recovery in the LDAP Gateway
- Support for Reconciling Space Character in TSO Command
- Dynamic Allocation of the Voyager DEBUGOUT Parameter
- Support for RACLINK Command
- Support for a New Diagnostic Tool
- Addition of New Parameters to Pioneer and Voyager
- Support for 256-Bit TCP/IP Encryption

Support for New Oracle Identity Governance Release

From this release onward, the connector can be installed and used on Oracle Identity Governance release12.2.1.3.0. Be sure to download and apply the 28682376 and 29133050 mandatory patches from My Oracle Support.

Support for New Target System Version

From this release onward, the you can install and use the connector with IBM RACF on z/OS 2.3.

Detailed Audit Logs

From this release onward, the connector provides a LOGGERX module that you can configure for detailed debug level log information on the Pioneer and Voyager agents. This detailed logging provides additional auditing and monitoring capabilities for your target system. In addition, you can choose to print or suppress log messages.

See Configuring Logging for more information.

Support for High Availability and Disaster Recovery in the LDAP Gateway

From this release onward, the LDAP gateway supports high availability and disaster recovery when you use OpenDS as the backend.

Support for Reconciling Space Character in TSO Command

From this release onward, the connector reconciles TSO commands that contain space characters.

Dynamic Allocation of the Voyager DEBUGOUT Parameter

From this release onward, the connector dynamically allocates the value of the DEBUGOUT parameter for Voyager.

Support for RACLINK Command

The connector can now issue RACLINK (administer user ID associations) commands for certain provisioning operations.

Support for a New Diagnostic Tool

From this release onward, a new diagnostic tool, ENVINFO, for the mainframe agents Pioneer and Voyager is available for use.

Addition of New Parameters to Pioneer and Voyager

The EBCDIC_COUNTRY_CODE and EBCDIC_TILDE_CHR parameters have been added to Pioneer and Voyager. You must use these parameters in conjunction with the gateway configuration property `_mainframeCodePage_` that is available within the racf.properties file.

Note that the value of the EBCDIC_TILDE_CHR parameter must be the HEX value 'BC' on the target system if it is used.

Support for 256-Bit TCP/IP Encryption

The connector supports TCP/ IP with 256-bit encryption between the LDAP gateway and mainframe agents Pioneer and Voyager.

Documentation-Specific Updates

These are the updates made to the connector documentation.

Documentation-Specific Updates in Release 9.1.0.22.0

- **ER 35267151 - RACF Provision and read connect owner group attribute**

  This provides the ability to view the connect-owner field for a user with connected groups listing. Currently only user's with connected groups are displayed without connect-owner details. While viewing the connected groups and the respective connect-owners, the ability to model such group connections using OIM child forms is required. This form currently does not have a connect-owner field, and the group connection is provisioned with secure-id defined for Pioneer STC. Hence there is a need for a new field (connect-owner) on the child form while adding a user to a group. The following are the Impacted Components:

  - RACF Advanced Custom Adapter – OIM
  - RACF LDAP gateway Changes

  IDF OIM RACF Advanced Custom Adapter changes are as follows:

  Reconciliation changes

  1. While reconciling all user’s data, the OIM RACF Advanced custom adapter will be expecting the Connect-owner attribute along with each user's connected group.

2. The parsing logic would be enhanced to read each connected group's connect-owner attribute and display it on the user profile.

Provisioning changes

1. While adding/connecting a user to a group using OIM child form/Entitlement, currently only the group display name is displayed as part of the lookup and the user can select the group to connect.
2. This child form/entitlement is enhanced to provide an additional field **connect-owner** which allows user details to populate once the group name is selected from the lookup values. This new connect-owner field being free-form, accepts any valid group name to be specified during connect operation.
3. Connect-owner field to be defined on the OIM child form is an optional field. If the value is not populated, the default behavior of setting the secure id as connect-owner will continue to work.

RACF Reconcile All LDAP Users

The RACF Reconcile All LDAP Users scheduled task is used to reconcile users from the internal LDAP store to Oracle Identity Manager. When you configure this scheduled task, it runs at specified intervals and fetches a list of users within the internal LDAP store and reconciles these users to Oracle Identity Manager. The following table provides the attribute values to be updated if ConnectOwner needs to be reconciled to users.

Attribute	Description
MultiValuedAttributes	Enter a comma-separated list of multivalued attributes that you want to reconcile. Do not include a space after each comma. Sample value: attributes,memberOf,groupConnectOwner

RACF Reconcile All Users

The RACF Reconcile All Users scheduled task is used to reconcile user data in the target resource (account management) mode of the connector. This scheduled task runs at specified intervals and fetches create or modify events on the target system for reconciliation. The following table provides the attribute values to be updated if ConnectOwner needs to be reconciled to users.

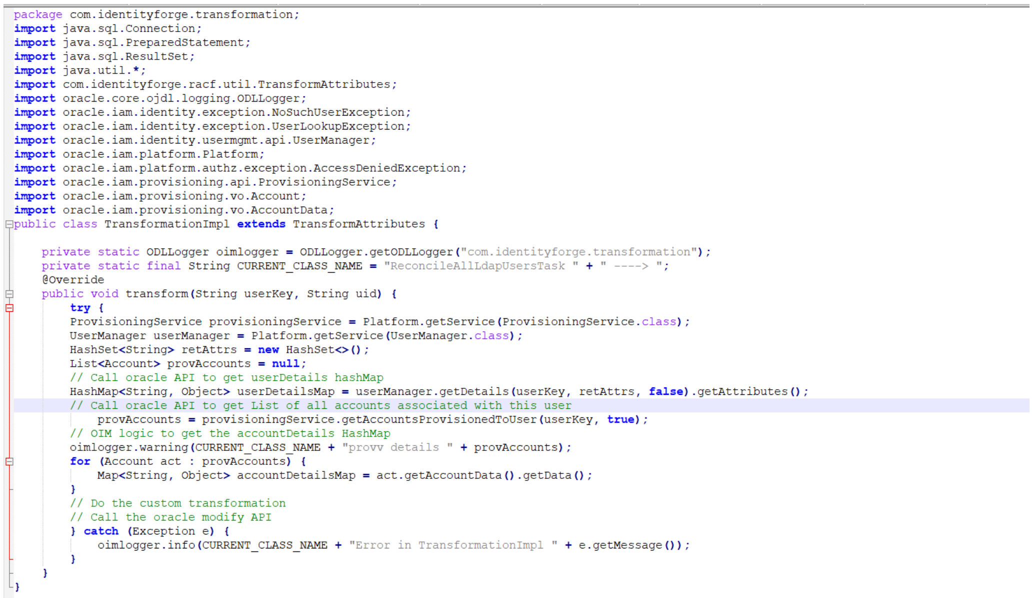
Attribute	Description
MultiValuedAttributes	Enter a comma-separated list of multivalued attributes that you want to reconcile. Do not include a space after each comma. **Sample value** attributes,memberOf,groupConnectOwner

ER 33479596 - RACF Reconciliation with Custom Transformation

This update provides the ability to customize transformation during RACF reconciliation to derive the values for some process form attributes on OIM. To check the OIM data and to be able to do some custom checks and to write custom transformation logic inside the transform method. This custom transformation is called before the reconciliation event is being triggered. The following are the steps for the changes to be done to the IDF OIM RACF Advanced Custom Adapter:

1. Download the release build package - *IBM_RACF_Adv_9.1.0.22.zip*.
2. Unzip the release build package.

3. Unzip the **IBM_RACF_Adv_Connector.zip** file.
4. Go to `IBM_RACF_Adv_Connector\transformation` folder.
5. Refer the `README.md` for further instructions.
   Below is the template code snippet for `TransformationImpl.java` class.



```
package com.identityforge.transformation;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.*;
import com.identityforge.racf.util.TransformAttributes;
import oracle.core.ojdl.logging.ODLLogger;
import oracle.iam.identity.exception.NoSuchUserException;
import oracle.iam.identity.exception.UserLookupException;
import oracle.iam.identity.usermgmt.api.UserManager;
import oracle.iam.platform.Platform;
import oracle.iam.platform.authz.exception.AccessDeniedException;
import oracle.iam.provisioning.api.ProvisioningService;
import oracle.iam.provisioning.vo.Account;
import oracle.iam.provisioning.vo.AccountData;
public class TransformationImpl extends TransformAttributes {

    private static ODLLogger oimlogger = ODLLogger.getODLLogger("com.identityforge.transformation");
    private static final String CURRENT_CLASS_NAME = "ReconcileAllLdapUsersTask " + " ----> ";
    @Override
    public void transform(String userKey, String uid) {
        try {
        ProvisioningService provisioningService = Platform.getService(ProvisioningService.class);
        UserManager userManager = Platform.getService(UserManager.class);
        HashSet<String> retAttrs = new HashSet<>();
        List<Account> provAccounts = null;
        // Call oracle API to get userDetails hashMap
        HashMap<String, Object> userDetailsMap = userManager.getDetails(userKey, retAttrs, false).getAttributes();
        // Call oracle API to get List of all accounts associated with this user
            provAccounts = provisioningService.getAccountsProvisionedToUser(userKey, true);
        // OIM logic to get the accountDetails HashMap
        oimlogger.warning(CURRENT_CLASS_NAME + "provv details " + provAccounts);
        for (Account act : provAccounts) {
            Map<String, Object> accountDetailsMap = act.getAccountData().getData();
        }
        // Do the custom transformation
        // Call the oracle modify API
        } catch (Exception e) {
            oimlogger.info(CURRENT_CLASS_NAME + "Error in TransformationImpl " + e.getMessage());
        }
    }
}
```

Documentation-Specific Updates in Release 9.1.0.21.0

NA

Documentation-Specific Updates in Release 9.1.0.20.0

NA

Documentation-Specific Updates in Release 9.1.0.19.0

NA

Documentation-Specific Updates in Release 9.1.0.18.0

NA

Documentation-Specific Updates in Release 9.1.0.17.0

The following documentation-specific update has been made in revision "17" of the guide:

Table 2-2 updated with details for the following properties: `resourceReadFromStaticFile`.

Documentation-Specific Updates in Release 9.1.0.16.0

NA

Documentation-Specific Updates in Release 9.1.0.15.0

NA

Documentation-Specific Updates in Release 9.1.0.14.0

NA

Documentation-Specific Updates in Release 9.1.0.13.0

NA

Documentation-Specific Updates in Release 9.1.0.12.0

The following documentation-specific update has been made in revision "12" of the guide:

- RACF Reconcile Datasets To Internal LDAP and RACF Reconcile Resources To Internal LDAP updated with details for z/OS 2.3 and 2.4.
- Performing Full Reconciliation updated with details for z/OS 2.4.
- Table 2-2 updated with details for the following properties: `batchReconForDatasetAndResourceUsingExtract`.
- Steps to generate a custom secret key added to Customizing AES Encryption Key.

Documentation-Specific Updates in Release 9.1.0.11.0

The following documentation-specific update has been made in revision "11" of the guide:

- `<HLQ>.JCLLIB.RACFRC` added to Installing the Mainframe Agents.
- RACF Reconcile Datasets To Internal LDAP and RACF Reconcile Resources To Internal LDAP scheduled tasks added.
- RACF Reconcile All LDAP Users updated with details for `MultiValuedAttributes`.
- Table 2-2 updated with details for the following properties: `updateDefaultGroupFailsIfUserIsNotMember`, `batchReconUpdatesUsersAndGroupsWithDatasetAccess`, `batchReconUpdatesUsersAndGroupsWithResourceAccess`, and `batchReconUpdateEntriesOnlyIfModified`

Documentation-Specific Updates in Release 9.1.0.10.0

The following documentation-specific update has been made in revision "10" of the guide:

- New section Configuring Memory Pool Settings for LDAP Gateway v8.x.x added.

Documentation-Specific Updates in Release 9.1.0.9.0

The following documentation-specific update has been made in revision "10" of the guide:

The `POST_PROC_ALIAS` and `JWAIT` parameters have been updated in Table 4-4

The following messages were added to Pioneer Messages : `IDFRPI050 IDFRPI051 IDFRPE025 IDFRPW010 IDFRPW011`.

Attribute **LDAP Time Zone** description amended to **OIM Server Time Zone** in Table 5-8 since OIM Server TZ is required rather than LDAP Server TZ as previously documented.

Documentation-Specific Updates in Release 9.1.0.6.0

The following documentation-specific update has been made in revision "09" of the guide:

A new parameter called IP has been added to Table 4-4

A new parameter called IP has been added to Table 4-5

The LDAP Time Zone attribute description in Table 5-8 has been updated.

Documentation-Specific Updates in Release 9.1.0.5.0

The following documentation-specific update has been made in revision "08" of the guide:

A new property called Filter has been added to Table 5-5 and Table 5-8

Scheduled Tasks for Lookup Field Synchronization has been updated to include RACF Reconcile Groups To Internal LDAP and RACF Find All LDAP Groups.

Documentation-Specific Updates in Release 9.1.0.4.0

There are no documentation-specific updates in revision "07" of the guide.

The following documentation-specific update has been made in revision "06" of the guide:

A new property called sendAltGrpWithMembershipUpdate has been added to Table 2-2.

Documentation-Specific Updates in Release 9.1.0.3.0

There are no documentation-specific updates in revision "05" of the guide.

Documentation-Specific Updates in Release 9.1.0.2.0

The following documentation-specific update has been made in revision "04" of the guide:

Activating and Deactivating Reconciliation Exits has been updated.

Documentation-Specific Updates in Release 9.1.0.1.0

The following documentation-specific updates have been made in revision "03" of the guide:

- The "Target systems" row of Table 1-1 has been updated to include z/OS 2.4.
- Configuring Memory Pool Settings has been added.

The following documentation-specific updates have been made in revision "02" of the guide:

- Table 2-2 has been updated to include the agentMetaRecon and agentCachingRecon properties.
- The name of the file and its location for managing LDAP Gateway logging operations has been updated in Enabling Logging for the LDAP Gateway.
- The following topics have been updated to clarify the encryption requirement for the connector:
  - The "Infrastructure requirement for the message transport layer between Oracle Identity Manager and the mainframe environment" row of Table 1-1
  - Description of the Message Transport Layer component in Understanding the Connector Components
  - Encrypted Communication Between the Target System and Oracle Identity Manager
  - The "Message Transport Layer" row of Table 4-1
- Table 3-1 has been updated to include missing parameters.
- Information about Environmental Settings and Other Requirements has been updated in Installation Requirements for Agents.

- Installing the Mainframe Agents has been updated to remove jobstreams that are not related to this connector.
- Configuring the Provisioning Agent has been updated.
- Table 5-1 has been updated to include info about the Find All Resources scheduled task.
- Pioneer Post-Processing Commands has been updated.

Documentation-Specific Updates in Release 9.1.0.0.0

The following documentation-specific update has been made in revision "01" of the guide:

This is the first release of the connector in this release track. Therefore, there are no documentation-specific updates in this release.

1 About the IBM RACF Advanced Connector

The IBM RACF Advanced connector integrates Oracle Identity Manager with a RACF target system.

This guide discusses the connector that enables you to use IBM RACF as a managed (target) resource of identity data for Oracle Identity Manager.

This chapter contains the following topics:

1.1 Introduction to the Connector

Oracle Identity Manager automates access rights management, security, and provisioning of IT resources. Oracle Identity Manager connectors are used to integrate Oracle Identity Manager with external, identity-aware applications. The advanced connector for IBM RACF provides a native interface between IBM RACF installed on an IBM z/OS mainframe and Oracle Identity Manager. The connector functions as a trusted virtual administrator on the target system, performing tasks related to creating and managing users.
The connector allows information about users created or modified directly on the target system to be reconciled into Oracle Identity Manager. In addition, you can use Oracle Identity Manager to perform provisioning operations on the target system.

In the IBM RACF context, the term **user profile** is synonymous with **user account**. If IBM RACF is configured as a target resource, then user profiles on IBM RACF correspond to accounts or resources assigned to OIM Users.

1.2 Certified Components

These are the software components and their versions required for installing and using the connector.

Table 1-1 Certified Components

Item	Requirement
Oracle Identity Manager or Oracle Identity Governance	You can use one of the following releases of Oracle Identity Manager or Oracle Identity Governance: • Oracle Identity Governance 14c (14.1.2.1.0) **Note:** For Oracle Identity Governance 14c (14.1.2.1.0) the RACF Advanced 9.1.0.24 Patch 37497000 or later must be installed. You can download the patch from My Oracle Support. • Oracle Identity Governance 12c PS4 (12.2.1.4.0) • Oracle Identity Governance 12c (12.2.1.3.0) with the 28682376 and 29133050 mandatory patches installed. You can download the mandatory patches from My Oracle Support. • Oracle Identity Manager 11*g* release 2 PS3 (11.1.2.3.0) with last version of OIM RACF v9.1.0.10 supported.
JDK	The JDK version can be one of the following: • For Oracle Identity Governance release 12.2.1.3.0 or later, use JDK 1.8.0_131+ . • For Oracle Identity Manager release 11.1.2.*x* or later, use JDK 1.6 update 31 or later.
Target systems	IBM RACF on z/OS 2.5 and 3.1
Infrastructure requirement for the message transport layer between Oracle Identity Manager and the mainframe environment	The infrastructure requirements can be one of the following: • TCP/IP • z/OS AES encryption
Target system user account for reconciliation and provisioning operations	IBM Authorized Program Facility (APF) authorized account with System Administrators privileges
Product Libraries	The following are the product libraries: • z/OS standard Load Libraries. These libraries must be APF authorized. • IRREVX01 resides in the Product Library.
Pioneer and Voyager	Pioneer and Voyager are written in single thread LE Cobol. They were developed to run above the 16M line. Options that can adversely affect these STCs are LE run options: ALL31(OFF) instead of ON STACK(,,,BELOW,,) instead of STACK(,,,ANYWHERE,,)
LDAP Gateway	The computer hosting the LDAP Gateway must run the following software: • Operating system: Microsoft Windows Server 2012, or Red Hat Enterprise Linux 7 (64-bit) • Oracle Java JRE 1.8 or 1.7

1.3 Certified Languages

These are the languages that the connector supports.

- Arabic
- Chinese (Simplified)
- Chinese (Traditional)
- Danish
- English
- French
- German
- Italian
- Japanese
- Korean
- Portuguese (Brazilian)
- Spanish

1.4 Connector Architecture

Connectors require certain architecture consisting of Gateways and Provisioning Agent.

This section contains the following topics:

- Understanding the Connector Components
- Understanding the Connector Operations

1.4.1 Understanding the Connector Components

The IBM RACF Advanced connector contains the following components:

- **LDAP Gateway**: The LDAP Gateway receives instructions from Oracle Identity Manager in the same way as any LDAP version 3 identity store. These LDAP commands are then converted into native commands for IBM RACF and sent to the Provisioning Agent. The response, which is also native to IBM RACF, is parsed into an LDAP-format response and returned to Oracle Identity Manager.

  During reconciliation, the LDAP Gateway receives event notification, converts the events to LDAP format, and then forwards them to Oracle Identity Manager, or events can be stored in the LDAP Gateway internal store and pulled into Oracle Identity Manager by a scheduled task.
- **Provisioning Agent (Pioneer)**: The Pioneer Provisioning Agent is a mainframe component. It receives native mainframe IBM RACF identity and authorization change events from the LDAP Gateway. These events are processed against the IBM RACF authentication repository, in which all provisioning updates from the LDAP Gateway are stored. The response is parsed and returned to the LDAP Gateway.

> **Note:**
>
> At some places in this guide, the Provisioning Agent is referred to as **Pioneer.**

- **Reconciliation Agent (Voyager):** The Reconciliation Agent captures mainframe events by using exits, which are programs run after events in IBM RACF are processed. These events include the ones generated at TSO logins, the command prompt, batch jobs, and other native events. These events are stored in the subpool cache area that is established by a supplied, standard z/OS procedure (STARTUP). The Reconciliation Agent captures these events, transforms them into LDAPv3 protocol notification messages, and then sends them to Oracle Identity Manager through the LDAP Gateway.

> **Note:**
>
> At some places in this guide, the Reconciliation Agent is referred to as **Voyager.**

- **Message Transport Layer**: This connector supports a message transport layer by using the TCP/IP protocol, which is functionally similar to proprietary message transport layer protocols. In addition, the connector provides AES encryption for messages sent and received through the transport layer.

  The AES encryption is performed using 128-bit cryptographic keys. In addition, Encryption and Decryption programs are supplied in the Distribution Load Library. The encryption or decryption does not require any network software or hardware.

1.4.2 Understanding the Connector Operations

Provides an overview of the provisioning and reconciliation operations.

This section contains the following topics:

- Full Reconciliation Process
- Initial LDAP Population and Reconciliation Process
- Provisioning Process

1.4.2.1 Full Reconciliation Process

Full reconciliation involves fetching existing user profile data from the mainframe to Oracle Identity Manager.

If you configure the target system as a target resource, then this user profile data is converted into accounts or resources for OIM Users.

The following is a summary of the full reconciliation process:

1. You set values for the attributes of the RACF Reconcile All Users scheduled task.
2. You run the scheduled task. The task sends a search request to the LDAP Gateway.
3. The LDAP Gateway encrypts the search request and then sends it to the Provisioning Agent on the mainframe.
4. The Provisioning Agent encrypts user profile data received from RACF and then passes this data to the LDAP Gateway.

5. The LDAP Gateway decrypts the user profile data. If the user profile data does not include any changes when compared to the OIM user's existing resource data, then the event is ignored and reconciliation continues with the next user on the target system. If the user profile data includes a change, then the LDAP Gateway passes the data on to Oracle Identity Manager.
6. The user profile data is converted into accounts or resources for OIM Users.

1.4.2.2 Initial LDAP Population and Reconciliation Process

This reconciliation process allows for a faster reconciliation based on an Extracted file configured on the Mainframe that will be used to populate the internal LDAP store, which OIM can then use a normal scheduled task to reconcile all the data to Oracle Identity Manager.

The following is a summary of the full reconciliation process:

> **Note:**
>
> For detailed instructions on how to perform full reconciliation, see Performing Full Reconciliation.

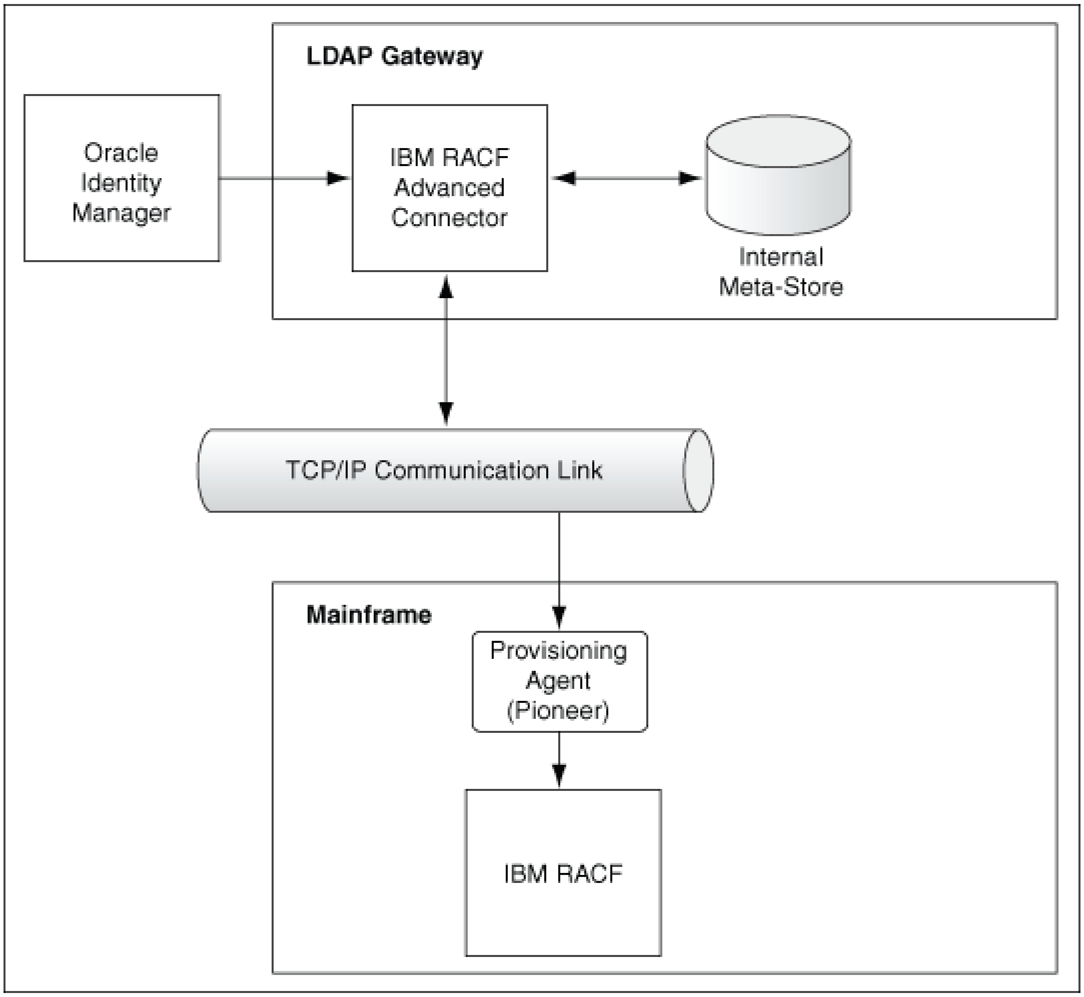
1. Use IBM utility to EXTRACT user data to a file.
2. Configure Pioneer to use this file when needed.
   Once this file has been created and used by OIM it will become stale and must be deleted. The file can be generated again if needed for re-populating or updating the Internal LDAP for Oracle Identity Manager to reconcile the latest data.
3. Once the above file is generated, run the RACF Reconcile Users To Internal LDAP scheduled task to populate the LDAP Gateway internal store.
4. After the LDAP Gateway internal store is populated, run the RACF Reconcile All LDAP Users scheduled task with one of the following settings:
   a. To reconcile all users, set the value of the Last Modified Timestamp attribute to 0.
   b. To reconcile all users that have changed since that date, set the value of the Last Modified Timestamp attribute to a date range.

   > **Note:**
   >
   > If the _internalEnt_ property, located in the *LDAP_INSTALL_DIR*/conf/racf.properties file, is set to `true`, then the LDAP internal store will also be populated on an ongoing basis by the "real-time" event capture using Voyager and the EXITs. So after initial population and reconciliation the process will still continue to use the RACF Reconcile All Ldap Users Task scheduled job using a Date range to reconcile these "real-time" event changes from data captured in the LDAP internal store.

1.4.2.3 Provisioning Process

Figure 1-1 shows the flow data during provisioning.

Figure 1-1 Provisioning Process


LDAP Gateway
Oracle Identity Manager
IBM RACF Advanced Connector
Internal Meta-Store
TCP/IP Communication Link
Mainframe
Provisioning Agent (Pioneer)
IBM RACF

The following is a summary of the provisioning process:

1. Provisioning data submitted from Oracle Identity Self Service is sent to the LDAP Gateway.
2. The LDAP Gateway converts the provisioning data into mainframe commands, encrypts the commands, and then sends them to the mainframe computer over TCP/IP.
3. The Provisioning Agent installed on the mainframe computer decrypts and converts the LDAP message from ASCII to EBCDIC.
4. The Provisioning agent executes the commands, runs them on the mainframe and within the Pioneer STC (Started Task) using the RACF API (IRRSEQ00).
5. The Provisioning Agent converts the RACF API output to ASCII and encrypts the message prior to sending back to the LDAP Gateway.
6. The outcome of the operation on the mainframe is displayed in Identity Self Service. A more detailed message is recorded in the connector log file.

1.5 Connector Features

The features of the connector are discussed in the following topics:

- Full and Incremental Reconciliation
- Encrypted Communication Between the Target System and Oracle Identity Manager
- High Availability Feature of the Connector

1.5.1 Full and Incremental Reconciliation

After you deploy the connector, you can perform full reconciliation to bring all existing user data from the target system to Oracle Identity Manager. After the first full reconciliation run, change-based or incremental reconciliation is automatically enabled and active. Incremental reconciliation is a real-time process. User changes on the target system are directly sent to Oracle Identity Manager or stored in the LDAP Gateway internal store.

You can perform a full reconciliation run at any time. See Configuring Incremental Reconciliation and Performing Full Reconciliation for more information.

1.5.2 Encrypted Communication Between the Target System and Oracle Identity Manager

AES-128 encryption is used to encrypt data that is exchanged between the LDAP Gateway, and the Reconciliation and Provisioning Agents on the Mainframe. This encryption is taken care by the Mainframe agents.

1.5.3 High Availability Feature of the Connector

The following are component-failure scenarios and the response of the connector to each scenario:

- **Scenario 1: The Reconciliation Agent is running and the LDAP Gateway stops responding**
  1. The Reconciliation Agent stops sending messages (event data) to the LDAP Gateway.
  2. Messages that are not sent are stored in the subpool cache.
  3. When the LDAP Gateway is brought back online, the Reconciliation Agent reads data from the subpool cache and then sends messages to the LDAP Gateway.
- **Scenario 2: The LDAP Gateway is running and the Reconciliation Agent stops responding**
  1. Event data is sent to the subpool cache.
  2. When the Reconciliation Agent is brought back online, it reads data from the subpool cache and then sends messages to the LDAP Gateway.

  > **Note:**
  >
  > During SHUTDOWN, there is a possibility that events that had been sent to the LDAP might be saved and re-sent again once the Agent is brought back online. This is to ensure no data lose and this process will re-list the event data to provide the most current view.

- **Scenario 3: The LDAP Gateway is running and the mainframe stops responding**
  1. Messages that are in the subpool cache are written to disk.

2. When the mainframe is brought back online, event data written to disk is again stored in the subpool cache.
3. The Reconciliation Agent reads data from the subpool cache and then sends messages to the LDAP Gateway.

> **Note:**
>
> During SHUTDOWN, there is a possibility that events that had been sent to the LDAP might be saved and re-sent again once the Agent is brought back online. This is to ensure no data lose and this process will re-list the event data to provide the most current view.

- **Scenario 4: The LDAP Gateway is running and the Provisioning Agent or mainframe stops responding**

  The process task that sends provisioning data to the LDAP Gateway retries the task.

- **Scenario 5: The subpool is stopped by an administrator**

  If the subpool is stopped by an administrator, then it shuts down the Reconciliation Agent, thereby destroying any messages that are not transmitted. However, the messages in the AES-encrypted file are not affected and can be recovered.

1.6 Connector Objects Used During Reconciliation and Provisioning

Information about connector objects used during reconciliation and provisioning are discussed in the following topics:

- Supported Functions for Target Resource Reconciliation
- Supported Functions for Provisioning
- User Attributes for Target Resource Reconciliation and Provisioning
- GROUP Attributes for Target Resource Reconciliation and Provisioning
- Security Attributes for Provisioning
- DATASET Profile Attributes for Provisioning
- Resource Profile Attributes for Provisioning
- Reconciliation Rule
- Reconciliation Action Rules
- Viewing the Reconciliation Action Rules for IBM RACF Advanced Connector

1.6.1 Supported Functions for Target Resource Reconciliation

The connector supports reconciliation of user data from the following events:

- Create user
- Modify user
- Revoke user
- Resume user

- Delete user
- Add user to group
- Delete user from group

1.6.2 Supported Functions for Provisioning

These are the provisioning functions that the connector supports.

Table 1-2 Supported Functions for Provisioning

Function	Description	Mainframe Command
Create users	Adds new users on IBM RACF	ADDUSER
Create groups	Adds new group on IBM RACF	ADDGRP
Modify users	Modifies user information on IBM RACF	ALTUSER
Change password	Changes user passwords on IBM RACF in response to password changes made on Oracle Identity Manager through user self-service	ALTUSER
Reset passwords	Resets user passwords on IBM RACF The passwords are reset by the administrator.	ALTUSER
Revoking user accounts	Sets IBM RACF user to a REVOKED state	ALTUSER
Resuming user accounts	Sets IBM RACF user to an ENABLED state	ALTUSER
Add user to group	Connects user with an IBM RACF group	CONNECT
Remove user from group	Disconnects user from an IBM RACF group	REMOVE
Permit user to dataset	Permits user to be part of the data set ACL and gives them access rights to the data set	PERMIT
Remove user from dataset	Removes user from the data set ACL	PERMIT
Permit user to access general resource	Permits user to be part of the resource ACL and gives them access rights to the resource	PERMIT
Remove user from general resource	Removes user from the resource ACL	PERMIT
Grant security attribute to user	Provides non-value security attribute privileges to user	ALTUSER
Grant user to TSO segment	Provides TSO access and information to user	ALTUSER
Grant user to OMVS segment	Provides OMVS information to users	ALTUSER
Delete user	Deletes user from IBM RACF	DELUSER

1.6.3 User Attributes for Target Resource Reconciliation and Provisioning

Table 1-3 lists attribute mappings between IBM RACF and Oracle Identity Manager for target resource reconciliation and provisioning. The OnBoardRacfUser and ModifyUser adapters are used for the Create User and Modify User provisioning operations, respectively.

Table 1-3 User Attributes for Target Resource Reconciliation and Provisioning

Process Form Field	IBM RACF Attribute	Description
cn	NAME	Full name You can specify the format in which Full Name values are stored on the target system.
cicsOpclass	CICS_OPCLASS	Operator class
cicsOpident	CICS_OPIDENT	Operator ID
cicsOpprty	CICS_OPPRTY	Operator priority
cicsRslkey	CICS_RSLKEY	Resource key 0–99
cicsTimeout	CICS_TIMEOUT	Timeout value
cicsTslkey	CICS_TSLKEY	Type key 1–99
cicsXrfsoff	CICS_XRFSOFF	Transaction off (Force|NoForce)
dfltGrp	DEFAULT-GROUP	Default group for the user
instdata	DATA	Installation-defined data for the user
netviewConsname	NETVIEW_CONSNAME	Console name
netviewCtl	NETVIEW_CTL	Control
netviewDomains	NETVIEW_DOMAINS	Domain name
netviewIc	NETVIEW_IC	Command|Command List
netviewMsgrecvr	NETVIEW_MSGRECVR	Message receiver
netviewNgmfadmn	NETVIEW_NGMFADMN	Administration (Y|N)
netviewNgmfvspn	NETVIEW_NGMFVSPN	View span
netviewOpclass	NETVIEW_OPCLASS	Operator class
omvsAssizemax	OMVS_ASSIZEMAX	Address space size
omvsAutouid	OMVS_AUTOUID	Generate auto user identifier
omvsCputimemax	OMVS_CPUTIMEMAX	CPU time
omvsFileprocmax	OMVS_FILEPROCMAX	Files per process
omvsHome	HOME	Homelocation
omvsMemlimit	OMVS_MEMLIMIT	Non-shared memory size
omvsMmapareamax	OMVS_MMAPAREAMAX	Memory map size
omvsProcusermax	OMVS_PROCUSERMAX	Processes per UID
omvsProgram	PROGRAM	Program
omvsShared	OMVS_SHARED	Shared user identifier
omvsShmemmax	OMVS_SHMEMMAX	Shared memory size
omvsThreadsmax	OMVS_THREADSMAX	Threads per process
omvsUid	UID	UID
owner	OWNER	Owner of the user profile
resumeDate	RESUME DATE	Future date from which the user will be allowed access to the system
revokeDate	REVOKE DATE	Future date from which the user's access to the system will be revoked
revoke	REVOKE|RESUME	Status of the user
tsoAcctNum	ACCTNUM	Default TSO account number on the TSO/E logon panel

Table 1-3   (Cont.) User Attributes for Target Resource Reconciliation and Provisioning

Process Form Field	IBM RACF Attribute	Description
tsoCommand	COMMAND	Command to be run during TSO/E logon
tsoDest	DEST	Default SYSOUT destination
tsoHoldclass	HOLDCLASS	Default hold class
tsoJobclass	JOBCLASS	Default job class
tsoMaxSize	MAXSIZE	Maximum region size the user can request at logon
tsoMsgclass	MSGCLASS	Default message class
tsoProc	PROC	Default logon procedure on the TSO/E logon panel
tsoSize	SIZE	Minimum region size if not requested at logon
tsoSysoutclass	SYSOUTCLASS	Default SYSOUT class
tsoUnit	UNIT	Default UNIT name for allocations
tsoUserdata	USERDATA	TSO-defined data for the user
uid	User	Login ID
userPassword	PASSWORD	Password used to log in
waaccnt	WAACCNT	Account number for APPC or IBM z/OS processing
waaddr1	WAADDR1	Address line 1 for SYSOUT delivery
waaddr2	WAADDR2	Address line 2 for SYSOUT delivery
waaddr3	WAADDR3	Address line 3 for SYSOUT delivery
waaddr4	WAADDR4	Address line 4 for SYSOUT delivery
wabldg	WABLDG	Building for SYSOUT delivery
wadept	WADEPT	Department for SYSOUT delivery
waname	WANAME	User name for SYSOUT delivery
waroom	WAROOM	Room for SYSOUT delivery

1.6.4 GROUP Attributes for Target Resource Reconciliation and Provisioning

The connector supports reconciliation and provisioning of the GROUP multivalued attribute. For any particular user, a child form is used to hold values of the GROUP attributes listed in the table. The AddUserToGroup and RemoveUserFromGroup adapters are used for GROUP provisioning operations.

Table 1-4 lists GROUP attribute mappings between IBM RACF and Oracle Identity Manager.

Table 1-4   GROUP Attribute Mappings for IBM RACF Connector

Child Form Field	IBM RACF Attribute	Description
MEMBER_OF	GROUP	UID Of the group being assigned to User

1.6.5 Security Attributes for Provisioning

The connector supports provisioning of the SECURITY ATTRIBUTE multivalued attribute. For any particular user, a child form is used to hold values of the SECURITY ATTRIBUTE attributes listed in the table.

The following list shows the bit flag security attributes that are supported for provisioning operations between Oracle Identity Manager and IBM RACF:

- ADSP
- AUDITOR
- CICS
- DCE
- DFP
- EXPIRED
- GRPACC
- NETVIEW
- OIDCARD
- OMVS
- OPERATIONS
- OPERPARM
- OVM
- PROTECTED
- PROXY
- RESTRICTED
- SPECIAL
- TSO
- UAUDIT

Table 1-5 Security Attribute for Target Resource Reconciliation and Provisioning

Child Form Field	IBM RACF Attribute	Description
ATTRIBUTE	Security Attribute	Attribute access authority for user

1.6.6 DATASET Profile Attributes for Provisioning

The connector supports provisioning of the DATASET multivalued attribute. For any particular user, a child form is used to hold values of the DATASET attributes listed in the table.

Table 1-6 lists DATASET attribute mappings between IBM RACF and Oracle Identity Manager.

Table 1-6   DATASET Attribute Mappings for IBM RACF Advanced Connector

Child Form Field	IBM RACF Attribute	Description
Dataset Name	PROFILE NAME	Prodile ID
Dataset Access	ACCESS	User's access level to the dataset
Dataset Generic	GENERIC	Treat the dataset as a generic name

1.6.7 Resource Profile Attributes for Provisioning

(The connector supports reconciliation and provisioning of the RESOURCE PROFILE multivalued attribute. For any particular user, a child form is used to hold values of the RESOURCE PROFILE attributes listed in the table.

Table 1-7   Resource Profile Attributes for Target Resource Provisioning

Child Form Field	IBM RACF Attribute	Description
RESOURCE PROFILE ID	RESOURCE PROFILE NAME& CLASS NAME	Profile ID and class name combinations
RESOURCE ACCESS	RESOURCE ACCESS	User's access level to resource profile

1.6.8 Reconciliation Rule

During target resource reconciliation, Oracle Identity Manager tries to match each user fetched from IBM RACF with existing IBM RACF resources provisioned to OIM Users. This is known as process matching. A reconciliation rule is applied for process matching. If a process match is found, then changes made to the user on the target system are copied to the resource on Oracle Identity Manager. If no match is found, then Oracle Identity Manager tries to match the user against existing OIM Users. This is known as entity matching. The reconciliation rule is applied during this process. If an entity match is found, then an IBM RACF resource is provisioned to the OIM User. Data for the newly provisioned resource is copied from the user profile.

The following is the reconciliation rule for target resource reconciliation:

Rule name: IdfReconUserRule

Rule element: User Login Equals uid

In this rule element:

- User Login is the User ID field on the process form and the OIM User form.
- uid is the USER attribute on IBM RACF.

After you deploy the connector, you can view this reconciliation rule by performing the following steps:

1. On the Design Console, expand **Development Tools** and then double-click **Reconciliation Rules**.
2. Search for and open the **IdfReconUserRule** rule.

1.6.9 Reconciliation Action Rules

Reconciliation action rules specify actions that must be taken depending on whether or not matching IBM RACF resources or OIM Users are found when the reconciliation rule is applied. Table 1-8 lists the reconciliation action rules.

Table 1-8 Reconciliation Action Rules for IBM RACF Advanced Connector

Rule Condition	Action
No Matches Found	None
One Entity Match Found	Establish Link
One Process Match Found	Establish Link

1.6.10 Viewing the Reconciliation Action Rules for IBM RACF Advanced Connector

After you deploy the connector, you can view the reconciliation action rules for target resource reconciliation by performing the following steps:

1. On the Design Console, expand **Resource Management** and then double-click **Resource Objects**.
2. Search for and open the **OIMRacfResourceObject** resource object.
3. Click the **Object Reconciliation** tab, and then click the **Reconciliation Action Rules** tab. The Reconciliation Action Rules tab displays the action rules defined for this connector. Reconciliation Action Rules shows the reconciliation action rule for target resource reconciliation.

2

Installing and Configuring the LDAP Gateway

The LDAP Gateway acts as the intermediary between Oracle Identity Manager and the connector components on the mainframe. You can install the LDAP Gateway either on a Microsoft Windows or RHEL Linux platform.

2.1 Hardware Requirements for Installing the LDAP Gateway

These are the recommended hardware requirements that are designed to give you optimal system performance from the LDAP gateway.

Table 2-1 Hardware Requirements for Installing the LDAP Gateway

Requirement Type	Processor	RAM	Hard Disk	Network Interface
Minimum hardware requirement	2 GHz single-core processor	4 GB RAM	10GB hard disk drive	1
Recommended hardware requirement	2 GHz multicore processor	16 GB RAM	50GB hard disk drive	1

2.2 Installing the LDAP Gateway

You can install the LDAP Gateway on Windows and Linux platforms.

See Hardware Requirements for Installing the LDAP Gateway and the "LDAP Gateway" row of Certified Components to ensure that the computer on which you want to install the LDAP Gateway meets the recommended specifications.

To install the LDAP Gateway:

1. Download and save the connector installation package (for example, IBM_RACF_Adv_9.1.0.0.zip) to any directory on the computer that will host the LDAP

Gateway. You can download the connector installation package from the OTN website at http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html.

2. Extract the contents of the connector installation package to any directory on the computer. This creates a directory named *CONNECTOR_NAME-RELEASE_NUMBER*.
3. Extract the contents of the etc/LDAP Gateway/IDF_LDAP_GATEWAY_vX.X.X.zip file from the connector installation package to a temporary directory on the computer hosting the LDAP gateway.
4. Depending on the operating system computer on which you want to install the LDAP Gateway, run one of the following files:
    - Microsoft Windows: IDFLDAPGateway-X-windows-oracle-vX.X.X.exe
    - Linux: IDFLDAPGateway-X-linux-x64-oracle-vX.X.X.run
5. On the Setup - LDAP Gateway screen, click **Next** to proceed with installation.
6. On the License Agreement screen, select **I accept the agreement** if you agree with the terms of the agreement, and then click **Next**.
7. On the Installation Location screen, specify the location where the LDAP Gateway must be installed.
    - For Linux:
      When you install the gateway as a normal user, the default location is inside the Home folder `(home/ubuntu/IDFLDAPGateway-X)`.

      When you install the gateway as a sudo or root user, the default location is `/opt/IDFLDAPGateway-X`.
    - For Microsoft Windows, the default location is Program files `(…\ProgramFiles (x86)\IDFLDAPGateway-X)`
8. Click **Next** to proceed.
9. On the License File screen, browse to the location containing the license.lic file, select it and then click **Next**. For the license.lic file please contact the Oracle team.

   The Ready to Install window is displayed.
10. Click **Next** to proceed.

    The Installing screen with a progress indicator bar for the installation is displayed.
11. On the Completing the LDAP Gateway Setup Wizard screen, select **View Readme File** if you want to read the enhancements made to the gateway. Click **Finish** to complete the installation process.

2.3 Upgrading the LDAP Gateway

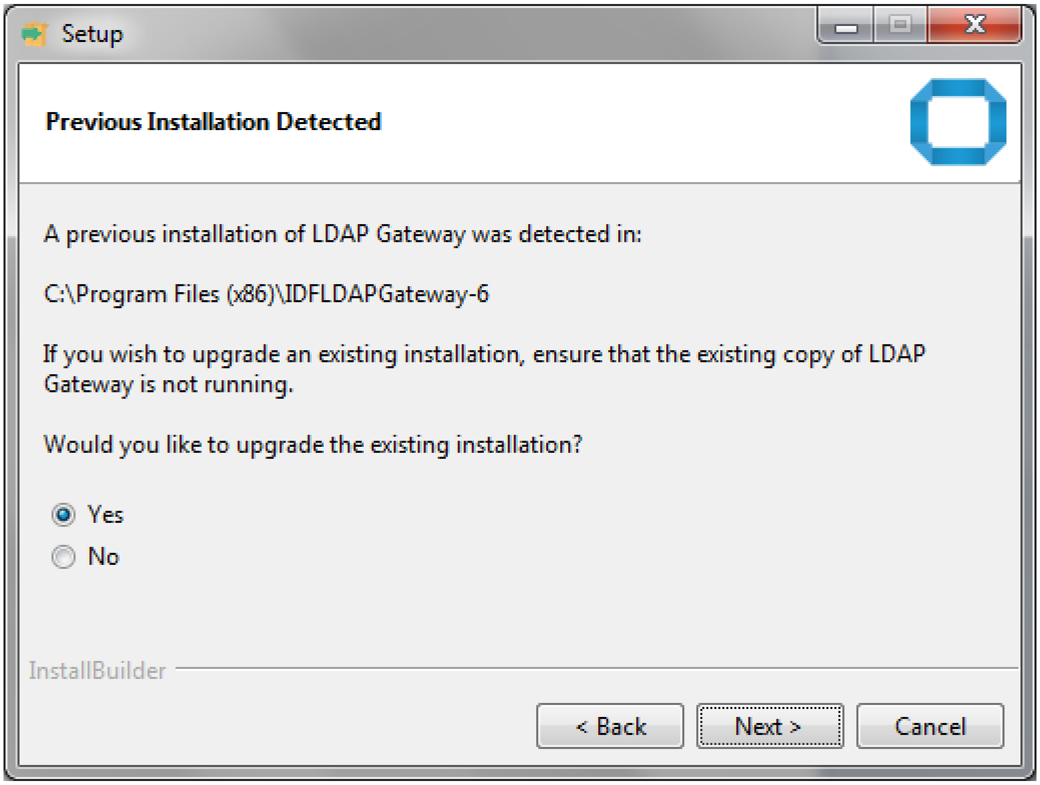
If you already have an earlier version of the LDAP Gateway (for example, version 5.*x*), then you can upgrade it to the latest version 6.x by running the LDAP gateway installer.

> **Note:**
>
> Before you begin the upgrade procedure:
>
> - On the computer hosting the gateway, stop the running instance of the gateway. If you are using a Microsoft Windows Service to run the gateway, then uninstall the Windows service.
> - In the target system environment, shut down any agents (for example, Pioneer or Voyager) that may be running.
> - Disable any cron jobs.

To upgrade the LDAP Gateway, do the following:

1. Download and save the connector installation package to any directory on the computer that will host the LDAP Gateway. You can download the connector installation package from the OTN website at http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html.
2. Extract the contents of the IDF_LDAP_GATEWAY_v6.4.0-rc.2.zip file from the connector installation package to a temporary directory on the computer hosting the LDAP gateway.
3. Depending on the operating system of the computer on which the LDAP gateway is installed, run one of the following files:
   - For Linux: IDFLDAPGateway-6-linux-x64-v6.4.0-rc.2.run
   - For Microsoft Windows: IDFLDAPGateway-6-windows-v6.4.0-rc.2.exe
4. On the Setup - LDAP Gateway screen, click **Next** to proceed with upgrade.
5. On the License Agreement screen, select **I accept the agreement** if you agree with the terms of the agreement, and then click **Next**.

   The installer detects the earlier installation of the gateway as shown in the following image:



Previous Installation Detected

A previous installation of LDAP Gateway was detected in:

C:\Program Files (x86)\IDFLDAPGateway-6

If you wish to upgrade an existing installation, ensure that the existing copy of LDAP Gateway is not running.

Would you like to upgrade the existing installation?

● Yes
○ No

6. On the Previous Installation Detected screen, when you are prompted whether you want to upgrade the existing installation, select one of the following options:

   - select **Yes** if you want to upgrade, and click **Next** to proceed. Then, on the Ready to Install screen, click **Next** to proceed with the upgrade.
   - Select **No** if you want to perform a fresh installation, and then click **Next** to proceed.

   > **Note:**
   >
   > To upgrade from version 5.x to 6.x, you need to provide the location of the existing installation folder location and the path of the valid license file. If the installation folder location is same, then the installer detects and creates a backup of the entire folder of the previous version with a suffix pre- and a timestamp. This can be verified at the installation location. The backup of the entire folder happens only once when you are upgrading from version 5.x to version 6.x. For example, if you already have a Gateway version 5.3 installed on your system, and you want to install Gateway version 6, then a backup folder for the files of 5.3 is created at the installation location.

   The Ready to Install window is displayed.

7. If you selected **No** on the Previous Installation Detected screen, then on the Installation Directory screen, specify the location where the gateway must be installed.

   a. For Linux:

      When you install the gateway as a normal user, the default location is inside the Home folder `(home/ubuntu/IDFLDAPGateway-6)`.

When you install the gateway as a sudo or root user, the default location is `/opt/IDFLDAPGateway-6`.

b. For Microsoft Windows, the default location is Program files `(…\ProgramFiles (x86)\IDFLDAPGateway-6)`

> **Note:**
>
> If the installation directory points to a location containing an existing gateway, that gateway is automatically upgraded during the installation process.

8. Click **Next**. In the Upgrade Previous Install dialog box, click **Yes** to confirm that you want to upgrade your existing installation of the gateway.
9. In the Ready to Install screen, click **Next** to proceed with the upgrade.

   The Installing screen with a progress indicator bar for the installation is displayed.
10. On the Completing the LDAP Gateway Setup Wizard screen, select **View Readme File** if you want to read the enhancements made to the gateway. Click **Finish** to complete the upgrade process.

2.4 Configuring the LDAP Gateway

Configure the LDAP gateway to connector to the target system and access the data.

The following topics describe the procedure to configure the LDAP Gateway:

> **Note:**
>
> The following procedures are for a fresh installation only. If you already have a running setup or if you want to upgrade, then you do not have to perform these procedures.

- Setting Connection Properties
- Creating the Connector Configuration
- Configuring the LDAP Gateway for Multiple Installations of the Target System
- Overriding the Default System Configuration

2.4.1 Setting Connection Properties

The *`LDAP_INSTALL_DIR`*`/conf` directory contains the racf.properties.example file that contains sample entries and is used as the basis for configuring the gateway.

The racf.properties.example file is only a sample file. Therefore, create a separate properties file (for example, racf.properties) in the *`LDAP_INSTALL_DIR`*`/conf` location by creating a copy of the racf.properties.example file. Use this properties file to specify connection information that the gateway uses to connect to your target system. To do so:

1. In the *`LDAP_INSTALL_DIR`*`/conf` directory, create a copy of the *`LDAP_INSTALL_DIR`*`/conf/racf.properties.example` file and rename it to for example, racf.properties.

> **Note:**
>
> If you are configuring the gateway for multiple instances of the target system, then you must create a copy of the *LDAP_INSTALL_DIR*/conf/racf.properties.example file and rename it for each target system instance. Ensure that the names of the renamed files are not the same.

2. In a text editor, open the racf.properties file for editing and set values for properties such as host, port, user credentials and so on to point to your environment.

   The following table describes these properties.

Table 2-2 Properties in the racf.properties File

Property	Description
agentPort	Enter the port number on the LDAP Gateway host computer that you are going to reserve for messages sent from the mainframe by the Reconciliation Agent, Voyager. The LDAP Gateway will receive real-time reconciliation messages using this port. This value should match the value of the PORT parameter in the Voyager agent control file.
agentMetaRecon	This property specifies whether Voyager must reconcile user data from the target system into the legacy meta store that is located in the LDAP gateway. The reconciled data is stored in the `OU=racf` subtree of the `OU=People` tree that is located in the system backend (`DC=System,DC=Backend`). Enter `true` to reconcile user data into the legacy meta store. Otherwise, enter `false`. The default value of this property is `true`. **Note:** If you upgraded the LDAP gateway from release 5.x to 6.x, then this property is not available in the racf.properties file by default. If you want to use this property, then you must add it to the racf.properties file manually.

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
agentCachingRecon	This property specifies whether Voyager must reconcile user data from the target system into the caching store that is located in the LDAP gateway. The reconciled data is stored in the `OU=people` subtree of the `OU=racf1` tree that is located in the system backend (`DC=System,DC=Backend`). Enter `true` to reconcile user data into the caching store. Otherwise, enter `false`. The default value of this property is `true`. **Note:** • If you upgraded the LDAP gateway from release 5.x to 6.x, then this property is not available in the racf.properties file by default. If you want to use this property, then you must add it to the racf.properties file manually. • If you set the value of this property to `true`, then ensure that the caching layer is enabled. If the caching layer is not enabled, then data is reconciled into the legacy meta store instead of the caching store. See Understanding the Caching Layer for information on how to enable the caching layer.
allowUpdateForDatasetAttributesOtherThanUniqueMember	Flag to determine if dataset attributes other than uniquemember can be modified on ou=Datasets. `(true|false)` default is `true`
attributesForAltGroupToBeSentAfterAddGroup	List of comma separated attributes to be sent in the ALTGROUP command post an ADD. These attributes can be sent in the LDAP Add request and will be sent in the ALTGROUP command to Racf, for example, NOTSO,NOCICS,NOOMVS.
attributesForAltUserToBeSentAfterAddUser	List of comma separated attributes to be sent in the ALTUSER command post an ADD. These attributes can be sent in the LDAP Add request and will be sent in the ALTUSER command to Racf, for example, NOTSO,NOCICS,NOOMVS.

configDNames	This property holds the display names of RACF fields that are defined in the CSDATA segment and used during user reconciliation operations. If entering more than one value, separate each value with a vertical bar (|) character. Each display name should have a corresponding configAttrs entry.
	For example, if you define a field with a display name of $PST15 and VEND ID, then you would enter:
	`# CUSTOM CSDATA RACF ATTRIBUTE DISPLAY NAME`
	`_configDNames_=$PST15 =|VEND ID =|`

Table 2-2   (Cont.) Properties in the racf.properties File

Property	Description

configAttrs	This property holds the field names of any custom target system fields that are defined in the CSDATA user segment and used during user provisioning operations. If entering more than one value, separate each value with a vertical bar (|) character. Each field name should have a corresponding configDNames entry. This step is mentioned in the following sections:
	• Adding Custom Fields for Target Resource Reconciliation
	• Adding Custom Fields for Provisioning for IBM RACF Advanced Connector
	For example, if you define fields with a display name of $PST15 and VEND ID, then you would enter:
	`# CUSTOM CSDATA RACF ATTRIBUTE FIELD NAME`
	`_configAttrs_=$PST15|VEND ID`

defaultDelete	Enter one of the following as the value of this property:
	• Set `revoke` as the value if you want the user to be disabled on the target system as the outcome of a Delete User provisioning operation.
	• Set `delete` as the value if you want the user to be deleted from the target system as the outcome of a Delete User provisioning operation.
	For example:
	`# DEFAULT ACTION WHEN DELETE FUNCTION USED`
	`_defaultDelete_=delete`

executeAltGroupAfterAddGroup	Flag to determine if ALTGROUP is to be executed after ADDGROUP. The attributes to be updated in the ALTGROUP need to be configured in `attributesForAltGroupToBeSentAfterAddGroup` property. `(true|false)` default is `true`.
host	Enter the host name or IP address of the computer that must connect to Pioneer. For example, `_host_=localhost`.
port	Enter the number of the port on the Mainframe that you are going to reserve for Pioneer. The LDAP Gateway will send provisioning messages to this port. This value should match the PORT parameter specified in the Pioneer provisioning agent STC. For example, `_port_=5790`.

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
stcID	This property allows the real-time agent to ignore events that have been submitted to the target system by the Pioneer STC (such as by request from Oracle Identity manager). Enter the name given to the Pioneer STARTED TASK.
auditOn	This property is used to store audit data from IBM RACF. Default setting is `false`.
batchMetaRecon	Batch Reconciliation will reconcile to the Legacy Meta Store (true|false - default is true)
batchCachingRecon	Batch Reconciliation will reconcile to the Caching Store (true|false - default is true)
domainOu	This property stores users in the specified subtree under the ou=People tree of the internal LDAP store. This entry needs to be unique and specific for each system if multiple systems are used within one LDAP Gateway. Default setting is `domainOu=racf`
executeAltUserAfterAddUser	Flag to determine if ALTUSER is to be executed after ADDUSER. The attributes to be updated in the ALTUSER need to be configured in `attributesForAltUserToBeSentAfterAddUser` property. `(true|false)` default is `true`.
internalEnt	This property allows the real-time agent to store user data in the LDAP Gateway internal store. Values: `[true|false]`
internalGrpEnt	This property is used to allow the real-time agent to store groups in the LDAP internal store. Values: `[true|false]`
internalCREnt	This property is used to allow the real-time agent to store connect and remove commands in the LDAP internal store. Values: `[true|false]`
isStreamingUsers	This property is used by the RACF Reconcile Users to Internal LDAP scheduled task. If you set the value of this property to `true`, then the LDAP gateway will process the USER EXTRACT data from the mainframe. If you set the value of this property to `false`, then the LDAP gateway will not process any USER EXTRACT data. Default value: `true`

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
isStreamingGroups	This property is used by the RACF Reconcile Users to Internal LDAP scheduled task. If you set the value of this property to `true`, the LDAP gateway will process the GROUP EXTRACT data from the mainframe. If you set the value of this property to `false`, the LDAP gateway will not process any GROUP EXTRACT data. Default value: `true`
configExtractAttrs	Use this property to list any custom CSDATA fields for RACF. Use this when using 'useExtractUser=true' property above. **Note:** The value in this property must match the RACF CSDATA segment. Sample value: `EMPSER :|NETAN :|`
allowDeleteDS	This property is used for default action when a delete request occurs that will delete dataset profiles for user being deleted. If the property is set to `true`, deleting a user will delete both the user and the user's datasets.
passUnknownAttrs	Whether unknown attributes will be included in commands that are sent to the target system. `(true|false)` default is `false`. For existing customers before v8.1, this property will be migrated as true to keep the existing behaviour.
refreshResourceAfterPermit	Flag to determine if refresh is required for resource after create or update. `(true|false)` default is `true`.

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
retrieveAllResourceClasses	Get all resources classes and allow to provisionresources of all classes. `(true|false)` default is `false` • If `false` the resource classes mentioned in the supportedResourceClasses property will be available for search and provision. • If `true`, all general resources from RACF system will be retrieved and new resources can be provisioned as well. • Make sure to execute a batch job on mainframe (sample supplied in <hlq>.JCLLIB(RACFRC)) as a pre-requisite to achieve this functionality, without which all the resource classes won't be retrieved. **Note:** Periodic batch reconciliation for Resources is required for this feature. Batch reconciliation of resource is done by reading the database-unload utility on RACF. Batch reconciliation for resources will overwrite few attributes which are extracted during the read operation. The attributes provided by batch reconciliation are different than those extracted from the read operation.
resourceReadFromStaticFile	This flag determines whether or not to read the resources from the static file. `(true|false)` default is `false`. **Note:** If this property is set as `true` then RACFRC job needs to be set as: `resourceReadFromStaticFile` .
secretKeyValue	This property contains the custom encryption key. This key should match the secretKey value used by the mainframe agents. See Customizing AES Encryption Key for more information about using this property.

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
treatPasswordAsPhrase	Determines whether values specified for the `userPassword` attribute are passed to RACF as a passphrase instead of password. `(true|false)` default is `false` This can be used if phrase attribute cannot be mapped.
truncatePassword	Flag to determine if the password sent needs to be truncated to the max length specified by _pwdMaxLength_. `(true|false)` default is `false` This is introduced for backward compatibility with v5 gateway. This should be set to true for customers who upgraded from v5 and need to truncate the password the length specified by _pwdMaxLength_. **Note:** This property is deprecated and setting this property to true is not recommended
useUnivGrp	Use this property to specify whether to use universal groups instead of normal groups on the target system. Universal groups can have an unlimited number of AUTH(USE) userIDs connected to it. Values: `(true|false)`
resumeOnReset	This property is used when resetting a user's password. If you set the value of this property to true, the user will be enabled during a reset password operation. If you set the value of this property to false, the user will not be enabled during a reset password operation. Default value: `true`
trimOmvsUid	This property is used with the omvsUid attribute. If you set the value of this property to true, the LDAP gateway will trim leading zeros, "0", from the omvsUid value. If you set the value of this property to false, the LDAP gateway will not trim any leading zeroes from the omvsUid value. Default value: `true`
trimNum	This property is used with the trimOmvsUid property and specifies the number of leading zeroes to trim from a user's omvsUid attribute. Default value: `2`

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
newOmvsUidAttr	This property specifies the new name to use for the omvsUid property. Default value: `OmvsUidEmplNumber`
usePwdComplexLength	This property is used to control the length of passwords. If you set the value of this property to true, the LDAP gateway will use the properties file password length settings. If you set the value of this property to false, the LDAP gateway will use the standard password length. Default value: `true`
idMinLength	This property specifies the minimum ACID length in characters. Default value: `1`
idMaxLength	This property specifies the maximum ACID length in characters. Default value: `8`
pwdMinLength	This property specifies the minimum password length for an ACID. Default value: `1`
pwdMaxLength	This property specifies the maximum password length for an ACID. Default value: `8`
phraseMinLength	Enter the minimum length for the passphrase as per IBM RACF documentation. Sample value: `9`
phraseMaxLength	Enter the maximum length for the passphrase as per IBM RACF documentation. Sample value: `100`
dateFormat	Enter the format for the date coming in from Mainframe in Read Response. For example, `MM/dd/yy`.
supportedResourceClasses	Enter the list of supported RACLIST resource classes on which the connector must perform CRUD operations. If you have to enter more than one resource class, then separate each value with a vertical bar (|) character. Sample value: `FACILITY|CDT|OPERCMDS`
type isencrypted timeout authretries requestorId CPF CPF-WAIT	These properties are no longer used in Oracle installations. Do not modify their values.

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
sendAltGrpWithMembershipUpdate	This parameter is used to determine if other group attributes can be modified along with the membership update. If you set the value of this property to `true`, AltGroup will be executed. If the value of the property is set to `false`, AltGroup will not be executed. Default value: `true`
updateDefaultGroupFailsIfUserIsNotMember	Flag to determine whether defaultGroup can be updated based on whether it is a part of `memberOf`. Values are `true|false` Default value: `false`
batchReconUpdatesUsersAndGroupsWithDatasetAccess	Flag to determine if batch reconciliation for Datasets should update the user and group profiles with dataset access. Values are `true|false` Default value: `false` **Note:** this is a performance intensive operation. Batch recon will take more time if there are a lot of memberships for datasets.
batchReconUpdatesUsersAndGroupsWithResourceAccess	Flag to determine if batch reconciliation for Resources should update the user and group profiles with resource access. Values are `true|false` Default value: `false` **Note:** this is a performance intensive operation. Batch recon will take more time if there are a lot of memberships for resources.
batchReconUpdateEntriesOnlyIfModified	Batch Reconciliation will reconcile updates only if the target attributes are modified. Values are `true|false` Default value: `false`

Table 2-2 (Cont.) Properties in the racf.properties File

Property	Description
batchReconForDatasetAndResourceUsingExtract	Batch Reconciliation for datasets and resource based on extract functionality. Values are `true|false` Default value: `true` **Note:** For z/OS 2.3 : If `false` then dataset and resource recon will use the dbUnload functionality. For z/OS 2.4 : If `true` then dataset and resource recon will use the extract functionality.

3. If you want to include custom segment as a part of the TSS LIST command set, then set a value for the `_configDatasets_` property.

   Use the following components to set a value for the `_configDatasets_` property:

   - Use fn to represent the first name.
   - Use sp to represent the space character.
   - Use ln to represent the last name.
   - Use a comma (,) to represent the comma.
   - Use a period (.) to represent the period.
   - Use the vertical bar (|) as the separator for the other components.

4. Save and close the file.

2.4.2 Creating the Connector Configuration

To allow the gateway to work with the target system, you must create and configure the customer-configuration.properties file for the type of connector and its related parameters for the operations.

> **Note:**
>
> In this guide, *LDAP_INSTALL_DIR* is the standard term used to refer to the directory in which the gateway has been installed. For example, for a Microsoft Windows host machine, the default installation directory for the gateway is . .\Program Files (x86)\IDFLDAPGateway-6\.

1. Create an empty customer-configuration.properties text file in the `LDAP_INSTALL_DIR/conf` directory.

> **Note:**
>
> If you have upgraded the gateway, then skip this step as the customer-configuration.properties file already exists and contains all the connector configurations present in the beans.xml file.

2. Navigate to the *`LDAP_INSTALL_DIR`*`/conf` directory and then locate and open the customer-configuration.properties.example file.

   The customer-configuration.properties.example file contains sample definitions (configuration properties) in various sections for each connector that the LDAP Gateway can be used with.

3. Search for and copy the following snippet from the customer-configuration.properties.example file, and paste it into the customer-configuration.properties file located in the *`LDAP_INSTALL_DIR`*`/conf` directory.

   ```
   cnctr.racf.class=com.identityforge.idfserver.backend.racf.RacfModule
   cnctr.racf.racf1.schema=schemas
   cnctr.racf.racf1.suffix=dc=racf,dc=com
   cnctr.racf.racf1.adminUserDN=cn=idfRacfAdmin,dc=racf,dc=com
   cnctr.racf.racf1.adminUserPassword=idfRacfPwd
   cnctr.racf.racf1.altAdminUserDN=cn=oimRacfAdmin,dc=racf,dc=com
   cnctr.racf.racf1.altAdminUserPassword=oimRacfPwd
   cnctr.racf.racf1.configLocation=../conf/racf.properties
   cnctr.racf.racf1.allowAnonymous=false
   cnctr.racf.racf1.metaBackend=ldapds
   cnctr.racf.racf1.agent=true
   cnctr.racf.racf1.customSchemaLocation=
   cnctr.racf.racf1.people.multiCallAttributes=userpassword,attributes,uid,use
   rpassword|userpassword,userpassword|passwordexpire|passwordexpiredays
   # Simple equality filters using the following attributes will be passed
   through to the target
   cnctr.racf.racf1.cachingAllowedTargetFilterAttributes=uid,objectClass,allda
   ta
   ```

4. In the customer-configuration.properties file, rename the connector qualifier for the newly pasted entries to match the name of the connection properties file that you created in Setting Connection Properties. Suppose you created a properties file named racfadv.properties, then rename all instances of racf in the newly pasted configuration entries to racfadv. For example, in the `cnctr.`**`racf`**`.racf1.suffix=dc=racf,dc=com` property, rename racf to `racfadv`. So the entry will now be `cnctr.`**`racfadv`**`.racf1.suffix=dc=racf,dc=com`

5. Similarly, rename the instance ID for all the configuration properties. For example, in the `cnctr.racf.racf1.schema=schemas` property, rename racf1 to `racf2`.

6. Edit the value of the `cnctr.racf.racf1.configLocation=` property to point to the connection properties file that you created in Setting Connection Properties. For example, if you created a file named racfadv.properties, then replace `cnctr.racf.racf1.configLocation= ../conf/racf.properties` with `cnctr.racf.racf1.configLocation= ../conf/`**`racfadv.properties`**

7. Change the default system administrator credentials that the gateways uses to connect to the target system as follows:

a. Locate the following properties:

   ```
   cnctr.racf.racf1.adminUserDN=cn=idfRacfAdmin,dc=racf,dc=com
   cnctr.racf.racf1.adminUserPassword=idfRacfPwd
   ```

b. Set new values for the adminUserDN and adminUserPassword properties and note them down. You must enter the same values for the idfPrincipalDn and idfPrincipalPwd parameters of the IT resource.

   > **Note:**
   >
   > - By default, all sensitive data is automatically encrypted when you start the gateway.
   > - For the adminUserDN property:
   >   - It is mandatory to that you use `cn` as the RDN identifier.
   >   - If you put spaces after the commas in the DN, then you must match that when using that ID to connect to the gateway. For example, if the required format is `cn=adminId,dc=racf,dc=com`, then `dc=racf,dc=com` must match the suffix property.

8. Save and close the file.
9. Restart the gateway for the changes to take effect.

2.4.3 Configuring the LDAP Gateway for Multiple Installations of the Target System

You can instantiate the same type of connector multiple times to represent multiple different endpoints of the same target system. This is in addition to the gateway supporting the ability to run connectors for various target systems within a single gateway instance.

If you have already configured a single instance of the connector for one target system installation and want to configure an additional instance, then:

1. For each target system installation in your environment, create a properties file in the `*LDAP_INSTALL_DIR*/conf` directory by creating a copy of the `*LDAP_INSTALL_DIR*/conf/racf.properties` file. Then, edit the newly created properties file to specify all connection properties.
2. Open the customer-configuration.properties.example file located in the `*LDAP_INSTALL_DIR*/conf` directory, copy the following configuration properties specific to your connector and paste it into the `*LDAP_INSTALL_DIR*/conf/customer-configuration.properties` file, below the existing set of configuration properties.

   ```
   cnctr.racf.class=com.identityforge.idfserver.backend.racf.RacfModule
   cnctr.racf.racf1.schema=schemas
   cnctr.racf.racf1.suffix=dc=racf,dc=com
   cnctr.racf.racf1.adminUserDN=cn=idfRacfAdmin,dc=racf,dc=com
   cnctr.racf.racf1.adminUserPassword=idfRacfPwd
   cnctr.racf.racf1.altAdminUserDN=cn=oimRacfAdmin,dc=racf,dc=com
   cnctr.racf.racf1.altAdminUserPassword=oimRacfPwd
   ```

```
cnctr.racf.racf1.configLocation=../conf/racf.properties
cnctr.racf.racf1.allowAnonymous=false
cnctr.racf.racf1.metaBackend=ldapds
cnctr.racf.racf1.agent=true
cnctr.racf.racf1.customSchemaLocation=
cnctr.racf.racf1.people.multiCallAttributes=userpassword,attributes,uid,use
rpassword|userpassword,userpassword|passwordexpire|passwordexpiredays
# Simple equality filters using the following attributes will be passed
through to the target
cnctr.racf.racf1.cachingAllowedTargetFilterAttributes=uid,objectClass,allda
ta
```

Close the customer-configuration.properties.example file.

3. In the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file, rename the instance ID for all the newly pasted configuration properties. For example, in the `cnctr.racf.racf1.schema=schemas` property, replace `racf1` with `racf2`.

   > **Note:**
   >
   > Ensure that the connector name qualifier in the configuration properties matches the one that you specified while performing the procedure described in Creating the Connector Configuration. For example, in the `cnctr.racf.racf1.suffix=dc=racf,dc=com` configuration property, if you renamed racf to `racfadv`, then you must do the same for all newly added configuration properties here.

4. Modify the following properties:
   - `adminUserPassword` - change the default value for security reasons.
   - `suffix` - Enter the unique baseDN that you want to use in OIM. The default value is `dc=racf,dc=com`. You can change the default value to a baseDN of your choice.
   - `adminUserDN` - Enter the full DN of an administrative user account that is allowed to use the connector for reconciliation and provisioning operations. Note that the DN suffix must match the value that you set for `suffix` property.
   - `altAdminUserDN` - Enter the full DN of the alternative administrative user account that is allowed to use the connector for reconciliation and provisioning operations. Note that the DN suffix must match the value that you set for `suffix` property.
   - `configLocation` - Enter the location of the property file (created in Step 1) for the instance of the target system. For example, . . conf/racf10.properties. If the intent is to point these two connectors to different target systems, then the configLocation property should point to a different connector properties file (created in Step 1) for each target system instance. The new properties file can be a copy of the original properties file with changes in the necessary properties to point to the new system.
5. Save and close the `customer-configuration.properties` file and then restart the gateway for the changes to take effect.

2.4.4 Overriding the Default System Configuration

You can override the default system configuration by modifying the `LDAP_INSTALL_DIR/conf/customer-configuration.properties` file.

To change the default system properties, locate that property in the `configuration.properties` file (located in the `conf/` folder) and copy it to `customer-configuration.properties` file and provide a new value.

> **Note:**
>
> Not all properties can be modified and must be done in consultation with Support.

By default, all system configurations are stored in the `LDAP_INSTALL_DIR/conf/configuration.properties` file. If required, you can override any of these system configurations by copying relevant properties from the `LDAP_INSTALL_DIR/conf/configuration.properties` file to the `LDAP_INSTALL_DIR/conf/customer-configuration.properties` file, and then providing a new value.

> **Note:**
>
> Do not edit `LDAP_INSTALL_DIR/conf/configuration.properties` file directly as it will be overwritten when you upgrade the gateway.

There can be several reasons when you want to override the default system configuration. For example, you may want to change the default passwords for the system backend persistence store or change the listening port when the default collides with another service or when the policies of the company require using a different port.

- To change the default system backend passwords, add the following properties to the `LDAP_INSTALL_DIR/conf/customer-configuration.properties` file:

  ```
  cnctr.proxy.ldapds.adminUserPassword=<admin-password>
  cnctr.proxy.ldapds.altAdminUserPassword=<alt-admin-password>
  ```

  In the preceding lines, replace ***<admin-password>*** with the password for accessing the system backend. Similarly, replace ***<alt-admin-password>*** with the alternative password for accessing the system backend (dc=system,dc=backend)

  Not all properties can be modified and must be done in consultation with Support.

- To change the default port, add the following properties to the `LDAP_INSTALL_DIR/conf/customer-configuration.properties` file:

  ```
  system.port=6389
  system.ssl_port=7389
  ```

  In the preceding lines, replace 6386 with the desired listening port for LDAP. Similarly, replace 7389 with the desired listening port for LDAPS.

2.5 Configuring the Windows Service for the LDAP Gateway

The Windows Service for the LDAP Gateway is installed using an IdentityForge batch file (IDF-Win-Service) that is included in the installation media.

- Installing and Configuring the Windows Service for the LDAP Gateway
- Uninstalling the Windows Service for the LDAP Gateway
- Configuring Memory Pool Settings

2.5.1 Installing and Configuring the Windows Service for the LDAP Gateway

You can install the Windows Service by running the `IDF-Win-Service install` command.

To install the Windows service, switch to the *LDAP_INSTALL_DIR*/win_service directory in a command window and then run the `IDF-Win-Service install` command. If you encounter any issues with the installation, then uncomment the `CG_PATH_TO_JVM` variable in the*LDAP_INSTALL_DIR*/win_service/IDF-Win-Service.bat file and ensure that the path is accurate. The following is the code snippet from the *LDAP_INSTALL_DIR*/win_service/IDF-Win-Service.bat file that you need to uncomment:

```
rem -- 7. Set this if you want to use a different JVM than the one
configured in your registry, or if it is not configured in the windows
registry
rem set CG_PATH_TO_JVM=C:\Program Files\Java\jre7\bin\server\jvm.dll
```

If you need to modify the Windows service settings, then it is recommended to first uninstall the service, make the modifications, and then reinstall the service until it installs and runs correctly.

After installing the service, you can start, stop, or restart it anytime by using the Windows Services console. Alternatively, run the following command to start the service:

```
> net start IdentityForgeService
```

Run the following command to stop the service:

```
> net stop IdentityForgeService
```

2.5.2 Uninstalling the Windows Service for the LDAP Gateway

Uninstall the Windows service for the LDAP Gateway by running the `IDF-Win-Service remove` command.

To uninstall the Windows service, switch to the *LDAP_INSTALL_DIR*/win_service directory in a command window and then run the `IDF-Win-Service remove` command.

2.5.3 Configuring Memory Pool Settings

You can configure the memory pool size for the Windows service by setting values for the `CG_JVMMS` and `CG_JVMMX` variables in the *`LDAP_INSTALL_DIR`*`/win_service/IDF-Win-Service.bat` file.

By default, the `CG_JVMMS` and `CG_JVMMX` variables are set to 1024 MB and 2048 MB, respectively. If the LDAP gateway processes a large number of records, then you might encounter the "Out of memory" exception. In such a scenario, you can allocate higher memory for your Windows service by increasing the values of the `CG_JVMMS` and `CG_JVMMX` variables.

To do so:

1. Stop the LDAP gateway Windows service and then uninstall it.
2. In a text editor, open the *`LDAP_INSTALL_DIR`*`/win_service/IDF-Win-Service.bat` file for editing.
3. Set the JVM minimum and maximum values by modifying values for the following lines:

   ```
   rem Initial memory pool size in MB.
   set CG_JVMMS=1024

   rem Maximum memory pool size in MB.
   set CG_JVMMX=2048
   ```

   > **Note:**
   >
   > When you receive the "Out of memory" exception, start with increasing the minimum and maximum values to 2048 and 4096, respectively. If the number of records is greater than 40k, then use higher minimum and maximum values.

4. In the *`LDAP_INSTALL_DIR`*`/conf/log4j.properties` file, set the gateway debug level to `ERROR` as follows:

   ```
   rootLogger.level = ERROR
   ```

5. Install the LDAP gateway Windows service.
6. Start the LDAP gateway through the Windows service.

2.5.4 Configuring Memory Pool Settings for LDAP Gateway v8.x.x

You can configure memory pool settings for LDAP Gateway v8.x.x by following the steps outlined below:

1. Ldap Gateway: Please update the Ldap Gateway with the provided installers in IDF_LDAP_GATEWAY_<version>.zip.

> **Note:**
>
> It is strongly recommended that you create a backup of the Gateway before applying the patch.

> **Note:**
>
> JDK1.8 or later is required for the LDAP Gateway.

2. If you use Windows service to start LDAP Gateway and are upgrading from v5/v6 to v8.x.x, please perform the steps below after the upgrade is complete.
   - Stop the Windows service if it is already running
   - Open a command prompt (cmd)
   - From the command line, execute the command IDF-Win-Service.bat remove in the win_service/ directory. Close the command prompt
   - Set the following system environment variable. This should be set as a system variable so that it is preserved after the server restart.
     `CG_JVMOPTS` - Set the initial and maximum java heap size in MB with the Java memory flags e.g : `CG_JVMOPTS = --JvmMs 4096 --JvmMx 8192`

     > **Note:**
     >
     > If the environment variable `CG_JVMOPTS` is not set, then it defaults to minimum heap size of 2048 and max heap size of 4096

     > **Note:**
     >
     > If there were any customizations made to the Windows service, then those customizations need to be applied to IDF-Win-Service.bat file.

   - Open a command prompt (cmd)
   - From the command line, execute the command `IDF-Win-Service.bat install` in the win_service/ directory.
   - Start the Windows service - `IdentityForgeService`
3. If you use run.bat on Windows or run.sh on Linux to start LDAP Gateway and are upgrading from v5/v6/v8.x.x to v8.x.x, please perform the steps below after the upgrade is complete.
   - Set up environment variable `JVM_OPTS` to specify the minimum and maximum Java heap memory sizes, for example : `JVM_OPTS="-Xms2048m -Xmx4096m"`

     > **Note:**
     >
     > If this variable is not set, Java defaults for the heap size will apply.

- Execute the `run.bat` script on Windows or `run.sh` on Linux

2.6 Configuring Transformation of the LDAP Gateway Attributes

You can configure transformation of LDAP Gateway attributes in search results by adding relevant entries to the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file.

You must include the transformation rule within the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file as an inline Jtwig template. For more information about Jtwig templates, see http://jtwig.org/documentation.

For example, you can add a transformation rule to use the `givenname` and `sn` attributes to create a value for the `cn` attribute in the People OU. To do so, you must add the following line in the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file:

```
cnctr.racf.racf1.transformation.People.read.cn.template.inline={{givenname}}
{{sn}}
```

This entry will render the value of the `cn` attribute as a concatenation of the `givenname` and `sn` attributes.

To configure transformation of LDAP gateway attributes:

1. In a text editor, open the customer-configuration.properties file located in the *`LDAP_INSTALL_DIR`*`/conf` directory.
2. Add the transformation rule in the following format:

   ```
   cnctr.CONNECTOR_QUALIFIER.INSTANCE_ID.transformation.OU.read.ATTR_NAME.temp
   late.inline=JTwig_TEMPLATE
   ```

   In this format, replace:

   - *CONNECTOR_QUALIFIER* with the name of the connection properties file that you created in Setting Connection Properties.
   - *INSTANCE_ID* with the instance ID for your target.
   - *OU* with the organizational unit against which the connector must perform transformation. The supported OU values are `People`, `Groups`, `Resources`, and `Datasets`.
   - *ATTR_NAME* with the name of the LDAP Gateway attribute in which the transformed value must be stored.
   - *Jtwig_TEMPLATE* with the Jtwig template for transformation.
3. Save and close the file.

2.7 Configuring Multiple Instances of the LDAP Gateway

You can configure and run multiple instances of the LDAP Gateway on the same host by entering unique port values for each instance of the LDAP Gateway.

To do so, install and configure the LDAP Gateway for each instance that you want to run. While installing the LDAP Gateway, ensure that the installation directory is different for each instance of the gateway.

Then, update the default values for each property listed in Table 2-3 so that the value is unique for each instance of the LDAP Gateway that is installed on the host. Suppose you are using the default values in the property files for instance 1, then for instance 2, replace the default value with a unique value for the property. For example, for instance 2, change the default value `6398` of the system `system.port` property in the*LDAP_INSTALL_DIR*`/conf/customer-configuration.properties` file to a unique value such as `8389`.

Table 2-3   Property Values To Be Updated for Running Multiple Instances of the LDAP Gateway

Property Name and Location	Property Description	Default Value
The `system.port` property in the*LDAP_INSTALL_DIR*`/conf/customer-configuration.properties` file	Gateway listening port (LDAP)	`6389`
The `system.ssl_port` property value in the *LDAP_INSTALL_DIR*`/conf/customer-configuration.properties` file	Gateway listening port (LDAPS)	`7389`
The `ds-cfg-listen-port` property under `dn: cn=LDAP Connection Handler,cn=Connection Handlers,cn=config` in the *LDAP_INSTALL_DIR*`/dsroot/config/config.ldif` file	OpenDJ listening port (LDAP)	`1389`
Set the value of the `ldap.port` property in the *LDAP_INSTALL_DIR*`/conf/ldapds.properties` file to the value set for the `ds-cfg-listen-port` property (in the preceding row)	Gateway config to read OpenDJ	`1389`
The `ds-cfg-listen-port` property under `dn: cn=LDAPS Connection Handler,cn=Connection Handlers,cn=config` in the *LDAP_INSTALL_DIR*`/dsroot/config/config.ldif` file	OpenDJ listening port (LDAPS)	`1636`
The `ds-cfg-listen-port` property under `dn: cn=Administration Connector,cn=config` in the *LDAP_INSTALL_DIR*`/dsroot/config/config.ldif` file	OpenDJ Administration port	`4444`
The `ds-cfg-replication-server` property under `dn: cn=localhost,cn=domains,cn=Multimaster Synchronization,cn=Synchronization Providers,cn=config` in the *LDAP_INSTALL_DIR*`/dsroot/config/config.ldif` file	OpenDJ Replication Server	`localhost:8989`

Table 2-3 (Cont.) Property Values To Be Updated for Running Multiple Instances of the LDAP Gateway

Property Name and Location	Property Description	Default Value
The `ds-cfg-replication-port` property value under `dn: cn=replication server,cn=Multimaster Synchronization,cn=Synchronization Providers,cn=config` in the *`LDAP_INSTALL_DIR`*`/dsroot/config/config.ldif` file	OpenDJ Replication Server Port	`8989`

2.8 Encrypting Data

Learn about encryption performed by the LDAP gateway and how to configure it.

- Understanding Encryption
- Configuring Encryption

2.8.1 Understanding Encryption

The *`LDAP_INSTALL_DIR`*`/conf/encryption.properties` file allows the ability to configure what properties, associated with the connector, must the LDAP Gateway manage as encrypted values.

The *`LDAP_INSTALL_DIR`*`/conf/encryption.properties` file is a common file containing properties of various modules that need to be securely protected. Use this file to define and encrypt any property located in the following files:

- connection properties file (created in Setting Connection Properties)
- *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties`

When the LDAP gateway starts, it uses the encryption.properties file to examine the properties that it must represent in encrypted format.

For example, when the LDAP gateway starts, it reads the following entry from the `encryption.properties` file:

```
file.customer-configuration=adminUserPassword,altAdminUserPassword
```

This entry implies that there exists a properties file called customer-configuration.properties that contains sensitive properties `adminUserPassword` and `altAdminPassword`. The LDAP gateway searches for the customer-configuration.properties file, and if found, replaces any clear-text values for the `adminUserPassword` and `altAdminPassword` properties with an encrypted version.

Similarly, at start up, the LDAP gateway also reads the following entry from the `encryption.properties` file:

```
class.RacfModule=_secretKeyValue_
```

This entry implies that there exists a connector called RacfModule and its associated properties file (the one created in Setting Connection Properties) contains the sensitive property `_secretKeyValue_` . The LDAP gateway serach for this properties file and replaces the clear-text value for the `_secretKeyValue_` property with an encrypted value.

Encrypted values within property files are always represented using the `ENC(ENCRYPTED_STRING)` format. To add or replace an existing encrypted value with a new value, replace the entire encryption string if present (including the `ENC(ENCRYPTED_STRING)`) with a new clear-text value, and then restart the gateway. Once the gateway restarts, the newly added clear-text value goes through an encryption process with the result being written back out to the property file replacing the original clear-text value.

During the encryption process, the encryption framework that the gateway uses automatically detects the highest level of encryption possible by examining the version of the Java Virtual Machine running, along with any additional encryption libraries that may have been installed alongside the JVM. By default, Java 1.8 supports 128-bit AES encryption and Java 1.7 supports 40-bit AES encryption. You can install additional encryption libraries by BouncyCastle into the JVM allowing for up to 256-bit AES encryption.

The encryption process in the LDAP gateway also allows for automatic migration of encryption values from a lower bit strength to a higher strength as it becomes available. For example, if the gateway is initially deployed on a system running Java 1.7 with 40-bit AES and that system is upgraded to Java 1.8 running 128-bit AES, then upon the next restart of the gateway, all encrypted values remaining at the 40-bit AES level are automatically re-encrypted at the higher 128-bit and stored back out in the property files. This process eliminates the need to manually replace the values in every property file in order to take advantage of the higher bit strength.

The gateway uses the private key located in the `LDAP_INSTALL_DIR/conf/idf.properties` file for all the encryption and decryption that it performs. The idf.properties file is created in the conf directory when the LDAP gateway is started for the first time. It is recommended that access to this file is restricted.

> **Note:**
>
> Once the gateway is deployed and started for the first time, the value of the autogenerated encryption key in the idf.properties file should not be changed. However, you can change the file name and its location. For example, to store the idf.properties file to a more secure location, the default location (where the gateway resides) can be overwritten and defined as `system.idfprops.filepath=ABSOLUTE_PATH_OF_THE_NEW_FILE` in the customerconfiguration. properties file.

2.8.2 Configuring Encryption

You can configure encryption by editing the `encryption.properties` file located in the `LDAP_INSTALL_DIR/conf/` directory.

By default, the LDAP gateway encrypts the values of:

- the `adminUserPassword` and `altAdminPassword` properties in the `LDAP_INSTALL_DIR/conf/customer-configuration.properties` file.
- the `_secretKeyValue_` property in the connection properties file (created in Setting Connection Properties).
- If you want to encrypt additional properties in the `customer-configuration.properties` file, then you must include them as a comma-separated list in the following property of the `encryption.properties` file: `file.customer-configuration=adminUserPassword,altAdminUserPassword`

For example, if you want to encrypt the `schema` and `suffix` properties of the `customer-configuration.properties` file, then include them in the`file.customer-configuration` property of the `encryption.properties` file as follows:

```
file.customer-
configuration=adminUserPassword,altAdminUserPassword,schema,suffix
```

- If you want to encrypt additional properties in the connection properties file, then include them as a comma-separated list in the following property of the `encryption.properties` file: `class.RacfModule=_secretKeyValue_`

  For example, if you want to encrypt the `_host_` and `_port_` properties of the connection properties file, then include them in the `class.RacfModule=_secretKeyValue_` property of the `encryption.properties` file as follows:

  ```
  class.RacfModule=_secretKeyValue_,_host_,_port_
  ```

- If you want to change the values any encrypted properties, then remove the `ENC` along with the value and then add the new value.

  For example, if the value of the `adminUserPassword` property in the `customer-configuration.properties` file is encrypted, then from the `adminUserPassword=ENC(t8+B0TbafPKyFFf0KoTlAmde82aRnwtf)` value, remove `ENC(t8+B0TbafPKyFFf0KoTlAmde82aRnwtf)` and replace it with the new value, without the prefix ENC. Whenever the gateway is restarted, it automatically overwrites the clear-text value with its encrypted counterpart.

2.9 Understanding the Caching Layer

The LDAP gateway features an optional and configurable caching layer, which is a temporary storage area where frequently accessed data is stored for rapid access.

An expiration policy defines the time dependency for the cached resource. For example, the `cachingMaxAge` parameter specifies the maximum time in minutes when the data is not in sync with the target system. You can pair the caching layer with an incremental reconciliation (to maintain the most recently updated data in the caching layer. This improves the performance of the LDAP gateway. In addition, the caching layer opens the LDAP gateway for more advanced features defined by the LDAPv3 RFC.

Benefits of Using the Caching Layer

Using the caching layer provides the following benefits:

- Faster search operations (when the cache is primed)
- A unified Base DN for both provisioning and reconciliation data

When paired with an embedded directory server, the caching layer offers these additional benefits:

- The ability to perform advanced LDAP search filters against the gateway.
- The ability to query an RFC compliant ChangeLog for delta reconciliation.

> **Note:**
>
> In an environment where the items noted above may not be required, you can disable the caching layer.

Considerations for Using the Caching Layer

The LDAP gateway can suffer a performance penalty when all of the following conditions are met:

- There is no data in the cache, or the cache is stale based on the configuration.
- An LDAP search operation is performed to retrieve the children of an Organizational Unit. For example, the contents of `ou=People`. Such an LDAP search operation returns only DNs (along with RDN components).
- The connector only returns *key* information when returning a list of objects.
- The `cachingIterateBehavior` property in the *`LDAP_INSTALL_DIR`*`/conf/configuration.properties` file remains set to the default of `AUTO` and not overwritten within the `customer-configuration.properties` file.
  In such a scenario, an LDAP search operation initially retrieves the list of results, containing only DN and RDN values. The caching layer then iterates through each result, fetching and caching the details from the target system. Finally, the full set of results are returned to Oracle Identity Manager.

To avoid this scenario, it is recommended that you use the caching layer in combination with scheduled reconciliation. With reconciliation setup and the staleness settings configured properly the above conditions will not be met.

How to Enable or Disable the Caching Layer?

The caching layer is enabled by default. To override this default setting or disable the caching layer, copy the `cnctr.coreBean.nexus.cachingEnabled` property from the *`LDAP_INSTALL_DIR`*`/conf/configuration.properties` file to the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file and then set its value to `false`.

You can enable the caching layer by setting the value of the `cnctr.coreBean.nexus.cachingEnabled` property in the *`LDAP_INSTALL_DIR`*`/conf/customer-configuration.properties` file to `true` .

2.10 Configuring Scheduled Reconciliation

Scheduled reconciliation allows for establishing a periodic synchronization between the Identity Store associated with the LDAP Gateway and that represented by your target system reachable by way of the connector.

The Scheduled Recon Utility (provided by *`LDAP_INSTALL_DIR`*`/dist/scheduled-recon.jar`) is a tool that ships with the IdentityForge LDAP Gateway. It provides the ability to perform a full recon against a configurable target system, placing the results in the internal identity store of the gateway. This utility provides a basic scheduling service for kicking off the built-in batched reconciliation of the connector on a configurable interval.

An example properties file, scheduled-recon.properties.example file that defines the reconciliation setup and behavior is available in the *`LDAP_INSTALL_DIR`*`/conf` folder. Use this file to configure scheduled reconciliation.

1. In the *`LDAP_INSTALL_DIR`*`/conf` directory, create a copy of the *`LDAP_INSTALL_DIR`*`/conf/scheduled-recon.example` file and rename it `scheduled-recon.properties`.
2. If required, open the *`LDAP_INSTALL_DIR`*`/conf/scheduled-recon.properties` file in a text editor and configure it to meet your requirements.

3. Run the *LDAP_INSTALL_DIR*/bin/run-recon.bat file to start the scheduled recon utility.

   You can run this batch file with the following options

Argument	Description
-h	Use this argument for help.
-loglevel <*level*>	Use this argument to define the logging level. Possible values are: • severe • warning • info • fine • finer • finest Default value is warning.
-logfile <*filepath*>	Use this argument to specify the path to the log file.
-p <*properties filepath*>	Use this argument to specify the path to the scheduled-recon.properties file.

   The following is the basic command structure for executing this batch file:

```
…\ldapgateway6\bin>run-recon.bat -loglevel "warning" -logfile <location of the log file> -p "D:\ldapgateway6\conf\scheduled-recon.properties"
```

2.11 About Parsing Grammar Protocol 1.0

Grammar is necessary for properly parsing user and group listings that come into the gateway from the mainframe agent during search requests and reconciliation events.

The grammar represents line-by-line parsing instructions that convert the semi-structured textual data into LDAP attributes and their respective values. Each line (ending in CRLF) of the listing received from the agent can be represented by an individual grammar definition and specified in the grammar file.

Grammar files with the default grammar are present in the *LDAP_INSTALL_DIR*/conf/parser-grammars/racf directory. It parses user and group listings that come into the gateway from the mainframe agent during search requests and reconciliation events.

For example, following is the user listing for IBM RACF Advanced:

```
USER=HBCMXJHB  NAME=HBCMXJHB
 OWNER=IDFAGNT  CREATED=19.214
 DEFAULT-GROUP=IDFSGRP
 PASSDATE=00.000 PASS-INTERVAL=180 PHRASEDATE=N/A
 ATTRIBUTES=NONE
 REVOKE DATE=NONE  RESUME DATE=NONE
 LAST-ACCESS=UNKNOWN
 CLASS AUTHORIZATIONS=NONE
 INSTALLATION-DATA=NEW VALUE
 NO-MODEL-NAME
 LOGON ALLOWED   (DAYS)          (TIME)
 ---------------------------------------------
 ANYDAY                          ANYTIME
 GROUP=IDFSGRP   AUTH=USE      CONNECT-OWNER=IDFAGNT   CONNECT-DATE=19.214
 CONNECTS=    00  UACC=NONE     LAST-CONNECT=UNKNOWN
 CONNECT ATTRIBUTES=NONE
 REVOKE DATE=NONE   RESUME DATE=NONE
 SECURITY-LEVEL=NONE SPECIFIED
 CATEGORY-AUTHORIZATION
 NONE SPECIFIED
 SECURITY-LABEL=NONE SPECIFIED
 TSO INFORMATION
 ***
---------------
ACCTNUM= 23456
HOLDCLASS= M
JOBCLASS= I
MSGCLASS= Q
PROC= TEST
SIZE= 00000001
MAXSIZE= 00000006
SYSOUTCLASS= Y
USERDATA= 6789
COMMAND= newcmd

CICS INFORMATION
----------------
 OPCLASS= 002
 OPIDENT= 1
IRR52117I LISTING exit DFHSNPTO not found.
 OPPRTY= 00001
 TIMEOUT=
 XRFSOFF= FORCE
 RSLKEYS= 00099
 TSLKEYS= 00099
***
```

Using the above listing, if you want to parse out the OPCLASS value from the listing and assign it to an LDAP attribute called “opcls”, then you can construct the following <Line> element in the grammar file:

```
<Line id=”opclassVal” enabled=”yes” sig=”[ ]*OPCLASS = (?&lt;opcls&gt;.*)"/>
```

The signature attribute (sig) in the Line element above is a regex that represents the rules for pulling out the value and assigning it to an LDAP attribute. Regex named groups are used as the convention for assigning the discovered values to LDAP attributes exposed through the connector.

The following table lists the attributes of a line element. The allowed values for these attributes are **yes**or **no**.

String	Mandatory?	Definition
id	Yes	Unique ID that is given to the line definition. Used primarily for internal referencing purposes, such as with the 'dependson' attribute. Values allowed: `any`

String	Mandatory?	Definition
enabled	No	Specifies whether the line is eligible for participating in the parsing process. Use this flag to override files (turn off lines). Default value: yes
signature	Yes	Defines the rules for what values are to be extracted for each line of the listing and which LDAP attributes should be assigned the values.
required	no	Defines whether an attribute is required or not. Default to: `yes`
multiline_sig	No	An optional regex expression to define the signature of a follow-on line that could represent whether the value was wrapped around two additional lines in the document. Values allowed: Any valid regex containing attribute matching key and attribute name. Defaults to: `empty value`
repeats	No	Represents whether the line can show up multiple times in the document. If set to `no`, then once the line is found, this Line definition is not evaluated again for the rest of the document. Defaults to: `No`
overflow	No	Represents whether data for an associated attribute can overflow to the next line. In case of an overflow, the final value of an attribute is derived by concatenating all values. Defaults to: `No`
multivalue_parser	No	An optional regex expression that defines how the found values are to be parsed out and turned into a multivalued list, such as using '(\S+)' to parse values that are space delimited. Values Allowed: Any valid regex Defaults to: `empty value`
applyCompositeRef	no	An optional comma-separated list of composite attributes to be built immediately after processing the line. Each value in the comma-separated list must correspond to the "id" attribute of a CompositeAttribute definition.

String	Mandatory?	Definition
defaultvalue	No	Defines the default value for an attribute. If this line does not match with any line of input, then this default value will be assigned to attribute.

Customizing Grammar Rules

You can apply new grammar rules to append to or override rules that are available by default in the *LDAP_INSTALL_DIR*/conf/parser-grammars/racf directory.

To define new grammar rules or override the existing rules, you must create a custom grammar file (for example, racf_FindAllPeople.cust) in the *LDAP_INSTALL_DIR*/conf/parser-grammars/racf directory.

> **Note:**
>
> - If the Id of the existing attribute matches with the attribute in the grammar line, it overrides the existing grammar definition.
> - If the Id of the existing attribute does not match with the attribute in the grammar line, it creates a new grammar definition.

Key Considerations

- The parser-grammars.cust grammar file must be at the same location where the default grammar files are located (*LDAP_INSTALL_DIR*/conf/parser-grammars/racf).
- The name of the grammar file must be the same except the cust extension. For example, if you need to customize the grammar for the *LDAP_INSTALL_DIR*/conf/parser-grammars/racf/racf_FindAllPeople.xml file, then create a custom grammar file *LDAP_INSTALL_DIR*/conf/parser-grammars/racf/racf_FindAllPeople.cust.
- For the grammar definitions to override, the ID attribute from both the files should match.

Nomenclature of the parsing grammar files

Each grammar file is named for the type of operation and listing it is responsible for parsing.

For example, for RACF, use the following for user extraction:

- racf_FindAllPeople.xml – fetches the IDs of all users.
- racf_ExtractUserById.xml – fetches all the details of a single user (for the given ID).

Overriding default existing grammar definitions

The grammar definitions specified in the custom grammar file override the default grammar definitions. To enable overriding of the particular line, the ID attribute in the custom provided attribute should match with the default grammar definition.

For example, if the default grammar definition in the property file and the definition specified in the custom grammar file is as shown in the following lines, then the definition is disabled and the line is not parsed.

```
<Protocol><Lines>
<Line id="elId" enabled="yes" sig="[ ]*ELID[ ]*=[ ]*(?<ELID>.*)"/>
</Lines>/Lines>


<Protocol><Lines>
<Line id="elId" enabled="no" sig="[ ]*ELID[ ]*=[ ]*(?<ELID>.*)"/>
</Lines></Protocol>
```

New grammar definitions

New grammar definitions can be specified in the custom grammar file. For example, the following grammar definition is used to get values of `DEPT_ACID=001 DEPT_NAME=hr`.

```
<Protocol><Lines>
="deptAcid" enabled="yes" sig="[ ]*DEPT_ACID[ ]*=[ ]*(?&lt;deptacid&gt;.*?)
[ ]*DEPT_NAME[ ]*=[ ]*(?&lt;department&gt;.*)" />
</Lines></Protocol>
```

2.12 Configuring IDF LDAP Gateway to Use SSL for Messaging Between Gateway and Pioneer/Voyager

Configuring IDF LDAP Gateway to use SSL for messaging between Gateway and Pioneer/Voyager involves the following steps:

> **Note:**
>
> - LDAP Gateway requires JAVA JDK 1.7 or above.
> - Oracle recommends installing JAVA in a directory whose name is without spaces, for example, c:\software.

- Configuring SSL for Messaging Between Gateway and Pioneer
- Configuring SSL for Messaging Between Gateway and Voyager
- Enabling AT-TLS for RACF Pioneer and Voyager

2.12.1 Configuring SSL for Messaging Between Gateway and Pioneer

To configure SSL for messaging between Gateway and Pioneer:

1. Import certificate in LDAP Gateway's trust store. To do so:
   a. Get the certificate from Pioneer agent in form of PEM file. See step 1 through step 9 in Enabling AT-TLS for RACF Pioneer and Voyager.

   b. Import the certificate into a trust store using the JAVA keytool command, as shown:

      ```
      keytool -import -file <certificate file> -keystore <keystore-file>
      ```

      For example:

      ```
      keytool -import -file pioneer-cert.dat -keystore cacerts
      ```

2. In the `racf.properties` file, enable SSL by using the following config parameters:

   ```
   # Set sslEnabled to true if the agent supports SSL messaging
   sslEnabled=false
   # set the TLS version if sslEnabled is set to true
   # This can be set to TLSv1.1, TLSv1.2 and can be extended to TLSv1.3 in future if
   required
   tlsVersion=TLSv1.2
   ```

2.12.2 Configuring SSL for Messaging Between Gateway and Voyager

To configure SSL for messaging between Gateway and Voyager:

1. Generate a certificate for Voyage by running the following command:

   ```
   keytool -genkey -keyalg RSA -alias <certificate alias> -keystore < keystore-file> -
   keysize 2048
   ```

   For example:

   ```
   keytool -genkey -keyalg RSA -alias gatewayCert -keystore keystore.jks -keysize 2048
   ```

2. Export the certificate for Voyage to a file by running the following command:

   ```
   keytool -export -alias <certificate alias> -file <certificate file> -keystore
   <keystore-file> -keysize 2048
   ```

   For example:

   ```
   keytool -export -alias gatewayCert -file gatewayCert.dat -keystore keystore.jks -
   keysize 2048
   ```

3. Import the certificate into Voyage agent. See step 10 in Enabling AT-TLS for RACF Pioneer and Voyager.
4. While starting the gateway, specify the keystore and the password along with the java executable command using the parameters `-Djavax.net.ssl.keyStore` and `-Djavax.net.ssl.keyStorePassword`.

2.12.3 Enabling AT-TLS for RACF Pioneer and Voyager

Using AT-TLS or TLS with Voyager and Pioneer require z/OS system definitions and RACF definitions.

To enable AT-TLS for RACF Pioneer and Voyager:

1. Create PAGENT STC (Started Task).
2. Create SYSLOGD STC (Started Task).
3. Create required parameters for PAGENT and SYSLOGD.
4. Modify TCPIP STC (Started Task) Profile to support TTLS.
5. Generate the Certificate on z/OS Unix System Services using `gskkyman`.

6. Add a Certificate as `RSA`, `keysize = 2048`, and `Private Key = YES`.
7. Create a RACF definitions for PAGENT, SYSLOGD, Pioneer, and Voyager.

   The following are sample PAGENT config parameters used in testing on z/OS 2.2 Put1606.

```
OMVS - /etc/pagent.env
LIBPATH=/usr/lib
TZ=EST5EDT
PAGENT_CONFIG_FILE=/etc/pagent.conf
PAGENT_LOG_FILE=SYSLOGD
GSK_RENEGOTIATION=ALL
GSK_PROTOCOL_TLSV1_2=ON

OMVS - /etc/pagent.conf
loglevel 255
TcpImage TCPIP /etc/pagent_policy.conf FLUSH PURGE

OMVS - /etc/pagent_policy.conf
# Shows AT-TLS events and result of each System SSL call
TTLSGroupAction grp_Diagnostic
{
TTLSEnabled On
Trace 6 # Log Error, Info, Event and Flow to syslogd
}
TTLSRule Pioneer_Server
{
LocalPortRange 6000 # Pioneer STC for IDMWORKS
Direction Inbound
Priority 1 # Base Priority
TTLSGroupActionRef grp_Diagnostic
TTLSEnvironmentActionRef Pioneer_List_Env
}
TTLSEnvironmentAction Pioneer_List_Env
{
HandshakeRole Server
TTLSKeyRingParmsRef PIONEERING
TTLSCipherParmsRef Pioneer_cipher_list
}
TTLSKeyRingParms PIONEERING
{
Keyring PIONEERING
}
TTLSCipherParms Pioneer_cipher_list
{
V3CipherSuites TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
```

```
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_RSA_WITH_RC4_128_SHA
V3CipherSuites TLS_RSA_WITH_RC4_128_MD5
V3CipherSuites TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
V3CipherSuites TLS_RSA_EXPORT_WITH_RC4_40_MD5
}
TTLSRule Voyager_Client
{
RemotePortRange 5197 # Voyager STC for IDMWORKS
Direction Outbound
TTLSGroupActionRef grp_Diagnostic
TTLSEnvironmentActionRef Voyager_List_Env
}
TTLSEnvironmentAction Voyager_List_Env
{
HandshakeRole Client
TTLSKeyRingParmsRef PIONEERING
TTLSCipherParmsRef Voyager_cipher_list
}
TTLSKeyRingParms PIONEERING
{
Keyring PIONEERING
}
TTLSCipherParms Voyager_cipher_list
{
V3CipherSuites TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_RSA_WITH_RC4_128_SHA
V3CipherSuites TLS_RSA_WITH_RC4_128_MD5
V3CipherSuites TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
```

```
V3CipherSuites TLS_RSA_EXPORT_WITH_RC4_40_MD5
}
```

RACF required definitions are:

```
//PAGRACD JOB SYSTEMS,MSGLEVEL=(1,1),MSGCLASS=X,CLASS=A,PRTY=8,
// NOTIFY=&SYSUID,REGION=4096K
//TSOX EXEC PGM=IKJEFT01,DYNAMNBR=99
//SYSTSPRT DD SYSOUT=*
//SYSTSIN DD *
ADDUSER PAGENT NAME('PAGENT-ATTLS') DFLTGRP(OMVSGRP) -
OMVS(UID(0) HOME('/'))
ADDUSER SYSLOGD NAME('SYSLOGD-UNIX') DFLTGRP(OMVSGRP) -
OMVS(UID(0) HOME('/'))
RALTER STARTED PAGENT.* STDATA(USER(SYSLOGS)
SETROPTS CLASSACT(OPERCMDS)
RDEFINE OPERCMDS (MVS.SERVMGR.PAGENT) UACC(NONE)
PERMIT MVS.SERVMSG.PAGENT CLASS(OPERCMDS) ACCESS(CONTROL) -
ID(PAGENT)
SETROPTS RACLIST(OPERCMDS) REFRESH
SETROPTS CLASSACT(STARTED)
SETROPTS RACLIST(STARTED)
SETROPTS GENERIC(STARTED)
RDEFINE STARTED PAGENT.* UACC(NONE)
SETROPTS RACLIST(STARTED) REFRESH
SETROPTS GENERIC(STARTED) REFRESH
SETROPTS CLASSACT(DIGTCERT DIGTRING)
RDEFINE FACILITY IRR.DIGTCERT.LISTRING UACC(NONE)
RDEFINE FACILITY IRR.DIGTCERT.LIST UACC(NONE)
PERMIT IRR.DIGTCERT.LIST CLASS(FACILITY) ID(TCPIP) ACCESS(READ)
PERMIT IRR.DIGTCERT.LISTRING CLASS(FACILITY) ID(TCPIP) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
RACDCERT ID(PIONEER) -
GENCERT SUBJECTSDN(CN('IDMWORKS.COM') -
O('IDMWORKS') -
OU('IDF ZOS22 SERVER') -
C('US')) -
WITHLABEL('PIONEER TESTER')
RACDCERT ID(PIONEER) ADDRING(PIONEERING)
RACDCERT ID(PIONEER) CONNECT(ID(PIONEER) -
LABEL('PIONEER SERVER') -
RING(PIONEERING) -
DEFAULT -
USAGE(PERSONAL)
/*
```

8. Import Certificate into RACF:

   ```
   RACDCERT ID(PIONEER) EXPORT(LABEL('PIONEER TESTER')) - FORMAT(CERTB64)
   DSN('PIONEER.CERT.FILE')
   ```

9. FTP the Certificate DSN=PIONEER.CERT.FILE to the LDAP.
10. FTP LDAP Certificate to z/OS into a sequential file.
11. CONNECT Certificate to Voyager:

```
RACDCERT CERTAUTH ADD('VOYAGER.CERT.LDAP') TRUST -
WITHLABEL('LABEL00000001')
RACDCERT ID(VOYAGER) CONNECT(CERTAUTH -
LABEL('LABEL00000001') -
RING(VOYAGERING))
```

2.13 Configuring Replication

You can achieve a highly available LDAP Gateway by setting up two instances of embedded OpenDJ in a master-to-master replication model.

The IdentityForge LDAP Gateway features an optional and configurable replication setup. The LDAP Gateway uses Embedded DS to store objects in internal store. You can set up two instances of Embedded DS (within the gateway) in master to master replication. In order to use the replication benefits, the connector must use internal Embedded DS to store the objects. This is configured by using the `metaBackend` property in the `customer-configuration.properties` file. The `metaBackend` property must use ldapds to indicate the internal store provider. For example `cnctr.saphr.saphr1.metaBackend=ldapds`.

The replication set up that offers the following benefits:

- Provides a way to use load balancer to distribute load between two gateway instances.
- Achieves high availability. The master to master replication ensures that at any given point in time, the copies in both instances are in sync, and if one of the machines or instances goes down, the service is not interrupted.

> **Note:**
>
> Before you perform the replication procedure, ensure that:
>
> - You have LDAP Gateway version 6.6.6 or later installed.
> - You create a backup of the internal store of the LDAP Gateway.

> **Note:**

To set up replication between two instances (for example server 1 and server 2) of the LDAP Gateway:

1. Stop the LDAP Gateway on server 1.
2. Create a backup of the *LDAP_INSTALL_DIR*`/dsroot` directory by copying and storing its contents outside the *LDAP_INSTALL_DIR*`/dsroot` directory.
3. Delete all the files and folders within the *LDAP_INSTALL_DIR*`/dsroot` directory.
4. Copy the contents of *LDAP_INSTALL_DIR*`/doc/samples/replication/replicated-dsroot-server1` directory to the *LDAP_INSTALL_DIR*`/dsroot` directory.
5. In a text editor, open the *LDAP_INSTALL_DIR*`/dsroot/config/config.ldif` file for editing, and then:
    a. Search for and replace all instances of <*this_server_ip_address_or_hostname*> with the hostname or IP address of the current server (for example, server 1).

   b. Search for and replace all instances of <*replication_server_ip_address_or_hostname*> with the hostname or IP address of the replicated server (for example, server 2).

   c. Save and close the config.ldif file.

6. Restart the LDAP Gateway.

7. Repeat Steps 1 through 6 on server 2 with the following difference:

   While performing Step 4 of this procedure, instead of copying the contents of the `LDAP_INSTALL_DIR/doc/samples/replication/replicated-dsroot-server1` directory, copy the contents of the `LDAP_INSTALL_DIR/doc/samples/replication/replicated-dsroot-server1` directory to the `LDAP_INSTALL_DIR/dsroot` directory.

3
IBM RACF Connector Deployment on Oracle Identity Manager

The LDAP Gateway acts as the intermediary between Oracle Identity Manager and the connector components on the mainframe. The following sections of this chapter describe the procedure to deploy some components of the connector, including the LDAP Gateway, on the Oracle Identity Manager host computer:

> **Note:**
>
> The procedure to deploy the mainframe components of the connector is described in the next chapter.

- Running the Connector Installer
- Configuring the IT Resource
- Configuring Oracle Identity Manager

3.1 Running the Connector Installer

Perform the following steps to run the Connector Installer:

1. Ensure you have downloaded the connector installation package from the OTN website at http://www.oracle.com/technetwork/middleware/id-mgmt/downloads/connectors-101674.html and extracted its contents.
2. Copy the contents of the connector installation package into the following directory:

   OIM_HOME/server/ConnectorDefaultDirectory
3. Log in to Oracle Identity System Administration.
4. In the left pane, under Provisioning Configuration, click **Manage Connector**.
5. In the Manage Connector page, click **Install.**
6. From the Connector list, select **IBM RACF Advanced *RELEASE_NUMBER*.** This list displays the names and release numbers of connectors whose installation files you copy into the default connector installation directory in Step 2.

   If you have copied the installation files into a different directory, then:

   a. In the **Alternative Directory** field, enter the full path and name of that directory.
   b. To repopulate the list of connectors in the Connector list, click **Refresh.**
   c. From the Connector list, select **IBM RACF Advanced *RELEASE_NUMBER*.**
7. Click **Load.**
8. To start the installation process, click **Continue.** In a sequence, the following tasks are automatically performed:

a. Configuration of connector libraries.

b. Import of the connector Target Resource user configuration XML file (by using the Deployment Manager).

c. Compilation of adapters.

On successful completion of a task, a check mark is displayed for the task. If a task fails, then an X mark and a message stating the reason for failure are displayed. If a task fails, make the required correction and then perform one of the following steps:

- Retry the installation by clicking **Retry.**
- Cancel the installation and begin again from Step 2.

9. If all three tasks of the connector installation process are successful, then a message indicating successful installation is displayed.
10. Click **Exit** to close the installation page.

When you run the Connector Installer, it copies the connector files and external code files to destination directories on the Oracle Identity Manager host computer. These files are listed in Files and Directories in the IBM RACF Advanced Connector Package.

3.2 Configuring the IT Resource

You must specify values for the parameters of the RacfResource IT resource as follows:

1. Log in to the Oracle Identity System Administration.
2. In the left pane, under Configuration, click **IT Resource.**
3. In the IT Resource Name field on the Manage IT Resource page, enter `RacfResource` and then click **Search.**
4. Click the edit icon for the IT resource.
5. From the list at the top of the page, select **Details and Parameters.**
6. Specify values for the parameters of the IT resource as described in the following table:

Table 3-1 IT Resource Parameters for IBM RACF Advanced Connector

Parameter	Description
AtMap User	This parameter holds the name of the lookup definition containing attribute mappings that are used for provisioning. Value: `AtMap.RACF` **Note:** You must not change the value of this parameter.
idfBackendDn	Enter the user ID that the connector will use to connect to the LDAP Gateway backend. Sample value: `cn=Directory Manager,dc=system,dc=backend`
idfBackendPassword	Enter the password of the user ID that the connector will use to connect to the LDAP Gateway backend. You also set this password in the configuration.properties file of the LDAP Gateway. **Note:** Do not enter an encrypted value.
idfbackendContext	Enter the root context for LDAP Gateway backend. Sample Value: `dc=system,dc=backend`

Table 3-1 (Cont.) IT Resource Parameters for IBM RACF Advanced Connector

Parameter	Description
idfConnectTimeoutMS	Enter an integer value that specifies the number of milliseconds after which an attempt to establish a connection between the LDAP Gateway and Oracle Identity Manager times out. If you do not enter a value for this parameter, then the connector uses a default time out of `300000` ms (that is, 5 minutes).
idfPrincipalDn	Set a user ID for an account that the connector will use to connect to the LDAP Gateway. Format: `cn=`*`USER_ID`*`,dc=racf,dc=com` Sample value: `cn=idfRacfAdmin,dc=racf,dc=com`
idfPrincipalPwd	Set a password for the account that the connector will use to connect to the LDAP Gateway. You also set this password in the files listed in the description of the idfPrincipalDn parameter. **Note:** Do not enter an encrypted value.
idfReadTimeoutMS	Enter an integer value that specifies the number of milliseconds after which an attempt to read data from the target system times out. If you do not enter a value for this parameter, then the connector uses a default time out of `1800000` ms (that is, 30 minutes).
idfRootContext	This parameter holds the root context for IBM RACF. Value: `dc=racf,dc=com` **Note:** You must not change the value of this parameter.
idfServerHost	This parameter holds the host name or IP address of the computer on which you install the LDAP Gateway. For this release of the connector, you install the LDAP Gateway on the Oracle Identity Manager host computer. Default value: `localhost` **Note:** Do not change the value of this parameter unless you have installed the LDAP Gateway on a different machine from the Oracle Identity Manager host computer.
idfServerPort	Enter the number of the port for connecting to the LDAP Gateway. Sample value: `5389`
idfSsl	This parameter determines whether the LDAP Gateway will use SSL to connect to the target system. Enter `true` if using SSL. Otherwise, enter `false.` Sample value: `true`
idfTrustStore	This parameter holds the directory location of the trust store containing the SSL certificate. This parameter is optional, and should only be entered when using SSL authentication. This must be the full path to the directory location. Sample value: `/app/home/ldapgateway/conf/idf.jks`
idfTrustStorePassword	This parameter holds the password for the SSL trust store. This parameter is optional, and should only be entered when using SSL authentication.
idfTrustStoreType	This parameter holds the trust store type for the SSL trust store. This parameter is optional, and should only be entered when using SSL authentication. Sample value: `jks`

Table 3-1 (Cont.) IT Resource Parameters for IBM RACF Advanced Connector

Parameter	Description
Last Modified Time Stamp	The most recent start time of the RACF Reconcile All LDAP Users reconciliation scheduled task is stored in this parameter. See RACF Reconcile All LDAP Users for more information about this scheduled task. The format of the value stored in this parameter is as follows: MM/dd/yy hh:mm:ss a In this format: MM is the month of the year. dd is the day of the month. yy is the year. hh is the hour in am/pm (01-12). mm is the minute in the hour. ss is the second in the minute. a is the marker for AM or PM. Sample value: `05/07/10 02:46:52 PM` Default value: `0` The reconciliation task will perform full LDAP user reconciliation when the value is 0. If the value is a non-zero, standard time-stamp value in the format given above, then incremental reconciliation is performed. Only records that have been created or modified after the specified time stamp are brought to Oracle Identity Manager for reconciliation. **Note:** When required, you can manually enter a time-stamp value in the specified format.

7. To save the values, click **Update.**

3.3 Configuring Oracle Identity Manager

Configuring Oracle Identity Manager involves the following procedures:

> **Note:**
>
> In an Oracle Identity Manager cluster, you must perform these steps on each node of the cluster.

- Creating Additional Metadata, Running Entitlement, and Catalog Synchronization Jobs
- Localizing Field Labels in UI Forms
- Clearing Content Related to Connector Resource Bundles from the Server Cache for Oracle Identity Manager Connector
- Enabling Logging for IBM RACF Advanced Connector

3.3.1 Creating Additional Metadata, Running Entitlement, and Catalog Synchronization Jobs

You must create additional metadata, such as a UI form and an application instance. In addition, you must run entitlement and catalog synchronization jobs. These procedures are described in the following sections:

- Creating and Activating a Sandbox
- Creating a New UI Form
- Creating an Application Instance
- Publishing a Sandbox
- Harvesting Entitlements and Sync Catalog
- Updating an Existing Application Instance with a New Form

3.3.1.1 Creating and Activating a Sandbox

Create and activate a sandbox as follows:

1. On the upper navigation bar, click **Sandboxes.** The Manage Sandboxes page is displayed.
2. On the toolbar, click **Create Sandbox.** The Create Sandbox dialog box is displayed.
3. In the Sandbox Name field, enter a name for the sandbox. This is a mandatory field.
4. In the Sandbox Description field, enter a description of the sandbox. This is an optional field.
5. Click **Save and Close.** A message is displayed with the sandbox name and creation label.
6. Click **OK.** The sandbox is displayed in the Available Sandboxes section of the Manage Sandboxes page.
7. From the table showing the available sandboxes in the Manage Sandboxes page, select the newly created sandbox that you want to activate.
8. On the toolbar, click **Activate Sandbox.**

   The sandbox is activated.

3.3.1.2 Creating a New UI Form

Create a new UI form as follows:

1. In the left pane, under Configuration, click **Form Designer.**
2. Under Search Results, click **Create.**
3. Select the resource type for which you want to create the form, for example, **OIMRacfResourceObject.**
4. Enter a form name and click **Create.**

3.3.1.3 Creating an Application Instance

Create an application instance as follows:

1. In the System Administration page, under Configuration in the left pane, click **Application Instances.**
2. Under Search Results, click **Create.**
3. Enter appropriate values for the fields displayed on the Attributes form and click **Save.**
4. In the Form drop-down list, select the newly created form and click **Apply.**

5. Publish the application instance to an organization to make the application instance available for requesting and subsequent provisioning to users.

3.3.1.4 Publishing a Sandbox

Before publishing a sandbox, perform the following procedure as a best practice to validate all sandbox changes made till this stage as it is difficult to revert the changes after a sandbox is published:

1. In Identity System Administration, deactivate the sandbox.
2. Log out of Identity System Administration.
3. Log in to Identity Self Service using the xelsysadm user credentials and then activate the sandbox that you deactivated in Step 1.
4. In the Catalog, ensure that the Concur application instance form appears with correct fields.
5. Publish the sandbox. See Publishing a Sandbox in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager*.

3.3.1.5 Harvesting Entitlements and Sync Catalog

To harvest entitlements and sync catalog:

1. Run the scheduled jobs for lookup field synchronization. See Scheduled Tasks for Lookup Field Synchronization for more information about these scheduled jobs.
2. Run the Entitlement List scheduled job to populate Entitlement Assignment schema from child process form table.
3. Run the Catalog Synchronization Job scheduled job.

> **See Also:**
>
> Predefined Scheduled Tasks in *Oracle Fusion Middleware Administering Oracle Identity Governance* for a description of the Entitlement List and Catalog Synchronization Job scheduled jobs.

3.3.1.6 Updating an Existing Application Instance with a New Form

For any changes you do in the Form Designer, you must create a new UI form and update the changes in an application instance. To update an existing application instance with a new form:

1. Create a sandbox and activate it as described in Creating and Activating a Sandbox.
2. Create a new UI form for the resource as described in Creating a New UI Form.
3. Open the existing application instance.
4. In the **Form** field, select the new UI form that you created.
5. Save the application instance.
6. Publish the sandbox as described in Publishing a Sandbox.

3.3.2 Localizing Field Labels in UI Forms

Perform the following steps to localize field labels that you add to in UI forms:

1. Log in to Oracle Enterprise Manager.
2. In the left pane, expand **Application Deployments** and then select **oracle.iam.console.identity.sysadmin.ear.**
3. In the right pane, from the Application Deployment list, select **MDS Configuration.**
4. On the MDS Configuration page, click **Export** and save the archive to the local computer.
5. Extract the contents of the archive, and open the following file in a text editor:

   SAVED_LOCATION\xliffBundles\oracle\iam\ui\runtime\BizEditorBundle.xlf
6. Edit the BizEditorBundle.xlf file as follows:
   a. Search for the following text:

      ```
      <file source-language="en"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

   b. Replace with the following text:

      ```
      <file source-language="en" target-language="LANG_CODE"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

      In this text, replace *LANG_CODE* with the code of the language that you want to localize the form field labels. The following is a sample value for localizing the form field labels in Japanese:

      ```
      <file source-language="en" target-language="ja"
      original="/xliffBundles/oracle/iam/ui/runtime/BizEditorBundle.xlf"
      datatype="x-oracle-adf">
      ```

   c. Search for the application instance code. The original code will be in the following format:

      ```
      <trans-unit id="$
      {adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']
      ['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.UD_<Fiel
      d_Name>__c_description']}">
      <source><Field_Label></source>
      <target/>
      </trans-unit>
      <trans-unit
      id="sessiondef.oracle.iam.ui.runtime.form.model.<UI_Form_Name>.entity.
      <UI_Form_Name>EO.UD_<Field_Name>__c_LABEL">
      <source><Field_Label></source>
      <target/>
      </trans-unit>
      ```

      For example, the following sample code show the update that should be made for the FULL NAME field on a UI form named RacfUserFormv1:

      ```
      <trans-unit id="$
      {adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']
      ['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.UD_IDF_R
      ACF_ADV_CN__c_description']}">
      <source>FULL NAME</source>
      ```

```
<target/>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.RacfUserFormv1.entity.RacfUserFor
mv1EO.UD_IDF_RACF_ADV_CN__c_LABEL">
<source>FULL NAME</source>
<target/>
</trans-unit>
```

d. Open the resource file from the /resources directory in the connector installation media, for example Racf-Adv_ja.properties, and get the value of the attribute from the file, for example global.udf.UD_IDF_RACF_ADV_CN=\u6C0F\u540D.

e. Replace the original code shown in Step 6.c with the following:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.UD_<Fiel
d_Name>__c_description']}">
<source>< global.udf.UD_Field_Name></source>
<target/>enter Unicode values here</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.<UI_Form_Name>.entity.
<UI_Form_Name>EO.UD_<Field_Name>__c_LABEL">
<source><Field_Label></source>
<target/>enter Unicode values here</target>
</trans-unit>
```

As an example, the code for FULL_NAME field translation would be:

```
<trans-unit id="$
{adfBundle['oracle.adf.businesseditor.model.util.BaseRuntimeResourceBundle']
['persdef.sessiondef.oracle.iam.ui.runtime.form.model.user.entity.userEO.UD_IDF_R
ACF_ADV_CN__c_description']}">


<source>FULL_NAME</source>
<target>\u6C0F\u540D</target>
</trans-unit>
<trans-unit
id="sessiondef.oracle.iam.ui.runtime.form.model.RacfUserFormv1.entity.RacfUserFor
mv1EO.UD_IDF_RACF_ADV_CN__c_LABEL">
<source>FULL_NAME</source>
<target>\u6C0F\u540D</target>
</trans-unit>
```

f. Repeat Steps 6.c through 6.e for all attributes of the process form.

g. Save the file as BizEditorBundle_LANG_CODE.xlf. In this file name, replace *LANG_CODE* with the code of the language to which you are localizing. Sample file name: `BizEditorBundle_ja.xlf`.

7. Repackage the ZIP file and import it into MDS.

8. Log out of and log in to Oracle Identity Manager.

3.3.3 Clearing Content Related to Connector Resource Bundles from the Server Cache for Oracle Identity Manager Connector

When you deploy the connector, the resource bundles are copied from the resources directory on the installation media into Oracle Identity Manager database. Whenever you add a new resource bundle to the connectorResources directory or make a change in an existing resource bundle, you must clear content related to connector resource bundles from the server cache.

To clear content related to connector resource bundles from the server cache:

1. In a command window, switch to the *OIM_HOME*/server/bin directory.
2. Enter one of the following commands:

> **Note:**
>
> You can use the PurgeCache utility to purge the cache for any content category. Run `PurgeCache.bat` *`CATEGORY_NAME`* on Microsoft Windows or `PurgeCache.sh` *`CATEGORY_NAME`* on UNIX. The *`CATEGORY_NAME`* argument represents the name of the content category that must be purged.
>
> For example, the following commands purge Metadata entries from the server cache:
>
> ```
> PurgeCache.bat MetaData
> ```
>
> ```
> PurgeCache.sh MetaData
> ```

   - On Microsoft Windows: `PurgeCache.bat All`
   - On UNIX: `PurgeCache.sh All`

   When prompted, enter the user name and password of an account belonging to the SYSTEM ADMINISTRATORS group. In addition, you are prompted to enter the service URL in the following format:

   ```
   t3://OIM_HOST_NAME:OIM_PORT_NUMBER
   ```

   In this format:

   - Replace *`OIM_HOST_NAME`* with the host name or IP address of the Oracle Identity Manager host computer.
   - Replace *`OIM_PORT_NUMBER`* with the port on which Oracle Identity Manager is listening.

3.3.4 Enabling Logging for IBM RACF Advanced Connector

The IBM RACF Advanced connector supports two forms of logging, namely LDAP gateway-level logging and Oracle Identity Manager-level logging. This section discusses the following topics:

- Enabling Logging for the LDAP Gateway
- Event Logging in Oracle Identity Manager

3.3.4.1 Enabling Logging for the LDAP Gateway

LDAP Gateway logging operations are managed by the log4j2.properties file, which is located in the *LDAP_INSTALL_DIR*/conf/ directory. In the log4j2.properties file, edit the rootLogger log level:

```
rootLogger.level = INFO
```

The following is a list of log levels that can be used:

- ALL

  This level enables logging for all events.
- DEBUG

  This level enables logging of information about fine-grained events that are useful for debugging.
- INFO

  This level enables logging of messages that highlight the progress of the application at a coarse-grained level.
- WARN

  This level enables logging of information about potentially harmful situations.
- ERROR

  This level enables logging of information about error events that might allow the application to continue running.
- FATAL

  This level enables logging of information about very severe error events that could cause the application to stop functioning.
- OFF

  This level disables logging for all events.

Multiple log files are available for use with the connector. Table 3-2 lists the name, location, and contents of each LDAP gateway log file.

Table 3-2 Log Files and their Contents

Log File	Description
nohup.out	This log file contains the console window output from the LDAP Gateway. This file is primarily used in conjunction with the run.sh script (instead of the run.bat file) **Location:** .../`ldapgateway/bin/`
idfserver.log.0	This log file contains provisioning and reconciliation logging messages from the LDAP Gateway and is the primary log file used by the gateway component. **Location:** .../`ldapgateway/logs/`

3.3.4.2 Event Logging in Oracle Identity Manager

Oracle Identity Manager uses Oracle Java Diagnostic Logging (OJDL) for logging. OJDL is based on java.util.logger. This section contains the following topics:

Understanding the Log Levels

Configuring Logging in Oracle Identity Manager

3.3.4.2.1 Understanding the Log Levels

To specify the type of event for which you want logging to take place, you can set the log level to one of the following:

- ERROR:1
- WARNING:1
- NOTIFICATION:1
- TRACE:1
- TRACE:16
- TRACE:32

Oracle Identity Manager level logging operations are managed by the logging.xml file which is located in the following directory:

```
DOMAIN_NAME/config/fmwconfig/servers/SERVER_NAME/
```

Loggers are used to configure logging operations for the Oracle Identity Manager functions of the connector.

3.3.4.2.2 Configuring Logging in Oracle Identity Manager

OIM level logging operations are managed by the logging.xml file, which is located in following directory:

DOMAIN_NAME/config/fmwconfig/servers/*SERVER_NAME*/

Loggers are used to configure logging operations for the connector's OIM functions. To configure loggers:

1. In the text editor, open the *DOMAIN_NAME*/config/fmwconfig/servers/*SERVER_NAME*/logging.xml file.
2. Locate the logger you want to configure. If adding a logger for the first time, you must create the logger definition. Table 3-3 lists the Oracle Identity Manager loggers for this connector.

Table 3-3 Logger Parameters

Logger	Description
COM.IDENTITYFORGE.IDFUSEROPERATIONS	Logs events related to provisioning operations from Oracle Identity Manager to the LDAP gateway, such as user creation and modification events.
COM.IDENTITYFORGE.UTIL.RACF.IDFLDAPOPERATIONSIMPL	Logs events related to basic LDAP functions, including connecting to and disconnecting from the LDAP gateway.
COM.IDENTITYFORGE.RACF.TASKS.DELETERECONCILEOIMUSERSTASK	Logs events related to the RACF Delete OIM Users scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.FINDALLDATASETSTASK	Logs events related to the Find All Datasets scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.FINDALLGROUPSTASK	Logs events related to the Find All Groups scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.FINDALLSOURCESTASK	Logs events related to the Find All Sources scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.FINDALLSECURITYATTRIBUTESTASK	Logs events related to the RACF Find All Security Attributes scheduled task.

Table 3-3 (Cont.) Logger Parameters

Logger	Description
COM.IDENTITYFORGE.RACF.TASKS.RECONCILEALLLDAPUSERSTASK	Logs events related to the Reconcile All LDAP Users scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.RECONCILEALLUSERSTASK	Logs events related to the Reconcile All Users scheduled task
COM.IDENTITYFORGE.RACF.TASKS.RECONCILEDELETEDLDAPUSERSTASK	Logs events related to the RACF Reconcile Deleted LDAP Users scheduled task.
COM.IDENTITYFORGE.RACF.TASKS.RECONCILEUSERSTOINTERNALLDAPTASK	Logs events related to the RACF Reconcile Users to Internal LDAP scheduled task.

3. Define the <logger> element and its handlers. You can use the standard odl-handler as the log handler, or write your own.

   The following is an example of a logger definition for the Reconcile All Users scheduled task:

   ```
   <logger name="COM.IDENTITYFORGE.RACF.TASKS.RECONCILEALLUSERSTASK" level='TRACE:32'>
   <handler name='odl-handler'/>
   </logger>
   ```

4. Save the changes and close the file.
5. Restart the Oracle Identity Manager server for the changes to take effect.

Log statements will be written to the path that is defined in the log handler that you assigned in the logger definition. For example, in the above logger definition for the Reconcile All Users scheduled task (in step 3), the handler is odl-handler, which has the following default output file path:

```
${domain.home}/servers/${weblogic.Name}/logs/${weblogic.Name}-diagnostic.log'
```

4 Installing and Configuring the Agents of the IBM RACF Connector on the Mainframe

Install the Provisioning Agent - Pioneer and Reconciliation Agent - Voyager of the IBM RACF connector on the mainframe. These agents communicate with the LDAP Gateway during connector operations.

- Installation Requirements for Agents
- Installing the Mainframe Agents
- Configuring the Mainframe Agents
- Configuring Logging
- Customizing the Reconciliation Exit
- Activating and Deactivating Reconciliation Exits

4.1 Installation Requirements for Agents

These are the software and environmental setting requirements for installing the Provisioning Agent - Pioneer and Reconciliation Agent - Voyager.

Verifying Installation Requirement

Ensure that the mainframe system on which you intend to install Pioneer and Voyager meet the following requirements:

Table 4-1 Requirements

Item	Requirement
Operating System	IBM z/OS 2.5 and 3.1
Message Transport Layer	TCP/IP
IBM RACF Advanced Identity Repository	Verify that the current patch for z/OS is installed.
Target system user account for the Provisioning Agent - Pioneer and Reconciliation Agent - Voyager.	IBM RACF Advanced-authorized user account with System Administrators privileges.

Note:

Both the Voyager and Pioneer Agents must have IBM RACF ACIDs defined on the IBM RACF database. These ACIDs must have at least the permissions of the System Administrators group on the mainframe. These user accounts have permissions above those of ordinary administrators on the mainframe, which include Read, Write, Execute, and Modify privileges. Voyager and Pioneer use Language Environment. The following are the recommended Language Environment runtime options that avoid issues when installing Voyager and Pioneer:

- ALL31(ON)
- HEAP(32768,32768,ANYWHERE,KEEP,8192,4096)
- STACK(131072,131072,ANYWHERE,KEEP,524288,524288)

Environmental Settings and Other Requirements

Ensure that the following requirements are met on the mainframe:

- A subpool is created to contain reconciliation changes for Voyager to access and send to the LDAP gateway. The subpool is located in the ECSA Subpool 231. The Subpool size is governed by the parameter passed by Voyager using the parameter `SUBPOOL_SIZE=`. This parameter accepts values in the range of `0200K` to `7500K`. The number of messages stored is based on the amount of SUBPOOL allocated. For example, an allocation of `0200K` will yield 2048 – 100 byte messages,and an allocation of `7500K` will yield 76800 – 100 byte messages. It is generally a temporary staging area for reconciliation requests and if there is an outage, Voyager saves the subpool to a `//CACHESAV` disk file in the Voyager STC. When Voyager is restarted and the subpool is rebuilt, the `CACHESAV` file is reloaded into the subpool. Once the LDAP connects, the subpool data is sent to the LDAP.
- It is recommended that an automation product be used to intercept messages from Voyager to indicate that the LDAP is unavailable. This facilitates correct action to ensure a safe shutdown of Voyager.
- Both Pioneer and Voyager require LE runtime options to be set as listed below. IBM defaults are without an existing CEEPRM member in the 'SYS1.PARMLIB'.
  - **ALL31(ON)** - This is the default provided by IBM. ALL31 specifies whether an application can run entirely in AMODE 31 or the application has one or more AMODE 24 routines.
  - **ANYHEAP(16K,8K,ANYWHERE,FREE)** - This is the default provided by IBM.
  - **HEAP(32K,32K,ANYWHERE,KEEP,8K,4K)** - This is the default provided by IBM.
  - **STACK(128K,128K,ANYWHERE,KEEP,512K,128K)** - This is the default but we need more STACKStorage so we use STACK(128K,128K,ANYWHERE,KEEP,512K,512K).

  To temporarily override the LE runtime options only for Pioneer and Voyager STC (Started Task), add a //CEEOPTS DD * in both STCs and add the above options. These options are only in effect during the duration of the STC. After the STCs shutdown, the LE runtime options overridden go back to the normal IBM defaults.
- Verify if the permanent environment has the following options:

  ```
  All31(ON)
  ANYHEAP(16384,8192,ANYWHERE,FREE)
  HEAP(32768,32768,ANYWHERE,KEEP,8192,4096)
  STACK(131072,131072,ANYWHERE,KEEP,524288,524288)
  ```

If the permanent environment does not have the preceding options, you can temporarily override by using the following:

```
//PIONEER  PROC HLQ=++hlq++,STEPLIB=++linklib++,
//             REXXLIB=++rexxlib++
.....
//PIONEER  EXEC PGM=PIONEERX,PARM='/RPTSTG(ON),RPTOPTS(ON)',
//         REGION=0M,TIME=1440
.....
//CEEOPTS  DD  *
All31(ON)
ANYHEAP(16384,8192,ANYWHERE,FREE)
HEAP(32768,32768,ANYWHERE,KEEP,8192,4096)
STACK(131072,131072,ANYWHERE,KEEP,524288,524288)
```

- You can use a normal non-privileged RACF user ID for Pioneer and Voyager.
- A RACF group containing the SPECIAL attribute RACF userids is recommended.
- A RACF facility class is required for usage with the SPECIAL attribute.
- A RACF userid with SPECIAL must be created for Pioneer to issue commands to RACF.
- Pioneer plays the role of a central site administrator and performs all the related tasks. Voyager collects the RACF event data by using 3 modified RACF passive exits such as `ICHPWX01`, `ICHRIX02`, and `IRREVX01`. There are no changes in the RACF storage or system storage due to these exits.
- The exits `ICHPWX01` and `ICHRIX02` reside in a load library and must be in the Linked Pack Area (LPA). However, the `IRREVX01` exit must not reside in the LPA.

  The RACF command exit `IRREVX01` must not reside in the LPA as recommended by IBM. Otherwise, an `S0C4` abend will occur. The exits `ICHPWX01` and `ICHRIX02` both internal cache RACF IDs and passwords and the caching routine adds them to the subpool created by STARTUP. If the exits are enabled and the subpool is not available, then a message indicating that the subpool is not available is displayed and the messages are lost.

  The exit `IRREVX01` caches RACF commands and calls an external caching routine called `LOGCACHE`. You can dynamically activate `IRREVX01` using the z/OS command `T PROG=75` to activate the `IRREVX01` command exit. To deactivate or remove the `IRREVX01` exit, issue the z/OS command `T PROG=76`.

  Each cache message is 100 bytes long. When Voyager connects to the LDAP gateway, Voyager polls cache and reads all the messages up to a maximum size of the subpool. At this point the cache is empty. Once installed, you must IPL the z/OS to bring in the `ICHPWX01` and `ICHRIX02` exits.

  You can dynamically activate the `IRREVX01` exit by using the z/OS command `T PROG=75`. To deactivate or remove the `IRREVX01` exit, issue the z/OS command `T PROG=76`. Once activated, all RACF commands are cached.

  > **Note:**
  >
  > It is mandatory for Voyager to have RACF permissions to issue an `IRR.RADMIN.LISTUSER` call through the RACF API service. This call is issued through the `IRRSEQ00` module of IBM. This is the only required permission in addition to the PERMIT for the `CACHESAV` file that Voyager reads and writes while it is active.

4.2 Installing the Mainframe Agents

The IBM RACF Advanced connector is shipped with a pair of agents, one for the provisioning (Pioneer) and one for real-time reconciliation (Voyager). If real-time reconciliation is not required, then install and start only the provisioning agent.

> **Note:**
>
> If the mainframe agents are already installed, and you are planning to install a new version, then take a back-up of the following files:
>
> - `<HLQ>.pioneer.control.file`
> - `<HLQ>.voyager.control.file`
>
> When you install a new version, both the control files are set to default settings. You can restore the contents of the control files from the back-up copies.

1. On the computer hosting the mainframe, extract the contents of the `RACF-<TIMESTAMP>-<VERSION>.zip` file located in the connector installation media.

   The following XMIT files are extracted:

   - `CLISTLIB.XMIT`
   - `JCLLIB.XMIT`
   - `LINKLIB.XMIT`
   - `PARMLIB.XMIT`
   - `PROCLIB.XMIT`

2. Transmit the extracted XMIT files to z/OS by using the following specifications:

   - `RECFM=FB`
   - `LRECL=80`
   - `BLKSIZE=3120`
   - `DSORG=PS`

   For example, you can use 3270 or FTP to transfer the files.

   The following datasets will exist on z/OS:

   - `<`*`HLQ`*`>.CLISTLIB.XMIT`
   - `<`*`HLQ`*`>.JCLLIB.XMIT`
   - `<`*`HLQ`*`>.LINKLIB.XMIT`
   - `<`*`HLQ`*`>.PARMLIB.XMIT`
   - `<`*`HLQ`*`>.PROCLIB.XMIT`

   > **Note:**
   >
   > In the preceding list, *<HLQ>* is the high-level-qualifier used when transmitting the files to z/OS.

3. For each of the XMIT files that have been transmitted, execute the following command at the TSO prompt: `TSO RECEIVE INDA('<HLQ>.<FILE>.XMIT')` .

   When prompted to specify restore parameters, enter `DA('<HLQ>.<FILE>')` .

   For example, if the high-level qualifier is IDF and the file is `CLISTLIB.XMIT`, then execute the following command:

   `TSO RECEIVE INDA('IDF.CLISTLIB.XMIT')`
   When prompted, respond with: `DA('IDF.CLISTLIB')`

   The following datasets will exist on z/OS:

   - `<HLQ>.CLISTLIB`
   - `<HLQ>.JCLLIB`
   - `<HLQ>.LINKLIB`
   - `<HLQ>.PARMLIB`
   - `<HLQ>.PROCLIB`

   > **Note:**
   >
   > In the preceding list, `<HLQ>` is the high-level-qualifier used when receiving the previously transmitted files.

4. Edit each of the following installed job streams to replace any placeholders in them with actual values.

   - `<HLQ>.CLISTLIB.ENVINFO`
   - `<HLQ>.JCLLIB.CREATDSN`
   - `<HLQ>.JCLLIB.IEBCOPYL`
   - `<HLQ>.JCLLIB.IEBCOPYP`
   - `<HLQ>.JCLLIB.IEBCPYPR`
   - `<HLQ>.JCLLIB.KEYMODR`
   - `<HLQ>.PARMLIB.PROGID`

     > **Note:**
     >
     > In the preceding job stream, update the ++vol++ placeholder with the VOLUME from where you have received LINKLIB.

   - `<HLQ>.PROCLIB.PIONEER`
   - `<HLQ>.PROCLIB.STARTUP`
   - `<HLQ>.PROCLIB.VOYAGER`
   - `<HLQ>.PROCLIB.WRAPUP`
   - `<HLQ>.JCLLIB.IEBCPYCL`
   - `<HLQ>.JCLLIB.LOADDSN`
   - `<HLQ>.JCLLIB.RACFDEF`

- <*HLQ*>.JCLLIB.RACFDEL
- <*HLQ*>.JCLLIB.RACFRC
- <*HLQ*>.JCLLIB.RACFRCOG
- <*HLQ*>.JCLLIB.RACFRCOU
- <*HLQ*>.JCLLIB.SECUTLD
- <*HLQ*>.JCLLIB.SECUTLE
- <*HLQ*>.JCLLIB.PROG75
- <*HLQ*>.JCLLIB.ASMJCL

> **Note:**
>
> In the preceding list, <*HLQ*> is the high-level-qualifier used when receiving the previously transmitted files.

The following table lists the installation placeholders found in job streams, their description, and example.

Table 4-2 Installation Placeholders

Placeholder	Description	Example
++hlq++	The high-level qualifier where the mainframe agent is to be installed. You must include all the multiple segments, if any.	IDF.PROD
++hlq1++	The top-most segment of the high-level qualifier where the mainframe agent is to be installed	IDF
++vol++	The volume where the mainframe agent is to be installed.	SDWRK1
++lpalib++	The DSN of the data set that contains customized lpalibs. Customize based on the z/OS environment.	USER.LPALIB
++parmdtr++	The name of the PARMLIB XMIT that was trasmitted to z/OS (without the .XMIT).	<*HLQ*>.PARMLIB
++parmlib++	The DSN of the data set that contains customized parmlibs. Customize based on z/OS environment.	USER.PARMLIB
++procdtr++	The name of the PROCLIB XMIT that was transmitted to z/OS (without the .XMIT).	<HLQ>.PROCLIB
++proclib++	The DSN of the data set that contains customized proclibs. Customize based on z/OS environment.	USER.PROCLIB

Table 4-2 (Cont.) Installation Placeholders

Placeholder	Description	Example
`++linkdtr++`	The name of the `LINKLIB` XMIT that was trasmitted to z/OS (without the `.XMIT`).	`<`*`HLQ`*`>.LINKLIB`
`++linklib++`	The DSN where the `LINKLIB` XMIT that was received.	`<`*`HLQ`*`>.LINKLIB`
`++rexxdtr++`	The name of the `CLISTLIB` XMIT that was transmitted to z/OS (without the `.XMIT`).	`<`*`HLQ`*`>.CLISTLIB`
`++rexxlib++`	The DSN where the `CLISTLIB` XMIT that was received.	`<`*`HLQ`*`>.CLISTLIB`
`++pionprms++`	The DSN of the control (configuration) file for the provisioning agent.	`PIONEER.CONTROL.FILE`
`++voyprms++`	The DSN of the control (configuration) file for the reconciliation agent.	`VOYAGER.CONTROL.FILE`
`++pionlog++`	The DSN of the control log (configuration) file for the LOGGERX feature of provisioning agent.	`PIONEER.CONTROL.LOG`
`++voyglog++`	The DSN of the control log (configuration) file for the LOGGERX feature of reconciliation agent.	`VOYAGER.CONTROL.LOG`
`++pstcuserid++`	The ACID of the user to be created for running the provisioning agent STC.	`PIONEER`
`++vstcuserid++`	The ACID of the user to be created for running the reconciliation agent STC.	`VOYAGER`
`++pstcnm++`	The name / description for the provisioning agent STC.	`'PIONEER STARTED TASK'`
`++vstcnm++`	The name / description for the reconciliation agent STC.	`'VOYAGER STARTED TASK'`
`++pstcuid++`	The OMVS UID assigned to the provisioning agent STC. Customize based on z/OS environment.	`80`
`++vstcuid++`	The OMVS UID assigned to the reconciliation agent STC. Customize based on z/OS environment.	`90`
`++stcgrp++`	The group assigned to the provisioning and reconciliation agent STCs. Ensure the group has UID(0) or BPX.SUPERUSER assigned. Customize based on z/OS environment.	`OMVSGRP`
`++secgrp++`	The Secure ID user default group.	`IDFSGRP`
`++secuid++`	The Secure ID user ACID.	`IDFAGNT`

Table 4-2 (Cont.) Installation Placeholders

Placeholder	Description	Example
`++secidnm++`	The Secure ID name.	`SECURE_ID`
`++cailink++`	The CA Linklist Library DSN. Customize based on RACF environment.	`CAI.CAKOLINK`

For example, in the following snippet from `CREATEDSN`, replace the placeholders `++hlq++` and `++vol++` with values such as `IDF.PROD` and `SDWRK1`:

```
//*
//S1       SET  PHLQ=++hlq++.PIONEER
//S2       SET  VHLQ=++hlq++.VOYAGER
//S3       SET  PVOL=++vol++
//S4       SET  VVOL=++vol++
//*
```

The following snippet displays the placeholders replaced with values:

```
//*
//S1       SET  PHLQ=IDF.PROD.PIONEER
//S2       SET  VHLQ=IDF.PROD.VOYAGER
//S3       SET  PVOL=SDWRK1
//S4       SET  VVOL=SDWRK1
//S5       SET THLQ=IDF.PROD
//*
```

5. Execute each of the following job streams in the order as shown in the following table to complete installation.

Table 4-3 Job Streams to Execute

Job Stream	Description
`<HLQ>.JCLLIB.IEBCOPYP`	Copies PARMLIB members to user PARMLIB.
`<HLQ>.JCLLIB.IEBCPYPR`	Copies PROCLIB members to user PROCLIB.
`<HLQ>.JCLLIB.IEBCPYCL`	Copies Rexx execs to user Rexx library.
`<HLQ>.JCLLIB.IEBCOPYL`	Copies exit routines to use LPA library.
`<HLQ>.JCLLIB.CREATDSN`	Allocates run time data sets, deleting the data sets first if they already exist.
`<HLQ>.JCLLIB.LOADDSN`	Copies PIONEER & VOYAGER configuration (control) files.
`<HLQ>.JCLLIB.RACFDEL`	Deletes pre-existing user accounts and privileges on the user accounts required to execute agent STCs.
`<HLQ>.JCLLIB.RACFDEF`	Defines users and permissions required to run the mainframe agent STCs.

The installation of the provisioning and reconciliation agents, Pioneer and Voyager, is complete. At this point, you can optionally remove the XMIT datasets that were originally transmitted to z/OS.

4.3 Configuring the Mainframe Agents

After installing Pioneer and Voyager, you must configure the mainframe agents to receive requests from and send responses to the LDAP gateway.

This section discusses the following topics:

- Configuring the Provisioning Agent
- Configuring the Reconciliation Agent

4.3.1 Configuring the Provisioning Agent

You must configure the provisioning agent to receive requests from the LDAP gateway, which originates from Oracle Identity Manager.

Edit the `<HLQ>.PIONEER.CONTROL.FILE` file to configure the behavior of the provisioning agent. Here, *`<HLQ>`* is the high-level-qualifier that you specified while installing the agents.

Table 4-4 Parameters of the Pioneer Control File

Parameter	Value	Description
TCPN	`TCPIP`	The name of the TCP/IP STC where the agent is executing.
IPAD	`0.0.0.0`	Do not change.
PORT	`9999`	The TCP/IP port that the agent will listen on.
CRLF	`Y` or `N`	If this flag is set to `Y`, then mainframe sends a response with carriage line feed. You must set the value of this parameter to `Y` for version 6+ of the LDAP Gateway. Set to `N` for version 5.
ESIZE	`16`	This is the only valid value. This parameter is for the AES128 encryption and decryption.

Table 4-4   (Cont.) Parameters of the Pioneer Control File

Parameter	Value	Description
POST_PROC_ALIAS	`T` or `F`	If you set the value of this parameter to `T`, then all LDAP **Alias** requests are processed using the IDCAMS JCL (from INJCLR DD) submission to the internal reader. If you set it to `F`, then all LDAP **Alias** requests are processed internally by the Pioneer Agent. **Note:** With MF Agent version 6.10.0 onwards, the default value for this parameter will be set to `F`, rather than `T`. The **Alias** process will continue to wor

Table 4-4 (Cont.) Parameters of the Pioneer Control File

Parameter	Value	Description
		k as before. If you set the value of this parameter to `T`, then **Alias** processing will be performed using IDCAMS JCL present in the dataset referenced by Pioneer DD INJCLR.
RWAIT	`0` or `999` (in seconds)	Enter the number of seconds the agent must wait before executing the jobs submitted by the batch recon.

Table 4-4 (Cont.) Parameters of the Pioneer Control File

Parameter	Value	Description
JWAIT	`0` or `999` (in seconds)	Controls how long the agent waits for the execution of the IDCAMS job (submitted for **Alias**.processing).
QUEUE_DSN	`IDF.SEARCH`	Max 44 character DSN used with RWAIT for recons. This DSN does not need allocated or deleted.
EXPORT_MON	`NO` or `YES, REC=nnnnn`	If you do not want to monitor user or group imports, then set the value of this parameter to `NO`. Set the value of this parameter to `YES` in the following format if you want to monitor user or group imports displaying a message every *nnnnn* records: `YES,REC=`*`nnnnn`* In this format, ensure that you replace *nnnnn* with a 5-digit format. For example, to monitor progress for every 5000 records, set the value of this parameter to `YES,REC=05000`
IP	`V4` or `V6`	IP version to be used for communication between LDAP gateway and PIONEER agent. Value `V4` would be used as an IPv4 based IP address or hostname for communication. (e.g. In tops.properties value for *host*=192.168.100.0) Value `V6` would be used as an IPv6 based IP address or hostname for communication. (e.g. In tops.properties value for *host*=FE80::A0:A001:A0:A0A0%tap0) Default value is `V4`, when not specified in `<HLQ>.PIONEER.CONTROL.FILE`.
DEBUG	`Y` or `N`	This parameter is deprecated.
IDLEMSG	`Y` or `N`	This parameter is deprecated.
DEBUGOUT	`SYSOUT, CLASS(X)`	This parameter is deprecated.
SPIN_CLASS	`X`	This parameter is deprecated.
AUDIT_LOG	`YES` or `NO`	This parameter is deprecated.

4.3.2 Configuring the Reconciliation Agent

You must configure the reconciliation agent to send incremental responses to the LDAP gateway.

Edit the `<HLQ>.VOYAGER.CONTROL.FILE` file to configure the behavior of the reconciliation agent. `<HLQ>` is the high-level qualifier that you specified while installing the agents.

Table 4-5 Parameters of the Voyager Control File

Parameter	Value	Description
TCPN	`TCPIP`	The name of the TCP/IP STC where the agent is executing.
IPAD	`999.999.999.999` or `ldap.example.com`	LDAP destination IP address or hostname (up to 40 characters).
PORT	`9999`	LDAP destination port that is listening to the incoming agent messages.
CRLF	`Y` or `N`	If this flag is set to `Y`, then mainframe sends a response with carriage line feed. You must set the value of this parameter to `Y` for version 6+ of the LDAP Gateway. Set to `N` for version 5.
ESIZE	`16`	This is the only valid value. This parameter is for the AES128 encryption and decryption.
CACHE_DELAY	`0` to `999`	This is the number of seconds that Voyager waits before issuing a write socket to the LDAP Gateway.
VOYAGER_ID	`VOYAGER`	This value will be included in the LDAP logs for diagnostic
CONNECT_RETRY	`999`	The number of times to retry when the LDAP connection is down.
CONNECT_INTV	`10`	The number of seconds between retries when the LDAP connection is down.

Table 4-5 (Cont.) Parameters of the Voyager Control File

Parameter	Value	Description
IP	V4 or V6	IP version to be used for communication between LDAP gateway and VOYAGER agent. Value V4 would be used as an IPv4 based IP address or hostname for communication. (e.g. IPAD entry in VOYAGER.CONTROL.FILE = 192.168.100.10) Value V6 would be used as an IPv6 based IP address or hostname for communication. (e.g. IPAD entry in VOYAGER.CONTROL.FILE = fe80::74c3:eeff:fe1e:60fd)
PIONEER_DELETE_MSGS	Not applicable	The parameter is deprecated.
RECOVERY_INTERVAL	Not applicable	The parameter is deprecated.
DNS_RECOVERY_INTERVAL	Not applicable	The parameter is deprecated.
DEBUG	Y or N	This parameter is deprecated.
DEBUGOUT	SYSOUT, CLASS (X)	This parameter is deprecated.
CONNECT_MSGS	Y or N	This parameter is deprecated.
MSGID01	NO or YES,IDMV602E,X	This parameter is deprecated.

4.4 Configuring Logging

You can configure logging for both Pioneer and Voyager by editing the <HLQ>.PIONEER.CONTROL.LOG and <HLQ>.VOYAGER.CONTROL.LOG files, respectively, and setting values for various log parameters based on your requirement. For example, you can have complete control over the messages that you want to print or suppress and also the device over which the message must be printed. A separate control file is designed and used to control the functionality of logging through LOGGERX.

Logging Parameters

LOGGERX requires initial parameters setup for operating. This is achieved by using a control file (different from the control file for Pioneer). The parameters of this control file described in the following table.

Table 4-6 Logging Parameters

Parameter	Accepted Value	Description
LOGGERX_MSGID01	NO or YES,IDMV602E,X	If you want to suppress the IDMV602E recovery message, then set the value of this parameter to NO. If you want to display the IDMV602E recovery message, then set the value of the parameter to YES in the following format: YES,IDMV602E,*X* In this format, replace *X* with any number between 0 through 99, which specifies the number of times the recovery message IDMV602E must be displayed. For example, YES,IDMV602E,6. **Note:** This parameter is applicable only to the <HLQ>.VOYAGER.CONTROL.LOG file.
LOGGERX_SYSOUT_CLASS	A through Z	The value in this parameter determines the class where the SYSOUT messages must be rolled to. For example, if you set the value of this parameter to A, then all SYSOUT messages will be directed to class A. If you do not specify a value for this parameter, then by default, all SYSOUT messages are rolled to class A.
LOGGERX_LEVEL_ROUTING	*MSG_TYPE:DEVICE* In this format, replace: • *MSG_TYPE* with types of messages such as INFO, WARN, ERR, or DBG. • *DEVICE* with any combination of SYSOUT, CONSOLE, FILE, or NONE by using a vertical bar (|) as the delimiter.	This parameter controls the message logging based on message type. The value of this parameter must contain the message type and the devices on which it is to be printed. For example, if you set the value of this parameter to INFO:SYSOUT|CONSOLE, then it means that all Informational messages will be written on to SPOOL/SYSOUT and the mainframe operator console. The same is applicable for message types – WARN(Warning), EROR(Error) and DEBG(DEBUGOUT).

Table 4-6 (Cont.) Logging Parameters

Parameter	Accepted Value	Description
LOGGERX_XXXX where XXXX can be either INFO, WARN, EROR, DEBG, AUDT, or PARM	`SYSOUT`	Use this parameter to specify SYSOUT when the value of *DEVICE* in the LOGGERX_LEVEL_ROUTING parameter is `FILE`. When the value is passed as SYSOUT, the file is created in the SPOOL as part of job output. For example, consider that the value of the LOGGERX_LEVEL_ROUTING parameter is set to `WARN:FILE`. In such a case, the entry `LOGFILE_WARN=SYSOUT` means that the job output will contain a file by the name WARNOUT that will contain warning messages.
LOGGERX_MSG_ROUTING	*MSGID:DEVICE* In this format, replace: • *MSGID* with the message ID corresponding to a message text. • *DEVICE* with any combination of `SYSOUT`, `CONSOLE`, `FILE`, or `NONE` by using the vertical bar (`|`) as the delimiter.	Use this parameter to redirect messages to a different device or suppress individual message based on message IDs. This parameter overrides the message levels set in the LOGGERX_LEVEL_ROUTING parameter. For example, the entries LOGGERX_MSG_ROUTING=IDFRPI001:NONE and LOGGERX_MSG_ROUTING=IDFRPI002:FILE combined with LOGGERX_LEVEL_ROUTING=INFO:CONSOLE mean that all Informational messages will go out on CONSOLE except, IDFRPI001(suppressed) and IDFRPI002(written on a file). You can provide 999 message IDs for each agent. In other words, you can choose to override, suppress, or redirect any number of messages. For a comprehensive list of message IDs and the corresponding message text, see Pioneer Messages and Voyager Messages.

Table 4-6 (Cont.) Logging Parameters

Parameter	Accepted Value	Description
LOGGERX_FILE_MSG	SYSOUT	This parameter is used when FILE is specified as the Device type in the LOGGERX_MSG_ROUTING parameter to route all message ID- specific messages to MSGOUT in the spool. This parameter accepts a value of SYSOUT. When the value is passed as SYSOUT, the file is (MSGOUT) created in the SPOOL as part of job output. For example, the entry LOGFILE_MSG=SYSOUT means that the job output will contain a file by the name MSGOUT that contains messages corresponding to the message ID provided in the value of the LOGGERX_MSG_ROUTING parameter with the destination device as FILE.
LOGGERX_DEBUG	Y or N	This parameter is deprecated in v6.0.0 and later versions of the Mainframe agents.
LOGGERX_SPIN_CLASS	X	This parameter is deprecated in v6.0.0 and later versions of the Mainframe agents.
LOGGERX_AUDIT_LOG	YES or NO	This parameter is deprecated in v6.0.0 and later versions of the Mainframe agents.
LOGGERX_CONNECT_MSGS	Y or N	This parameter is deprecated in v6.0.0 and later versions of the Mainframe agents.

Important Use Case of the Log File

1. LOGGERX_LEVEL_ROUTING=INFO:FILE
   - LOGGERX_LEVEL_ROUTING=AUDT:FILE
   - LOGGERX_LEVEL_ROUTING=WARN:FILE
   - LOGGERX_LEVEL_ROUTING=ERR:FILE
   - LOGGERX_LEVEL_ROUTING=DBG:FILE
   - LOGGGERX_FILE_WARN=SYSOUT
   - LOGGGERX_FILE_INFO=SYSOUT
   - LOGGGERX_FILE_AUDT=SYSOUT
   - LOGGGERX_FILE_DEBG=SYSOUT
   - LOGGGERX_FILE_EROR=SYSOUT

The above combination results in all INFO, AUDT, WARN, ERR, and DBG messages written onto INFOUT, WARNOUT, ERROUT, and DBGOUT in spool/Sysout.

2. LOGGERX_LEVEL_ROUTING=INFO:FILE|SYSOUT
   - LOGGERX_LEVEL_ROUTING=AUDT:FILE|SYSOUT
   - LOGGERX_LEVEL_ROUTING=WARN:FILE|SYSOUT
   - LOGGERX_LEVEL_ROUTING=ERR:FILE|SYSOUT
   - LOGGERX_LEVEL_ROUTING=DBG:FILE|SYSOUT
   - LOGGGERX_FILE_WARN=SYSOUT
   - LOGGGERX_FILE_INFO= SYSOUT
   - LOGGGERX_FILE_AUDT=SYSOUT
   - LOGGGERX_FILE_DEBG= SYSOUT
   - LOGGGERX_FILE_EROR= SYSOUT

   The above combination results in all INFO, AUDT, WARN, ERR, DBG messages written onto INFOUT, WARNOUT, ERROUT, and DBGOUT in spool and all the messages will also be written onto SYSOUT file in job output.

3. LOGGERX_LEVEL_ROUTING=INFO:FILE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=AUDT:FILE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=WARN:FILE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=ERR:FILE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=DBG:FILE|SYSOUT|CONSOLE
   - LOGGGERX_FILE_WARN=SYSOUT
   - LOGGGERX_FILE_INFO= SYSOUT
   - LOGGGERX_FILE_AUDT= SYSOUT
   - LOGGGERX_FILE_DEBG= SYSOUT
   - LOGGGERX_FILE_EROR= SYSOUT

   The above combination results in all INFO, AUDT, WARN, ERR, and DBG messages written onto INFOUT, WARNOUT, ERROUT, and DBGOUT in spool and all the messages will also be written onto SYSOUT file in job output and on the mainframe operator console.

4. LOGGERX_LEVEL_ROUTING=INFO:NONE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=AUDT:NONE
   - LOGGERX_LEVEL_ROUTING=WARN:NONE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=ERR:NONE|SYSOUT|CONSOLE
   - LOGGERX_LEVEL_ROUTING=DBG:NONE|SYSOUT|CONSOLE
   - LOGGGERX_FILE_WARN=SYSOUT
   - LOGGGERX_FILE_INFO= SYSOUT
   - LOGGGERX_FILE_AUDT=SYSOUT
   - LOGGGERX_FILE_DEBG= SYSOUT
   - LOGGGERX_FILE_EROR= SYSOUT
   - LOGGERX_MSG_ROUTING=IDMP000I :CONSOLE

- LOGGERX_MSG_ROUTING=IDMP010I :CONSOLE
- LOGGERX_MSG_ROUTING=IDMP300I :CONSOLE
- LOGGERX_MSG_ROUTING=IDMP001E:CONSOLE

The above combinations results in all INFO, AUDT, WARN, ERR, and DBG messages being suppressed. Since NONE is specified it does not matter if other devices are specified too, the messages will be suppressed. However, as LOGGERX_MSG_ROUTING is also specified, the messages IDs IDMP000I, IDMP010I, IDMP300I, and IDMP001E are not suppressed and are displayed on the CONSOLE. This establishes that at any point of time, the LOGGERX_MSG_ROUTING parameter has a higher priority in deciding the message's output device, than its corresponding LEVEL ROUTING

> **Note:**
>
> In the sample control log files, for **Parm** message output, logging is routed based on message IDs IDMP400I, IDMP401E, and IDMV400I. These are set to route to 'SYSOUT' device and needs to maintain to get the PARMOUT dataset created in SPOOL.

4.5 Customizing the Reconciliation Exit

You can customize the default IRREVX01 exit to meet any special requirements in your environment.

> **Note:**
>
> Perform the procedure described in this section only if you use a custom RACF command exit (IRREVX01).

The IRREVX01 exit is invoked each time a RACF command is executed and the specific data from the command is written to subpool 231. This data is used during reconciliation. This connector lets you use the default reconciliation exit (IRREVX01) that is shipped with the connector installation package as well as any custom command exit that you may have created in your environment.

This connector is shipped with two sample modules namely IRREVX1I and CUSTINSX that are available in the `<HLQ>.JCLLIB` library. The CUSTINSX module is a sample source that also includes Write-to-operator (WTO) statements to test the execution flow. Include any site-specific changes or customization in the CUSTINSX module. The IRREVX1I source module is a sample driver module source that includes a call to `IDFINSTX`, which is IDMWORKS's modified logic and a commented call to the CUSTINSX module. Use the sample IRREVX1I module to call the CUSTINSX module that includes your custom logic. Then, activate the IRREVX01 exit to add the IRREVX1I load module dynamically to the IRREVX01 exit point.

To call a custom exit module:

1. Open the IRREVX1I member module, located in the `<HLQ>.JCLLIB` library, for editing.
2. Delete the * symbol from column 1 to uncomment the statements to call the CUSTINSX module as shown in the code snippet below:

```
*----------------------------------------------------------------------*
UNCOMMENT BELOW COMMENTED BLOCK TO CALL CUSTOMER MODIFIED IRREVX01
```

```
REFER TO PDF ADMIN GUIDE (Documentation) BEFORE MAKING CHANGE
*-----------------------------------------------------------------------*
L R15,=V(CUSTINSX) LOAD CUSTOMER IRREVX01 ENTRY POINT ADDR
BASR R14,R15 INVOKE CUSTOMER IRREVX01
*-----------------------------------------------------------------------*
UNCOMMENT ABOVE COMMENTED BLOCK TO CALL CUSTOMER MODIFIED IRREVX01
*-----------------------------------------------------------------------*
```

3. Save the file.
4. Refer to the sample CUSTINSX module source supplied in the `<HLQ>.JCLLIB` library, and then add your custom code within the same CUSTINSX module. Ensure to remove the default WTO statements (included to track the execution flow) in the CUSTINSX module to avoid flooding the system log with messages.
5. Assemble and link-edit the CUSTINSX module and the IRREVX1I driver-source module by using the ASMJCL job that is supplied in the `<HLQ>.JCLLIB` library. Change placeholders such as `++linklib++` and `++hlq++` with the site-specific load library and source dataset high-level qualifier, respectively and then submit the job. Ensure that the job completes with MAX-CC of 0 or 4.
6. Check for and delete any dynamic module (such as IRREVX1I or LOGEVX01) that might already exist at the IRREVX01 exit point as follows:

   > **Note:**
   >
   > From LDAP Gateway version 6.3.1 onward, the name of the exit module provided by IDMWORKS has been changed from LOGEVX01 to IRREVX01. If you have upgraded to LDAP Gateway version 6.3.1 or later versions, then you must ensure to delete the LOGEVX01 module from the dynamic exit point. The procedure to delete the LOGEVX01 module is described later in this topic.

   a. Run the following command to check whether the IRREVX1I or LOGEVX01 module already exists at the IRREVX01 exit point:

   ```
   /D PROG,EXIT,EXITNAME=IRREVX01
   ```

   b. If the IRREVX1I module exists, then delete it by using the PROG76 PARMLIB member. Run the following command from the z/OS operator interface:

   ```
   /T PROG=76
   ```

   c. If the LOGEVX01 module exists, then delete it by updating the PROG76 parmlib member to specify the module as LOGEVX01 as shown below:

   ```
   EXIT,DELETE,EXITNAME=IRREVX01,MODNAME=LOGEVX01
   ```

   Then, issue the `/T PROG=76` operator command.
7. Add the updated IRREVX1I load module (assembled and integrated with the CUSTINSX module) dynamically to the IRREVX01 exit point by using the PROG75 PARMLIB member. To do so, you must activate the IRREVX01 exit as described in Activating and Deactivating Reconciliation Exits.

4.6 Activating and Deactivating Reconciliation Exits

To make use of real-time reconciliation and the reconciliation agent, you must activate system exits for capturing and reacting to changes in the target system.

Activate the system exits to capture target system changes in real-time. To do so, run the following command from the z/OS operator interface:

```
T prog=75
```

When you run this command, the IRREV1I load module that is supplied with the connector is dynamically added to the IRREVX01 exit point using the PROG75 PARMLIB member that supplied with the `<HLQ>.PARMLIB` library.

Deactivate the system exits to disable the reconciliation of real-time changes to the target system. To do so, run the following command from the z/OS operator interface:

```
T prog=76
```

4.7 Operator Interface for Mainframe Agents

Both provisioning and reconciliation agents have an operator interface, and you can control the agents by passing commands through the interface.

The following topics are discussed in this section:

- Provisioning Agent Commands
- Reconciliation Agent Commands

4.7.1 Provisioning Agent Commands

Pass the Pioneer provisioning agent commands through the operator interface to control Pioneer.

Table 4-7 Provisioning Agent Commands

Command	Description
`T PROG=ID`	APF authorizes `<HLQ>.LINKLIB` - required to start the agent.
`S PIONEER`	Starts the agent.
`F PIONEER,SHUTDOWN`	Shuts down the agent.
`F PIONEER,STATUS`	Sends a status request to the agent.
`F PIONEER,DEBUG=Y`	Enables debug-level (detailed) log output.
`F PIONEER,DEBUG=N`	Disables debug-level (detailed) log output.

Note:

This interface through the z/OS modify command is a *single-threaded* system. Commands are queued and may take a few seconds before the agent acknowledges them.

4.7.2 Reconciliation Agent Commands

Pass the Voyager reconciliation agent through the operator interface to control Voyager.

Table 4-8 Reconciliation Agent Commands

Command	Description
`T PROG=ID`	APF authorizes `<HLQ>.LINKLIB` - *required to start the agent*.
`T PROG=75`	Activates system exits - required for real-time reconciliation as described in Activating and Deactivating Reconciliation Exits.
`S VOYAGER`	Starts the agent.
`F VOYAGER,SHUTDOWN`	Shuts down the agent.
`F VOYAGER,STATUS`	Sends a status request to the agent.
`F VOYAGER,DEBUG=Y`	Enables debug-level (detailed) log output.
`F VOYAGER,DEBUG=N`	Disables debug-level (detailed) log output.
`F VOYAGER,IPAD=999.999.999.999,PORT=9999`	Changes the IP address and port of the target LDAP Gateway.

Note:

The interface through the z/OS modify command is a *single-threaded* system. Commands are queued and take a few seconds before the agent acknowledges them.

5 Using the IBM RACF Advanced Connector

You can use the IBM RACF Advanced connector for performing reconciliation and provisioning operations after configuring it to meet your requirements.

The procedure to use the IBM RACF Advanced connector can be divided into the following topics:

5.1 Guidelines on Using the IBM RACF Advanced Connector

Apply the following guidelines while using the connector:

- The LDAP Gateway does not send the full attribute value when provisioning attribute values that contain one or more space characters. If this problem occurs, surround the attribute value in single quotation marks when populating the form field.
- The RACF connector LDAP gateway encrypts ASCII data transmitting the encrypted message to the mainframe. The mainframe decrypts this message, as the in bound message is in ASCII format, it is translated to EBCDIC for mainframe processing. As a result, any task that requires non-ASCII data transfer fails. In addition, there is no provision in the connector to indicate that the task has failed or that an error has occurred on the mainframe. To avoid errors of this type, you must exercise caution when providing inputs to the connector for the target system, especially when using a regional language interface.
- Passwords used on the mainframe must conform to stringent rules related to passwords on mainframes. These passwords are also subject to restrictions imposed by corporate policies and rules about mainframe passwords. Keep in mind these requirements when you create or modify target system accounts through provisioning operations on Oracle Identity Manager.
- The subpool must be started before starting the Reconciliation Agent. If the agent is started before the subpool, then an error message stating, "NO TOKEN FOUND", will be printed. Additionally, if the LDAP Gateway is not available when the Reconciliation Agent is started, then an error message is generated stating, "NO LDAP FOUND" will be printed.
- When you update the `TSO_SIZE` and `TSO_MAXSIZE` attributes during a provisioning operation, you must not include leading zeros in the value that you specify. For example, if you want to change the value of the `SIZE` attribute from `000001` to `000002`, then enter `2` in the `SIZE` field on the Identity Self Service.

5.2 Scheduled Tasks for Lookup Field Synchronization

The scheduled tasks for lookup field synchronization populate lookup tables with facility, dataset, group, or profiles IDs that can be assigned during the user provisioning process.

The following are the scheduled tasks for lookup field synchronization:

- RACF Find All Resources
- RACF Find All Datasets
- RACF Find All Groups

These scheduled tasks populate lookup fields in Oracle Identity Manager with resource profiles, datasets, or group IDs. Values from these lookup fields can be assigned during user provisioning operations and reconciliation runs. When you configure these scheduled tasks, they run at specified intervals and fetch a listing of all resource, dataset, or group IDs on the target system for reconciliation.

Table 5-1 describes the attributes of the 3 scheduled tasks.

Table 5-1 Attributes of the Find All Resources, Find All Datasets, and Find All Groups Scheduled Tasks

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RACFResource`
Resource Object	Enter the name of the resource object against which provisioning runs must be performed. Sample value: `OIMRacfResourceObject`
Lookup Code Name	Enter the name of the lookup code where OIM will store the results of the scheduled task. Sample value 1: `Lookup.profileNames` Sample value 2: `Lookup.ResourceNames`

Table 5-1 (Cont.) Attributes of the Find All Resources, Find All Datasets, and Find All Groups Scheduled Tasks

Attribute	Description
Recon Type	This attribute determines how resources, datasets, or group memberships from the target system are populated in Oracle Identity Manager lookup definitions. You can use one of the following options: • `Append` adds resources, datasets, or group membership entries from the target system that do not exist in the Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definitions. Any existing entries remain untouched. • `Replace` removes all the existing entries in Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definition and replaces them with resources, datasets, or group membership entries from the target system. • `Merge` handles entries in the following manner: – If you are using the connector for a single installation of the target system, then resources, datasets, and group membership entries that exist in both the target system and Oracle Identity Manager are updated in the Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definitions. Resources, datasets, and group membership entries that exist only in the target system are added to the Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definitions. – If you are using the connector for multiple installations of the target system, then only the resources, datasets, and group membership entries corresponding to the target system installation that you are using are updated or added. Entries that exist in both the target system and Oracle Identity Manager are updated in the Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definitions. Entries that exist only in the target system are added to the Lookup.ResourceNames, Lookup.DatasetNames, or Lookup.GroupNames lookup definitions. Default value: `Merge`
Resource Class Type **Note:** This attribute is available only in the RACF Find All Resources scheduled task.	Enter the name of the type of resource class you are reconciling. You can enter multiple resource class types as a comma-separated list. Sample value: `FACILITY,CDT,OPERCMDS` **Note:** Ensure that the resources list that you specify here matches the list that you have specified for the `supportedResourceClasses` property in the racf.properties file.

5.2.1 RACF Reconcile Groups To Internal LDAP

The RACF Reconcile Groups to Internal LDAP scheduled task is used to process the Group file extract from the target system to the internal LDAP store. When you configure this scheduled task, it runs at specified intervals and fetches a list of groups on the target system. Each of these groups is then reconciled to the internal LDAP store. No reconciliation to Oracle Identity Manager is performed.

Table 5-2 describes the attributes of the scheduled task.

Table 5-2 Attributes of the RACF Reconcile Groups To Internal LDAP Task

Attribute	Description
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored. Sample value: `racf`
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`

5.2.2 RACF Find All LDAP Groups

This scheduled task populates lookup fields in Oracle Identity Manager with resource group IDs from internal LDAP. Values from these lookup fields can be assigned during user provisioning operations and reconciliation runs. When you configure this scheduled task, it runs at specified intervals and fetches a listing of all group IDs on the internal LDAP for reconciliation.

Table 5-3 describes the attributes of the scheduled task.

Table 5-3 Attributes of the RACF Find All LDAP Groups Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource` Groups are reconciled from internal LDAP store using the IT Resource.
Secondary IT Resource	Enter the name of the secondary IT resource that was configured for the target system. Sample value: `SecondResource` Groups are stored in OIM in the following format:`<Secondary IT Resource>~<Group Name>`.
Filter	Filter to be specified in LDAP format. Sample value for Simple Filter: `(cn=<groupID>)` Sample value for Complex Filter: `(&(commandflag=ADD)(cn=<groupID>))`
Resource Object	Enter the name of the resource object against which provisioning runs must be performed. Sample value: `OIMRacfResourceObject`
Lookup Code Name	Enter the name of the lookup code where OIM will store the results of the scheduled task. Sample value: `Lookup.GroupNames`

Table 5-3 (Cont.) Attributes of the RACF Find All LDAP Groups Task

Attribute	Description
Recon Type	This attribute determines how group memberships from the target system are populated in Oracle Identity Manager lookup definitions. You can use one of the following options: • `Append` adds group membership entries from the target system that do not exist in the Lookup.GroupNames lookup definitions. Any existing entries remain untouched. • `Replace` removes all the existing entries in Lookup.GroupNames lookup definition and replaces them with group membership entries from the target system. • `Merge` handles entries in the following manner: If you are using the connector for a single installation of the target system, then group membership entries that exist in both the target system and Oracle Identity Manager are updated in the Lookup.GroupNames lookup definitions. Group membership entries that exist only in the target system are added to the Lookup.GroupNames lookup definitions. If you are using the connector for multiple installations of the target system, then only the group membership entries corresponding to the target system installation that you are using are updated or added. Entries that exist in both the target system and Oracle Identity Manager are updated in the Lookup.GroupNames lookup definitions. Entries that exist only in the target system are added to the Lookup.GroupNames lookup definitions. Default value: Merge
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored. Sample value: `ou=racf,ou=Groups`
AttrsToReturn	Enter a comma-separated list of object attributes that the connector must retrieve from internal LDAP. For example, enter a comma-separated list of group attributes that the connector must fetch from LDAP and load into Oracle Identity Manager. . Sample value: `cn,commandFlag` **Note:** cn and commandFlag attributes are mandatory

Table 5-3 (Cont.) Attributes of the RACF Find All LDAP Groups Task

Attribute	Description
DescTemplate	By default, when lookup reconciliation is performed, the lookup description is same as the lookup value in the lookup window. Therefore, if required, use the DescTemplate attribute to specify the attribute whose value must be used as the lookup description and displayed in the lookup window.
SearchBaseDN	This should be kept blank. This is reserved for future use.

5.3 Configuring the Security Attributes Lookup Field

The Lookup.RacfSecurityAttributeNames lookup definition is one of the lookup definitions that is created in Oracle Identity Manager when you deploy the connector. This lookup field is populated with standard RACF nonvalue security attributes such as ADSP, AUDIT, SPECIAL, and so on.

The IBM RACF Advanced connector includes a scheduled task to automatically populate the lookup field used for storing RACF security attributes.

This section contains the following topics:

- Attributes of the Find All Security Attributes Scheduled Task
- Adding Additional Security Attributes for Provisioning and Reconciliation

5.3.1 Attributes of the Find All Security Attributes Scheduled Task

The IBM RACF Advanced connector includes a scheduled task to automatically populate the lookup field used for storing RACF security attributes.

> **Note:**
>
> The Find All Security Attributes scheduled task does not query the target system for data. Instead, the scheduled task automatically populates the lookup field with "itResourceKey~sourceName" pairs based on the IT Resource and Find All Security Attributes scheduled task property values.

Table 5-4 describes the properties of the Find All Security Attributes scheduled task.

Table 5-4 Attributes of the Find All Security Attributes Scheduled Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`
Security Attributes	Enter a comma-separated list of RACF non-value security attributes. Sample value: `ADSP, AUDIT, RESTRICTED, SPECIAL, UAUDIT`

Table 5-4 (Cont.) Attributes of the Find All Security Attributes Scheduled Task

Attribute	Description
Lookup Code Name	Enter the name of the lookup code where Oracle Identity Manager will store the source entries. Sample value: `Lookup.RacfSecurityAttributes`
Recon Type	This attribute determines how security attributes from the target system are populated in Oracle Identity Manager lookup definitions. You can use one of the following options: • `Append` adds security attributes from the target system that do not exist in the Lookup.RacfSecurityAttributes lookup definition. Any existing entries remain untouched. • `Replace` removes all the existing entries in the Lookup.RacfSecurityAttributes lookup definition and replaces them with security attributes from the target system. • `Merge` handles entries in the following manner: – If you are using the connector for a single installation of the target system, then security attributes that exist in both the target system and Oracle Identity Manager are updated in the Lookup.RacfSecurityAttributes lookup definition. Security attributes that exist only in the target system are added to the Lookup.RacfSecurityAttributes lookup definitions. – If you are using the connector for multiple installations of the target system, then only security attributes corresponding to the target system installation that you are using are updated or added. Security attributes that exist in both the target system and Oracle Identity Manager are updated in the Lookup.RacfSecurityAttributes lookup definition. Security attributes that exist only in the target system are added to the Lookup.RacfSecurityAttributes lookup definition. Default value: `Merge`

However, you can also manually add additional values. See Adding Additional Security Attributes for Provisioning and Reconciliation.

5.3.2 Adding Additional Security Attributes for Provisioning and Reconciliation

To add additional security attributes for provisioning and reconciliation:

1. Login to Oracle Identity Manager Design Console.
2. Expand **Administration**, and then double-click **Lookup Definition**.
3. Search for the **Lookup.RacfSecurityAttributesNames** lookup definition.
4. Click **Add**.
5. In the Code Key column, enter the name of the security attribute. Enter the same value in the Decode column. The following is a sample entry:

   Code Key: `ITResource~ADSP` Decode: `ITResource~ADSP`
6. Click the Save icon.

5.4 Configuring Reconciliation

The IBM RACF Advanced connector supports both incremental reconciliation (sometimes referred to as real-time reconciliation) and full reconciliation. This section discusses the following topics related to configuring reconciliation:

- Configuring Incremental Reconciliation
- Performing Full Reconciliation
- Reconciliation Scheduled Tasks
- Guidelines for Configuring Filtered Reconciliation to Multiple Resource Objects

5.4.1 Configuring Incremental Reconciliation

The Voyager agent and the LDAP gateway perform incremental reconciliation using the RACF Reconcile All LDAP Users scheduled task. To configure incremental reconciliation:

1. Ensure the racf.properties has the following set:
   - USE INTERNAL META STORE

     ```
     [true|false]_internalEnt_=true
     ```
   - USE GROUP INTERNAL META STORE

     ```
     [true|false]_internalGrpEnt_=true
     ```
2. Use the Last Modified Timestamp parameter of the IT resource to set a date range that will reconcile all users that have changed since that date.

> **Note:**
>
> If the _internalEnt_ property, located in LDAP_INSTALL_DIR/conf/racf.properties, is set to true, then the LDAP internal store will also be populated on an ongoing basis by the "real-time" event capture using Voyager and the EXIT(s). So after initial population and reconciliation the process will still continue to use the RACF Reconcile All LDAP Users scheduled task using a Date range to reconcile these real-time event changes from data captured in the LDAP internal store.

5.4.2 Performing Full Reconciliation

Full reconciliation involves reconciling all existing user records from the target system into Oracle Identity Manager.

After you deploy the connector, you must first perform full reconciliation. After first-time reconciliation, the connector will automatically switch to performing incremental reconciliation based on the time stamp value present in the IT resource.

To perform full reconciliation in a set up that involves LDAP gateway as an intermediary datastore between the RACF target system and Oracle Identity Manager, choose one of the options:

- If you are performing reconciliation for the first time, then:
  1. Generate an EXTRACT reconciliation file on the RACF target system. To do so:

a. For z/OS 2.3 :
On the mainframe, execute the RACFRCOU or RACFRCOG batch jobs for reconciling Users or Groups, respectively. These batch jobs populate user and group data in the &HLQ..PIONEER.IMPORTU.FILE dataset (referenced with DD name //FULLIMPU inside PIONEER STC Procedure) and &HLQ..PIONEER.IMPORTG.FILE dataset (referenced with DD name //FULLIMPG inside PIONEER STC Procedure), respectively.

   These batch jobs are a member of the <hlq>.JCLLIB dataset that is available in the etc/Provisioning and Reconciliation Connector/RACF-AGENTS-201905311134-6.0.0.zip file of the connector installation media.

   When Pioneer receives request for full reconciliation (user or group), it reads the corresponding dataset and sends the response back to gateway and clears the dataset. After each execution of full reconciliation, the corresponding file gets cleared. Therefore, if required, you must regenerate the EXTRACT file for populating the internal LDAP for Oracle Identity Manager to reconcile the latest data.

   For reconciling all Dataset profiles and General resource profiles from RACF DB, we are using RACF DB unload. In the distribution, we are shipping sample JCL, named RACFRC (<HLQ>.JCLLIB(RACFRC)). In this JCL, the first step is to take the RACF DB unload using IRRDBU00 utility. The details about this utility can be referred at: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.icha700/usdbum.htm (Z/OS V2.4)

   Following steps in the Job are different sort steps, manipulating the RACF DB unload data for Datasets and general resource profile records and arranging them in sequence. The DD name IMPORTD maps to the final file containing all dataset profiles. This file name must match with the file name in PIONEER STC DD name FULLIMPD. The DD name IMPORTR maps to the final file containing all general resource profiles. This file name must match with the file name in PIONEER STC DD name FULLIMPR.

b. For z/OS 2.4 :
On the mainframe, execute the RACFRCOU or RACFRCOG batch jobs for reconciling Users or Groups, respectively. These batch jobs populate user and group data in the &HLQ..PIONEER.IMPORTU.FILE dataset (referenced with DD name //FULLIMPU inside PIONEER STC Procedure) and &HLQ..PIONEER.IMPORTG.FILE dataset (referenced with DD name //FULLIMPG inside PIONEER STC Procedure), respectively.

   These batch jobs are a member of the <hlq>.JCLLIB dataset that is available in the etc/Provisioning and Reconciliation Connector/RACF-AGENTS-201905311134-6.0.0.zip file of the connector installation media.

   When Pioneer receives request for full reconciliation (user or group), it reads the corresponding dataset and sends the response back to gateway and clears the dataset. After each execution of full reconciliation, the corresponding file gets cleared. Therefore, if required, you must regenerate the EXTRACT file for populating the internal LDAP for Oracle Identity Manager to reconcile the latest data.

   For reconciling all Dataset profiles and General resource profiles from RACF DB, we are using RACF DB unload. In the distribution, we are shipping sample JCL, named RACFRC (<HLQ>.JCLLIB(RACFRCOD) and <HLQ>.JCLLIB(RACFRCOR)). In this JCL, the first step is to take the RACF DB unload using IRRDBU00 utility. The details about this utility can be referred at: https://www.ibm.com/support/knowledgecenter/SSLTBW_2.4.0/com.ibm.zos.v2r4.icha700/usdbum.htm (Z/OS V2.4)

Following steps in the Job are different sort steps, manipulating the RACF DB unload data for Datasets and general resource profile records and arranging them in sequence. The DD name IMPORTD maps to the final file containing all dataset profiles. This file name must match with the file name in PIONEER STC DD name FULLIMPD. The DD name IMPORTR maps to the final file containing all general resource profiles. This file name must match with the file name in PIONEER STC DD name FULLIMPR.

2. Set the value of the Last Modified Time Stamp parameter of the IT resource parameter to `0`.
3. Run the RACF Reconcile Users to Internal LDAP scheduled task.
4. Run the RACF Reconcile Groups To Internal LDAP scheduled task.
5. Run the RACF Reconcile Datasets To Internal LDAP scheduled task.
6. Run the RACF Reconcile Resources To Internal LDAP scheduled task.
7. Run the RACF Reconcile All LDAP Users scheduled task.

> **Note:**
>
> If you do not run the RACF Recon Users to Internal LDAP scheduled task with the EXTRACT recon file, then the RACF Reconcile LDAP Users scheduled task will always perform in incremental mode.

- If this not the first time that you are performing full reconciliation, then:
  1. Set the value of the Last Modified Time Stamp parameter of the IT resource parameter to `0`.
  2. Run the RACF Reconcile All LDAP Users scheduled task.

> **Note:**
>
> If updates for a user are complete and the user is reconciles, in order to to see the datasets/resources again you need to run the following jobs again
>
> - RACF Reconcile Datasets To Internal LDAP
> - RACF Reconcile Resources To Internal LDAP
> - RACF Reconcile All LDAP Users

This completes full reconciliation and from the next reconciliation run onward, the connector will automatically switch to incremental reconciliation by using the value in the Last Modified Time Stamp parameter of the IT resource.

To perform full reconciliation in a set up that does not involve LDAP gateway, run the RACF Reconcile All Users scheduled task. The scheduled job will always run in full reconciliation mode.

5.4.3 Reconciliation Scheduled Tasks

When you run the Connector Installer, these reconciliation scheduled tasks are automatically created in Oracle Identity Manager.

- RACF Reconcile All Users
- RACF Deleted User Reconciliation Using OIM
- RACF Reconcile Users to Internal LDAP
- RACF Reconcile All LDAP Users

5.4.3.1 RACF Reconcile All Users

The RACF Reconcile All Users scheduled task is used to reconcile user data in the target resource (account management) mode of the connector. This scheduled task runs at specified intervals and fetches create or modify events on the target system for reconciliation.

Table 5-5 describes the attributes of the scheduled task.

Table 5-5 Attributes of the RACF Reconcile All Users Scheduled Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`
Resource Object	Enter the name of the resource object against which reconciliation runs must be performed. Sample value: `OIMRacfResourceObject`
MultiValuedAttributes	Enter a comma-separated list of multivalued attributes that you want to reconcile. Do not include a space after each comma. Sample value:`attributes,member of`
SingleValueAttributes	Enter a comma-separated list of single-valued attributes that you want to reconcile. Do not include a space after each comma. Do not include attributes already listed in the MultiValueAttributes field. Sample value: `uid,owner,defaultGroup,waddr1,tsoMaxSize` **Note**: By default, Oracle Identity Manager's design form only allows entering up to 150 characters in a text field. To increase this limit, change the value of the TSA_VALUE column in Oracle Identity Manager database.
UID Case	Enter either "upper" or "lower" for the case for the UID attribute value. Sample value: `upper`
UsersList	Enter a comma-separated list of UIDs that you want to reconcile from the target system. If this property is left blank, all users on the target system will be reconciled. Sample value: `userQA01,georgeb,marthaj,RST0354`

Table 5-5 (Cont.) Attributes of the RACF Reconcile All Users Scheduled Task

Attribute	Description
Filter	Enter a filter criteria to search for and retrieve user records that match the given filter criteria. You can use any target system attribute to create the filter criterion. The filter criterion that you enter must be a valid filter according to RFC2254. The filter can be either simple or complex. A simple filter uses only a (uid=<userid>) condition whereas a complex filter is a combination of one or more attributes. Sample value for a simple filter: `(uid=<userid>)` Sample value for a complex filter: `(&(commandflag=UPDATE)(revoke=n))` This complex filter searches for and retrieves all user records whose commandflag attribute value is UPDATE and revoke is n. **Note**: If you specify a complex filter, then ensure that you have enabled the caching layer of the LDAP Gateway as described in Understanding the Caching Layer. If the caching layer is disabled, then the connector considers only the simple filter (uid=<userid>).

5.4.3.2 RACF Deleted User Reconciliation Using OIM

The RACF Reconcile Deleted Users to OIM scheduled task is used to reconcile data about deleted users in the target resource (account management) mode of the connector.

When you configure this scheduled task, it runs at specified intervals and fetches a list of users on the target system. These user names are then compared with provisioned users in Oracle Identity Manager. Any user profiles that exist within Oracle Identity Manager, but not in the target system, are deleted from Oracle Identity Manager.

Table 5-6 describes the attributes of the scheduled task.

Table 5-6 Attributes of the RACF Reconcile Deleted Users to Oracle Identity Manager Scheduled Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value:`RacfResource`
Resource Object	Enter the name of the resource object against which the delete reconciliation runs must be performed. Sample value: `OIMRacfResourceObject`
Recon Matching Rule Attributes	Enter a comma-separated list of attributes used in the matching rule. If the IT resource is used, enter `IT`. Sample value: `UID,IT`

5.4.3.3 RACF Reconcile Users to Internal LDAP

The RACF Reconcile Users to Internal LDAP scheduled task is used to process the CFILE extract from the target system to the internal LDAP store. When you configure this scheduled task, it runs at specified intervals and fetches a list of users and their profiles on the target system. Each of these users is then reconciled to the internal LDAP store. No reconciliation to Oracle Identity Manager is performed.

Table 5-7 describes the attributes of the scheduled task.

Table 5-7 Attributes of the RACF Reconcile Users to Internal LDAP Scheduled Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored.Sample value: `racf`

5.4.3.4 RACF Reconcile All LDAP Users

The RACF Reconcile All LDAP Users scheduled task is used to reconcile users from the internal LDAP store to Oracle Identity Manager. When you configure this scheduled task, it runs at specified intervals and fetches a list of users within the internal LDAP store and reconciles these users to Oracle Identity Manager.

Table 5-8 describes the attributes of the scheduled task.

Table 5-8 Attributes of the RACF Reconcile All LDAP Users Scheduled Task

Attribute	Description
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`
Secondary IT Resource	Enter the name of the secondary IT resource that was configured for the target system. Sample value: `SecondResource`
Resource Object	Enter the name of the resource object against which the delete reconciliation runs must be performed. Sample value: `OIMRacfResourceObject`
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored.Sample value: `racf`
MultiValuedAttributes	Enter a comma-separated list of multivalued attributes that you want to reconcile. Do not include a space after each comma. Sample value: `member of,attributes,datasets,resources`
SingleValueAttributes	Enter a comma-separated list of single-valued attributes that you want to reconcile. Do not include a space after each comma. Do not include attributes already listed in the MultiValueAttributes field. Sample value: `uid,owner,defaultGroup,waddr1,tsoMaxSize` **Note:** By default, Oracle Identity Manager's design form only allows entering up to 150 characters in a text field. To increase this limit, change the value of the TSA_VALUE column in the Oracle Identity Manager database.
LDAP Time Zone	Enter the full OIM server timezone database name value. Do not use the abbreviated timezone value. To find out the timezone database name value refer to List of tz database time zones. Sample value: `America/New York`

Table 5-8 (Cont.) Attributes of the RACF Reconcile All LDAP Users Scheduled Task

Attribute	Description
UID Case	Enter whether the user ID should be displayed in uppercase or lowercase. Sample value: `upper`
Filter	Enter a filter criteria to search for and retrieve user records that match the given filter criteria. You can use any target system attribute to create the filter criterion. The filter criterion that you enter must be a valid filter according to RFC2254. The filter can be either simple or complex. A simple filter uses only a (uid=<userid>) condition whereas a complex filter is a combination of one or more attributes. Sample value for a simple filter: `(uid=<userid>)` Sample value for a complex filter: `(&(commandflag=UPDATE)(revoke=n))` This complex filter searches for and retrieves all user records whose commandflag attribute value is UPDATE and revoke is n.

5.4.3.5 RACF Reconcile Datasets To Internal LDAP

The RACF Reconcile Datasets To Internal LDAP scheduled task is used to process the Datasets file extract from the target system to the internal LDAP store. When you configure this scheduled task, it runs at specified intervals and fetches a list of Datasets on the target system. Each of these Datasets is then reconciled to the internal LDAP store. No reconciliation to Oracle Identity Manager is performed.

Note:

- For z/OS 2.3 the <HLQ>.JCLLIB(RACFRC) job needs to be executed on the mainframe prior to executing this task.
- For z/OS 2.4 the <HLQ>.JCLLIB(RACFRCOD) job needs to be executed on the mainframe prior to executing this task.

Table 5-9 describes the attributes of the scheduled task.

Table 5-9 Attributes of the RACF Reconcile Datasets To Internal LDAP Task

Attribute	Description
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored. Sample value: `racf`
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`

5.4.3.6 RACF Reconcile Resources To Internal LDAP

The RACF Reconcile Resources To Internal LDAP scheduled task is used to process the Resources file extract from the target system to the internal LDAP store. When you configure this scheduled task, it runs at specified intervals and fetches a list of Resources on the target system. Each of these Resources is then reconciled to the internal LDAP store. No reconciliation to Oracle Identity Manager is performed.

Note:

- For z/OS 2.3 the <HLQ>.JCLLIB(RACFRC) job needs to be executed on the mainframe prior to executing this task.
- For z/OS 2.4 the <HLQ>.JCLLIB(RACFRCOR) job needs to be executed on the mainframe prior to executing this task.

Table 5-10 describes the attributes of the scheduled task.

Table 5-10 Attributes of the RACF Reconcile Resources To Internal LDAP Task

Attribuite	Description
Domain OU	Enter the name of the internally-configured directory in the LDAP internal store where the contents of event changes will be stored. Sample value: `racf`
IT Resource	Enter the name of the IT resource that was configured for the target system. Sample value: `RacfResource`

5.4.4 Guidelines for Configuring Filtered Reconciliation to Multiple Resource Objects

Some organizations use multiple resource objects to represent multiple user types in their system. The Resource Object property of the RACF Reconcile All Users scheduled task is used to specify the resource object used during reconciliation, and you can enter more than one resource object in the value of the Resource Object attribute. Further, you can include IBM RACF attribute-value pairs to filter records for each resource object.

See Also:

RACF Reconcile All Users for information about the RACF Reconcile All Users scheduled task

The following is a sample format of the value for the Resource Object attribute:

```
(ATTRIBUTE1:VALUE1)RESOURCE_OBJECT1,RESOURCE_OBJECT2
```

As shown by RESOURCE_OBJECT2 in the sample format, specifying a filter attribute is optional, but if more than one resource object is specified, you must specify a filter for each additional resource object. If you do not specify a filter attribute, then all records are reconciled to the first resource object in the list. Further, the filters are checked in order, so the resource object without a filter attribute should be included last in the list.

Filter attributes should be surrounded by parentheses.

Apply the following guidelines while specifying a value for the Resource Object attribute:

- The names of the resource objects must be the same as the names that you specified while creating the resource objects in the Oracle Identity Manager Design Console.
- The IBM RACF attribute names must be the same as the names used in the LDAP Gateway configuration files.
- The value must be a regular expression as defined in the java.util.regex Java package. Note that the find() API call of the regex matcher is used rather than the matches() API call. This means that a substring matching rule can be specified in the pattern, rather than requiring the entire string matching rule.

  Further, substring matching is case-sensitive. A "(tso)" filter will not match a user with the user ID "TSOUSER1".
- Multiple values can be matched. Use a vertical bar (|) for a separator as shown in the following example:

  (*ATTRIBUTE*:*VALUE1*|*VALUE2*|*VALUE3*)*RESOURCE_OBJECT*
- Multiple filters can be applied to the attribute and to the same resource object. For example:

  (*ATTRIBUTE1:VALUE1*)&(*ATTRIBUTE2*:*VALUE2*)*RESOURCE_OBJECT*

The following is a sample value for the Resource Object attribute:

```
(tsoProc:X)TSSR01,(instdata:value1|value2|value3)RacfResourceObject2,
(tso)RacfResourceObject24000,Resource
```

In this sample value:

- (tsoProc:X)TSSRO1 represents a user with X as the attribute value for the TSO Proc segment. Records that meet this criterion are reconciled with the TSSRO1 resource object.
- (instdata:value1|value2|value3)RacfResourceObject2 represents a user with value1, value2, or value3 as their INSTDATA attribute value. Records that meet this criterion are reconciled with the RacfResourceObject2 resource object.
- (tso)RacfResourceObject24000 represents a user with TSO privileges. A TSO attribute value is not specified. Records that meet this criterion are reconciled with the RacfResourceObject24000 resource object.
- All other records are reconciled with the resource object.

5.5 Configuring Account Status Reconciliation for IBM RACF Advanced Connector

> **Note:**
>
> This section describes an optional procedure. Perform this procedure only if you want reconciliation of user status changes on IBM RACF.

When a user is disabled or enabled on the target system, the status of the user can be reconciled into Oracle Identity Manager. To configure reconciliation of user status changes made on IBM RACF:

1. In the RACF Reconcile All Users scheduled task, add the Status attribute to the SingleValueAttributes property list.
2. Log in to the Design Console:
   - In the **OIMRacfSecretResourceObject** resource object, create a reconciliation field to represent the Status attribute.
   - In the **OIMRacfProvisioningProcess** process definition, map the field for the Status field to the OIM_OBJECT_STATUS field.

5.6 Scheduled Tasks for IBM RACF Advanced Connector

Table 5-11 lists the scheduled tasks that you must configure.

Table 5-11 Scheduled Tasks for Lookup Field Synchronization and Reconciliation for IBM RACF

Scheduled Task	Description
RACF Find All Resources	This scheduled task is used to synchronize the values of resource profile lookup fields between Oracle Identity Manager and the target system. For information about this scheduled task and its attributes, see Scheduled Tasks for Lookup Field Synchronization.
RACF Find All Datasets	This scheduled task is used to synchronize the values of dataset lookup fields between Oracle Identity Manager and the target system. For information about this scheduled task and its attributes, see Scheduled Tasks for Lookup Field Synchronization.
RACF Find All Groups	This scheduled task is used to synchronize the values of group lookup fields between Oracle Identity Manager and the target system. For information about this scheduled task and its attributes, see Scheduled Tasks for Lookup Field Synchronization.
RACF Find All Security Attributes	This scheduled task is used to automatically populate the security attributes lookup field with IT Resource Key~Security Attribute Name pairs. For information about this scheduled task and its attributes, see Configuring the Security Attributes Lookup Field.
RACF Reconcile All Users	This scheduled task is used to fetch user data during target resource reconciliation. For information about this scheduled task and its attributes, see RACF Reconcile All Users.
RACF Reconcile Deleted Users to OIM	This scheduled task is used to fetch data about deleted users during target resource reconciliation. During a reconciliation run, for each deleted user account on the target system, the RACF User resource is revoked for the corresponding OIM User. For information about this scheduled task and its attributes, see RACF Deleted User Reconciliation Using OIM.

Table 5-11 (Cont.) Scheduled Tasks for Lookup Field Synchronization and Reconciliation for IBM RACF

Scheduled Task	Description
RACF Reconcile Users to Internal LDAP	This scheduled task is used to reconcile users from the target system to the internal LDAP store. For information about this scheduled task and its attributes, see RACF Reconcile Users to Internal LDAP.
RACF Reconcile All LDAP Users	This scheduled task is used to reconcile users from the internal LDAP store to Oracle Identity Manager. For information about this scheduled task and its attributes, see RACF Reconcile All LDAP Users.

5.7 Configuring Reconciliation Jobs

Configure reconciliation jobs to perform reconciliation runs that check for new information on your target system periodically and replicates the data in Oracle Identity Governance.

You can apply this procedure to configure the reconciliation jobs for users and entitlements.

To configure a reconciliation job:

1. Log in to Identity System Administration.
2. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the scheduled job as follows:
   a. In the Search field, enter the name of the scheduled job as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
   b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. On the Job Details tab, you can modify the parameters of the scheduled task:
   - **Retries**: Enter an integer value in this field. This number represents the number of times the scheduler tries to start the job before assigning the Stopped status to the job.
   - **Schedule Type**: Depending on the frequency at which you want the job to run, select the appropriate schedule type. See Creating Jobs in *Oracle Fusion Middleware Administering Oracle Identity Governance*.

   In addition to modifying the job details, you can enable or disable a job.
5. On the **Job Details** tab, in the Parameters region, specify values for the attributes of the scheduled task.

   > **Note:**
   >
   > Values (either default or user-defined) must be assigned to all the attributes. If even a single attribute value is left empty, then reconciliation is not performed.

6. Click **Apply** to save the changes.

> **Note:**
>
> You can use the Scheduler Status page in Identity System Administration to either start, stop, or reinitialize the scheduler.

5.8 Performing Provisioning Operations

You create a new user in Identity Self Service by using the Create User page. You provision or request for accounts on the Accounts tab of the User Details page.

To perform provisioning operations in Oracle Identity Governance:

1. Log in to Identity Self Service.
2. Create a user as follows:
   a. In Identity Self Service, click **Manage**. The Home tab displays the different Manage option. Click **Users**. The Manage Users page is displayed.
   b. From the Actions menu, select **Create**. Alternatively, you can click **Create** on the toolbar. The Create User page is displayed with input fields for user profile attributes.
   c. Enter details of the user in the Create User page.
3. On the Account tab, click **Request Accounts**.
4. In the Catalog page, search for and add to cart the application instance for the connector that you configured earlier, and then click **Checkout**.
5. Specify value for fields in the application form and then click **Ready to Submit**.
6. Click **Submit**.

> **See Also:**
>
> Creating a User in *Oracle Fusion Middleware Performing Self Service Tasks with Oracle Identity Governance* for details about the fields on the Create User page

6

Extending the Functionality of the IBM RACF Advanced Connector

These are the optional procedures that you can perform to extend the functionality of the connector for addressing your business requirements.

- Adding Custom Fields for Target Resource Reconciliation
- Adding Custom Multivalued Fields for Reconciliation
- Adding Custom Fields for Provisioning for IBM RACF Advanced Connector
- Removing Attributes Mapped for Target Resource Reconciliation
- Using the Provisioning Agent to Run IBM z/OS Batch Jobs
- Configuring the Connector for Provisioning to Multiple Installations of the Target System
- Customizing Log File Locations
- LDAP Reconciliation Supported Queries
- Handling Pioneer Error Messaging Exceptions in the Gateway

6.1 Adding Custom Fields for Target Resource Reconciliation

To add a custom field for reconciliation, you must first update the connector reconciliation component you are using, and then update Oracle Identity Manager.

> **Note:**
>
> You must ensure that new attributes you add for reconciliation contain only string-format data. Binary attributes must not be brought into Oracle Identity Manager natively.

By default, the attributes listed in Table 1-3 are mapped for reconciliation between Oracle Identity Manager and the target system. If required, you can add new attributes for target resource reconciliation.

This section discusses the following topics:

- Adding Custom Fields for Reconciliation
- Adding Custom Fields to Oracle Identity Manager

6.1.1 Adding Custom Fields for Reconciliation

You can add custom fields for reconciliation by specifying a value for the SingleValueAttributes attribute of the RACF Reconcile All Users and RACF Reconcile All LDAP Users scheduled tasks.

To add a custom field for scheduled task reconciliation:

1. Log in to Identity System Administration.
2. In the left pane, under System Management, click **Scheduler**.
3. Search for and open the **RACF Reconcile All Users** and **RACF Reconcile All LDAP Users** scheduled tasks as follows:
    a. On the left pane, in the Search field, enter `RACF Reconcile All Users` or `RACF Reconcile All LDAP Users` as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.
    b. In the search results table on the left pane, click the scheduled job in the Job Name column.
4. Add the custom field to the list of attributes in the **SingleValueAttributes** scheduled task attribute.
5. Click **Apply**.

6.1.2 Adding Custom Fields to Oracle Identity Manager

After adding the custom field to the RACF Reconcile All users scheduled task (if using scheduled task reconciliation), you must add the custom field to the Oracle Identity Manager components.

To update Oracle Identity Manager with the custom field:

1. Log in to the Oracle Identity Manager Design Console.
2. Add the custom field to the list of reconciliation fields in the resource object as follows:
    a. Expand **Resource Management** and then double-click **Resource Objects.**
    b. Search for and open the **OIMRacfResourceObject** resource object.
    c. On the Object Reconciliation tab, click **Add Field.**
    d. In the Add Reconciliation Field dialog box, enter the details of the field.

       For example, if you are adding a RACF attribute called "Description", then enter `Description` in the Field Name field and select **String** from the Field Type list.
    e. Click **Save** and close the dialog box.
    f. Click **Create Reconciliation Profile.** This copies changes made to the resource object into MDS.
    g. Click **Save.**
3. Add the custom field on the process form as follows:
    a. Expand **Development Tools** and then double-click **Form Designer.**
    b. Search for and open the **UD_RACF_ADV** process form.
    c. Click **Create New Version,** and then click **Add.**
    d. Enter the details of the field.

       For example, if you are adding the Description field, then enter `UD_RACF_ADV_DESCRIPTION` in the Name field, and then enter the rest of the details of this field.
    e. Click **Save** and then click **Make Version Active.**

4. Create a reconciliation field mapping for the custom field in the provisioning process as follows:

   a. Expand **Process Management** and then double-click **Process Definition.**

   b. Search for and open the **OIMRacfProvisioningProcess** process definition.

   c. On the Reconciliation Field Mappings tab of the provisioning process, click **Add Field Map.**

   d. In the Add Reconciliation Field Mapping dialog box, from the Field Name field, select the value for the field that you want to add. For example, from the Field Name field, select **Description.**

   e. Double-click the **Process Data field**, and then select **UD_RACF_ADV_DESCRIPTION.**

   f. Click **Save** and close the dialog box.

   g. Click **Save.**

5. Create a new UI form and attach it to the application instance to make this new attribute visible. See Creating a New UI Form and Updating an Existing Application Instance with a New Form for the procedures.

6. If you are adding a custom attribute or custom dataset, then set values for the `_configAttrs_`, `_configDNames_`, and `_configDatasets_` properties in the racf.properties file. See Table 2-2 for information about these properties.

6.2 Adding Custom Multivalued Fields for Reconciliation

To add a custom multivalued field to reconciliation, you must first update the IDF reconciliation component you are using, and then update Oracle Identity Manager.

- Adding Custom Multivalued Fields for Reconciliation
- Adding Custom Multivalued Fields

6.2.1 Adding Custom Multivalued Fields to the Reconciliation Component

You can add custom multivalued fields for reconciliation by specifying a value for the MultiValuedAttributes property of the RACF Reconcile All Users and RACF Reconcile All LDAP Users scheduled tasks.

To add a custom field for scheduled task reconciliation:

1. Log in to Identity System Administration.

2. In the left pane, under System Management, click **Scheduler**.

3. Search for and open the **RACF Reconcile All Users** and **RACF Reconcile All LDAP Users** scheduled tasks as follows:

   a. On the left pane, in the Search field, enter `RACF Reconcile All Users` or `RACF Reconcile All LDAP Users` as the search criterion. Alternatively, you can click **Advanced Search** and specify the search criterion.

   b. In the search results table on the left pane, click the scheduled job in the Job Name column.

4. Add the custom field to the list of attributes in the `MultiValuedAttributes property`.

5. Click **Apply**.

6.2.2 Adding Custom Multivalued Fields

After adding the custom multivalued field to the RACF Reconcile All users scheduled task (if using scheduled task reconciliation), you must add the custom multivalued field to the Oracle Identity Manager components.

To update Oracle Identity Manager with the multivalued field:

1. Log in to the Oracle Identity Manager Design Console.
2. Create a form for the multivalued field as follows:
   a. Expand **Development Tools** and double-click **Form Designer.**
   b. Create a form by specifying a table name and description, and then click **Save.**
   c. Click **Add** and enter the details of the field.
   d. Click **Save** and then click **Make Version Active.** Figure 6-1 shows the multivalued field added on a new form.

Figure 6-1 Multivalued Field Added on a New Form

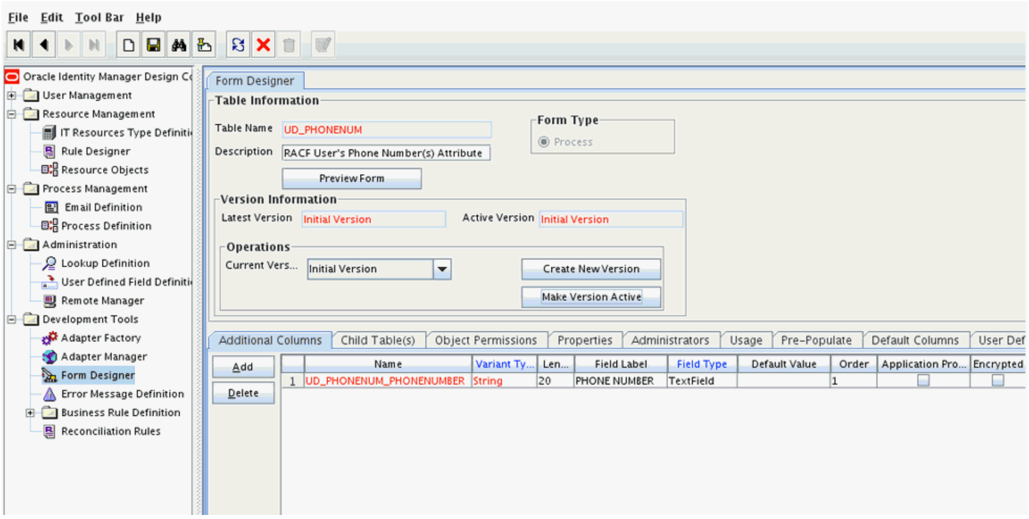


Table Information

Table Name: UD_PHONENUM

Description: RACF User's Phone Number(s) Attribute

Form Type: ● Process

Version Information

Latest Version: Initial Version

Active Version: Initial Version

Operations

Current Vers...: Initial Version

	Name	Variant Ty...	Len...	Field Label	Field Type	Default Value	Order	Application Pro...	Encrypted
1	UD_PHONENUM_PHONENUMBER	String	20	PHONE NUMBER	TextField		1	□	□

3. Add the form created for the multivalued field as a child form of the process form as follows:
   a. Search for and open the **UD_RACF_ADV** process form.
   b. Click **Create New Version.**
   c. Click the **Child Table(s)** tab.
   d. Click **Assign.**
   e. In the Assign Child Tables dialog box, select the newly created child form, click the **right arrow,** and then click **OK.**
   f. Click **Save** and then click **Make Version Active.** Figure 6-2 shows the child form added to the process form.

Figure 6-2 Child Form Added to the Process Form

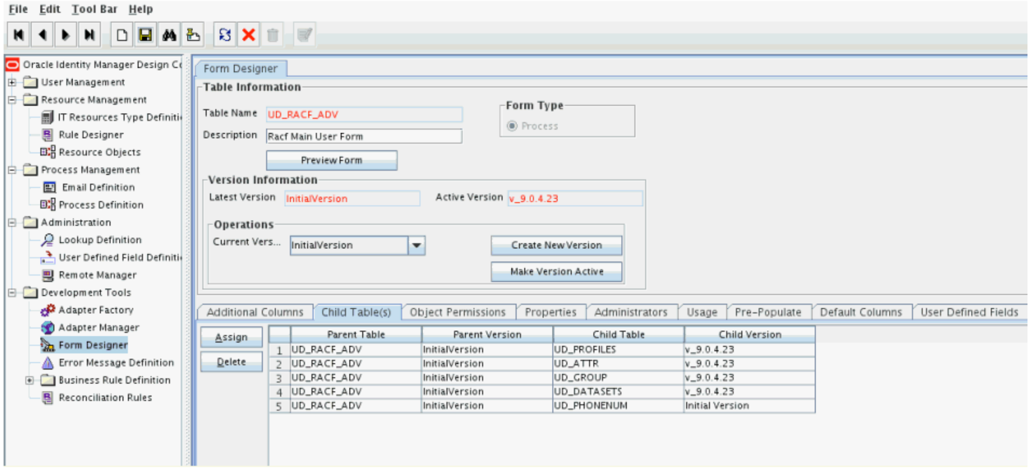


Table Information

Table Name: UD_RACF_ADV

Description: Racf Main User Form

Form Type: ● Process

Version Information

Latest Version: InitialVersion

Active Version: v_9.0.4.23

Operations

Current Vers...: InitialVersion

	Parent Table	Parent Version	Child Table	Child Version
1	UD_RACF_ADV	InitialVersion	UD_PROFILES	v_9.0.4.23
2	UD_RACF_ADV	InitialVersion	UD_ATTR	v_9.0.4.23
3	UD_RACF_ADV	InitialVersion	UD_GROUP	v_9.0.4.23
4	UD_RACF_ADV	InitialVersion	UD_DATASETS	v_9.0.4.23
5	UD_RACF_ADV	InitialVersion	UD_PHONENUM	Initial Version

4. Add the new multivalued field to the list of reconciliation fields in the resource object as follows:

   a. Expand **Resource Management** and then double-click **Resource Objects.**

   b. Search for and open the **OIMRacfObject** resource object.

   c. On the Object Reconciliation tab, click **Add Field.**

   d. In the Add Reconciliation Field dialog box, enter the details of the field.

      For example, enter `phoneNumber` in the Field Name field and select **Multi-Valued Attribute** from the Field Type list.

   e. Click **Save** and close the dialog box.

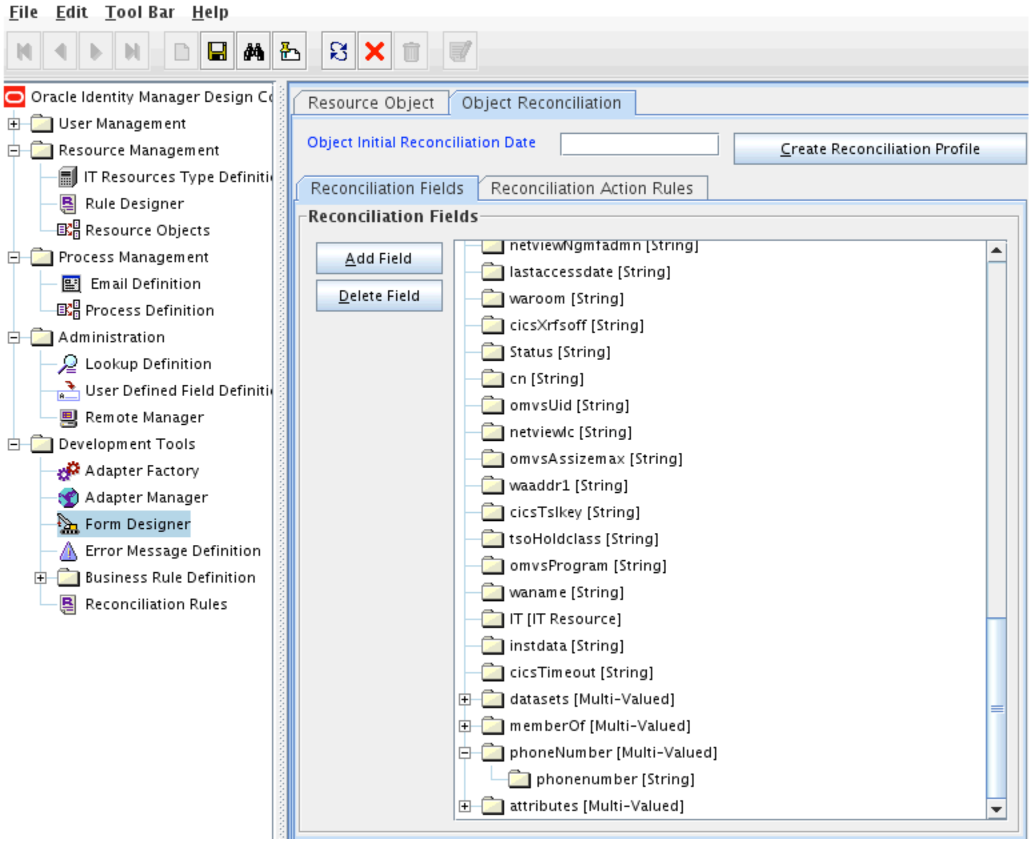
   f. Right-click the newly created field and select **Define Property Fields.**

   g. In the Add Reconciliation Fields dialog box, enter the details of the newly created field.

      For example, enter `phonenumber` in the Field Name field and select **String** from the Field Type list.

   h. Click **Save** and then close the dialog box. Figure 6-3 shows the new reconciliation field added in the resource object.

Figure 6-3 New Reconciliation Field Added in the resource Object



i. Click **Create Reconciliation Profile.** This copies changes made to the resource object into MDS.

5. Create an entry for the field in the `AtMap.Racf` lookup definition, as follows:

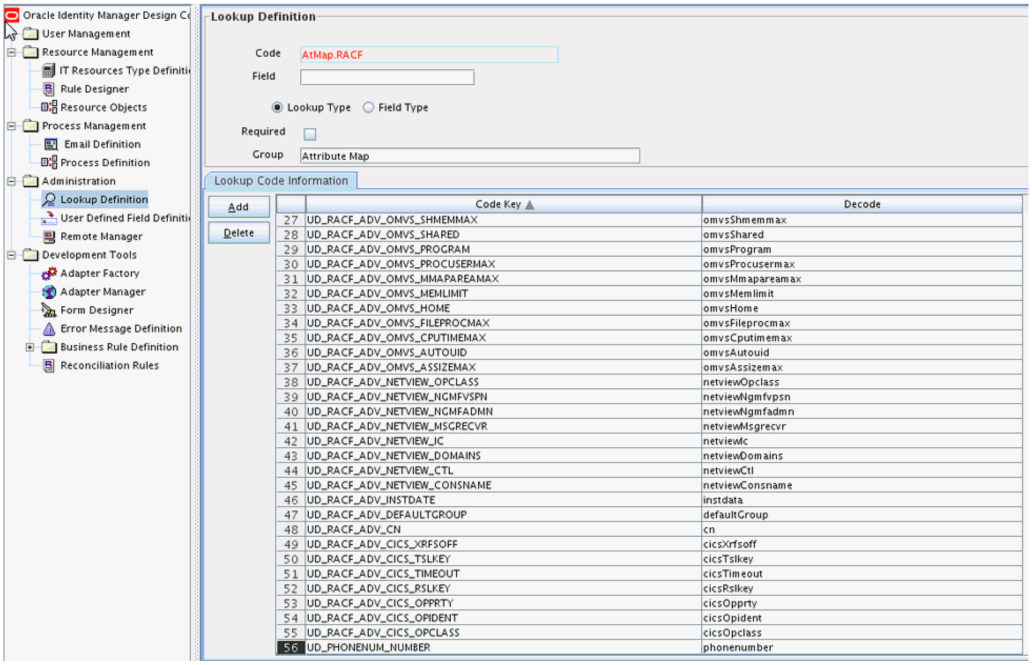
   a. Expand **Administration** and then double-click **Lookup Definition.**

   b. Search for the **AtMap.Racf** lookup definition.

   c. Click **Add** and enter the Code Key and Decode values for the field. The Code Key value is the name of the process form field that you created for the multivalued custom field in Step 3.d. The Decode value is the name of the target system field.

      For example, enter `UD_PHONENUM_PHONENUMBER` in the Code Key field and then enter `phonenumber` in the Decode field. Figure 6-4 shows the lookup code added to the lookup definition.

Figure 6-4 Entry Added in the Lookup Definition



Code: AtMap.RACF

Field:

● Lookup Type ○ Field Type

Required □

Group: Attribute Map

	Code Key	Decode
27	UD_RACF_ADV_OMVS_SHMEMMAX	omvsShmemmax
28	UD_RACF_ADV_OMVS_SHARED	omvsShared
29	UD_RACF_ADV_OMVS_PROGRAM	omvsProgram
30	UD_RACF_ADV_OMVS_PROCUSERMAX	omvsProcusermax
31	UD_RACF_ADV_OMVS_MMAPAREAMAX	omvsMmapareamax
32	UD_RACF_ADV_OMVS_MEMLIMIT	omvsMemlimit
33	UD_RACF_ADV_OMVS_HOME	omvsHome
34	UD_RACF_ADV_OMVS_FILEPROCMAX	omvsFileprocmax
35	UD_RACF_ADV_OMVS_CPUTIMEMAX	omvsCputimemax
36	UD_RACF_ADV_OMVS_AUTOUID	omvsAutouid
37	UD_RACF_ADV_OMVS_ASSIZEMAX	omvsAssizemax
38	UD_RACF_ADV_NETVIEW_OPCLASS	netviewOpclass
39	UD_RACF_ADV_NETVIEW_NGMFVSPN	netviewNgmfvpsn
40	UD_RACF_ADV_NETVIEW_NGMFADMN	netviewNgmfadmn
41	UD_RACF_ADV_NETVIEW_MSGRECVR	netviewMsgrecvr
42	UD_RACF_ADV_NETVIEW_IC	netviewIc
43	UD_RACF_ADV_NETVIEW_DOMAINS	netviewDomains
44	UD_RACF_ADV_NETVIEW_CTL	netviewCtl
45	UD_RACF_ADV_NETVIEW_CONSNAME	netviewConsname
46	UD_RACF_ADV_INSTDATE	instdata
47	UD_RACF_ADV_DEFAULTGROUP	defaultGroup
48	UD_RACF_ADV_CN	cn
49	UD_RACF_ADV_CICS_XRFSOFF	cicsXrfsoff
50	UD_RACF_ADV_CICS_TSLKEY	cicsTslkey
51	UD_RACF_ADV_CICS_TIMEOUT	cicsTimeout
52	UD_RACF_ADV_CICS_RSLKEY	cicsRslkey
53	UD_RACF_ADV_CICS_OPPRTY	cicsOpprty
54	UD_RACF_ADV_CICS_OPIDENT	cicsOpident
55	UD_RACF_ADV_CICS_OPCLASS	cicsOpclass
56	UD_PHONENUM_NUMBER	phonenumber

d. Click **Save.**

6. Create a reconciliation field mapping for the new multivalued field as follows:

   a. Expand **Process Management** and then double-click **Process Definition.**

   b. Search for and open the **OIMRacfProvisioningProcess** process definition.

   c. On the Reconciliation Field Mappings tab of the provisioning process, click **Add Table Map.**

   d. In the Add Reconciliation Table Mapping dialog box, select the **field name** and table name from the list, click **Save,** and then close the dialog box.

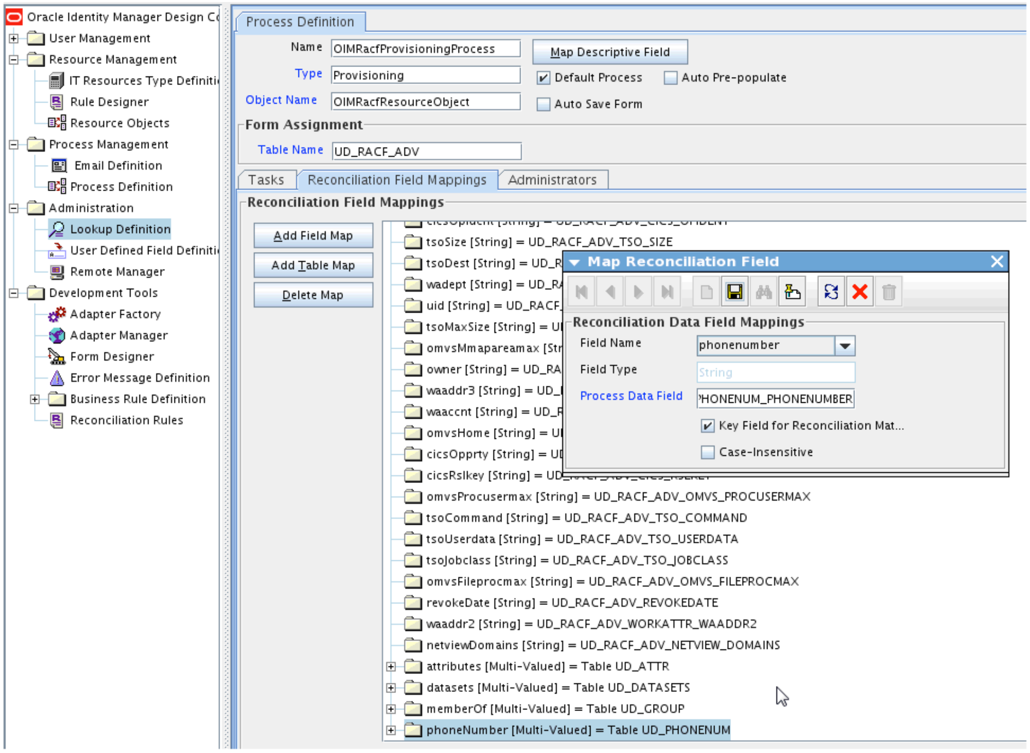
   e. Right-click the newly created field and select **Define Property Field Map.**

   f. In the Field Name field, select the value for the field that you want to add.

   g. Double-click the **Process Data** field, and then select **UD_PHONENUM_PHONENUMBER.**

   h. Select **Key Field** for Reconciliation Field Matching and click **Save.** Figure 6-5 shows the new reconciliation field mapped to a process data field in the process definition.

Figure 6-5 New Reconciliation Field Mapped to a Process Data Field



Name: OIMRacfProvisioningProcess
Type: Provisioning
Object Name: OIMRacfResourceObject
☑ Default Process
☐ Auto Pre-populate
☐ Auto Save Form

Form Assignment
Table Name: UD_RACF_ADV

Reconciliation Field Mappings

```
cicsOpident [String] = UD_RACF_ADV_CICS_OPIDENT
tsoSize [String] = UD_RACF_ADV_TSO_SIZE
tsoDest [String] = UD_R
wadept [String] = UD_R
uid [String] = UD_RACF
tsoMaxSize [String] = U
omvsMmapareamax [Str
owner [String] = UD_RA
waaddr3 [String] = UD_
waaccnt [String] = UD_
omvsHome [String] = U
cicsOpprty [String] = U
cicsRslkey [String] = UD_RACF_ADV_CICS_RSLKEY
omvsProcusermax [String] = UD_RACF_ADV_OMVS_PROCUSERMAX
tsoCommand [String] = UD_RACF_ADV_TSO_COMMAND
tsoUserdata [String] = UD_RACF_ADV_TSO_USERDATA
tsoJobclass [String] = UD_RACF_ADV_TSO_JOBCLASS
omvsFileprocmax [String] = UD_RACF_ADV_OMVS_FILEPROCMAX
revokeDate [String] = UD_RACF_ADV_REVOKEDATE
waaddr2 [String] = UD_RACF_ADV_WORKATTR_WAADDR2
netviewDomains [String] = UD_RACF_ADV_NETVIEW_DOMAINS
attributes [Multi-Valued] = Table UD_ATTR
datasets [Multi-Valued] = Table UD_DATASETS
memberOf [Multi-Valued] = Table UD_GROUP
phoneNumber [Multi-Valued] = Table UD_PHONENUM
```

Map Reconciliation Field — Reconciliation Data Field Mappings
Field Name: phonenumber
Field Type: String
Process Data Field: 'HONENUM_PHONENUMBER
☑ Key Field for Reconciliation Mat...
☐ Case-Insensitive

6.3 Adding Custom Fields for Provisioning for IBM RACF Advanced Connector

By default, the user attributes for target resource reconciliation and provisioning are mapped for provisioning between Oracle Identity Manager and the target system. If required, you can map additional attributes for provisioning.

To add a new attribute for provisioning:

1. Log in to the Oracle Identity Manager Design Console.
2. Add the new attribute on the process form as follows:

   If you have added the field on the process form by performing Step 4 of Adding Custom Fields to Oracle Identity Manager, then you need not add the field again. If you have not added the field, then:

   a. Expand **Development Tools.**

   b. Double-click **Form Designer.**

   c. Search for and open the **UD_RACF_ADV** process form.

   d. Click **Create New Version,** and then click **Add.**

   e. Enter the details of the attribute.

      For example, if you are adding the Description field, enter **UD_RACF_ADV_DESCRIPTION** in the Name field, and then enter the rest of the details of this field.

   f. Click **Save** and then click **Make Version Active.**

3. To enable update of the attribute during provisioning operations, create a process task as follows:

   a. Expand **Process Management,** and double-click **Process Definition.**

   b. Search for and open the **OIMRacfProvisioningProcess** process definition.

   c. Click **Add.**

   d. On the General tab of the Creating New Task dialog box, enter a name and description for the task and then select the following:

      Conditional

      Required for Completion

      Disable Manual Insert

      Allow Cancellation while Pending

      Allow Multiple Instances

   e. Click **Save.**

   f. Go to the Integration tab and click **Add.**

   g. In the Handler Selection dialog box, select **Adapter**, click **adpMODIFYUSER**, and then click the Save icon.

      The list of adapter variables is displayed on the Integration tab.

   h. To create the mapping for the first adapter variable:

      Double-click the number of the first row.

      In the Edit Data Mapping for Variable dialog box, enter the following values:

      Variable Name: `Adapter return value`

      Data Type: `Object`

      Map To: `Response code`

      Click the Save icon.

   i. To create mappings for the remaining adapter variables, use the data given in the following table:

Table 6-1 Values for the Variables, Map To, Qualifier, and Literal Value Lists for Each Variable

Variable Number	Variable Name	Map To	Qualifier
Second	idfResource	Process Data	LDAP_SERVER
Third	uid	Process Data	LoginId
Fourth	attrName	String Literal	Enter the LDAP attribute name in the Literal Value field. Example: `description`
Fifth	attrValue	Process Data	Select the process form field from the drop-down list. Example: DESCRIPTION

   j. On the Responses task, click **Add** to add at least the SUCCESS response code, with status C. This ensures that if the custom task is successfully run, then the status of task is displayed as Completed in Oracle Identity Manager.

k. Click the Save icon in the Editing Task dialog box, and then close the dialog box.

l. Click the Save icon to save changes to the process definition.

> **Note:**
>
> To enable Password Interval provisioning:
>
> - Use literal attrName "pwdInterval" for the modifyUser task. Value=0 (Note a value of 0 will set the command to NOINTERVAL).
> - Use literal attrName "pwdInterval" for the modifyUser task. Value=1 through nnn, where nnn is system accepted value range for INTERVAL (1) through INTERVAL (nnn).

4. Create a new UI form and attach it to the application instance to make this new attribute visible. See Creating a New UI Form and Section Updating an Existing Application Instance with a New Form for the procedures.

6.4 Removing Attributes Mapped for Target Resource Reconciliation

The SingleValueAttributes and MultiValuedAttributes attributes contain the list of target system attributes that are mapped for scheduled task reconciliation. These attributes are found in the RACF Reconcile All Users and RACF Reconcile All LDAP Users scheduled tasks.

If you want to remove an attribute mapped for scheduled task reconciliation, then remove it from the SingleValueAttributes or MultiValuedAttributes attributes.

6.5 Using the Provisioning Agent to Run IBM z/OS Batch Jobs

You can use the Provisioning Agent to run IBM z/OS batch jobs after provisioning operations. This feature provides an interface to the batch environment of IBM z/OS. For example, a CLIST script written in IBM REXX can be called through the standard TSO JCL. When it is called, the CLIST can perform user functions such as calling IBM DB2 UDB for database table updates, calling user programs to handle file updates, and generating reports.

To configure the Provisioning Agent to run IBM z/OS batch jobs:

1. Open the Provisioning Agent control file in a text editor.
2. In this file, create entries in the following format:

```
C=RACF_COMMAND,M=MEMBER_NAME,L=LIBRARY_NAME
P=USERID(Y),NAME(Y),CSDATA(003)
```

If you want to perform special post-processing, then a new feature has been added to only one parameter of the control file. The following is the definition for the new feature:

```
C=DELUSER,M=member-name,L=library_name,DEL=Y  or  DEL=N
DEL=Y  --  execute Rexx clist or z/OS job stream in library L=, M= and Perform the
actual deluser via RACF
DEL=N  --  execute Rexx clist or z/OS job stream in library L=,M= and DO NOT issue
the deluser to RACF
```

In the first line:

- *RACF_COMMAND* can be ADDUSER, ALTUSER, DELUSER, CONNECT, or REMOVE.
- *MEMBER_NAME* is the name of the IBM z/OS PDS that is submitted for execution in the IBM z/OS batch environment.
- *LIBRARY_NAME* is the name of the IBM z/OS PDS library name that contains the member specified by `MEMBER_NAME`.

The output of the submitted job is not sent back to the Provisioning Agent of the LDAP Gateway. You must take steps to ensure that the required action is taken based on the status of the operation. For example:

```
C=ADDUSER,M=ABCD,L=PDS.LIBRARY.ONE
P=USERID(Y),NAME(N)
```

The Provisioning Agent fetches the RACF user ID and passes it as a parameter to a REXX clist. The REXX clist must be set up to support parameters or arguments as shown in this example:

```
/* rexx */
Arg p1
```

Here, p1 is the RACF user ID and it can be used in the REXX clist. The same applies for NAME. If NAME(Y) and USERID(Y) are used, then the REXX clist can be similar to the following:

```
/* rexx */
Arg p1 p2
```

Here, p1 is the RACF user ID and p2 is the name.

If USERID(Y),NAME(N) is used, then only the user ID is passed. The csdata field can also be passed. The following example shows how to create and pass this field:

a. Define a csdata segment. See the *IBM RACF System Administrator's Guide* for information about the procedure.

b. To populate a CSDATA segment with one field:

   ```
   Altuser IDF004 CSDATA(EMPSER(100100))
   lu idf004 csdata noracf
   USER=IDF004
   CSDATA INFORMATION
    ------------------
   EMPLOYEE SERIAL= 0000100100
   ```

c. To populate a CSDATA segment with multiple fields:

   ```
   Altuser idf004 csdata(address('99 Main St, Anywhere, NJ, 08022')
   Phone(555-555-5555))
   lu idf004 csdata noracf
   USER=IDF004
   CSDATA INFORMATION
   ------------------
   EMPLOYEE SERIAL= 0000100100
   HOME ADDRESS = 99 Main St, Anywhere, NJ, 08022
   HOME PHONE = 555-555-5555
   For example:
   C=ADDUSER,M=ABCD,L=PDS.LIBRARY.ONE
   P=USERID(Y),NAME(N),CSDATA(001)
   ```

   The Provisioning Agent fetches the RACF user ID and passes it and the EMPLOYEE SERIAL csdata field to a REXX clist. This format has been changed and on CSDATA, the number of CSDATA fields need to be passed. The passed fields including userID, name and CSDATA cannot exceed 80 bytes. A CSDATA(001) will pass the first CSDATA field defined.

> **Note:**
>
> A hyphen must be added between the two names in this example and the length must be provided.

The REXX clist must be set up to support parameters or arguments as shown in the following example:

```
/* rexx */
Arg p1 p2
```

Here, p1 is the RACF user ID and p2 is Employee-Serial.

> **Note:**
>
> In this release of the Provisioning Agent, there is an 80-byte limit on the size of the field value that is passed. For example, if the user ID, name, and Employee-Serial are together over 80 bytes, one or two of these values must be removed so that the 80-byte limit is not exceeded.

3. Save and close the file.

The following sequence of steps takes place after a provisioning operation:

1. The Provisioning Agent opens the control file and reads the association between provisioning functions and the members specified in the file.
2. If there is an entry for the provisioning operation that was performed, then the corresponding member is submitted to the IBM z/OS batch environment.
   For example, suppose you had added the following entry in the control file:

   ```
   C=ALTUSER,M=MY_MEMBER,L=MY_LIBRARY
   ```

   At the end of a Modify User provisioning operation on the target system, the Provisioning Agent runs the MY_MEMBER member. This member performs the required operation on IBM z/OS.

6.6 Configuring the Connector for Provisioning to Multiple Installations of the Target System

You can configure the connector for multiple installations of the target system. You can also configure the connector for a scenario in which multiple logical partitions (LPARs), which are not associated with the first LPAR, are configured in the target system.

For each installation of the target system, you create an IT resource and configure an additional instance of the LDAP Gateway.

To configure the connector for the second installation of the target system:

> **Note:**
>
> Perform the same procedure for all installations of the target system.

1. Create an IT resource based on the OIMLDAPGatewayResourceType IT resource type.

   See Configuring the IT Resource for information about the parameters of the IT resource.
2. Copy the current *LDAP_INSTALL_DIR* directory, including all the subdirectories, to a new location on the Oracle Identity Manager computer.

   > **Note:**
   >
   > In the remaining steps of this procedure, *LDAP_INSTALL_DIR* refers to the newly copied directory.
3. Extract the contents of the *LDAP_INSTALL_DIR*/dist/idfserver.jar file.
4. In the beans.xml file, change the value of the port in the <property name="port" value="xxxx"/> line to specify a port that is different from the port used for the first instance of the LDAP Gateway. The default port number is shown in the following example:

   ```
   <bean id="listener" class="com.identityforge.idfserver.nio.Listener">
   <constructor-arg><ref bean="bus"/></constructor-arg>
   <property name="admin"><value>false</value></property>
   <property name="config"><value>../conf/listener.xml</value></property>
   <property name="port" value="5389"/>
   </bean>
   ```

   When you change the port number, you must make the same change in the value of the idfServerPort parameter of the IT resource that you create by performing Step 1.
5. Save and close the beans.xml file.
6. Open the *LDAP_INSTALL_DIR*/conf/racf.properties file and set values for the following parameters:
   - _host_= Enter the IP address or host name of the mainframe.
   - _port_= Enter the port number for the second instance of the Provisioning agent.
   - _agentPort_= Enter the port number for the second instance of the Reconciliation agent.

   > **Note:**
   >
   > The value of the _agentPort_ parameter must not be the same as that of the first instance if a second LPAR, which is not associated with the first LPAR, is configured in the target system. This value can be the same as the value of the idfServerPort parameter if you have two mainframe servers with IBM RACF running on each server.
7. Save and close the racf.properties file.
8. In a Linux or Solaris environment, if there are not enough socket file descriptors to open up all the ports needed for the server, then:

a. In a text editor, open the run script from the *LDAP_INSTALL_DIR*/bin directory.

b. Add the following line in the file:

```
-Djava.nio.channels.spi.SelectorProvider=sun.nio.ch.PollSelectorProvider
```

c. Save and close the file.

> **Note:**
>
> When you use Identity Self Service to perform provisioning, you can specify the IT resource corresponding to the IBM RACF installation to which you want to provision the user.

6.7 Customizing Log File Locations

The name and log location of the main LDAP gateway log file (idfserver.log) and the EXTRACT XML error log file (idf.xml.error.log) can be modified by adding additional arguments to the LDAP gateway server STARTUP command. These arguments are optional, and you can include one, both, or neither in the STARTUP command.

1. In a text editor, open the run script from the *`LDAP_INSTALL_DIR`*`/bin` directory. This run script is used to start and stop the LDAP gateway.
   - If using a Windows system, open the run.bat file.
   - If using a UNIX system, open the run.sh file.
2. Add the arguments to the start command, located at the end of the run script:
   - Add the arguments after the `-cp %CLASSPATH%` argument.
   - To modify the idfserver.log path, use the `-Didf.logpath=` argument.
   - To modify the idf.xml.error.log path, use the `-Didf.xmllogpath=` argument.

   In the following example, the start command will set the idfserver.log path to `C:/logs/ldap/idfserver.log` and the idf.xml.error.log path to `C:/logs/errors/idf.xml.error.log`:

   ```
   %JAVACMD% %DEBUG% %JVM_OPTS% %SECURE% -cp %CLASSPATH% -Didf.logpath="c:/logs/ldap/
   idfserver.log" -Didf.xmllogpath="c:/logs/errors/idf.xml.error.log" -
   Djava.library.path=%HOME%/lib com.identityforge.idfserver.Main %1 %2 %3 %4 %5 %6 %7
   %8 %9
   ```

6.8 LDAP Reconciliation Supported Queries

User Reconciliation Queries

- All User DNs and "uid" attribute
  - baseDn= ou=People,dc=racfxxx,dc=com
  - filter= (objectclass=*)
- Single User Search for all data
  - baseDn=ou=People,dc=racfxxx,dc=com

  - filter= (uid=idxxx)

Group Reconciliation Queries

- All Group DNs and "uid" attribute
  - baseDn= ou=Groups,dc=racfxxx,dc=com
  - filter= (objectclass=*)
- Single Group Search for all data
  - baseDn=ou=Groups,dc=racfxxx,dc=com
  - filter= (cn=idxxx)

Dataset Profiles for a given USER (uid) Reconciliation Queries

Dataset Profiles returned for a user

- baseDn= ou=Datasets,dc=racfxxx,dc=com
- filter= (uniqueMember=uid=idxxx,ou=People,dc=racfxxx,dc=com)i OR
- Filter= (uid=idxxx)

User-Defined Resources Reconciliation Queries

- Retrieve All User-Defined Resources: SEARCH CLASS (type)
  - baseDn= ou=Resources,dc=racfxxx,dc=com
  - Filter= (resourceType="YOUR CLASS TYPE")
    This returns all LDAP DN entries and each entry will contain the Resource ID via the 'cn' LDAP attribute.
- Retrieve Single User-Defined Resource: RLIST (cn) ALL
  - baseDn=ou=Resources,dc=racfxxx,dc=com
  - Filter= (cn=classID)

6.9 Handling Pioneer Error Messaging Exceptions in the Gateway

The error handling routines let you configure what error messages to look for when deciding that a request sent to Pioneer has succeeded or failed. Use these instructions to configure error handling.

Enable or Disable the Ability to Examine the Pioneer SAF Code

Some commands will return SAF or RACF codes whenever a command fails.

To enable the ability to automatically throw an error whenever codes greater than 0 are returned, add the `check-return-codes` property to the racf.properties file (created in Setting Connection Properties) and set its value to `yes`.

> **Note:**
>
> Warning codes may also show up as codes greater than 0 depending on the type of mainframe environment that you are using. Ensure to check for false positives with testing before determining whether this is an appropriate capability to turn on before deploying to a production environment.

Configuring Custom Error Messages

Many commands will require parsing out the return value looking for error messages. The error handling has been expanded to include a configuration file that allows for extending the set of error messages you might encounter.

Each error message which is being searched, is defined as a regex signature.

The IBM RACF Advanced connector comes with a default signatures file, errorMsgSignatures.xml, that you can extract from within the *`LDAP_INSTALL_DIR`*`/dist/backends/racf-connector.jar` compilation file. The errorMsgSignatures.xml file is located in the `com/identityforge/idfserver/backend/racf/repository/` directory of the `racf-connector.jar` compilation file.

You can add, overwrite, or disable the defaults in favor of custom messages.

To do so, in the *`LDAP_INSTALL_DIR`*`/conf` directory, create a new XML file representing the messages to add, replace, or disable. For example, create a new XML file *`LDAP_INSTALL_DIR`*`/conf/custom-racf-error-sig-file.xml` and add your custom messages. Then, in the *`LDAP_INSTALL_DIR`*`/conf/racf.properties` file, add a reference to the newly created XML file by setting a value for the `errormsg-sig-file` property. For example:

```
errormsg-sig-file=../conf/custom-racf-error-sig-file.xml
```

Restart the LDAP gateway for the changes to take effect. At runtime, the contents of the custom signature file are merged into the default signatures file and the overrides or additions will be applied.

The following are examples of custom signatures:

Example 1: Suppose you create a new XML file *`LDAP_INSTALL_DIR`*`/conf/custom-racf-error-sig-file.xml` in the *`LDAP_INSTALL_DIR`*`/conf` directory with the following entries:

```
<?xml version="1.0" encoding="utf-8"?>
<Signatures>
    <Signature id="custom1" regex="^C4R541E .*" enabled="yes"/>
    <Signature id="custom2" regex="^ICH02005I .*" enabled="yes"/>
    <Signature id="custom3" regex="^IKJ56701I .*" enabled="yes"/>
</Signatures>
```

In this example, the first signature looks for `C4R541E` located at the beginning of the returned message from Pioneer. If found, it would get flagged as an error and the message returned.

The second signature looks for `ICH02005I` located at the beginning of the returned message from Pioneer. If found, it would get flagged as an error and the message returned. Modify as needed for example, signature 3 regex="^IKJ56701I .* to indicate. If found, it would get flagged as an error and the message returned.

In the preceding example, the `enabled="yes"` entry implies that the messages defined in the regex patterns must not be considered as errors.

Example 2: Suppose you create a new XML file *LDAP_INSTALL_DIR*`/conf/custom-rcaf-error-sig-file.xml` in the *LDAP_INSTALL_DIR*`/conf` directory with the following entries:

```
<?xml version="1.0" encoding="utf-8"?>
<Signatures>
    <Signature id="custom1" regex="^ICH\d{5}I .*" enabled="yes">
        <Exception regex="^ICH01432I .*"/>
        <Exception regex="^ICH05555I .*"/>
        <Exception regex="^ICH01024I .*"/>
    </Signature>
    <Signature id="custom2" regex=".*INVALID DEPARTMENT.*" enabled="yes"/>
    <Signature id="e2" enabled="no"/>
</Signatures>
```

In this example, the first signature looks for the `ICHxxxxxxI` pattern located at the beginning of the returned message from Pioneer. If found, it then examines the exceptions defined. If the message begins with `ICH01432I` or `ICH05555I`, then it is marked as a warning and ignored. Otherwise, it is flagged as an error and the message returned.

The second signature looks for `INVALID DEPARTMENT` to show up anywhere in the returned message. If found, then it is flagged as an error and the message returned.

The third signature is an example of disabling an existing default signature. All default signatures start with `e` in the `id` attribute followed by a number. By referencing the id, the default signature's regex, enablement flag, and or exceptions can be replaced with a custom override. The `enabled="yes"` entry implies that the messages defined in the regex patterns must not be considered as errors.

At any given point in time, you can locate and open the `errorMsgSignatures.xml` file to obtain the list of default signatures currently deployed.

Note:

Given that according to the IBM RACF manual, "I" type messages are technically classified as informational and not error related, you need to make sure that it truly is a failure on the mainframe rather than somccething whereby the account gets created and Oracle Identity Manager considers it failed. We explicitly called out this RACF code as a warning as that is what the original implementation was doing.

7 Troubleshooting the IBM RACF Advanced Connector

These are some helpful tips to assist in resolving problems that you may encounter while using the connector.

Table 7-1 Troubleshooting Tips

Problem Description	Solution
Oracle Identity Manager cannot establish a connection with the target system.	• Ensure that the mainframe is running. • Verify that the required ports are working. • Due to the nature of the Provisioning Agent, the LDAP Gateway must be started first, and then the mainframe JCL started task must be started. This is a requirement based on how TCP/IP operates. Check that the IP address of the server that hosts the LDAP Gateway is configured in the Reconciliation Agent JCL. • Read the LDAP Gateway logs to determine if messages are being sent and received. • Examine the Oracle Identity Manager configuration to verify that the IP address, admin ID, and admin password are correct. • Check with the mainframe platform manager to verify that the mainframe user account and password have not been changed.
The mainframe does not appear to respond.	• Check the connection information that you have provided in the IT resource and the acf2Connection.properties file. • Check the logs. If any of the mainframe JCL jobs have reached an abnormal end, then make the required corrections and rerun the jobs.
A particular use case does not work as expected.	Check for the use case event in the LDAP Gateway logs. Then check for the event in the specific log assigned to the connector: • If the event has not been recorded in either of these logs, then investigate the connection between Oracle Identity Manager and the LDAP Gateway. • If the event is in the log but the command has not had the intended change on a mainframe user profile, then check for configuration and connections between the LDAP Gateway and the mainframe. Verify that the message transport layer is working.
The LDAP Gateway fails and stops working	If this problem occurs, then the Reconciliation Agent stops sending messages to the LDAP Gateway. Instead, it stores them in the subpool cache. When this happens, restart the LDAP Gateway instance so that the Reconciliation Agent reads the subpool cache and resends the messages.

Table 7-1 (Cont.) Troubleshooting Tips

Problem Description	Solution
The LDAP Gateway is running. However, the Reconciliation Agent fails and stops working	If this problem occurs, then all events are sent to the subpool cache. If the mainframe fails, then all messages are written to the disk. When this happens, restart the Reconciliation Agent instance so that it reads messages from the disk or subpool cache and resends the messages.
Voyager unable to connect to the LDAP	1. Can the LDAP server be pinged? 2. Is the LDAP up? 3. Is the LDAP listening on the correct port? Must be what is defined on PORT= on Voyager. 4. Can the Server where the LDAP resides Ping Voyager?
Voyager abends: S306-30 or Pioneer abends: S306-30	Review all RACF definitions. This abend is a incorrect definition.
Voyager or Pioneer abends other than S306-30 and SB37, SD37 or SE37	Open an Oracle SR and send the Voyager/Pioneer STC logs.
LDAP cant connect to Pioneer	1. Verify the listening port is correct on Pioneer, must be PORT= 2. Can the LDAP server ping Pioneer? 3. Can Pioneer ping the Server?
ADDUSER,ALTUSER,ADDGROUP,DELUSER submitted by LDAP and it fails.	Fails with SAF RC=8, RACF RC = 8 Incorrect RACF definitions for Pioneer. Must have access to all irr.radmin.* functions.
No Data in Voyager subpool. No events coming to the LDAP	Verify the three exits are up by: "D PROG,EXIT" the command exit should be active, "IRREVX01"

8 Known Issues and Workarounds for the IBM RACF Advanced Connector

These are the known issues associated with this release of the connector.

Multi-Threaded Batched Reconciliation Encounters an Error

When more than one open batched reconciliation operation is created (that is, when multi-threaded batched reconciliation is invoked) for a particular job and resource object, the following error is encountered:

```
Internal Exception: java.sql.SQLException: ORA-01422: exact fetch returns more than
requested number of rows
```

As a workaround, open the reconciliation profile of the resource object and set the value of the batchSize attribute to

```
0
```

. By default, the attribute has a value of either -1 or higher. This approach would result in single event processing. If the error is already encountered, open the RECON_BATCHES table for the particular job and resource object. Of all the multiple reconciliation batches in the Initiated status, manually update all the batches except one to Completed status. Then, open the reconciliation profile of the resource object and set the value of the batchSize attribute to 0.

Note that the batchSize attribute to be updated is the reconciliation profile attribute and not a scheduled job parameter.

A

Files and Directories in the IBM RACF Advanced Connector Package

These are the files and directories on the connector installation package that comprise the IBM RACF Advanced connector.

Table A-1 Files and Directories in the Installation Package

Files in the Installation Package Directory	Description
configuration/RacfAdv.xml	This XML file contains configuration information that is used during connector installation.
etc/LDAP Gateway/IDF_LDAP_GATEWAY_v6.4.0.zip	This ZIP file contains the files required to deploy the LDAP Gateway.
etc/Provisioning and Reconciliation Connector/RACF-AGENTS-201905311134-6.0.0.zip	This ZIP file contains the files required to deploy the Reconciliation and Provisioning Agents on the mainframe.
lib/racf-provisioning-adapter.jar	This JAR file contains the code for the adapters that are used during connector provisioning operations. During connector installation, this file is copied to the Oracle Identity Manager database.
lib/racf-scheduled-tasks.jar	This JAR file contains the code for the connector's scheduled tasks that perform lookup population and full reconciliation. During connector installation, this file is copied to the Oracle Identity Manager database.
Files in the resources directory	Each of these resource bundles contains locale-specific information that is used by the connector. During connector installation, this file is copied to the Oracle Identity Manager database. **Note:** A **resource bundle** is a file containing localized versions of the text strings that include GUI element labels and messages.
xml/oimRacfAdvR2Connector.xml	This XML file contains definitions of the connector components, such as the IT resource and resource object. These objects are created in Oracle Identity Manager when you import the XML file.

B
APF-Authorized Libraries

APF stands for Authorized Program Facility. In a z/OS environment, APF is a facility that permits the identification of programs that are authorized to use restricted functions.

APF-authorized programs must reside in one of the following authorized libraries:

- SYS1.LINKLIB
- SYS1.SVCLIB
- SYS1.LPALIB
- Authorized libraries specified by your installation

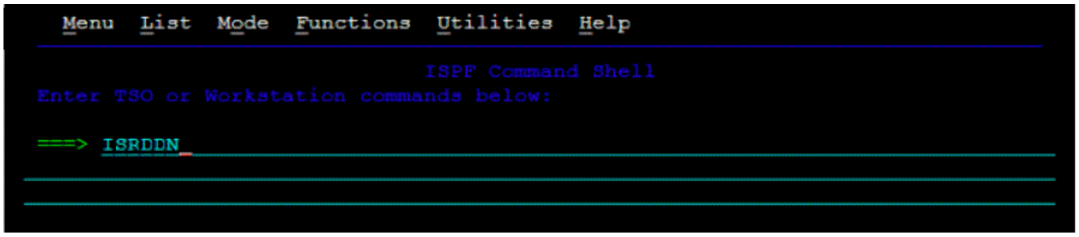
Authorized libraries are defined in an APF list, or in the link pack area (LPA). Any module in the LPA (pageable, modified, fixed, or dynamic) will be treated by the system as though it came from an APF-authorized library. The installation must ensure that it has properly protected SYS1.LPALIB and any other library that contributes modules to the link pack area to avoid system security and integrity exposures, just as it would protect any APF-authorized library.

APF also prevents authorized programs (supervisor state, APF-authorized, PSW key 0-7) from accessing a load module that is not in an APF-authorized library.

To find the datasets that have been APF authorized:

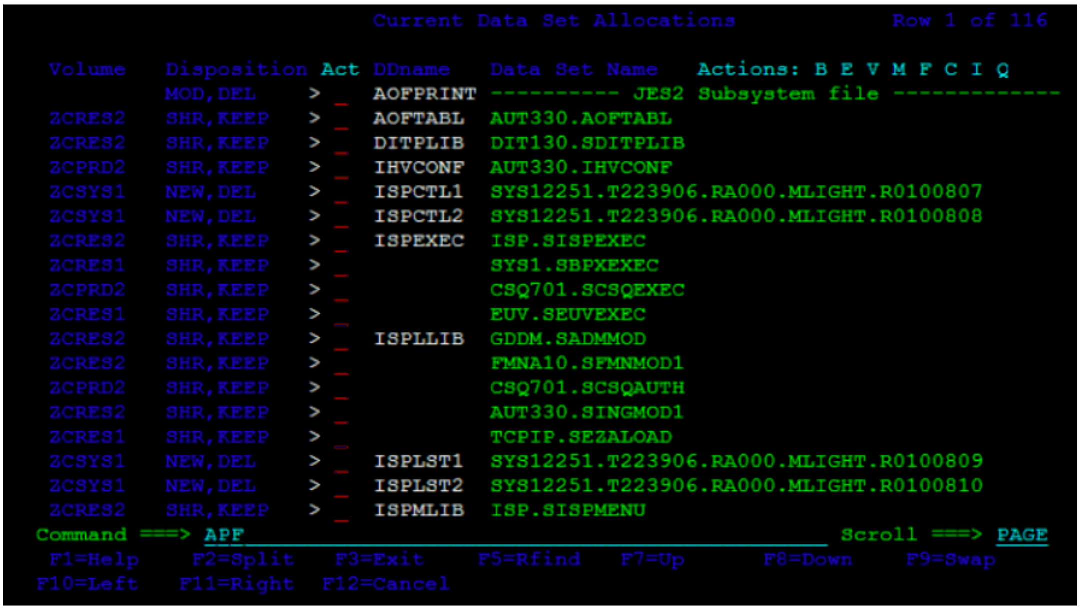
1. Type TSO ISRDDN in your ISPF session (some shops need just ISRDDN with no TSO prefix) and hit enter.
2. Type APF and hit enter. It'll bring up a list of all datasets that are APF authorized.

   Remember that, if you like to use an APF authorized dataset in a job STEPLIB, make sure all the datasets in the STEPLIB are APF authorized.



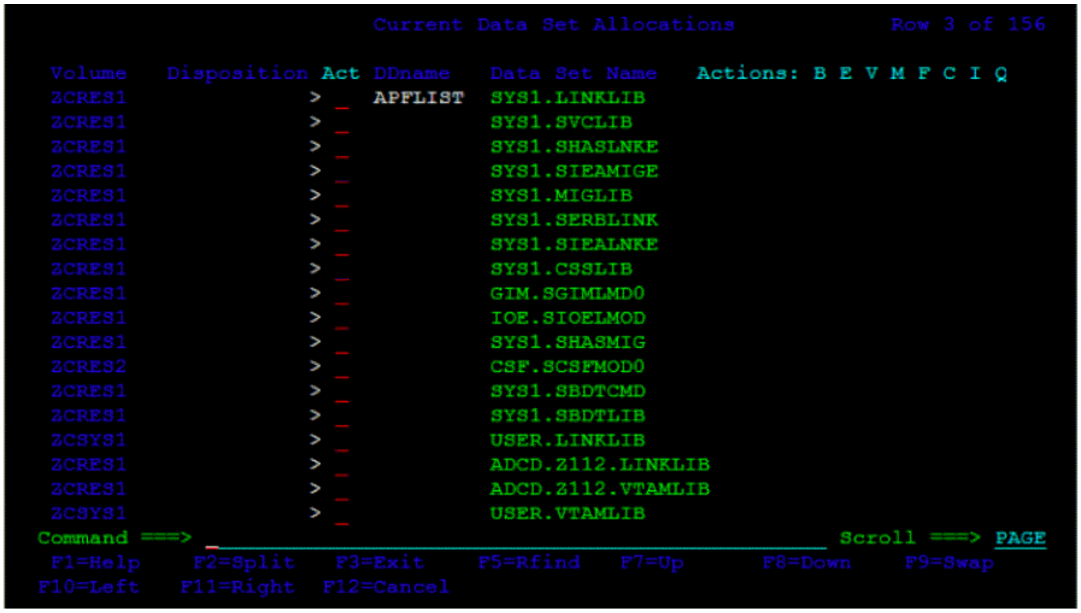
```
   Menu  List  Mode  Functions  Utilities  Help
                        ISPF Command Shell
Enter TSO or Workstation commands below:

===> ISRDDN_
```



```
                           Current Data Set Allocations            Row 1 of 116

 Volume   Disposition Act DDname   Data Set Name   Actions: B E V M F C I Q
          MOD,DEL      > _ AOFPRINT ---------- JES2 Subsystem file -------------
 ZCRES2   SHR,KEEP     > _ AOFTABL  AUT330.AOFTABL
 ZCRES2   SHR,KEEP     > _ DITPLIB  DIT130.SDITPLIB
 ZCPRD2   SHR,KEEP     > _ IHVCONF  AUT330.IHVCONF
 ZCSYS1   NEW,DEL      > _ ISPCTL1  SYS12251.T223906.RA000.MLIGHT.R0100807
 ZCSYS1   NEW,DEL      > _ ISPCTL2  SYS12251.T223906.RA000.MLIGHT.R0100808
 ZCRES2   SHR,KEEP     > _ ISPEXEC  ISP.SISPEXEC
 ZCRES1   SHR,KEEP     > _          SYS1.SBPXEXEC
 ZCPRD2   SHR,KEEP     > _          CSQ701.SCSQEXEC
 ZCRES1   SHR,KEEP     > _          EUV.SEUVEXEC
 ZCRES2   SHR,KEEP     > _ ISPLLIB  GDDM.SADMMOD
 ZCRES2   SHR,KEEP     > _          FMNA10.SFMNMOD1
 ZCPRD2   SHR,KEEP     > _          CSQ701.SCSQAUTH
 ZCRES2   SHR,KEEP     > _          AUT330.SINGMOD1
 ZCRES1   SHR,KEEP     > _          TCPIP.SEZALOAD
 ZCSYS1   NEW,DEL      > _ ISPLST1  SYS12251.T223906.RA000.MLIGHT.R0100809
 ZCSYS1   NEW,DEL      > _ ISPLST2  SYS12251.T223906.RA000.MLIGHT.R0100810
 ZCRES2   SHR,KEEP     > _ ISPMLIB  ISP.SISPMENU
Command ===> APF                                           Scroll ===> PAGE
 F1=Help     F2=Split    F3=Exit     F5=Rfind    F7=Up       F8=Down     F9=Swap
F10=Left    F11=Right   F12=Cancel
```



```
                           Current Data Set Allocations            Row 3 of 156

 Volume   Disposition Act DDname   Data Set Name   Actions: B E V M F C I Q
 ZCRES1                > _ APFLIST  SYS1.LINKLIB
 ZCRES1                > _          SYS1.SVCLIB
 ZCRES1                > _          SYS1.SHASLNKE
 ZCRES1                > _          SYS1.SIEAMIGE
 ZCRES1                > _          SYS1.MIGLIB
 ZCRES1                > _          SYS1.SERBLINK
 ZCRES1                > _          SYS1.SIEALNKE
 ZCRES1                > _          SYS1.CSSLIB
 ZCRES1                > _          GIM.SGIMLMD0
 ZCRES1                > _          IOE.SIOELMOD
 ZCRES1                > _          SYS1.SHASMIG
 ZCRES2                > _          CSF.SCSFMOD0
 ZCRES1                > _          SYS1.SBDTCMD
 ZCRES1                > _          SYS1.SBDTLIB
 ZCSYS1                > _          USER.LINKLIB
 ZCRES1                > _          ADCD.Z112.LINKLIB
 ZCRES1                > _          ADCD.Z112.VTAMLIB
 ZCSYS1                > _          USER.VTAMLIB
Command ===> _                                             Scroll ===> PAGE
 F1=Help     F2=Split    F3=Exit     F5=Rfind    F7=Up       F8=Down     F9=Swap
F10=Left    F11=Right   F12=Cancel
```

C

Pioneer Datasets

Table C-1 shows the relationship between the steps in the LOADDSN member and the file contents that are loaded into Pioneer's datasets. In these example datasets, PIONEER is used for the High-Level qualifier for Pioneer files and VOYAGER is used for the High-Level qualifier for Voyager files. The HLQ will have to be changed to meet installation standards. Table C-1 shows the relationship between the steps in the LOADDSN member and the corresponding file contents.

Table C-1 Relationship between the Steps in the LOADDSN Member and the File Contents

Steps	File Contents

Step#3	//STEP3 EXEC PGM=IEBGENER
	//SYSUT1 DD DSN=IDF.PROD.JCLLIB(PSAMPLE),DISP=SHR
	//SYSUT2 DD DSN=PIONEER.CONTROL.FILE,DISP=SHR
	//SYSPRINT DD SYSOUT=*
	//SYSIN DD DUMMY

PSAMPLE	TCPN=TCPIP
	IPAD=0.0.0.0
	PORT=5697
	DEBUG=N
	ESIZE=16
	LPAR=ZPDT-112
	POST_PROC_ALIAS=T
	IDLEMSG=N
	DEBUGOUT=SYSOUT,CLASS(S)
	SPIN_CLASS=K
	AUDIT_LOG=YES

Step#4	//STEP4 EXEC PGM=IEBGENER
	//SYSUT1 DD DSN=IDF.PROD.JCLLIB(VSAMPLE),DISP=SHR
	//SYSUT2 DD DSN=VOYAGER.CONTROL.FILE,DISP=SHR
	//SYSPRINT DD SYSOUT=*
	//SYSIN DD DUMMY

Table C-1 (Cont.) Relationship between the Steps in the LOADDSN Member and the File Contents

Steps	File Contents

VSAMPLE	SUBPOOL_SIZE=7500K
	TCPN=TCPIP
	* IPAD=192.168.1.999
	IPAD=192.168.1.100
	* IPAD=RACF.LEGACYIDM.COM
	PORT=5097
	DEBUG=N
	ESIZE=16
	* DELAY=00
	* STARTDELAY=10
	* PRTNCODE=SHUTRC
	CSDATA=N
	VOYAGER_ID=TESTVGER
	CACHE_DELAY=000
	AUDIT_LOG=YES
	PIONEER_ID=START2
	EXTRACT=Y

D

Creating Custom Scheduled Tasks

The following sections provide information about Java classes that you can use to create scheduled tasks for user reconciliation and lookup field synchronization:

- Code for Searching All Users and All User Data
- Code for Searching All Groups and All Group Data
- Code for Searching All Datasets and All Dataset Data

See Managing Scheduled Tasks i*n Oracle Fusion Middleware Administering Oracle Identity Manager* and Developing Lookup Definitions, UDFs, and Remote Manager in *Oracle Fusion Middleware Developing and Customizing Applications for Oracle Identity Manager* for detailed information about creating scheduled tasks and adding lookup fields for provisioning operations.

D.1 Code for Searching All Users and All User Data

Use the following class to create a scheduled task for fetching user data from the target system:

```
public void testSearchAllUsers() {

try {


SearchControls ctls = new SearchControls();
// SET COUNT LIMIT to 0 for all users //
ctls.setCountLimit(5);


// Search for objects that have those matching attributes - (objectclass=*)
//or (objectclass=idforgperson) is supported
NamingEnumeration answer =
ctx.search("ou=People,dc=racf,dc=com", "(objectclass=idforgperson)", ctls );


while( answer.hasMoreElements() ) {
     SearchResult result = (SearchResult)answer.nextElement();
     Attributes as = result.getAttributes();

}

} catch( NamingException nException ) {
  System.out.println(nException.toString());
}
```

D.2 Code for Searching All Groups and All Group Data

Use the following class to create a scheduled task for fetching group data from the target system. This data can be used to synchronize a group lookup field.

```
public void testSearchAllGroups() {

try {


SearchControls ctls = new SearchControls();
// SET COUNT LIMIT to 0 for all users //
ctls.setCountLimit(0);


// Search for objects that have those matching attributes - (objectclass=*)
//or (objectclass=idforggroup) is supported
NamingEnumeration answer =
ctx.search("ou=Groups,dc=racf,dc=com", "(objectclass=idforggroup)", ctls );


while( answer.hasMoreElements() ) {
     SearchResult result = (SearchResult)answer.nextElement();
     Attributes as = result.getAttributes();

}

} catch( NamingException nException ) {
  System.out.println(nException.toString());
}
```

D.3 Code for Searching All Datasets and All Dataset Data

Use the following class to create a scheduled task for fetching dataset data from the target system. This data can be used to synchronize a dataset lookup field.

```
public void testSearchAllDatasets() {

try {


SearchControls ctls = new SearchControls();
// SET COUNT LIMIT to 0 for all users //
ctls.setCountLimit(5);


// Search for objects that have those matching attributes - (objectclass=*)
//or (objectclass=idforgdataset) is supported
NamingEnumeration answer =
ctx.search("ou=Datasets,dc=racf,dc=com", "(objectclass=idforgdataset)", ctls
);


while( answer.hasMoreElements() ) {
     SearchResult result = (SearchResult)answer.nextElement();
     Attributes as = result.getAttributes();

}

} catch( NamingException nException ) {
  System.out.println(nException.toString());
}
```

E

Voyager and Pioneer Control File Parameters

Table E-1 lists Voyager control file parameters and the corresponding descriptions.

Table E-1 Voyager Control File Parameters

Voyager Control File Parameters	Description
SUBPOOL_SIZE=	The size of the cache in Subpool 231(ECSA) Voyager will allocate for event messages created by installed exits. The values are from 0200K to 7500K. Before allocating verify that there is enough ECSA storage available. Sample value: `1000K`
TCPN=	The TCPIP STC name where Voyager is executing. This is required for socket allocations. Sample value: `TCPIP`
IPAD=	The LDAP IP address or hostname. The hostname can only be 40 characters long. Sample value: `10.10.10.10`
PORT=	The port the LDAP is listening on for Voyager messages. **Note:** This is the value _agentPort_ as specified in racf.properties. Sample value: `5790`
DEBUG=	Enter `Y` to turn on debugging and enter `N` to turn it off. If you enter Y, then the output is sent to DEBUGOUT. **Note:** If you set DEBUG=Y produces enormous log. It is advised not to use DEBUG=Y in production. Default value: `N`
ESIZE=	The value of this parameter must be set to 16 always. This never changes.
CSDATA=	Enter `Y` if the RACF database is supporting CSDATA fields. Otherwise, enter `N`. If EXTRACT=Y, then set the value of this parameter to `N`. Sample value: `N`

Table E-1 (Cont.) Voyager Control File Parameters

Voyager Control File Parameters	Description
VOYAGER_ID=	The ID defined to LDAP for Voyager, if several Voyagers are using the same RACF database, then they must all have the same VOYAGER_ID. This parameter is used for the following purposes: • For user documentation. • Comes in handy when one VOYAGER PER LPAR. • Gateway uses VOYAGER_ID= and some consideration is given to configure to exclude gateway events by VOYAGER_ID if needed. Sample value: `VOYID`
CACHE_DELAY=	The delay (in seconds) Voyager will use when issuing a write socket to the LDAP. This parameter is used in conjunction with communications from the Mainframe Connector Portion to Oracle Identity Manager Appserver Parent Product. Sample value: `005`
AUDIT_LOG=	Enter `YES` to turn on audit logging. Otherwise, enter `NO`. Sample value: `YES`
PIONEER_ID=	(Optional) Enter the RACF userID defined for Pioneer.When Voyager reads a subpool message and the issuer is this RACF userID, then Voyager will not send this message to the LDAP.If you do not specify a value for this parameter, no action takes place.
EXTRACT=	Enter Y to utilize a RACF Extract versus a RACF LISTUSER. Otherwise, enter `N`. This Parameter should be `Y` unless told otherwise by Oracle Support Representative. Sample value: `Y`
CONNECT_RETRY=	The number of times Voyager will attempt to reconnect. You can specify a value from `001` through `999`. A value of `999` indicates unlimited retries. Sample value: `009`
CONNECT_INTV=	The number of seconds between each reconnect attempt. The value can range from `01` through `99`. If CONNECT_RETRY=010 and CONNECT_INTV=10, then Voyager will retry the connection to the LDAP for 100 seconds. After the 100 seconds have elapsed, Voyager will shutdown. Sample value: `05`

Table E-1 (Cont.) Voyager Control File Parameters

Voyager Control File Parameters	Description
EBCDIC_COUNTRY_CODE	This parameter represents the EBCDIC Country Code page override. **Note:** Do not specify values for special reserved usage parameters unless directed by Oracle Support. These parameters are available only for specific custom usage, and their sample values are not available. These parameters must be used along with the gateway configuration property file (_mainframeCodePage_), which is available inside the racf.properties file.
EBCDIC_TILDE_CHR	This parameter represents the EBCDIC HEX value tilde character that indicates the end of data override. **Note:** Do not specify values for special reserved usage parameters unless directed by Oracle Support. These parameters are available only for specific custom usage, and their sample values are not available. These parameters must be used along with the gateway configuration property file (_mainframeCodePage_), which is available inside the racf.properties file.

Table E-2 lists Pioneer control file parameter and the corresponding descriptions.

Table E-2 Pioneer Control File Parameters

Pioneer Control File Parameter	Description
TCPN=	The TCPIP STC name where Voyager is executing. This is required for socket allocations. Sample value: `TCPIP`
IPAD=	This is the Reserved value. The value of this parameter must be set to `0.0.0.0` always.
PORT=	The port at which Pioneer is listening for LDAP messages. Sample value: `5709`
DEBUG=	Enter `Y` to turn on debugging and enter `N` to turn it off. If you enter `Y`, then the output is sent to DEBUGOUT. **Note:** Setting this flag to `Y` will create enormous log. Do not use it on production systems. Sample value: `N`
ESIZE=	The value of this parameter must be set to `16` always. This never changes.
LPAR=	Enter a 20-byte unique name for the LPAR of the Voyager system. Sample value: `zOS-2.2-TEST`

Table E-2 (Cont.) Pioneer Control File Parameters

Pioneer Control File Parameter	Description
POST_PROC_ALIAS=	If you set the value of this parameter to `T`, then Pioneer will honor all DEFINE/DELETE alias requests from the LDAP. If you set the value of this parameter to `F`, Pioneer will ignore all requests for DEFINE or DELETE aliases. Sample value: `T`
IDLEMSG=	Can take a value of either `Y` or `N`. If you set the value of this parameter to `Y`, then for every 60 minutes Pioneer is idle, it displays an IDLE message. Sample value: `N`
DEBUGOUT=	This parameter is valid only if you set DEBUG=Y. If you have set DEBUG=N, then this parameter is ignored. If the output must be sent to SYSOUT, then use the following format: `SYSOUT,CLASS(x)` In this format, `x` represents the JES2 output class desired, please use a lettered SYSOUT CLASS available at your installation versus "*". Sample value: `IBM z/oS JCL` **Note:** The usage of SYSOUT,CLASS(*) has been noted to cause IKJ56231I FILE DEBUGOUT NOT ALLOCATED, SYSTEM OR INSTALLATION ERROR+ when used with a Control card.
SPIN_CLASS=	The output SPIN class for DEBUGOUT when Pioneer shutdown or debugging is turned off through Operator command. Sample value: `x`
AUDIT_LOG=	If you set the value of this parameter to `YES`, then the Audit log is turned on and the output goes to AUDTLOG ddname of Pioneer. If you set the value of this parameter to `NO`, then auditing will not be in effect. Sample value: `YES`

Table E-2 (Cont.) Pioneer Control File Parameters

Pioneer Control File Parameter	Description
SECURE_ID=	YES is required. NO is not accepted and will fail. SECURE id can run in one of the three following modes: • SECURE_ID = YES, DEFAULT = YES This mode uses RACF userid IDFAGNT as the default userid. This must have 'SPECIAL' as a coded attribute. • SECURE_ID = YES, DEFAULT = NO, ENCRYPT = NO, ID = racfuserid This mode uses the RACF userid for RACF API calls and must have 'SPECIAL' coded on that RACF userid. • SECURE_ID = YES, DEFAULT = NO, ENCRYPT = YES This mode uses the RACF userid that was encrypted using the new IDFSECUT program. This encrypted RACF userid will be used for all RACF API calls. Sample value: `YES, DEFAULT=YES`
SMP\F=	A value of either N or Y is required. If you set the value of this parameter to N, then SMF recording is not turned off, but custom SMF TYPE 245 subtype 1 and 2 records are not created when the SECURE_ID is invoked. If you set the value of this parameter to Y, then every time the SECURE_ID is invoked, custom SMF TYPE 245 subtype 1 and 2 records are created. In addition, SMFPRMxx of z/OS SYS1.PARMLIB must be reviewed to verify that this SMF Type record will be written. Also, Pioneer must have the z/OS authority to write custom SMF type 245 entries. Sample value: N
EBCDIC_COUNTRY_CODE	This parameter represents the EBCDIC Country Code Page override.
EBCDIC_TILDE_CHR	This parameter represents the EBCDIC HEX value Tilde Character that indicates the end of data override.

F

Configuring RACF Starter User ID and Access for Voyager Agent and Pioneer Agent Started Tasks

Pioneer Started Task no longer supports or requires a RACF userid attribute 'SPECIAL'. A normal RACF userid as shown below can be used.

There are various modes that you can use. The modes and the required RACF definitions are shown below. Note that the normal RACF userid is italicized.

> **Note:**
>
> Depending on the requirement, select one of the **modes** between **1**, **2**, or **3**.

One of the following 3 modes can be used:

- **Mode:**

  ```
  SECURE_ID=YES,DEFAULT=YES
  ```

  This mode uses RACF userid IDFAGNT as the default userid. This must have SPECIAL as a coded attribute.

  Default Pioneer control file parameter is `SECURE_ID=YES,DEFAULT=YES`

  – ADDGROUP SECGRP
  – *ADDUSER PIONEER NAME(PIONEER) DFLTGRP(SECGRP) NOPASS NOPHRASE*
  – ADDUSER IDFAGNT NAME(DEFAULT-ID) DFLTGRP(SECGRP) NOPASS NOPHRASE SPECIAL
  – PW USER(PIONEER) NOINTERVAL
  – ALU PIONEER AUDITOR
    This is used for list type commands like LISTUSER, LISTGRP, and other similar commands.
  – RDEFINE FACILITY IDFADMIN.CMD UACC(NONE)
  – PERMIT IDFADMIN.CMD ID(PIONEER) ACCESS(READ)
  – CONNECT PIONEER GROUP(grpname)
    The grpname must be the same grpname used for FTPD. It must have a OMVS segment and a Permit for using BPX.DAEMON without which the Pioneer RACF Userid will fail as shown below:

    ```
    0090 IDMP006I - PIONEER DETECTS DEBUGGING IS ACTIVE
    0090 IDMP011I - PIONEER DETECTS CPUID 1006112064
    0090 IDMP012I - PIONEER DETECTS SYSPLEX SYSNAME ADCD113S
    0090 IDMP013I - PIONEER DETECTS LPARNAME AS SYST
    0090 IDMP014I - PIONEER DETECTS COUNTRY CODE OF US
    0090 IDMP009I - PIONEER DETECTS ENCRYPTION ENABLED
    ```

```
0090 IDMP016I - PIONEER APF LIBRARY IS GOOD
0281 ICH408I JOB(PIONEER ) STEP(PIONEER ) CL(PROCESS ) 251
0281 OMVS SEGMENT NOT DEFINED
0090 IDMP402I PIONEER HAS NO OPEN SOCKETS
0090 IDMP402I PIONEER DID NOT OPEN TCPIP API
0090 IDMP402I PIONEER IS ENDING DUE TO ERRORS
0090 IDMP402I PIONEER - REVIEW SYSLOG OR PARMOUT
0090 IDMP402I PIONEER ENDS RC= 100
0090 IEF404I PIONEER - ENDED - TIME=10.26.13
0281 $HASP395 PIONEER ENDED
0281 IEA989I SLIP TRAP ID=X33E MATCHED. JOBNAME=*UNAVAIL, ASID=0037.
```

- **Mode:**

  ```
  SECURE ID=YES,DEFAULT=NO,ENCRYPT=NO,ID=IDMSECU
  ```

  This mode uses the RACF userid for RACF API calls and must have 'SPECIAL' coded on that RACF userid.

  Using a user defined RACF secure id:

  Pioneer parameter is `SECURE ID=YES,DEFAULT=NO,ENCRYPT=NO,ID=IDMSECU`

  - ADDGROUP SECGRP
  - *ADDUSER PIONEER NAME(PIONEER) DFLTGRP(SECGRP) NOPASS NOPHRASE*
  - PW USER(PIONEER) NOINTERVAL
  - ALU PIONEER AUDITOR
    This is used for list type commands like LISTUSER, LISTGRP, and other similar commands.
  - ADDUSER IDMSECU NAME('SECURE-ID') DFLTGRP(SECGRP) NOPASS NOPHRASE SPECIAL
  - RDEFINE FACILITY IDFADMIN.CMD UACC(NONE)
  - PERMIT IDFADMIN.CMD ID(PIONEER) ACCESS(READ)
  - See Pioneer CONNECT above

- **Mode:**

  ```
  SECURE_ID=YES,DEFAULT=NO,ENCRYPT=YES
  ```

  This mode uses the RACF userid that was encrypted using the new IDFSECUT program. This encrypted RACF userid will be used for all RACF API calls.

  Using a encrypted RACF userid:

  Pioneer parameter is `SECURE_ID=YES,DEFAULT=NO,ENCRYPT=YES`

  - ADDGROUP SECGRP
  - *ADDUSER PIONEER NAME(PIONEER) DFLTGRP(SECGRP) NOPASS NOPHRASE*
  - PW USER(PIONEER) NOINTERVAL
  - ALU PIONEER AUDITOR
    This is used for list type commands like LISTUSER, LISTGRP, and other similar commands.
  - ADDUSER <your-secure-id-that was encrypted> NAME('SECURE-ID') DFLTGRP(SECGRP) NOPASS NOPHRASE SPECIAL
  - RDEFINE FACILITY IDFADMIN.CMD UACC(NONE)
  - PERMIT IDFADMIN.CMD ID(PIONEER) ACCESS(READ)

See Pioneer CONNECT above

You can encrypt and decrypt the RACF userid, and implement the SECUREID process. To do so, perform the following procedures:

- Procedure to encrypt the RACF userid:
  Execute IDFSECUT. In the sample below, JCL is supplied in the distribution JCLLIB. The 'DFLEOUT' ddname dataset must match the ddname//SECUREID of Pioneer. The member name of JCLLIB is 'SECUTLE' which is the encryption utility of JCL. Then, only the parameters are visible and the ID=XXXXX is the RACF userid that has to be encrypted.

  ```
  //IDFSECUT JOB SYSTEMS,MSGLEVEL=(1,1),
  //   MSGCLASS=X,CLASS=A,PRTY=8,
  //       NOTIFY=&SYSUID,REGION=4096K
  //* ID=XXXXX IS THE RACF USER THAT HAS SPECIAL ATRIIBUTES
  //* FOR USE WITH PIONEER
  //STEP1    EXEC PGM=IDFSECUT,PARM='ID=XXXXX,FUNC=ENCRYPT'
  //STEPLIB  DD   DSN=<YOURHLQ.PROD.LOADLIB,DISP=SHR
  //DFLEOUT  DD   DSN=<YOURHLQ>.SECUREID.FILE,DISP=SHR
  //LINEOUT  DD   SYSOUT=*
  //SYSPRINT DD   SYSOUT=*
  ```

- Procedure to decrypt the RACF userid:
  Execute IDFSECUT. In the sample below, JCL is supplied in the distribution JCLLIB. The 'DFLEOUT' ddname dataset must match he ddname//SECUREID of Pioneer. The member name of JCLLIB is 'SECUTLE' which is the encryption utility of JCL. The parameters are the only ones that are displayed.

  ```
  //IDFSECUT JOB SYSTEMS,MSGLEVEL=(1,1),
  //   MSGCLASS=X,CLASS=A,PRTY=8,
  //       NOTIFY=&SYSUID,REGION=4096K
  //* ID=NONE IS TO VERIFY WHAT RACF USER ID IS CONTAINED IN
  //* THE SECUREID FILE
  //STEP1    EXEC PGM=IDFSECUT,PARM='ID=NONE,FUNC=DECRYPT'
  //STEPLIB  DD   DSN=<YOURHLQ.PROD.LOADLIB,DISP=SHR
  //DFLEOUT  DD   DSN=<YOURHLQ>.SECUREID.FILE,DISP=SHR
  //LINEOUT  DD   SYSOUT=*
  //SYSPRINT DD   SYSOUT=*
  ```

- Procedure to implement the SECUREID process:
  - Select the RACF userid desired to perform the Pioneer RACF API calls to R_admin.
  - Define it to RACF as shown in Step 3.
  - Encrypt it using the IDFSECUT as shown in the above Step.
  - Start Pioneer.
    Pioneer reads the SECURE_ID file and stores the encrypted id.

    Pioneer also first receives the RACF command and accesses the RACF facility 'MYADMN.CMD'. If access is granted, Pioneer uses the encrypted id with which it decrypts all RACF calls.

The following steps are required to use all the modes as these are common for each mode.

Perform the following steps after you select the mode:

1. RACF Facility must be changed as mentioned below in order to start Pioneer:

   ```
   RDEF STARTED PIONEER.* UACC(NONE) OWNER(xxxxxxx)
   ```

```
RALT STARTED PIONEER.* AUDIT(FAILURES(READ))

RALT STARTED PIONEER.* STDATA(USER(PIONEER) GROUP(SYS1) PRIVILEGED(NO) TRACE(NO))
```

2. Pioneer (Other RACF definitions):

```
RDEFINE FACILITY IRR.RADMIN.* UACC(NONE)

PERMIT IRR.RADMIN CLASS(FACILITY) ID(<your-RACF-non-secure-id>) ACCESS(READ)

ADDSD '<yourhlq>.CONTROL.FILE' UACC(NONE)

PERMIT '<yourhlq>.CONTROL.FILE' ID(<your-RACF-non-secure-id>) ACCESS(READ)

ADDSD '<yourhlq>.REXXOUT.FILE' UACC(NONE)

PERMIT '<yourhlq>.REXXOUT.FILE' ID(<your-RACF-non-secure-id>)ACCESS(UPDATE)

ADDSD '<yourhlq>.RECON.FILE' UACC(NONE)

PERMIT '<yourhlq>.RECON.FILE' ID (<your-RACF-non-secure-id>)ACCESS(UPDATE)

ADDSD '<yourhlq>.RECON.LIBRARY' UACC(NONE)

PERMIT '<yourhlq>.RECON.LIBRARY' ID (<your-RACF-non-secure-id>)ACCESS(READ)

ADDSD '<yourhlq>.IMPORTU.FILE' UACC(NONE)

PERMIT '<yourhlq>.IMPORTU.FILE' ID (<your-RACF-non-secure-id>)ACCESS(UPDATE)

ADDSD '<yourhlq>.IMPORTG.FILE' UACC(NONE)

PERMIT '<yourhlq>.IMPORTG.FILE' ID (<your-RACF-non-secure-id>) ACCESS(UPDATE)

ADDSD '<yourhlq>.ALIAS.LSTOUT' UACC(NONE)

PERMIT '<yourhlq>.ALIAS.LSTOUT' ID(<your-RACF-non-secure-id>) ACCESS(UPDATE)

ADDSD '<yourhlq>.IDCAMS.CTL' UACC(NONE)

PERMIT '<yourhlq>.IDCAMS.CTL' ID (<your-RACF-non-secure-id>) ACCESS(UPDATE)
```

G

Customizing AES Encryption Key

Perform this procedure to configure and customize an AES encryption key.

1. In order to use your own key with the LDAP gateway, you must add it to the properties file for the particular mainframe connector that you are using. The property files used for the the IBM RACF connector is racf.properties. It will be a 32 character HEX key.

   Define a property called `_secretKeyValue_` to store the key you want to use.

   The value defined is the same in all the property files.

   For example, _secretKeyValue_=**52810283F6B4E0A5D82FDE935E23ED7C**

   To generate a custom secret key perform the following tasks on your platform:

   a. Windows
      Open a command prompt and run:

      ```
      java -cp <path of lib folder>\*; <path of idfserver.jar file>
      com.identityforge.idfserver.util.AESCipherUtil <16 char test>
      ```

      For example:

      ```
      java -cp "C:\8.1.2\ldapgateway8.0-oracle-v8.1.2\lib\*;C:
      \8.1.2\ldapgateway8.0-oracle-v8.1.2\dist\idfserver.jar"
      com.identityforge.idfserver.util.AESCipherUtil ABCD12344321TEST
      ```

   b. Linux
      Open a terminal session and run:

      ```
      java -cp <path of lib folder>\*; <path of idfserver.jar file>
      com.identityforge.idfserver.util.AESCipherUtil <16 char test>
      ```

      For example:

      ```
      java -cp /Users/ldapgateway/v8.1.2/lib/*:/Users/ldapgateway/v8.1.2/dist/
      idfserver.jar com.identityforge.idfserver.util.AESCipherUtil
      EFGH12344321TEST
      ```

   > **Note:**
   >
   > The LDAP Gateway will have to be restarted for the new key to take effect.

2. Once you have defined the key in the LDAP property file, you will need to set the key on the mainframe side.

   A MVS Job called KEYMODR will set the key on the mainframe side. It will ship with the distribution JCL files in the JCLLIB.xmi library as follows:

   ```
   //ADCDZZAP JOB ,SYSTEMS,CLASS=A,MSGCLASS=X,
   // MSGLEVEL=(1,1),REGION=4096K,TIME=1440,NOTIFY=&SYSUID
   //ZAPKEY EXEC PGM=AMASPZAP
   //SYSPRINT DD SYSOUT=*
   //SYSLIB DD DISP=SHR,DSN=MLIGHT.MY.LOAD
   //SYSIN DD *
   NAME IDFRINFO IDFRINFO
   ```

```
* VERIFY EYECATHER IS PRESENT
VER 0080 C9C4,C6D9,C9D5,C6D6 'IDFRINFO'
* SET KEYLEN = 100
REP 0088 0064
* SET 1ST 16 BYTES WITH YOUR KEY
REP 008A 7CC7,3006,074D,E87A,A647,2FC4,3BA4,5DB1
* SET 2ND 16 BYTES WITH ANYTHING (FOR FUTURE USE)
REP 009A D2D3,D4D5,D6D7,D8D9,E2E3,E4E5,E6E7,E8E9
* SET 3RD 16 BYTES WITH ANYTHING (FOR FUTURE USE)
REP 00AA F6F7,F8F9,F9F9,F9F9,F9F9,F9F9,F9F9,F9F9
* SET 4th 16 BYTES WITH THE DATE (2013082013200000)
REP 00BA F2F0,F1F3,F0F8,F2F0,F1F3,F3F0,F0F0,F0F0
//
```

3. To use the BATCH JCL, perform the following procedure:

   a. Change the job card to conform to the standards of your system.

   b. Change the below line to set the DSN where you have the linklib for the mainframe agent:

      ```
      //SYSLIB DD DISP=SHR,DSN=MLIGHT.MY.LOAD
      ```

   c. Change the below line to set your key value:
      * SET 1ST 16 BYTES WITH YOUR KEY

      REP 008A 7CC7,3006,074D,E87A,A647,2FC4,3BA4,5DB1

      Do not change the beginning of the line REP 008A. However, you can change the rest of the line to match your key. Use 4 characters at a time followed by a comma, as shown above.

   d. Change the below line to set the date for your key:
      * SET 4th 16 BYTES WITH THE DATE (2013082013200000)

      REP 00BA F2F0,F1F3,F0F8,F2F0,F1F3,F3F0,F0F0,F0F0

      Do not change the beginning of the line REP 008A. However, you can change the rest of the line to match the date you changed the key.

      Note:

      EBCDIC HEX values for the numbers 0 through 9 are used. They are F0 through F9.
      The format for the date is YYYYMMDDHHMMSSMM (Year Month Day Hour Minutes Seconds Miliseconds). This is optional, but it will help in identifying the key.

      After you have made the changes, you will need to submit the Job to set your changes.

      Additionally, note that Pioneer and Voyager will have to be restarted for the new key take effect.

      If AMASPZAP is not allowed, then follow the instructions mentioned below:

      The procedure to change the key is very similar to the directions for the KEYMODR jcl. The first line for KEYBYTES will be changed after which the fourth line for the key date change will have to be changed.

EBCDIC HEX values for the numbers 0 through 9 are used. They are F0 through F9. The format for the date is YYYYMMDDHHMMSSMM (Year Month Day Hour Minutes Seconds Miliseconds). This is optional, but it will help in identifying the key.

In addition to changing the Jobcard, you must change the following in IDFRINFO:

EYECATCH DC C'IDFRINFO'

INFOLEN DC H'100'

KEYBYTES DC X'D880D7614C07088BC2D51A1945FDB6B4': Ensure that you change this key.

DC X'D8D9E2E3E4E5E6E7E8E9F0F1F2F3F4F5': This is reserved for later use. Keep it as is.

DC X'F6F7F8F9F9F9F9F9F9F9F9F9F9F9F9F9': This is reserved for later use. Keep it as is.

DC X'F2F0F0F6F1F0F2F6F1F4F2F5F0F0F0F0': This is the date key was changed.

DC 86X'0'

END IDFRINFO

Once they have changed the Key and have assembled/linked IDFRINFO, then they will have to replace the IDFRINFO that we supply.

The LDAP gateway server settings must also be updated to use the new key. To configure the LDAP gateway, perform the following steps:

a. Stop the LDAP gateway server (if it is running).

b. Open the racf.properties file, located in the *LDAP_INSTALL_DIR*/conf directory.

c. Modify the value of the _secretKey_ property to match the new key.

d. Save and close the file.

e. Restart the LDAP gateway server.

H

Mainframe Language Environment Runtime Options

For the IBM RACF Advanced Connector you need to set the Mainframe Language Environment runtime options.

This appendix contains the following topics:

- Setting Runtime Options for IBM RACF
- Run Time Options, Defaults and Recommendations for IBM RACF

H.1 Setting Runtime Options for IBM RACF

If the following settings are not properly set, they can cause random S806 or S0C4 conditions.

1. Add the following CEEOPTS DD to your PIONEER and or VOYAGER Task (or other modules through STC/JCL) as needed.

   Example (this may vary by site requirements):

   ```
   //CEEOPTS DD DISP=SHR,
   //DSN=&SYSPLEX.OIDM.VOYAGER.CONTROL.PARMLIB(CEEPRM00)
   ```

2. Where the CEEPRM00 PDS member contains:
   a. RPTOPT(ON)
   b. RPTSTG(ON)
3. When you run the offending STC/JCL again you will get a list of the options in affect.
4. Compare the output of the current JES LOG and look for one of the following literals, so one may review the current options in place.
   a. "LAST WHERE SET"
   b. "IBM-supplied default"
   c. "ALL31"
5. Note that all LE options should all be reviewed (not only ALL31) as noted in Run Time Options, Defaults and Recommendations for IBM RACF.
6. The options can be overridden within the CEEOPTS DD through the CEEPRM00 PDS member (or site specific implementation), as follows:
   - Where CEEPRM00
   - ALL31(ON)
   - RPTOPT(ON)
   - RPTSTG(ON)
   - STACK(128K,128K,ANYWHERE,KEEP,512K,512K)
7. When the anomaly is addressed, the RPT* lines can be removed, if desired:
   - Where CEEPRM00

- ALL31(ON)
- STACK(128K,128K,ANYWHERE,KEEP,512K,512K)

H.2 Run Time Options, Defaults and Recommendations for IBM RACF

Customizing Language Environment run time options Z/OS Language Environment Customization: Info gathered from IBM Manual # SA22-7564-13.

Table H-1 lists Language Environment run time options, defaults and recommendations.

Table H-1   Language Environment Run Time Options, Defaults and Recommendations for IBM RACF

Option	Default	Recommended	IDF's
ABPERC	NONE	NONE	NONE
ABTERMENC	ABEND	ABEND	ABEND
AIXBLD	OFF	OFF	OFF
ALL31	ON	ON	ON
ANYHEAP	16K,8K,ANY,FREE	16K,8K,ANY,FREE	16K,8K,ANY,FREE
ARGPARSE	ARGPARSE	ARGPARSE	ARGPARSE
AUTOTASK	NOAUTOTASK	NOAUTOTASK	NOAUTOTASK
BELOWHEAP	8K,4K,FREE	8K,4K,FREE	8K,4K,FREE
CBLOPTS	ON	ON	ON
CBLPSHPOP	ON	N/A	ON
CBLQDA	OFF	OFF	OFF
CEEDUMP	60,SYSOUT=*,FREE-END,SPIN-UNALLOC	60,SYSOUT=*,FREE-END,SPIN-UNALLOC	60,SYSOUT=*,FREE-END,SPIN-UNALLOC
CHECK	ON	ON	ON
COUNTRY	US	User defined	US
DEBUG	OFF	OFF	OFF
DEPTHCONDLMT	10	0	10
DYNDMP	*USERID,NODYNAMIC,TDUMP	*USERID,NODYNAMIC,TDUMP	*USERID,NODYNAMIC,TDUMP
ENV	No default	User default	No default
ENVAR	"	"	"
ERRCOUNT	0	0	0
ERRUNIT	6	6	6
EXECOPS	EXECOPS	EXECOPS	EXECOPS
FILEHIST	ON	ON	ON
FILETAG	NOAUTOCVT,NOAUTOTAG	NOAUTOCVT,NOAUTOTAG	NOAUTOCVT,NOAUTOTAG
HEAP	32K,32K,ANY,KEEP,8K,4K	32K,32K,ANY,KEEP,8K,4K	32K,32K,ANY,KEEP,8K,4K

Table H-1 (Cont.) Language Environment Run Time Options, Defaults and Recommendations for IBM RACF

Option	Default	Recommended	IDF's
HEAP64	1M,1M,KEEP,32K,32K,KEEP,4k,4K,FREE	N/A	N/A
STACK	128K,128K,ANY,KEEP,512K,128K	128K,128K,ANY,KEEP,512K,128K	128K,128K,ANY,KEEP,512K,128K

There are many more run time options that are not applicable to this situation.

Pioneer Post-Processing Commands

Pioneer can post-process RACF commands. The commands that are enabled through Pioneer for Post-Processing are `ADDUSER`, `ALTUSER`, `CONNECT`, `DELUSER`, and `REMOVE`.

Post-processing is used to pass any USERID or command that is successfully processed by the provisioning agent (Pioneer) to a REXX/CLIST member for further processing. Post-processing is triggered by separate configuration entries in the Pioneer control file (<HLQ>.PIONEER.CONTROL.FILE). It accepts commands in the format `C=`*`SUPPORTED_COMMAND`*, where *SUPPORTED_COMMAND* can be `ADDUSER`, `ALTUSER`, `DELUSER`, `CONNECT`, or `REMOVE`.
When you specify a `C=`*`SUPPORTED_COMMAND`* command with the `L=` (library) and `M=` (member) parameters along with a required parameter (for example, `USERID=Y` or `CMD=Y`), the PDS in `L=` and `M=` is submitted by Pioneer through the Intrd. Therefore, `M=` from `L=` must be a batch Job to execute a REXX/CLIST exec. The parameter `USERID=Y` indicates the user ID to be passed and `CMD=Y` indicates the RACF command to be passed for post-processing. If you use `CMD=N`, then the user ID alone is passed for post-processing. The name of the REXX/CLIST member should be same as that of `M=member`.
To determine if there is a post-process, Pioneer will read the control file (DDNAME=PARMFLE) and looks for the `C=`*`SUPPORTED_COMMAND`* property.

ADDUSER Command in Post-Processing

Below are the valid configurations for post-processing the `ADDUSER` command. You must add only one of the following entries to the <HLQ>.PIONEER.CONTROL.FILE file:

```
C=ADDUSER,M=XXXXXX,L=XXX.XXX.XXX,USERID=Y
C=ADDUSER,L=XXX.XXX.XXX,M=XXXXXX,USERID=Y
C=ADDUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=Y
C=ADDUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=Y
C=ADDUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=N
C=ADDUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=N
```

The following is an example of the `ADDUSER` command:

```
C=ADDUSER,M=TEST,L=YOUR.PDS.LIBRARY,USERID=Y
```

- In this example, `USERID=Y` specifies that the RACF user ID must be passed as an argument to the CLIST/REXX exec in `M=` member. The RACF user ID is added as provided from LDAP.
- Pioneer receives the `ADDUSER` command through the LDAP and dynamically allocates PDS library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`.
- Pioneer Reads the PDS(`L=`) member(`M=`). As mentioned earlier, this must be a batch job to execute REXX/ CLIST. The following is a sample JCL:

  ```
  //REXXBTH JOB,SYSTEMS,CLASS=A,MSGCLASS=X,
  // MSGLEVEL=(1,1),REGION=4096K,NOTIFY=&SYSUID
  //STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
  //SYSTSPRT DD SYSOUT=*
  ```

```
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSPROC DD DSN=PIONEER.CLIST.LIBRARY,DISP=SHR
//SYSTSIN DD *
```

- Pioneer then adds %TEST (value in M= parameter) USERID (RACF USERID) as input to SYSSTSIN DD statement to execute %TEST
- Pioneer punches each of the above JCL statements and the `%TEST USERID` statement to the z/OS Intrdr.
- Pioneer closes library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`
- Pioneer frees library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`. The `% TEST Rexx` list checks for the USERID argument and processes further as post process.
- A simple Rexx clist (for example, `TEST`) is as follows:

  ```
  /* rexx */
  Arg p1
  ```

  When this rexx clist is executed, p1 contains the USERID passed. The Rexx list then passes this User ID as desired by your programmer.

ALTUSER Command in Post-Processing

If you require post-processing only for the `ALTUSER` command, then you must add only one of the following valid configuration entries to the <HLQ>.PIONEER.CONTROL.FILE file:

```
C=ALTUSER,M=XXXXXX,L=XXX.XXX.XXX,USERID=Y
C=ALTUSER,L=XXX.XXX.XXX,M=XXXXXX,USERID=Y
C=ALTUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=Y
C=ALTUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=Y
C=ALTUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=N
C=ALTUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=N
```

The following is an example of the `ALTUSER` command:

```
C=ALTUSER,M=TEST,L=YOUR.PDS.LIBRARY,USERID=Y
```

- In this example, `USERID=Y` specifies that the RACF user ID must be passed as an argument to the CLIST/REXX exec in `M=` member. The RACF user ID is added as provided from LDAP.
- Pioneer receives the `ALTUSER` command through the LDAP and dynamically allocates PDS library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`.
- Pioneer reads the PDS(`L=`) member(`M=`). The following is a sample JCL:

  ```
  //REXXBTH JOB,SYSTEMS,CLASS=A,MSGCLASS=X,
  // MSGLEVEL=(1,1),REGION=4096K,NOTIFY=&SYSUID
  //STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=20
  //SYSTSPRT DD SYSOUT=*
  //SYSPRINT DD SYSOUT=*
  //SYSUDUMP DD SYSOUT=*
  //SYSPROC DD DSN=PIONEER.CLIST.LIBRARY,DISP=SHR
  //SYSTSIN DD *
  ```

- Pioneer adds %TEST (value in M= parameter) USERID (RACF USERID) as input to SYSSTSIN DD statement to execute %TEST
- Pioneer punches each of the above JCL statements and the `%TEST USERID` statement to the z/OS Intrdr.
- Pioneer closes library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`
- Pioneer frees library `dsn='YOUR.PDS.LIBRARY'`, `member=TEST`. The `% TEST Rexx` list checks for the USERID argument and processes further as post process.
- A simple Rexx clist (for example, `TEST`) is as follows:

  ```
  /* rexx */
  Arg p1
  ```

  When this rexx clist is executed, p1 contains the USERID passed. The Rexx list then passes this User ID as desired by your programmer.

DELUSER Command in Post-Processing

If you require post-processing only for the `DELUSER` command, then you must add only one of the following valid configuration entries to the <HLQ>.PIONEER.CONTROL.FILE file:

```
C=DELUSER,M=XXXXXX,L=XXX.XXX.XXX,DEL=Y
C=DELUSER,L=XXX.XXX.XXX,M=XXXXXX,DEL=N
C=DELUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=Y
C=DELUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=Y
C=DELUSER,M=XXXXXX,L=XXX.XXX.XXX,CMD=N
C=DELUSER,L=XXX.XXX.XXX,M=XXXXXX,CMD=N
```

The following is an example of the `DELUSER` command:

```
C=DELUSER,M=TESTD,L=YOUR.PDS.LIBRARY,DEL=Y
```

In this example, the user ID is deleted from RACF and the job or script described in `M=,L=` is executed. The RACF user ID is passed as an argument to REXX exec as described in the examples for the `ADDUSER` and `ALTUSER` commands.

The following is another example of the `DELUSER` command:

```
C=DELUSER,M=TESTD,L=YOUR.PDS.LIBRARY,DEL=N
```

In this example, the user ID is not deleted from RACF and the job or script described in `M=,L=` is executed. The RACF user ID is passed as an argument to REXX exec as described in the examples for the `ADDUSER` and `ALTUSER` commands.

CONNECT Command in Post-Processing

If you require post-processing only for the `CONNECT` command, then you must add only one of the following valid configuration entries to the <HLQ>.PIONEER.CONTROL.FILE file:

```
C=CONNECT,M=XXXXXX,L=XXX.XXX.XXX,CMD=Y
C=CONNECT,L=XXX.XXX.XXX,M=XXXXXX,CMD=Y
C=CONNECT,M=XXXXXX,L=XXX.XXX.XXX,CMD=N
C=CONNECT,L=XXX.XXX.XXX,M=XXXXXX,CMD=N
C=CONNECT,M=XXXXXX,L=XXX.XXX.XXX,USERID=Y
C=CONNECT,L=XXX.XXX.XXX,M=XXXXXX,USERID=Y
```

The following is an example of the `CONNECT` command:

```
C=CONNECT,M=TEST,L=YOUR.PDS.LIBRARY,USERID=Y
```

This example specifies that for each `CONNECT` command execution, the job or script described in `M=,L=` is executed and the RACF user ID is passed as an argument to REXX exec as described in the examples for the `ADDUSER` and `ALTUSER` commands.

The following is another example of the `CONNECT` command:

```
C=CONNECT,M=TEST,L=YOUR.PDS.LIBRARY,CMD=Y
```

This example specifies that for each `CONNECT` command execution, the job or script described in `M=,L=` is executed and the RACF command is passed as an argument to REXX exec. A maximum of 69 characters from the `CONNECT` command is passed to REXX exec.

REMOVE Command in Post-Processing

If you require post-processing only for the `REMOVE` command, then you must add only one of the following valid configuration entries to the <HLQ>.PIONEER.CONTROL.FILE file:

```
C=REMOVE,M=XXXXXX,L=XXX.XXX.XXX,CMD=Y
C=REMOVE,L=XXX.XXX.XXX,M=XXXXXX,CMD=Y
C=REMOVE,M=XXXXXX,L=XXX.XXX.XXX,CMD=N
C=REMOVE,L=XXX.XXX.XXX,M=XXXXXX,CMD=N
C=REMOVE,M=XXXXXX,L=XXX.XXX.XXX,USERID=Y
C=REMOVE,L=XXX.XXX.XXX,M=XXXXXX,USERID=Y
```

The following is an example of the `REMOVE` command:

```
C=REMOVE,M=TEST,L=YOUR.PDS.LIBRARY,USERID=Y
```

This example specifies that for each `REMOVE` command execution, the job or script described in `M=,L=` is executed and the RACF user ID is passed as an argument to REXX exec as described in the examples for the `ADDUSER` and `ALTUSER` commands.

The following is another example of the `CONNECT` command:

```
C=REMOVE,M=TEST,L=YOUR.PDS.LIBRARY,CMD=Y
```

This example specifies that for each `REMOVE` command execution, the job or script described in `M=,L=` is executed and the RACF command is passed as an argument to REXX exec. A maximum of 69 characters from the `REMOVE` command is passed to REXX exec.

J

Pioneer SMF Process

The following is a brief outline of the startup and initialization process of Pioneer:

1. The Provisioning Agent - Pioneer reads the control file and validates the parameters.
2. The Provisioning Agent - Pioneer calls the following:
   - `IDFGETIF`: Extracts Jobname, Jobid, and Userid
   - `IDFCHKAU`: This new program issues RACROUTE macros to validate access the RACF userid that starts Pioneer’s access to `IDFRADMIN.CMD` facility.
3. If the Return code from `IDFCHKAU` is not 0, then error messages are issued and the Provisioning Agent - Pioneer terminates with `RC=300`.
4. If the Return code from `IDFCHJKAU` is 0, then IDFCHKIR is called using RACROUTE macros to validate access of the RACF userid that starts the Provisioning Agent - Pioneer. It has access to `IRR.RADMIN.*` that is required for RACF List functions performed by Provisioning Agent - Pioneer's called program MYRADMIN.
5. If Return code from IDFCHKIR is not 0, then error messages are issued and Provisioning Agent - Pioneer terminates with a `RC=300`.
6. When the previous steps are executed, the Provisioning Agent - Pioneer initializes the TCPIP interface and goes into a Listen state on the `PORT=` port.
7. When a message arrives and passes validation, then MYRADMIN is called if the functions are RACF LIST and SEARCH type functions. If the RACF functions are not LIST or SEARCH, then programs are called based on the SMF= parameter. If SMF=Y, then IDFRADMNS is called using the `SECURE_ID` and after the command completes a SMF TYPE 245 TYPE 1, 2 records are written. If SMF=N then IDFRADMN is called using the `SECURE_ID` and no SMF record is written.
8. When all the steps are completed, the output is sent back to the LDAP.

K
Pioneer Messages

Table K-1 lists the Pioneer messages.

> **Note:**
>
> All Provisioning Agent messages are prefixed with IDFRP. The next character after IDFRP defines the message type followed by 3 digit number that uniquely identifies the message in its specific sub-genre.

Table K-1 Pioneer Messages

Message ID	Message Text	Type
IDMP000I	Pioneer Starting	Informational
IDMP001I	Pioneer Input parameters are good	Informational
IDMP002I	Pioneer detects IDF-BUILD <Build info>	Informational
IDFRPI063	Pioneer detects Audit log is now: <value>	Informational
IDMP003I	Pioneer detects jobname: <value>	Informational
IDMP004I	Pioneer detects TCPIP IP Address <value>	Informational
IDMP005I	Pioneer detects TCPIP Port <value>	Informational
IDMP006I	Pioneer detects Debugging is <value>	Informational
IDMP009I	Pioneer detects encryption is enabled	Informational
IDMP011S	Pioneer is using <value> for security calls	Informational
IDMP011I	Pioneer detects CPUID of <value>	Informational
IDMP012I	Pioneer detects sysplex sysname of <value>	Informational
IDMP013I	Pioneer detects LPAR Name as <value>	Informational
IDMP014I	Pioneer detects country code of <value>	Informational
IDMP016I	Pioneer has APF authorization	Informational
IDMP017I	Pioneer found secured racf id	Informational
IDMP020I	Pioneer accepting messages on <value>	Informational

Table K-1 (Cont.) Pioneer Messages

Message ID	Message Text	Type
IDMP020A	Pioneer operatoe has issued shutdown command	Informational
IDMP030I	Pioneer INITAPI was successful	Informational
IDMP031I	Pioneer GETCLIENTID was successful	Informational
IDMP032I	Client name is <value>	Informational
IDMP033I	Client task is <value>	Informational
IDMP035I	Pioneer bind socket was successful	Informational
IDMP036I	Pioneer Listening port is <value>	Informational
IDMP037I	Pioneer listening address is <value>	Informational
IDMP038I	Pioneer listen socket call was successful	Informational
IDMP038A	Pioneer is ready for messages	Informational
IDMP039I	Pioneer write socket call was succesful	Informational
IDMP041I	Pioneer socket accept was successful	Informational
IDMP050A	Pioneer closing IP connection	Informational
IDMP051I	Pioneer close socket call was successful	Informational
IDMP052I	Pioneer shutdown socket call was successful	Informational
IDMP070I	Pioneer <filename> is now open	Informational
IDMP071I	Pioneer <filename> is now closed	Informational
IDMP100I	Pioneer (IN) MSGS processed is: <value>	Informational
IDFRPI064	Pioneer message (READ) bytes: <value>	Informational
IDFRPI065	Pioneer message (write) bytes: <value>	Informational
IDMP102I	Pioneer terminating	Informational
IDMP201I	Pioneer <parm> status=good	Informational
IDMP206I	Pioneer jobname is <value>	Informational
IDMP207I	Pioneer jobid is <value>	Informational
IDMP208I	Pioneer RACF userid is <value>	Informational
IDMP209I	Pioneer RACF userid <value> authorized for IDF ADMIN.CMD	Informational
IDMP210I	Pioneer RACF userid <value> authorized for IRR.RADMIN.*	Informational
IDMP500I	*AUDIT* - <message,info,txt>	Audit
IDMP500I	*HEADER* - AUDIT FUNC <value> RACF CMD <value> RACF USR <value> FUNCSTAT <value> OTHER-INFO <value>	Audit
IDMP500I	<Blank> AUDIT MESSAGE	Audit

Table K-1 (Cont.) Pioneer Messages

Message ID	Message Text	Type
IDMP400I	*PARMS* - <text,info,value>	Informational
IDMP400I	*PARMS* - <text,info,value>	Informational
IDMP305I	Pioneer debugging was turned <value>	Informational
IDMP306I	Pioneer-polloper: <value>	Informational
IDMP307I	Pioneer received a status query and is alive	Informational
IDFRPI050	*** SUBMISSION JCL END ***	Informational
IDFRPI051	<file-name> CLOSED OK	Informational
IDFRPI052	EZACIC09 OUTPUT : CANONICAL NAME <VALUE> NAME LENGTH <VALUE> NAME <VALUE> NEXT ADRINFO <VALUE>	Informational
IDFRPI053	DEBUG: <value>	Informational
IDFRPI054	Timer Wait Failed	Informational
IDFRPI055	SOCK# <value> is OFF /ON	Informational
IDFRPI056	PIONEERX: <Function value> RETCODE = <value> ERRNO = <value>	Informational
IDFRPI057	USERID=(<value>) RC=<value> OOPS(1)=<retcode 1> OOPS(2)=<retcode 2> OOPS(3) = <retcode 3>	Informational
IDFRPI058	<IMPORTU/G file type> records read : <value>	Informational
IDFRPI059	LDAP Command <value>	Informational
IDFRPI060	PIONEER-BUILD : <IDF-BUILD Value>	Informational
IDFRPI061	Write Socket MSGS : <value>	Informational
IDFRPI062	RACF CMD: <value> FOR USER <value> SAF Return code <value> RACF Return code <value> RACF Reason code <value>	Informational
IDFRPI066	Batch Reconciliation: User reconciliation status information	Informational
IDFRPI067	Batch Reconciliation: Group reconciliation status information	Informational
IDMP001E	Pioneer <Error> <Info>	Error
IDMP009E	Pioneer detects encryption is disabled	Error
IDMP016E	Pioneer has no APF authorization and is required	Error
IDMP030E	Pioneer INITAPI failed RC/ ERRNO: <value>	Error
IDMP040E	Pioneer translation was not successful from-to: <value>	Error

Table K-1 (Cont.) Pioneer Messages

Message ID	Message Text	Type
IDFRPE023	Pioneer socket accept was not successful RC: <value>	Error
IDMP209E	Pioneer RACF userid <value> not authorized for IDFADMIN.CMD	Error
IDMP210E	Pioneer RACF userid <value> not authorized for IRR.RADMIN.*	Error
IDMP211E	Pioneer Translation failed from <value> using Country code <value> reason <value>	Error
IDMP208E	Pioneer module IDFGETIF failed to extract JOB info - Pioneer will terminate	Error
IDMP401E	*PARMS* - <text,info,value>	Error
IDMP402E	Pioneer is ending due to errors	Error
IDFRPE024	Pioneer ends RC: <value>	Error
IDFRPE025	(PIONEER) DDNAME (DDname) IS EMPTY. POPULATE AS DESCRIBED IN MANUAL. - PIONEER WILL ABEND WITH RC = 100	Error
IDMP403E	Pioneer found secure-ID errors - Found: <value>	Error
IDFRPE015	Pioneer found a syntax error in a post-parameter Error occurred with <value> Post Error is <value> in-parm <value> Closing PARMFLE	Error
IDFRPE016	Function= <value> RETCODE=<value> ERRORNO=<value>	Error
IDFRPE017	INVALID USER <value> RETCODES = <value1>, <value2>, <value3>	Error
IDFRPE018	Error Calling Program <name>	Error
IDFRPE019	DEBUGOUT Did not allocate OK RC: <value>	Error
IDFRPE020	DYNAMIC CALL Failed BPXWDYN-RC : <value>	Error
IDFRPE021	PIONEER COULD NOT FREE DEBUGOUT	Error
IDFRPE022	<FILE> Failed to <ACTION> RC: <value>	Error
IDMP080E	Pioneer called RACF with SMF=Y and SMFPARMXX had no record=245 defined a critical error	Error
IDMP300I	*DEBUG* <message,text,value>	Debug
IDMP048I	Pioneer the LDAP connection timed out	Warning
IDMP049I	Pioneer has been idle for 10 minutes	Warning

Table K-1 (Cont.) Pioneer Messages

Message ID	Message Text	Type
IDMP402I	Pioneer No open TCP connections found	Warning
IDFRPW005	Max Timeouts Triggered, Going to Exit	Warning
IDFRPW006	Pioneer control file must have control records in it see admin manual for details Pioneer will abend	Warning
IDFRPW007	NO DDNAME Found	Warning
IDFRPW008	DINDX Invalid	Warning
IDFRPW009	Pioneer could not open <file> RC: <value>	Warning
IDFRPW010	ALIAS FUNCTION TYPE NOT SUPPORTED	Warning
IDFRPW011	FILE-FREE TYPE NOT SUPPORTED	Warning

L

Voyager Messages

Table L-1 lists the Voyage messages.

Note:

All Reconciliation Agent messages are prefixed with IDFRV. The next character after IDFRV defines the message type followed by 3 digit number that uniquely identifies the message in its specific sub-genre.

Table L-1 Voyager Messages

Message ID	Message Text	Message Type
IDMV000I	Voyager reconciliation agent starting	Informational
IDFRVI068	Voyager is executing from an APF authorized library	Informational
IDFRVI069	Voyager found <value> security subsystem	Informational
IDFRVI070	Voyager subpool initialization Ok	Informational
IDMV001I	Voyager subpool size is : <value> K	Informational
IDMV002I	Voyager subpool will hold <value> messages	Informational
IDMV003I	Voyager SP231 allocated OK	Informational
IDMV004I	Voyager storage token built OK	Informational
IDMV006I	Voyageer build level is at <value>	Informational
IDMV008I	Voyager detects subpool - 100 byte version	Informational
IDMV009I	Voyager detects (TCPIP) jobname <value>	Informational
IDMV010I	Voyager detects (TCPIP) IP address <value>	Informational
IDMV011I	Voyager detects (TCPIP) IP port <value>	Informational
IDMV012I	Voyager detects encryption is ON	Informational
IDMV015I	Voyager detects cache file opened OK	Informational
IDMV016I	Voyager computing cache timer delay successful	Informational
IDMV017I	Voyager detects encryption KVER spaces	Informational
IDMV019I	Voyager detects debugging is OFF	Informational
IDMV020I	Voyager detects MVS retcodes is	Informational

Table L-1 (Cont.) Voyager Messages

Message ID	Message Text	Message Type
IDFRVI020	Voyager detects MVS retcodes is	Informational
IDMV022I	Voyager detects hostname of <value>	Informational
IDMV023I	Voyager initialization of TCP API was successful	Informational
IDMV024I	Voyager initialization of GET client ID was successful	Informational
IDMV025I	Voyager accpeting messages on <value>	Informational
IDMV027I	Voyager connected to gateway server was successful	Informational
IDMV032I	Voyager connection start timer begins	Informational
IDMV033I	Voyager connection start timer ends	Informational
IDMV050I	Voyager cache polling begins	Informational
IDMV070I	Voyager <filename> is now open	Informational
IDMV071I	Voyager <filename> is now closed	Informational
IDMV100I	Voyager shutdown started	Informational
IDMV103I	Voyager found storage allocation	Informational
IDMV104I	Voyager storage freed OK	Informational
IDMV110I	Voyager reconciliation agent has terminated	Informational
IDMV111I	Voyager has ended with zero return code	Informational
IDFRVI071	Voyager sent messages <value> received messages <value>	Informational
IDMV105I	Voyager subpool messages read : <value>	Informational
IDMV151I	Voyager DNS request <value>	Informational
IDMV152I	Voyager IP connect request <value>	Informational
IDMV155I	Voyager cachesave was read <value> messages	Informational
IDFRVI072	Voyager wrote <value> messages	Informational
IDMV200E	Voyager startup parameter error	Informational
IDMV200I	Voyager unable to connect to gateway	Informational
IDMV202E	Voyager no storage token found	Informational
IDMV202I	Voyager unable to connect to NEW IP/PORT	Informational
IDMV210I	Voyager <poll message value>	Informational
IDMV400I	*PARM* <value, info, text>	Informational
IDMV500I	*AUDIT* <value, info, text>	Audit
IDMV109I	Voyager write succesful - MSG = <value>	

Table L-1   (Cont.) Voyager Messages

Message ID	Message Text	Message Type
IDMV111I	Voyager probing LDAP ……….	
IDMV112I	Polling Cache 100 processed	
IDMV113I	Messages sent <value>	
IDMV300I	END saving cache to DASD	
IDMV114I	Voyager RACF Listuser/Group call	
IDFRVI056	SAF Return Code : <value>	
IDFRVI057	RACF Return code : <value>	
IDFRVI058	RACF Reason code: <value>	
IDMV001E	Voyager has terminated. See sysout for details	
IDFRVI060	Voyager detects TCP write is Off	
IDFRVI061	Voyager Control File must have control records in it see admin manual for control record format sequenece Voyager will now Abend	
IDFRVI062	Voyager cache found phase : <value>	
IDFRVI063	SOCKET-NUM : <value> SOCKETS-USED : <value> SOCKETSUSED: <value> %	
IDFRVI064	MAXNo= <value> SOCKETS-USED=<value>	
IDFRVI065	Select call modified MAXSNO to : <value>	
IDFRVI066	NO DDNAME FOUND	
IDFRVI067	<File Name> <Status msg> RC : <value>	
IDMV300I	*DEBUG* <value, info, text>	Debug
IDMV000E	Voyager has exceeded the connect itnv/retry OF : - <value> secs <value> times	Error
IDFRVE022	Voyager is not executing from an APF auhtoirzed library	Error
IDFRVE023	Voyager does nto have AES encryption active and AES active is required	Error
IDFRVE024	Voyager subpool initialization failed storage obtain RC: <value>	Error
IDMV004E	Voyager storage token build failed IEANTCR RC: <value>	Error
IDMV005E	Voyager input control file is empty	Error
IDMV015E	Voyager detects cache file <file name> Failure RC: <value>	Error
IDFRVE025	Voyager detects cache file read error RC: <value>	Error

Table L-1 (Cont.) Voyager Messages

Message ID	Message Text	Message Type
IDMV022E	Voyager detects bad hostname of <value>	Error
IDMV023E	Voyager initialization of TCP API failed RC: <value>	Error
IDMV024E	Voyager initialization of GET client ID failed RC: <value>	Error
IDMV026E	Voyager initialization of PTON failed	Error
IDMV070E	Voyager could not open <filename> RC: <value>	Error
IDMV112E	BAD cache record - Leaving	Error
IDMV113E	TCP Err messages - Leaving	Error
IDFRVE016	Voyager cache has an invalid phase	Error
IDFRVE017	LDAP Gateway error <TCP Add> has not responded in <retry value> secs	Error
IDFRVE018	Voyager COBOL LE Timer function failed <value>	Error
IDFRVE019	Destination IP/PORT failure reason : <value> Voyager will retry connection	Error
IDFRVE020	Connection to LDAP has failed TCPIP recovery to LDAP has failed recovery has occurred: <failure retry value>	Error
IDFRVE021	Reserved	Error
IDMV211E	Voyager Translation failed from :<val> using country code <val> reason : <value>	Error
IDMV013I	Voyager detects cache delay set to <value> secs	Warning
IDMV014I	Voyager detects cache read delay <value> secs	Warning
IDMV018I	Voyager detects debugging is ON	Warning
IDMV103I	Voyager has ended with a non-zero return code	Warning
IDFRVW005	Cache Overflow	Warning

M

Features of the Mainframe Agents

This appendix contains the following topics:

- Functions Supported by the Pioneer Provisioning Agent
- Functions Supported by the Voyager Reconciliation Agent

M.1 Functions Supported by the Pioneer Provisioning Agent

The Pioneer Provisioning Agent supports the following functions:

Standard IBM RACF user profile commands:

- [ADDUSER]: Creates a IBM RACF user profile
- [ALTUSER]: Modifies a IBM RACF user profile
- [DELUSER]: Deletes a IBM RACF user profile
- [PASSWORD]: Modifies password data for IBM RACF user profile

Standard IBM RACF group profile commands:

- [ADDGRP]: Creates a IBM RACF group profile
- [ALTGRP]: Modifies a IBMRACF group profile
- [DELGRP]: Deletes a IBM RACF group profile
- [CONNET]: Adds an IBM RACF user to a group. This command works based on the variables that set access rights.
- [REMOVE]: Removes an IBM RACF user from a group.

Standard IBM RACF data set and resource profile commands:

- [PERMIT]: Provides data set or resource profile access to a user
- [SETROPTS]: Refresh access to resource

Standard IBM RACF searching:

- SEARCH CLASS(USER)
- SEARCH CLASS(GROUP)
- SEARCH CLASS(DATASET)
- [RLIST]: Retrieve Resource information

Standard IBM RACF alias commands:

- [DEFINE ALIAS]: Defines user to catalog
- [DELETE ALIAS]: Removes user from catalog
- [LISTC ENTRIES]: List Master Catalogs By User
- [LISTC LEVEL]: List Datasets By User

Proprietary IdentityForge IBM RACF Authentication and Change Password Commands:

- CHKUSER
- CHKUSERPWD

Table M-1 lists the functions supported by the Provisioning Agent - Pioneer.

Table M-1 Functions Supported by the Provisioning Agent - Pioneer

Function	Description
Authenticate Users	Validates users LoginId and Password.
Create Users	Adds new users IBM RACF.
Modify Users	Modifies user information in IBM RACF.
Change Passwords	Changes user passwords on IBM RACF in response selfservice change password.
Reset Passwords	Resets user passwords IBM RACF. The passwords are reset by the administrator.
Change Passphrase	Changes user passphrase on IBM RACF in response self-service change passphrase.
Reset passphrase	Resets user passphrase on IBM RACF. The passphrase are reset by the administrator.
Disable User Accounts	Disables users in IBM RACF.
Enable User Accounts	Enables users in IBM RACF.
Delete Users	Removes users from IBM RACF.
Create Groups	Adds new groups to IBM RACF.
Modify Groups	Modifies group information in IBM RACF.
Delete Groups	Removes groups from IBM RACF.
Search All Users	Retrieves all users with current data from IBM RACF.
Search All Groups	Retrieves all groups with current data from IBM RACF.
Search All Datasets	Retrieves all datasets with current data from IBM RACF.
Search All Dataset Profiles by User	Retrieves all dataset profiles for a given RACF,LOGIN ID.
Search Resource	Retrieve RACF resource with current data.
Grant Users Access to Datasets and General Resources	Adding/Removing the user to an IBM RACF dataset or resources.
Grant Users Access to Privileges (TSO, SPECIAL)	Provides TSO login access to users or other Privileges.
Grant User TSO attributes	Provides TSO information.
Grant User NETVIEW attributes	Provides NETVIEW information.
Grant User CICS attributes	Provides CICS information.
Grant User CSDATA attributes	Provides CSDATA user-defined information.
Grant User OMVS attributes	Provides OMVS information.
Grant Users Access to Groups	Adding the user to an IBM RACF group.

M.2 Functions Supported by the Voyager Reconciliation Agent

The Voyager Reconciliation Agent supports reconciliation of changes that are made to user profiles by using commands such as ADDUSER or ALTUSER. These commands also contain users' passwords for reconciliation, if any.

The Voyager Reconciliation Agent supports the following functions:

- Change passwords
- Create user data
- Modify user data
- Password Interval changes
- Disable users
- Delete users
- Enable users
- Group Membership Changes
- Create group data
- Modify group data
- Delete groups
- Audit information

N
Custom Data Field (CSDATA)

Custom Data Fields (CSDATA) are user-defined segments available on the User and Group profiles. These fields provide an alternative to Installation Data or use of other predefined segments to keep the information in the RACF database.

CSDATA is defined in the CFIELD General Resource Class using the standard RACF commands. Multiple fields can be defined in the CSDATA. CSDATA may contain any information such as: Department, Employee ID, Email address, Physical Location (City/State), and so on.

This appendix contains the following topics:

- Adding CSDATA Fields
- Parsing CSDATA Fields

N.1 Adding CSDATA Fields

RACF administrators can create customized user-defined fields for specific segments.

For example, to add a field with 8 characters all numeric:

```
RDEFINE CFIELD USER.CSDATA.EMPSER UACC(NONE)
           CFDEF(TYPE(NUM)
           FIRST(NUMERIC) OTHER(NUMERIC)
           MAXLENGTH(8)
           MINVALUE(100000)
           MAXVALUE(99999999)
           HELP('EMPLOYEE SERIAL, 6-8 Digits')
           LISTHEAD('Employee serial=')
```

To enter more than one value, separate each value with a vertical bar (|) character. Each field name should have a corresponding configDNames entry.

To include CSDATA in a LISTUSER command [true|false] `-useCSDATA_=true/`.

> **Note:**
>
> To include CSDATA as part of the 'LISTUSER' command, set the value of the command to true.

To add a custom attribute or custom dataset, set values for the `_configAttrs_`, `_configDNames` and `_configDatasets_`properties in the `connector.properties` file. Here:

- `_configAttrs_`: specifies the property that holds the field names of any custom target system fields that are defined in the CSDATA user segment and used during user provisioning operations. To enter more than one value, separate each value with a vertical bar (|) character. Each field name should have a corresponding configDNames entry. For example:

```
# CUSTOM CSDATA RACF ATTRIBUTE FIELD NAME
_configAttrs_=$PST15|VEND ID|
```

- `_configDNames_`: specifies the property that holds the display name(s) of RACF field(s) that are defined in the CSDATA segment and used during user reconciliation operations. If entering more than one value, separate each value with a vertical bar (|) character. Each display name should have a corresponding configAttrs entry. For example, to define a field with a display name of $PST15 and VEND ID, then enter:

```
# CUSTOM CSDATA RACF ATTRIBUTE DISPLAY NAME
_configDNames_=$PST15 =|VEND ID =|
```

N.2 Parsing CSDATA Fields

To properly parse the fields from the CSDATA section of a user extract, the following parameter must be defined:

```
_configExtractAttrs_=<fld1>|<fld>
```

Here, <fld> = 9-character field name including ending colon.

For example, if the user extract produces the following CSDATA listing:

```
Segment: CSDATA Fields:03
EMPSER :100101
ADDRESS1:20 Main St., Anytown, PA., 08011
PHONE :555-444-7777
```

The definition of the property would be: `_configExtractAttrs_=EMPSER :|ADDRESS1:|PHONE :`

It is important to make sure that the colon character is in the ninth character position on all fields being parsed from a user extract as that is considered the standard format of all field listings in a user extract.

To add a field with 8 characters all numeric:

```
RDEFINE CFIELD USER.CSDATA.EMPSER UACC(NONE)
           CFDEF(TYPE(NUM)
           FIRST(NUMERIC) OTHER(NUMERIC)
           MAXLENGTH(8)
           MINVALUE(100000)
           MAXVALUE(99999999)
           HELP('EMPLOYEE SERIAL, 6-8 Digits')
           LISTHEAD('Employee serial=')
```