Governance, Risk, and Compliance Controls Suite

Configuration Controls Governor Apps Administration Guide

Software Version 5.1.1

ORACLE®

Integra Apps User's Guide

Part No. IA0001-511A

Contents

Prerequisites and assumptions

This guide assumes that:

All instructions in the *Integra Apps 5.1 Installation Guide* have been performed.

Integra Apps administrators are also ERP application administrators with superuser or user privileges.

Intended audience

- Personnel who must configure Integra Apps for the first time after installation
- Integra Apps administrators

Documents about Integra Apps 5.1

Release Notes provides a quick overview of the new features offered in Integra Apps 5.1.

Installation Guide gives step-by-step instructions for installing Integra Apps 5.1.

Administration Guide explains how to configure and administer Integra Apps 5.1.

User Guide shows how to use Integra Apps 5.1 once it has been installed and configured.

Terminology

Systems and Databases

Integra Agent	Integra-installed and owned schema in the ERP instance. Used by Integra components when performing tasks that require access to the instance.
Integra component	Any one of the following: Informia Archive, Informia Subset, Informia Reorg, Informia Data Masking, Integra Apps, Integra Codebase, Integra Access, Integra Transaction, or Integra Forms.
Integra Home instance	Database instance where Integra's components are installed, including the Integra Home schema and component-specific tablespaces. The instance may also contain non-Integra schemas.
Integra Home schema (AMHOME)	Contains all data about your Integra users, security and deployments. Usually named AMHOME.
Integra UI	User interface of Integra's components; i.e., the portion of the components that most users and administrators work with directly.
ERP application	An enterprise-scale application, such as "General Ledger."
ERP application UI	The ERP application's user interface.
ERP instance	Database instance that contains an ERP application's data.
ERP module	Synonymous with ERP application.
ERP system	Joins or integrates your ERP applications.

Using Integra

Definition	Collection of one or more programs and their user-specified parameters.

Job	A sequence of one or more definitions and/or standalone programs scheduled for execution. **Future** jobs are those whose execution has not begun; **current** jobs are those being executed now, and **past** jobs have either been completely executed or terminated.
Occurrence	Generated when a definition is executed. Contains data about the definition's execution, and reports generated by the definition's programs.
Program	A batch of steps or instructions that can be scheduled for execution in a job. **Standalone** programs can be scheduled independently of other programs (in contrast to the programs that are part of a definition and therefore always scheduled together).

Using Integra Apps

Change Tracker	Displays all changes collected by Change Tracking definitions.
Change Tracking Definition	Apps definition type that monitors changes to ERP application setups.
Change Tracking Alert	Email message that describes changes to ERP application setups. Based on Change Tracking Queries.
Change Tracking Query	Filters the data displayed in the Change Tracker.
Change Tracking Trigger	Installed in the ERP instance by the Change Tracking definition; notes events you specify (inserts, updates, deletes)
Comparison	Report that identifies differences between two Snapshot occurrences.
Displayed Field	Field that is visible to ERP system users.
Field Description	Prompt or label on a field that is visible to ERP system users.

Migration Definition	Apps definition type that moves ERP application setup data from one instance to another.
Non-Displayed Field	Field not visible to ERP system users, but contained in an ERP table.
Object	Group of setup tables to be monitored. Helps you choose the information to be monitored without having to grapple with data model or entity relationships. An Object is typically associated with an ERP application setup form or page.
Snapshot Definition	Apps definition type that reports ERP application setup data.
Template	Determines the data presented in Snapshot occurrences.

Quick Start

This section shows you step-by-step how to configure Integra Apps 5.1 for the first time. If you've already configured Integra Apps 5.1, see How to... (p. 31) for tutorials, or Reference (p. 55) for quick overviews of features and options.

In This Section

Log in

1 Open a web browser window and visit the Integra **Secure Login** page:

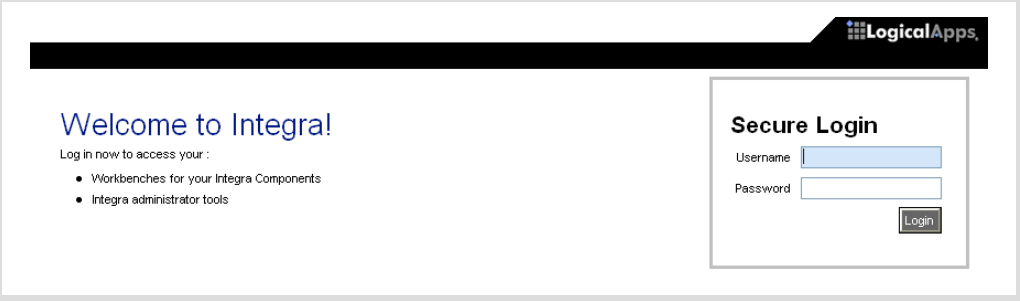
`http://`*`hostname`*`/`*`web-application-name`*

...where:

`hostname` is the domain name of the web server configured to serve Integra

`web-application-name` is the name of the Integra web application (usually `integra`).

You might also have an Integra icon on your desktop; double-clicking it opens this page.



2 If you already have your own Integra username and password, enter them now. However, if this is the first time your organization has logged into Integra, use the following default login information:

Username: `amadmin`

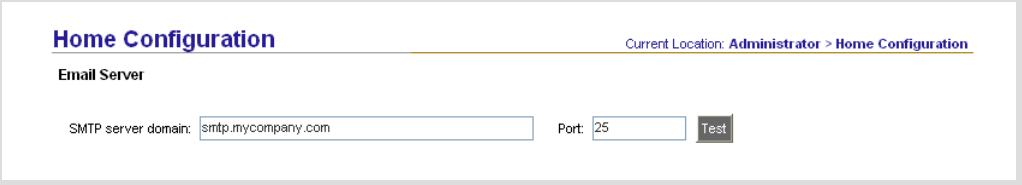
Password: `amadmin` (case sensitive)

You must be assigned the **Administrator** role to access the **Home Configuration** page. The user `AMADMIN` has this role.

Configure Email Alerts

Integra's Email Alerts feature is used by all Integra components. Follow the steps below to configure the email server.

1 Select **Administrator > Home Configuration** from the menu bar. The **Home Configuration** page appears.



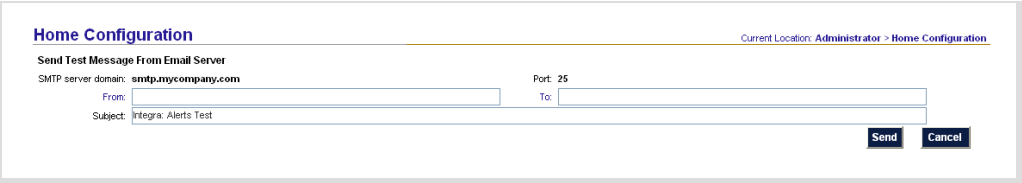
2 Enter the following information:

Email Server

SMTP server domain	The domain name of your mail server. If left blank, Integra cannot send email notifications.
Port	The port number of your mail server.

3 Click **Save**.

4 Click **Test...**. The **Send Test Message from Email Server** page appears.



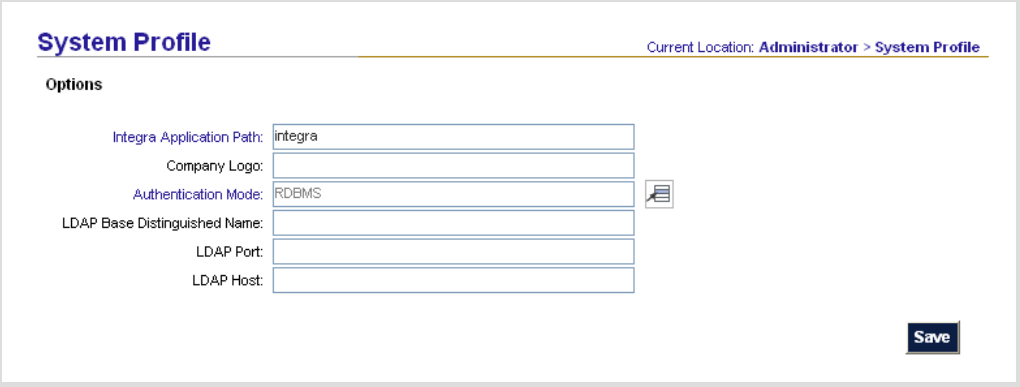
Enter **From** and **To** addresses, and optionally a **Subject**, and click **Send**. The page is refreshed, displaying **Status** information. If the test message is sent successfully, but you don't receive it, first verify that the message isn't being treated erroneously as spam by your incoming mail server or mail client.

When you're finished, click **Back**. The **Home Configuration** page reappears.

Integra does not use the Oracle Alerts product. It sends email messages directly from the database, using Oracle's UTL_SMTP feature.

Configure the System Profile

1 Select **Administrator > System Profile** from the menu bar. The **System Profile** page appears.



2 Enter this information:

Options

Integra Application Path	Final portion of the URL that summons the **Secure Login** page. The entire URL is: `http://hostname/web-application-name` ...where: `hostname` is the domain name of the Integra UI server The default value of `web-application-name` is **integra**
Company Logo	Name of the image file that is displayed in the upper left corner of each page. The file must be located in the **images** subdirectory of the directory on the Integra UI server where the Integra web application is deployed.
Authentication Mode	There are two options for validating users: **RDBMS** uses an Integra RDBMS module. **LDAP** uses an LDAP directory maintained by your organization. This eliminates the need for Integra users to enter their password on the **Secure Login** page. If you choose this option, the following fields are required:

LDAP Base Distinguished Name	LDAP entry that identifies an authorized user of the LDAP server.
LDAP Host	Machine name or IP address where the LDAP server is hosted.
LDAP Port	Port number to use when communicating with the host.

3 Click **Save**.

Configure Integra for each ERP instance

You must configure Integra to work with each ERP instance that you want to manage. Broadly speaking, you will perform these tasks for each instance:

Define a new instance in Integra.

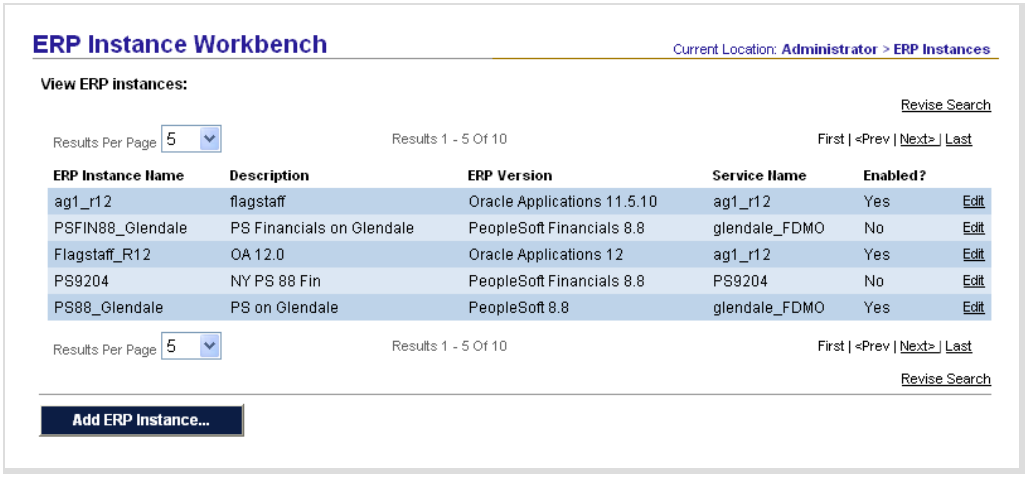
Create Integra Agent tablespaces for the instance. The Integra Agent lets Integra components perform tasks that require access to the instance's data.

Configure Integra Agent parameters.

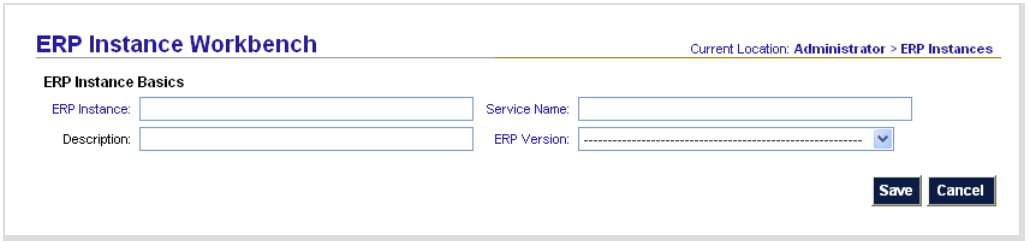
Configure the component's parameters.

Define a new ERP instance

1 Select **Administrator > ERP Instances** from the menu bar. If one or more instances have been defined already, a list of instances appears.



2 Click **Add ERP instance...**. A new definition page appears.



3 Enter the following information:

Basics

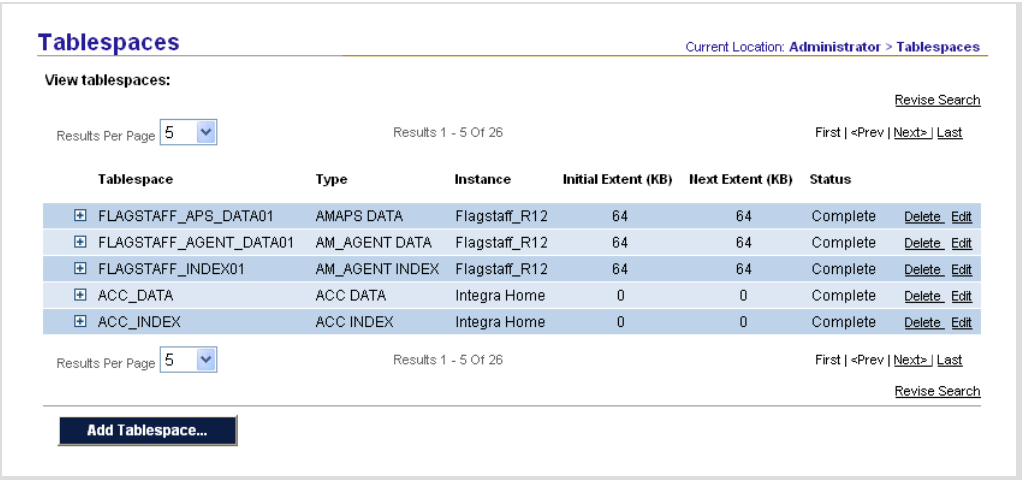
ERP Instance	Short name that identifies the ERP instance
Description	Longer, more descriptive name for the ERP instance. For example, if **Instance name** is `Dev`, **Description** could be `Development Instance`
Service Name	TNSNAMES service name that will be used when Integra creates a database link from the Integra Home schema to the new Integra Agent schema on the ERP instance. This name must be present in the TNSNAMES configuration for the Integra Home instance's Oracle Home.
ERP Version	ERP application version

4 Click **Save**. The page is refreshed. Your installed Integra components will be listed in the **Components** section.

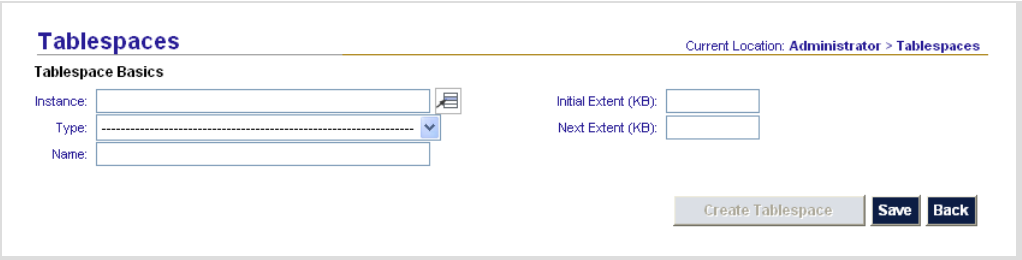
Create tablespaces

If any the following tablespaces already exist, there is no need to recreate them.

1 Select **Administrator > Tablespace Workbench** from the menu bar. The **Tablespace Workbench** appears.



2 Click **Add Tablespace...**. The **Tablespaces** page appears.



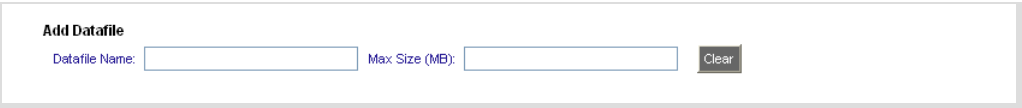
3 Enter the following information:

Tablespace Basics

Instance	Click and select the ERP instance
Type	Click and select **AM_AGENT DATA**
Name	`AM_AGENT_D`
Initial extent (KB)	`64`
Next extent (KB)	`64`

Tablespaces will be locally managed with user-defined extents. If you must define a tablespace differently, create it manually and enter its name in the **Name** field; there will be a warning the tablespace exists - choose yes to continue.

4 Click **Save**. The page refreshes, with a new **Datafiles** section below **Tablespace Basics**. The new section contains these fields:



5 Enter the following information:

Add datafile:

Datafile Name	Enter the full path from the root directory to a new datafile named `AM_AGENT_D1.DBF`
Max Size (MB)	`1000`

About **Max Size**: The datafile's size will initially be 10% of this value, and will automatically increase in 10% increments until **Max Size** is reached.

6 Click **Save**. You will be asked for the system password. After you enter it, the page refreshes again.

7 Click **Create tablespace**. A job to create the tablespace will be submitted for immediate execution.

8 Select **View > View Current Jobs**, and monitor the job's progress.

9 Once the job has completed, return to the **Tablespace Workbench** by selecting **Administrator > Tablespace Workbench**.

10 Create the following additional tablespace(s) using the steps above:

Tablespace Basics

Instance	The ERP instance
Type	**AM_AGENT INDEX**
Name	`AM_AGENT_X`
Initial extent (KB)	`64`
Next extent (KB)	`64`

Add datafile:

Datafile Name	Enter the full path from the root directory to a new datafile named `AM_AGENT_X1.DBF`
Max Size (MB)	`1000`

Integra Apps Tablespaces

Instance	The ERP instance	The ERP instance
Type	**APS_DATA**	**APS_INDEX**
Name	`APS_AGENT_D`	`APS_AGENT_X`
Initial extent (KB)	`64`	`64`

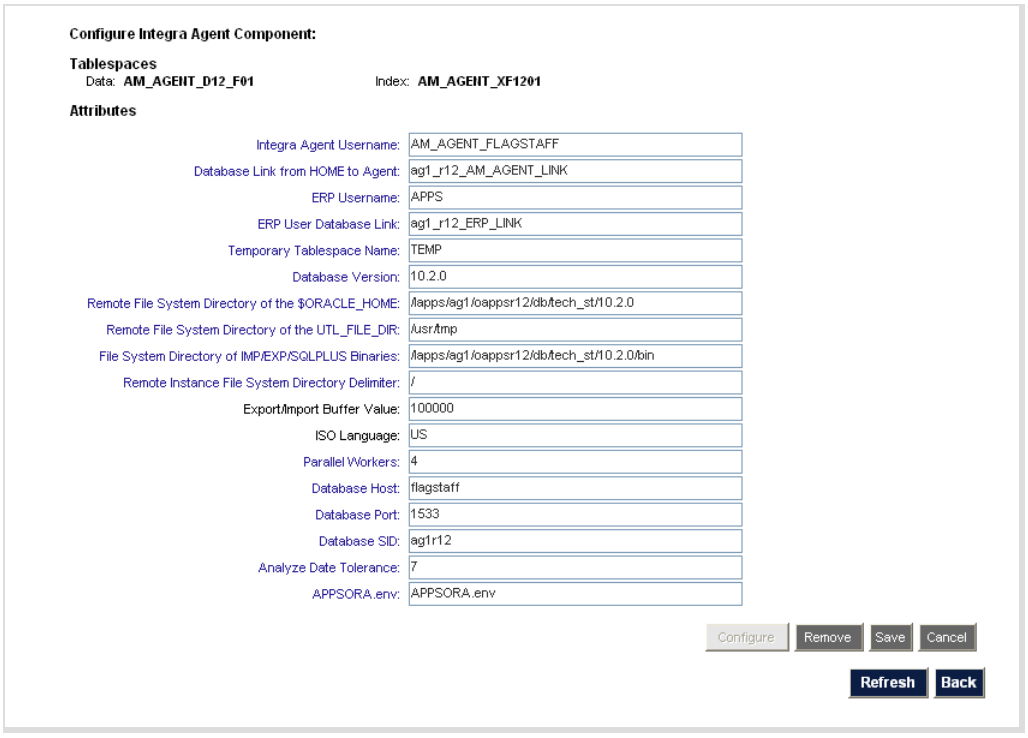
Next extent (KB)	64	64
Add datafile:		
Name	Enter the full path to a new datafile named APS_AGENT_D1.DBF	Enter the full path to a new datafile named APS_AGENT_X1.DBF
Max Size (MB)	500	500

Configure Integra Agent

1 Select **Administrator > ERP Instance Workbench** from the menu bar.

2 Click **Edit** to the right of the ERP instance's name.

3 To the far right of **Integra Agent**, click **Edit**. The **Configure Integra Agent Component** section appears.



4 Revise the following information if necessary:

Attributes

Integra Agent Username	For the Integra Agent schema. Created on the ERP instance if it does not exist. Default value: `AM_AGENT`
Database Link from HOME to Agent	Needed to transfer metadata from the Integra Home instance to the ERP instance, and to execute any process on the ERP instance. Created on the Integra Home instance if it does not exist. The value must contain no more than 20 characters.
ERP Username	Existing ERP instance user with full access to the ERP instance's data. ***Oracle E-Business Suite:*** The typical value is `APPS` ***PeopleSoft Enterprise:*** The typical value is `SYSADM` ***Siebel CRM Applications:*** The typical value is `SIEBEL`
ERP User Database Link	Needed to access ERP application information. Created on the ERP instance if it does not exist. The value must contain no more than 20 characters.
Temporary Tablespace Name	Name of the ERP instance's temporary Tablespace.
Database Version	ERP instance's database version. Example: `8.1.7`
Remote File System Directory of the $ORACLE_HOME	File system directory that corresponds to the $ORACLE_HOME of the ERP instance.

Remote File System Directory of the UTL_FILE_DIR	File system directory specified by the UTL_FILE_DIR parameter in the ERP instance's INIT.ORA file. Tells Integra where to read and write files that reside on the ERP instance.
File System Directory IMP/EXP/SQLPLUS Binaries	File system directory path where these executables reside on the ERP instance.
Remote Instance File System Directory Delimiter	The character used by your ERP instance's file system to separate elements within pathnames. ***Unix/Linux:*** `/` (forward slash) ***Windows:*** `\` (backslash)
Export/Import Buffer Value	Sets the BUFFER parameter, which is used when invoking the IMP and EXP utilities. Default value: `100000`
ISO language	Default value: `US`
Parallel Workers	Default value: `4`
Database Host	Server where database resides.
Database Port	Port for installing and launching the database.
Database SID	Name of the database that contains the ERP application schema.
Analyze Date Tolerance	Default value: `7` ***Informia Subset users:*** Subset will **not** perform a row count if there is an existing one that was created within this many days.
APPSORA.env	Default value: `APPSORA.env`

5 Click the **Save** button near the bottom of the page. You will be prompted for the following passwords - be sure to record your entries for later use: **AM_AGENT**, **ERP USER**, **SYSTEM**.

6 Click the **Configure** button. An ERP instance configuration job is submitted, and the page is refreshed. The **Latest Status** column will indicate the job's status. To update the **Latest Status** column, click the **Refresh** button at the bottom of the page.

7 When the job completes, the **Latest Status** indicates that configuration of the Integra Agent either succeeded or failed. If it failed:

   a. Click **Edit** to the right of **Integra Agent**.

   b. Click the **Remove** button near the bottom of the page.

   c. Select **Jobs > View Current Jobs** from the menu bar. Find the instance configuration job that was created when you clicked the **Configure** button, and view the job's logs/details to determine the cause of the failure.

   d. Resolve the cause of the failure.

   e. Select **Administrator > ERP Instance**, edit the instance definition, click **Edit** to the right of **Integra Agent**, and click the **Configure** button.

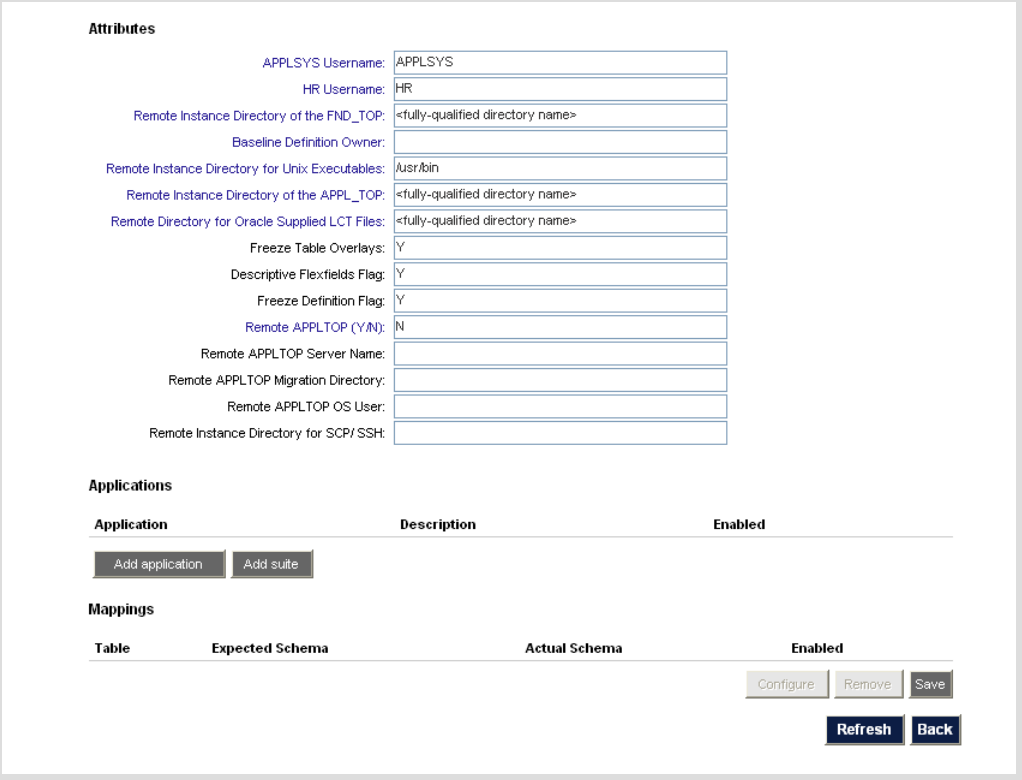
When you configure the Integra Agent, a job containing a number of programs is scheduled for immediate execution. Information about those programs is provided below. All programs must complete without error to fully prepare the instance for Integra products (other than the noted exception).

Configure Integra Agent [*instance name*]	The main program that calls other programs to perform specific tasks.
Drop SYSTEM Database Link	Drops the database link in the Integra Home schema to the SYSTEM user on the ERP instance, if one exists. **If one does not exist, this job will end in error. You can safely ignore this error.**
Create SYSTEM Database Link	Creates a database link for the Integra Home schema to connect as SYSTEM on the ERP instance. This link is used to create the Integra Agent schema on the ERP instance.
Create Integra Agent User	This user (or schema) will contain all Integra packages and tables.
Grant User Privileges and Roles	Grants necessary privileges and roles to the Integra Agent user.

Create Integra Agent Database Link	Creates a database link for the Integra Home schema user to connect to the Integra Agent schema on the ERP instance. This link is used to create Integra Agent objects.
Create Integra Agent Schema Objects	Creates tables and indexes used by Integra Agent.
Compile Target Packages	Compiles all packages used by the Integra Agent.
Install Java Classes and Allocate File Permissions	Installs Java components used by the Integra Agent.
Create ERP Database Link	Creates a database link for the Integra Home schema user to connect to the ERP User schema on the ERP instance.
Populate Integra Agent Seed data	Populates the Integra Agent objects' seed data.
Drop SYSTEM Database Link	Drops the database link from the Integra Home schema to the SYSTEM user on the ERP instance.

Configure Integra Apps

1 To the far right of **Integra Apps**, click **Edit**. The page is refreshed, and the lower portion displays fields for configuring Integra Apps (depending on your installation, tables might be listed in the **Mappings** section).



2 Enter the following information:

Attributes

APPLSYS Username	***Oracle E-Business Suite users:*** ERP instance's **APPLSYS** username. ***PeopleSoft Enterprise users:*** ERP instance's **SYSADM** username.
HR Username	***Oracle E-Business Suite users:*** ERP instance's **HR** username.
Remote Instance Directory of the FND_TOP	***Oracle E-Business Suite users:*** ERP instance's **$FND_TOP** directory.
Baseline Definition Owner	Integra Apps User who will own the Snapshot and Change Tracking definitions that will be generated automatically when you complete this configuration process.

Remote Instance Directory for Unix Executables	***Oracle E-Business Suite users:*** Directory where Unix command executables are located (usually `/bin`).
Remote Instance Directory of the APPL_TOP	***Oracle E-Business Suite users:*** ERP instance's **$APPL_TOP** directory.
Remote Directory for Oracle Supplied LCT Files	***Oracle E-Business Suite users:*** Directory that stores the ERP instance's Oracle-supplied **.lct** files.
Freeze Table Overlays	If you have moved or renamed any ERP tables or views from their defaults, enter `N` to map them (which will make them visible to Integra Apps). The default value is `Y`.
Descriptive Flexfields Flag	***Oracle E-Business Suite users:*** Enter `Y` to capture all specific values entered in every Descriptive Flexfield of every setup Object in this ERP instance. The default value is `N`. **Note:** To report on the structure or definition of a Descriptive Flexfield, use the **System Administration** application's **Descriptive Flexfield Segments** object.
Remote APPLTOP (Y/N)	***Oracle E-Business Suite users:*** Enter `Y` if your **APPL_TOP** is on a different physical server than the ERP instance. The default value is `N`.
Remote APPLTOP Server Name	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Name of server.
Remote APPLTOP Migration Directory	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Migration directory.
Remote APPLTOP OS User	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Operating system user.
Remote Instance Directory for SCP/SSH	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Directory where the **scp** and **ssh** commands are located; sometimes differs from **Remote Instance Directory for Unix Executables** above.

3 Click **Save**.

4 Specify the applications that Integra Apps will be able to monitor on this ERP instance. You can:

Click **Add application...** to add applications one-by-one, clicking **Save** after selecting each application.

Click **Add suite...** to add all applications in a product suite.

5 Click the **Configure** button. You will be prompted for the following passwords - be sure to record your entries for later use: **AM_AGENT**, **ERP USER** and **SYSTEM**.

6 Once you supply the passwords, an ERP instance configuration job is submitted, and the page is refreshed. The **Latest Status** column will indicate the job's status. To update the **Latest Status** column, click the **Refresh** button at the bottom of the page.

7 When the job completes, the **Latest Status** indicates that configuration of the Integra Apps either succeeded or failed. If it failed:

a. Click **Edit** to the right of **Integra Apps**.

b. Click the **Remove** button near the bottom of the page.

c. Select **Jobs > View Current Jobs** from the menu bar. Find the instance configuration job that was created when you clicked the **Configure** button, and view the job's logs/details to determine the cause of the failure.

d. Resolve the cause of the failure.

e. Select **Administrator > ERP Instance**, edit the instance definition, click **Edit** to the right of **Integra Apps**, and click the **Configure** button.

When you configure Integra Apps, a job containing a number of programs is scheduled for immediate execution. Information about those programs is provided below. All programs must complete without error to fully prepare the instance for Integra products (other than the noted exception).

Configure Integra Apps	Main configuration program that calls all other programs to perform specific tasks.
Create Target Schema Objects (OA 10.7-specific)	***Oracle E-Business Suite 10.7 users:*** Creates Integra Apps Views and Synonyms.
Create Target Schema Objects (OA 11.0.3 and 11i-specific)	***Oracle E-Business Suite 11.0.3 and 11i users:*** Creates Integra Apps Views and Synonyms.

Create Target Schema Objects	Creates Integra Apps database objects (e.g., tables, sequences, and indexes).
Create tick_ct_target_pkg	Creates Integra Apps Change Tracking package specification and body.
Run Target Synchronize	Copies metadata to the ERP instance and verifies that there are no conflicts with the Oracle Data Dictionary.

8 If Integra Apps cannot locate tables or views due to customizations, or locates the same table name within more than one schema, it will populate the **Mappings** section.

a. ***For each item:*** Click **Edit** to the item's right and specify the object that Integra Apps should use. If two schemas own tables with the same name, enable only one table.

b. Click the **Configure** button again and follow the preceding steps.

Create users, roles and security groups

Typically, each person who uses Integra has his or her own username and password. This lets you control each user's login access, roles and permissions. **Roles** determine the Integra menu items the user sees; **Permissions** control the ERP data the user can access. Rather than granting permissions directly to users, you grant them to **Security Groups**, then assign users to the security groups. Users who do not belong to any security groups will not be able to access any ERP data. Similarly, users who do not have any roles will not see any Integra menu items.

If you are configuring Integra for the first time, use Create Integra User to create the users and roles specified in your implementation plan. Then use To add a Security Group to create security groups and assign your users to them.

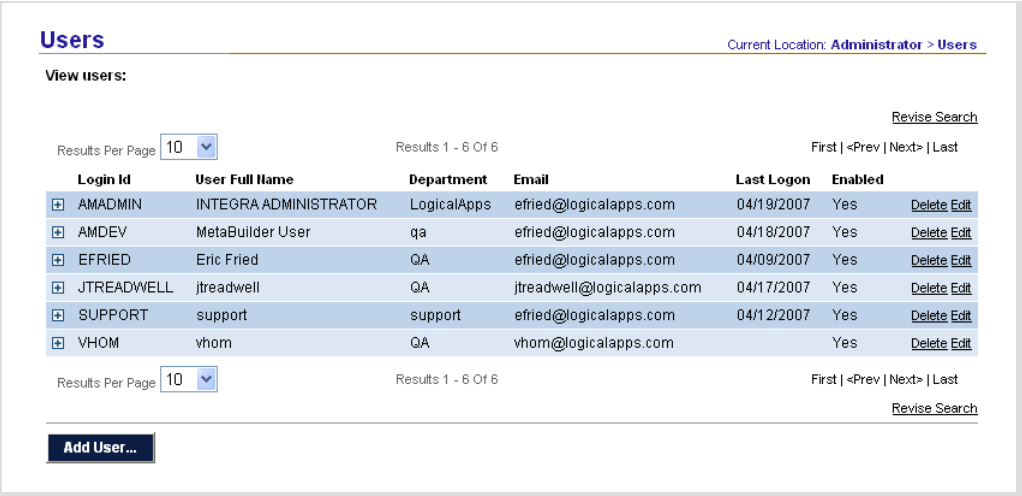
At minimum, you must:

- Create at least one new user (unless you will be the only person using Integra).
- Assign each possible role to at least one user (unless there are roles that you do not wish to assign to any user).
- Create at least one security group.
- Assign each user to at least one security group (except for users who should not have access to any ERP data).

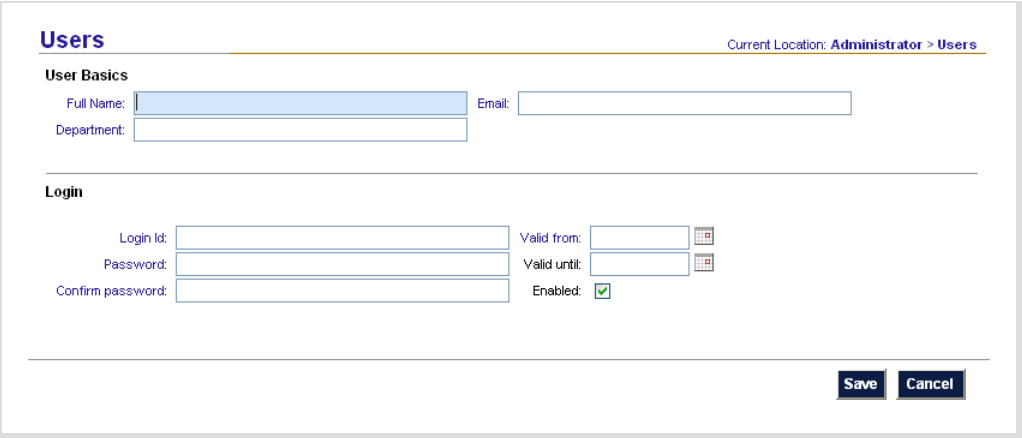
The simplest possible setup would be to assign all roles to yourself, create one security group with access to all ERP data, and assign yourself to that group.

Create Integra User

1 Select **Administrator > Users** from the menu bar. The **Users** page appears.



2 Click the **Add user...** button. A blank definition appears.



3 Enter the following information:

User Basics

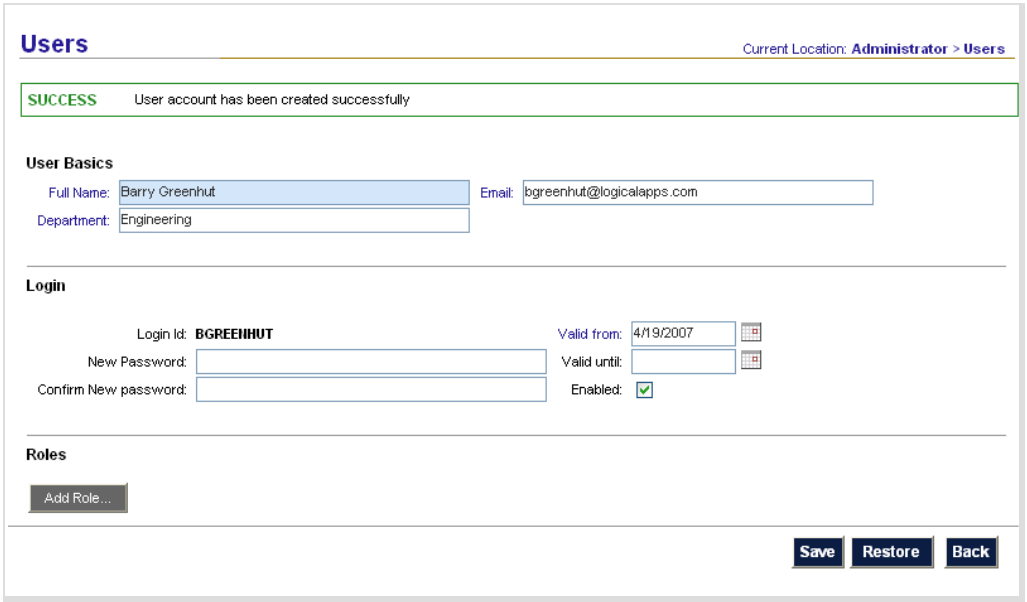
Full name	Integra user's full name. The name may consist of letters, spaces, ' (apostrophes), and – (dashes); other characters are not allowed.
Department	User's department or organization. The department may consist of letters, digits, spaces, and - (dashes); other characters are not allowed.
Email	The user will receive alerts at this address.

Login

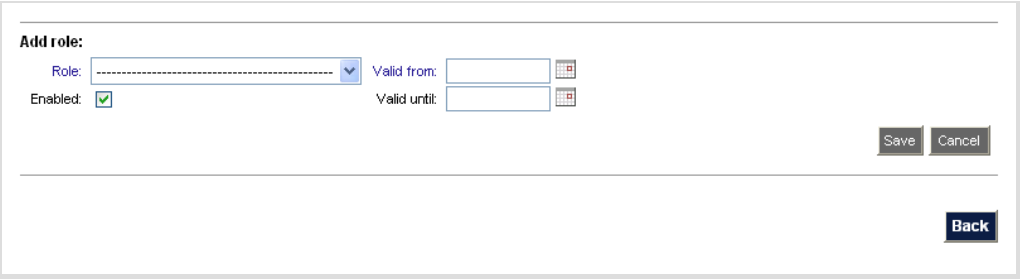
Login ID	User's Integra Login ID. The ID may consist of letters and digits only; spaces and other characters are not allowed.

Password	Initial password for the new user (40 character length limit). This data will be encrypted before it is stored. Not required if LDAP Authentication mode is used. See System Profile (p. 81) for more information. The following characters are not allowed: ! | / = $ & @ " ' ` ,
Confirm password	Re-enter the password to confirm
Valid from	First day user can log in
Valid until	*Optional:* Last day user can log in
Enabled	Enables the user to log in; if not checked, the user cannot log in.

4 Click **Save**. The page is refreshed.



5 Click **Add role...**. The **Roles** page appears.



6 Enter the following information to assign roles to the user:

Add role:

Role	Select a role from the drop-down list.
Enabled	Enables this role for this user.
Valid from	First day the user will have this role.
Valid until	*Optional:* Last day the user will have this role.

The following roles are available for Integra Apps 5.1.

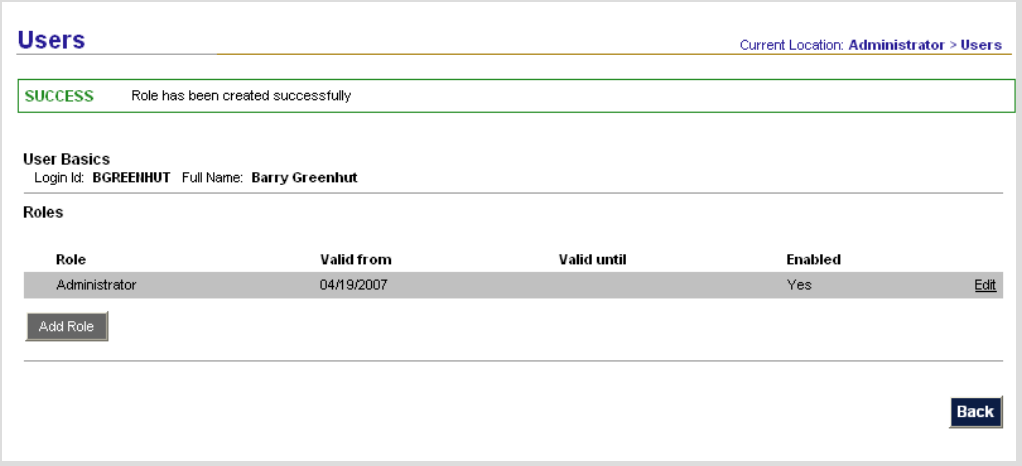
For the convenience of administrators of previous versions, we also show 4.x roles here; your 4.x users' roles have been replaced with the corresponding 5.1 roles.

Apps 5.1 Role		***Apps 4.x Role***
Apps User	Create and edit Snapshot, Change Tracking, and Migration definitions.	*Apps User*
Snapshot Scheduler	Schedule Snapshot definitions for execution, and compare Snapshot occurrences.	*Apps User*
Change Tracking Scheduler	Schedule Change Tracking definitions for execution.	*Change Tracking Manager*
Migration Scheduler	Schedule Migration definitions for execution.	*Migration User*
Apps Developer	Create and edit Templates.	*Template Builder*

Apps 5.1 Role		*Apps 4.x Role*
Apps Metadata Manager	Use MetaBuilder.	*Metabuilder User*
	This 4.x role is not converted (all administration is conducted by the Administrator role):	*Apps Administrator*

The **Administrator** role gives the user access to all **Administrator** menu bar items.

7 Set all options and click the **Save** button. The page refreshes, displaying a summary of the user's roles.



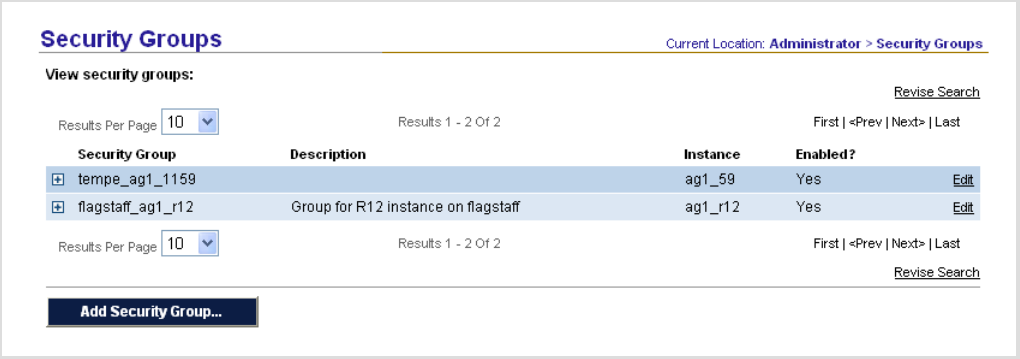
8 To add another role, click **Add role...** again. The new role is added when you click the **Save** button. You can add as many roles as you wish.

9 Click the **Back** button. The definition page reappears.

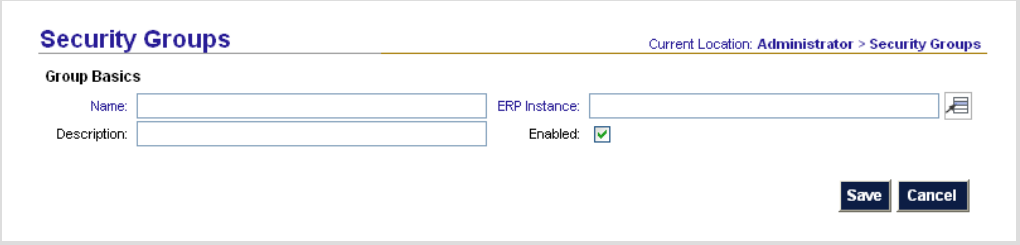
10 Click the **Save** button. The new definition is saved and the **Users** page appears.

To add a Security Group:

1 Select **Administration > Security Groups** from the menu bar. The **Security Groups** page appears.



2 Click **Add security group...**. A blank definition appears.

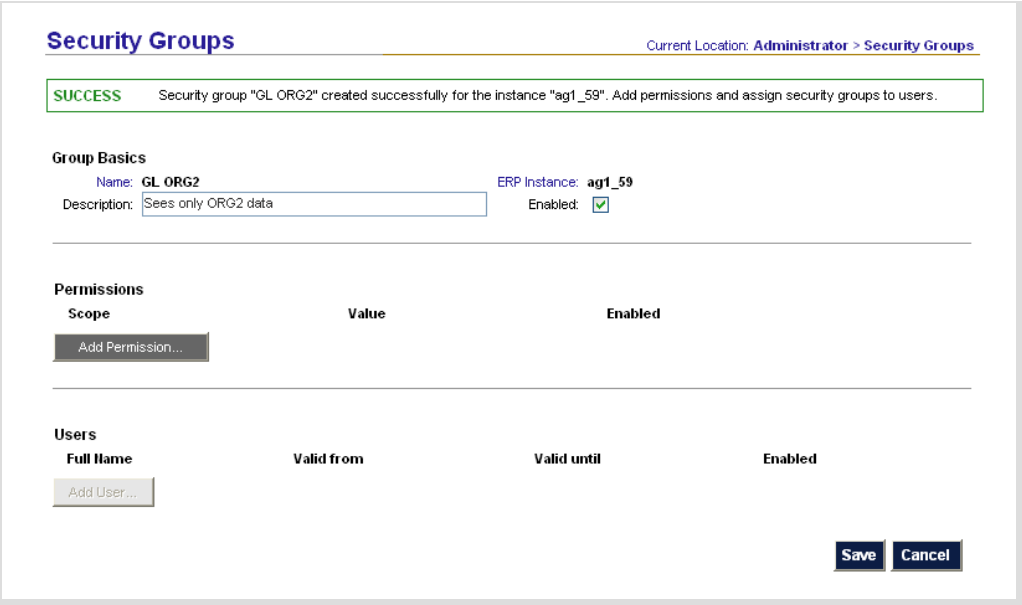


3 Enter the following information:

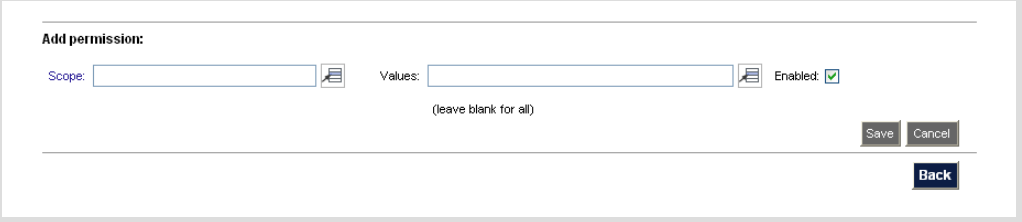
Group Basics

Name	The group's name.
Description	A short description of the group.
ERP Instance	The ERP instance that the group can work with.
Enabled	When checked, the group is active.

4 Click **Save**. The page is refreshed, displaying the **Permissions** and **Users** sections.



5 Click **Add Permission...**. The **Permissions** page appears.

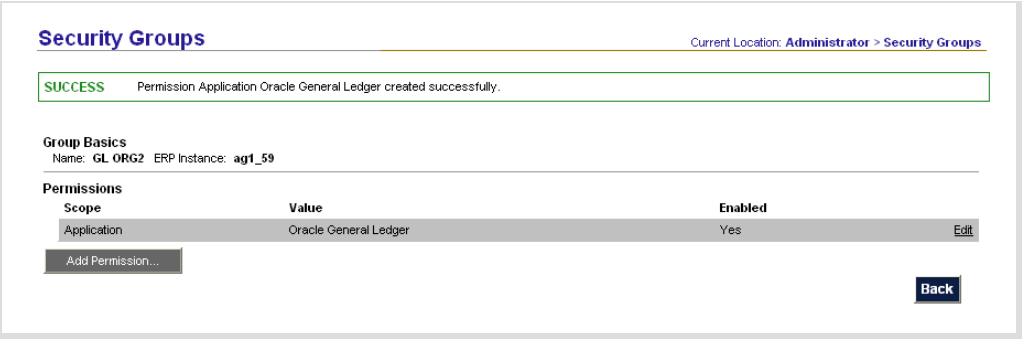


6 Enter the following information:

Add permission:

Scope & Values	Access is granted to data where *Scope IS IN Values*. To match all values, leave **Values** blank.
Enabled	When checked, the permission is active.

7 Click **Save**. The page refreshes, displaying a summary of the Permission.

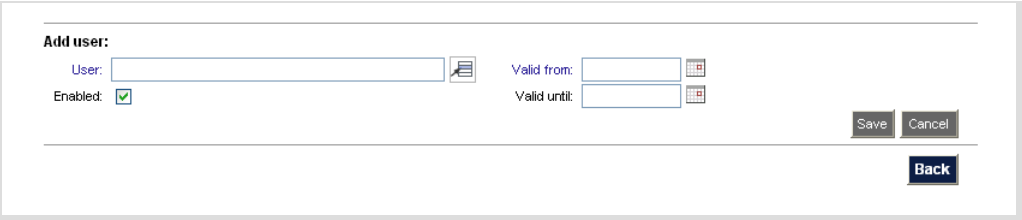


8 To add another permission, click **Add permission...** again. The new permission is added when you click **Save**. You may add as many permissions as you wish.

> Before you can add users to a group, you must add a permission for each **Scope**.
>
> Add all desired permissions before adding the first user. Once you add a user, you cannot change the group's permissions.

9 When you are done adding permissions, click **Back**. The definition page reappears.

10 Click **Add user...**. The **Users** page appears.



11 Enter the following information:

Add user:

User	Select a user from the pop-up window.
Enabled	When checked, this user is an active member of the group.
Valid from	First day the user will be a member of the group.
Valid until	*Optional:* Last day the user be a member of the group.

12 Click **Save**. The page refreshes.

13 To add another user, click **Add user...** again. The user is added when you click **Save**. You may add as many users as you wish.

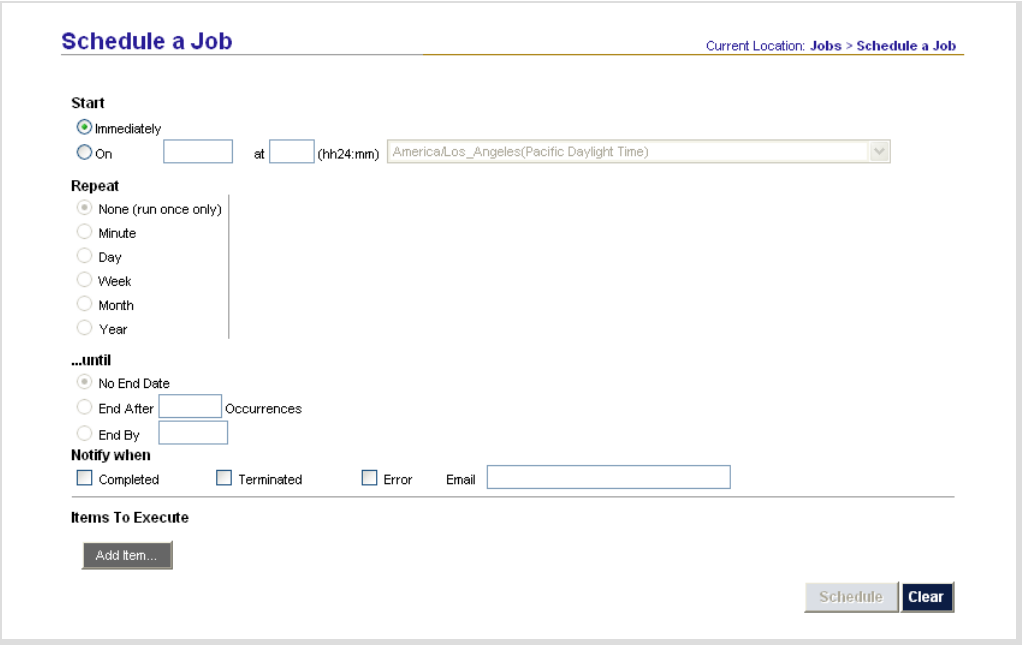
14 Click the **Back** button. The definition page reappears.

15 Click **Save**. The new definition is saved and the **Security Groups** page appears.

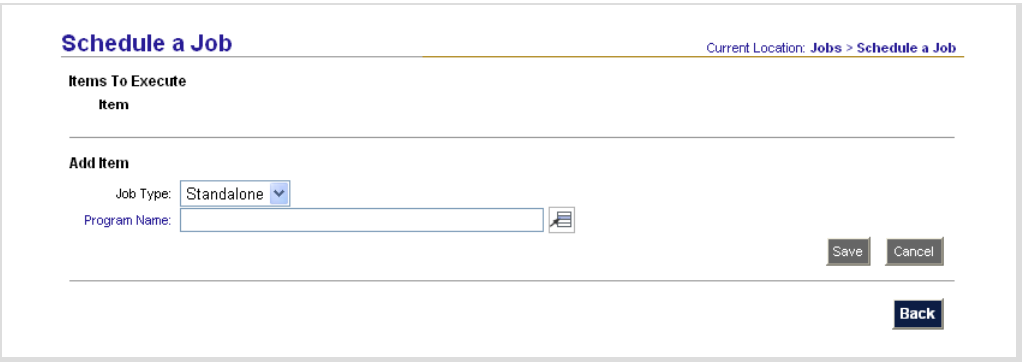
Schedule Change Tracking Transfer

The **Change Tracking Transfer** program transfers change tracking data from the ERP instances to Integra Apps. When users view Integra Apps' Change Tracker, they see only the data that has been transferred by this program. You are advised to schedule this program to run as frequently as possible:

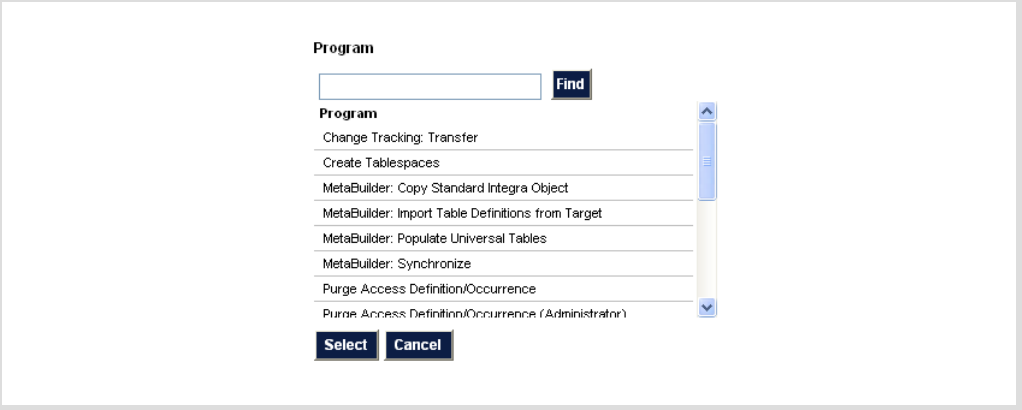
1 Select **Jobs > Schedule a Job** from the menu bar. The **Schedule a Job** page appears.



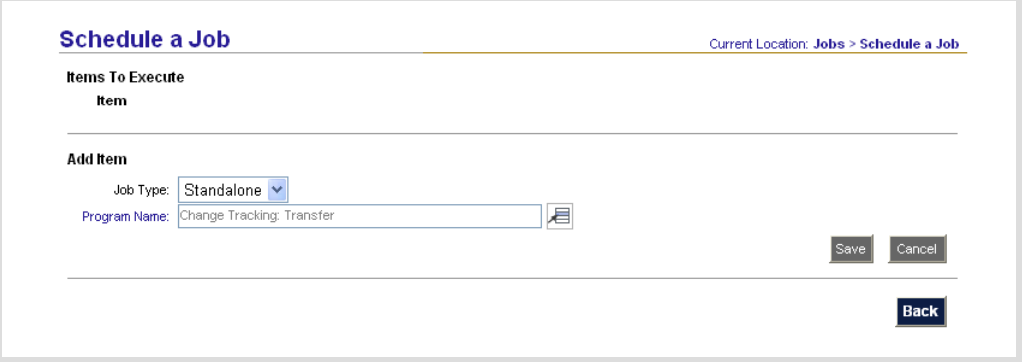
2 Click **Add item...**. The **Add Item** page appears.



3 Click to the right of **Program Name**. An LOV (List of Values) window appears, displaying all standalone programs.



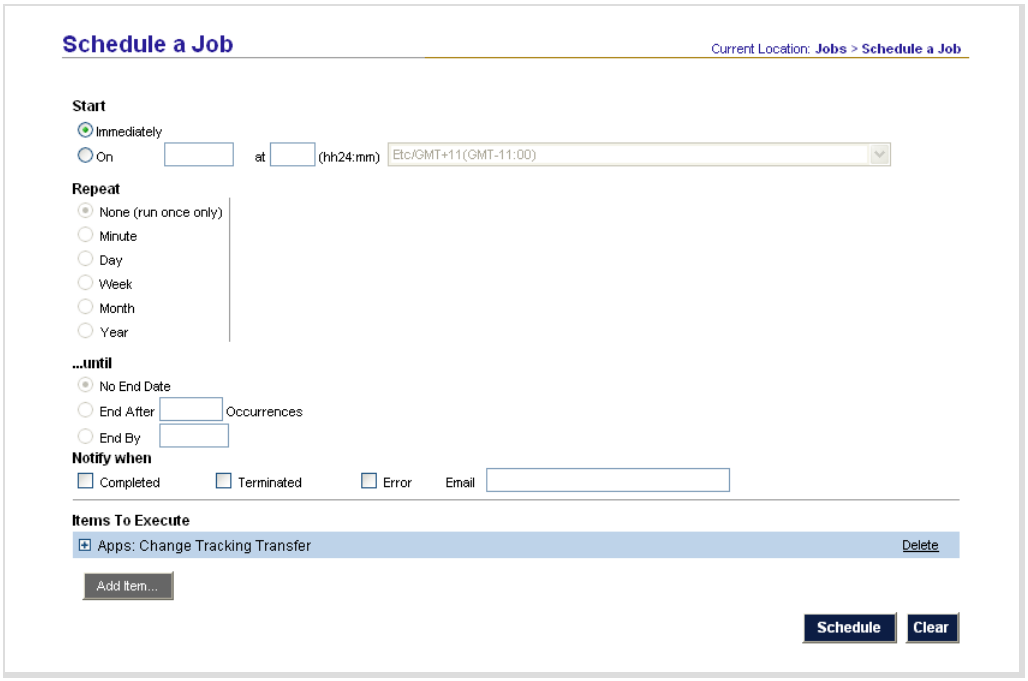
4 Highlight **Change Tracking: Transfer** and click the **Select** button. The LOV window disappears, and fields for configuring the program appear.



5 Click **Save**.

6 Click **Back**. The Schedule a Job page reappears.

7 Set all scheduling options as desired. You are advised to choose the **Repeat: Minute** option for frequent updating (perhaps every 60 minutes or less).



8 Click **Schedule**.

How to...

In This Section

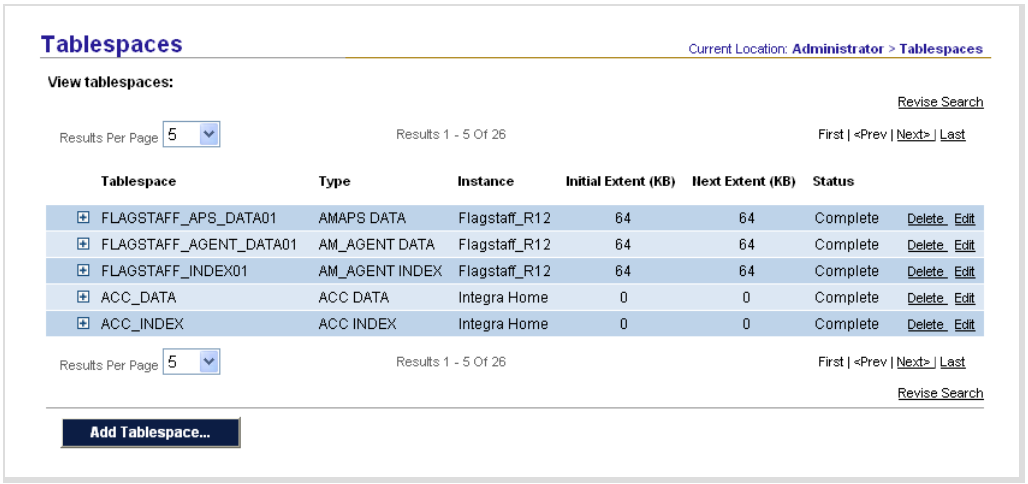
Create tablespaces

Integra requires the creation and maintenance of several tablespaces. Use the **Tablespace Workbench** to work with tablespaces.

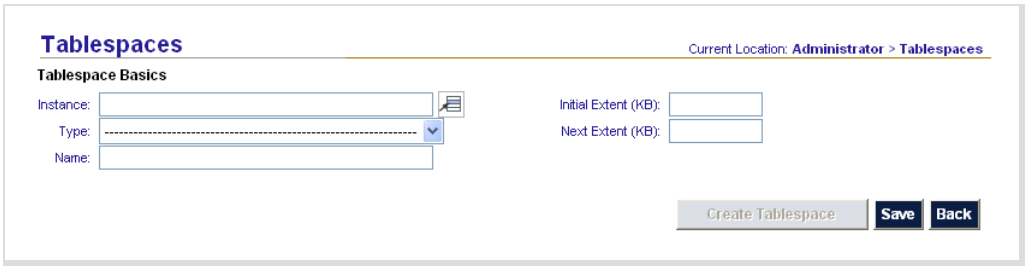
> All tablespaces created by Integra are locally managed.

Create a tablespace

1 Select **Administrator > Tablespace Workbench** from the menu bar. The **Tablespace Workbench** appears.



2 Click **Add Tablespace...**. The **Tablespaces** page appears.



3 Enter the following information:

Tablespace Basics

Instance	Where the tablespace will reside.
Type	Choose the appropriate type.

Name	The tablespace's name.
Initial extent	Size of the tablespace's initial extent.
Next extent	Size of the tablespace's subsequent extents.

4 Click **Save**. The page refreshes, with a new **Datafiles** section below **Tablespace Basics**.

5 Enter the following information:

Datafile Basics

Datafile Name	The datafile's name, including the file system directory path for where the datafile will be stored.
Max Size	The maximum size of the datafile. Its initial size will be 10% of the amount specified here; it will be allowed to AUTOEXTEND in increments of 10% or 200MB, whichever is less, until reaching the size entered here.

6 Click **Save**. The page refreshes.

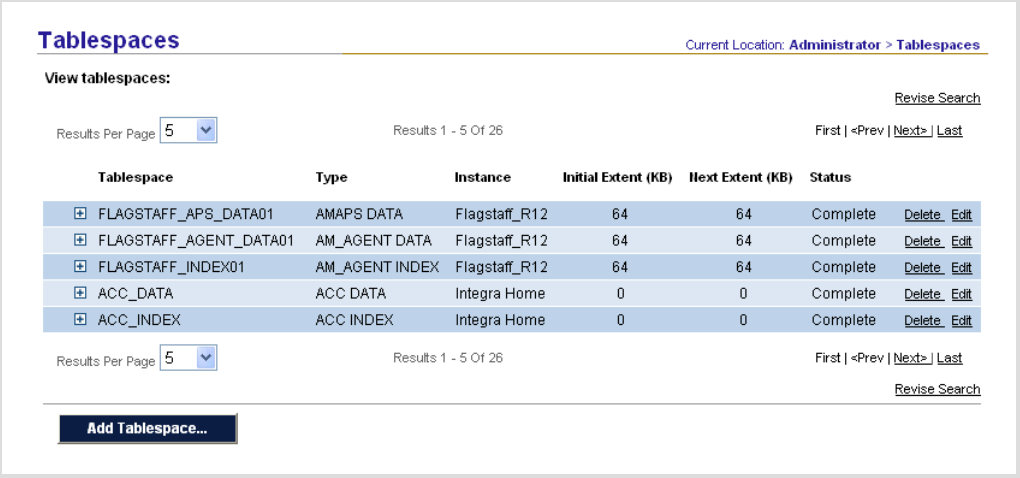
7 To add another datafile to the tablespace, fill in the fields and click **Save** again. You may add as many datafiles as you wish.

8 Click **Create Tablespace**. A job to create the tablespace will be submitted for immediate execution; you can monitor it on the **Jobs > View Current Jobs** page.

9 Repeat the preceding steps to create more tablespaces.

View or edit a tablespace

1 Select **Administrator > Tablespace Workbench** from the menu bar. A list of tablespaces appears.



2 Click **Edit** to the right of the tablespace's name.

Add a datafile to an existing tablespace

1 Edit the tablespace.

2 Fill in the **Datafile Name** and **Max Size** fields.

3 Click **Save**. The page refreshes.

4 Click **Alter Tablespace** to add the datafile.

Schedule definitions and work with jobs

Many actions you take when using Integra, such as scheduling a definition, cause the creation of **jobs**. Each job is a sequence of one or more **programs**. (When you schedule a definition, you're really scheduling a job that contains the sequence of programs that you saw when you created or edited the definition.)

Because the programs are executed in a sequence, if a program cannot be completed, the remaining programs in the sequence will not be executed.

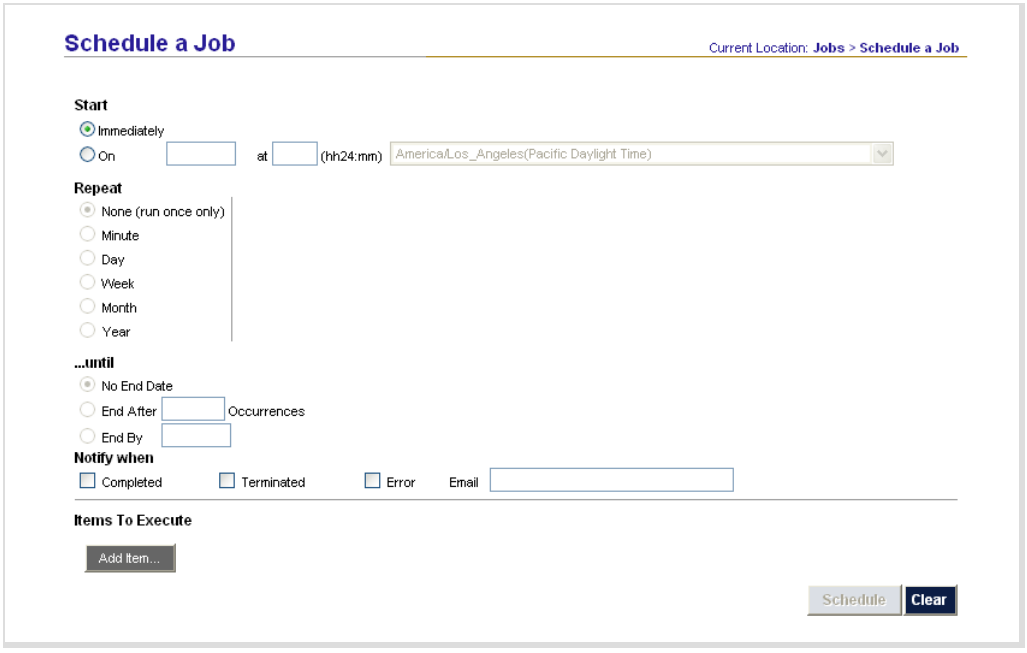
Each job is scheduled to run on one or more occasions. See Schedule a job (p. 35) for more information.

All jobs that are scheduled for the future are listed on the **View Future Jobs** page (select **Jobs > Future Jobs** from the menu bar). All jobs that are being executed, or that already have been executed, are listed on the **View Current Jobs** page (**Jobs > Current Jobs**). See View a job's status or details (p. 39) for more information.

Schedule a job

1 Prepare all definitions to be included in the job.

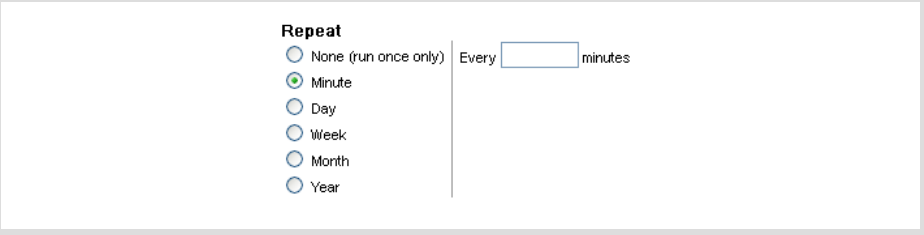
2 Go to the **Schedule a Job** page (either select **Jobs > Schedule a Job** from the menu bar, or click **Schedule...** while editing a definition or performing an administrative function).



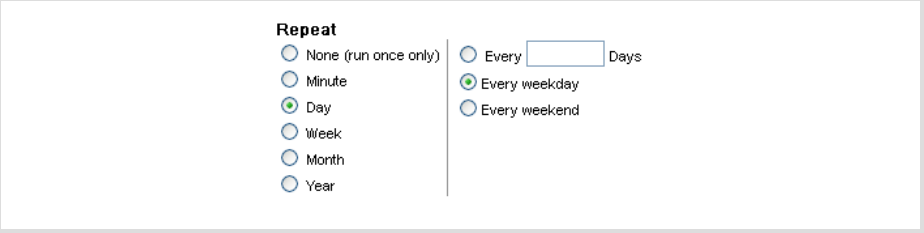
3 In the upper part of the page, set all schedule options. The default is to run the job once, immediately.

4 If you schedule the job to start **On...**, you can choose these **Repeat** options:

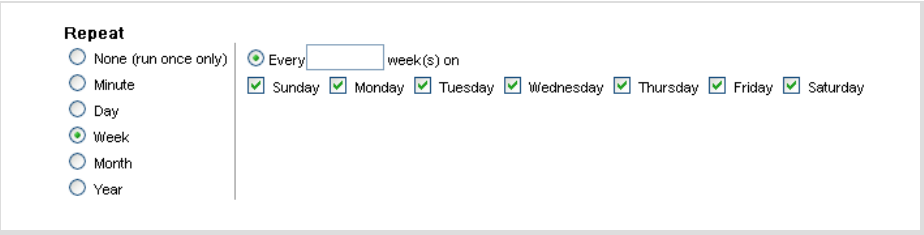
- **Minute** - repeat every *n* minutes



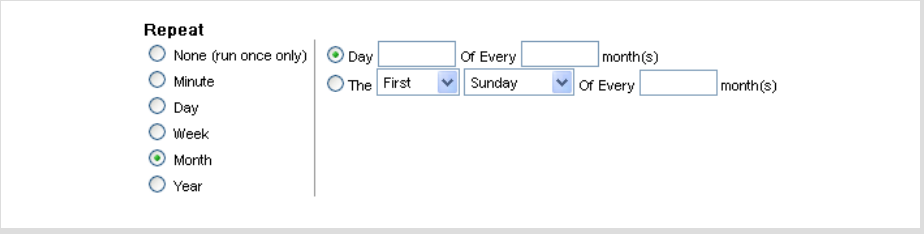
- **Day** - repeat every *n* days, every weekday, or every weekend day



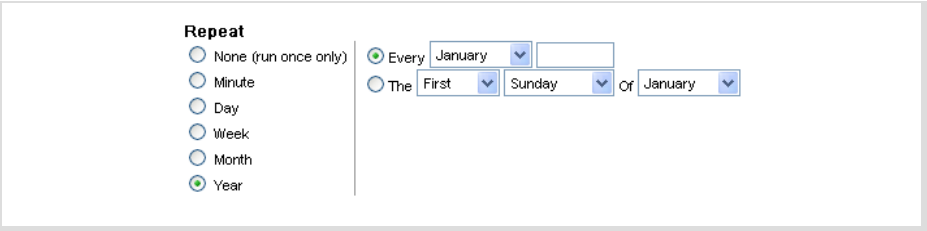
- **Week** - repeat every *n* weeks on specified days



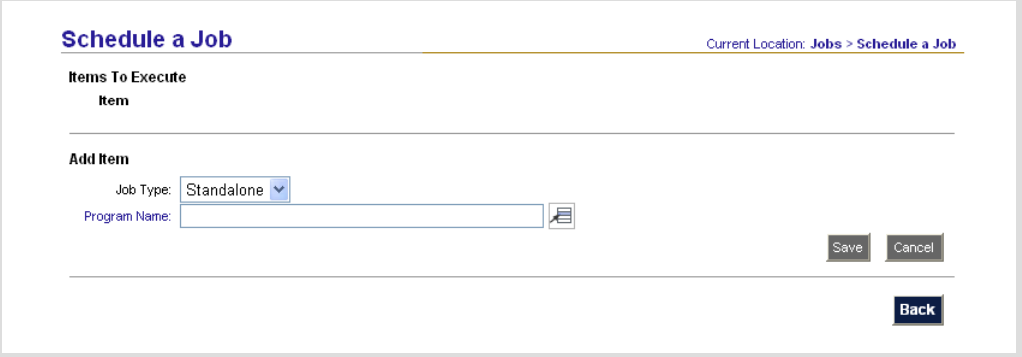
- **Month** - repeat every *n* months on a specified day



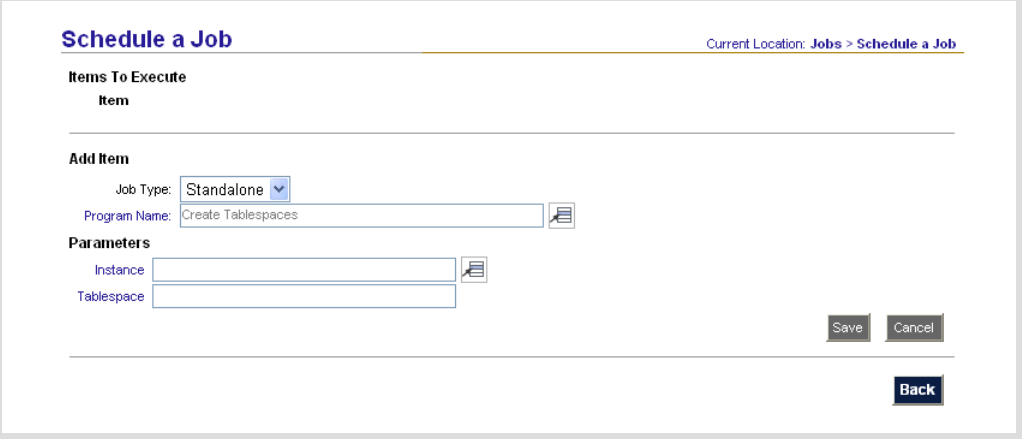
- **Year** - repeat every year on the specified day



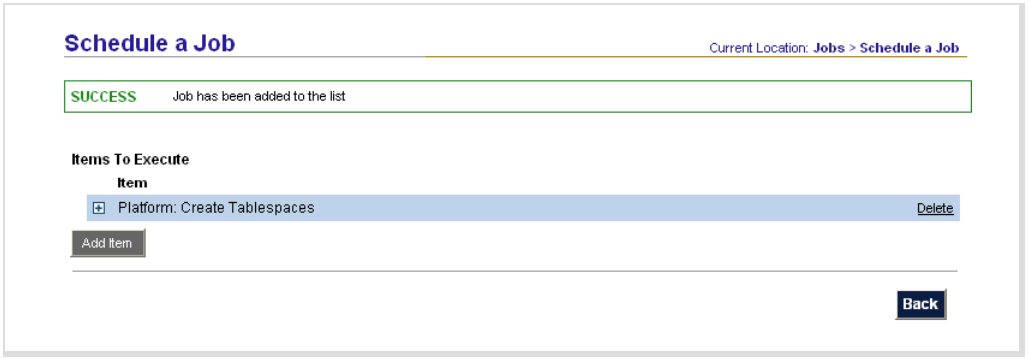
5 In the lower part of the page, include all items to be executed, in the desired order. When you click **Add Item...**, the **Add Item to Job** page appears.



When you select a **Program Name**, additional fields might appear.

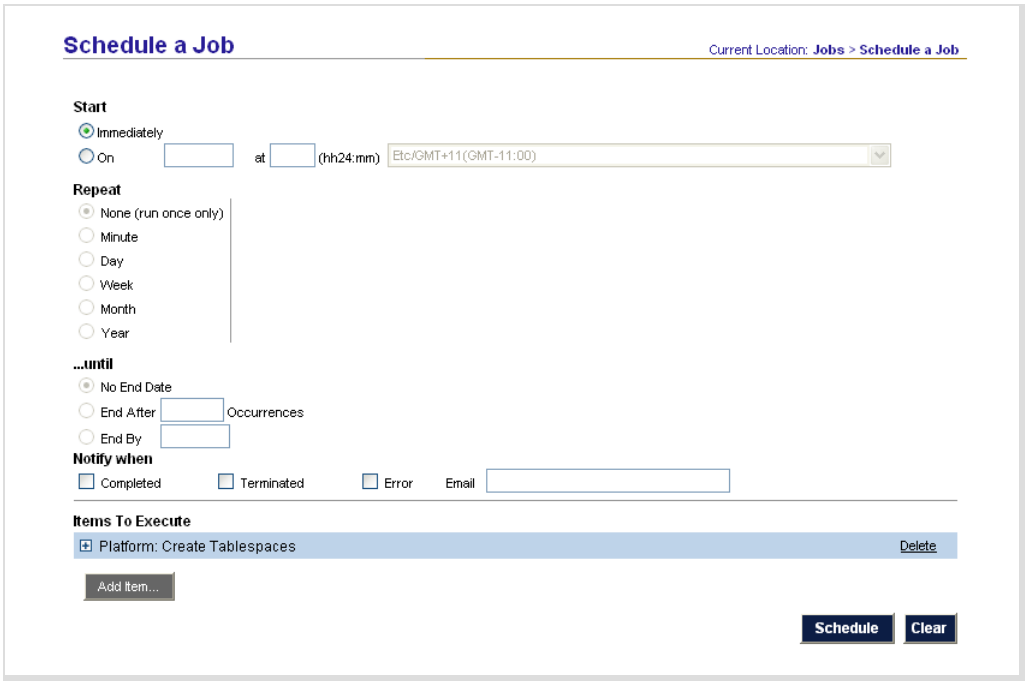


Set all additional items, and click **Save** to add the item to the job.

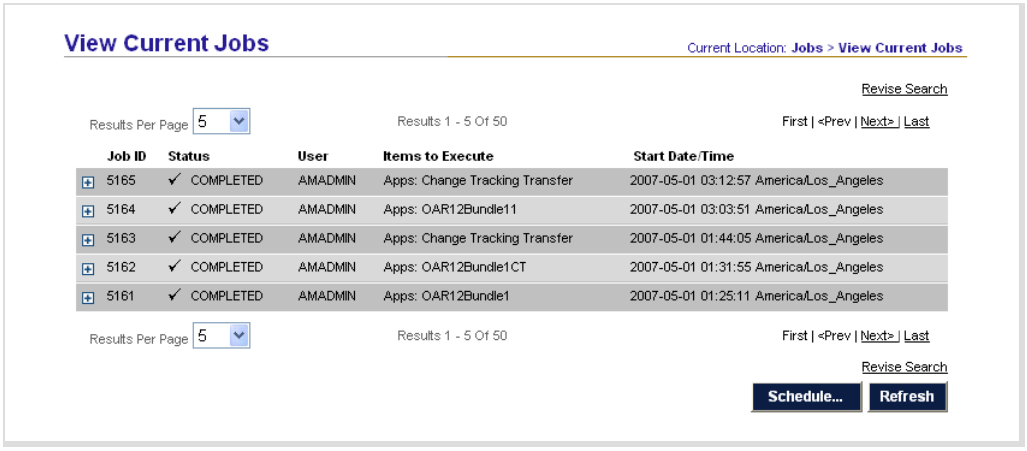


To add another item to execute, click **Add Item**. You can add as many items as you like.

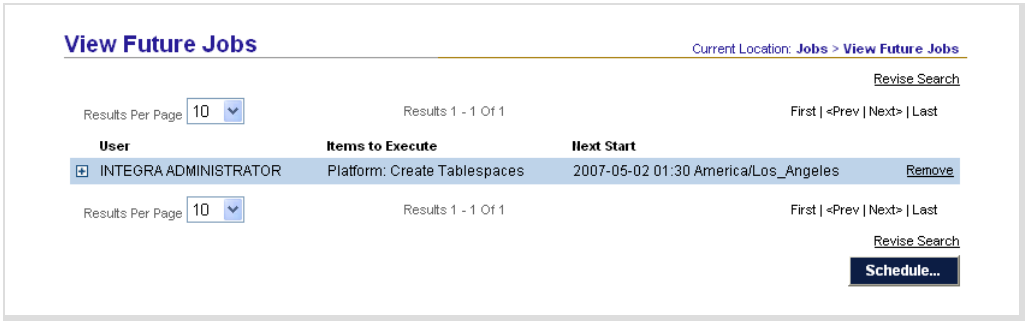
When you are done adding items, click **Back**. The **Schedule a Job** page reappears.



6 When you are done, click **Schedule**. If the job is scheduled to run immediately, the **View Current Jobs** page appears.



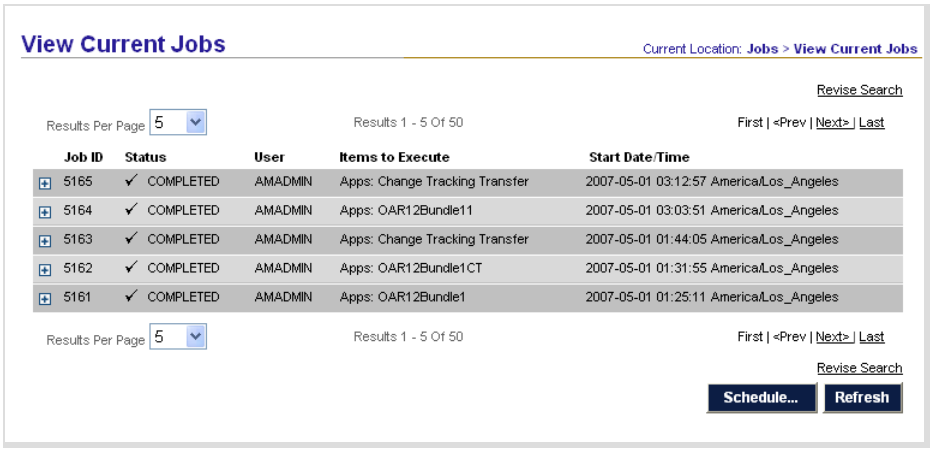
If the job is scheduled to run in the future, the **View Future Jobs** page appears.



View a job's status or details

To see the status of:

A job that's running now, or that ran in the past: Select **Jobs > View Current Jobs** from the menu bar. The **View Current Jobs** page appears; the most recently scheduled jobs are listed, sorted by Job ID, newest ID first.



A job that will run in the future: Select **Jobs > View Future Jobs** from the menu bar. The **View Future Jobs** page appears; jobs are sorted by start date/time, most imminent jobs first.

In either case, if you don't see the job you're looking for, you can:

Increase the number of **Results Per Page**

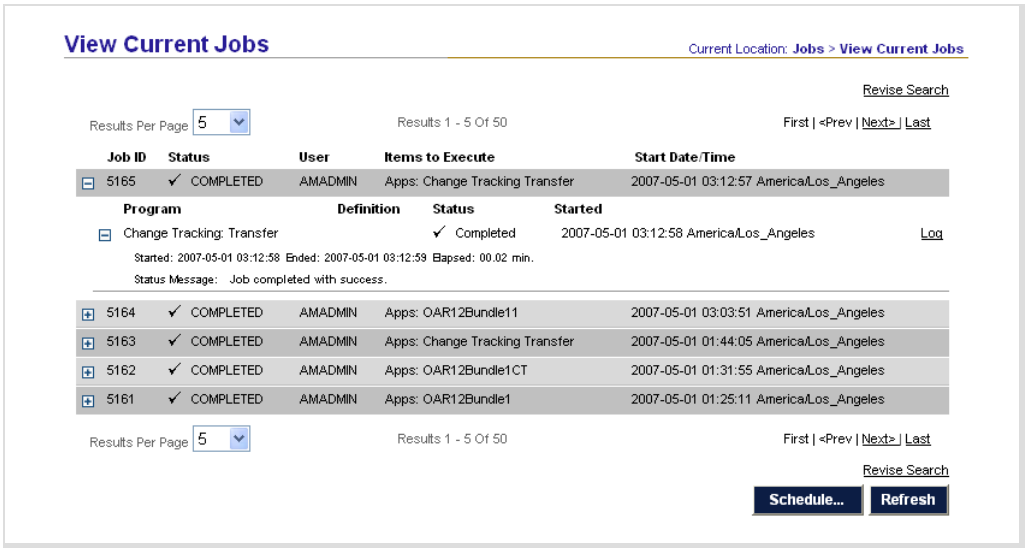
Click **Next >** to see the next page of jobs

Click **Revise Search**, fill in the search form and click the **Search** button

On the **View Current Jobs** page, the following job statuses are possible:

Statuses are not displayed on the **View Future Jobs** page because all jobs have the same status (**Pending**).

Click ⊞ to see the job's details and control buttons.



Depending on the job's status, some of these buttons could appear:

If the job's execution has been paused: Click **Resume Job** to resume execution of the paused job.

If the job is being executed, or if its execution has been paused: Click **Terminate Job** to cancel the job. *Note:* Not all jobs can respond to a termination request.

For current/past jobs, one or both of the following links will appear to the right of each program:

Click **Log** to view log information, including error messages.

Click **Details** to view additional details about the program's execution, and all reports generated by the program.

Cancel the current execution of a job

1. Find the job on the **View Current Jobs** page.
2. Click the job's **Terminate job** button. The job's status changes to **Terminating**. Once the currently running program terminates or completes, the job's status changes to **Terminated**.

Cancel future execution of a job

1. Find the job on the **View Future Jobs** page.
2. Click the job's **Remove** link.

Examine a job's Occurrences

1. Find the job on the **View Current Jobs** page.
2. Click **Details** to the right of any of the definition's programs.

Change a job's schedule

1. Cancel future execution of the job (see Cancel future execution of a job (p. 41)).
2. Schedule a new job.

Schedule a job again (after all occurrences have completed)

1. Schedule a new job

Edit a definition that is scheduled for use in future jobs

1. On the **View Future Jobs** page, remove all jobs that use the definition.

2 Edit the definition.

3 Reschedule the jobs that you removed in Step 1.

Add/remove definitions or programs from a job, or change their order of execution

1 Cancel future execution of the job.

2 Schedule a new job.

Reconfigure an instance

You will need to reconfigure the Integra Agent on ERP instances when either:

The Integra Agent no longer exists (e.g., because you have refreshed the instance).

You have changed the passwords on the instance for any of these users: ERP user, SYSTEM or AM_AGENT.

When a password has changed

If you have changed the password for the ERP user, SYSTEM or AM_AGENT:

1. Select **Administrator** > **ERP Instances** from the menu bar. A list of instances appears.
2. Click **Edit** to the right of the instance where the password changed.
3. For each component installed - but not Integra Agent:
    a. To the right of the component's name, click **Edit**.
    b. Click **Remove** to remove the component from the business application instance. The page refreshes, and the **Latest Status** column is updated.
    c. If **Latest Status** indicates that removal of the component failed:
        i. Select **Jobs > View Current Jobs** from the menu bar. Find the instance removal job that was created when you clicked the **Remove** button, and view the job's logs/details to determine the cause of the failure.
        ii. Resolve the cause of the failure.
        iii. Repeat the steps above.
4. Repeat Step 3 for the Integra Agent.
5. Reconfigure the Integra Agent with the new passwords.
6. Reconfigure each component.

When an instance has been refreshed

During the Integra testing cycle, you may be required to refresh your ERP instance. This usually means you must reconfigure the Integra Agent and recreate its tablespaces.

The initial steps must be performed **before** refreshing the ERP instance.

BEFORE refreshing the ERP instance:

1. Select **Administrator** > **ERP Instances** from the menu bar. A list of instances appears.
2. Click **Edit** to the right of the instance where the password changed.
3. For each component installed - but not Integra Agent:
    a. To the right of the component's name, click **Edit**.
    b. Click **Remove** to remove the component from the business application instance. The page refreshes, and the **Latest Status** column is updated.
    c. If **Latest Status** indicates that removal of the component failed:
        i. Select **Jobs > View Current Jobs** from the menu bar. Find the instance removal job that was created when you clicked the **Remove** button, and view the job's logs/details to determine the cause of the failure.
        ii. Resolve the cause of the failure.
        iii. Repeat the steps above.
4. Repeat Step 3 for the Integra Agent.

You may now refresh the ERP instance.

AFTER refreshing the ERP instance:

5. Make note of the datafile information for each Integra tablespace on the ERP instance. This includes:

    AM_AGENT_D

    AM_AGENT_X

    All component-specific tablespaces

6. Delete those tablespaces.
7. Recreate those tablespaces.
8. Reconfigure the Integra Agent.
9. Reconfigure each component.

Your refreshed ERP instance is now reconfigured to run the appropriate Integra components.

Maintain the Integra Home schema

Overview

The Integra Home schema can affect many operations within your organization, because it stores information about:

- The Integra system as a whole
- Each Integra component and ERP instance

When creating your maintenance plans for the Integra Home database, be sure to consult all:

- Integra Administrators
- IT and business unit managers who use or rely on Integra
- Functional and technical superusers who use Integra
- Designated ERP application end users
- Database administrators

These decision-makers must agree, among other things, whether to retain or purge the information in the Integra Home during maintenance.

Your DBAs should monitor the size of the Integra Home tables and indexes, to ensure sufficient tablespace capacity to fulfill your organization's needs.

Database passwords

Integra's operation is impacted when any these passwords are changed:

Integra Home schema (AMHOME)	The Integra Home schema database password on your Integra Home instance
Integra Agent schema (AM_AGENT)	The Integra Agent schema database password on any of your ERP instances
Integra Public schema (AMHOME_PUBLIC)	***If your Integra UI is served by Oracle Forms and Reports servers:*** The Integra Public schema database password on your Integra Home instance
ERP instance system	System password for the ERP instance; e.g., System/Manager.

ERP schema password	Typical values: ***Oracle E-Business Suite:*** `APPS` ***PeopleSoft Enterprise:*** `SYSADM` ***Siebel CRM Applications:*** `SIEBEL`

Once you understand the effect of database password changes on Integra, you will be able to change passwords on schedule or on demand, while keeping Integra available without disruption.

> However, you are advised to change these database passwords only during scheduled downtime, or after you have broadcast an announcement indicating that Integra will be temporarily unavailable.

Change the Integra Home schema password

To ensure that the Integra Home database remains secure, you may choose to change the Integra Home schema database password from time to time. Since end users do not typically perform ad-hoc queries of the Integra Home database, it is unusual for the Integra Home schema database password to be disclosed to anyone other than the database administrator. However, it is easy to change the Integra Home schema password when necessary:

1. Log into Product Download Center.
2. Click **Tools**.
3. Click **Alter Integra Home Schema Password - for AM Platform 4.5 and newer**:

   The file **amhome_reset_45.zip** is downloaded.

4. Extract the file **amhome_reset_45.sql** from **amhome_reset_45.zip.**
5. Choose a new password for the Integra Home schema.

> The password's maximum length is 30 characters. These characters are not allowed: `!` `|` `/` `=` `$` `&` `@` `"` `'` `` ` `` `,`

6 Connect to the Integra Home database as `SYSTEM`, and change the Integra Home schema database password. You can use this syntax:

```
ALTER USER AMHOME IDENTIFIED BY <NEW_PASSWORD>;
```

...or use Oracle Enterprise Manager or other methods to change the password, according to your usual database administration procedure.

7 Run this script:

```
@am_home_encrypt.sql
```

...against the Integra Home schema. This script will prompt you for connection information (including the new password that you designated).

Change the Integra Agent schema password or ERP schema password

To ensure security of the Integra Agent schema(s), you may choose to change the Integra Agent schema database password from time to time, according to your organization's database security policies.

Changing the ERP instance schema password will impact the installed Integra components on that instance.

Follow these steps to change the Integra Agent schema or ERP schema passwords:

1 Select **Administrator** > **ERP Instances** from the menu bar. A list of instances appears.

2 Click **Edit** to the right of the instance where the password changed.

3 For each component installed - but not Integra Agent:

   a. To the right of the component's name, click **Edit**.

   b. Click **Remove** to remove the component from the business application instance. The page refreshes, and the **Latest Status** column is updated.

   c. If **Latest Status** indicates that removal of the component failed:

      i. Select **Jobs > View Current Jobs** from the menu bar. Find the instance removal job that was created when you clicked the **Remove** button, and view the job's logs/details to determine the cause of the failure.

      ii. Resolve the cause of the failure.

      iii. Repeat the steps above.

4 Repeat Step 3 for the Integra Agent.

5 Reconfigure the Integra Agent with the new passwords.

6 Reconfigure each component.

Patches

Integra patches let your organization take advantage of added functionality, increased performance, fixes, and/or changes related to ERP system patches and upgrades.

There are four patchable elements in the Integra environment. Each time you apply a patch, the patch can affect some or all of them:

- Data structure and database packages
- Servers
- Metadata
- Web application

The Integra Home schema contains a table that lists the installed patches. You can see this information by selecting **Help > About products** from the menu bar.

From time to time, **patch sets** (sets of recommended patches) are released. Each patch set is accompanied by a **Readme** file, which contains a description of the patches in the set, prerequisites for successful installation, and installation instructions.

Patches are created routinely so that Integra Apps remains compatible with ERP system patches and upgrades. Whenever you patch or upgrade your ERP system, verify that you have installed the latest Integra patch set(s).

Create templates

Templates let you customize the reports generated by Snapshot definitions. You can streamline the information presented in Snapshot reports based on your preferences, and display information specific to your needs. Each template governs the display of one Integra Apps object (see Terminology (p. v) for a definition of "object").

To summarize, you will:

Create a template.

Run a Snapshot definition that includes the template's object.

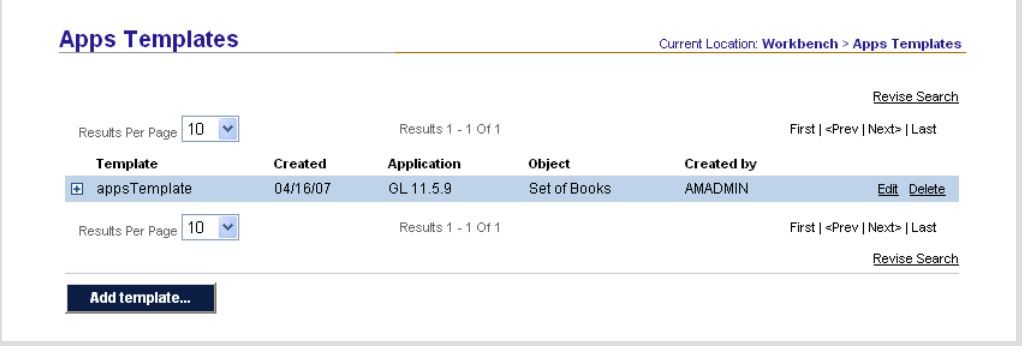
View the report(s), applying the new template.

Here are detailed instructions:

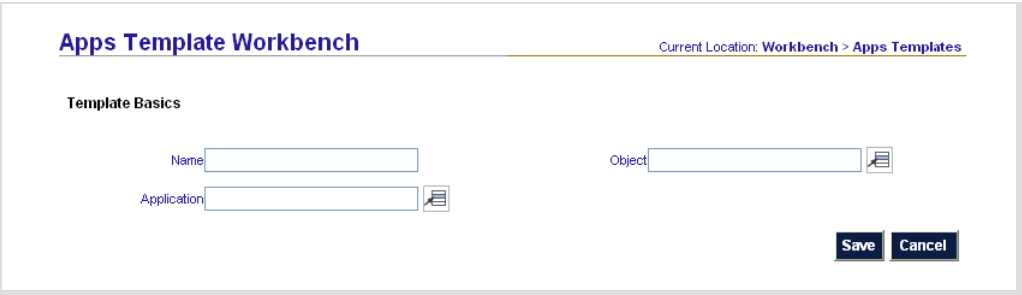
Create a Template

1 Select **Workbench > Apps Template** from the menu bar.

2 If templates already exist, a list of them appears.



3 Click **Add template...**. A blank definition appears.

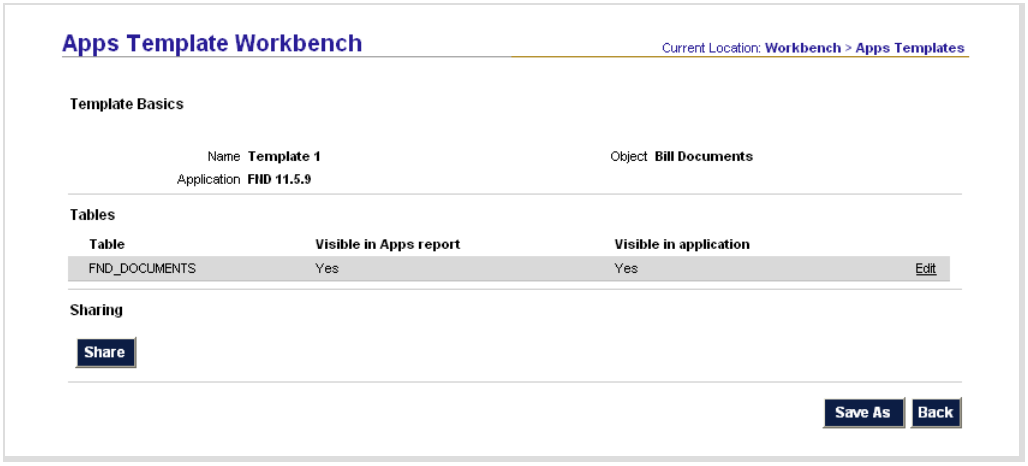


4 Enter the following information:

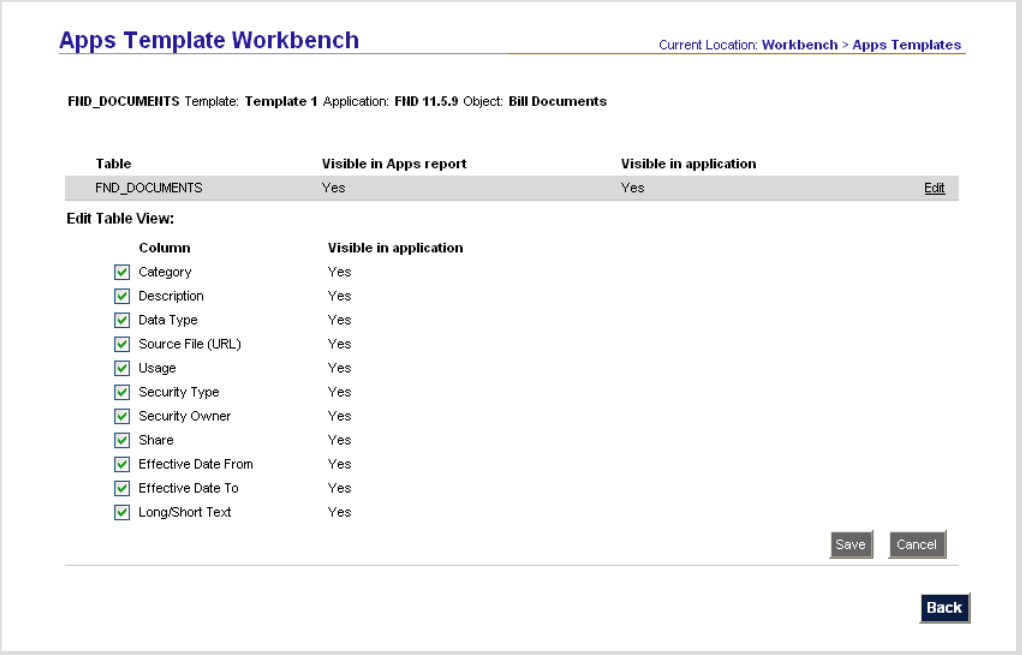
Definition Basics

Name	A short name that will identify this definition.
Application	ERP application whose setup data will be recorded in the Snapshot.
Object	Group of ERP application setup tables that helps you select information to monitor without having to consider the complexity of the data model or entity relationships. An object is typically associated with an ERP application setup form.

5 Click **Save**. The page is refreshed based on **Object**, listing the **Tables** belonging to the object; here is an example.



6 Click **Edit...** to the right of any **Table** to change how the template handles the table's data. A page similar to this appears:



7 Set the following fields if they appear:

Edit Table:

Column	Checking/unchecking this box toggles all the following checkboxes.
Field1, Field2, ... FieldN	When checked, the column's data is included in the Snapshot report.

8 Click **Save**. The page refreshes.

9 Repeat steps 6-8 for other tables, as desired.

10 Click **Back**. The definition page reappears.

11 Click **Save**. You have the option to share the template with other Integra users.

Run a Snapshot definition that uses the template's object
View the resulting Snapshot report(s)

12 See "How to... Create Snapshots of ERP application setups" in the *Integra Apps 5.1 User Guide* for instructions about creating and scheduling Snapshot definitions, and viewing the Snapshot's reports.

Purge occurrences and definitions

The following standalone programs are available:

Purge: Change Tracking Data	Purges unwanted Change Tracking data from the Integra Home schema. The data to be purged is selected by ERP instance and end date.
Purge: Change Tracking Definitions	Purges unwanted Change Tracking occurrences and definitions from the Integra Home schema. You can purge only occurrences, or both occurrences and their definitions. The data to be purged is selected by ERP instance and Integra Apps user.
Purge: Migration Definitions	Purges unwanted Migration definitions from the Integra Home schema. You can purge only occurrences, or both occurrences and their definitions. The data to be purged is selected by ERP instance.
Purge: Snapshots and Comparisons (Administrator)	Purges unwanted Snapshot definitions and occurrences, and Comparisons from your Integra Home schema. In the case of Snapshots, you can purge only occurrences, or both occurrences and their definitions. When you remove a Snapshot definition associated with a Comparison, you will also remove the Comparison. The data to be purged is selected by ERP instance and Integra Apps user.
Purge: Snapshots and Comparisons	This program functions similarly to **Purge: Snapshots and Comparisons (Administrator)**, but purges only data created by the user who schedules this program. The data to be purged is selected by ERP instance.

To schedule either program:

1. Select **Jobs > Schedule a Job** from the menu bar.
2. Click **Add item...**.

3 Set **Job Type** to **Standalone**.

4 Set **Program Name** to the desired program's name.

5 Click **Save**.

6 Click **Back**.

7 Set any other options desired, and click **Schedule**.

Reference

In This Section

Login

All Integra users must begin by logging in on the **Secure Login** page.

The page's URL is:

`http://`*`hostname`*`/`*`web-application-name`*

...where:

`hostname` is the domain name of the web server configured to serve Integra

`web-application-name` is the name of the Integra web application (usually `integra`).

You might also have an Integra icon on your desktop; double-clicking it opens this page.

The upper portion of the page has these fields:

Username	Your Integra Administrator creates your username.
Password	Your Integra Administrator creates your password. To change it, log in, then select **Home > User Profile** from the menu bar.

The upper portion of the page has this button:

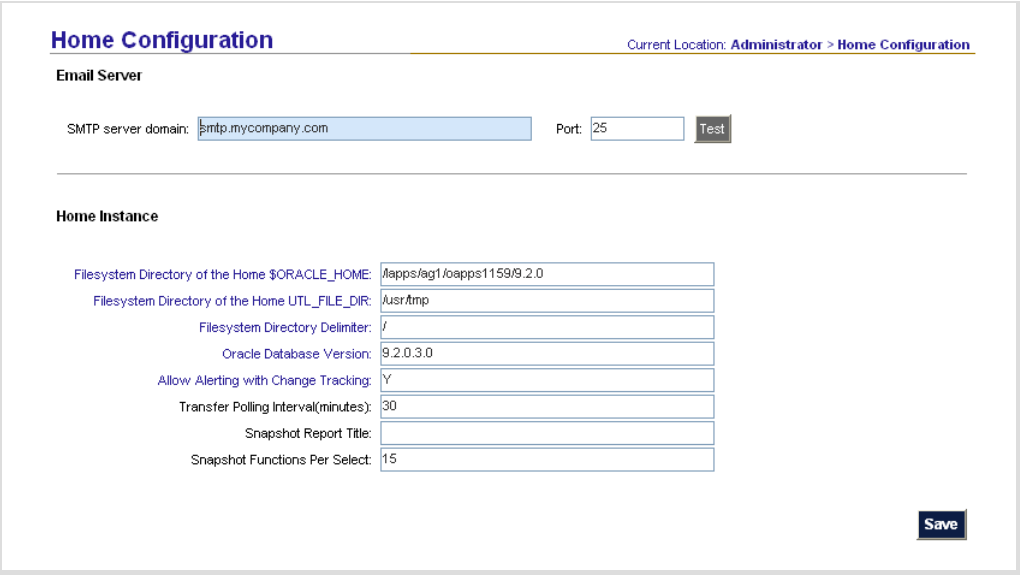
Log in	Verifies your username and password, and displays your Welcome Page. To change your Welcome Page, log in, then select **Home > User Profile** from the menu bar.

The lower portion of the page displays helpful information about Integra. This content is updated regularly if your server's firewall permits HTTP transactions across the Internet.

Security Notice: The **Secure Login** page is the only Integra page that attempts to send or receive information using the Internet. Once you log in, all communications are restricted to your organization's internal networks.

Home Configuration

The **Home Configuration** page contains settings that apply to all of Integra.



To view the page, select **Administration > Home Configuration** from the menu bar.

The page contains these fields:

Email Server

SMTP server domain	The domain name of your mail server. If left blank, Integra cannot send email notifications.
Port	The port number of your mail server.

Home Instance

Filesystem Directory of the AMHOME $ORACLE_HOME	File system path to the directory that holds the ERP application's database.

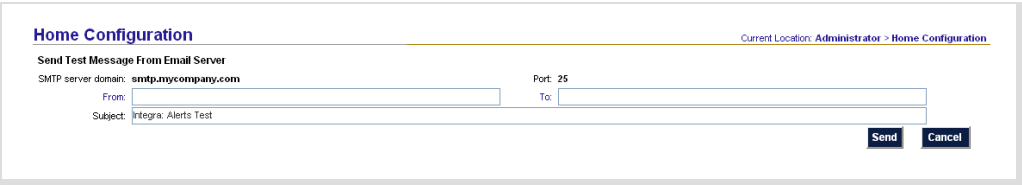
Filesystem Directory of the AMHOME UTL_FILE_DIR	***Oracle 8i users:*** Use the value of the **util_file_dir** parameter in the **INIT.ORA** file on the ERP instance. ***Oracle 9i users:*** Use any valid file system path. We recommend you create a new directory for this purpose. Ensure that the value is accurate and that the permissions on the specified directory are set properly. An invalid entry or insufficient privileges will result in errors.
Filesystem Directory Delimiter	Backslash (\) for Windows systems and forward slash (/) for Unix/Linux.
Oracle Database Version	The Integra Home instance's database version.
Allow Alerting with Change Tracking	***Integra Apps 5.x users:*** When set to `Y`, Integra Apps' Change Tracker lets you configure Alerts.

The page contains these buttons:

Test...	Displays the **Send Test Message from Email Server** page (see Email alerts (p. 59)).
Save	Saves the changes you made on this page.

Email alerts

Integra's optional Email Alert feature is used by all Integra components. When you click **Test...** on the **Home Configuration** page, the **Send Test Message from Email Server** page appears.



Enter **From** and **To** addresses, and optionally a **Subject**, and click **Send**. The page is refreshed, displaying a **Status** field.

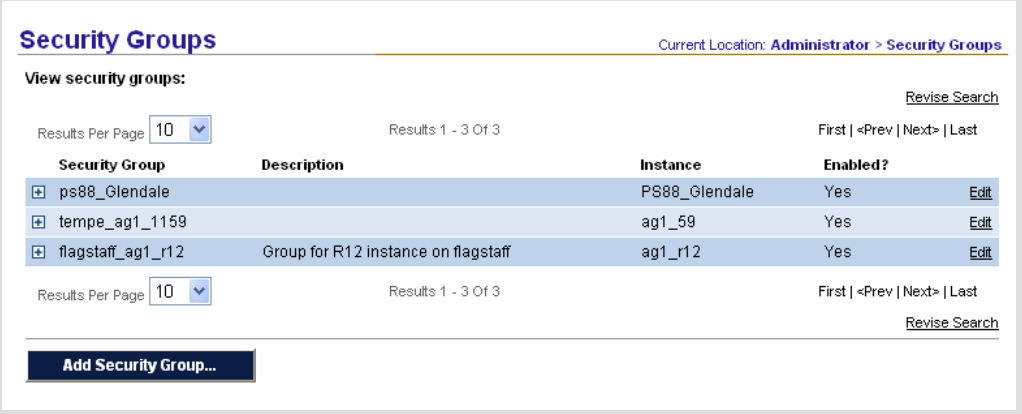
To update **Status**, click **Refresh**. If the test message is sent successfully, but you don't receive it, first verify that the message isn't being treated erroneously as spam by your incoming mail server or mail client.

When you're finished, click **Back**. The **Home Configuration** page reappears.

Integra does not use the Oracle Alerts product. It sends email messages directly from the database, using Oracle's UTL_SMTP feature.

Security Groups

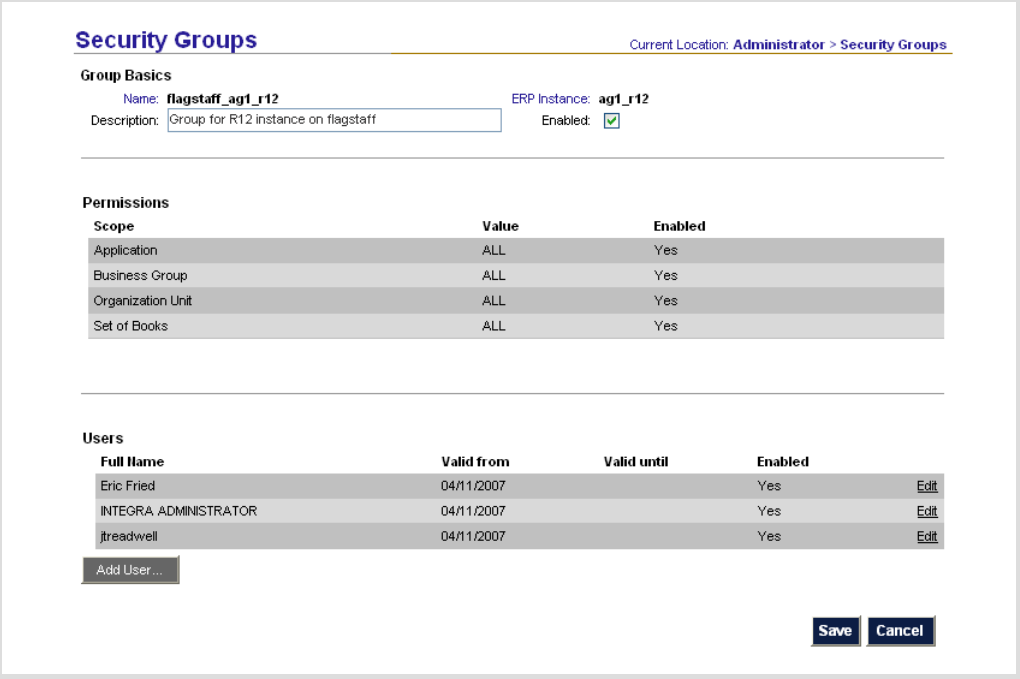
Security Groups control users' access to ERP data. To view existing Security Groups or add new ones, select **Administrator > Security Groups** from the menu bar. You will see a list of existing Security Groups.



Like all Integra search results, you can see more (or different) results by changing **Results Per Page**, clicking **Next >**, or clicking **Revise Search**.

Click **Edit** to view an existing group, or **Add security group...** to add a new one. In either case, the **Security Groups** page appears.

Security Groups definition



The page contains these fields:

Group Basics

Name	The group's name.
Description	A short description of the group.
ERP Instance	The ERP instance that the group can work with.
Enabled	When checked, the group is active.

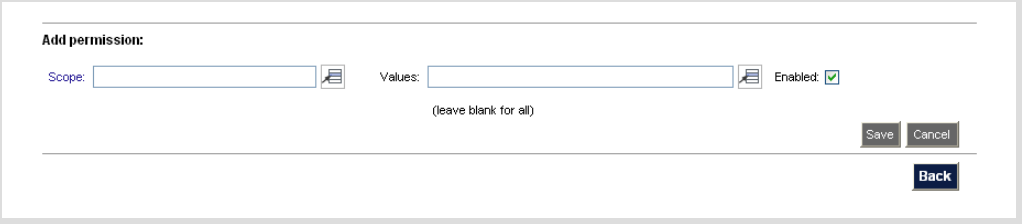
The page contains these buttons and links:

Edit	*Appears to the right of each permission or user that has been added.* Displays the appropriate page - **Group Permissions** or **Group Users**, and lets you revise the permission or user.
Add permission...	*Visible until the first user is added.* Displays the **Group Permissions** page and lets you add a permission to the group.
Add user...	*Disabled until you have added at least one permission with each Scope.* Displays the **Group Users** page and lets you add a user to the group.

Save	Saves the changes made to the fields on this page. Does not affect changes to permissions and users, which are saved or discarded while on the **Group Permissions** and **Group Users** pages, respectively.
Cancel	Discards the changes made to the fields on this page. Does not affect changes to permissions and users, which are saved or discarded while on the **Group Permissions** and **Group Users** pages, respectively.

Permissions

Permissions control the group's access to data.



Before you can add users to a group, you must add a permission for each **Scope**.

Add all desired permissions before adding the first user. Once you add a user, you cannot change the group's permissions.

The page contains these fields:

Add permission:

Scope & Values	Access is granted to data where *Scope IS IN Values*. To match all values, leave **Values** blank.
Enabled	When checked, the permission is active.

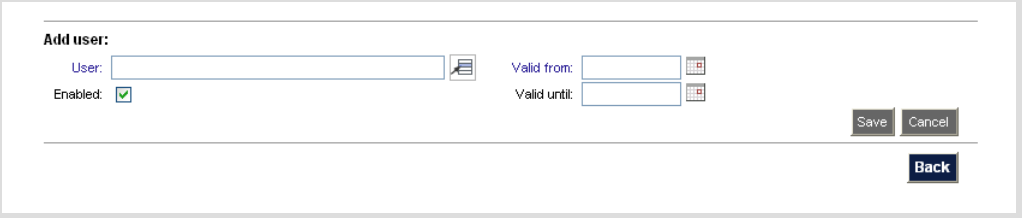
The page contains these buttons and links:

Edit	Refreshes the page and lets you revise a permission.
Save	Saves your changes to the permission and refreshes the page.

Cancel	Discards your changes to the permission and refreshes the page.
Add permission...	Refreshes the page and lets you add a permission.
Back	Discards any unsaved changes and displays the **Security Groups** page.

Users

Users must belong to security groups in order to have access to Integra's components and their functions.



The page contains these fields:

Add user:

User	Select a user from the pop-up window.
Enabled	When checked, this user is an active member of the group.
Valid from	First day the user will be a member of the group.
Valid until	*Optional:* Last day the user be a member of the group.

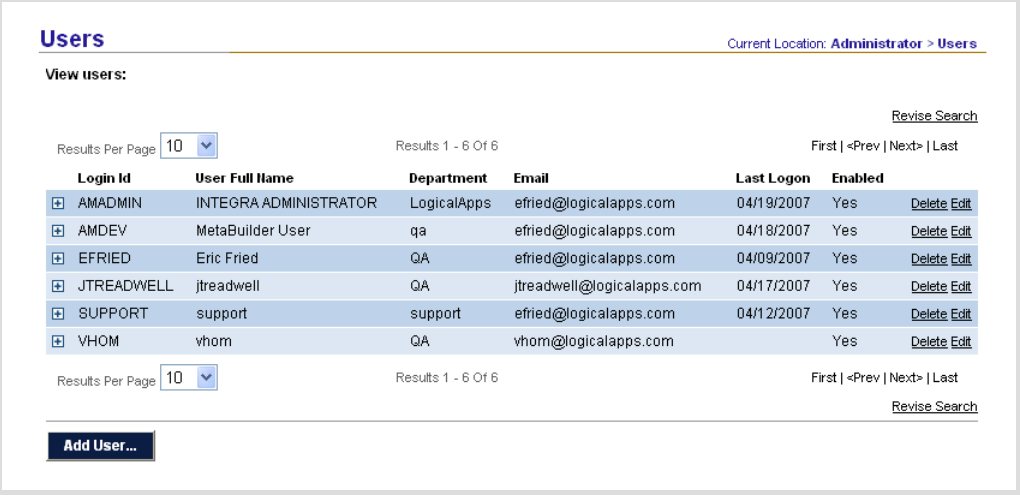
The page contains these buttons and links:

Edit	Refreshes the page and lets you revise a user's information.
Save	Saves your changes to the user's information and refreshes the page.
Cancel	Discards your changes to the user's information and refreshes the page.
Add user...	Refreshes the page and lets you add a user.

Back	Discards any unsaved changes and displays the **Security Groups** page.

Users and Roles

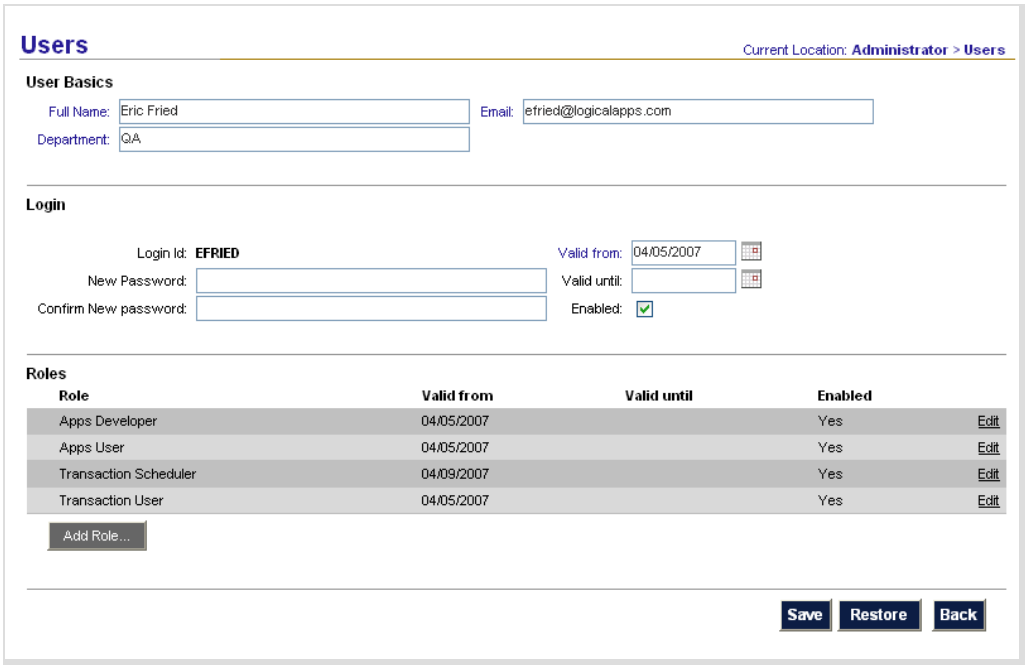
Use the **Users** page to configure a user's login access and roles. You can also modify a user's access or roles, or disable a user's access. To view existing users or add new ones, select **Administrator > Users** from the menu bar.



Like all Integra search results, you can see more (or different) results by changing **Results Per Page**, clicking **Next >**, or clicking **Revise Search**.

Click **Edit** to view an existing user, or **Add User...** to add a new one. In either case, the **Users** page appears.

Users definition



The page contains these fields:

User Basics

Full name	Integra user's full name. The name may consist of letters, spaces, ' (apostrophes), and – (dashes); other characters are not allowed.
Department	User's department or organization. The department may consist of letters, digits, spaces, and - (dashes); other characters are not allowed.
Email	The user will receive alerts at this address.

Login

Login ID	User's Integra Login ID. The ID may consist of letters and digits only; spaces and other characters are not allowed.

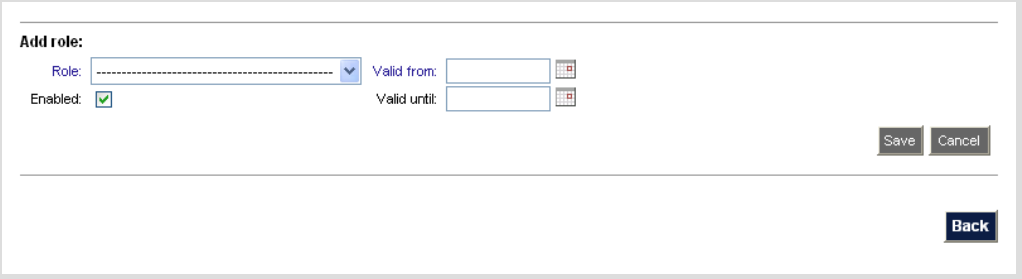
Password	Initial password for the new user (40 character length limit). This data will be encrypted before it is stored. Not required if LDAP Authentication mode is used. See System Profile (p. 81) for more information. The following characters are not allowed: `! | / = $ & @ " ' ` ,`
Confirm password	Re-enter the password to confirm
Valid from	First day user can log in
Valid until	*Optional:* Last day user can log in
Enabled	Enables the user to log in; if not checked, the user cannot log in.

The page contains these buttons and links:

Edit	*Appears when one or more roles have been assigned.* Displays the **Roles** page (see Roles (p. 69)), where you can revise the role.
Add role...	Displays the **Roles** page and lets you add a role.
Save	Saves the changes made on this page. Does not affect changes to roles, which are saved or discarded while on the **Roles** page.
Restore	Discards the changes made on this page. Does not affect changes to roles, which are saved or discarded while on the **Roles** page.
Back	Displays the list of Users.

Roles

Each user can have one or more roles. To add a role, click **Add role...** on the **Users** page. To revise a role, click **Edit** to the right of the role's name on the **Users** page.



The page contains these fields:

Add role:

Role	Select a role from the drop-down list.
Enabled	Enables this role for this user.
Valid from	First day the user will have this role.
Valid until	*Optional:* Last day the user will have this role.

The following roles are available for Integra Apps 5.1.

For the convenience of administrators of previous versions, we also show 4.x roles here; your 4.x users' roles have been replaced with the corresponding 5.1 roles.

Apps 5.1 Role		***Apps 4.x Role***
Apps User	Create and edit Snapshot, Change Tracking, and Migration definitions.	*Apps User*
Snapshot Scheduler	Schedule Snapshot definitions for execution, and compare Snapshot occurrences.	*Apps User*
Change Tracking Scheduler	Schedule Change Tracking definitions for execution.	*Change Tracking Manager*
Migration Scheduler	Schedule Migration definitions for execution.	*Migration User*

Apps 5.1 Role		*Apps 4.x Role*
Apps Developer	Create and edit Templates.	*Template Builder*
Apps Metadata Manager	Use MetaBuilder.	*Metabuilder User*
	This 4.x role is not converted (all administration is conducted by the Administrator role):	*Apps Administrator*

The **Administrator** role gives the user access to all **Administrator** menu bar items.

The page contains these buttons and links:

Edit	Refreshes the page and lets you revise a role.
Save	Saves your changes to the role and refreshes the page.
Cancel	Discards your changes to the role and refreshes the page.
Add role...	Refreshes the page and lets you add a role.
Back	Discards any unsaved changes and displays the **Users** page.

ERP instances

You must configure Integra to work with each ERP instance that you want to manage. Broadly speaking, you will perform these tasks for each instance:

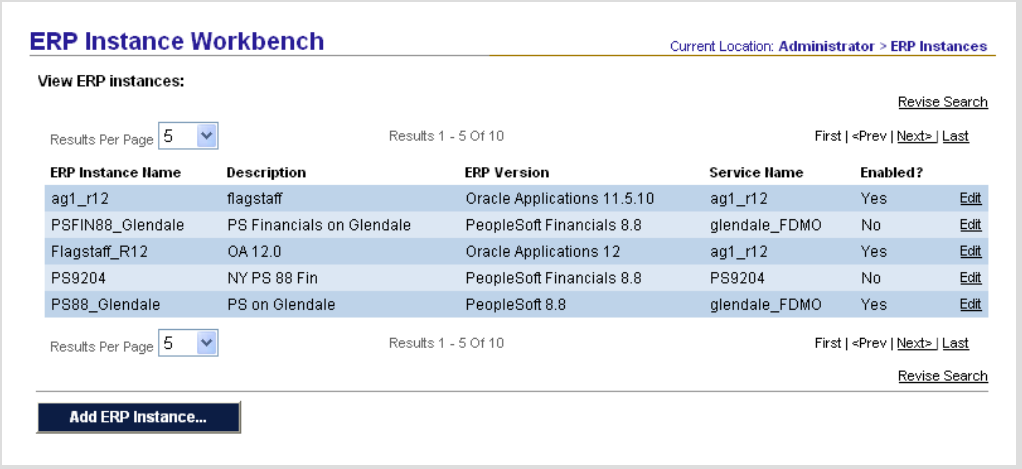
Define a new instance in Integra.

Create Integra Agent tablespaces for the instance. The Integra Agent lets Integra components perform tasks that require access to the instance's data.

Configure Integra Agent parameters.

Configure the component's parameters.

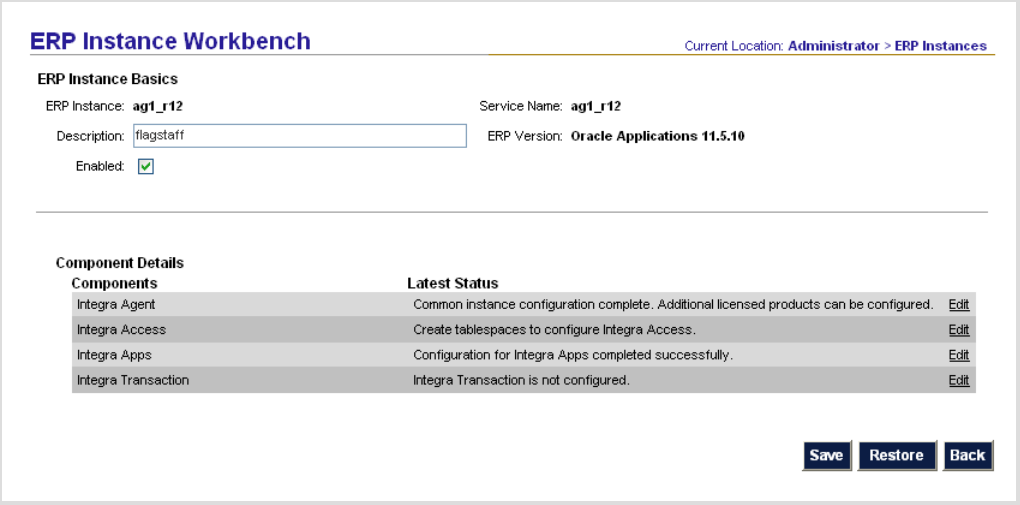
To view ERP instances or add new ones, select **Administrator > ERP Instances** from the menu bar. A list of ERP instances appears.



Like all Integra search results, you can see more (or different) results by changing **Results Per Page**, clicking **Next >**, or clicking **Revise Search**.

Click **Edit** to view an existing instance, or **Add instance...** to add a new one. In either case, the **ERP Instances** page appears.

ERP Instances definition



The page contains these fields:

Basics

ERP Instance	Short name that identifies the ERP instance
Description	Longer, more descriptive name for the ERP instance. For example, if **Instance name** is `Dev`, **Description** could be `Development Instance`
Service Name	TNSNAMES service name that will be used when Integra creates a database link from the Integra Home schema to the new Integra Agent schema on the ERP instance. This name must be present in the TNSNAMES configuration for the Integra Home instance's Oracle Home.
ERP Version	ERP application version

The page contains these buttons and links:

Edit	Displays the **Components** page (see Component Details (p. 73)), where you can configure, remove or reconfigure the component.
Save	Saves the changes made on this page. Does not affect changes to components, which are performed while on the **Components** page.
Restore	Discards the changes made on this page. Does not affect changes to components, which are performed while on the **Components** page.

Back	Displays the list of ERP instances.

Component Details

Integra has a number of components (Access, Apps, Codebase, Forms and Transaction). Three of them - Access, Apps and Transaction - can be installed in the same Integra Home instance; the **ERP Instances** page displays the components you have installed, along with an additional component used by all other components: the Integra Agent.

The **Component Details** area lets you configure or remove any component. The area contains these buttons and links:

Remove	*Appears when the component has already been configured.* Removes the component from the instance and refreshes the page.
Configure	Configures the component on the instance and refreshes the page. If the component has already been configured, this button updates the configuration. If you have made changes to values on this page, click **Save** before clicking **Configure**.
Save	Saves your changes to the fields on this page without configuring the component, and refreshes the page.
Cancel	Discards your changes and refreshes the page.

These buttons appear at the bottom of the ERP Instances page when working with Component Details:

Refresh	Retrieves the latest status of the instance.
Back	Discards unsaved changes and displays the **ERP Instances** page.

The fields on this page vary by component. They are listed below.

Integra Agent

The fields for Integra Agent are:

Tablespaces

Data	The data tablespace used by the Integra Agent. Normally, the tablespace is named **AM_AGENT_D**.
Index	The index tablespace used by the Integra Agent. Normally, the tablespace is named **AM_AGENT_X**.

Attributes

Integra Agent Username	For the Integra Agent schema. Created on the ERP instance if it does not exist. Default value: `AM_AGENT`
Database Link from HOME to Agent	Needed to transfer metadata from the Integra Home instance to the ERP instance, and to execute any process on the ERP instance. Created on the Integra Home instance if it does not exist. The value must contain no more than 20 characters.
ERP Username	Existing ERP instance user with full access to the ERP instance's data. ***Oracle E-Business Suite:*** The typical value is `APPS` ***PeopleSoft Enterprise:*** The typical value is `SYSADM` ***Siebel CRM Applications:*** The typical value is `SIEBEL`
ERP User Database Link	Needed to access ERP application information. Created on the ERP instance if it does not exist. The value must contain no more than 20 characters.
Temporary Tablespace Name	Name of the ERP instance's temporary Tablespace.

Database Version	ERP instance's database version. Example: `8.1.7`
Remote File System Directory of the $ORACLE_HOME	File system directory that corresponds to the $ORACLE_HOME of the ERP instance.
Remote File System Directory of the UTL_FILE_DIR	File system directory specified by the UTL_FILE_DIR parameter in the ERP instance's INIT.ORA file. Tells Integra where to read and write files that reside on the ERP instance.
File System Directory IMP/EXP/SQLPLUS Binaries	File system directory path where these executables reside on the ERP instance.
Remote Instance File System Directory Delimiter	The character used by your ERP instance's file system to separate elements within pathnames. ***Unix/Linux:*** `/` (forward slash) ***Windows:*** `\` (backslash)
Export/Import Buffer Value	Sets the BUFFER parameter, which is used when invoking the IMP and EXP utilities. Default value: `100000`
ISO language	Default value: `US`
Parallel Workers	Default value: `4`
Database Host	Server where database resides.
Database Port	Port for installing and launching the database.
Database SID	Name of the database that contains the ERP application schema.
Analyze Date Tolerance	Default value: `7` ***Informia Subset users:*** Subset will **not** perform a row count if there is an existing one that was created within this many days.

APPSORA.env	Default value: `APPSORA.env`

Integra Apps

The fields for Integra Apps are:

Tablespaces

Data	Data tablespace used by the Integra Agent. Typically named **APS_AGENT_D**.
Index	Index tablespace used by the Integra Agent. Typically named **APS_AGENT_X**.

Attributes

APPLSYS Username	***Oracle E-Business Suite users:*** ERP instance's **APPLSYS** username. ***PeopleSoft Enterprise users:*** ERP instance's **SYSADM** username.
HR Username	***Oracle E-Business Suite users:*** ERP instance's **HR** username.
Remote Instance Directory of the FND_TOP	***Oracle E-Business Suite users:*** ERP instance's **$FND_TOP** directory.
Baseline Definition Owner	Integra Apps User who will own the Snapshot and Change Tracking definitions that will be generated automatically when you complete this configuration process.
Remote Instance Directory for Unix Executables	***Oracle E-Business Suite users:*** Directory where Unix command executables are located (usually `/bin`).
Remote Instance Directory of the APPL_TOP	***Oracle E-Business Suite users:*** ERP instance's **$APPL_TOP** directory.
Remote Directory for Oracle Supplied LCT Files	***Oracle E-Business Suite users:*** Directory that stores the ERP instance's Oracle-supplied **.lct** files.

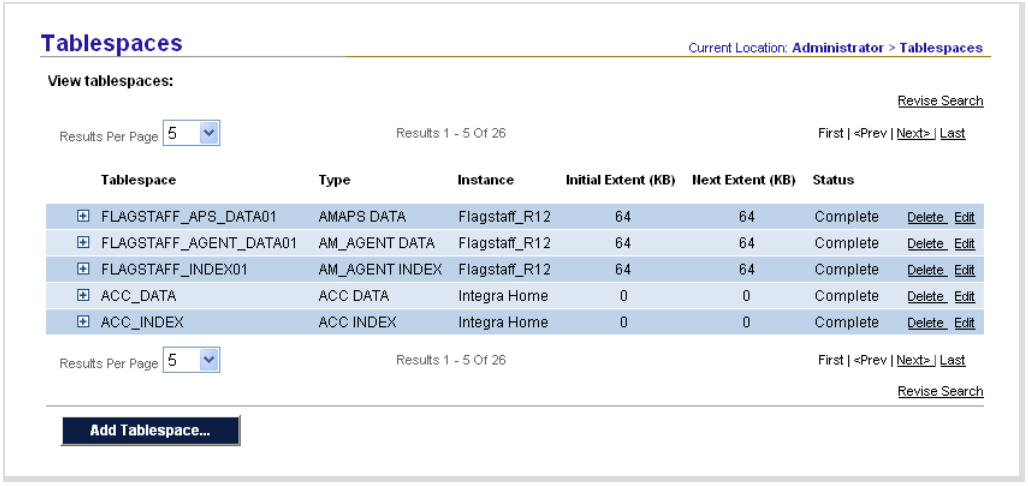
Freeze Table Overlays	If you have moved or renamed any ERP tables or views from their defaults, enter `N` to map them (which will make them visible to Integra Apps). The default value is `Y`.
Descriptive Flexfields Flag	***Oracle E-Business Suite users:*** Enter `Y` to capture all specific values entered in every Descriptive Flexfield of every setup Object in this ERP instance. The default value is `N`. **Note:** To report on the structure or definition of a Descriptive Flexfield, use the **System Administration** application's **Descriptive Flexfield Segments** object.
Remote APPLTOP (Y/N)	***Oracle E-Business Suite users:*** Enter `Y` if your **APPL_TOP** is on a different physical server than the ERP instance. The default value is `N`.
Remote APPLTOP Server Name	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Name of server.
Remote APPLTOP Migration Directory	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Migration directory.
Remote APPLTOP OS User	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Operating system user.
Remote Instance Directory for SCP/SSH	***Oracle E-Business Suite users: If your APPL_TOP is on a different computer than the ERP instance:*** Directory where the **scp** and **ssh** commands are located; sometimes differs from **Remote Instance Directory for Unix Executables** above.

Tablespaces

Integra requires the creation and maintenance of several tablespaces. Use the **Tablespace Workbench** to work with tablespaces.

> All tablespaces created by Integra are locally managed.

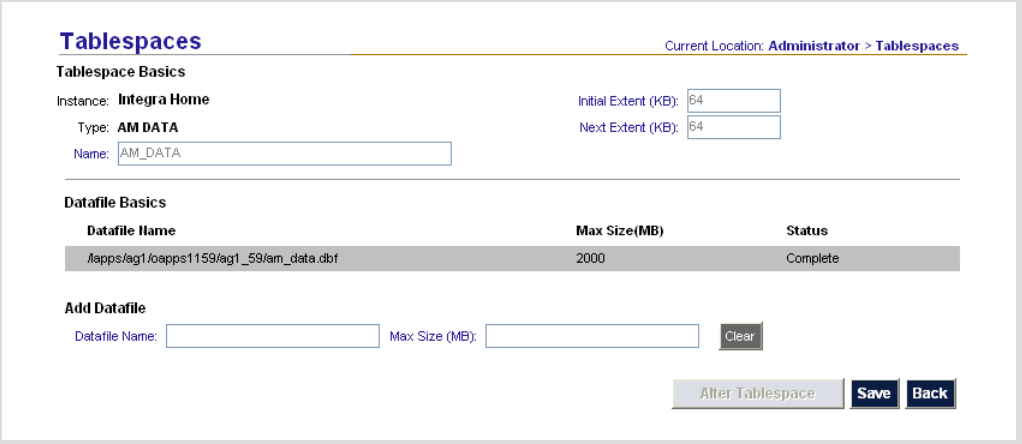
To use the **Tablespace Workbench**, select **Administrator > Tablespaces** from the menu bar. A list of existing tablespaces appears.



> Like all Integra search results, you can see more (or different) results by changing **Results Per Page**, clicking **Next >**, or clicking **Revise Search**.

Click **Edit** to view an existing tablespace, **Delete** to delete a one, or **Add tablespace...** to add a new one. If you click **Edit** or **Add tablespace...**, the **Tablespaces** page appears.

Tablespaces definition



The **Tablespace Basics** section contains these fields:

Tablespace Basics

Instance	Where the tablespace will reside.
Type	Choose the appropriate type.
Name	The tablespace's name.
Initial extent	Size of the tablespace's initial extent.
Next extent	Size of the tablespace's subsequent extents.

The **Tablespace Basics** section contains these buttons:

Create tablespace **Alter Tablespace**	Creates all the datafiles listed, and refreshes the page. If some of the datafiles already exist, this button updates their configurations.
Save	Saves the changes made on this page. Does not affect changes to datafile configuration settings.
Back	Discards the changes made on this page. Does not affect changes to datafile configuration settings.

Datafile Basics

Each tablespace uses one or more datafiles, and has exclusive use of its datafiles.

The **Datafile Basics** section contains these fields:

Datafile Basics

Datafile Name	The datafile's name, including the file system directory path for where the datafile will be stored.
Max Size	The maximum size of the datafile. Its initial size will be 10% of the amount specified here; it will be allowed to AUTOEXTEND in increments of 10% or 200MB, whichever is less, until reaching the size entered here.

The section contains this button:

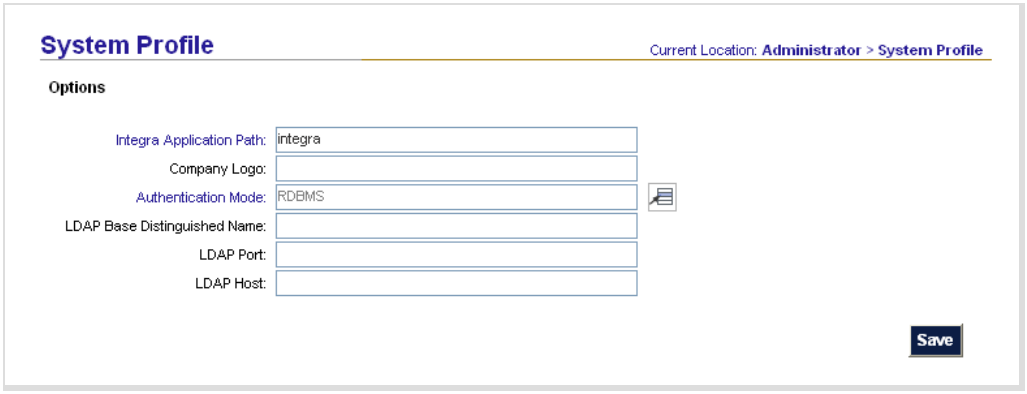
Clear	Clears the **Datafile Name** and **Max Size** fields.

The section contains these links:

Edit	*Appears to the right of undeployed datafile definitions.* Displays the **Datafiles** page, and lets you revise the datafile's configuration.
Delete	*Appears to the right of undeployed datafile definitions.* Deletes the datafile.

System Profile

To change the System Profile, select **Administrator > System Profile** from the menu bar. The System Profile page appears.



The page contains these fields:

Options

Integra Application Path	Final portion of the URL that summons the **Secure Login** page. The entire URL is: `http://hostname/web-application-name` ...where: `hostname` is the domain name of the Integra UI server The default value of `web-application-name` is **integra**
Company Logo	Name of the image file that is displayed in the upper left corner of each page. The file must be located in the **images** subdirectory of the directory on the Integra UI server where the Integra web application is deployed.
Authentication Mode	There are two options for validating users: **RDBMS** uses an Integra RDBMS module. **LDAP** uses an LDAP directory maintained by your organization. This eliminates the need for Integra users to enter their password on the **Secure Login** page. If you choose this option, the following fields are required:

LDAP Base Distinguished Name	LDAP entry that identifies an authorized user of the LDAP server.
LDAP Host	Machine name or IP address where the LDAP server is hosted.
LDAP Port	Port number to use when communicating with the host.

The page contains these buttons:

Save	Saves changes made on this page.
Restore	Discards changes made on this page.

Help

Selecting **Help > About products** from the menu bar displays the following information about each installed component:

Product (component name)

Release Number

Patch Number (latest patch release applied, and patch history)

Metadata release

Metadata Patch Number (latest metadata patch release applied), and Patch History

Customizing Integra Apps with MetaBuilder

Metadata is detailed information about the data structure of the ERP application's database.

Integra Apps' predefined metadata covers many ERP application versions and modules. MetaBuilder lets you customize this metadata to better serve your business needs.

Because MetaBuilder lets you create metadata for any application, you can also use it to build metadata for applications that are not predefined in Integra.

In This Section

How to create metadata

The following pages describe how to create metadata in step-by-step detail. You must perform all steps, in the order shown.

Here is a summary of the steps you will perform:

Set up the structure needed to store custom metadata

Step 1: Define a Product Family

Step 2: Define Suites

Step 3: Define Applications

Step 4: Assign Applications to Suites

Step 5: License Suites

Step 6: Assign Applications to ERP Instance

Step 7: Assign Users to Security Groups

Step 8: Import Data Dictionary into MetaBuilder

Create the metadata needed for snapshots, comparisons and change tracking

Step 9: Create Schemas

Step 10: Assign Schemas to an Application

Step 11: Add Tables to a Schema

Step 12: Configure Columns within a Table/View
Insert Phantom Columns
Insert Missing Columns

Step 13: Assign Translation Details to Columns
Option 1: Column
Option 2: Function (Existing)
Option 3: Function (New): Create a Translation Function; Test the function; Apply the function; Add multiple SELECTS to the function

Step 14: Create Objects
Cross-Reference an Object to another Application
Maintain Column Filters: Security Filters; Suggested Filters
Copy a Standard Integra Object

Step 15: Synchronize

Test the metadata by verifying that Integra Apps returns the correct information when using the new or updated metadata

Step 16: Test Metadata

Preparation

Before creating metadata, you must identify non-standard application Objects and their data sources.

Identify non-standard application Objects

An **Object** is a group of application setup tables that helps users select information to monitor. As a MetaBuilder user, you must identify all Objects that you want to work with. Your choice of each Object's contents facilitates the use of Integra Apps by those who are familiar with the ERP application. To provide an intuitive look and feel, Objects should be closely aligned with the organization of information seen in the ERP application's menus.

For example, Oracle E-Business Suite's **General Ledger Define Set of Books** form contains fields that are stored in different tables. This is not apparent, nor typically relevant, to managers and end-users. Integra Apps contains a **Set of Books** Object that aggregates the information in these tables, facilitating reporting and change tracking.

Identify Object data sources

Once you identify an Object, you must identify the tables associated with it.

Oracle E-Business Suite users: Forms are usually based on database views. Each view usually contains one table that serves as the primary source of information (since it stores a majority of the data), but other tables might need to be included in the Object's definition too. You can usually see a setup form's table or view information using the **Help...Record History** menu item; if not, you must do more research. One useful set of resources are the eTRMs (electronic technical reference manual).

PeopleSoft Enterprise users: Pages can be based on either tables or views; use Application Designer to identify them. Typically, objects containing a suffix of **_VW** indicate that the page is based on a view.

Set up metadata structure

Before creating metadata, you must set up the necessary Product Family, Suites, Applications, Application licensing and ERP instance configuration, and import your data dictionary.

> The following instructions assume that your Integra Administrator has given you the **Integra MetaBuilder User** role. If you do not have this role, contact your Integra Administrator.

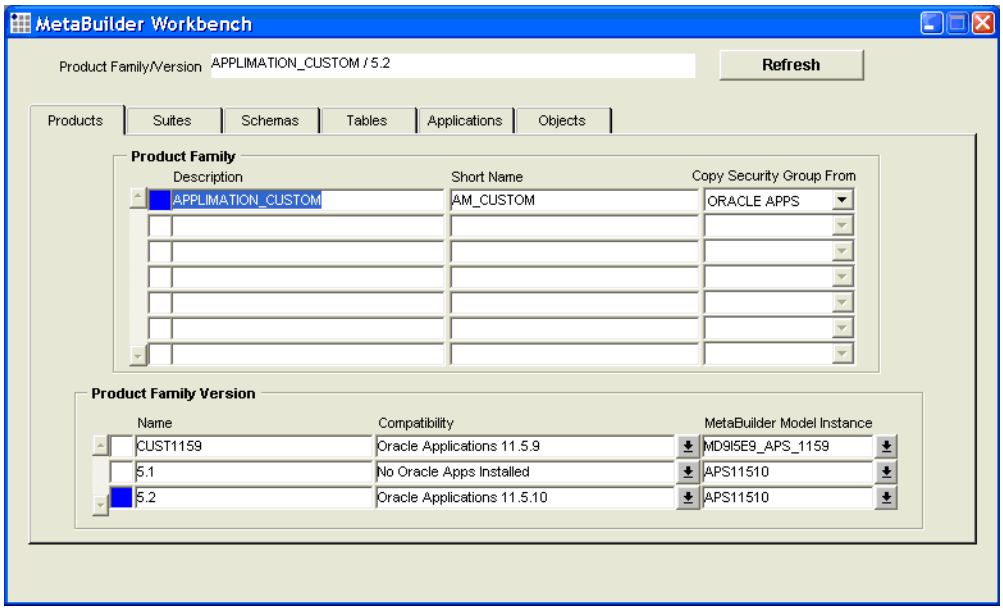
Step 1: Define a Product Family

A **Product Family** is a software product that provides a business solution and supports different business functions; Oracle E-Business Suite is an example of a Product Family.

A Product Family often contains more than one **Product Family Version**. Software makers typically provide more functionality with each new Product Family Version they release.

1 Log into MetaBuilder.

2 From the Navigator, under the **Integra MetaBuilder User** role, select **Navigate...Apps Metadata...MetaBuilder Workbench**. MetaBuilder opens the MetaBuilder Workbench.



3 Select the **Products** tab to define a new Product Family. Use the following information to enter your Product Family details:

Description	Name of the Product Family
Short Name	Abbreviated version of the Product Family name. You cannot change this value once you have defined the Product Family; if you need to change the name, change the **Description** instead.

4 Enter the following information in the **Product Family Version** zone to define a Product Family Version:

Name	Used within MetaBuilder to describe the Product Family.
Compatibility	ERP application suite version that the metadata will be compatible with.
MetaBuilder Model Instance	Database instance where the data dictionary information that you use for metadata development is kept. Contact your Administrator if you want to use a new database instance.

5 Select **Action...Save** from the menu to save your work.

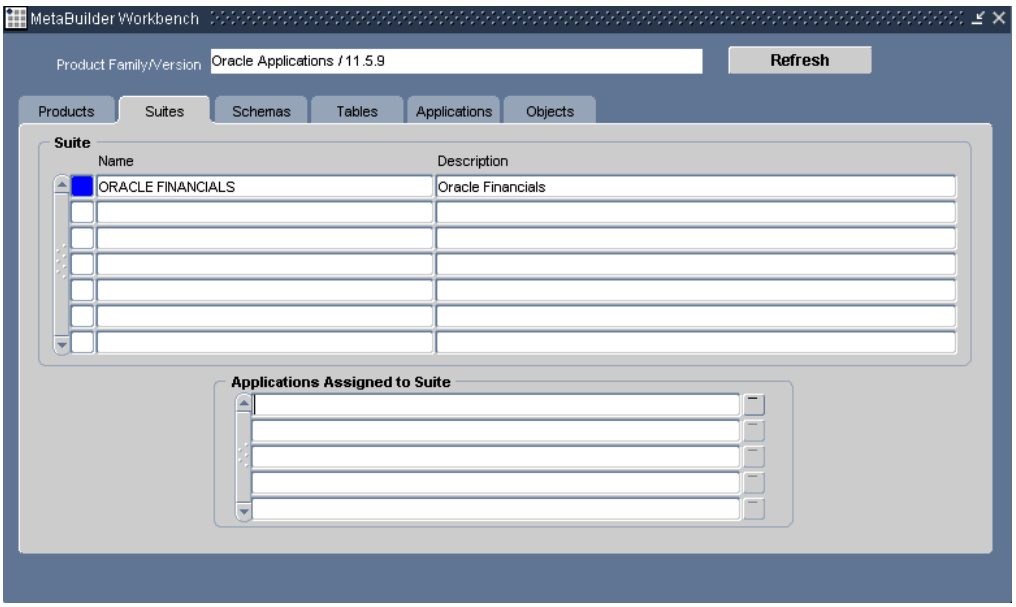
Step 2: Define Suites

In this step, you will create **Suites** to assign to your Product Family/Version. A Suite is comprised of a number of Applications within a Product Family, usually corresponding to a group of interrelated business processes (e.g., Financials or Manufacturing).

Follow these steps to create a Suite:

1 On the MetaBuilder Workbench, select the Product Family and the Version in the **Products** tab for which you want to define Suites.

2 Navigate to the **Suites** tab. The MetaBuilder Workbench displays the Product Family/Version that you selected from the **Products** tab in the **Product Family/Version** field.



3 Enter the following information in the **Suite** zone to define a Suite:

Name	Name of the Suite.
Description	Description of the Suite you are creating. This Description should represent all the Applications that you plan to assign to this Suite.

4 Select **Row...Insert** from the menu and repeat Step 3 for each Suite that you want to create under this Product Family/Version.

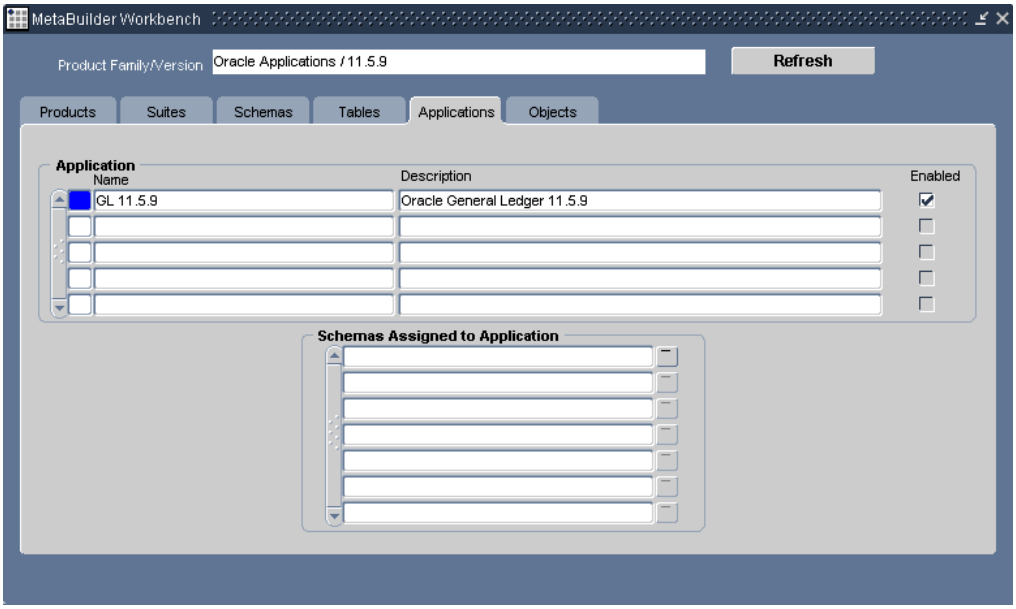
5 Select **Action...Save** from the menu to save your work.

Step 3: Define Applications

In this step, you will create **Applications** to assign to Suites. MetaBuilder Suites are comprised of Applications, which usually satisfy a particular business function (e.g., Payables or Receivables).

Follow these steps to define an Application:

1 On the MetaBuilder Workbench, navigate to the **Applications** tab pertaining to a specific Product Family/Version.



2 Enter the following information in the **Application** zone to define a new Application:

Name	Name of the Application.
Description	Description of the Application.
Enabled	Select this checkbox to enable the Application.

3 Select **Row...Insert** from the menu and repeat Step 2 for each Application that you want to create under this Product Family/Version.

4 Select **Action...Save** from the menu to save your work.

Step 4: Assign Applications to Suites

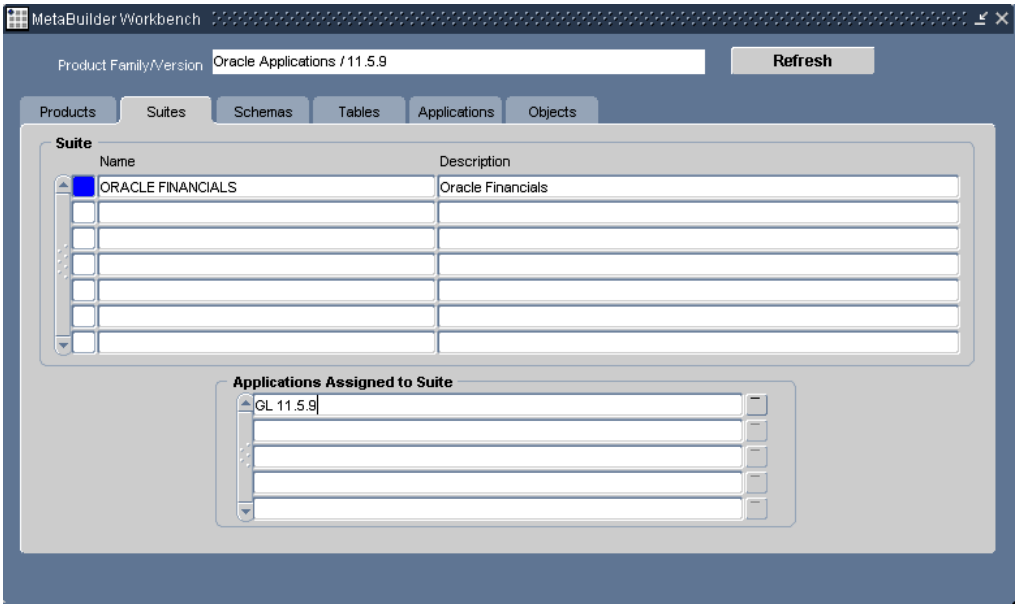
Once you define your Applications, you must assign them to one or more Suites. An Application may be assigned to one or more Suites, and a Suite may contain one or more Applications.

Follow these steps to assign Applications to Suites:

1 On the MetaBuilder Workbench, navigate to the **Suites** tab pertaining to a specific Product Family/Version.

2 In the **Suite** zone, select a Suite to which you want to assign your Applications.

3 In the **Applications Assigned to Suite** zone, select an Application from the list of values to assign to the selected Suite.



> The list of Applications contains only those created in Step 3: Define Applications (p. 91).

4 Select **Row...Insert** from the menu and repeat Step 3 for each Application that you want to assign to this Suite.

5 Select **Action...Save** from the menu to save your work.

Step 5: License Suites

After assigning your Applications to a Suite, you must add the Suite to the list of Licensed Suites and Applications on Integra's **Application Licensing** page; this allows your site's ERP instances to report on the Suite's Applications.

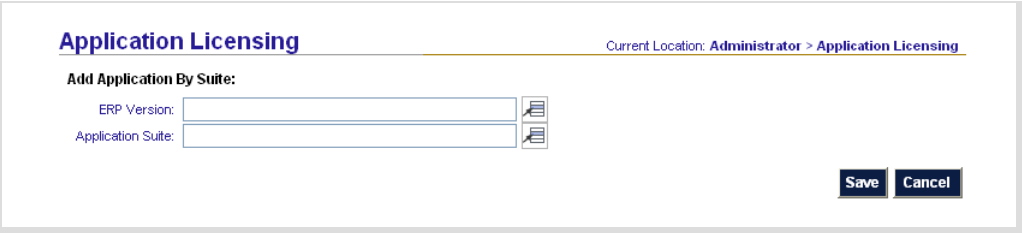
> You must have access to Integra's **Administrator** menu to complete this step. If you do not, contact your Integra Administrator.

Follow these steps to assign your Suites to the Home Configuration:

1 Log into Integra.

2 Select **Administrator > Application Licensing** from the menu bar.

3 Click **Add Application By Suite**. The **Add Application By Suite** page appears.



4 Select the appropriate **ERP Version**.

5 Select the new **Application Suite**.

6 Click **Save**.

Step 6: Assign Applications to ERP Instance

After licensing Suites, you must assign them to each ERP instance where you want to be able to report and track changes.

> You must have access to Integra's **Administrator** menu to complete this step. If you do not, contact your Integra Administrator.

Follow these steps to assign your Applications to an ERP instance:

1 Select **Administrator > ERP Instances** from the menu bar. A list of configured ERP instances appears.

2 Click **Edit** to the far right of the ERP instance you want to assign Applications to. Information about the ERP instance configuration appears.

3 Click **Edit** to the far right of **Integra Apps**. The page is refreshed, and the lower portion now displays fields for configuring Integra Apps.

4 Specify the applications that Integra Apps will be able to monitor on this ERP instance. You can:

Click **Add application...** to add applications one-by-one, clicking **Save** after selecting each application.

Click **Add suite...** to add all applications in a product suite.

5 Click the **Configure** button. For more information, see Configure Integra Apps (p. 15).

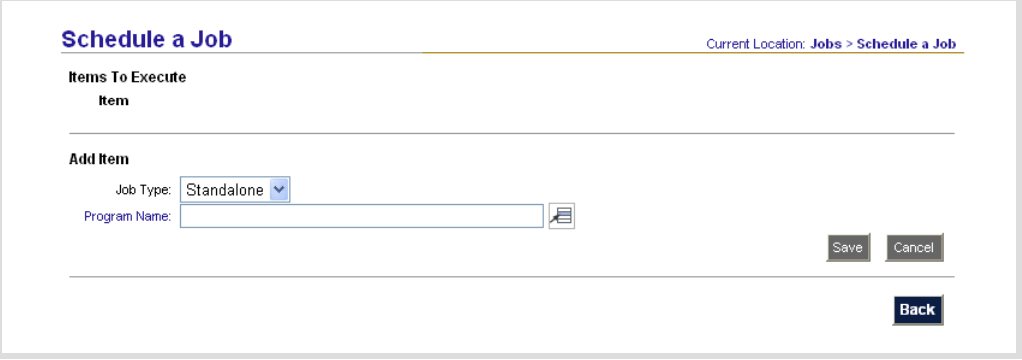
Step 7: Import Data Dictionary into MetaBuilder

Before you can create metadata for your database tables and columns, MetaBuilder needs information for the **MetaBuilder Model Instance**. To retrieve this information, you must run a job to import information stored in the ERP instance's data dictionary.

Follow these steps to run this job:

1 Select **Jobs > Schedule a Job**. The **Schedule a Job** page appears.

2 Click **Add Item...** near the bottom of the page. The **Schedule a Job** page appears.



3 Set **Program Name** to **MetaBuilder: Import Table Definitions from Target**.

4 Enter the following information:

Development Instance	ERP instance where you want to import your data dictionary from.
Target Schema Name	*Optional:* Schema that you want to import from. You should enter this value to avoid performance and tablespace issues. A list of values does not exist for this field. You must type your Schema name *precisely* as it appears in your database dictionary.
Table/View Name	*Optional:* Table or view name to from. Leave this field empty if you want to import *all* of the metadata associated with a Schema.

5 Press the **Save** button. The page is refreshed.

6 Press the **Back** button. The **Schedule a Job** page reappears.

7 Click **Schedule**. The job is scheduled to run immediately.

8 Select **Jobs > View Current/Past Jobs** to view the status of your job.

9 Repeat above steps if you want to import more table or view definitions for a Schema.

Create metadata

Once you have successfully set up your metadata structure, you can begin creating metadata. You can create metadata pertaining to database schemas, tables and columns.

Step 8: Create Schemas

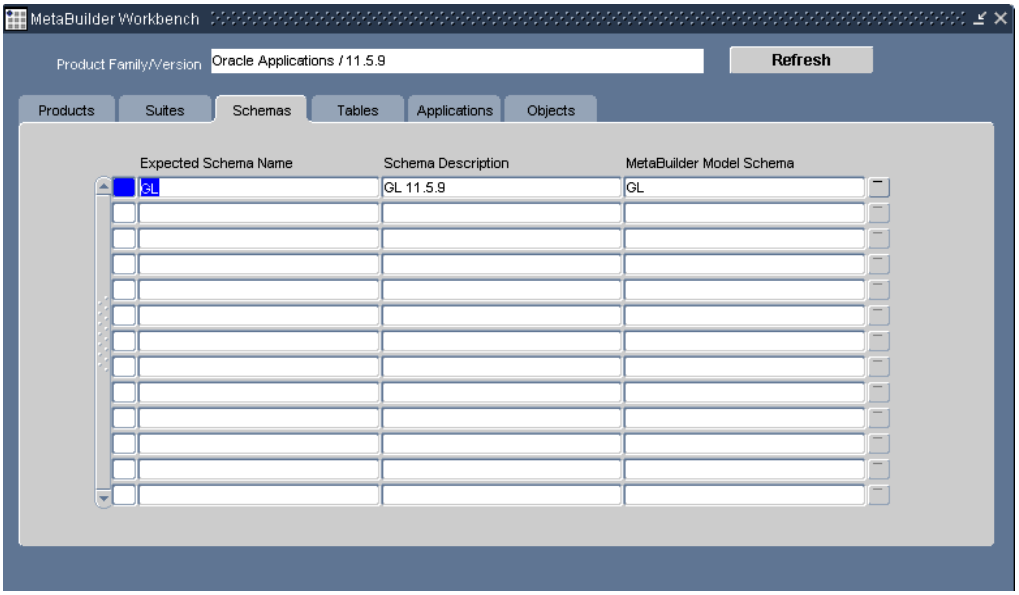
A **Schema** owns a collection of database objects (tables, stored procedures and functions, triggers, synonyms). GL is an example of an Oracle E-Business Suite schema; it contains the database objects associated with the Oracle General Ledger Application.

Follow these steps to create a Schema for your Application:

1 Log into MetaBuilder.

2 From the Navigator, under the **MetaBuilder** role, navigate to the **MetaBuilder Workbench**.

3 On the **MetaBuilder Workbench**, navigate to the **Schemas** tab pertaining to a specific Product Family/Version.



4 Enter the following information to create a Schema, using list of values where applicable:

Expected Schema Name	The precise Schema name found in your database dictionary.
Schema Description	Description of the Schema.
MetaBuilder Model Schema	Schema name pertaining to your Expected Schema Name. The **MetaBuilder Model Schema** list of values contains only database schema names for which you have run the **MetaBuilder: Import Table Definitions from Target** job.

5 Select **Row...Insert** from the menu and repeat Step 4 for each Schema that you want to create under this Product Family/Version.

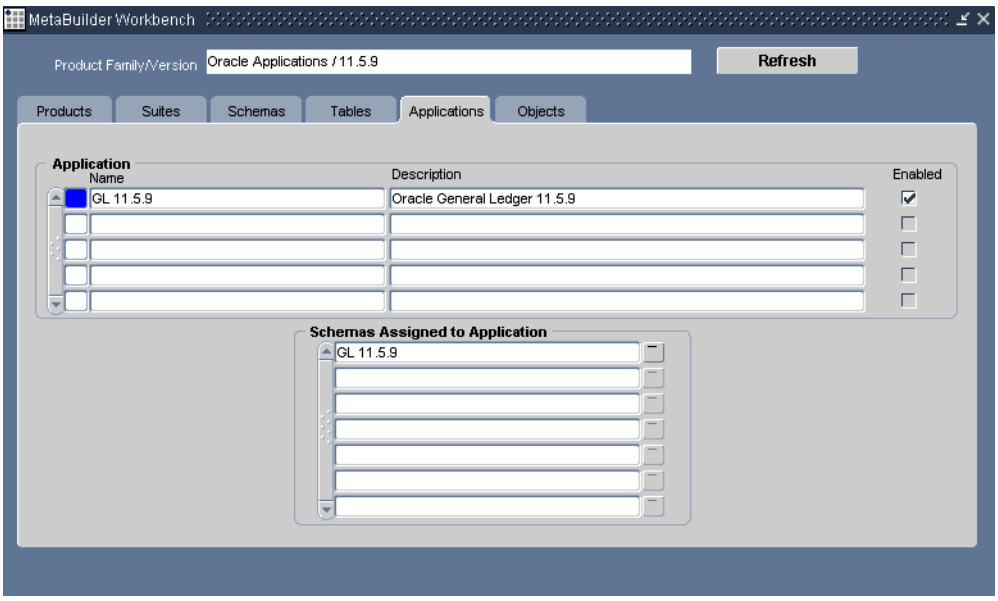
6 Select **Action...Save** from the menu to save your work.

Step 9: Assign Schemas to an Application

An Application is typically comprised of one or many schemas. In order for your Applications to connect to the appropriate Schema(s) in MetaBuilder, you must create relationships between Applications and Schemas.

Follow these steps to assign Schemas to an Application:

1 On the MetaBuilder Workbench, navigate to the **Applications** tab pertaining to a specific Product Family/Version.



2 In the **Application** zone, select an Application to which you want to assign a Schema.

3 In the **Schemas Assigned to Application** zone, select the Schema that you want to assign to this Application.

4 Select **Row...Insert** from the menu and repeat Step 3 for each Schema that you want to add to this Application.

5 Select **Action...Save** from the menu to save your work.

Step 10: Add Tables to a Schema

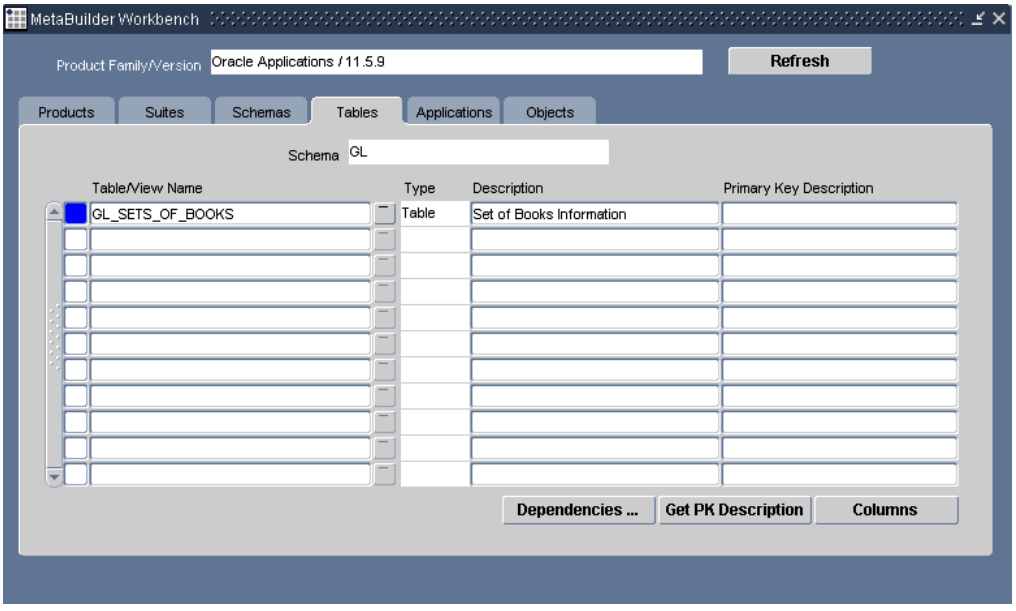
Follow these steps to add Tables owned by a Schema.

> You must add the Tables for which you want to create metadata.

1 On the MetaBuilder Workbench, navigate to the **Schemas** tab pertaining to a specific Product Family/Version.

2 Select the **Expected Schema Name** to which you want to assign a table.

3 Navigate to the **Tables** tab. The MetaBuilder Workbench displays the Schema that you selected from the **Schemas** tab in the **Schema** field.



> You can click the **Dependencies...** button to view the dependencies from a Table/View. MetaBuilder opens the **Table *table-name* Dependencies** window. This is useful if more than one object is based on the same Table/View.

4 Enter the following information to assign a Table to a Schema:

Table/View Name	Table that you want to include in your selected Schema. The list of values for this field is based on the Tables and Views that you imported in Step 7: Import Data Dictionary into MetaBuilder (p. 95).
Type	*Read-only:* MetaBuilder populates this field when you save, based on the information you selected for the **Table/View Name** field. The potential values of this field are **Table** and **View**.
Description	Description of the table.
Primary Key Description	Primary key description for your table or view. You can enter this field after you complete Step 11: Configure Columns within a Table/View (p. 100).

5 Select **Row...Insert** from the menu and repeat Step 4 for each table that you want to add under this Schema.

6 Select **Action...Save** from the menu to save your work.

Step 11: Configure Columns within a Table/View

Once you have added the Tables/Views to your metadata, you must also add/modify Columns.

Follow these steps to add/modify Columns from a Table/View:

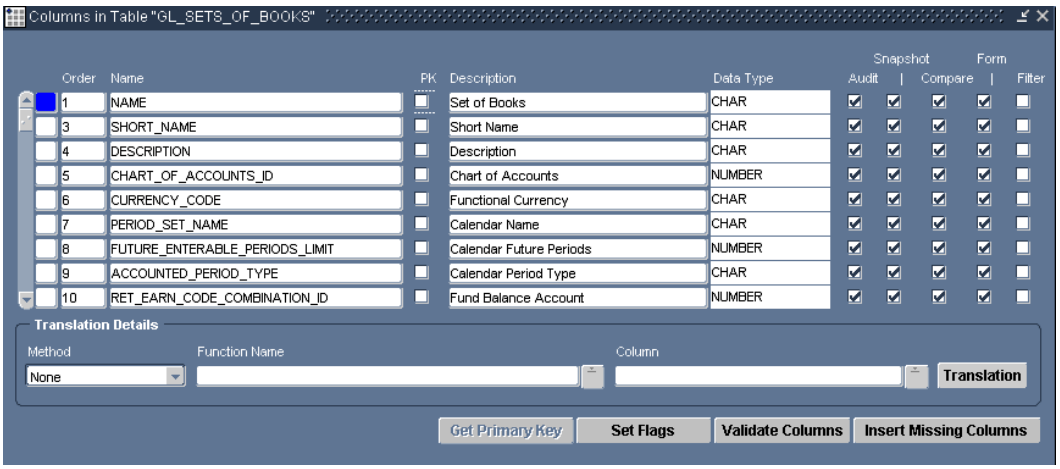
1 On the MetaBuilder Workbench, navigate to the **Schemas** tab pertaining to a specific Product Family/Version.

2 Select the Schema from which you want to select a table.

3 Navigate to the **Tables** tab. The **Schema** field displays the Schema that you selected from the **Schemas** tab.

4 Select the Table to which you want to add/modify Columns.

5 Click the **Columns** button. MetaBuilder opens the **Columns in Table *table-name*** window, listing all the Table's Columns.



6 Enter/modify the following information to define metadata for the Columns:

Order	Order in which you want this Column to appear on Integra Apps reports.
Name	MetaBuilder pre-populates this field with data dictionary information.
PK	Select this checkbox if the Column is part of the Table's primary key. If you do not know whether this Column is part of the Table's primary key, use Step 9 below to retrieve the Table's primary key.
Description	The name of the Column as it will appear in Integra Apps reports.
Data Type	MetaBuilder pre-populates this field with the Column's data type from the data dictionary.

7 Select **Action...Save** from the menu to save your work.

8 Repeat Steps 6 and 7 for each Column in this Table that you want to modify.

9 Click the **Get Primary Key** button to select the Column(s) that will make your record unique. MetaBuilder opens the **Choose PK Constraint** window, containing a list of unique constraints and their Columns.

10 Select the **Constraints Name** that you want to specify as your Table's primary key.

11 Click the **OK** button. MetaBuilder enables the **PK** checkbox for each Column that belongs to the unique constraint you selected. You can reset these checkboxes manually at any time.

When creating Snapshot reports, Integra Apps uses the Columns selected as **PK** to group records. If your Table does not have a primary key or a unique constraint defined, please ensure that your selection of **PK** columns makes each record within the Table unique. Once you define the **PK** columns, navigate back to the **Tables** tab and click the **Get PK Description** button to populate this field.

12 Click the **Set Flags** button to set the **Audit**, **Snapshot**, **Compare**, **Form** and **Filter** checkboxes based on predefined rules for each Column. You can reset these checkboxes manually at any time.

13 Use the following information to decide whether to enable each of these checkboxes:

Audit	Select this checkbox to be able to track changes against the data contained within this Column. ***Oracle E-Business Suite users:*** Your table must meet the following conditions to be able to use Integra Apps' Change Tracking functionality: 1. Your table must contain the standard Who columns (CREATION_DATE, CREATED_BY, LAST_UPDATE_DATE, LAST_UPDATED_BY). 2. Your table's CREATED_BY and LAST_UPDATED_BY columns must be foreign keys of the Oracle Users table (FND_USER).
Snapshot	Select this checkbox to include this column in Integra Apps' Snapshot Reports when you select **Displayed & non-Displayed Fields** option when running a report.
Compare	Select this checkbox to be able to compare this column across points in time or database instances.
Form	Select this checkbox to include this column in Integra Apps' Snapshot Reports when you select the **Displayed fields only** option when running a report. You should not select this checkbox if you do not select the **Snapshot** checkbox.

Filter	Select this checkbox to let Integra Apps users filter on this Column when running a Snapshot Report. You should only use this checkbox when the associated table is a child table of an Object. The columns in the Object's Parent Table are automatically available for filtering whenever **Form** is checked.

14 Select **Action...Save** from the menu to save your work.

15 Press the **Validate Columns** button to perform a validation on your Column metadata configuration. MetaBuilder verifies your checkbox selections, your metadata, and that you designated a Primary Key.

16 Select **Action...Save** from the menu to save your work if you made any changes.

Insert Phantom Columns

Phantom Columns allow you to add another instance of an existing column to your Table in the **Columns in Table...** window. It may be the case that you need to use an existing Column for another data translation for your Snapshot reports. We refer to columns of this type as Phantom Columns.

Follow these steps to add a Phantom Column:

1 Select an **Order** field of the Column record prior to the order in which you want add the Phantom Column.

2 Select **Row...Insert** from the menu to insert a Phantom Column.

3 MetaBuilder inserts a record and increments the **Order** field for the subsequent Columns.

4 Navigate to the **Name** field and select the Column that you want to add again using the list of values.

5 Configure this Phantom Column. Refer to the instructions above.

6 Select **Action...Save** from the menu to save your work.

Insert Missing Columns

It may be the case that you inadvertently deleted a Column or you have upgraded your ERP application version. You can add this new or deleted Column using the **Insert Missing Columns** button.

If you have upgraded your ERP application version and you want to add missing Columns, you must verify that you have changed your **MetaBuilder Model Instance** to the upgraded ERP instance, and you must import the Table information again using the **MetaBuilder: Import Table Definitions from Target** program.

Follow these steps to insert a missing Column:

1 In the **Columns in Table...** window, press the **Insert Missing Columns** button.

2 Configure this Column. Refer to the instructions above.

3 Select **Action...Save** from the menu to save your work.

Step 12: Assign Translation Details to Columns

Your Table's Columns might require information that is stored in another Table, or you might want to use data from a different Column within the same table. Use the **Translation Details** zone to generate this information.

You have three options for translating data:

Column: Assign the Column data from another Column of the same Table

Function (New): Create and assign the Column a new Integra Translation Function.

Function (Existing): Assign the Column a predefined Integra Translation Function.

Each method is detailed in the sections below.

Column

You use this option if you have an inactive Column and want to display another Column data of the same table that you already placed a Translation Function or you already assigned another Column data. Thus, you do not have to insert a Phantom Column.

1 Select the Column to which you want to assign the data of another Column.

2 In the **Translation Details** zone, enter the following information:

Method	Select **Column** from the drop-down list.
Function Name	MetaBuilder disables this field for this option.

Column	Select the Column containing the data you want to display.

3 Select **Action...Save** from the menu to save your work.

Function (Existing)

Use this option if the data that you want to display is stored in more than one table, or if you store data as an ID rather than a value, and would like to provide the full description of the ID in your Snapshot reports. An example of a Translation Function is the conversion of a Code Combination ID in the Oracle General Ledger to a multi-segment accounting flexfield value.

1 Select the Column that you want to assign a Translation Function.

2 In the **Translation Details** zone, enter the following information:

Method	Select **Function** from the drop-down list.
Function Name	Select the Translation Function that you want to apply to this field. To verify or test the selected **Function Name**, click the **Translation** button to view the Translation Function's definition. Integra's metadata uses the following name convention: `tick_app_version.function-name` ...where: `app` is the ERP application's short name `version` is the abbreviated version number (e.g., `107`, `1103`, `1157`, `1158`, `1159`) `function-name` is the descriptive name of the function You can find existing functions by navigating to the **Translation Function** window and clicking the **Find** button. MetaBuilder opens the **Find Function** window. Refer to "Function (New)" below for more information. You can also check which Columns, if any, are applying this Translation Function. Navigate to the **Translation Function** window, and press the **Referenced By** button.

Column	MetaBuilder disables the **Column** field for this method.

3 Select **Action...Save** from the menu to save your work.

Function (New)

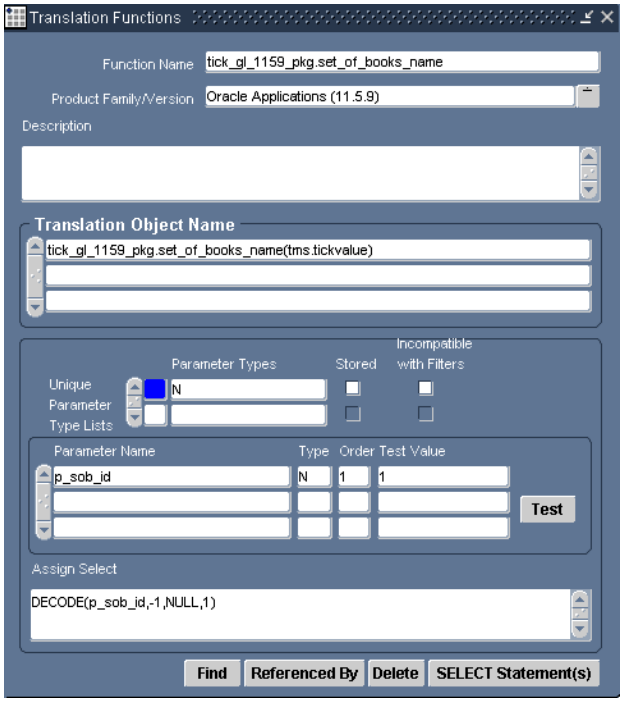
This option is equivalent to **Function (Existing)** above in all respects but one: the Translation Function that you need does not exist yet. You will create a new Translation Function and apply it to the Column that you want to translate.

> You may need the assistance of a technical resource (e.g., a developer) to create a data translation using this option.

Follow these steps to create and apply a new Translation Function:

Create a Translation Function

1 From the Navigator, under the MetaBuilder Role, select **Navigate...Maintain Metadata...Translation Functions**. MetaBuilder opens the Translation Functions form.



2 Enter the following information to create a Translation Function:

Function Name	Name of your Translation Function. You should name your Translation Functions logically in order to facilitate the process of finding and reusing these functions.
Product Family/Version	Select a Product Family/Version to which you want to apply this function.
Description	Description of this function.

Translation Object Name	This is usually a concatenation of a Function Name and parameter names to be passed to the Function.
	
	You must enter the **Translation Object Name** in this form:
	
	`name(tms.tickvalue)`
	
	...where:
	
	`name` is the Function Name
	
	The variable **tms.tickvalue** references the Table Column record on which you apply the function. If you specify the actual column name instead of using this variable, you will not be able to apply this function to a column within another table (e.g., one that stores the same information but has a different column name).
	
	You can include a maximum of four parameters per Translation Function:
	
	`name(tms.tickvalue, tms.Column1, tms.Column2, tms.Column3)`
	
	...where:
	
	`Column1`, `Column2`, `Column3` are the column names in the source table. As shown above, precede the column names with `tms.`

Parameter Types

Enter the Column data types for the parameters that you pass to your Translation Function:

Enter:	...for data type:
C	Character
N	Number
D	Date

If you have multiple parameters for your Translation Object Name, separate the data types with a comma (,).

This form does not support parameters of other types.

Stored

If you must store the Translation Function on your ERP instances, follow the instructions in "Creating a Stored Function" below, then select this checkbox.

Generally, you do not need to do this. You should only store functions on your ERP instances if the **Maintain Translation Functions** form cannot support the complexity of your function. For example, IF statements and multiple SUBSTRING/INSTRING operations are typically created as stored functions.

Incompatible with Filters

Select this checkbox if the information in the **Select Statement(s) for** window (see Step 4 below) is not compatible with your stored procedure. This disables your ability to update the data in the **Select Statement(s) for** window.

This checkbox only works when the **Stored** checkbox is selected.

Parameter Name

Name of the parameter.

Type

Data type of the parameter.

Order

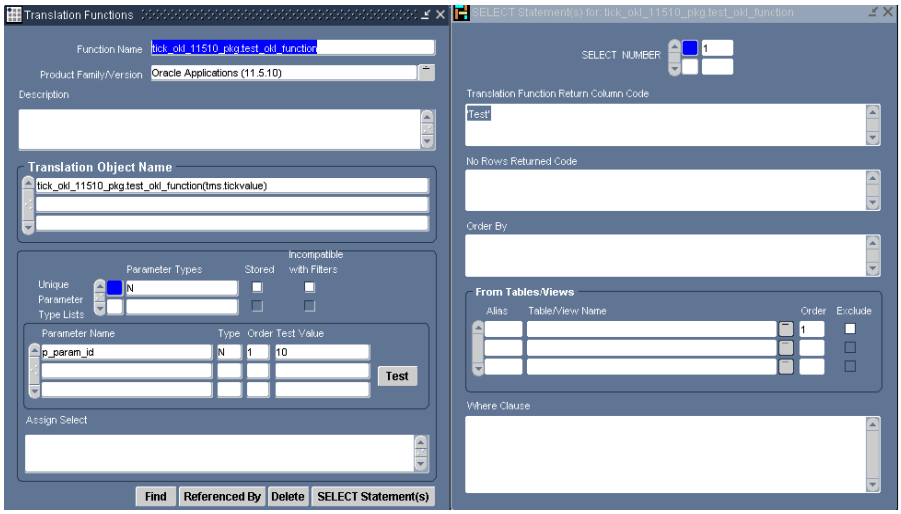
Order in which you are passing the parameter in the **Translation Object Name** field.

Assign Select Enter a function if your translation function contains multiple SELECT statements or you want to manipulate the parameter data that you are passing.

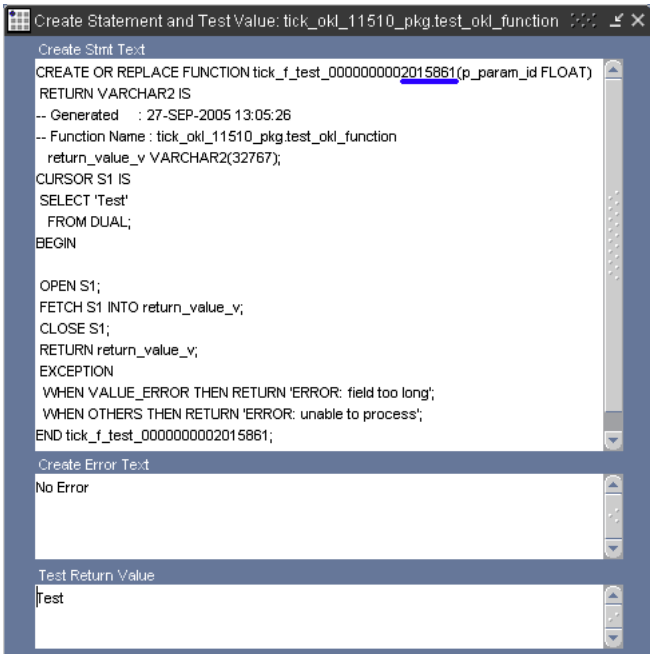
Typically, your translation functions will contain only one SELECT statement. In this case, you should leave this field blank.

Creating a Stored Function:

a. Click the **SELECT Statement(s)** button and enter `'test'` (or any other string surrounded by single-quote marks) in **Translation Function Return Column Code**.



b. Click the **Test** button and make note of the long number that appears in the first line of **Create Stmt Text** (underlined in this figure; discard the leading zeroes).



c. Using your database editor, find the row in the Integra Home schema's **TICK_META_FUNCTION_OVERLOADS** table where **meta_function_overload_id** equals the value noted in the preceding step.

d. In the same row, the **create_stmt_text** column holds the statement that defines the function. Modify the statement as needed to perform the desired function.

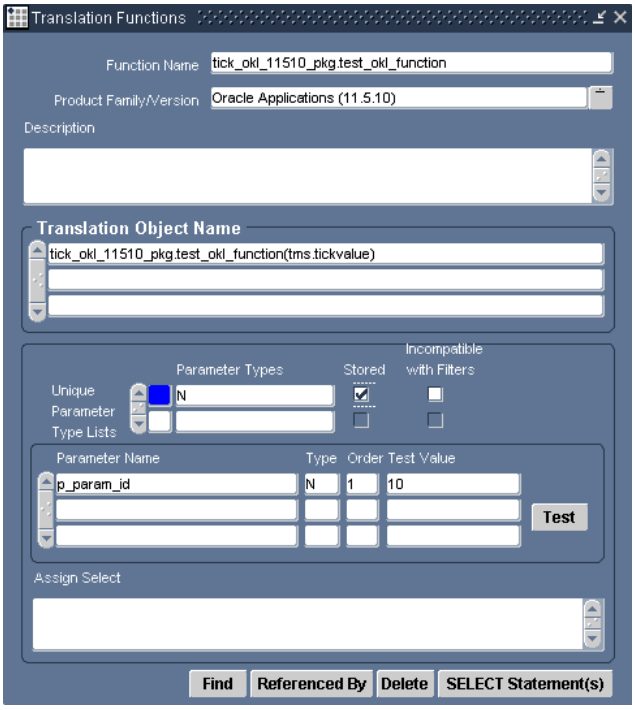
e. Integra uses an **internal function name** instead of the name designated in MetaBuilder, employing this format:

tick_f_test_################

Remove the string **test_** from each occurrence of the internal function name (in the first and last lines of the statement).

f. Copy the new function name to the **function_overload_name** column.

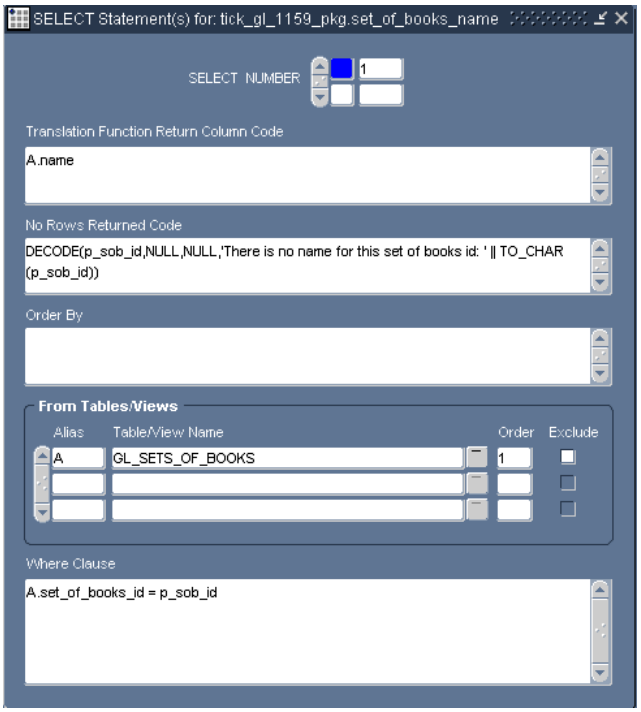
g. Return to MetaBuilder, and check the Stored checkbox.



3 Select **Action...Save** from the menu to save your work.

Press the **Delete** button if you want to delete your Translation Function. MetaBuilder raises a message to confirm your command.

4 Press the **SELECT Statement(s)** button to create a SELECT statement for your Translation Function. MetaBuilder opens the **SELECT Statement(s) for** window.



5 Enter the following information to create a SELECT statement for your Translation Function:

SELECT NUMBER	MetaBuilder defaults this field to **1**. Retain this value for your first SELECT statement. If you need to create another SELECT statement, press the down arrow key and increment the number.
Translation Function Return Column Code	Database column to be returned by the Function. You also have the option to perform SQL functions on a database column.
No Rows Returned Code	Enter a message if you want to return a message when your function input value(s) return a NULL value. Your message must be enclosed in single quotes ('). If your Translation Function is a valid function and its input values are NULL, MetaBuilder returns a NULL value.

Order By	Database column name used to sort the results of the SELECT statement. Try to make your SELECT statement identify a unique value. If your statement returns more than one record, Integra Apps uses the first translated value from the sort order.
Alias	Alias of the table or view from which you are selecting values. You must denote any reference to any table's columns with the column name preceded by the Alias.
Table/View Name	Database table or view from which you are selecting values. If the table or view does not exist in the list of values, you must add your table in the MetaBuilder Workbench and run **MetaBuilder: Synchronize** to synchronize your ERP instance prior to creating the function.
Order	Number that specifies the order of the tables in the SELECT statement.
Exclude	Select this checkbox for any table from which you are not directly selecting information.
Where Clause	Condition(s) of your SELECT statement. Do not include the word `WHERE` in this field.

6 Select **Action...Save** from the menu to save your work.

Test the Translation Function

After you define your Translation Function, you must verify that it compiles. You have the option to specify test values for each input parameter so that you can verify the Function's results.

First, synchronize the ERP instance that you will test the function on:

7 Log into Integra.

8 Select **Jobs > Schedule a Job**. The **Schedule a Job** page appears.

9 Click **Add Item...** near the bottom of the page. The **Items To Execute** page appears.

10 Set **Program Name** to **MetaBuilder: Synchronize**.

11 Enter the following information:

Target Instance Name	Select the ERP instance that you want to synchronize.

12 Press the **Save** button. The page is refreshed.

13 Press the **Back** button. The **Schedule a Job** page reappears.

14 Click **Schedule**. The job is scheduled to run immediately.

15 Select **Jobs > View Current/Past Jobs** to view the status of your job.

> Wait until this program has completed successfully prior to viewing your changes to the reports. The MetaBuilder: Synchronize program also checks that your metadata data structure complies with the data dictionary.

Next, return to MetaBuilder and follow these steps to test your new Translation Function:

16 To test the Translation Function using specific parameter values, enter the values in the **Translation Functions** window's **Test Value** fields. Otherwise, leave the **Test Value** fields empty.

17 Select **Action...Save** from the menu to save your work.

18 Click the **Test** button to test your Translation Function. MetaBuilder opens the **Create Statement and Test Value** window.

This window contains the following three fields:

Create Stmt Text:	Contains the code that Integra Apps runs for this Translation Function. Use this field to verify that your definition is correct and/or to identify possible mistakes.
Create Error Text	Lists the database error encountered if your Translation Function contains a syntax error. This field contains the value **No error** if your Translation Function compiles on the ERP instance without errors.
Test Return Value	Contains the value returned from your Translation Function using your Test Value(s). If you did not specify any input parameters, this field is empty.

Apply the New Translation Function

Once you have verified that your Translation Function compiles and returns the desired output, you must apply this function to your metadata Column.

19 In the MetaBuilder Workbench, navigate to the **Schemas** tab pertaining to a specific Product Family/Version.

20 Select the Schema that owns the Table containing the Column that you want to translate.

21 Select the Table containing the Column that you want to translate.

22 Click the **Columns** button. MetaBuilder opens the **Columns in Table** window, listing all the Columns that define this Table.

23 Select the Column to which you want to assign a Translation Function.

24 In the **Translation Details** zone, enter the following information:

Method	Select **Function** from the drop-down list.
Function Name	Select the new Translation Function. To verify or retest the **Function Name** that you selected, press the **Translation** button to view the Translation Function's definition.
Column	MetaBuilder disables the **Column** field for this method.

25 Select **Action...Save** from the menu to save your work.

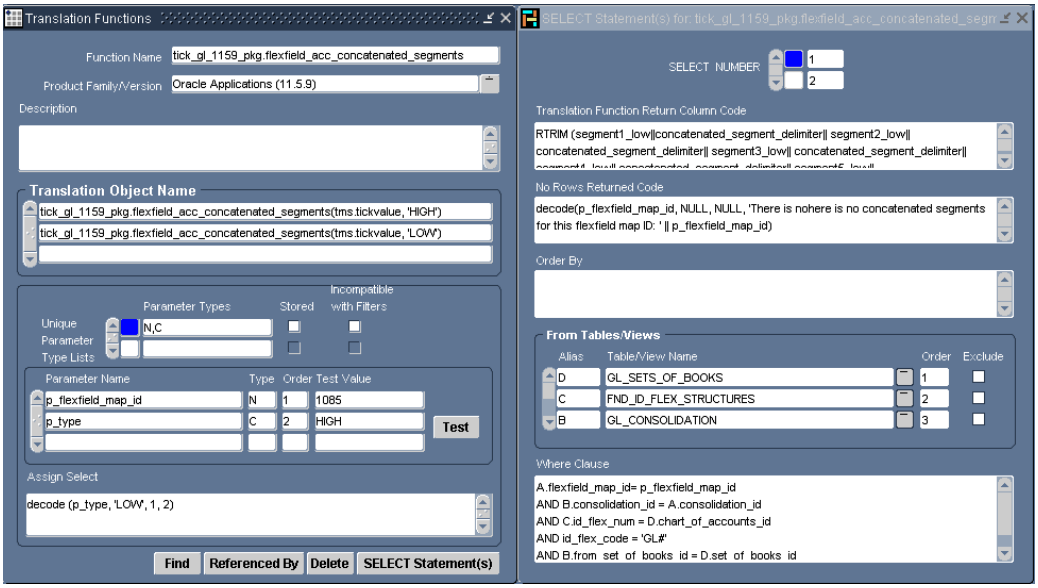
Add Multiple SELECTS to a Translation Function

The **Translation Functions** form has the capability to handle multiple SELECT statements for a Column translation. For example, you may need to create different SELECT statements based on the value of another Column in the table. In this case, you must use the **Assign Select** field to handle the multiple SELECT statements.

Follow these steps to add multiple SELECT statements to a Translation Function:

1. In the **MetaBuilder** role, navigate to the **Translation Functions** form.
2. Query the Translation Function to which you want to assign multiple SELECT statements.
3. In the **Assign Select** field, enter a SQL function that assigns the correct **SELECT NUMBER** to the Translation Function based on certain conditions. Typically, you should use the DECODE function.
4. Press the **SELECT Statement(s)** button to create additional SELECT statements for your function. MetaBuilder opens the **SELECT Statement(s)** window for your function.
5. In the **SELECT NUMBER** field, navigate to the next record to create another SELECT statement.
6. Increment the **SELECT NUMBER**.
7. Enter the information for your SELECT statement. Refer to "Create a Translation Function" above.
8. Repeat Steps 4-7 to create additional SELECT statements.
9. Select **Action...Save** from the menu to save your work.

In the following example, if the parameter p_ type = LOW, the Translation Function will use SELECT statement 1. Otherwise, the Translation Function will use SELECT statement 2.



Step 13: Create an Object

A MetaBuilder Object is a group of database tables that store information for specific business functions. You must define an Object within an Application.

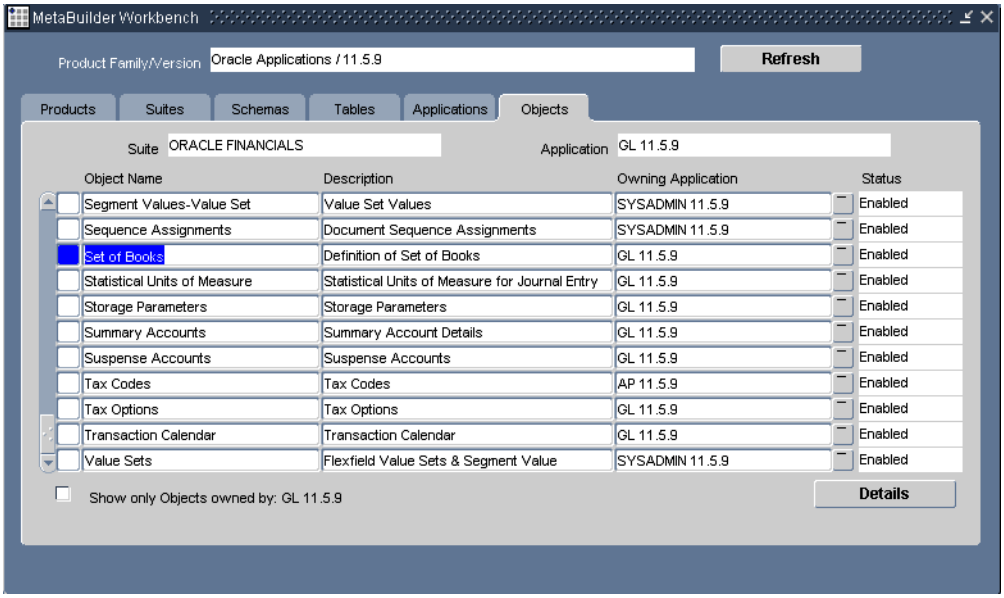
You can create Objects based on one or more tables. There is no limit to the number of tables that you can link within an Object.

Follow these steps to create an Object associated with your Application:

1 In the MetaBuilder Workbench, navigate to the **Applications** tab pertaining to a specific Product Family/Version.

2 Select the Application for which you want to create Objects.

3 Navigate to the **Objects** tab. The MetaBuilder Workbench displays the Application that you selected in the **Applications** tab along with its associated Suite information.

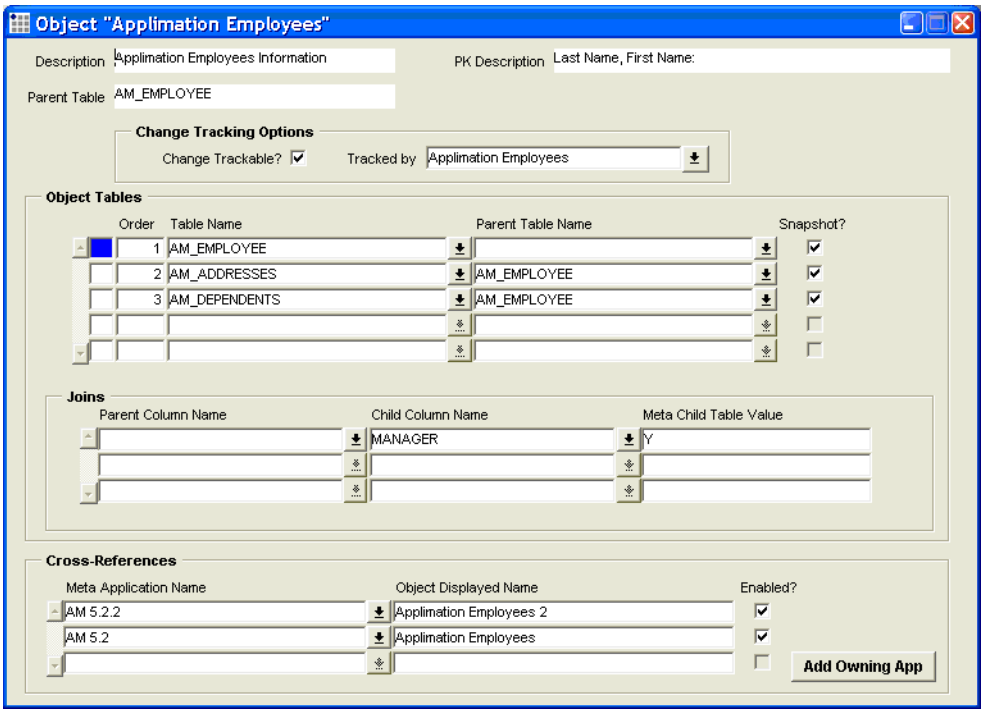


4 Enter the following information to create an Object:

Object Name	Identifiable name for the Object.
Description	Description of the Object.
Owning Application	Application that will own the Object.
Status	*Read-only:* MetaBuilder defaults this field to **Enabled**.

5 Select **Action...Save** from the menu to save your work.

6 Press the **Details** button to enter your Object's table information. MetaBuilder opens the **Object** window.



MetaBuilder populates the **PK Description** field and **Parent Table** field after you enter your Object's Table information (in the following step).

7 Select **Action...Save** from the menu to save your work.

8 Specify the Object's Parent Table in the **Object Tables** zone:

Order	Enter `1` to identify this table as the Object's Parent Table. All other tables must be descendents of this table.
Table Name	Parent Table. The list of values for this field contains only the tables that you have previously added in Step 10: Add Tables to a Schema (p. 99).
Parent Table Name	Leave this field empty, because the Parent Table has no parent.
Snapshot	Select this checkbox to include this table in Snapshot Reports.

9 Select the Change Tracking options for the table:

a. Select the **Change Trackable?** checkbox to allow Change Tracking for this Object. If this is an Object that contains tables included in other Objects that are Change Trackable, leave this checkbox unchecked.

Oracle E-Business Suite users: Your table must meet the following conditions to use Integra Apps Change Tracking functionality:

1. Your table must contain the standard Who columns (CREATION_DATE, CREATED_BY, LAST_UPDATE_DATE, LAST_UPDATED_BY).

2. Your table's CREATED_BY and LAST_UPDATED_BY columns must be foreign keys of the Oracle Users table (FND_USER).

b. Select a value for **Tracked by** field if this Object is not Change Trackable because another Object contains Tables from your Object. Select the other Object that contains your Object's Tables.

Be sure to include all Tables that you want to change track from the Object in the **Tracked by** field with the **Snapshot?** checkbox deselected.

10 Select **Action...Save** from the menu to save your work.

11 To add more Tables if an Object is comprised of more than one table:

a. Select **Row...Insert** from the menu

b. Enter the following information to specify an Object's Child Table information in the **Object Tables** zone:

Order	Enter a number for the order in which you want the table to appear in Snapshot Reports.
Table Name	Child Table to add to the Object.
Parent Table Name	Parent Table on which the Child Table is based. This value should be a table that has already been added to the Object.
Snapshot?	Select this checkbox to include this table in Snapshot Reports.

c. Enter the following information to link the Parent Table to the Child Table in the **Joins** zone:

Parent Column Name	Column from the Parent Table that has a foreign key reference to the Child Table.
Child Column Name	Column from the Child Table that has a foreign key reference to the Parent Table.
Meta Child Table Value	Enter a value if you want to constrain on the records returned based on this Column value. This field is case-sensitive.

d. Set Change Tracking for the table (see the preceding steps).

e. Select **Action...Save** from the menu to save your work.

You must verify that all Tables used in a Translation Function for the Object are included in the Object's Tables information, or accounted for in another Object.

f. Repeat Step 10 to insert additional Child Tables.

You must add any translated tables associated with your Parent and Child Tables. This allows you to enable Change Tracking for these translated tables. You should leave the **Snapshot?** checkbox deselected.

12 Click the **Add Owning App** button to allow the **Owning Application** field to view the Object. MetaBuilder creates an enabled record in the **Cross-references** zone for the Object.

13 Select **Action...Save** from the menu to save your work.

Cross-Reference an Object to another Application

You might have several Applications that share the same or similar configuration data. MetaBuilder lets you assign or cross-reference metadata Objects across Applications to eliminate the need to create duplicate metadata.

Follow these steps to cross-reference an Object to another Application:

1 In the **Object** window, navigate to the **Cross-References** zone.

2 Select **Row...Insert** from the menu to enter the following information to create a cross-reference Object:

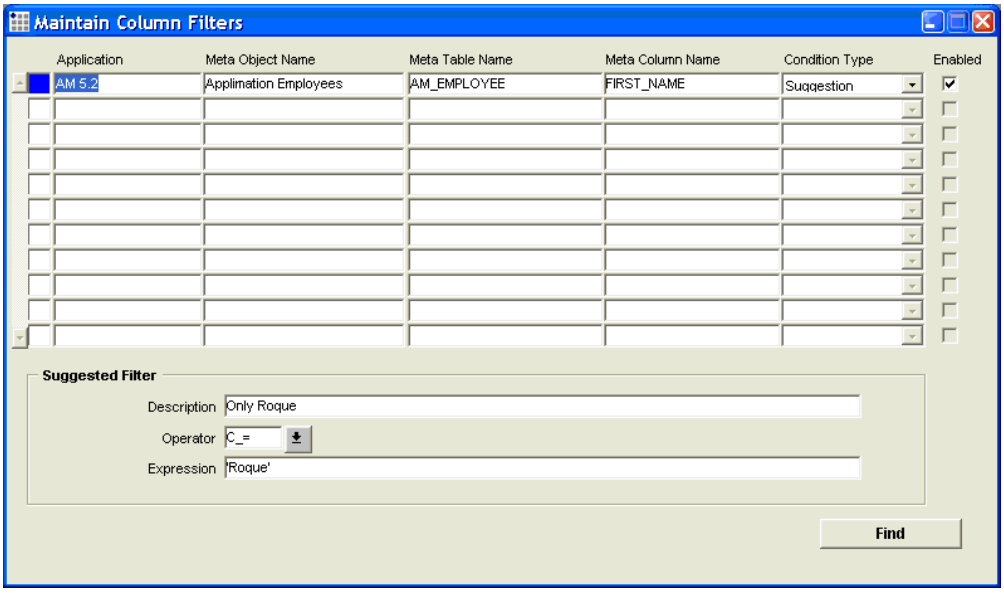
Meta Application Name	Application to which you want to cross-reference the Object.
Object Displayed Name	Name of the Object. Typically, this value equals the original Object Name that you are cross-referencing. You may change this value to reflect the exact Object name for the other Application.
Enabled?	MetaBuilder selects this checkbox automatically to enable the Object in the Meta Application specified. You may deselect the checkbox if you do not want to run Snapshot Report for that Application's Object.

3 To cross-reference additional Applications, repeat the preceding step.

4 Select **Action...Save** from the menu to save your work.

Maintain Column Filters

Security Filters. If you want to add data security to the Object, use the **Maintain Column Filters** form.



Oracle E-Business Suite users: If your custom Objects are associated with organizations and sets of books, you have two options for security:

ORG Security. Create a filter of this type to allow Oracle E-Business Suite organization security. You should only apply this filter type to Meta Column Names that are associated with the ORGANIZATION_ID column in the HR_ALL_ORGANIZATION_UNITS table.

SOB Security. Create a filter of this type to allow Oracle E-Business Suite set of books security. You should only apply this filter type to Meta Column Names that are associated with the SET_OF_BOOKS_ID column in the GL_SETS_OF_BOOKS table.

When you create the Object in the MetaBuilder Workbench, the workbench automatically adds a security record in this form for the columns ORGANIZATION_ID, ORG_ID and SET_OF_BOOKS_ID.

PeopleSoft Enterprise users: If your custom Objects are associated with SETID, the MetaBuilder Workbench also automatically adds a SETID Security record in this form pertaining to your Object.

Suggested Filters. You may want to add Suggested filters that users can remove when creating definitions. For example, PeopleSoft tables contain historic data that users might not want to view in Snapshots. A Suggested filter could remove that data; users who want to see only the latest and future data would leave the filter in place, while those who want to view historic data would remove it.

Follow these steps to create a Suggested filter:

1 From the Navigator, under the **Integra MetaBuilder User** role, select **Navigate...Apps Metadata...Maintain Column Filters**. MetaBuilder opens the Maintain Column Filters form.

2 Enter the following information (using **Edit...List of Values** from the menu where applicable):

Application	Name of the Application.
Meta Object Name	Name of the Object.
Meta Table Name	Name of the Table (within the Object) on which you want to apply the Suggested filter.
Meta Column Name	Name of the Column on which you want to apply the Suggested filter.
Condition Type	Select **Suggestion** from the dropdown list.

Enabled	Check this box to enable the Suggested filter.
Description	Description of the Suggested filter.
Operator	Operator for the condition that will be applied to the Meta Column Name during the Snapshot run. The list of values for this field changes based on the data type.
Expression	Enter the expression for the condition precisely as you would in a WHERE clause of a SELECT statement. For example, enter single quotes before and after literal values.

3 To create another Suggested filter on this Object, select **Row...Insert** from the menu and repeat the preceding step. You may create as many Suggested filters as you like.

4 Select **Action...Save** from the menu to save your work.

Copy a Standard Integra Object

You may want to modify the metadata of a standard Integra Object to better manage your business requirements. This Integra program lets you copy and modify a standard Integra Object instead of having to recreate metadata that already exists.

Follow these steps to copy and modify a standard Integra Object.

1 Log into Integra.

2 Select **Jobs > Schedule a Job**. The **Schedule a Job** page appears.

3 Click **Add Item...** near the bottom of the page. The **Items To Execute** page appears.

4 Set **Program Name** to **MetaBuilder: Copy Standard Integra Object**.

5 Enter the following information:

Copy From Product	Product Family Version from which you want your Application.
From Application	Application from which you want your Integra Object.
From Object	Integra Object that you want to copy.
Copy to Product	Product Family Version to which you want to your Application.

To Appplication	Application to which you want to create the object.
New Object Name	Name for your new Object.

6 Press the **Save** button. The page is refreshed.

7 Press the **Back** button. The **Schedule a Job** page reappears.

8 Click **Schedule**. The job is scheduled to run immediately.

9 Select **Jobs > View Current/Past Jobs** to view the status of your job.

10 Repeat the steps above to copy additional standard Integra Objects.

> Use this program only when you have not imported and manually added Tables/Views that are the underlying metadata of the Integra Objects you intend to copy.

Step 14: Synchronize

After you make changes or create new metadata within the MetaBuilder Workbench, you must run a synchronization process to update the metadata for your ERP instances.

> Integra Apps does not display metadata changes if you do not run the following program.

Follow these steps to run this program:

1 Log into Integra.

2 Select **Jobs > Schedule a Job**. The **Schedule a Job** page appears.

3 Click **Add Item...** near the bottom of the page. The **Items To Execute** page appears.

4 Set **Program Name** to **MetaBuilder: Synchronize**.

5 Enter the following information:

Target Instance Name	Select the ERP instance that you want to synchronize.

6 Press the **Save** button. The page is refreshed.

7 Press the **Back** button. The **Schedule a Job** page reappears.

8 Click **Schedule**. The job is scheduled to run immediately.

9 Select **Jobs > View Current/Past Jobs** to view the status of your job.

> Wait until this program has completed successfully prior to viewing your changes to the reports. The MetaBuilder: Synchronize program also checks that your metadata data structure complies with the data dictionary.

10 Repeat Steps 2-9 to synchronize additional ERP instances.

Step 15: Manage Security Groups

Security Groups control the ERP data that users can access. Each group has several Permissions; one of them determines the **Applications** whose data can be accessed.

Users who belong to groups with the **Applications** permission left blank can access data from all Applications, including the new Application you have created in the preceding steps. No other users will have access to the new Application's data unless you:

Add them to a group with the **Applications** permission left blank

...or:

Create a new Security Group with **Applications** set to the new Application, and add the users to the group

For more information about Security Groups, see Create users, roles and security groups (p. 19).

Step 16: Test Metadata

Once you have created your metadata, you must verify that Integra Apps monitors your Objects accurately when using it. Follow these steps:

1 Log into Integra.

2 Create and schedule one or more Snapshot definitions. Verify that their reports contain the data you see in your ERP applications. For instructions, see "How to... Create Snapshots of ERP application setups" in the *Integra Apps 5.1 User Guide*.

3 Compare two Snapshot reports and verify the results. For instructions, see "How to... Compare Snapshots" in the *Integra Apps 5.1 User Guide*.

4 Create and deploy one or more Change Tracking definitions. Schedule and run the **Change Tracking Transfer** program. Verify that the Change Tracker contains the correct data. For instructions, see "How to... Track changes to ERP application setups" in the *Integra Apps 5.1 User Guide*.

MetaBuilder glossary

Application An Application is a set of forms, menus, reports and other components that satisfies a particular business function. General Ledger is an example of an Application. The Application stores the accounting and tracking of fixed assets for a site.

Columns Columns are subdivisions of a table with a column name and data type.

Configure A Configure process configures your ERP instance for your Integra solution component. This process allows your Integra users to run Snapshots, Translation Functions, Change Tracking and other jobs.

Cross-references Two or more Applications can share configuration data; to eliminate the need to create metadata multiple times, Integra Apps let you assign (cross-reference) metadata objects across Applications.

Data Dictionary A data dictionary is a central source of information for the database itself and for its users. This dictionary contains a comprehensive set of all tables and views within the database.

MetaBuilder: Copy Standard Integra Object A program that allows a user to copy an object from Standard Integra Objects to the user's custom applications.

MetaBuilder: Import Table Definitions from Target A program used to import information from an ERP instance's database dictionary into an Integra table that stores data dictionary information for which a user would like to track via metadata.

MetaBuilder Model Instance Database instance where the data dictionary information that you use for metadata development is kept.

MetaBuilder: Synchronize A program that copies metadata to ERP instances.

Object An Object is a logical grouping of database tables that store information based on specific business functions within an Application. Payables Options is an example of a setup object on which a user can base a Snapshot definition.

Phantom Column A Phantom Column is a duplicate Column. Please refer to Columns.

Product Family A Product Family is a software product that provides a business solution. Product Families often contain different versions. Software companies typically provide increased functionality within their product with each version that they release. Oracle E-Business Suite is an example of a Product Family. It is a product that supports many different business functions.

Product Family Version A Product Family Version is a specific release of a Product Family. Most Applications usually denote versions with numbers. Version 11i of Oracle E-Business Suite is an example of a Product Family Version.

Schemas A Schema owns a collection of database objects, such as tables, stored procedures and functions, triggers, synonyms, etc. AP is an example of an Oracle E-Business Suite schema. Within the database, the AP schema contains all database objects associated with the Oracle Payables Application.

Suite A Suite is comprised of a number of Applications within a Product Family, usually corresponding to a group of inter-related business processes. Oracle Financials is an example of a Suite. The software contains a number of different Applications (GL, AP, AR, FA, etc.) to support the financial accounting of a business.

Table A Table is the basic data storage structure in a relational database management system. Tables consist of rows and columns.

Translation Function If your data is stored as an ID rather than as a literal value, and you would like to provide the full description of the ID on your Snapshot Definition output, you use a Translation Function. An example of a Translation Function is the conversion of a Code Combination ID (CCID) in the Oracle General Ledger to a multi-segment accounting flexfield value.