IBM TRIRIGA Application Platform
Version 3 Release 6

Application Building for the IBM TRIRIGA Application Platform

IBM

Contents

Chapter 1. Application building for the IBM TRIRIGA Application Platform

You use the IBM® TRIRIGA® Application Platform to run, build, and manage the IBM TRIRIGA applications. Application building for the IBM TRIRIGA Application Platform includes data modeling, data types, lifecycles, workflows, user interfaces, workflows, temporary data, hierarchies, events, integration with external applications, notifications, security, and offline forms.

Introduction to IBM TRIRIGA Application Platform

The IBM TRIRIGA Application Platform creates applications that are more valuable than applications that are built on other platforms because of its focus on business processes and its delivery of business objects.

Business processes

Unlike traditional technologies, the IBM TRIRIGA Application Platform allows a person who builds applications to focus on what an application is supposed to do rather than how to do it. By separating the programming details from the business processes, the result is usually an excellent fit for business needs.

Even better, many people need only a few weeks of training to be able to build applications with the IBM TRIRIGA Application Platform. Because years of programming training are not required, business people who know what the application is supposed to do can build an application without the services of a programmer.

Building applications on the IBM TRIRIGA Application Platform is a relatively rapid process that does not require the assistance of programmers in most cases. Customizing and modifying applications to meet changing business needs are also relatively rapid processes. The rapid building, customizing, and modifying applications by business people who need only a few weeks of training results in much lower business costs.

Business objects

The IBM TRIRIGA Application Platform is delivered with predefined business objects that are used to describe real-world entities. These business objects can be used to represent things that are needed by most business applications, such as people, currencies, and schedules. Building applications that use these predefined business objects saves people time that they would otherwise spend to create these objects themselves.

Building applications that use these predefined business objects has another more important benefit. Applications that use the same common business objects to represent the same data automatically share the data that is contained in the business objects. The data in these common business objects does not need to be entered twice.

Licenses

To build or customize applications on the IBM TRIRIGA Application Platform, you need to access many features that are available to you only if your user ID is assigned with an IBM TRIRIGA Application Platform license. If you do not have an IBM TRIRIGA Application Platform license, ask your platform administrator or IBM TRIRIGA representative about obtaining an IBM TRIRIGA Application Platform license.

Overview of IBM TRIRIGA Application Platform

The IBM TRIRIGA Application Platform is a self-contained environment for building and running business applications. Built into the platform is much of the common logic that is used for business applications.

To build a simple application that runs in the IBM TRIRIGA Application Platform environment, you need four things:
- A description of how the data in the application is organized.
- A description of what the user interface of the application looks like.
- The descriptions of the reports and queries that the application supports.
- Any custom logic for the business processes that the application supports.

Currently, the full set of builder tools is not yet globalized. To avoid the confusion of seeing some text in English and some text in another language, application builders must be logged in as US English users.

Data model

The IBM TRIRIGA Application Platform maintains a description of the data that applications use. This description is called the data model.

When you use the IBM TRIRIGA Application Platform to build an application, you start with the data model. The data model supports the other elements of the application such as the user interface, the reports, and the custom logic.

The data model is organized into five main parts:
- Field definitions describe individual pieces of data. A field contains an individual piece of data.
- Business objects describe sets of fields that are stored and manipulated together. Business objects are used to create records. A record contains actual data for the fields that are described by the business object.
- Associations describe how business objects relate to each other.
- Modules organize business objects. A module is a collection of business objects. Each business object belongs to exactly one module.
- State transitions control the lifecycle of records that are created from a business object. A typical lifecycle of a record follows a path of being created, modified, and deleted. But the specific lifecycle is determined by the state transitions in the business object.

The IBM TRIRIGA Application Platform provides the Data Modeler tool to describe and manage modules, business objects, associations, and field definitions. The Data Modeler is described in "Data modeling".

The Data Modeler also describes and manages the state transitions that are used by records. The IBM TRIRIGA Application Platform provides a library of predefined state transition families that you can reference when you set up your state transitions for a business object. This library is managed by the State Family Manager tool. The State Family Manager is described in "Life Cycles".

Data model is independent of the user interface

Typically, most business objects in a data model might closely match the structure of information that is visible in the user interface. However, this practice is not a requirement.

A business object might be used to create records that are used for behind-the-scenes processing only. As a result, the information in such records might not appear anywhere in the user interface.

User interface

The user interface of an application on the IBM TRIRIGA Application Platform has three essential elements: portals, navigation items, and forms. You access the main parts of an application through a portal. You use navigation items to access related groups of records. You use forms to create, view, and edit record information.

Portals

A portal is the home page for an application. When you sign in to the IBM TRIRIGA Application Platform, the first thing you see is your portal. The portal consists of the header region, the menu region, and the main content region, which can contain more complex graphical components.

The portal and menu regions that you see when you sign in are determined by the settings in your My Profile record. The My Profile records are described in "Security".

The menus are managed with the Navigation Builder tool, which is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

The Portal Builder tool is used to create and manage the named portals and to control the arrangement and creation of portal sections within each named portal. The Portal Builder is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Navigation items

You use navigation items to access related groups of records. You can configure a navigation item to display a collection of forms, results of a query, or hierarchical data.

Navigation items can be configured to display a default master/detail query, which is known as a manager query, that provides a standard way for the records to be displayed. If different roles need different views or ways to manipulate the same records, you can create multiple navigation items with customized queries to manage the same records in different ways.

The Navigation Builder is a tool that is provided by the IBM TRIRIGA Application Platform to create, maintain, and organize navigation items in menus or portals. The Navigation Builder gives access to navigation items that determine what kinds of records to display. Navigation items are also flexible and generic enough to

represent menus or portal quick link sections, display hierarchical data, run reports, link to other builder tools, and display UX applications.

You can specify an icon that is displayed next to a navigation item's label. The icon name is assigned after you upload the icon image to the `userfiles\images` folder of the installation directory. For the icon to display, the navigation item must be part of a Portal section whose Type is set to 'Quick Links' and whose 'Large View - Includes Icons' radio button is selected in the Portal Builder.

The Navigation Builder and navigation items are described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Forms

You use forms to create, view, and edit record information. Each form is associated with a business object. In addition, one form can display or edit data from any combination of records.

The only restriction on the records that can be manipulated by one form is the form's ability to access the records. The business object from which the records were created is not important. Different forms might access the same records to provide a different presentation of the contents of those records.

The IBM TRIRIGA Application Platform provides the Form Builder tool to define and manage forms that create, view, and edit the contents of records. The Form Builder is discussed in detail in "Form building".

Each form that is created by the Form Builder is associated with a business object. Multiple forms can be associated with the same business object. This relationship allows different forms to provide different views of the data in the same kinds of records.

The fields in a form can contain values for a record that is created from the business object with which the form is associated. A form also can display values from fields in other records. A form can even display label fields that are not connected with any record at all.

Forms use workflows to control the interaction between the user and the form. A workflow is a sequence of tasks that you can specify to be run automatically. Workflows can be used with forms to customize the behavior of the form in a number of important ways as follows:

- Workflows can copy values between records and forms.
- Workflows can be used to provide immediate feedback if a person enters a wrong value in a field.
- Workflows can dynamically change the display properties of a form. The possibilities include changing the color of a label, the font of a field, or even hiding and unhiding parts of the form.
- Workflows can evaluate the overall consistency and correctness of the data in a form before they allow the data to be saved.

Workflows can be run implicitly as a result of changing the value of a field. Workflows can be run explicitly by clicking an action.

Forms that are created by the Form Builder are organized into tabs that contain sections that contain the fields. Actions that run a workflow can be associated with

an entire form, an individual tab, or a section of the form. Also, buttons can be put anywhere in a form. A workflow can be run when someone clicks a button.

Queries and reports

The IBM TRIRIGA Application Platform has mechanisms to produce various reports or queries. These mechanisms can be divided into two categories: internal and external.

There are two internal report generating mechanisms, each producing a different report. One internal mechanism is for producing what is called a form report. A form report presents the contents of a single record. A form report might also include the contents of a record's multi-record smart sections or query sections.

Internally generated form reports are a flexible way to format the contents of a record. Form reports are described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

The other internal report generating mechanism is the Report Manager. The Report Manager generates tabular reports or graphs. The Report Manager is described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

The Report Manager generates three interesting features of reports:

- Reports that are generated by the Report Manager can be included in forms that are part of a user interface. Reports that are generated primarily for use within a user interface are called queries. The inclusion of queries in a user interface is under the control of the Form Builder tool.
- Reports that are generated by the Report Manager allow users to access the underlying records. When you click a piece of data in a report, an appropriate form pops up from which you can view or edit the contents of the source records.
- If a report or query is configured to allow it, you can edit the values that you see in a report directly in the report. With this feature, you can edit the values in any number of records all at the same time.

Editable reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Externally generated reports are produced with the IBM TRIRIGA Advanced Reporting tool. This tool allows more flexibility in formatting reports that the internally generated reports. Information about this tool can be found in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Workflows

A workflow is a specified sequence of tasks that when triggered, runs automatically. If the IBM TRIRIGA Application Platform does not already know how to manage your required tasks, you can use a workflow to specify the tasks that you need your application to run automatically.

Many concepts need to be understood to successfully create a workflow. Having a background in programming is helpful in learning to create workflows, but it is not necessary.

The major concepts that are needed to create a workflow are described in "Overview of workflows". The details that you need to know to create a working workflow are described in "Workflow building".

Platform features

In addition to the features that are needed to build and support the core functions of most business applications, the IBM TRIRIGA Application Platform provides many other features that range from exporting metadata to organizing records into a hierarchy.

- The platform has the ability to associate calendars and schedules with records. This feature can be used to track the availability of resources or to initiate actions at predetermined times. This feature is discussed in "Calendar and time-based events".
- The platform has the ability to export the metadata that makes up an application from the environment in which that application was developed to an XML file. The XML file then can be imported to other platform installations. This feature is discussed in the *IBM TRIRIGA Application Platform 3 Object Migration User Guide*.
- The platform has a variety of features to allow applications running on the platform to work with applications running outside the platform. The platform is able to work directly with a variety of technologies including Excel, database tables, and SOAP. There is an overview of these features in "Integration with external applications".
- The platform has the ability to organize records into a hierarchy. This is especially useful for organizing some types of data such as organization structures, location levels, and geographical subdivisions. The platform also has the ability to roll up totals through a hierarchy. Hierarchies are discussed in "Hierarchies".
- The platform has a variety of security related features to allow an administrator to exercise a great deal of control over who can access what kinds of data and what people can do with the data. The platform's security features are discussed in "Security".
- The platform has features to help adapt applications to work in different countries and in different languages. More information can be found in the *IBM TRIRIGA Application Platform 3 Globalization User Guide*.
- Information for application developers who are building or customizing an IBM TRIRIGA Workplace Performance Management application that runs on the IBM TRIRIGA Application Platform can be found in *Application Building for the IBM TRIRIGA Application Platform 3: Performance Framework*.

Common skills for platform tools

The IBM TRIRIGA Application Platform provides a variety of tools for building applications. There are some skills and concepts that are common to using many of these tools.

Custom menus

Some of the tools in the IBM TRIRIGA Application Platform use menus that look and work differently from most other menus. The following figure shows an example of one of these menus.
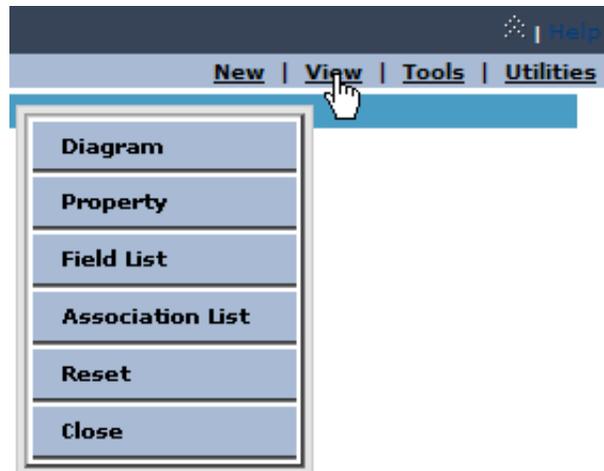
Figure 1. Custom menu

These custom menus may not appear directly under the word in the menu bar that is clicked to make the menu appear. The example shows the mouse on the word View and the menu appearing to the left of the word.

After one of these custom menus appears, you cannot make it disappear by moving the mouse. You must click one of the menu items to make it disappear.

At the bottom of these menus is an item labeled Close. The Close menu item is the menu item to click when you do not want to click a menu item. Clicking the Close menu item makes the menu disappear without doing anything else.

Create-Publish-Revise cycle

Applications that run on the IBM TRIRIGA Application Platform are made up of a variety of metadata, such as:

- business objects
- forms
- workflows

The pieces of metadata that make up an application have interdependencies. If you save a piece of metadata that is wrong or incomplete, then you break everything that uses that piece of metadata.

To avoid breaking an application, when you are editing a piece of metadata, you may decide to avoid saving the metadata until you are finished editing it. If the metadata in question is small and simple, this may not be an issue. However, if the metadata in question is large, it is a good practice to save your work frequently to minimize the consequences of any editing mishaps.

At first glance, there seems to be no way to achieve both of these goals at once. However, the IBM TRIRIGA Application Platform has a feature that allows you to achieve both goals at once.

Whenever you are editing a piece of metadata, after you have saved it the first time there will be a menu item in the Tools menu to publish the metadata. The changes you make to the metadata do not actually take effect until you publish the

metadata. No matter how many times you save metadata you are editing, the previously published version of the metadata will continue to be used until you publish the edited metadata.

When you first create a piece of metadata, it is not available for use until you publish it. You cannot publish a piece of metadata until after the first time it is saved.

The following life cycle diagram shows the typical life cycle of a piece of metadata.
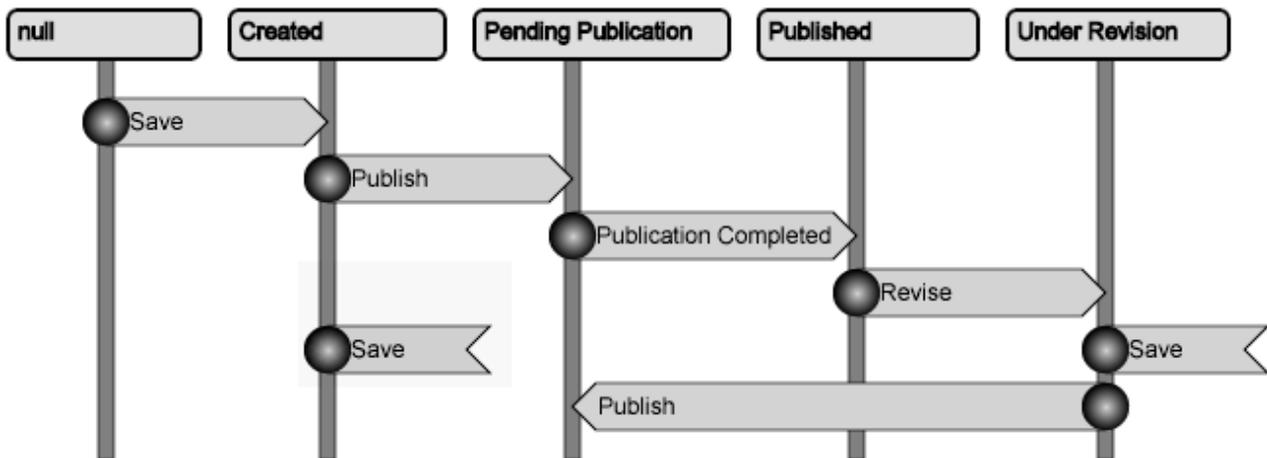
Figure 2. Metadata life cycle

Here are descriptions for the states of metadata shown in the figure.

null   A piece of metadata is in this state when it is first created. The metadata is not available for use while it is in this state. All you can do with metadata in this state is edit it and save it. Saving the metadata puts it in the Created state.

Created
>A piece of metadata is in this state after it is saved for the first time until it is published for the first time. The metadata is not available for use while it is in this state. While the metadata is in this state, you can edit it, save it, or publish it. Publishing the metadata puts it in the Pending Publication state.

Pending Publication
>A piece of metadata is in this state after a publish action has been performed on it. It stays in this state until the platform has finished its internal processes related to publishing the metadata. This process takes a few minutes for business objects. It is almost instantaneous for other kinds of metadata.

Published
>A piece of metadata is in this state after it has been published. While it is in this state, the piece of metadata is available for use. Also, while a piece of metadata is in this state, you cannot edit it.
>
>The only action you can perform on a piece of metadata in this state is **Revise**. Performing a **Revise** action on a piece of metadata puts it in the Under Revision state.

Under Revision

> A piece of metadata is in this state after a **Revise** action has been performed on it. While a piece of metadata is in this state, you can edit it. Also, while a piece of metadata is in this state, you can perform a **Save** or **Publish** action on the metadata.
>
> Performing a **Save** action on metadata in this state saves your edits but does not make any of the changes available for use. The metadata will continue to be used in exactly the same form it was in the last time it was published.
>
> Performing a **Publish** action on metadata in this state saves your edits, makes them available for use, and puts the metadata in the Pending Publication state.

Important Note: When you modify as-shipped IBM TRIRIGA objects to suit your business needs as opposed to creating your own objects, do not use a cst name prefix or follow the naming conventions. This guideline is a change from previous IBM TRIRIGA versions. The new object revisioning and object labeling features in IBM TRIRIGA Application Platform enable you to modify as-shipped objects without the need for naming conventions. Your object modifications are saved in revisions when you publish or save them.

Objects that are revisioned include modules, forms, business objects, queries/reports, UX metadata, workflows, navigation collections, navigation items, portals, portal sections, and security groups. If you use the naming conventions, the object revisioning and comparison features will not work.

For more information on object revisioning, see Object Labels and Revisions on the IBM TRIRIGA wiki.

For more information on UX metadata, see UX Articles on the IBM TRIRIGA wiki.

Panel manipulation

Many of the tools provided in the IBM TRIRIGA Application Platform environment have a user interface that is organized into panels. The following figure shows an example of this. The panels display different kinds of information.
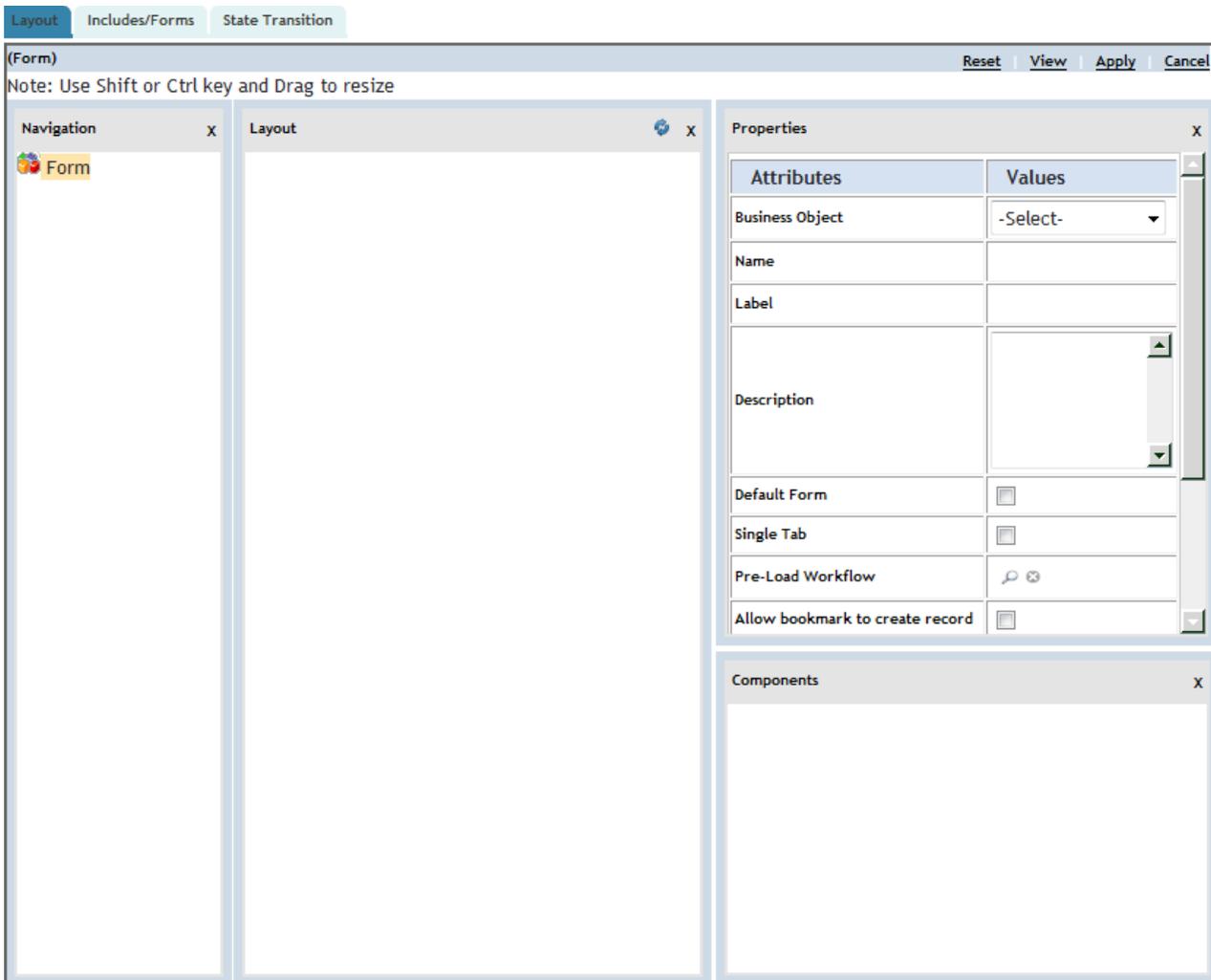
Figure 3. User interface organized into panels

There are five things that you are generally able to do with a tool's panels:
- Hide a panel so that it is not visible in the user interface.
- Force a hidden panel to be visible in the user interface.
- Move panels around in the user interface.
- Change the size of a panel.
- Restore the panels in a tool's user interface to their default configuration.

To hide a panel, click the X in panel's upper-right corner. After you click the X, the panel is no longer visible in the user interface.

To force a hidden panel to be visible again, use the View menu. Tools that have a panel-based user interface have a View menu that looks similar to a custom menu. Clicking a View menu item forces the panel with the same name as the item to become visible if hidden.

To drag a panel to a different position in the window, move the mouse to the rectangle at the top of the panel that contains the name of the panel and the X. If you hold down the left mouse button and move the mouse, you can drag the panel to where you want it in the window.

To change the size of a panel, use the symbol in the lower-right corner of the panel.

To restore the panels to their default configuration, click Reset in the menu.

Naming conventions

IBM TRIRIGA follows a set of naming conventions in its applications and in custom development that it does on behalf of its customers. IBM TRIRIGA suggests you also follow these naming conventions.

Following these conventions will reduce unexpected interactions between applications provided by IBM TRIRIGA and applications created by others. Following these conventions will help ensure that upgrades go smoothly.

Meanwhile, workflow naming standards are discussed in detail in "Workflow building".

Name Prefix:

Most objects in the as-shipped IBM TRIRIGA applications are prefixed with **tri**.

When you create your own objects, the convention in general is to assign a three-character **cst** prefix to the object name. If you add a new field or smart section to a business object or add a new form, for example, you prefix the name with **cst** such as cstMyForm.

When you create your own objects, use the guidelines in the *Naming conventions for design elements* table that follows.

Important Note: When you modify as-shipped IBM TRIRIGA objects to suit your business needs as opposed to creating your own objects, do not use a cst name prefix or follow the naming conventions. This guideline is a change from previous IBM TRIRIGA versions. The new object revisioning and object labeling features in IBM TRIRIGA Application Platform enable you to modify as-shipped objects without the need for naming conventions. Your object modifications are saved in revisions.

Objects that are revisioned include modules, forms, business objects, queries/reports, UX metadata, workflows, navigation collections, navigation items, portals, portal sections, and security groups. If you use the naming conventions, the object revisioning and comparison features will not work.

For more information on object revisioning, see Object Labels and Revisions on the IBM TRIRIGA wiki.

For more information on UX metadata, see UX Articles on the IBM TRIRIGA wiki.

Design element names:

Use the naming conventions in the table when you create your own objects. Do not using these naming conventions when you modify as-shipped IBM TRIRIGA objects.

Table 1. Naming conventions for design elements to follow when you create your own objects

Design element	Naming convention	Example name	Label convention	Label example
Module	cst + mixed case, no spaces	cstMyModule	none	n/a
Business object	cst + mixed case, no spaces[a]	cstMyBO	User readable name for the business object	Purchase Order
Form	[cst +] business object name + mixed case, no spaces[b]	cstMyForm	User readable name for the Graphical User Interface	Short Form Purchase Order
Form sections	cst + mixed case, no spaces	cstGeneral	User readable name for section	General
Smart/query section	[cst +] Business object name + mixed case, no spaces[d]	cstOrganization BillTo	User readable name for section	Bill To
Field	cst + mixed case, no spaces + field type suffix (See the table for field type suffixes)	cstAssignedFull NameTX	User readable name for field	Assigned Full Name
Navigation collection	cst + mixed case, no spaces[c]	cstPurchasing	User readable name for the navigation collection, suitable for a menu item	Purchasing
Navigation item	cst + mixed case, no spaces	cstAsset	User readable name for the navigation item, suitable for a menu item	Asset

Table 1. Naming conventions for design elements to follow when you create your own objects (continued)

Design element	Naming convention	Example name	Label convention	Label example
Portal	cst + Role – Type Sequence Types: Standard – Standard Company portal with specific report list for this role Graphic – Specific Portal for this role including graphs, queries, and report list for this role. TRIRIGA Performance Management for WPM Portals	cstTRIRIGA Contact Center Agent – Standard TRIRIGA Contact Center Agent – Graphic 2		

Table 1. Naming conventions for design elements to follow when you create your own objects (continued)

Design element	Naming convention	Example name	Label convention	Label example
Portal section	cst + Type + "-" + Business Object name – description	cstGraph – triBuilding Portfolio by Tenure (Pie Graph)	Display on Portal	Project Team Member Reports
	Types:	cstReports – Project Team Member		My Calendar
	Graph – Query section pointed to a graph	cstURL – My Calendar		
	Query – Query section pointed to a query	Examples:		
	Reports – Query List	triManager – WPM Move Planner		
	URL – External Notes:	triMetricSection – WPM Move Planner		
	CrystalGraph (portal section type = Query)	triMetricSection – WPM Move Planner		
	CrystalReport (portal section type = Query)	triPortalQuerySwitch – WPM Project Team Member		
	Manager – (portal section type = Manager) for WPM-Key Metrics	triQuickAdd – Performance Management – Move Planner		
	MetricSection – (portal section type = Query)	triShortcut – Performance Management – Move Planner		
	PortalQuerySwitch (portal section type – Query) – query with dropdown of other query/reports.			
	QuickAdd – (portal section type = Quick Add List)			
	Shortcut – (portal section type= Shortcut List)			

Design element	Naming convention	Example name	Label convention	Label example
Query, Report, Graph	cst + "-" + Form Name (or Business Object Name or Module Name if Report/Graph) - Keyword - Context[e] Or if the query references a new or customized object: Form Name (or Business Object Name or Module Name) - Keyword - Context (See the table for keywords)	cst-triPurchaseOrder - Workflow - POs for Current Year cstMyForm - BIRT - Employee Contact Details	Form Label (or Business Object Label or Module Label) - Context (What the report displays to user)	Employees - All Associated Active Employees
Security group	Customer Name + Role	IBM TRIRIGA Contact Center Agent	none	n/a
State family	cst + Context (mixed case, no spaces)	cstActionItem, cstData, cstDocument	none	n/a
State family state	cst + (Action) mixed case, no spaces[f]	cstDeleted	User readable name for the Action[g]	Deleted
State family action	cst + (Action) mixed case, no spaces	cstFinalDelete	User readable name for the Action (Transition)	Final Delete

(a) Business object names cannot be changed once the business object is published.

(b) This is the system name for the form. Each business object can have many forms.

(c) This is the system name for the navigation collection. Each business object can be in multiple navigation collections.

(d) A section name cannot exceed 30 characters, so there may not be enough space for the full business object name. Choose a name that reflects the intent of the section.

(e) No spaces between the "cst-" and the tri object name. Context should include filters to states, data filters, association filters, if editable etc. For queries tied to a $$RECORDID$$ or other platform key, make sure to include the type of context

record in the context part of the name. The ID of the report should be CUSTOM, or use your company's numbering standards for reports. SYSTEM is reserved for IBM TRIRIGA delivered reports. Keywords are described in the table for keywords.

(f) The 'null' state is the exception to this rule. A record is in a null state before it is created or permanently deleted. The system then permanently removes the records in the null state after a configured time period with the Cleanup Agent.

(g) State Transitions should not be labeled Cancel.

Field name suffixes:

The names of fields should have an uppercase suffix that corresponds to the field's type. These suffixes are shown in the following table.

Table 2. Field name suffix conventions

Field type	Suffix convention	Example field name
Action button	AB	cstLookupGeographyAB
Binary	BI	cstOrganizationBI
Boolean	BL	cstProcessFlagBL
Business object[a]	BO	cstOrganizationBO
Classification	CL	cstRollupLocHeadCL
Classification rollup	CR	cstBuildingCommonCR
Color	CO	cstFabricColorCO
Control number	CN	cstIdCN
Date	DA	cstEstimatedStartDateDA
Date and time	DT	cstActualStartDT
Duration	DU	cstMeetingDurationDU
Financial rollup	FR	cstCommitedCostsFR
Image	IM	cstPortraitIM
Label only[b]	LA	cstEnterNameHereLA
List	LI	cstYesNoLI
Note	NO	cstCommentsNO
Number	NU	cstRateNU
Password	PA	cstPasswordPA
System read only[c]	SY	triBusinessObjectIdSY
Text	TX	cstNameTX
Time	TI	cstStartTimeTI
UOM	UO	cstAreaUO
URL	UR	cstExternalInfoUR

(a) Only used for system-generated fields. Use Text locator fields that use Association strings and enable Find queries.

(b) Added with the Form Builder, not with the Data Modeler.

(c) There are IBM TRIRIGA fields for all of the system read only fields. You should not add versions of these with other prefixes. Use the **Find** action to add them if they do not exist in the business object.

Report, query, and graph keywords:

The keywords to be used in report, query, and graph names are shown in the following table.

Table 3. Report, query, and graph keyword conventions

Keyword	Description
Debug	Query to help isolate data issues
Display	Query to display data in form query sections
External	Query with linked IBM TRIRIGA Advanced Reporting document
Filter	Query to filter data for other queries via Association filters
Find	Query for **Find** action of locator field or section
FormMetric	Query for metric in a form
Formula	Query used to provide input values for extended formulas
Graph	Graph reports intended for end users
Graphics	Graphics Editor report
Patch	Query used in patch helpers
Portal	Query for portal section
Report	Report intended for end users
Reserve	Calendar and reservation-based query of data associated with record
Reserve Legacy	Calendar and legacy reservation-based query of data associated with record
Summary	Summary report intended for end users
Workflow	Query for workflow query task

Publish names:

The Publish Name defines a unique value that identifies each record for a business object. This value can be composed of one or more fields with combined values that define a record as unique.

Take care when selecting the control number as the published name as these records cannot be updated via IBM TRIRIGA Data Integrator. The control number should only be used as the publish name definition for Log or non-permanent data, such as action forms or helpers. If the application does not have a requirement for the starting value of control number, use 1000000.

If a business object has base currency fields, the Conversion Group mapping property should refer to a field with a name like triConversionGroupLI that specifies which conversion group to use for converting to base currency fields. The default value should be Default. Also, the Exchange Date property should refer to

a field with a name like triExchangeDT that specifies the effective date to use for converting to Base fields. The default value should be the current date.

When a user changes the publish name of a business object, the system requests confirmation of the change.

Chapter 2. Data modeling

The Data Modeler is a tool that is part of the IBM TRIRIGA Application Platform. The Data Modeler allows you to define the types of records that an application will use. It also allows you to define the kinds of relationships that the records may have.

All applications that run on the IBM TRIRIGA Application Platform rely on four basic facilities:

- Records that contain the application's data.
- Queries or reports that allow people to work with sets of records.
- Workflows that automate the manipulation of records.
- Forms that allow people to work with individual records.

The Data Modeler works by allowing you to edit metadata that describes records. Metadata is simply data that describes data. The metadata for records is organized in four ways.

Business objects
A business object describes the properties of a kind of record. To create a record, you use the business object that corresponds to the type of record you want to create.

Field definitions
A field contains an individual piece of data such as 4. Records contain fields that contain the individual pieces of data. A field definition defines a type of field with a specified name. A business object contains a list of field definitions that determine which fields will be in records created from the business object.

Modules
Business objects are organized into collections called modules. Each module contains one or more business objects.

Associations

In the IBM TRIRIGA Application Platform environment, there are two different types of associations. The first is at the business object (BO) level and is referred to as a BO level association. This is an association between two business objects. It serves as an association definition. The purpose of a BO level association is to define associations that applications may create or use between records that are created from the associated business objects.

The second type of association is a record level association. It defines a connection from one record to another record. If a record level association exists, an application looking at the first record can then navigate to another record it is associated with.

Associations are defined using phrases that are called association strings. Generally two different strings are used and they are categorized as either forward or reverse associations. Which string is used for the forward or reverse association depends on the perspective of the object being referred to. From the perspective of the first object, it has a forward and reverse association to the second object. However, from the perspective of the

second object, the associations are flipped. The first object's reverse association is the second object's forward association. The first object's forward association is the second object's reverse association.

If there is a BO level association between two business objects, a record level association with the same names may exist between two records created from the two business objects. The BO level association serves as a definition that allows the record level associations it defines to be created. Record level associations are usually instances of BO level associations. However, it is possible to have a record level association that is not defined by a BO level association.

Modules

A module is a collection of business objects. Every module has a name. Business objects with a similar structure or purpose are usually in the same module.

The IBM TRIRIGA Application Platform comes with a number of modules already defined. Here are some examples:

- The Geography module contains business objects used to create records that represent different levels of political or geographic entities. The business objects defined in the Geography module include City, Country, State, and World Region.
- The Document module contains business objects used to describe and manipulate documents and files. It contains business objects for describing individual documents, collections of documents, and file directories.
- The Meeting module contains business objects used to describe a meeting or meetings.

Before you can define a business object, you must decide which module it will be in. You may decide to put some business objects into modules that are already defined in the IBM TRIRIGA Application Platform and some into new modules. You must create a new module before you can put business objects in it.

Module organization

Use these rules to help you decide in which module to add a new business object:
- Business objects that contain some of the same fields should be in the same module.
- Business objects that can play the same role in the same business process should be in the same module.
- Business objects that have a similar business purpose should be in the same module.

Creating modules

Before you begin

Before you create a new module, you must decide on a name for the module. Choose a name that describes the sort of business objects that will be in the module. For example, if the business objects in a new module will be used to create records that describe materials used in training courses, it would be reasonable to choose a name like cstTrainingMaterial. Do not choose a name like X24.

Once you have decided on the name of a new module, you can create the module.

Procedure

1. To create a module, navigate to **Tools > Builder Tools > Data Modeler**. This makes the Data Modeler visible. The Data Modeler is organized into panels. The name of the panel that first appears is Object Browser. Other panels that appear as needed are named Data Modeler, Property, Field List, and Association List. You can rearrange and otherwise manage the panels using the procedures discussed in "Panel manipulation".

2. The panel labeled Object Browser contains a list of modules. Normally the Object Browser panel is collapsed so you cannot see what is in it. When you put the mouse pointer over the left-most part of the Object Browser panel where the words Object Browser are, it expands.

3. If you click a closed folder icon next to module, a list of the business objects in the module appears under the module's name. Also the closed folder icon changes to an open folder icon.

4. If you click an open folder icon next to a module, the list of business objects under the module's name disappears. Also, the open folder icon changes to a closed folder icon.

5. In the menu bar at the top of the Data Modeler, there is a menu named New. In this menu, click the New Module menu item, which causes the Property panel to appear and look like the following figure. This allows you to specify the properties needed to create a new module.
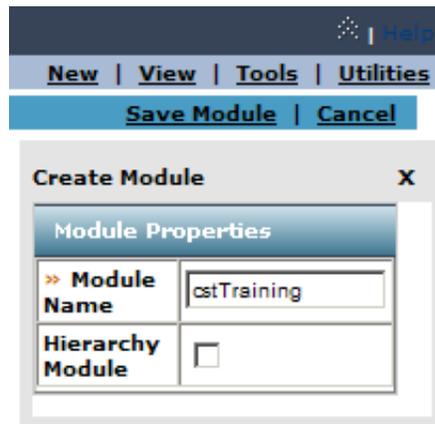
Figure 4. Properties for a new module

These are the properties to specify:

Module Name
The value of this property is the name of the module. Naming conventions are described in "Naming conventions".

Hierarchy Module
This check box should be checked if the module is going to be a hierarchy module. Hierarchy modules are described in "Hierarchies". If you are not sure what a hierarchy module is, assume all modules you create are not hierarchy modules; leave this check box unchecked.

6. Click the **Save Module** action near the top of the Data Modeler when you finish entering a new module's information. After you click the **Save Module** action, you are finished creating the module. You do not publish a module.

Modules are different from most other kinds of metadata in that they have no publish-revise cycle. The concept of a publish-revise cycle is discussed in "Create-Publish-Revise cycle".

Deleting modules

Procedure

To delete a module, first delete any business objects created in the module. Then select the **Delete Module** action.

Changing module properties

Procedure

If you want to edit the properties of an existing module, click the name of the module in the Object Browser panel. Aside from the label at the top of the Properties panel being Edit Module instead of Create Module, the experience of editing a module's properties is the same as creating a module.
Some of the other properties that display in the Module Properties panel are as follows. All of these properties are read-only.

- `Modified Date` shows the last time the module was revised.
- `Modified By` shows who last modified the module.
- `Revision` shows the current revision number of the module.
- All modules contain an identifier called an object label. `Object Label` shows the current object label for the module.

When you save a module by clicking **Save Module**, a revision is created for the module. To view a list of revisions to a module, select **Tools > Builder Tools > Data Modeler**. Select the module from the Object Browser and select the **List All Revisions** action.
You can compare two revisions of a module by selecting them and clicking **Compare Revisions**. Or you can compare two modules with different names by clicking **Compare Objects**. The results display in the Comparison Result section. To view the results in a new window, click the button next to the Compare menu. You can also export the comparison results to a text file by clicking **Export**.

Addition of business objects to a module

At this point, you can create business objects in the new module. The first business object you must create in a module is the module's base business object. The base business object should have the same name as the module.

After you have created and published a module's base business object and specified the base business object's fields, you can create other business objects in the module. When you create another business object in a module, the new business object is created with a copy of the sections, fields, association definitions, and state transitions in the module's base business object. Once a business object is created, you are free to add, remove, or modify its sections, fields, association definitions, and state transitions. Changing the newly created business object in any way has no effect on the base business object or on any other business object.

Creating business objects

Before you begin

Once a module exists, you can create business objects in the module. At a high level, these are the steps for creating a business object.

Procedure

1. Specify basic information about the business object, including its name and description. After this step, the business object exists. Its name and basic properties are established, but the IBM TRIRIGA Application Platform does not yet know what fields the records created from the business object will contain.
2. Specify associations between the business object and other business objects.
3. Specify the business object's fields.
4. Specify how records created from the business object will be uniquely identified.
5. Publish the business object.

Business object properties

To begin the creation of a business object by specifying its basic information, click the New Business Object menu item in the New menu of the Data Modeler.

There are a number of pieces of information you can enter into the properties panel. Some of the information does not have to be specified when you create a business object. It can be specified later or not at all.

When you create a business object, there are four pieces of information you must specify before the business object can exist. The first three pieces of information are the business object's name, display name, and description. The description can wait until later, but all business objects should have a description from the time they are first created.

The other piece of information you must specify at this time is whether the records created from the business object will be stand alone, embedded, or link records. The default is Stand Alone. This is controlled by radio buttons with these labels:

Stand Alone

Stand alone records contain their own data. Stand alone records can appear at the top-level organization of reports and queries (described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*). Stand alone records are the only form of record that can be managed directly from the results of a query. This is the default.

Embedded

Embedded records cannot be at the top-level organization of a query or report. Embedded records cannot be accessed from the results of a query. An embedded record can be accessed only through other records. Embedded records can exist only in a smart section or query section of another record. Smart sections are described in "Record organization".

Embedded records cannot be shared between two different smart sections. Any attempt to put the same embedded record in two different smart sections will cause a copy of the embedded record to be put in the second smart section.

Stand alone records can be referenced by any number of smart sections; embedded records can not be embedded in more than one.

Tip: Use a Stand Alone business object with a dependent association instead of an Embedded business object. IBM TRIRIGA uses Stand Alone business objects with a dependent association in standard IBM TRIRIGA applications.

Link

Link records were used to blur differences between the types of records referenced by a smart section. When a link record was created, it was linked to another record. A link record could be linked to any record created from a business object in the same module as the one the link was created from.

In the past, one of the uses of link business objects was to report on multiple business objects. Now, Report Builder queries can report on multiple business objects without using link business objects. Use Report Builder queries to report on multiple business objects.

In the past, one of the uses of link business objects was to allow different business objects to appear in multi-record smart sections. Now, query sections provide this functionality without using link business objects or multi-record smart sections. Use Query sections to display information from different business objects.

Externally Managed

This is not normally selected. The Externally Managed radio button should only be selected for Fact table business objects. Externally managed business objects are described in *Application Building for the IBM TRIRIGA Application Platform 3: Performance Framework.*

After you have specified one of the above four options, you will be able to create the business object by clicking the **Save BO** action. What follows are brief descriptions of check boxes and drop-down menus you can use to specify other properties of a business object:

Staging Table

Identifies business objects for which the IBM TRIRIGA system maintains staging tables. If the property is set to DataConnect or Generic and the business object is published, a staging table is either created or updated. If the property is set to None and the business object is published, the publish process deletes the staging table if it exists. By default the Staging Table property is set to None.

If this property is set to Generic, then the fields whose Staging Table key property has been set at the time of the initial BO publish will become the keys for the database table.

A staging table is created for a business object with the Staging Table property set to DataConnect or Generic when the business object is brought in through object migration and object migration publishes the business object.

IBM TRIRIGA DataConnect for Fact Tables uses staging tables as a mechanism for moving external data into the IBM TRIRIGA database. Read more about DataConnect for Fact Tables in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management.*

Has Calendar

> This is normally unchecked. If it is checked, records created from this business object will have a calendar associated with them.

> Calendars and calendar-based events are described in "Calendar and time-based events".

Audit Actions

> This is normally unchecked. If this is checked, a record is kept of every action as defined in the state transitions performed on records created from this business object. The record of actions is incorporated into an audit trail.

> If you are using the Request for Information (RFI) metrics and reports in IBM TRIRIGA Workplace Performance Management products, Audit Actions must be checked.

> Audit trails and other security-related issues are discussed in "Security".

Audit Access

> This is normally unchecked. If this is checked, a record is kept of every time someone views a record created from this business object.

The next three radio buttons define what auditing will be performed for data changes made to records created from this business object. Audit trails and other security-related issues are discussed in "Security".

No Audit

> No Audit is the most common choice. When selected, a record is not kept when a value change is made to a record created from this business object.

Audit Interactive Data

> When selected, a record is kept of every value change made by a user to a record created from this business object. The record of changes is incorporated into an audit trail.

Audit All Data

> When selected, a record is kept of every value change made to a record created from this business object, regardless of how the data is modified. This includes value changes made by a user, due to a formula calculation or rollup calculation, or by a workflow. However, this does not include value changes made if the record is modified by an editable query. Note that selecting this option may adversely affect the performance of the process.

Require Explanation

> This is normally unchecked. If this is checked and Audit Interactive Data is selected, when a user makes a change to a record, a popup appears requiring them to enter information about why they made the change.

Show Single Tab

> If this check box is checked, records created from this business object can be used with forms that have single tab operation. If this check box is not checked, you will not be able to use forms that only show a single tab with records created from this business object.

Approval History

> If this check box is checked, forms that work with records created from this business object can have a tab that shows the approval history of the

record. Approvals of records are under the control of Approval workflow tasks, which are discussed in "Approval task".

When you are finished setting the properties of a new business object, click the **Save BO** action to create the business object. Creating the business object does not make the business object available for use. The business object will not be available for use until you publish it. The create-publish-revise cycle is described in "Create-Publish-Revise cycle".

You will not be able to publish the new business object until you have completed two more steps:
- Specify the fields that will be in records created from the business object. This step is described in "Business object fields".
- Specify how a unique name will be determined for records created from the business object using values in the records' fields. The details of this step are specified in "Naming records".

Once you have done these things, you are ready to publish the business object. The details of publishing a business object are discussed in "Publishing business objects".

The immediate consequences of saving the properties of a business object for the first time are:
- The name of the module is included in the Object Browser panel.
- Some additional properties appear in the Property panel.

Since we are now working with an existing business object, the next step is editing the properties of an existing business object, such as adding fields.

Copying business objects

If you create a new business object by clicking the New Business Object menu item, the new business object is created with copies of the module's base business object's sections, fields, associations, and state transitions. Sometimes, it is more convenient for the new business object to be copied from a different business object.

You also can create a new business object by copying any non-base business object in a module. To do this, view the properties of the existing business object you want to copy. When the business object's properties appear, a **Copy BO** action also appears at the top of the Data Modeler. Clicking the **Copy BO** action causes a Name property to appear in the Data Modeler's Property panel. Set the value of the property to the name you want the new business object to have. Finish by clicking the **Ok** action at the top of the Property panel.

Business object properties revision

When a business object's displayed state is Published, you cannot modify the business object's properties, fields or anything else about the business object. This is explained in "Create-Publish-Revise cycle".

To be able to revise a business object, you must first change its state to Revision In Progress. To do this, first click Tools in the menu bar at the top of the Data Modeler. If the business object whose details are shown in the Diagram panel has

the displayed status Published, the Tools menu contains an item labeled **Revise BO**. Clicking the Revise BO menu item changes the displayed state of the business object to Revision In Progress.

Once the displayed state of the business object is Revision In Progress, you are able to make changes to the business object.

A revision is created for a business object every time you publish it by clicking **Publish BO**. The current revision for a business object displays at the bottom of the Business Object Properties panel.

To view a list of revisions to a business object, select **Tools > Builder Tools > Data Modeler**. Select the business object from the Object Browser and click **List All Revisions**. You can also access a business object's revision list by selecting **Tools > List All BO Revisions**. The revisions are listed in reverse chronological order in the Business Object Revisions section. You can revert to a previous revision of a business object by selecting the previous revision and clicking **Publish**.

You can compare two revisions of a business object by selecting them and clicking **Compare**. You can also compare to business objects with different names. The results display in the Comparison Result section. To view the results in a new window, click the button next to the Compare menu. You can also export the comparison results to a text file by clicking **Export**.

Business object fields

After a business object exists, you can specify a list of the fields that records created from the business object will contain. To access a business object's list of fields, the business object must be visible in the Diagram panel. The procedure for making a business object visible in the Diagram panel is described in "Viewing business object properties".

In the Data Modeler, when a business object is selected, its Field List panel becomes visible. The business object's list of fields appears in the Field List panel. The Data Modeler's Field List panel has columns showing the following properties of the fields.

Field Name
> The name of a field uniquely identifies it within the IBM TRIRIGA Application Platform environment. It is the name you see for the field within the Data Modeler, report filters, workflows, and other places a person does not see when using an application. Naming conventions are described in "Naming conventions".

Field Label
> A field's label is the name that appears next to it in a form or report. A label does not have to be unique.

Field Type
> Each field can hold only a specified kind of data. The kind of data that a field can hold is determined by the value of this property. Some of the data types are Number, Text, and Date. There is a summary of all data types in "Field data types".

You can use the Data Modeler's Field List panel to add fields to a business object, edit the properties of fields, and remove fields from business objects.

Adding fields to business objects

About this task

The procedure for adding a field varies slightly, depending on whether the IBM TRIRIGA Application Platform already has a definition for the field you want to add. If the field is already defined, add the field by searching for it. If the field is not defined, add the field by defining it.

Procedure

1. If you are not sure whether a field with a particular name is defined, you should find out by searching for the field. To search for a field, begin by looking at the Field List of the business object to which you want to add a field.

2. Click the **Find** action at the top of the Field List. This causes a Field Search window to appear in the Data Modeler's Property panel. A Field Search allows you to search for existing fields and add them to the selected business object.

3. To use a Field Search, enter the search criteria for the field to be found. The form allows you to specify the following search criteria.

   Module Name
   If you select a value for this, you restrict the search to only fields used by business objects that are part of the selected module.

   Business Object Name
   You can only select a value for this if you have specified a module. Selecting a business object restricts the search to only fields used by the selected business object.

   Name The name of a field is the unique name you see for the field within the Data Modeler, report filters, work flows and other places a person does not see when using an application. Specifying a value for Name restricts the search to fields whose name starts with the specified wildcard pattern.

   Label This is the default label that appears next to the field in a form or report. Specifying a value for Label restricts the search to fields whose label starts with the specified wildcard pattern.

   Type Each field can hold only a specified kind of data. The kind of data that a field can hold is determined by its type. Some of the data types are Number, Text, and Date. If you specify a data type other than All, only fields having the specified type will appear in the search results. There is a summary of all data types in "Field data types".

   Wildcard searches
   Sometimes you want to find fields that have a name or label that contains a word anywhere in the name or label. For example, you may want to find all field names that contain the word "start" anywhere in the name. You can do this by using a percent sign (%) in a search string. Percent signs in a search string are treated as a wildcard. The search string %start matches ActualStartDate, ProjectPlanStart and StartTime. You can also use a percent sign to find a name that has a particular beginning and end. Entering start%time matches Start Time and Start Date Time.

4. After you have finished specifying search criteria, click the **Search** action at the top of the Field Search. A list of the fields in the search results will appear in the bottom half of the Field Search.

5. If the fields you are looking for appear in the search results, you can add the fields to the business object by checking the check box to the left of the fields and then clicking the **Accept** action at the top of the Field Search. After you click the **Accept** action, the accepted fields are added to the business object are is displayed in the list of fields in the List panel.

6. If the label, default value, or most other things about the found fields does not match the properties you want, you can change them. The changes you make will affect only the business object's copy of the field. The changes will not have any affect on any other field or business object.

   - There are two things about an existing field you cannot change. You cannot change a field's name or its type. If you find a field with the right name but the wrong type, you must use a field with a different name.

   - Many business objects have a field with the same name. Even if fields in different business objects share the same name, most other things about the fields are allowed to be different. However, there is one thing that must be the same for every field with the same name: Every field that has the same name must also have the same data type.

7. If the Field Search does not find the field you are looking for, click the **Add** action at the top of the List panel.

   - The properties for a new field appear in the Property panel of the Data Modeler. The properties for a new field are about the same as the properties for an existing field, except that you can change the values of the properties labeled Name and Field Type.

   - You will see some different properties for the field if you select different types for the new field. The properties that go with a field of a particular type are described in "Field data types".

Field properties

You can edit the properties of a field by clicking its name in the Field List panel of the Data Modeler. When you click a field's name, the field's properties appear in the Data Modeler's Property panel.

The first property is the Section property. This is a read-only property. Section names are used to organize the fields that can be accessed in records created from a business object. When you are adding a field directly to a business object, it will always be part of a section named General. Other kinds of sections are discussed in "Record organization".

The value of the Field Type property determines the type of data that the field will hold. There is a summary of the possible values for Field Type in "Field data types".

Depending on the value selected for Field Type, additional properties may appear. You can find a description of the specific properties for different data types along with their descriptions in "Field data types".

If you are adding a new field, you can change the values of Field Type and Name. If you are editing an existing field, these properties are read-only and cannot be changed.

The Description and Purpose properties should contain whatever information you think will help people understand what will be stored in the field and what the field is for. If you are adding a new field, you can edit the Description and

Purpose properties in the normal way. If you are editing the properties of an existing field, you can still edit the Description and Purpose properties, but the procedure for editing them is unusual.

When you are editing the properties of a field, the Description and Purpose properties are grayed out. You cannot edit them directly. What you can do is first click the **Save Field** action at the top of the Data Modeler to ensure that any changes you have made to properties are not lost. Next, click the label of the Description property. The label of the Description property is a hyperlink. Clicking it causes a Field Description form to appear at the bottom of the Property panel under the properties.

The Field Description form contains the same information as the corresponding properties of the field. You can edit the values of the Description and Purpose properties in this form. When you click this form's **Ok** action, the form disappears and the Description and Purpose properties of the field are updated.

There is a reason for this unusual procedure for editing a field's Description and Purpose properties. These properties are intended to explain how the field should be used in business objects. They are not intended to explain how the field is used in a particular business object. Because the information is not specific to any business object, all fields with the same name have the same values for Description and Purpose. If you change a field's Description or Purpose, the change will be visible in the properties of the field in every business object that uses it.

The check box labeled Required determines how this field is treated when it appears in a form. If the Required check box is checked, a value for the field must be specified. If a value for a required field is not specified in a form, the contents of the form will not be accepted.

There are a few circumstances in which the platform supplies the initial value for a field in a new record by copying the value of a like-named field from an existing record. For example, if a field has no default value and the name of the field is the same as a name of a field in the triPeople business object, the initial value for the field comes from the triPeople business object that describes the person who is logged in. Other circumstances related to hierarchy modules are described in "Hierarchies".

This mechanism for automatically providing the initial value of a new field is called auto-populate. If you want to prevent a field from being auto-populated, check the check box labeled Do Not Auto Populate. It is a good practice to check this check box unless you specifically want a field to be auto-populated.

The check box labeled Result Column establishes a default for whether or not the field will be used in smart sections that refer to this business object. Smart sections of a business object are discussed in "Record organization". The Result Column check box also determines whether this field will be available for constructing a dynamic list. For information on dynamic lists, see "List creation". If this check box is checked, the default is that this field will be used in smart sections that refer to this business object; otherwise not.

If the Result Column check box is checked, another property appears immediately below it. This property is labeled Column Sequence. The number in this field is used to determine the default order in which fields appear in smart sections that refer to this business object.

The check box labeled Mobile Field is useful for mobile applications that run with clients that have a small display, such as a phone or PDA. The setting of this check box determines whether the field is included in a smaller view of the records created from this business object.

If the Mobile Field check box is checked, another property appears immediately under it. This field is labeled Mobile Field Seq. The number that appears in this field will determine the default order of fields in the smaller view.

If the Staging Table Field property is checked, the field is included in the business object's staging table. Changes to this property after a staging table is created are not reflected in the staging table until the business object is republished. By default the Staging Table Field property is not checked. The Staging Table Field property is supported in the following field types: Boolean, Business Object, Classification, Color, Date, Date and Time, Duration, List, Locator, Number, Password, Text, Time, UOM, Url. The Staging Table Field is checked for all Required fields of these types. Only fields in the General section are supported to be staging table fields. Fields in a smart section can be added to the staging table.

The Staging Table Key property identifies fields to be used as keys to find a record via Upsert or Update. The **Insert** action does not use the Staging Table Key property. By default, the Staging Table Key property is not checked. It can be selected only when the Staging Table Field property is selected. The Staging Table Key property is supported in the following field types: Boolean, Business Object, Classification, Color, Date, Date and Time, Duration, List, Locator, Number, Password, Text, Time, UOM, Url.

IBM TRIRIGA DataConnect for Fact Tables uses staging tables. Read more about DataConnect for Fact Tables in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management.*

When the Do Audit property is checked, changes in value for this field are stored in audit tables. If the business object's Audit Interactive Data property is selected and a field's Do Audit property is not checked, audit data for the field are not stored in audit tables.

Changing the Display Mask property in UOM List fields with any value other than Currency or in Number fields changes the decimals shown to the user but does not change the value in the database. The formatting in a Number field takes precedence over UOM formatting when displaying Number fields in query reports or forms. If a value has more digits to the right of the decimal place than shown in the Display Mask property, the platform uses Round Half Up to round the value in the display.

When the Read Only property is selected, the user cannot change the value for this field.

Requiring a field in only some forms

If you want a field to be required to have a value in every form it appears in, check its Required check box. If you want a field to be required to have a value in some but not all of the forms it appears in, do not check its Required check box. If the Required check box is checked, every form must require a value for the field. If the Required check box is not checked, forms may require the field to have a value or not.

Deleting fields

About this task

Adding fields or sections to an existing business object is usually a safe thing to do. Except for some unusual situations involving hierarchies and the auto-populate feature, adding fields to a business object will not break an application.

However, deleting fields or sections from a business object can be dangerous. If a workflow or anything else uses a field you have deleted, it will break. Only if you are certain that a field is not being used should you feel free to delete the field.

If you are not certain that a field is not being used by something, do not delete the field from the business object. Instead, leave the field in the business object but remove the field from all forms and reports that use the field. This way, users will not see that the field is still there, but workflows that expect it to be there will still see it.

When you are certain of the consequences, deleting fields from a business object is usually rather straightforward.

Procedure

1. Check the check box to the left of the field's name in the Field List panel and then click the **Delete** action at the top of the Field List panel. The platform may ask if you are sure you wish to delete the field and may respond to the **Delete** action by telling you that the delete was successful.
   - You cannot delete fields that are part of the published name. You have to remove that field from the BO Mapping before the system will let you delete that field. For information about BO Mapping, see "Naming records".
   - Another possible response to the **Delete** action is that the field cannot be deleted because it is in use somewhere. A field cannot be deleted from a business object if it is referenced by a smart section. If you do not know which business object and which smart section is using the field, you will have to find the smart section and business object that contains it. The IBM TRIRIGA Application Platform can assist you in this search.
2. If you check the check box to the left of the field name and then click the Field Finder menu item in the Data Modeler's Tools menu, a window pops up that contains a list of business objects that use the field. An example of this is shown in the following figure.
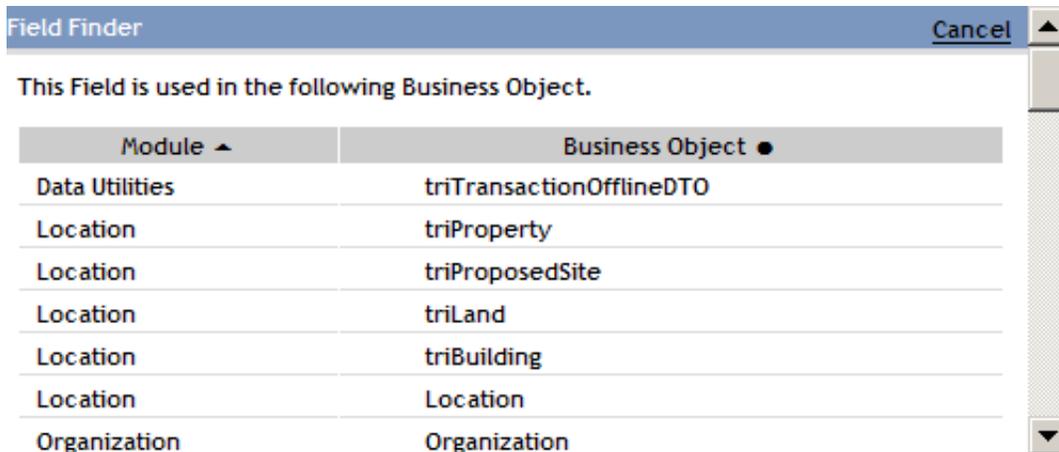
Figure 5. Field finder

3. There is another way to find where a field is used. Select the field in the Field List. Once the field's properties are displayed in the Field Properties panel, click the **Where Used** action. A window pops up showing what references or uses the business object / field. The display shows the following information: Name, Type, Module, Object, Form, Action, Additional Information.

   • Check the Where Used before you delete a field to be sure the field is not used in a report. Deleting a field that is used in a report causes an error the next time the report is run.

4. If you want to export the information in the Where Used window, click the **Export Usage** action at the top of the window. You will be offered the choice of saving or opening a csv file. For most object types, clicking the linked object will open the builder for the reference. The following lists the references found by Where Used:

   Business Object Field

   • Query (Display Columns, Order By, Group By, Sum, Filter)
   • Smart Sections
   • Locator
   • Form
   • Regular Formulas
   • Extended Formulas
   • Workflow Create Record Task (Map - Field, Source or Target)*
   • Workflow Modify Records Task (Map - Field, Source or Target)*
   • Workflow Schedule Task (Map - Field, Source or Target)*
   • Workflow Conditions - Start Task, Switch Task, Break Task*
   • Workflow Retrieve Records Task (Filter)*
   • Workflow Create Record/Modify Records/Schedule Tasks (Map - Field as part of Expression)*
   • Workflow Populate File Task*
   • Workflow Distill File Task*
   • Workflow Variable Assignment Task (Value is from an expression)*

   Form

   • Form Field Action
   • Form Section Action - Popup Form, non Popup Form

- Portal Section, Record Add
- Navigation Item (Master Detail Report, Record Add, Master Detail Default)
- Navigation Item Action
- Navigation Item Security Override
- Query
- Query Action
- Workflow Modify Metadata Task*
- Workflow Create Task*
- State Transitions (Form)
- System Add
- System Delete

Query

- Form Section Find Action
- Form Field Action (onClick, onChange)
- Form Popup Section Action
- Form Query Section (Query, Availability, Availability Legacy, Calendar, Gantt, Group By)
- Portal Section Query
- Scorecard
- Navigation Item (Master Detail Report, Record Open, Report, Master Detail Hierarchy, Scorecard)
- Navigation Item Dynamic Labels
- Navigation Item Security Override
- Query Action
- Query in a Sub Report
- Query in an Associated Query (Hierarchical Query)
- Query Association Filters
- Workflow Query Task*
- Extended Formulas

Workflow

- Query Action (Pre-Create Workflow)
- Form Field Action
- Form Field Action (Pre Form and Post Form Workflows)
- Form Section Action (Pre Form and Workflow)
- Form Section (Select and Stack Pre-Move)
- Form Section (Synchronous and Stack On-Save)
- Form (Pre-Load Workflow)
- State Transition (BO Sub Action - Workflow Select)
- BO (Pre-Create Workflow)
- Smart Section (Initialize Record)
- Call Workflow*
- Portal Section, Record Add
- Navigation Item (Record Add)
- Navigation Item Action

- Navigation Item Security Override

Note * For workflow references, the system displays the latest workflow version containing the reference. The workflows checked for references are in the following statuses: Revision In Progress, Retired, Published, and In Progress.

Naming records

Before you begin

The name of a record can come from multiple fields. A text field can contain up to 1000 characters. Even though the fields that contain a record's name may contain many more than 100 or even 1000 characters, the platform only looks at the first 100 characters of a record's name. If the first 100 characters of two records created from the same business object would be the same, the platform will complain about a duplicate name and will not create the record.

About this task

The IBM TRIRIGA Application Platform requires that for every record there is a field or combination of fields whose value uniquely identifies the record. This value or combination of values is called the record's name. The Name property is used to identify the field or combination of fields whose value(s) will uniquely identify the records. Naming conventions are described in "Naming conventions".

Procedure

1. This Name property is one of a business object's mapping properties. To access a business object's mapping properties, go to the Data Modeler's Tools menu and click its BO Mapping menu item. Clicking the BO Mapping menu item causes the business object's mapping properties to appear in the Data Modeler's Property panel.

2. To set the Name property, click its Find link. When you click the Find link, a Select Field form appears under the mapping properties in the Property panel.

3. You use this form to identify a field whose value will be used in the name of a record. It allows you to identify a field by its smart section and name. Smart sections are described in more detail in "Record organization". For the purposes of this form, fields that are not actually part of a smart section will appear to be part of a smart section named General.

4. After you select a field, click the Select Field form's **Ok** action. The Select Field form disappears and the text of the Find link is replaced with the name of the selected field. This is shown in the following figure.

Mapping Properties   X

Map Fields				
›› **Name**	Actual Amount		**Add**	**Delete**
Cost	Find	+ ▼	Find	+ ▼
	Find	+ ▼	Find	
Quantity	Find			
Image	▼			
Conversion Group	▼			
Exchange Date	▼			
Control Number	☐ Based on Prefix			
Prefix	Find		**Add**	**Delete**
Suffix	Find		**Add**	**Delete**
Start With	1			
Delimiter	▼			

Figure 6. Name setting with one Name field

5. If a record's entire name comes from a single field, you are finished specifying the name. Be sure to click **Save Mapping** action at the top of the Data Modeler to save your changes to the mapping properties.

6. In most cases, a record's name is the value of a single field. However, the name of some kinds of records must come from more than one field. To specify an additional field for the name, click the **Add** action to the right of the Name link. Clicking this action adds a drop-down list and a Find link to the Name property. This looks like the following figure.

Figure 7. Name setting with two Name fields

7. You can use this Find link to specify the additional field. If you decide it was a mistake to click the **Add** action, click the **Delete** action and the most recently added drop-down list and link are removed.

   - If a name of records created from a business object comes from values of more than one field, the values in the name are separated by a character selected in the drop-down list. The drop-down list between each of the field links allows you to choose from these characters.

   - When a person is using a form to edit a record, the record's name appears on the top of the form. If the record's name comes from multiple fields, the selected punctuation appears as part of the name.

   - The meaning of the Name property value is that the name of triPeople records will consist of the value of the Last Name field followed by a comma, followed by the value of the First Name field followed by a hyphen, followed by the value of the ID field. You can see the values of these fields and the name formed from them at the top of the form.

Control numbers

Control Number is a data type for fields that is described in "Control number fields". The value of a Control Number field can be the next number in an automatically generated sequence of numbers. A business object can contain at most one Control Number field.

The system generates a record's control number when the record transitions from the null state to any other state. The sequence of numbers generated in a business object's Control Number field is controlled by some of the business object's mapping properties. You can see these properties located to the right of and below the words Control Number.

The main part of a control number is a sequence number. If a Control Number field's Generate on Create check box is checked, then each time a new record is created from the business object, a sequence number one greater than previous number is used to form the control number. The number that is the value of the business object's mapping property labelled Start With will be used as the next sequence number.

It is common for a control number to have more parts than only its sequence number. The control number may have a prefix or suffix that is a fixed piece of text or the value of one of the record's fields. A control number may have any number or prefixes or suffixes.

To add a prefix or suffix to a control number, click the Find link to the right of the Prefix or Suffix label. When you click one of these Find links, a Type Field form appears under the mapping properties in the Data Modeler's Property panel. This Type Field form has more options on it than the Type Field form used for a business object's name.

In the Type Field form, while the Attribute Type radio button is selected, we can specify that the value of the control number prefix/suffix will come from the field specified by the selected Section Name and Field Name values.

If we want the prefix or suffix to be a fixed piece of text, rather than the value of a field, we can select the Constant Value radio button. Whatever text we supply for its Constant Text field will be the contents of the prefix or suffix being specified. When we select the Constant Value radio button, the Type Field form looks like the following figure.
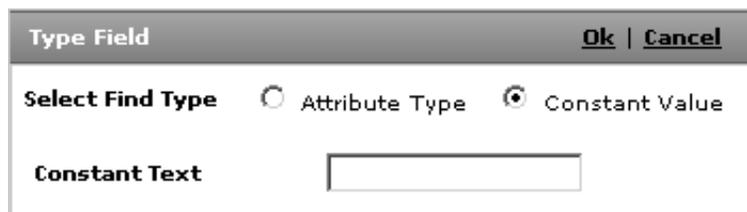
Figure 8. Fixed text for a control number suffix or prefix

In order to make sense of a control number that has prefixes or suffixes, there must be a way to tell where each part of the control number begins and ends. This is done by specifying that a particular character will be used as punctuation between

each part of the control number. The character chosen for this purpose is called the delimiter. For example, a control number that has been specified to have a hyphen as its delimiter might look like 3243-56-R.

To specify which character will be used as the delimiter, we would select a character for the value named Delimiter. The possible choices for delimiter character are: space ( ), hyphen (-), period (.), and comma (,).

One last setting that is sometimes of use is the check box labeled Based on Prefix. When this check box is not checked, the sequence number part of a new record's control number is always one greater than the sequence number for the previous record created from the same business object. When the Based on Prefix check box is checked, the sequence number for a new record is one greater than the sequence number for the previous record that had the same prefix.

For example, suppose that you want line items in a purchase order to have a control number that is prefixed by the purchase order's control number. The line item numbers in a purchase order with control number 3452 might look like 3452.1, 3452.2, 3452.3, and so on. The difficulty is that the line items for a purchase order may not be created at the same time.

It may happen that the first few line items for a purchase order are created, then another purchase order is created with its own line items, and then more line items are added to the first purchase order. The line item numbers for the first purchase order are still expected to look like 3452.1, 3452.2, 3452.3, and so on.

To arrange for this to happen, you would first make sure that the business object for purchase order line items has a smart section that contains the purchase order's control number. We could then specify the purchase order's control number as the prefix for the line item's control number. By checking the Based on Prefix check box, we would cause the sequence number portion of the line item's control numbers to start from the first value for each purchase order and continue in sequence.

Cost and quantity properties

Two other mapping properties are labeled Cost and Quantity. These properties help control the way totals are rolled up through records organized in a hierarchy. These properties and the rolling up of totals in a hierarchy are described in "Hierarchies".

The Image property is used when defining the default fields displayed in a link.

The Conversion Group and Exchange Date properties are used to help convert between different currencies. These properties and currency conversion are described in "Money".

For any of the above properties the drop-down list next to the property allows you to select the field whose value will be used for the purposes described above. Notice that the Cost and Quantity properties only allow selection of Number fields; the Image property only allows selection of Image fields; the Conversion Group property, List fields; and the Exchange Date property, Date or Date/Time fields.

Standard fields

There are three fields that the Data Modeler adds to every business object. You cannot see these three fields within the Data Modeler, but they are visible from other platform tools that work with a business object's fields.

The three fields are classification fields that can be used to associate every record with a geography, a location and an organization. The names of the fields are GeographyName, LocationName, and OrgName. The Business Object data type is discussed in "Business object fields".

In addition to the many uses these three fields have in business applications, the platform has a security-related use for the GeographyName and OrgName fields. These security-related uses are described in "Organization and geography".

Two things must be done to take advantage of these fields:
* Geographies, locations and organizations must be entered into the platform. Entering information about geographies, locations and organizations is usually done by the people who use or administer the application because they know the context in which the application will be used.
* The other thing that must be done to take advantage of the three fields is to have a way to set the value of each of the fields.

The simplest way to set the value of the fields is to take advantage of the IBM TRIRIGA Application Platform auto-populate feature, which will copy the initial values for these fields from the triPeople or other record that describes the person who initiated the creation of the record. The relevant portions of these records are described in "Authentication".

If the auto-populate feature is not sufficient to initialize the three fields to whatever values are needed, you will need to provide workflows to initialize the fields.

Publishing business objects

About this task

A new business object is not available for use until you publish it. Once you have done all the previously described things to a new business object and added a state transition diagram, you are ready to publish it. Read about state transition diagrams in "Life cycles".

If you have placed a published business object under revision, none of the changes you make to the business object affect any records until you publish the business object again.

When you publish a business object, if any records were previously created from the business object, they will be made to conform to the new definition of the business object. If you added fields to the business object, publication will add the fields to all records created from the business object. If you deleted fields from the business object, publication will delete the fields from all records created from the business object.

Publishing a business object makes the changes made to the business object available for the creation of new records. Records cannot be created from a

business object until the first time the business object is published. Once the business object is published, it becomes possible to create records from the business object.

Clicking the **Publish Business Object** action for a business object with the Staging Table property set to DataConnect triggers the following activities:

* If there are any jobs in Processing or Ready state that use the staging table and business object, the publish fails and a message is posted to the user immediately.
* Because Publish is handled by an agent and is not an immediate action, the Publish agent checks the job state again in case a new job has started or the state of an existing job has changed in the interim. If there are any jobs in Processing or Ready state that use the staging table and business object, the publish fails, a notification is posted to the user, and details about why the publish failed are written to the server.log.

For more information about publishing business objects with the Staging Table field, see "Business object properties" and *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

It is always best to publish business objects on a quiet system. This is because runtime components using the business object during the publish process could experience unpredictable behavior since there are points in time where the database definition and the business object metadata are not in sync.

After you publish a business object, you can make changes to the business object without being concerned that a partially completed set of changes will be reflected in a newly created record. The changes you make to a business object are not reflected in new records until after you publish the business object again.

You can publish individual business objects by using the Procedure below.

You can also publish multiple business objects at the same time by selecting **Utilities** > **BO Bulk Publish** in the Data Modeler menu. When the Business Object Publisher loads, business objects that are not published and are not pending publication are highlighted. By default, the embedded objects are displayed. If you click **Show in Manager**, you see the unpublished business objects. Verify the business objects that you want to publish are selected, then click **Publish BOs**. The status changes to Pending Publication. The publish of these business objects might take some time to complete. When the page refreshes, the object ID is displayed for each BO that was not published.

The Procedure shows how to publish individual business objects.

Procedure

1. Make sure that the details of the business object are visible in the Data Modeler.
2. Click **Tools** > **Publish BO**.
   * While a business object's mapping properties are visible, if the business object is under revision, a **Publish** action will be at the top of the Data Modeler. Clicking this action is another way to publish a business object.
   * The publication of the business object does not happen immediately. Publication of business objects happens in the background.

3. You can check whether publication of a business object has finished in two ways:
   - Click **View** in the menu and click **Diagram**. The business object's details appear in the Diagram panel. If the state of the business object is Pending Publication, the publication is still underway. If the state of the business object is Published, the publication has finished.
   - When the publication business object has finished being published, a notification is placed in the notification section of your portal. When you see the notification, you know that publication is finished. The subject of the notification will be one of the following, depending on the outcome of the publish:
     - Publication of [BO Name] completed successfully.
     - Publication of [BO Name] completed with warning(s).
     - Publication of [BO Name] completed with error(s).
     - Publication of [BO Name] completed with warning(s) and error(s).

     In the event that the notification subject reported warnings and/or errors, the body of the notification contains more detail about those warnings and/or errors. The Message ID at the end of each message corresponds to a detailed stack trace in the server.log. More information about the server.log can be found in *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

4. When an existing business object is republished, the system drops database columns that are no longer used.

Deleting business objects

About this task

Deleting a Business Object with the Staging Table property set to DataConnect triggers the following activities:
- If there is a job in Processing state that uses the staging table and Business Object, the delete fails and a message is posted to the user immediately.
- If there is a job in Waiting state, the Business Object and staging table are deleted and the state of the job is changed to Obsolete.

Procedure

1. First, make sure that the details of the business object are visible in the Data Modeler.
2. Click the Delete BO menu item in the Tools menu.

Associations

Most records are of limited use all by themselves. Only when they are associated with other records is their full usefulness realized. For example, by itself, a record that describes a training course has some usefulness. Associating it with other records that represent such things as course materials needed for the course, scheduled sections of the course, and prerequisites for the course makes the record that describes the training course much more useful.

An association is a connection between records. An association between two records allows the IBM TRIRIGA Application Platform to navigate from one of the

records to the other. Each end of the association has a name that we use to identify the association from that end. There can be any number of records on each end of an association.

There are a number of ways that an association can be created between two records. Here are a few of the ways:

- A person can use the Association tab of a form to explicitly associate two records.
- A person can put a record in a smart section of another record. This implicitly creates an association between the records. Smart sections are described in "Record organization".
- A person can put a record in a locator field, business object field, or classification field of another record. This implicitly creates an association between the records. Locator fields are described in "Locator fields". Business object fields are described in "Business object fields". Classification fields are described in "Classification fields".
- A workflow is a sequence of automated tasks for the platform to perform automatically. A workflow can create an association between records. Workflows are discussed in "Overview of workflows".

A person cannot explicitly create an association between records until it is defined in the IBM TRIRIGA Application Platform environment.

Association definition

In general, the platform allows the creation of an association between records only if there is an association definition that allows it. Exceptions to this rule are discussed later.

Association definitions are connected to two business objects. Association definitions also have an association name connected with each end of the association definition. These are often referred to as the forward association and reverse association strings.

If you create an association definition named Belongs To from the Course Section business object to the Course business object and an association named Has in the other direction, you will then be able to create corresponding associations between Course Section records and Course records.

There are two steps to creating an association definition:

1. Ensure that the list of association types includes the names you need for the association.
2. Tell the platform to create the association definitions.

To ensure that the names needed to describe the associations are in the list of association types, navigate to the List Manager by clicking Tools > Administration > Lists.

To see the list of association types in the List Manager, set Manage By to Name and then select the radio button next to Association Types. The Association Types list is large, so you may notice a delay while it loads.

Review the list of association types to see if it contains names suitable for describing the associations. Some of the more common association types, such as

Has, Belongs To, Uses, and Is Used By, are already in the list. If you want to use names that are not in the list, you can handle such names in one of two ways:

- If a name with a similar meaning is already in the list, you can choose to use it. Using a name that is already in the list may result in more consistent names for associations.
- You can add names to the list. The details of how to add new names to a list are explained in "List management".

Once all the names you want to use for describing associations are in the list, you are ready to define the associations. You can define an association with either of the following tools:

- Data Modeler
- Association Manager

With either tool, you define the association in the Associate Business Object panel. Regardless of which tool you use to access the Associate Business Object panel, the data to be entered is the same.

Exceptions to defining associations

There are some exceptions to the general rule of defining an association between records. For example, exceptions include Associate Record tasks in workflows described in "Associate records task"; temporary associations for the section actions of forms described in "Section actions"; field and button actions described in "Field and button actions"; and Data Integrator for single-direction record-level associations described in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*. Unless the association only is used temporarily for custom logic, it is best practice to explicitly define the association.

Associate business object panel

The Associate Business Object panel defines an association between two business objects. The properties for the new association are shown in the following figure.

Figure 9. Associate business object

The details of the association are defined by specifying its properties.

- For the Module property, select the module in the drop-down list that contains the business object that will be at one end of the association definition. This defaults to the module that contains the selected business object.

- For the Business Object property, select the business object in the module named by the Module property that will be at one end of the association. This defaults to the selected business object.

- For the Association property, select the name that will be on the same side of the association definition as the business object named by the Business Object property.

- For the Associate Module property, select the module that contains the business object that will be on the other end of the association definition.

- For the Associate Business Object property, select the business object in the module named by the Association Module property that will be on the other end of the association definition.

- For the Reverse Association property, select the name that will be on the other side of the association definition.

- If the Dependent Flag property is checked, it means that the existence of records on the other end of the association depends on the existence of the record on this side of the association. This is a one-way, asymmetric relationship.

Attention: Do not check the Dependent Flag check box unless you thoroughly understand the following explanation. Otherwise, you may find many records being deleted that should not be.

Suppose that the Dependent Flag property for an association definition is checked and that the association definition has been used to associate a record created from the business object specified by the Business Object to other records. If the record is deleted, the dependent records it is associated with are also deleted.

Similarly, if the record is copied, then the copy of the record becomes associated with copies of the dependent records, if the association or corresponding record is identified to be copied. If records on the other side of the association are deleted or copied, it has no effect on the record on this side of the association.

For example, you may have Line Items which are dependent on a Purchase Order. You would create an association definition named Has from the Purchase Order business object to the Line Item business object and named Belongs To in the other direction. This association would have the Dependent Flag checked. If you defined the reverse association named Belongs To from the Line Item business object to the Purchase Order business object and named Has in the other direction. This association would NOT have the Dependent Flag checked. With this set up, the line items are dependent on the purchase order, but the purchase order is NOT dependent on the line items.

If you need a workflow run when a record is deleted due to the deletion of a record on the other end of a dependent association, arrange for the workflow to be launched by a De-Associate system event. There is an overview of workflows in "Overview of workflows". De-Associate system events are discussed in "System events that trigger workflows".

The Project Containment Disabled check box is visible only when the Dependent Flag check box is selected. When checked, the Project Containment Disabled check box disables forcing a child record into the same project as its parent. The RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties controls whether or not child records are forced into the same project as their parent record. When the Project Containment Disabled flag is off (the default), child records are forced to the project of their parent record. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

There are a couple of scenarios where the association definition may not cover a child. They are as follows:

- The RECORD_PROJECT_CONTAINMENT property is set to Y. A dependent section exists and the association definition backing the dependent section is not a dependent association. When a row is added to the dependent section, the child record is forced into the same project as the parent record. If this is not the desired behavior, change the association definition to be dependent and use the Project Containment Disabled flag to control the behavior.
- The RECORD_PROJECT_CONTAINMENT property is set to Y. The user or a workflow makes one record a hierarchical child of another, creating an Is Parent Of association between the two records. The parent record is not the root of the hierarchy. When an Is Parent Of association is made, the child record will be forced into the same project as the parent record. If this is not the desired behavior, change the association definition to be dependent and use the Project Containment Disabled flag to control the behavior.

The Cascade Read-Only check box is visible only when the Dependent Flag is selected. When selected, it propagates the read-only state of a parent record to any dependent child records. This creates a dynamic read-only condition for the dependent child records. The actual state of the child records does not change.

This sets the child records to read-only regardless of the reason that the parent is read-only. If the parent is read-only due to security, record state, or any other reason, the child becomes read-only.

When a record is not already read-only, the platform looks for a dependent association with the Cascade Read-Only property selected. If the platform finds more than one direct parent on which the current record is dependent, it puts a warning in the log noting that this scenario is not supported and the platform does not cascade any read-only state. If the platform finds exactly one record, it looks at the record at the other end of the association to see if it is read-only. If no records are found or the parent is not read-only, the platform continues to look recursively for a parent with the Cascade Read-Only property checked; this continues for a maximum of five levels deep.

The Cascade Read-Only feature only considers dependent associations. It does not consider any other way that a child record may be dependent on a parent record.

The Cascade Read-Only property is disabled by default.

An example of the use of the Cascade Read-Only property is in real estate contracts. In the as-delivered software, when a user creates a real estate contract with payment schedules, the payment schedules have dependent associations to the real estate contract with the Cascade Read-Only property selected. When the user activates the real estate contract, the payment schedules become read-only, which prevents a user from changing the payment terms. Using this technique is much more efficient for the application developer and for system performance than creating workflows and hidden state transition actions to accomplish the same task.

Adding associations from the Data Modeler

About this task

You can use the Data Modeler to define associations between business objects. Before you define an association, you may want to see if it is already defined.

Procedure

1. Select a business object in the Data Modeler's Object Browser panel
2. Click the word View in the menu bar. The view menu appears.
3. Click the Diagram menu item in the view menu. The business object's details display in the Diagram panel.
4. When you select a business object, its details are displayed in the Diagram panel, including all defined associations connected to the selected business object.
   - A simple box with the name of the selected business object appears at the center of the diagram. For each association that is defined between the selected business object and other business objects, there is a line and a box.
   - Each of the other boxes in the diagram corresponds to an association. The name of the other business object connected to the association definition appears in the middle of the box. The association name on the side of the selected business object appears on the top of the box. The name on the other side association appears on the bottom of the box.
   - The middle part of boxes that correspond to an association are either dark blue or light blue. A dark blue middle means that there is a smart section based on the association. A light blue middle means the association is a simple association with no corresponding smart section.

5. If you want to change the selected business object to a business object named in the middle of one of the boxes, click the name of the business object in the diagram.

6. You can get more information about the association definitions connected to a business object by looking at the Data Modeler's Association List panel. The Association List panel does not automatically appear. To see it, click the Association List menu item in the View menu.

   • All association definitions shown in the Association List panel have the selected business object on one end of the association. The information displayed for each association definition is:

     – The name of the module that contains the other business object.

     – The name of the association on the side of the selected business object.

     – The name of the other business object.

   • For example, an association in the Association List for the triBuilding business object is Geography Cost Index (triCostEstimate - Has Cost Index - triGeographyCostIndex). This defines an association named Has Cost Index that is also connected to the triGeographyCostIndex business object in the triCostEstimate module.

   • The Association List panel initially displays all association definitions connected to the selected business object. This list is divided into two parts by a section bar labeled Other Associated Objects. The associations that appear above the bar are those being used to support a structural feature of the data model. Association definitions not being used to support a structural feature appear below the bar.

   • If the size of the Association List panel is so large as to be inconvenient, you may want it to not display the non-structural associations. There is an arrowhead on the right side of the Other Associated Objects section bar. When the entire list is visible in the Association List panel, the arrowhead points up. If you click it while it is pointing up, the association definitions in the Other Associated Objects section become hidden and the arrowhead points down. If you click the arrowhead again, the associations in the Other Associated Objects section are visible again and the arrowhead points up.

7. If you did not find the association you were looking for in the Diagram panel or the Association List panel, you need to define the association.

8. To add an association, select the New Association menu item in the Data Modeler's New menu. This causes the Association Properties panel for a new association to appear. Also, a **Save Association** action appears at the top of the Data Modeler. For detailed information about entering the properties defining an association, see the description in "Associate business object panel".

9. After you have finished specifying the properties of a new association definition, create the association definition by clicking the **Save Association** action at the top of the Data Modeler.

10. As you create association definitions, they appear in the Diagram panel and also in the Association List panel in the Other Associated Objects section.

11. To edit the properties of an existing association definition, click its name at the top of one of the boxes in the Diagram panel or on the association definition in the Association List panel. This causes the association's properties to appear and the **Save Association** action to appear at the top of the Data Modeler.

12. When you are finished editing the association's properties, be sure to click the **Save Association** action to save the edits.

Adding associations from the Association Manager

About this task

The Association Manager defines associations and enforces the relationships between the existing business objects in the system. The association created is bi-directional, as the association refers to the section of the selected business object. For example, assume that a relationship must be described from business object A's perspective (where the association is created for that particular business object) and business object B's perspective (where the section of that particular association is related). In such a case, the reverse relationship from business object B's perspective also must be defined.

Occasionally a business object may need to have more than one kind of relationship with another business object. For example, an Employee (person) may belong to two office locations. One office is used full time (primary), while the other office is a remote location (shared) and is used infrequently.

The list of association types is maintained in the Association Type list through the List Manager. The Association Type list is a system list. You can add values to the list as required to describe the relationship between business objects.

Procedure

1. To access the Association Manager, navigate to Tools > Builder Tools > Association Manager. The Association Manager panel appears. You will see the list of existing modules in the Module panel on the left.
2. When you click the radio button to the left of any module, all associations for that module display in the Associate Business Object List panel on the right.
   - The IBM TRIRIGA applications are delivered with a set of pre-defined associations. When you require a new association, you can create the association using the Association Manager.
   - **Attention:** Modifying or deleting an association that is used in conjunction with other data or processes can produce unwanted or unexpected results.
3. To add an association from the Association Manager, scroll the Module panel on the left until you find the module to which the association is to be added. Click the radio button to the left of the module name and click the **Add** action on the Associate Business Object List section bar. The Associate Business Object panel appears. This panel and its fields are described in "Associate business object panel".

Associations with base business objects

A base business object is the first business object created in a module, and should have the same name as its module. If the business objects specified by an association definition are not base business objects, the association definition allows associations to be created only between records created from the specified business objects.

If one side of an association definition specifies a base business object, the association definition will allow records created from any business object in the specified module to be associated.

Record organization

The fields of a record are organized into sections. Fields that are directly contained by a record are always in a section named *General*. Every record and business object contains exactly one section named *General*.

In addition to containing a *General* section, records also can contain *smart sections*. Each smart section in a record has a name. A smart section can reference fields in one or multiple records, depending on the smart section's properties.

A smart section contains fields. A smart section contains a set of fields for each record it references. Each field in a smart section corresponds to a field in a referenced record. The fields in a smart section contain either a reference or a copy of the value in the corresponding field of the referenced record depending on the smart section properties.

Each smart section in a record is based on an association between the record that contains the smart section and another record. If a smart section references a record, these two statements are always true:

- The record that contains the smart section has an association with the record that the smart section references.
- Something was done, either by a user or a workflow, to cause the smart section to refer to the record. If the record that contains the smart section does not already have an association with a record that is added to a smart section, the association is created automatically.

There are two fundamental varieties of smart sections:

Single-record smart sections

> A single-record smart section can reference one record or none. The presentation of fields in a single-record smart section is usually very form-like, with a label next to each field and fields arranged in different rows and columns.

> If you add a record to a single-record smart section that already references a record, the new record replaces the old one.

> A dependent single-record smart section can contain a locator field. The attributes of the locator field are defined in the corresponding locator field in the associated business object. Locator fields are discussed in "Locator fields".

Multiple-record smart sections

> A multiple-record smart section in a record can reference any number of records. The fields in a multiple-record smart section are usually presented in a table-like way, with the fields arranged in columns and the fields for each record in a different row. A smart section also can be organized vertically, with the fields in rows and the records in columns.

> When you add a record to a multiple-record smart section, the smart section simply references one more record than it did before.

> IBM TRIRIGA has replaced most pre-configured multiple-record smart sections with Query sections (described in "Query sections"), which are more flexible and powerful. Before you use a multi-record smart section, review "Pros and cons of using smart sections".

A business object contains descriptions of the smart sections that will be in records created from the business object.

The Data Modeler treats both kinds of smart sections the same way. You tell the Data Modeler which kind you want by specifying a value for one of the smart section's properties.

To create either kind of smart section, first click the association that the smart section will be based on in the *Diagram* panel or the *Association List* panel. If the association does not exist, follow the procedure described in "Association definition" to define the association.

When you are looking at the association's properties, you will see an action at the top of the association properties labeled *Create Section*. Click this action to create a smart section based on the association. If the **Create Section** action is grayed out, it is usually because the association has not been created yet. Click the **Save Association** action at the top of the Data Modeler to solve this problem.

When you click the *Create Section* action, properties of the new smart section appear in the Data Modeler's *Property* panel. The exact set of properties that a smart section has depends on whether the smart section will contain stand alone, embedded, or link records. The properties that are displayed if the smart section will contain stand alone or link records are shown in the following figure.

Figure 10. Stand alone smart section properties

Begin by specifying the value of the *Section Name* property. This is a name used internally to identify the section. The name must be unique within the business object.

The next thing to specify is the value of the *Section Label* property. This is the default text for a user interface to display to identify the smart section. It is usually displayed in a bar above the section. It does not need to be unique.

The value of the *Associated Business Object* should already have the correct value, which is the association that the smart section will be based on.

The value of the *Temporary Association* property is used by the workflow named by the *Workflow to Initialize Record* property. If you are not going to set a value for the *Workflow to Initialize Record* property, you do not need to set a value for the *Temporary Association* property.

The value of the *Temporary Association* property is the name of an association the workflow can use to navigate from a record that contains the smart section to the record that was just added to the smart section.

Temporary associations differ from permanent associations in a few ways:

- Temporary associations are temporary. They are created for the benefit of a workflow. After the workflow is finished, the association is removed.
- Temporary associations are identified solely by their name, not by the type of record on the other end.
- Temporary associations are implicitly created by the platform in response to certain situations. They cannot be explicitly created by a person or workflow.
- No association definition is required for a temporary association.

The next thing to specify about a smart section is whether it will be a single-record smart section or a multiple-record smart section. If you select the *Associate One Record* radio button, the smart section will be a single-record smart section that can reference either zero or one records. If you select the *Associate Multiple Record* radio button, the smart section will be a multiple-record smart section that can reference any number of records.

If you selected the *Associate One Record* radio button, you will not be able to specify the *Vertical Section* property. The check box for the *Vertical Section* property will be grayed out and forced to the unchecked state.

If you selected the *Associate Multiple Record* radio button, you will be able to specify the *Vertical Section* property. A user interface usually presents a multiple-record smart section with each row of the presentation corresponding to a different record and each column to a different field. If you check the *Vertical Section* check box, you are requesting the user interface to switch the usual presentation around, so that each column of the presentation corresponds to a different record and each row to a different field.

The next property you can specify is the *Dependent* property. Smart sections with the Dependent flag checked are known as dependent smart sections and can be affected by project security.

Attention: Do not check the *Dependent* check box unless you thoroughly understand the following explanation. Otherwise, you may find many records being deleted that should not be.

If the *Dependent* check box is checked, the continued existence of a record referenced by the smart section depends on its continuing to be referenced by the smart section. If a record is removed from a smart section that has the *Dependent* check box checked, the record is deleted. Notice that if you checked the Dependent flag for the association, the Dependent check box in Section Properties is checked and cannot be changed.

The next property for you to specify determines if values in the smart section's fields may be modified by a user. If you select the *Reference Only* radio button, values in the fields of the smart section may not be modified by a user.

If you select the *Reference With Modify* radio button, the smart section's fields may be modified. The values of fields in the underlying record are used to initialize the values of the smart section's fields. After a record becomes referenced by a smart section that has its *Reference With Modify* radio button selected, there is no further connection between their values. Changes to the smart section's fields do not affect the underlying record's fields and changes to the underlying record's fields do not affect the smart section's fields.

The *Live Link* check box is visible only if the *Reference Only* radio button is selected. If the *Live Link* check box is checked, the corresponding fields in the underlying record are always directly referenced to the associated record in the smart section. If the *Live Link* check box is not checked, modified values for corresponding fields in the underlying record are copied to corresponding fields in the smart section and changes to values of the underlying associated record have no effect on the values in the smart section's fields.

If you want a workflow to run when a record is added to the smart section, set the value of the *Workflow to Initialize Record* property to the name of the synchronous workflow that is to run when a record is added to the smart section. Synchronous workflows are discussed in "Synchronous versus asynchronous workflows".

When you add a record to a smart section, there may be some special initialization you want done. For example, you may want to set some fields in the newly added record based on the fields in the same record that contains the smart section.

The workflow you specify in the *Workflow to Initialize Record* property can be responsible for any initialization or bookkeeping that is needed when a record is added to the smart section. The workflow is launched from the record that contains the smart section. It is able to navigate from the record that contains the smart section to the newly added record by traversing a temporary association with the name you specified as the value of the *Temporary Association* property.

To specify a workflow, click the search icon. Clicking the search icon causes a list of synchronous workflows associated with the selected business object to appear under the smart section's properties. You can then select from the list the workflow that you want to associate with the smart section.

If a workflow is already associated with a smart section and you want to clear the value of the *Workflow to Initialize Record* property so that no workflow is associated with the smart section, click the X icon.

In the Data Modeler you can associate an existing workflow with a section of a record. You cannot use the Data Modeler to create a workflow. You must use the Workflow Builder to create a workflow. You cannot put tasks in a workflow that refer to a section until the section is defined. To define a section with an associated workflow, first create the workflow without any tasks. Next, create the section in the business object, specifying the name of the workflow. After the section is created, add tasks to the workflow.

The Used by DataConnect property indicates that this smart section can receive values from DataConnect. Select the Values Required property to indicate values brought in from DataConnect are required. Values Required does not affect the smart section unless Used by DataConnect is checked and the DataConnect action is either Insert or Upsert/Insert. You can read more about DataConnect in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

After you finish specifying values for the smart section's properties, you are ready to specify which fields of the underlying records will be part of the smart section. You can see the fields available for use in the smart section by clicking the down-pointing arrowhead on the Field List section bar under the smart section's properties.

After you click the down-pointing arrowhead, it becomes an up-pointing arrowhead and the list of available fields becomes visible. An example of what this

looks like is shown in the following figure.

Figure 11. Stand alone smart section with field list showing

The Field List displays the fields in the business object that will be used to create the records that the smart section will contain. There are four columns in the Field List:

Select

Checking a check box in the Select column means that the field will be in the smart section.

Sum

>   Number fields have a check box in the Sum column of the Field List. Use this check box to indicate that you want a sum computed for the field.

>   You can arrange for records to have a field that contains the sum of a field used in a records section.

>   Under the section's properties there is a Field List. There is an example of a Field List in the previous figure. The Field List allows you to specify which fields will be in the section by checking a check box in the Select column.

>   For Number fields, you also can specify that the business object should have a field that contains the sum of the values of a field in the section. You specify this by checking the corresponding check box in the Sum column.

>   If you request a sum field, the field is added to the business object's General section. The name of the sum field is composed from the name of the section and the name of the field to be summed. For example, if there is a field named One in a section named Roster, the sum field for One will be named Roster One Total.

DataConnect Key

>   The Field List contains the DataConnect Key column when the Used by DataConnect property is selected. Only data types that do not require a lookup to find their value can be DataConnect keys. DataConnect keys can be Number, Text, or Control Number data types.

>   DataConnect keys do not need to be in the smart section; they can be in the referenced business object. Selecting the check box in the DataConnect Key column identifies a field as a DataConnect lookup key. Once published, the staging table will have fields for these columns. DataConnect will use the values from these staging table fields to look up entries to add to the smart section. This allows DataConnect to populate the smart section from inbound data.

>   You can learn more about DataConnect in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

Field

>   Contains the label defined for the field in the business object that is the source of the field.

When you first look at the Field List for a smart section, some check boxes in the Select column may already be checked and greyed out. The fields initially included in a smart section are the fields in the underlying business object that are defined with their *Result Column* check box checked. The *Result Column* check box is discussed in "Field properties".

If a field is UOM-managed, be sure to select the corresponding UOM source field. This ensures that the correct UOM is available in the smart section for display to the user.

Smart section sequence numbers

When you are looking at the properties of an existing multiple-smart section, it has a property labeled *Table Sequence*. Single-smart sections do not have this property.

Smart sections that have not been saved for the first time do not have this property. The following figure shows the properties of an existing multiple-record smart section.

Section Properties

Section Details

» Section Name	triBuildingClassDataAttributes
» Section Label	Building Class Data Attributes
Associated Business Object	triIntermediate - Has Building Class Data Value - triData/
Temporary Association	---Select a Temporary Association---
Associate One Record	○
Associate Multiple Record	◉
Vertical Section	☐
Table Sequence	▾
Dependent	✓
Add Through Find Query	☐
Workflow to Initialize Record	🔍 ✕
Used by DataConnect	☐
Values Required	☐

Field List

Figure 12. Existing multiple-record smart section

The purpose of the *Table Sequence* field is to solve a problem related to the order that information from records appears in reports. Reports are described in the *IBM TRIRIGA Application Platform Reporting 3 User Guide*.

You can generate reports that show the contents of a multiple-record smart section ordered by the contents of any field in the record. If the records in question have fields named *Date* or *Title*, you can have the report order the information by its date or title.

Sometimes you want the contents of a multiple-record smart section to be listed in an order that does not correspond to the data it contains but rather to the order that records were added to a smart section. The *Table Sequence* field gives you a simple way of facilitating reports that show records in the order they were added to a multiple-record smart section.

The basic strategy for facilitating reports that show records in the order they were added to a multiple-record smart section is to add a *Number* field to the records for this purpose. The values put in this field will reflect the order in which their record is added to a smart section. We call these values *sequence numbers*.

After you have added a field to a business object to contain sequence numbers for the records created from it, you need a way to put the sequence numbers in the field. This is what the *Table Sequence* property is for.

The *Table Sequence* property contains a list of the *Number* fields in the multiple-record smart section. If you select the name of a *Number* field as the value of the *Table Sequence* property, the IBM TRIRIGA Application Platform will put sequence numbers in the specified field.

Embedded smart sections

Smart sections that contain embedded records or have the dependent flag checked have a different set of properties than other smart sections. The following figure shows the properties of a smart section that contains embedded records.

Figure 13. Embedded smart section properties

Embedded records exist only in the smart section they are created in. Because the existence of embedded records always depends on the existence of the record that contains them, the *Dependent* check box is always checked and grayed out.

For similar reasons, the properties for a smart section that will contain embedded records has no *Reference Only* radio button, no *Live Link* check box and no *Reference With Modify* radio button.

There is a property that is unique to smart sections that will contain embedded records. The property is displayed as a check box labeled *Add Through Find Query.* Checking this check box designates that the user is going to Find one type of record and a workflow is going to take the information of the records found to create the records that will be shown in the section. This option is available only in conjunction with the Associate Multiple Record option.

Add Through Find Query is a rather elaborate mechanism for adding records to a smart section that contains embedded records. There is a simpler mechanism available for adding records to smart sections that contain other kinds of records. If a smart section contains records that are not embedded, you can add records to it by finding existing records and selecting them.

Existing embedded records cannot be added to a smart section in this straightforward way because embedded records exist only in the smart section they were created in. If the information you want to add to a smart section as an embedded record is in another record, you can add a copy of the record to the smart section if it is the right kind of record. If the information is in a different kind of record, you will need to create the embedded record and set its values appropriately.

Because of the many different ways that may be needed to go from a found record to adding an embedded record to a smart section, the IBM TRIRIGA Application Platform does not provide any built-in logic to handle it. Instead, the platform requires you to provide the logic in a workflow that handles all the details.

The workflow you specify in the *Workflow to Initialize Record* property has more responsibilities when the *Add Through Find Query* check box is checked. Normally, the workflow is responsible only for initialization or bookkeeping. If the *Add Through Find Query* check box is checked, the workflow is also responsible for creating the embedded record and adding it to the smart section.

Another difference in the responsibilities of the workflow if the *Add Through Find Query* check box is checked is the way it is expected to use the temporary association specified by the *Temporary Association* property. Instead of using the temporary association to navigate to a new record, the workflow is expected to use the temporary association to navigate to the records found that contain the data that will be the basis for the contents of the embedded records the workflow is expected to create.

Pros and cons of using smart sections

Because multiple-record smart sections and query sections (described in "Query sections") are similar in a number of ways, we thought it would be helpful to summarize their differences to help you decide which is most appropriate for a need.

If you need to be able to explicitly add or remove records from a smart section to keep an historical record of what the values of the referenced record were at the time it was added, use a multiple-record smart section. Otherwise, if merely having an association with the containing record is sufficient for inclusion in the section, a query section is usually the better choice.

A multiple-record smart section contains exactly those records that have been put in the section. A query section contains the results of a query.

Users and workflows may explicitly add records to a multiple-record smart section or remove them. There is no explicit add or remove operation for a query section. Records appear in a query section because of the values they contain or the associations they have.

When the content of a multiple-record smart section is to be based on the values or associations of the records it contains, using a query section is the better choice. It results in a simpler and faster application.

If you want to use a field from an associated business object, in a business object's mapping properties (*Name*, *Cost*, *Quantity*, *Prefix*, or *Suffix*), you must include it in a single-record smart section that references the associated business object.

Locator fields are similar to fields in smart sections because a locator field's value can be populated with the value of a field from another record. If your situation calls for a need for a live link to the referenced record's most up-to-date value, you may want to use a smart section. Locator fields are beneficial if a record containing a locator field needs to retain the original value of the referenced record as it was when the locator field was populated with the referenced record. Locator fields are discussed in "Locator fields".

Note sections

Note sections are a feature that was supported in older versions of the platform. This feature is now obsolete. Business objects created in an older version of the platform that use note sections will continue to work as always, but your note section is turned into a note field.

To allow notes of arbitrary length with formatting to be included in a record, use a field that has the data type *Note*. The *Note* data type is described in "Note fields".

Discussion thread sections

A Discussion Thread section allows people to write sequences of comments and attach them to a record. The comments can have comments, so the whole discussion is arranged in a hierarchy.

To create a discussion section, first define an association between the business object that will contain the discussion section and the *Discussion Thread* business object in the *System* module. It is not important what names you choose for the association.

As soon as you specify that the other end of the association will be connected to the *Discussion Thread* business object, the properties for the discussion appear below the association's properties. This is shown in the following figure.

Figure 14. Properties for an Association and a Discussion Thread section

These are the properties to specify when creating a Discussion Thread section:

Section Name
> This is the name that the platform uses internally to identify the section. This name must be unique among the business object's sections.

Section Label
> This is the default label text that will appear in a user interface to identify the section.

After you have finished specifying the properties of the association and the Discussion Thread section, click the *Save Association* action. This creates both the association and the Discussion Thread section.

Links

As mentioned previously, a link was a special kind of record that blurred distinctions between types of records created from different business objects. Link records existed only in a smart section.

In the past, one of the uses of link business objects was to report on multiple business objects. Now, Report Builder queries can report on multiple business objects without using link business objects. Use Report Builder queries to report on multiple business objects.

In the past, another of the uses of link business objects was to allow different business objects to appear in multi-record smart sections. Now, query sections provide this functionality without using link business objects or multi-record smart sections. Use query sections to display information from different business objects.

We will explain how a link works by first explaining how the other kinds of records, stand alone and embedded, organize their fields.

Consider the following diagram. It shows the organization of two kinds of records that are not links. One is a *Textbook* record. The other is a *Sample Application* record. The diagram shows three fields of each record.

Some fields in the records have the same name. They both have a field named *Name* and another field named *Description*. Each record also has fields with names that the other record does not have. The *Textbook* record has a field named *Publisher* that the *SampleApplication* record does not. The *Sample Application* record has a *Revision Date* field that the *Textbook* record does not.
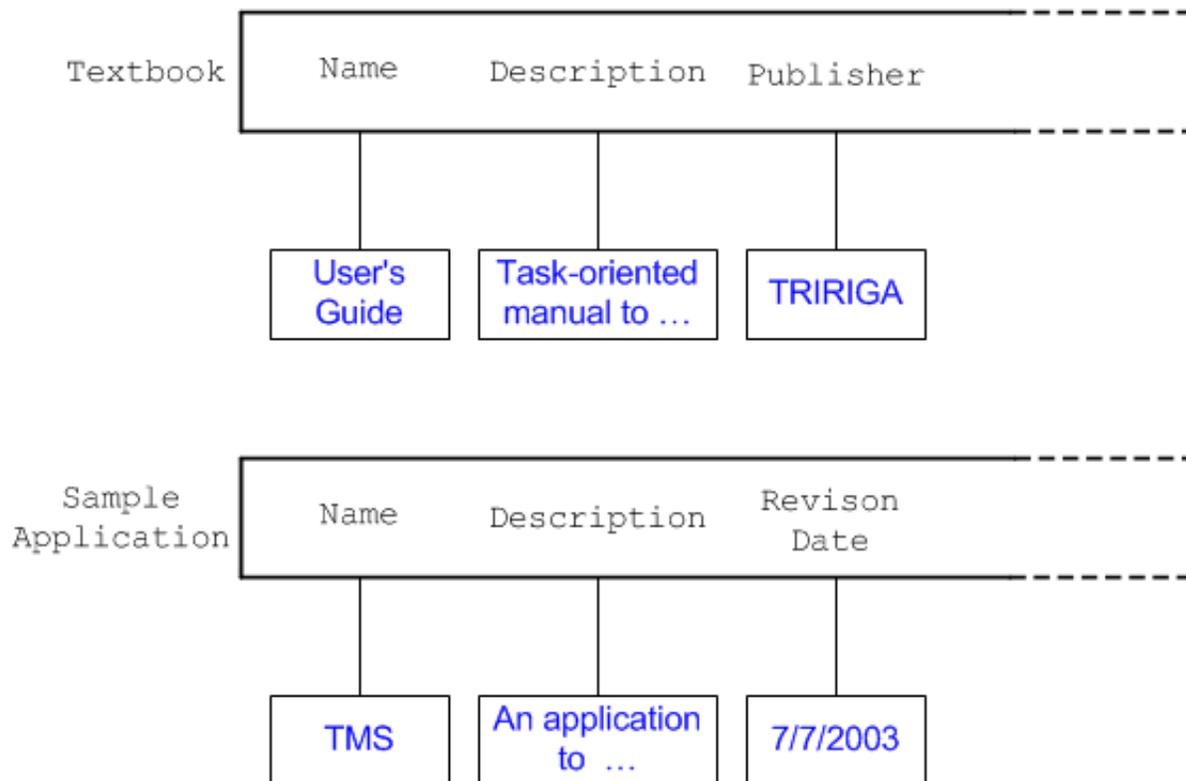
Figure 15. Records that are not links

Clearly, these two records are different. Because they have different fields, they must be treated differently. Looking for a *Publisher* field in a *Sample Application* record does not work.

Link records are created when a person or workflow adds a record to a section based on an association with a link business object. Instead of causing the section to reference the record being added, the section creates a new link record that is linked to the record being added to the section. The section references the link record rather than the record it was asked to reference.

If the name of a field in a link record is different from any field name in the record it is linked to, the field's initial value is determined normally. However, fields in the link record that have the same name as a field in the linked record have their initial value copied from the field with the same name in the linked record. This

organization is shown in the following diagram.
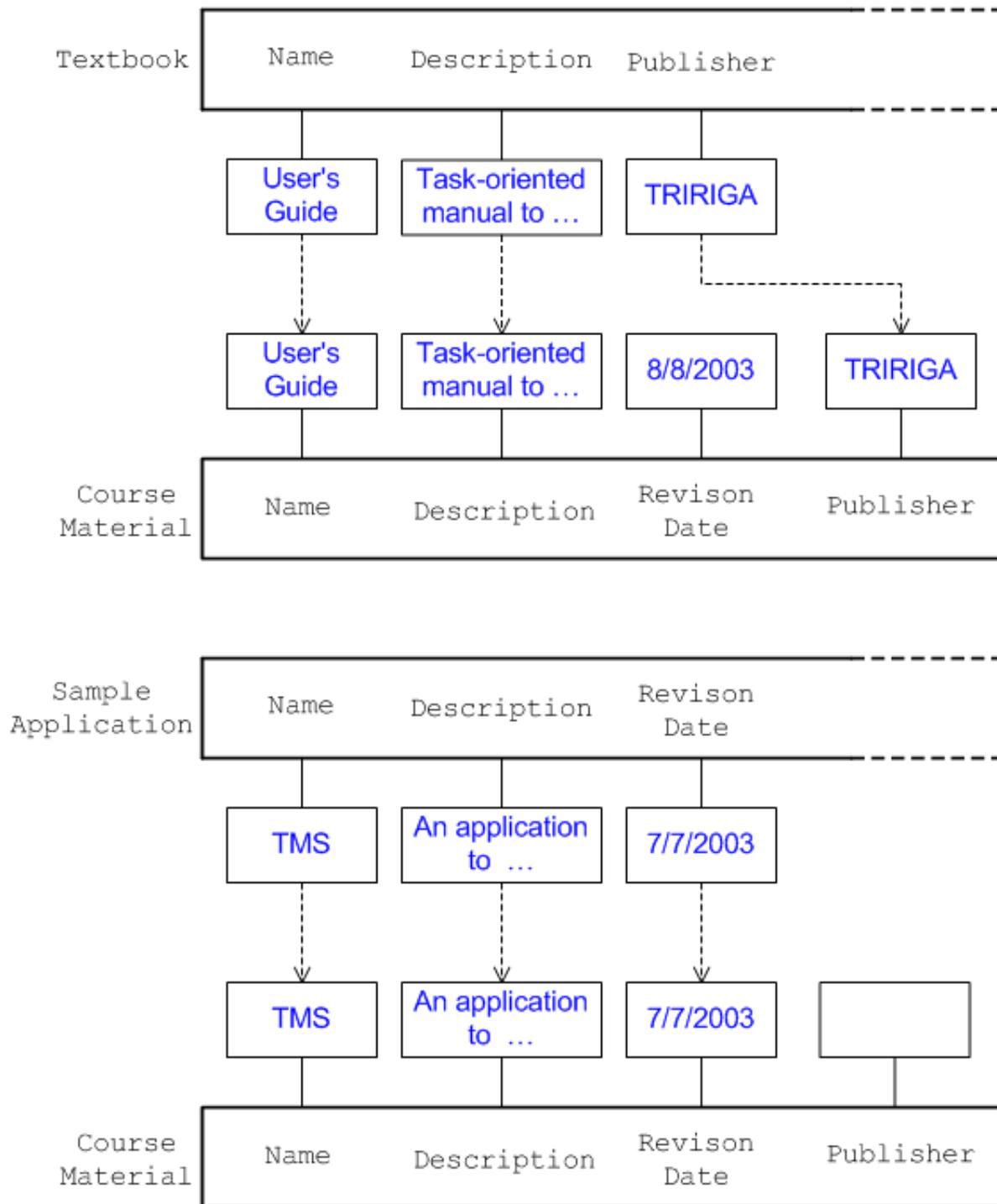
Figure 16. Links

The second diagram shows the same records we saw in the first diagram, each linked to a link record created from a business object named *Course Material*. The *Course Material* records are links that have fields with some of the same names as the *Textbook* and *Sample Application* records.

The second diagram shows how the fields of each *Course Material* link record are given initial values from the record that it is linked to. The *Name*, *Description* and *Publisher* fields of the *Course Material* record linked to the *Textbook* record have their initial value copied from like-named fields in the linked record. The *Revision Date* field does not get its initial value from the linked record because the record it is linked to does not have a field named *Revision Date*. Similarly, the *Name*, *Description* and *Revision Date* fields of the *Course Material* link that is linked to the *Sample Application* record have their initial values copied from the like-named fields in the linked record. The *Publisher* field does not get its initial value copied from the linked record because the record it is linked to does not have a field named *Publisher*.

If a section references records through links rather than directly, all records in the section appear to have the same number of fields with the same names and data types.

After the fields of link records are set to their initial values, what happens to the values after that is determined by the properties of the section that contains the link records. Whether or not a user interface should allow the values of the link record's fields to be modified and whether or not changes in the linked record will be copied to the fields of the link record are things that are controlled in the usual way by the section's properties: the *Reference Only* radio button, the *Live Link* check box and the *Reference With Modify* radio button.

Whether or not a user can click a hyperlink to the linked record in a single-record smart section is set by the smart section's Show Embedded Link check box.

There is a restriction on the kinds of records that a link can be linked to. A link record can be linked only to a record that was created from a business object in the same module as the business object used to create the link record.

Chapter 3. Field data types

A property of every field is its data type. A field's data type determines the type of data that the field can contain. A good understanding of data types is necessary for defining fields and for most things that applications do with fields.

The following table summarizes the data types available in the IBM TRIRIGA Application Platform. The description of each data type includes a discussion of what is involved in specifying a field with that type. Most fields have the usual properties described in "Field properties". Only additional properties or properties that are somehow different will be described.

Table 4. Summary of data types

Data type	Description	Suffix
Binary	Used to contain an arbitrary sequence of bytes that the platform cannot directly manipulate.	BI
Boolean	Used to contain a value that is either true or false.	BL
Business object	Used to associate a Location, Organization, or Geography with a record.	BO
Classification	Used to contain a value selected from a defined hierarchy of values.	CL
Classification rollup	Used for rolling up sums by classification.	CR
Color	Used to contain a color.	CO
Control number	Used to generate and contain unique ID numbers.	CN
Date	Used to contain a date.	DA
Date and time	Used to contain a time and date combination.	DT
Duration	Used to contain the length of a time interval.	DU
Financial rollup	Used to contain totals from other fields involved in financial transactions.	FR
Image	Used to contain an image.	IM
Label only	Used to specify a label with no corresponding field.	LA
List	Used to select a value from a list of values.	LI
Note	Used to contain arbitrary length formatted text.	NO
Number	Used to contain numbers.	NU

Table 4. Summary of data types (continued)

Data type	Description	Suffix
Password	Used to contain a text value that can be modified but not displayed in a user interface.	PA
System read only	Used to access information about a record and the business object used to create it.	SY
Text	Used to contain text values	TX
Time	Used to contain a time of day	TI
UOM	Used to contain a Unit of Measure	UO
Url	Used to contain a URL	UR

Binary fields

A Binary field contains a value of arbitrary length that the IBM TRIRIGA Application Platform cannot directly manipulate. For example, a Binary field can be used to contain an Excel spreadsheet. The exception to this is when the binary field stores an Excel spreadsheet and used in conjunction with the Distill File and Populate File workflow tasks discussed in "Offline Excel spreadsheets".

A Binary field has all of the usual field properties, which are described in "Field properties".

Tip: Binary field names should end with BI for easy identification later, e.g., cstDataTemplateBI.

Boolean fields

A Boolean field contains a value that is either true or false. There are two most common uses for Boolean fields:

- Boolean fields may be used to indicate whether the real world entity represented by a record is classified in a certain way or not. For example, a Boolean field may be used to indicate if a task is a milestone, if a lease includes a particular service, or if a country uses the Euro as its currency.

For classifications having more than two possibilities, it is usually simpler and less confusing to use the List or Classification data types.

- Boolean fields may be used as an alternative to state transitions to note things about the current state of something. Boolean fields are sometimes more useful for this purpose if you need to reason about combinations of states. For example, boolean variables may be used to note if the data in a record is up to date, if a document has been reviewed, or if a piece of equipment needs to be picked up rather than being delivered. Having three Boolean fields to represent these things would avoid having at least eight states in the state transition family to represent the eight possible combinations of possibilities.

Boolean fields display as checked/unchecked check boxes in non-editable queries and reports.

Boolean fields can be selected as runtime filters or as static filters in metric queries.

To filter a Boolean field, use the values TRUE and/or FALSE.

The form uses all of the usual field properties, which are described in "Field properties".

To minimize user confusion, do not check the *Required* check box for a Boolean field. *Required* is useful for data types that allow a distinction to be made between fields that contain a value and those that do not. A Boolean field always contains a true or false value. Since there is always a value in a Boolean field, making it required is meaningless.

Tip: Boolean field names should end with BL for easy identification later, e.g., cstDefaultBL.

Business object fields

A Business Object field can contain a *Geography*, *Location* or *Organization* record, depending on how the field is specified. You may also see this type of field referred to as a Smart Object field. A Smart Object is an instance of a Business Object. If a triPeople Business Object is the definition of a person, a triPeople Smart Object is a specific instance of a person in the system.

A Business Object field uses all the usual field properties, which are described in "Field properties". There are some additional properties.

Module

The value of this property can be set to *Geography*, *Location*, or *Organization*. This setting determines which of these three types of records this field will refer to.

Full Path

Geography, *Location*, and *Organization* records are managed as hierarchies by the IBM TRIRIGA Application Platform's Geography Manager, Location Manager, and Organization Manager, respectively. If the *Full Path* check box is not checked, the Geography, Location, or Organization that is the value of this field is displayed using its own name. This might look like: Atlanta

If the *Full Path* check box is checked, the Geography, Location, or Organization that is the value of this field is displayed using its name and every other name between it and the root of its hierarchy. This might look like: \World\North America\USA\Georgia\Atlanta

Hierarchies are discussed in greater detail in "Hierarchies".

Association String

It is not necessary to define any associations to use a Business Object field, but it is a good thing to do. If any associations between this field's business object and a business object in the module specified by the *Module* property are defined, the association names will appear in this property's drop-down list. If you choose the name of an association in the drop-down list to be this property's value, the platform creates the named association

between a record that contains this field and the record that this field references as well as the reverse association between these two records.

Tip: Business Object field names should end with BO for easy identification later, e.g., cstOrganizationBO.

Classification fields

Classifications are represented as records presented in a hierarchical fashion. If the type of a field is Classification, a record in the *Classification* hierarchy can be chosen as the value of the field. The general purpose way of handling hierarchies is described in "Hierarchies".

You can allow a Classification field's value to be chosen from any record in the *Classification* hierarchy. A Classification field can be defined that may contain any record shown in this hierarchy.

Usually, a field is restricted to some part of the hierarchy. For example, a field might be restricted to the classification *Discipline*, which includes everything under *Discipline*, such as *Civil*, *Electrical*, and *Electrical Lighting*.

Classifications like the ones under *Discipline* are useful simply for their name and what their name implies to people. Because classifications are represented by records, they can have additional fields that contain data used to determine the outcome of computations. For example, a classification that indicates what type of equipment something is may have a field whose value is the phone number to call if the equipment needs repair.

The List data type is sometimes a good alternative to Classification. There is a comparison of the two types in "Lists versus classifications".

Tip: Classification field names should end with CL for easy identification later, e.g., cstFunctionalRoleCL.

Classification field properties

A Classification field has all of the usual field properties, which are described in "Field properties". The form also has additional properties.

Full Path

> If the *Full Path* check box is not checked, the classification that is the value of this field will be displayed in the user interface using just the classification's name. For example, if the classification labeled Owned Asset is selected it will be displayed as: *Owned Asset*

> If the *Full Path* check box is checked, the names of all classifications in the hierarchy above the selection are included in what is displayed. For example, if the classification labeled Owned Asset is selected and the value of the *Root Classification* property is *Asset Ownership* it will be displayed as: *\Classifications\Asset Ownership\Owned Asset*

Root Classification

> The value of this property determines from which portion of the Classification hierarchy values for this field can be selected. When you click the search icon to the right of this property, a Classification Selector

pops up. When you select a classification as the value for *Root Classification* , it means the field's value may be that classification or classifications under it.

Keep in mind that the classification you specify here allows you to restrict what values are appropriate for this field. For example, if you specify Classifications here, any classification could be the value of this field. If you specify Discipline, then you can restrict the values of this field to more specific classifications, such as Civil and Electrical.

Association String

It is not necessary to define an association to use a *Classification* field. However, if any associations are defined between this field's business object and the business object specified by the *Root Classification* property or a business object under it in the classification hierarchy, then the names of the associations appear in this property's drop-down list. If you choose the name of an association in the drop-down list to be the value of this property, then the platform will create the named association between a record that contains this field and the record that this field references as well as the reverse association between these two records.

If this property has no value, the IBM TRIRIGA Application Platform still creates an association between a record that contains this field and the record that this field references. The name of the association will be *Classified by* followed by the name of the Classification record that is the value of the *Root Classification* property. For example, if this property has no value and the value of the *Root Classification* property is *Brands*, then the name of the association will be *Classified by Brands*. Since there is no reverse association defined, the reverse association will not be created.

Creating and editing classification fields

When creating a new Classification field, you may find that some of the classifications you want as possible values for the field are not available in the Classification Selector panel. If all that is missing are classifications you want to see in an existing part of the Classification hierarchy, you can simply add the classifications using the Classification Manager.

Procedure

1. Sometimes you want to add a whole new branch to the Classification hierarchy. These are the conceptual steps to adding a new branch to the Classification hierarchy:

   a. Define a new business object in the *Classification* module that will be used to create the records that will form the new branch of the Classification hierarchy.

   b. Create an *Is Parent Of* association from the new business object to itself so that you can create a sub-hierarchy using just records created using the new business object.

   c. Publish the new business object.

   d. All Classification hierarchies have a common parent that is created from a Classification business object. You must create an *Is Parent Of* association from the Classification business object to the new business object to allow records created from the new business object to be in the hierarchy under the common parent.

   e. Create a form to create records created from the new business object and edit them.

f. Modify the form used to edit Classification records to allow the creation of the records created from the new business object under Classification records.

g. Define the new branch of the hierarchy using the new business object.

2. We explain the details of these conceptual steps using an example. Suppose we want to add the classifications shown in the following list to the Classification hierarchy. These classifications will allow one kind of record to represent all kinds of public transportation modes. The application will use a Classification field in a record to know what kind of public transportation mode it is.

- Public Transportation Mode
  - Airline
    - Regional
    - Major
  - Bus
  - Ferry
  - Train
    - Commuter
    - Metro/Underground/Elevated
    - Tram

3. To accomplish the first conceptual step, we will create a business object in the Classification module named *cstPublicTransportationMode*. The steps in doing this will be:

a. Navigate to the Data Modeler.

b. Select the Classification module in the Data Modeler's *Object Browser* panel. Click the *New Business Object* menu item in the *New* menu of the Data Modeler. This causes properties for a new business object to appear in the *Property* panel.

c. Set the *Name* of the new business object to *cstPublicTransportationMode* and the *Display Name* to *Public Transportation Mode*. For the Description enter Classifications for different public transportation modes.

d. Specify that the new business object is *Stand Alone*. All business objects used for hierarchies should be *Stand Alone*.

e. Click the Data Modeler's *Save BO* action. This creates the *cstPublicTransportationMode* business object and causes it to appear in the *Data Modeler*.

4. The next conceptual step is creating an *Is Parent Of* association.

a. Click the *New Association* menu item in the *New* menu. This causes the properties for a new association to appear in the *Property* panel.

b. Set the value of the *Association* property to *Is Parent Of*. Set the value of the *Associate Business Object* property to *cstPublicTransportationMode*.

c. Click the Data Modeler's *Save Association* action.

5. The next conceptual step is to publish the new business object.

a. Click the *BO Mapping* menu item of the Data Modeler's *Tools* menu. The Mapping Properties of the *cstPublicTransportationMode* business object appear in the *Property* panel.

b. Click the *Find* hyperlink in the *Name* property. This causes a Type Field form to appear under the Mapping Properties.

c. Make sure that the fields contain the correct values. The Section Name field should contain *RecordInformation*. The Field Name field should contain *Name*.

d. Click the Type Field form's *Ok* action and the Type Field form disappears.

e. Click the *Save Mapping* action near the top of the Data Modeler.

f. While the Mapping Properties are visible, a *Publish BO* action is visible near the top of the Data Modeler. Click the *Publish BO* action to publish the *cstPublicTransportationMode* business object.

6. The next conceptual step is to create an *Is Parent Of* association from the Classification business object to the new business object to allow you to create new cstPublicTransportation Mode records under the Classification hierarchy.

a. Click the *Classification* business object in the Data Modeler's Object Browser panel. Details of the *Classification* business object appear.

b. Click the *New Association* menu item in the *New* menu. This causes the properties for a new association to appear in the *Property* panel.

c. Set the value of the *Association* property to *Is Parent Of*. Set the value of the *Associate Business Object* property to *cstPublicTransportationMode*.

d. Click the Data Modeler's *Save Association* action.

e. Click the *Publish BO* menu item of the *Tools* menu.

7. The next conceptual step is to create a form for the new business object.

a. Navigate to the Form Builder by clicking the *Form Builder* menu item in the *Tools > Builder Tools* menu.

b. Select the radio button for the *Classification* module in the left side of the Form Builder. A list of forms associated with the *Classification* module appears in the right side of the Form Builder.

c. Click the *New* action at the top of the Form Builder. A Form Wizard window pops up to allow you to define a new form. Another option is to Copy an existing Classification form. Starting from a copy could save you some time.

d. Initially, the Form Wizard's properties tab shows the properties of the form itself. Set the value of the *Business Object* property to *cstPublicTransportationMode*. Check the check box for the *Default Form* property.

e. Click the *Apply* action at the top of the Form Wizard's *Layout* tab. The name shown in the *Navigation* tab will change from *Form*, which is just a placeholder name, to *cstPublicTransportationMode*.

f. Click the *Add Tab* action at the top of the Form Wizard's *Layout* tab. This causes the properties for a new tab to appear in the *Properties* panel of the Form Wizard's *Layout* tab.

g. Fill in the properties of the new tab as shown in the following table.

Table 5. Properties for General tab

Attributes	Values
Name	General
Label	General
Tab Information	Required
Instruction	Describe the mode of public transportation
Visible	Selected

h. Click the *Apply* action at the top of the Form Wizard's *Layout* tab. The General tab now appears in the Navigator panel.

i. Click the *Add Section* action at the top of the Form Wizard's *Layout* tab. This causes the properties for a new tab to appear in the *Properties* panel of the Form Wizard's *Layout* tab.

j. Fill in the properties of the new section as shown in the following table.

Table 6. Properties for General section

Attributes	Values
Type	Form
Name	General
Label	General
Visible	Selected
Expand Section	Selected
Show Title Bar	Selected
Start Row	1
Row Span	1
Start Column	1
Col Span	12

k. Click the *Apply* action at the top of the Form Wizard's *Layout* tab. A list of the fields in the *cstPublicTransportationMode* business object appears in the *Components* panel.

l. Check the check box for the *triNameTX* and *triDescriptionTX* fields. Click the *Add* action at the top of the *Components* panel. This adds two fields to the *General* section that are based on the *triNameTX* and *triDescriptionTX* fields in the business object. The *General* section and its two fields appear in the *Layout* panel of the Form Wizard's *Layout* tab.

m. Navigate to the *Includes/Forms* tab of the Form Wizard.

n. Click the *Add* action on the *Includes* section bar. A Select Item(s) panel pops up to allow the selection of records that may be created under a *cstPublicTransportationMode* record.

o. Select the check box next to *cstPublicTransportationMode* and click the *Ok* action at the top of the Select Item(s) panel. This causes the Select Item(s) panel to disappear and the *cstPublicTransportationMode* business object to appear in the *Includes* section.

p. Navigate back to the *Layout* tab of the Form Wizard. Click the *Publish* action at the top of the *Layout* tab. This causes the form to be published. When the publication of the form is done, the *Publish* action at the top of the *Layout* tab is replaced with a *Revise* action.

q. Click the *Cancel* action to close the Form Wizard window.

8. The next conceptual step is to enable the *Classification* form to create records under Classification using the new business object.

a. Navigate to the Form Builder.

b. Select the radio button for the *Classification* module in the left side of the Form Builder. A list of forms associated with the *Classification* module appears in the right side of the Form Builder.

c. Click the hyperlink on the right side of the Form Builder for the *Classification* form. Look for triClassification. A Form Wizard window pops up for viewing and editing the properties of the *Classification* form.

d. Click the *Revise* action at the top of the Form Wizard's *Layout* tab. Wait for the *Revise* action to be replaced with a *Publish* action.

e. Navigate to the Form Wizard's *Includes/Forms* tab.

f. Click the *Add* action at the top of the *Includes* section. A Select Item(s) panel appears.

g. In the Select Item(s) panel, select the check box for the *cstPublicTransportationMode* business object. Click the *Ok* action at the top of the Select Item(s) panel. The Select Item(s) panel disappears. The *cstPublicTransportationMode* business object is now in the list in the *Includes* section.

h. Navigate back to the *Layout* tab of the Form Wizard. Click the *Publish* action at the top of the *Layout* tab. When the publication of the form is done, the *Publish* action at the top of the *Layout* tab is replaced with a *Revise* action.

i. Click the *Cancel* action to close the Form Wizard window.

9. The final conceptual step is to add records created from the new *cstPublicTransportationMode* business object to the Classification hierarchy.

a. Navigate to Tools > Classifications.

b. The *Hierarchy* initially displays the top level of the *Classification* hierarchy.

c. In the *Hierarchy* panel, click the word *Classifications* at the top. Click the *New* action on bar below the *Hierarchy* section bar. A list of possible types of records you can create under the *Classifications* record appears.

d. Click the list item labeled *Public Transportation Mode*. The system displays the *Public Transportation Mode* form, which allows you to create the new *Public Transportation Mode* record.

e. Fill in the fields of the form: In the *Name* field enter *Public Transportation Mode*. In the Description field enter Root classification for Public Transportation Mode.

f. Click the form's *Create* action. The record is created. The form disappears. The *Hierarchy* now contains *Public Transportation Mode*.

g. Click *Public Transportation Mode* in the *Hierarchy* and click the *New* action. A list appears of the kinds of classifications we can add under *Public Transportation Mode*. The only item in this list is *Public Transportation Mode*.

h. Click *Public Transportation Mode*. A form appears so we can edit the new *Public Transportation Mode* classification.

i. Enter *Airline* in the *Name* field and fill in the Description field. Click the *Create* action. The form disappears. The Classification tree redisplays. Notice that the icon next to *Public Transportation Mode* now looks like a plus sign instead of a dot. This is because there is now something under it. Clicking the icon (not the word) alternately opens and closes the classification, showing and hiding what is under *Public Transportation Mode*.

j. Repeat the last few steps to enter the rest of the Classification hierarchy.

Add fields to classification records

In the preceding example, we added records to the Classification hierarchy. As we added them, the records served only to identify different public transportation modes.

We could add additional fields to *cstPublicTransportationMode* records to control things the application will do based on the public transportation mode.

For example, we could include fields to contain text that is to be printed on a report described by a *cstPublicTransportationMode* record. There also could be fields to control the color of different parts of the text.

Other kinds of records are created in whatever is currently the active project. Classification records always are created in the Company Level project, so that they are accessible to everyone. Projects are described in "Projects".

Classification rollup fields

Classification Rollup fields are used to intelligently roll up totals of data in a hierarchy, based on classification fields.

The properties and use of a Classification Rollup field are fully described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

Tip: Classification Rollup field names should end with CR for easy identification later, e.g., cstBuildingCommonCR.

Color fields

Color fields contain a color. Color values can be used to control the color of some things that are displayed.

A Color field has all the usual field properties, which are described in "Field properties", except for *Read Only*. It is not possible to specify that a Color field is read only.

The *Default Value* property is used to specify the initial value for the field. To specify a default value click the search icon. A palette of colors pops up. After you have selected a color by clicking it, the color palette disappears and a rectangle of the selected color appears to the left of the search icon.

You may have noticed that above the palette of colors there were two tabs, Basic and Advanced. We just set the color using the Basic Color Picker. The alternative is to use the Advanced Color Picker. Click the search icon again. Now click the Advanced tab. You have four ways to specify the color: By selecting the area to the left and the color spectrum in the middle; by entering the r, g, and b values directly; by entering the h, s, and v values directly; or by entering the hex value directly (do not forget to put the # in front of the hex value). Note that if you are entering the values directly, you will need to click inside the Color Picker or press tab for the color to update. When you are satisfied, click Apply and the selected color appears in the Default Value property. Most other places in IBM TRIRIGA that have color fields also provide the option of using the Advanced Color Picker.

Tip: Color field names should end with CO for easy identification later, e.g., cstDisplayColorCO.

Control number fields

A Control Number field contains a string that uniquely identifies a record. The use for a control number is described more fully in "Naming records". The system generates a record's control number when the record transitions from the null state to any other state. A Control Number field uses all the usual field properties, which are described in "Field properties".

The *Do not Auto Populate* and *Read Only* check boxes are grayed out and always checked. You cannot uncheck either check box.

If the *Generate on Create* check box is checked, the Control Number field is set to a generated value the first time that the record transitions to a state other than *null*. This is usually desirable.

The generated value is based on a counter associated with the business object. There are other settings that control the format of generated control numbers described in "Control numbers". If we use the default values for these settings, the control numbers generated for records created from the same business object are 1, 2, 3, etc.

If the *Generate on Create* is not checked, no value is automatically put in the Control Number field. This is useful only if you have a workflow to set a value in the field.

It is not usually a good idea to check a Control Number field's *Required* check box. The value of a Control Number field is usually not set until the first time the record that contains the Control Number field is saved. If the *Required* check box is checked, a form will not allow the record to be saved until the Control Number field has a value. Since control number fields are always read-only, you will be unable to put a value in the field or save the record.

IBM TRIRIGA Data Integrator does not properly handle updates to records for business objects containing publish names that include a Control Number field with the Generate on Create check box selected. In this scenario, Data Integrator is unable to determine that a record included in an upload file already exists in the system, causing new records to be created each time the file is uploaded. For more information, see *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

Tip: Control Number field names should end with CN for easy identification later, e.g., cstControlNumberCN.

Date fields

A Date field contains a date. A Date field has all the usual field properties, which are described in "Field properties". There are some additional properties.

Select the Relative property to indicate that this date is not fixed. A relative date is a date that is not bound to any time zone. For example, the beginning of the new year in 2011 was January 1, 2011 00:00. That does not represent a fixed point in time, rather it represents a particular point of time in the context of a time zone.

A benefit of a relative date is that an administrator can define it and reuse it many times. For example, an administrator defines a calendar named US Holiday to contain New Year's Day, President's Day, Independence Day, and other standard holiday observances. The administrator can then use the US Holiday calendar for resources in any of the time zones covered by the US and know that at runtime a relative time is used in the context of the resource and its time zone.

The *Default Value* property is used to specify a fixed or relative initial value for the field. To specify that the initial value of the field should be relative to the current date, select the radio button labeled *Current Date*.

If you leave the value of zero, then the default value is the current date. If you put in a value greater than zero, then the default value is that many days before or

after the current date. What determines whether it is a number of days before or after the current date is whether the value selected in the drop-down list is + or -.

To specify that the initial value should be a fixed date, select the lower radio button and enter a date into the field next to it.

The *Validation* property is used to specify what validation should be applied to values people enter for the field in a user interface. Only one value is supported for this property: *Valid Date*.

The *Formula Type* property is visible only if the *Formula* check box is checked. If the *Formula* check box is checked, it means that the value of the field will be determined by a formula and the field is always read only. The details of how to specify a formula are described in "Formulas".

The value in a Date field is stored in the database in milliseconds and contains both the date and the time. Before you perform duration calculations with Date fields, you must strip out the time portion with the DateFromDateTime function. The DateFromDateTime function returns "0:00" (GMT midnight) in the time portion.

Tip: Date field names should end with DA for easy identification later, e.g., cstProjectedStartDateDA.

Date and time fields

A Date and Time field contains a date and a time. Values are stored in the database in milliseconds. A Date and Time field uses all the usual field properties, which are described in "Field properties". There are additional properties.

Select the Relative property to indicate that this date and time is not fixed. A relative date and time is a date and time that is not bound to any time zone. For example, the beginning of the new year in 2011 was January 1, 2011 00:00. That does not represent a fixed point in time, rather it represents a particular point of time in the context of a time zone.

A benefit of a relative date and time is that an administrator can define it and reuse it many times. For example, an administrator defines a calendar named US Holiday to contain New Year's Day, President's Day, Independence Day, and other standard holiday observances. The administrator can then use the US Holiday calendar for resources in any of the time zones covered by the US and know that at runtime a relative time is used in the context of the resource and its time zone.

The *Default Value* property is used to specify a fixed or relative initial value for the field. To specify that the initial value of the field should be relative to the current date and time, select the radio button marked *Current Date & Time*.

You can select the default value to be the specified number of days, hours, minutes, or seconds before or after the current time. What determines whether it is before or after the current time is whether the value selected in the drop-down list is + or -.

To specify that the initial value should be a fixed date and time, select the lower radio button and enter a date and time into the field next to it.

The *Formula Type* property is visible only if the *Formula* check box is checked. If the *Formula* check box is checked, it means that the value of the field will be determined by a formula and the field is always read only. The details of how to specify a formula are described in "Formulas".

Date and Time fields mapped into Date fields are truncated to midnight (00:00:00).

Tip: Date and Time field names should end with DT for easy identification later, e.g., cstActualEndDT.

Duration fields

A Duration field contains a value that is the length of a period of time expressed in years, months, weeks, days, hours, minutes and seconds. The form uses all the usual field properties, which are described in "Field properties". There are some additional properties.

The *Default Value* property is used to specify an initial value for the field. To enter an initial value, click the *Enter* hyperlink. A small form pops up in which you to enter the quantity of years, months, weeks, days, hours, minutes, and seconds that are the field's initial value.

The *Formula Type* is visible only when Formula is selected. It is determined by a pair of radio buttons labeled Regular and Extended. Selecting the Regular radio button brings the Formula Field property. How to specify a regular formula is described in "Formulas". Selecting the Extended radio button and then the Enter hyperlink brings the Extended Formula panel. How to specify an extended formula is described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations.*

Duration is stored in the IBM TRIRIGA 10 database as a long value. When saving duration values with a workflow or IBM TRIRIGA Connector for Business Applications, use the following format to ensure your users see the correct value in a form. The formula is as follows:

(100000000000000 * ((years * 12) + months)) + (weeks * 604800000) + (days * 86400000) + (hours * 3600000) + (minutes * 60000) + (seconds * 1000) + milliseconds

Tip: Be careful when comparing duration fields in months or years in a formula. A year is equal to 365 days in milliseconds. A month is equal to 365 days in milliseconds divided by 12. When doing fencing comparisons, the length of a month is the same for any month, which can be misleading if comparing against actual dates since there are several months that do not have this exact number of days. The same is true for years since a leap year has an extra day.

Tip: Duration field names should end with DU for easy identification later, e.g., cstMeetingDurationDU.

Financial rollup fields

Financial Rollup fields are used to accumulate or summarize transaction results.

The properties and use of a Financial Rollup field are fully described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

Tip: Financial Rollup field names should end with FR for easy identification later.

Image fields

An Image field contains a value that is an image. An Image field has the usual field properties, which are described in "Field properties".

Tip: Image field names should end with IM for easy identification later, e.g., cstPortraitIM.

Label only fields

A Label Only field does not contain any value. A Label Only field is useful for adding labels to forms with no field associated with them.

Older versions of the platform required this field type to place a label on the form. Since you now can add a label directly to the form without storing it for each record, you should not use this field type.

Tip: Label Only field names should end with LA.

List fields

A List field can have a value that is chosen from a list of values. Classification is often a good or better alternative to List. There is a comparison of the two types in "Lists versus classifications". A List field uses all the usual field properties, which are described in "Field properties". There are some additional properties.

If the *Dependent* check box is checked, it means the list of values that may be selected for this field depends on the value selected for another field from another list. For example, suppose that a business object contains two List fields named *Country* and *State*. If the value selected for *Country* is *United States*, then the list of values that should be available for *State* should be a list of the US states. If the value selected for *Country* is *Canada*, then the list of values that should be available for *State* should be a list of the Canadian provinces.

There is more discussion of what it means for the *Dependent* check box to be checked. The properties that appear below the *Dependent* check box are different if the check box is checked. The following paragraphs assume that the *Dependent* check box is not checked.

The *Module* and *List* properties are used to identify the list of values that are available for this field. Lists of values are associated with a module. You must identify the module that the list is associated with before you can find the list.

The *Default Value* property is used to specify what the initial value of this field will be.

Tip: List field names should end with LI for easy identification later, e.g., cstCategoryLI.

Dependent lists

When the *Dependent* check box is checked, additional properties become available on the bottom of the form.

Parent

> The value of the *Parent* property is the name of the parent field and the list that it uses. The choices in this drop-down list are the names of only List fields that are already in the business object and have dependent lists. If the parent you want to select is not in this list, then check to see if it first needs to be added to the business object.

List

> The value of the *List* property is the name of the dependent list that values for this field will be chosen from.

> Each item in a dependent list is paired with an item from the parent list. When a value for this field is chosen, it must be chosen from items in the dependent list that are paired with the currently selected item in the parent list. For example, in the case of the list named *State or Province*, the list is set up as dependent on a list named *Country*.

> The list named *Country* contains the names of countries including *United States* and *Canada*. The items in the list named *State or Province* include the US states and the Canadian provinces. Each of the US state names is paired with the name *United States*. Each of the Canadian province names is paired with the name *Canada*. When it is time to use the *State or Province* list to select an item, only those items that match the currently selected *Country* are offered as possible values.

> If the current value of *Country* is *United States*, only the US states are offered as possible values because they are paired with *United States*. If the current value of *Country* is *Canada*, only the Canadian provinces are offered as possible values because they are paired with *Canada*.

List management

To add, change or remove items in a list, use the IBM TRIRIGA Application Platform List Manager. With the List Manager you can change the contents of a list and create new lists.

To navigate to the List Manager, select Tools > Administration > Lists.

By clicking the drop-down box below the Manage By section bar, you can choose to view by Name or by Type.

To modify a list, begin by clicking the radio button next to the name of the list in the *Manage By* section. After you click the radio button, the list of items in the selected list appears in the right side of the List Manager.

To add an item to a list, enter the text of the item in the field labeled *Value* and click the *Save* Entries action.

Items are normally added to a list in alphabetical order. To change the sequence of the items in the list, click an arrow in the *Order* column. Clicking an upward pointing arrow moves the item up a row; clicking a downward pointing arrow

moves the item down a row. To re-sort the list in alphabetical order click the Sort List action. To save the new sequence, click the Save Sequence action.

To delete an item from a list, check the check box next to the item and click *Delete*.

If a list is a dependent list, an additional field appears at the top of the list section. You can select any item in the parent list to be the value of the *Parent List* field. The columns below only contain items that are paired with the specified item in the parent list. If we add an item to a dependent list, the new item is paired with the specified item in the parent list.

Filtering the list of lists

In the upper left side of the List Manager is the list of lists. You must find the list you want to modify in this list before you can modify it. The list of lists is long. It may be inconvenient to find the list you are looking for.

In the lower left side of the List Manager is a section labeled Filter By. This section contains a list of modules with a check box next to each one. If none of the check boxes are checked, the list of lists contains all of the lists. If any of the check boxes are checked, the list of lists contains only lists associated with the modules whose check boxes are checked.

List creation

To create a new list, begin by clicking the List Manager's *New List* action.

Attention: Be very careful about how you fill out this form. Once you click the OK action and save what you enter, there is no way to change the information you entered.

Text that people will see in a user interface to identify the list goes in the *Label* field.

The unique name for the list that is used within the IBM TRIRIGA Application Platform to identify the list goes in the *Name* field.

A description of the list goes in the *Description* field.

The *Type* field can be used to group related lists in the List Manager. The List Manager can show the lists in order of their name or in order of their type. For example, this is useful if we have a few different lists that are related to training. In this situation we would specify *Training* as the *Type* for training related lists. Then when we ask the List Manager to show the lists ordered by type, the training lists will show up together.

The *System List* check box should be unchecked. If it is checked, the list cannot be deleted. Do not check this check box. Once you save a list's properties, you cannot change this check box.

The *Language* field is used for applications that need to function in more than one language. The simplest way to set this field is to US English or whatever language the application will be used in. A full explanation of how to internationalize applications can be found in the *IBM TRIRIGA Application Platform 3 Globalization User Guide*.

The List Manager organizes lists by the module they are associated with. We specify the module that the new list will be associated with by selecting the name of the module in the *Module* field.

If the *Dependent List* check box is checked it means that the new list will be a dependent list. Checking the *Dependent List* check box also causes additional fields to appear at the bottom of the form. Creating dependent lists is discussed in "Dependent list creation".

The value of the *Source Type* field can be *Manual* or *Dynamic*. If the *Dependent List* check box is checked, the value of *Source Type* is forced to *Manual*.

If the value of *Source Type* is *Manual*, it means that the contents of the list will be manually maintained through the List Manager, as discussed previously in "List management".

If the value of *Source Type* is *Dynamic*, it means that the contents of the list will be dynamically generated from the contents of records. In this case, the ordering will be alphabetical.

When the value of *Source Type* is *Dynamic*, additional fields appear at the bottom of the form.

When the value of the *Source Type* field is *Dynamic*, the *Dependent List* field is grayed out and cannot be checked. A list cannot be both dynamic and dependent.

Together, the *Source Module* and *Object Type* fields identify a business object. Use the *Source Module* field to select the name of the module that contains the business object. Use the *Object Type* field to select the business object.

Data from all records created from the specified business object is used to populate the list. The rows of data to the right of the *Fields* label identify the fields in the business object that have their *Result Column* check box checked.

The text of the items in the list comes from the contents of the selected fields in each record. If just the *Name* field is selected, this means that the items of the list will consist of contents of the *Name* field of every record that has been created from the *Country* business object. For example, the contents of the list might look like this:
- Canada
- Mexico
- United States

If more than one field is selected, the contents of the selected fields are assembled to form the text of each list item. The order in which they are assembled is determined by the numbers in the *Sequence* column. For example, suppose that both fields of *Country* are selected and the sequence number for *Name* is 1 and the sequence number for *Short Name (ISO-A2)* is 2. The contents of the list might look like this:
- Canada CA
- Mexico MX
- United States US

If the value of a field used in a dynamic list changes, the dynamic list does not update until the metadata cache is cleared. When a record that is the source of a dynamic list is deleted, the value no longer appears in the dynamic list.

Attention: Deleting a list that is used in conjunction with other data may produce unwanted or unexpected results.

Dependent list creation

When the *Dependent List* check box is checked, the *Source Type* field is forced to the value *Manual* and grayed out. Two additional fields appear at the bottom to identify the parent list.

The *Source Module* and *Master List* fields jointly identify the parent list. The *Source Module* field contains the name of the module that the List Manager associates with the parent list. The *Master List* field contains the name of the parent list.

Note fields

A Note field contains a formatted piece of text. There is no limit to the length of the text. A Note field has the usual field properties, which are described in "Field properties".

When the Localizable property is selected, it indicates that this field should be exported during instance data export for translation. Selecting this property specifies that this field is meant to have secondary language values. By default, this property is not selected.

Note field values are stored in the database in binary format. The string representation of the binary value sent for translation includes HTML tags. These HTML tags must be retained during translation.

Should you change an existing Note field from localizable to not localizable after the field has existing language values and internal values, you will see a warning message. The stale values remain in the language table.

An editable Note field displays to a user in a WYSIWYG editor with basic formatting and basic editing features. Users also can insert/update a table, embed an image, and view/edit the source HTML. Use the Style Manager to configure settings for a Note field's editor format options. The Style Manager is discussed in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*. A read-only Note field displays the HTML defined in the field without editor controls.

The audit data store has a limit of 4000 characters for any audited field. If an audited Note field's data exceeds 4000 characters, its corresponding audit record reflects the first 4000 characters of that Note field's value.

Note: Use pure HTML reports when you are building HTML form reports with Note data.

Tip: Note field names should end with NO for easy identification later, e.g., cstCommentsNO.

> **Attention:** A Note field cannot be used as any part of a record's name. Also, the IBM TRIRIGA Application Platform search tool does not find text in a Note field and the Report Manager does not allow reporting on this field. If these limitations are unacceptable, consider using the Text data type.

Number fields

A Number field contains a number. A Number field uses all the usual field properties, which are described in "Field properties". There are some additional properties.

The *Default Value* property is used to specify an initial value for the field. If blank, the user will be required to input a value.

The *UOM List* property indicates what sort of measurement the number in the field is for; e.g., area, length, quantity. It is possible for you to customize the list of unit of measure types. This is described in "Units of measure".

The *Create Base Field* property is visible only if the value of the *UOM List* property is *Currency*. If the check box in the *Create Base Field* property is checked, then when the properties for this field are saved, a base currency field will be created for this field. Base currency fields are discussed in "Base currency fields".

The name of the base currency field will be the name of this field followed by *Base*. For example if the name of a field is *CanadianCost*, then the name of its base field would be *CanadianCostBase*.

The *Default UOM* property specifies the default unit of measure to be associated with values in this field. The values in the drop-down list are the UOM Values for the UOM Type selected in the UOM List property.

Use the *Validation* property to specify the validation that should be applied to values entered for this field through a user interface. The possibilities are:

Accept Decimal
> This means numbers that do or do not contain a decimal point will be valid.

Accept Only Number
> This means that only numbers that do not contain a decimal point will be valid.

When the UOM List property is blank, the Display Mask, Storage Precision, and Rounding Rule properties are visible.

Display Mask controls the formatting of the display of values for this number field. The values to put in the Display Mask property are described in "Number formats". The formatting in a Number field takes precedence over UOM formatting when displaying Number fields in query reports or forms.

Storage Precision defines how many decimal places the platform is to use when computing and storing numbers entered in this number field. Enter the number of digits to the right of the decimal point. The maximum value is 12.

In Rounding Rule specify the rounding rule to be applied when rounding numbers entered in this number field to the number of decimal places in the Storage Precision property. The default value is Half Up. The options are as follows:

Ceiling
> Round the fractional digit so the value moves toward positive infinity. For example, if Storage Precision is 2, 1.234 => 1.24, 1.12 => 1.13, -1.234 => -1.23.

Down   Round the fractional digit so the value moves toward zero. The operation effectively truncates the value at the number of decimal places in Storage Precision. For example, if Storage Precision is 2, 1.2345 => 1.23, 1.328 => 1.32, -2.125 => -2.12.

Floor   Round the fractional digit so the value moves toward negative infinity. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.23, 1.235 => 1.23, -1.234 => -1.24.

Half Down
> Round the fractional digit using the digit to the right to determine the direction of the round. If the digit to the right is 6 or greater, round up. If the digit to the right is 5 or less, round down. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.23, 1.23 => 1.23.

Half Even
> Round the fractional digit up or down as needed to make it even. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.24, 1.245 => 1.24, 1.23 => 1.23.

Half Up
> Round the fractional digit using the digit to the right to determine the direction of the round. If the digit to the right is 5 or greater, round up. If the digit to the right is 4 or less, round down. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.24, 1.23 => 1.23.

Up   Round the fractional digit so the value moves away from zero. For example, if Storage Precision is 2, 1.234 => 1.24, 1.23 => 1.23, -1.234 => -1.24.

When the UOM List property is defined, the Use Custom Precision and Mask property is visible. This property specifies whether or not this number field will use the properties for Display Mask, Storage Precision, and Rounding Rule as they are defined on the UOM.

When Use Custom Precision and Mask is not checked, the platform uses the Display Mask, Storage Precision, and Rounding Rule as they are defined on the UOM.

When the Use Custom Precision and Mask property is checked, the Display Mask, Storage Precision, and Rounding Rule properties are visible and initially set to the values defined on the UOM. Any change you make only applies to this number field.

The *UOM Source Attribute* property contains a drop-down list that allows you to specify the name of a field in the business object. If the *Formula* check box is *not* checked, the drop-down list will contain names of UOM fields only. The unit of measure for this field will be determined by the value of the named UOM field.

If the *Formula* check box *is* checked, the drop-down list will contain names of UOM and Number fields. The unit of measure associated with the named field is the unit of measure of the formula's result.

The value of the *UOM Source Attribute* property is the name of another Number field in this business object. If the *UOM Source Attribute* property has no value, then the unit of measure for the field is defaulted to the value specified in the Default UOM property and may be overridden by the user at runtime.

If you specify a value for the *UOM Source Attribute* property, the named field's unit of measure becomes the unit of measure for this field. Also, this field's *Default UOM* property is grayed out and set to the value of the other field's *Default UOM* property.

If you specify a value for the *UOM Source Attribute* property, the *UOM List* property and the *Default UOM* property are grayed out and their values ignored.

When there is a value in the Threshold Source Attribute property, this number field is a scored number field, meaning it will be scored (as described with the Comparison property in "Smart sections"). The score is displayed as a red, yellow, or green image to the left of the number value.

The value of the Threshold Source Attribute property must be a locator field that references the Threshold business object. This locator field must be in either a General section or a single-record smart section. For a number field that is in a smart section, the linked locator field must reside on the referenced business object. The threshold record defines the threshold ranges and contains a UOM value that is used to convert the scored value.

Select the linked locator field in the Threshold Source Attribute drop-down list. The list only shows locator fields in the same business object that references the Threshold business object.

A scored number field can be used in a form section, non-table smart section, table smart section, or vertical table section.

The score for the field is calculated during the rendering of the field and is not stored. The platform compares the UOM of the number field with that of the threshold record. If a conversion is needed, the platform converts the number value before comparing it with the threshold value to calculate the score.

During runtime, the system uses the threshold record that is currently selected by the linked locator fields. If the locator field does not have a value, the number field displays without a score.

The score is updated when the tab is reloaded. This means that if the value of the number field, or the threshold record, or the locator field that links to the threshold record, is changed, the score will not update (if needed) until the tab is reloaded.

If the check box in the *Formula* property is checked, it means that the value of the field will be determined by a formula and will be read only. This property is discussed in "Formulas".

The *Sum this Field* check box is grayed out unless the field is part of a smart section. If this check box is checked, it causes a summary field to be generated in the business object that contains the smart section. There is more detail about sums in "Record organization".

Number field display and storage

The ability to override the Display Mask and Storage Precision properties with the Use Custom Precision and Mask property provides clarity in the display and storage precisions for number fields and UOM fields and a higher level of granularity.

For example, suppose for efficiency or reporting reasons you want all area measurement fields to have 2 decimal places and a storage precision of 6. You could set the corresponding UOM appropriately. Now suppose you want some specific fields in a business object, for internal calculation for example, to have more decimal places and higher precision. With IBM TRIRIGA, you can use the custom display mask feature and not have to update all affected fields.

When a property is selected to be overridden, the initial values are from the values defined in the next level, as illustrated in the following figure.
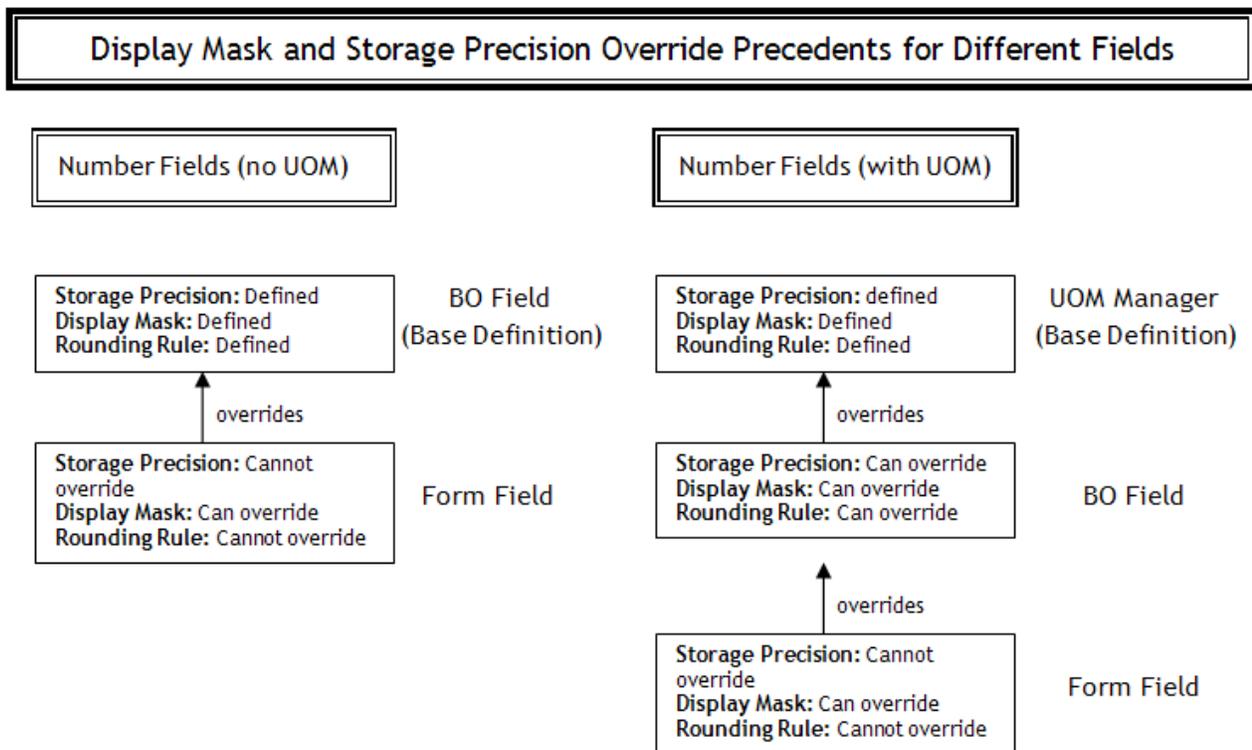
Figure 17. Display mask and storage precision override precedents

At the base definition level, should default values be employed, the platform uses the default values shown in the following table for each property.

Table 7. Base definition default values

Property	Default Value
Storage Precision	12

Table 7. Base definition default values (continued)

Property	Default Value
Display Mask	#.####
Rounding Rule	ROUND_HALF_UP

Tip: Number field names should end with NU for easy identification later, e.g., cstCapacityNU.

Password fields

A Password field contains text. Fields in user interfaces that allow users to type a value for a Password field do not display the actual value of the field or the text that is entered. A Password field uses all the usual field properties, which are described in "Field properties". There are two additional properties.

The *Validation* property may have no value, in which case anything entered in a user interface for the value of the Password field is acceptable. The *Validation* property may contain the value *Alpha Only No Spaces*, in which case what is entered in a user interface for a Password field must contain only letters to be acceptable. Anything entered that contains a number, spaces, or punctuation is not acceptable.

If you set the *Validation* for a Password field to *Alpha Only No Spaces*, it encourages people to make the password a single word that will be easy for them to remember. The drawback to this is that passwords that are regular words are more easily guessed.

Setting no Validation for a password field by leaving the *Validation* property blank allows the password to be multiple words that can contain numbers and punctuation. Such passwords may be harder to remember but also are harder to guess.

The *Reversible Encryption* property defines how the platform will store the password in the database. The three options are as follows:

Do Not Encrypt
This option stores the password as plain text in the database.

Encrypt - Reversible
This option encrypts the password using a reversible technique and stores that encrypted value in the database.

Encrypt Non-Reversible
This option encrypts the password using a non-reversible technique and stores that encrypted value in the database.

Once a Password field has been saved with the Reversible Encryption property set to Encrypt Non-Reversible, the Reversible Encryption property is disabled. If you wish to change the Reversible Encryption property, you must delete the field and add it again.

A Password field can be mapped to a Text field, for example using a workflow. If the Reversible Encryption property is Do Not Encrypt or Encrypt - Reversible, a Text field mapped from a Password field can be displayed as plain text. If the

Reversible Encryption property is set to Encrypt Non-Reversible, a Text field mapped from a Password field can be displayed but the output will be unreadable.

Password fields set to Encrypt Non-Reversible with identical content will not compare as equal in a workflow. Instead, try the following procedure in a password change/verification scenario where the user enters the password twice to make sure they entered it correctly: In the temp record used to receive and compare the two fields containing the new password, set the fields to Encrypt - Reversible. After the workflow validation is complete, have the workflow use one of the values to set/update the password on the user's profile in a different field that is Encrypt Non-Reversible. Then discard the values of the Encrypt - Reversible fields.

Tip: Password field names should end with PA for easy identification later, e.g., cstPasswordPA.

System read only fields

A System Read Only field contains a value generated internally within the IBM TRIRIGA Application Platform for the record that contains the field. System Read Only fields are always read-only. Their value cannot be directly changed from a user interface. A System Read Only field uses all the usual field properties, which are described in "Field properties". There is one additional property.

The value of the *Sub Attribute Type* property determines which of the record's internal information the IBM TRIRIGA Application Platform places in the field. The Sub Attribute Type property is required and its value cannot be changed after the field has been saved and created. Reuse existing System Read Only fields instead of creating new ones. Existing fields are noted in parentheses below. The values for Sub Attribute Type follow:

BO Record Id

> This is a string of text generated internally by the IBM TRIRIGA Application Platform that uniquely identifies the record.
>
> (triRecordIdSY)

BO Type Id

> This is a string of text that uniquely identifies the business object that was used to create the record.
>
> (triBusinessObjectIdSY)

BO Type Name

> This is the name of the business object that was used to create the record.
>
> (triBusinessObjectNameSY)

BO Label

> This is the label of the business object that was used to create the record.
>
> (triBusinesObjectLabelSY)

Created DateTime

> This is the date and time when this record was created.
>
> (triCreatedSY)

Form Id

> This is a string of text generated internally by the IBM TRIRIGA Application Platform that uniquely identifies the form associated with this record.
>
> The form associated with a record is initially the form that was used to create the record. If the record was created by a workflow task, the task specifies the form initially associated with the record.
>
> The form associated with a record can be changed by a workflow using the Modify Metadata workflow task described in "Modify metadata task".
>
> (triFormIdSY)

Form Label

> This is the label of the form associated with this record.
>
> (triFormLabelSY)

Form Name

> This is the name of the form associated with this record.
>
> (triFormNameSY)

Modified DateTime

> This is the date and time when this record was last modified, stored as a string. Modified DateTime cannot be used as a DateTime field in query filters.
>
> (triModifiedSY)

Object State

> This is the name of the record's current state from its state transitions. State transitions are discussed in "Life cycles".
>
> (triRecordStateSY)

Parent Id

> If the business object used to create this record is part of a hierarchy module, then this is a string of text that uniquely identifies the record's parent in the hierarchy. Hierarchies are described in "Hierarchies".
>
> (triParentIdSY)

Path

> If the business object used to create this record is part of a hierarchy module, then this is the path from the root of the hierarchy to this record.
>
> (triPathSY)

Project Id

> This is a string of text that uniquely identifies the project that the record belongs to. Projects are discussed in "Projects".
>
> (triProjectIdSY)

Project Name

> This is the name of the project that the record belongs to. Projects are discussed in "Projects".
>
> (triProjectNameSY)

Record Name

> The value used as this record's unique name. This is explained in "Naming records".
>
> (triRecordNameSY)

Modified DateTime (Number)

> This is the date and time when this record was last saved, stored as a number. Modified DateTime (Number) can be used as a DateTime field in query filters.
>
> **Tip:** System Date fields may not display accurately on reports or account for time zones for end users. In IBM TRIRIGA 10 applications, these fields are typically named triModifiedSY and triCreatedSY and are stored as text strings in the server's time zone. A field of type Modified DateTime (Number) must be created and added to objects on which you wish to report. This field type can be retrieved through IBM TRIRIGA Connector for Business Applications. It can be used as a dynamic query filter through the API.

Created DateTime (Number)

> This is the date and time when this record was created, stored as a number. Created DateTime (Number) can be used as a DateTime field in query filters.
>
> **Tip:** System Date fields may not display accurately on reports or account for time zones for end users. In IBM TRIRIGA 10 applications, these fields are typically named triModifiedSY and triCreatedSY and are stored as text strings in the server's time zone. A field of type Created DateTime (Number) must be created and added to objects on which you wish to report. This field type can be retrieved through IBM TRIRIGA Connector for Business Applications. It can be used as a dynamic query filter through the API.

> **Tip:** System Read Only field names should end with SY for easy identification later.

Text fields

A Text field contains text. A Text field uses all the usual field properties, which are described in "Field properties". There are additional properties.

When the Localizable property is selected, it indicates that this field should be exported during instance data export for translation. Selecting this property specifies that this field is meant to have secondary language values. By default, this property is cleared.

Locator fields cannot be localized and have the Localization property cleared and read-only. A locator field contains a value from another record. The localization of the value is controlled by that other record. If the other record is localizable, the language values stored in the referenced business object record are used when rendering the translated value to the user. To localize a locator field, set the Localizable property in the referenced business object field.

Should you change an existing Text field from localizable to not localizable after the field has existing language values and internal values, you will see a warning message. The stale values remain in the language table.

The *Default Value* property is used to specify an initial value for the field.

The *Size* property is used to specify the maximum number of characters which can be stored in this field.

The *Validation* property may have no value, in which case anything entered in a user interface for the value of the Text field is acceptable. These are the other possible values you can select for *Validation* property:

Make Lower Case
> If the value of this property is *Make Lower Case*, any value for this field is acceptable. However, any upper case letters entered into this field are converted to lower case.

Make Upper Case
> If the value of this property is *Make Upper Case*, any value for this field is acceptable. However, any lower case letters entered into this field are converted to upper case.

Alpha Only No Spaces
> If the value of this property is *Alpha Only No Spaces*, then whatever is entered in a user interface as a value for this field may contain only letters. Anything that contains numbers, spaces or punctuation is not acceptable.

Only Numeric
> If the value of this property is *Only Numeric*, then only digits may be entered in a user interface as a value for this field.

The hyperlinks in the *Formula* property allow us to enter a formula that determines the value of this field. The details of how to specify a formula are described in "Formulas".

Attention: There is a 1000 character limit on the length of the text in a Text field. If you need a field that can contain an unlimited number of characters, consider using a Note field. Note fields are discussed in "Note fields".

Tip: If you encounter instances of special characters in Text fields saving as question marks, this can be caused by database encoding. To check, run the following query on your database:

```
select value from NLS_DATABASE_PARAMETERS
where parameter='NLS_CHARACTERSET'
```

If the value returned is not a type UTF-8 or UTF-16 encoding, this may be the cause of the issue. For example a common default encoding for NLS_CHARACTERSET is WE8MSWIN1252, which is a 1 byte per character encoding that cannot store as many character types as UTF-8. Changing the character encoding for the database should resolve this issue but also will result in slower performance.

Tip: Text field names should end with TX for easy identification later, e.g., cstNameTX.

Locator fields

If the *Locator Field* check box is selected, it means that this field can be populated with the value of a field from another record and a reference and association (note, the reference and the association are two separate things) to that record are created from the record containing the locator field.

This establishes the value of this field at the time it was populated from the other record. However, should the value in the referenced field change and be saved, the value in the locator field is not updated to the new value of the referenced field.

For example, the triProduct business object has a locator field named triVendorCompanyLookupTX that is set to create a reference to an Organization and to display the Organization's hierarchy path. If a triProduct record's triVendorCompanyLookupTX field were populated with a vendor whose hierarchy path value was Organizations\External Companies\Renovations, the locator field's value would be Organizations\External Companies\Renovations. If, after populating the locator field, the vendor's Name field (which makes up the last part of the hierarchy path value) was changed to ZetaBank, the locator field's value in the triProduct record would not be changed to the new value and would still have the value Organizations\External Companies\Renovations. This is because a locator field receives a copy of the referenced record's field value and not a link to the value.

Locator fields are beneficial if a record containing a locator field needs to retain the original value of the referenced record as it was when the locator field was populated with the referenced record. If your situation calls for a live link to the referenced record's most up-to-date value, you may want to use a smart section. Smart sections are discussed in "Record organization".

At runtime, if an alternate form is defined for the locator and the user selects the control to open the locator, then the alternate form displays.

When the *Locator Field* check box is selected, you use the *Locator Module* field to specify which field of which business object populates the locator. Begin by selecting the name of the module the field's business object is defined in. Select that module name as the value for the *Locator Module* field.

After you select a module name as the value of the *Locator Module* property, you specify the business object and field to be associated with this field. You also specify the association to be used to connect the two business objects.

Business Object Name

> The value of this property is the name of the business object that contains the field whose value populates this field. Choose the value of this field from a drop-down list of the business objects in the module that was specified for the value of the *Locator Module* property.

> The drop-down list also contains the value *All*. The meaning of *All* is explained below.

Associated String

> The value of this property is the name of an association that connects this field's business object to the business object whose name is the value of the Business Object Name property. The choices available in the drop-down list are the business object-level association definitions that have been created

from the object that contains the locator field to the object that is specified in the Business Object Name property.

If the value of the Business Object Name property is *All*, the value of this property is the name of an association between this field's business object and the base business object of the module specified as the value of the *Locator Module* property. Such an association allows records created from this field's business object to be associated with records created from any business object in the module named by the *Locator Module* property.

Locator Field

This property's value is the name of the field that this field is populated from in the business object named by the *Business Object Name* property. The name is prefixed with the name of the section that contains the field. If the field is not in a smart section, the name *General* is used for the section name.

The drop-down list contains the names of fields in the business object named by the value of the *Business Object Name* property. If the value of the *Business Object Name* property is *All*, the drop-down list contains the names of fields in the base business object of the module specified as the *Locator Module* property's value.

After you finish specifying properties values for this form, click the form's *Ok* action. After you save the locator field properties, additional field properties named *Locate Using* and *Locator String* appear.

Locate Using

The *Locate Using* property has two hyperlinks. The text of the upper hyperlink contains the values specified for the *Business Object Name* and *Locator Field* locator field properties. Clicking the upper hyperlink causes the form used to specify these properties to reappear below the field properties.

The lower hyperlink in the *Locate Using* property has the text *Edit Mapping*. Clicking this lower hyperlink causes a form to appear in the List panel in which you to specify that the value of other fields in this business object should be taken from the same record associated with this locator field.

If you click the search icon next to a field's name in the Locator Mapping form, a list pops up in which you specify which field of the record associated with this locator field is be used to set the value of the field in this business object whose icon you clicked.

Locator String

The *Locator String* property is read-only. Its value is the value you specified for the *Associated String* locator field property.

There is some similarity between smart sections and locator fields. For example, setting the value of a locator field is done in the same way as setting the value of a smart section, and this is different from setting other field types. The way that a workflow sets the value of a record's contents is called Object Mapping and is discussed in "Object mapping".

When a locator field is initially created or is cleared, the objectID of the referenced object is set to null.

Autocomplete in locator fields

When a locator field is used in a form, the user sees the field's label, a text box, the search (locator picker) icon, and the clear (X) icon. Autocomplete's goal is to reduce the user clicks needed to reach the final search target.

Autocomplete functionality is available for all locator fields. Specify whether or not your users have this feature in the USE_AUTO_COMPLETE_IN_LOCATOR_FIELD property in the TRIRIGAWEB.properties file. The default is for autocomplete to be enabled. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

The AUTO_COMPLETE_MIN_CHAR property in the TRIRIGAWEB.properties file sets the minimum number of characters a user must type to trigger an autocomplete search. The default is 3 characters. Another condition for activating autocomplete is the user's keystroke speed. The system measures when to display the autocomplete list using a 400 millisecond lag. A slow typist, after typing the minimum number of characters, should see an indicator icon and, within a reasonable time frame, the list. A fast typist may need to pause after typing the last character in order to trigger the list.

Autocomplete is not case sensitive. As the autocomplete functionality is triggered, a list appears under the text box.

The list shows suggested items. The first suggested item in the list is highlighted. For each suggested item, if the locator field has defined locator mapping fields, those field values display. The characters in the main field value that match the user's input are bold.

The user can perform the following actions at this stage:
* Accept the first suggested item by pressing the Tab key or Enter.
* Use the up/down arrow keys to select an item. Pressing Enter or the Tab key completes the selection.
* Scroll and click an item. The selection completes automatically.
* Type in more characters to narrow the suggested items list.
* Delete characters, which may broaden the suggested items list.
* Key in a different set of characters.
* Press the Esc key to clear the field and hide the list.

A user can take advantage of autocomplete and their familiarity with their company's data by entering one or two characters and pressing the Tab key. The system performs a query and one of the following:
* If the query returns one record, that record completes the selection.
* If the query returns more than one record, the system presents the records in a list.
* If the query returns no records, the user sees a "no record match" message and the cursor stays in the field waiting for the user's next action.

Users that are using a screen reader do not see the data in the autocomplete list because the focus is on the input text field. In order to take advantage of autocomplete, such a user can follow the steps below.
* Enter the exact string into the input text field and tab out of the field. This selects a unique record for the value of the field.

- If multiple records display for the string that was entered, tab out of the field and use the search icon to select a value.
- If no record matches the string that was entered, tabbing out of the field is not allowed. If the screen reader changes focus to the Tab key, select the Esc key. The user can then tab to the next element.

After the user completes the selection, the system populates all related fields. The system also updates the field's label to show the anchor link. If the user clicks this hyperlink, the system displays the selected record's detail.

The list disappears when the user clicks within the text box, tabs out, clicks outside the text box, presses Esc, or makes a selection.

If the user clicks a text box that has a value, presses a key to initiate an autocomplete search, and then clicks outside the text box, the original value in the text box is retained.

After selecting a record, the user can perform another search without clearing the field with the clear (X) icon.

Instead of using autocomplete, the user can click the field's search icon to popup the locator query full search window. This method is preferred for users that use a screen reader.

The autocomplete functionality also is available on single-record smart section text fields, as described in "Smart sections".

Locator field overloading

One benefit of using locator fields to reference other records (as opposed to smart sections) is that a locator field can be populated with record types other than those they are defined to receive in the Data Modeler. This ability, called overloading, is beneficial in a few instances. First, the ability to overload a locator field is beneficial when you need to populate a locator field with records from more than one Business Object, or even from more than one Module. Similarly, it also is beneficial when all of the types of records you may need to populate into a locator field are not known at the time the locator field definition is created.

For example, the triApproval business object is used in the IBM TRIRIGA approval engine and has been set up to work with most kinds of business objects. During the approval process, a triApproval record is created and associated to the record being sent for approval. The association is created by populating a locator field named triLinkedRecordTX with the record being sent for approval via a workflow. By populating the locator field, both a direct reference and an association from the triApproval record to the record being sent for approval are created. However, since triApproval records can potentially be created for an almost unlimited number of types of records, the corresponding number of types of values that may need to be referenced by the single triLinkedRecordTX locator field is also virtually infinite, and hence, the need to overload locator fields is sometimes necessary.

When overloading a locator field with a value it is not defined in the Data Modeler to receive, certain steps need to be taken to ensure the process works correctly:

First, since a locator field can only be defined to create an association to one type of business object in the Data Modeler, when overloading a locator field the type of association that will be created when overloading a locator field is still determined

by the Association String property defined in the locator field, although the association type also should be specified by the workflow. This is done by setting the Association Type property in the Object Mapping window of a Create Record or Modify Records task in a workflow (this setting is described in "Object mapping"). The Association Type property only shows those association types for which business object-level association definitions have been defined, from the perspective of the business object that contains the locator field to be overloaded (e.g., the triApproval object in our scenario above) to the Business Object whose records will be referenced by, and mapped into, the locator field (for more information on creating association definitions, see "Association definition"). Even though using Source mapping to populate the locator field usually creates the association as defined in the locator field in the Data Modeler, this step ensures the association gets created in the event that the record you are populating into the locator field is still in the null state (e.g., for a new record whose first state transition has yet to be invoked). Note, that in this scenario, the workflow mapping results in the locator field not being populated and having a blank value, since there is no Record Name or other field value yet for the record to map into the locator field.

The second requirement to overload locator fields is to make sure the locator field is populated using the Source option in the Object Mapping window in the locator field. The process of populating locator fields is described in "Object mapping".

Time fields

A Time field contains a value that is a time, such as 3:21 PM. A Time field uses all the usual field properties, which are described in "Field properties". There are some additional properties.

The *Default Value* property is used to specify a fixed initial value for the field. The initial value is specified as an hour in the range 0-23, a minute in the range 0-59 and a second in the range 0-59.

Tip: Time field names should end with TI for easy identification later, e.g., cstStartTimeTI.

UOM fields

A UOM (Unit of Measure) field identifies a unit of measurement such as pounds, feet or gallons. UOM fields are usually used with Number fields. One UOM field can be used to specify the unit of measure for any number of Number fields.

Though a Number field can be used to determine the unit of measure of other number fields, it can be more natural for people to specify a unit of measure for multiple number fields by using a field that is not associated with any single Number field.

A UOM field can also be used to determine the kind of measurement that a number is for, such as weight, length, or volume. A UOM field uses all the usual field properties, which are described in "Field properties". There are some additional properties.

The *UOM List* property is used to select the kind of unit that the field will allow to be selected, such as weight, length or volume. The type of unit that can be selected can be changed if the *Edit UOM List* check box is checked.

The *Default UOM* property specifies the initial value of this field in new records.

The Edit UOM List property specifies that at runtime the user can change the type of units measured not just the unit value. For example the user could change the UOM List from Length to Area which would change the unit values available from feet and meters to square-feet and square-meters.

Tip: UOM field names should end with UO for easy identification later, e.g., cstAreaUO.

Url fields

A Url field contains string of text that should be a valid URL. Fields in user interfaces that render a Url field also provide a way to open the URL specified by the contents of the field. A Url field uses all the usual field properties, which are described in "Field properties". There are additional properties.

The *Default Value* property can be used to specify the initial value of this field in a new record. The system allows the following prefixes: http://, https://, file://, and ftp://. If the protocol is not http://, https://, file://, or ftp://, the system prefixes http://

Tip: Use the full URL when entering external URLs; e.g., http://www.ibm.com.

Tip: Url field names should end with UR for easy identification later, e.g., cstExternalInfoUR.

Lists versus classifications

The Classification and List data types are similar: both allow someone to select a single item from multiple possibilities. To help you decide which data type is the better choice for a particular field, we have provided this description of the differences between the data types.

- Lists are usually a static set of choices. In general, the set of choices presented by a list is determined in advance and is not normally changed by an application. (The exception is a dynamic list that is dynamically generated from the contents of records, as described in "List creation".) Classifications can be a dynamic set of choices that reflect data the application can control.
- Lists are easier to set up and are a good choice to represent single options that will never change. Classifications are harder to set up but, being record data, support future changes better.
- Classifications make it easy to organize kinds of things into a hierarchy. Some hierarchical organization is possible with dependent lists. However, managing hierarchies with more than three levels is not practical to do with dependent lists.
- Classifications may be used to identify live records. Lists are simply names.
- Classifications can have data associated with them that can be used to influence the outcome of a computation. This relationship between a selected classification and the outcome of a computation can be indirect. Because a List is just a selection of text strings, a choice of a list value can only have a direct relationship to the outcome of a computation.

It is not possible to make a blanket statement that Classifications are better than Lists or that Lists are better than Classifications. The better choice depends on the application and the field. Be sure to consider all of the pros and cons before deciding.

Units of measure

Number fields have a unit of measure associated with them. The unit of measure is associated with a Number field as part of its definition. The definition of a Number field is discussed in "Number fields".

A number's unit may be displayed in a form or not. For example, in a form that displays textbook information, the textbook height may consist of two fields, one for the height value (such as 7) and one for the unit of measure (such as inches).

The units of measure are displayed as the selected item in a list. The list contains other units of measure that are used to measure the same kind of thing (such as length or weight). For example, with inches, the kind of thing being measured is length. Some of the other units of measure in the list may be feet, yards, meters and centimeters.

When a form is initially displayed, it shows numbers with the unit of measure that they are stored with. If the *Width* and *Height* fields in a form initially display with inches selected, we know that these numbers are stored as inches.

If we select a different unit of measure, the number is converted to the selected unit of measure. For example if we change the unit for *Width* from inches to centimeters, the number changes from 7 to 17.78. You will need to save the record before the conversion applies.

When someone enters a value for a number and selects its unit of measure, that number and the selected unit of measure are stored in the record's field.

Based on the ability of the IBM TRIRIGA Application Platform to handle different kinds of measurements, we see that it knows about different kinds of measurement such as length, angles, and mass. It knows about specific units used for each kind of measurement. For example it knows that length is measured in units such as inches, feet, yards, meters and kilometers. It knows that area is measured in units such as acres and square yards. It is also clear that it knows how to convert between different units that measure the same kind of thing.

The IBM TRIRIGA Application Platform uses two kinds of records to represent its knowledge about units of measure:

- *UOM_Type* records are used to describe kinds of things that can be measured.
- *UOM_Values* records are used to describe actual units of measure and how to convert between them.

The IBM TRIRIGA Application Platform comes with records that describe most common kinds of measurements and most common units of measure. If all the kinds of measurement and units of measure your application will need are already in the IBM TRIRIGA Application Platform, feel free to skip ahead to "Number formats".

If your application needs to use measurements that are not already in the IBM TRIRIGA Application Platform environment, see the following "Example: Managing units of measure".

Example: Managing units of measure

Both kinds of records used to represent knowledge about measurements can be found by navigating to Tools > Administration > Unit of Measure (UOM). Clicking Types brings up a list of existing UOM types.

We will explain how to add a new kind of measurement and units of measure by working through an example. The kind of measurement will be Digital Data. The units of measure will be byte, kilobyte, megabyte, gigabyte, etc.

To add a new kind of measurement, click the *Add* action, which causes a UOM_Type form to appear.

Put the name of the kind of measurement in the field labeled *UOM Type*. This is the only field we need to fill in at this time.

UOM Type Code is a read-only field. It is used internally by the IBM TRIRIGA Application Platform.

The *Base UOM* field will be filled in automatically when we create the first unit of measure for *Digital Data*. The unit of measure whose name appears in the *Base UOM* field is used to organize conversions between different units of measure.

Tip: In the unlikely event that you have an invalid Base UOM for a UOM Type, the system may not function properly. If the following SQL returns any UOM Types, it is an indication that the UOM Type has an invalid Base UOM and should be fixed:

```
SELECT * FROM uom_types a WHERE base_uom IS NOT NULL
AND base_uom != '' AND NOT EXISTS
(SELECT 'x' FROM uom_values b WHERE a.UOM_TYPE_CODE =
b.UOM_TYPE_CODE and a.base_uom = b.UOM_VALUE)
```

One way to organize conversions between different units of measure is in a matrix as shown in the following table. This organization supplies a conversion factor to convert from every unit of measure to every other unit of measure.

Table 8. Digital data unit conversion matrix

	To Bytes	To Kilobytes	To Megabytes	To Gigabytes
From Bytes	1	0.0009765625	0.0000009536 7431640625	0.0000000009 3132257461 5478515625
From Kilobytes	1024	1	0.0009765625	0.0000009536 7431640625
From Megabytes	1048576	1024	1	0.0009765625
From Gigabytes	1073741824	1048576	1024	1

Using this matrix organization, to convert from one unit to another, find the correct conversion factor and multiply by it. For example, to convert from gigabytes to kilobytes, the table example tells us to multiply by 1048576.

The unfortunate thing about this organization is that it may require an unreasonably large number of conversion factors. The table example shows 4 units of measure and requires 16 conversion factors. This is not so bad, but it gets worse with more units. For example, 10 units require 100 conversion factors, whereas 25 units require 625 conversion factors!

To avoid needing so many conversion factors, the IBM TRIRIGA Application Platform uses a different organization that requires only one column of the matrix. The unit of measure that corresponds to the matrix column we choose to use is called the *base UOM*.

Using this organization, conversion from one unit to another is done by first converting from the original unit of measure to the base UOM. Then the base UOM is converted to the target unit. The conversion from the base unit to the target unit is done by dividing by the conversion factor instead of multiplying. We will make this clearer by working through an example.

Suppose we have decided that the base UOM for Digital Data will be kilobytes and that we want to convert from gigabytes to megabytes. We would first convert from gigabytes to kilobytes by multiplying by 1048576 and then convert from kilobytes to megabytes by dividing by 1024 like this:

```
1048576 / 1024 = 1024
```

This works out the same as if we had set up the entire matrix, at the expense of needing an extra division.

Attention: Be sure you understand the concept of a base UOM before you define any units of measure for a new type of measurement. The first unit of measure that you create for a type of measurement will be its base UOM.

In theory, given a set of units used to measure the same kind of thing, you could you pick any of the units to be the base UOM. Since every unit can be converted to every other unit, it should work no matter which unit you pick. However, it turns out that some units may be a better choice for base UOM than others.

The reason that some units of measure may be a better choice for base UOM than others has to do with the way the IBM TRIRIGA Application Platform stores numbers. There is a limit to the size of number that it stores. The IBM TRIRIGA Application Platform can store a number with up to 20 digits to the left of the decimal point and up to 12 digits to the right of the decimal point.

Because of the limitation on the sizes of numbers that the IBM TRIRIGA Application Platform can store, choose a base UOM that allows all the conversion factors to be represented exactly. Looking at the conversion matrix in the same table example, we see that converting to megabytes or gigabytes involves conversion factors that have more than 12 digits to the right of the decimal point. This means that both megabytes and gigabytes would be bad choices for the base UOM because the IBM TRIRIGA Application Platform would not be able to store all the digits of some of the conversion factors. The results of those conversions would not be correct.

The conversion factors for bytes and kilobytes do fit into what the IBM TRIRIGA Application Platform can store. This makes either of them a good choice for base UOM.

Returning to our example, we decide to make Kilobytes the base UOM for Digital Data.

- To do this, create a new *UOM_Values* record.
- To create a new *UOM_Values* record, navigate to Tools > Administration > Unit of Measure (UOM) > Values, and click the *Add* action.
- Set the value of the *UOM Type* field to *Digital Data*. The system completes the rest of the *General* section with information from the *UOM_Type* selected.
- Next, fill in fields of the UOM Details section, which describes information about the unit of measure.
- Enter the unit of measure's name in the *UOM* field and put the unit of measure's abbreviation in the *UOM ABBREVIATION* field.
- The default value for the *Conversion Factor* field is 1. The value of *Conversion Factor* for a base UOM should be 1.
- The rest of the fields are not relevant for Kilobytes, so we finish by clicking the *Create* action. *Kilobytes* becomes the base UOM for Digital Data because it is the first unit of measure defined for Digital Data.
- To complete the example, create *UOM_Values* records to describe bytes, megabytes, and gigabytes.

There are additional fields in the UOM_Values form that we did not use to define kilobytes. The first of this is *Conversion Offset*. The *Conversion Offset* field is used to specify a value that needs to be added before multiplying by the value of the *Conversion Factor* field.

To see how the *Conversion Offset* field is used, look at the example of degrees-fahrenheit. The base unit that the platform uses for temperature is degrees-celsius. The formula to convert from Fahrenheit to Celsius is:

```
ºC = (ºF - 32) x 5/9
```

The record that describes degrees-fahrenheit is shown in the following figure.

Figure 18. Degrees-Fahrenheit

The remaining fields are most often used with units of money (currency), but may be used for any kind of unit.

The value of the *Display Mask* field is used to determine how numbers with the unit of measure will be formatted when displayed. If no value is supplied for the *Display Mask* field, default formatting is used. The default formatting follows these rules:

- If a number is negative, it is formatted with a minus sign (-) at the beginning.
- The number will be formatted with no leading zeros, unless the magnitude of the number is less than 1. If the magnitude of the number is less than 1, then there will be one leading zero.
- If the number has a fractional part, it will be formatted with a decimal point followed by one or more digits.
- If a number has a fractional part, it will be formatted with only as many digits after the decimal point as are needed. It will have no trailing zeros.
- If a value has more digits to the right of the decimal place than shown in the Display Mask property, the platform defaults to using ROUND_HALF_UP to round the value in the display.

Here are examples of numbers with default formatting:

- 0
- -2
- 34
- -0.234
- 0.234
- 123.4565

See "Number formats" for a description of the values to put in the Display Mask field to control the display formatting of numbers.

Storage Precision defines how many decimal places the platform is to use when computing and storing numbers defined with this UOM. Enter the number of digits to the right of the decimal point. The maximum value is 12.

In Rounding Rule specify the rounding rule to be applied when rounding numbers to the number of decimal places in Storage Precision. The default value is ROUND_HALF_UP. The options are as follows:

ROUND_CEILING
> Round the fractional digit so the value moves toward positive infinity. For example, if Storage Precision is 2, 1.234 => 1.24, 1.12 => 1.13, -1.234 => -1.23.

ROUND_DOWN
> Round the fractional digit so the value moves toward zero. The operation effectively truncates the value at the number of decimal places in Storage Precision. For example, if Storage Precision is 2, 1.2345 => 1.23, 1.328 => 1.32, -2.125 => -2.12.

ROUND_FLOOR
> Round the fractional digit so the value moves toward negative infinity. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.23, 1.235 => 1.23, -1.234 => -1.24.

ROUND_HALF_DOWN

Round the fractional digit using the digit to the right to determine the direction of the round. If the digit to the right is 6 or greater, round up. If the digit to the right is 5 or less, round down. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.23, 1.23 => 1.23.

ROUND_HALF_EVEN

Round the fractional digit up or down as needed to make it even. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.24, 1.245 => 1.24, 1.23 => 1.23.

ROUND_HALF_UP

Round the fractional digit using the digit to the right to determine the direction of the round. If the digit to the right is 5 or greater, round up. If the digit to the right is 4 or less, round down. For example, if Storage Precision is 2, 1.234 => 1.23, 1.236 => 1.24, 1.235 => 1.24, 1.23 => 1.23.

ROUND _UP

Round the fractional digit so the value moves away from zero. For example, if Storage Precision is 2, 1.234 => 1.24, 1.23 => 1.23, -1.234 => -1.24.

The *UOM Decimal* field controls the formatting of numbers that contain a decimal point. If no value is in the *UOM Decimal* field, a period (.) will be used to indicate a decimal point. If a value is in the *UOM Decimal* field, that value is used as the decimal point. This is useful for formatting numbers for countries that use a comma (,) as a decimal point.

The value of the *Currency Symbol* field is not used in the IBM TRIRIGA Application Platform. If the value of the Display Mask field specifies that numbers should be formatted with a currency symbol, it will always use a dollar sign ($) for the currency symbol.

The value in the *Display Mask* field may indicate that the digits of a number should be separated in to groups like this: 123,456,789. If no value is specified in the *UOM Delimiter* field, the character used to separate groups of digits is a comma (,). If a value is specified in the *UOM Delimiter* field, that value is used instead of a comma to separate groups of digits. This is useful for formatting numbers for countries that use a period (.) or a space to separate groups of digits.

Number formats

The value supplied for the Display Mask field of the UOM_Values form or the Number field form is used to determine the way that numbers with the unit of measure are formatted when displayed. The formatting in a Number field takes precedence over UOM formatting when displaying Number fields in query reports or forms. If a value has more digits to the right of the decimal place than shown in the Display Mask property, the platform uses Round Half Up to round the value in the display.

The value supplied for the Display Mask field is treated as a sequence of characters with each character having a particular meaning. Different characters have different meanings in a format. The following explains how formats work one character at a time.

The first character is 0 (zero). A zero is used to indicate a position that contains a digit. If a number has a format that consists of a sequence of zeros, it will be formatted to have at least that many digits. This is shown in the following table.

Table 9. Numbers formatted with 000

Value	Format	Formatted Value
0	000	000
3	000	003
45	000	045
678	000	678
12345	000	12345

Another character that has a special meaning in a number format is # (known as hash mark, number sign or octothorp). A # is used to indicate a position in a format that can contain a digit or a space. If a position that corresponds to a # would contain a leading zero, then it contains a space; otherwise it contains a digit. The following table shows some examples of numbers formatted with combinations of # and 0.

Table 10. Numbers formatted with # and 0

Value	Format	Formatted Value
0	##0	0
0	#00	00
0	###	
3	##0	3
45	##0	45
678	##0	678
12345	##0	12345

A number format may contain one period to indicate a decimal point in formatted numbers. The following table shows some example formats with a period.

Table 11. Numbers formatted with a decimal point

Value	Format	Formatted Value
0	##0.000	0.000
123.453	##0	123
123.453	##0.	123.
123.453	##0.0	123.5
123.453	##0.00	123.45
123.453	##0.000	123.453
123.453	##0.0000	123.4530

If a format does not allow enough digits after the decimal point to express a number exactly, the number is rounded to fit within the format. If there are more zeros after the decimal point in the format than are needed to represent a number exactly, the number is formatted with trailing zeros.

Use # after the decimal point to avoid trailing zeros. There are examples of this in the following table.

Table 12. Numbers formatted to avoid trailing zeros

Value	Format	Formatted Value
0	##0.0##	0.0
123	##0.0##	123.0
123.4	##0.0##	123.4
123.45	##0.0##	123.45
123.453	##0.0##	123.453
123.4538	##0.0##	123.454

A comma (,) can be used to specify that the digits in a formatted number should be divided into groups. The following table has examples of this.

Notice that if a number is longer than the format, the separation of digits into groups continues with groups the same size. The size of the groups can be three, four, or any other size that is desired. The last example in the table shows digits formatted into groups of four.

Table 13. Numbers formatted to group digits

Value	Format	Formatted Value
0	#,###,###	0
123	#,###,###	123
1234	#,###,###	1,234
123456	#,###,###	123,456
1234567	#,###,###	1,234,567
1234567890	#,###,###	1,234,567,890
1234567890	####,####	12,3456,7890

Use an E to indicate numbers should be formatted using scientific notation. The following table shows some examples of this.

Table 14. Numbers formatted to use scientific notation

Value	Format	Formatted Value
1234	0.###E0	1.234E3
0.00123	00.###E0	12.3E-4

Do not use a comma (,) and an E in the same number format.

Put a % (percent sign) or ? (per-mill sign) at the beginning or end of number pattern. These characters are displayed as themselves. A % has the affect of multiplying the formatted version of the number by 100. A ? has the affect of multiplying the formatted version of the number by 1,000. Examples of these are shown in the following table.

Table 15. Numbers formatted with percent and per-mill

Value	Format	Formatted Value
0.8432	0.####	0.8432
0.8432	##0.0#%	84.32%
0.8432	%##0.0#	%84.32
0.8432	##0.0#?	843.2 ?
0.8432	%##0.0#	?843.2

Do not use both a % (percent sign) and a ? (per-mill sign) in the same number format.

Use a ¤ (currency sign) at the beginning or end of a number format to indicate that a currency symbol should be in the formatted number. This is always $ (dollar sign). The following table shows some examples of this.

Table 16. Numbers formatted with currency symbol

Value	Format	Formatted Value
1234.5	¤#,##0.00	$1,234.50
1234.5	#,##0.00¤	1,234.50$

Characters that have no special meaning in a number format can be at the beginning or end of a number format. Such characters are just copied into formatted numbers. This is shown in the following table.

Table 17. Numbers formatted with non-special characters

Value	Format	Formatted Value
1234.5	**###0.00**	**1234.50**

Number formats can specify how negative numbers are to be formatted. If a number format does not specify how negative numbers will be formatted, then a negative number is formatted the same as a positive number with a minus sign before it.

You can explicitly specify a how negative numbers are to be formatted by following the number format with a ; (semicolon) followed by another number format that specifies formatting for negative numbers. This is shown in the following table.

Table 18. Numbers formatted with semicolon

Value	Format	Formatted Value
1234.5	#,##0.00	1,234.50
-1234.5	#,##0.00	-1,234.50
1234.5	#,##0.00;(#,##0.00)	1,234.50
-1234.5	#,##0.00;(#,##0.00)	(1,234.50)

Use a ' (single quotation mark) to indicate that the character following it should not be treated with any special meaning, for example, for literal symbols. The following table shows some examples of this.

Table 19. Numbers formatted with single quotation mark

Value	Format	Formatted Value
12	'#0	#12
12	#0 o''clock	12 o'clock

Note: The display mask for UOM values with type not set to *Currency* must be in the US English or Java-accepted format. For example, #.####.

Money

Units that are used to measure things like length, volume, and temperature have something in common. The conversion factors between units used to measure each of these things do not change. Units that are used to measure money are different.

The conversions between different currencies that countries use to measure money change over time. Hence, IBM TRIRIGA treats currencies differently from other kinds of measurements.

Evaluate your use of currencies during implementation and add or remove currencies pertinent to your company needs prior to adding data records. Failure to do so before creating records could cause conversion issues or data loss on those records.

The IBM TRIRIGA Application Platform can maintain conversion rates between different currencies that apply only to a range of time that you specify. To manage conversion rates for currencies, you use the Currency Conversion Manager.

To access the Currency Conversion Manager, navigate to Tools > Administration > Currency Conversions.

Each conversion rate is from one specific currency to another specific currency. Each conversion rate applies only to transactions that occur between a specified start time and end time.

The conversion rates you enter into the Currency Conversion Manager are used when the IBM TRIRIGA Application Platform generates financial transactions. Financial transactions are discussed in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

Conversion rates are also used when computing the value to automatically put in a base currency field. Base currency fields are discussed later.

Another use for conversion rates is with formulas. The conversion rates used in a formula are determined by the mapping properties of the business object that contains the field that the formula is for. Mapping properties for selecting conversion rates are discussed in "Base currency mapping properties". Formulas are discussed in "Formulas".

Manually entering conversion rates into the Currency Conversion Manager may not be convenient. It is common to automate the process by having another program feed conversion rates to the IBM TRIRIGA Application Platform by using IBM TRIRIGA Connector for Business Applications. For further information, see *IBM TRIRIGA Connector for Business Applications 3 Technical Specifications*.

Base currency fields

Some organizations need to track monetary amounts in different currencies. However, the organization's accounting practices may require that all monetary amounts be converted to a common currency that the organization uses internally for its accounting. This common currency used for accounting purposes is usually called the base currency.

The IBM TRIRIGA Application Platform has features to support the use of a base currency. The support for a base currency is accomplished using these features of the platform:

- Establish the currency that will be used as the base currency. Set this from the platform's administrative interface as one of its configuration properties. The platform is shipped with this value set to *US Dollars*. If you choose the default value of *US Dollars*, it must not be renamed or edited. To use a different name for the US currency, create a new UOM record that defines the new name.

- The name of the property that controls the base currency is *BaseCurrency*. It is in the TRIRIGAWEB.properties file. The details of how to use the platform's administrative interface to set the value of configuration properties are described in the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

- It is also possible for the value of the *BaseCurrency* property to be set at runtime. For more information, see *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

- For each number field with a UOM that may be a different currency from the base currency, create a corresponding base currency field. The base currency field will contain the amount in the main field converted to the base currency.

- Edit the mapping properties of the business object that contains the fields to be converted and their corresponding base currency fields. The details of how to edit these mapping properties are described in "Base currency mapping properties".

Creating base currency fields

Procedure

1. To create a base currency field, begin by editing the properties of the field to which the base currency field will correspond. Editing field properties is discussed in "Business object fields".

2. You can create a base currency field for a Number field only if the value of its *UOM List* property is *Currency*. When the value of a Number field's *UOM List* property is *Currency*, a property that is otherwise hidden appears under the *UOM List* property. The name of this property is *Create Base Field*.

3. The *Create Base Field* property contains a check box. If the check box in the *Create Base Field* property is checked, then when the properties for the field are saved a base currency field will be created for this field. The name of the base currency field will be the name of the corresponding field followed by *Base*.

   - For example, suppose that the name of a field is *CanadianCost* and we check its *Create Base Field* check box. When we save the field's properties, a base currency field named *CanadianCostBase* will be created if it did not already exist.

   - When someone enters a value into the *CanadianCost* field, that value is converted to the base currency. The converted value is stored in the *CanadianCostBase* field.

4. It is common not to include base currency fields in forms (discussed in "Form building") that are used to edit records. Usually the people entering data in currencies other than the base currency do not need to see the amount converted into the base currency and may find it confusing.

5. If you do include a base currency field in a form, it should be read-only. Allowing a base currency field to be set directly from a form defeats the fundamental purpose of the base currency field.

Base currency mapping properties

The platform can perform automatic currency conversions for base currency fields and for the results of a formula. To convert currencies, the platform must use a conversion rate in the Currency Conversion Manager (described in "Money").

To select a conversion rate, the platform needs to know more than just the currencies it is converting to and from. It also needs the name of a conversion group and a date. You provide the platform with this information through the mapping properties of the business object that contains the field that is the destination of the converted amount.

To access a business object's mapping properties, navigate to the Data Modeler. Select the business object. Go to the Data Modeler's *Tools* menu and click its *BO Mapping* menu item. Clicking the *BO Mapping* menu item causes the business object's mapping properties to appear in the Data Modeler's *Property* panel.

There are two mapping properties related to currency conversion. The value of the *Conversion Group* property is used to determine the conversion group that the conversion rate should be found in. This is done indirectly.

The value of the *Conversion Group* property is the name of a List field in the business object. The drop-down list in the *Conversion Group* property contains the names of the List fields in the business object. The value of the list field named by the *Conversion Group* property is treated as the name of the conversion group that the conversion rate must belong to.

To make this work, you will need to add a list to the List Manager that contains the names of the conversion groups that may be used for this purpose. The List Manager is described in "List management".

The other mapping property that is related to currency conversion is the *Exchange Date* property. The value of the *Exchange Date* property is used to determined the date for the conversion rate.

The value of the *Exchange Date* property is the name of a Date and Time field or the name of a Date field. The drop-down list in the *Exchange Date* property contains the names of the Date fields and Date and Time fields in the business object. The value of the field named by the *Exchange Date* property is the date that is used to select the conversion rate.

If these properties are not specified the platform will use the "Default" Conversion Group and the date/time that the record is saved as the Exchanged Date.

Formulas

Some kinds of fields can have their value determined by a simple formula. The simple formula mechanism works about the same way for each data type it can be used with.

There also is an extended formula mechanism. Extended formulas are more powerful and complicated than simple formulas. Extended formulas are described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*. The following paragraphs describe simple formulas.

A simple formula determines the value of a field from the values of other fields in the same record. For example, we may want to have a formula determine when a training session will end based on the time that it starts and its duration.

Some formulas may involve the use of a constant. For example, you may want the amount of money you budget for a certain kind of project to be 10% greater than the estimated amount. To do this, we need to multiply by 1.1. Simple formulas work with fields, not constants, so we cannot put 1.1 directly in a simple formula.

What you can do is add to the business object a read-only field that has a default value of 1.1. You would not normally include such a field in a user interface. Generally, creating fields that are constants is not a good idea since the system will have to store the value for each record that is created.

Now that you know what simple formulas can do, we are ready to look at the mechanics of specifying a simple formula. Though some details vary from one data type to another, the idea is that you select a field and then an operator and then a field until you have entered the entire formula. To make this clearer, we will work through the example of defining a field to contain a budget amount that is 10% more than an estimated amount.

Let us suppose a business object named *RepairProject* already has a field named *EstimatedCost* to contain the estimated cost of a repair. Also suppose that the business object has a read-only field named *ErrorMargin* whose value is 1.1. We want to define a field named *BudgetCost* that will contain the budget amount. Check the *Formula* check box and selected the Regular Formula Type.

To enter the formula, click the first *Find* hyperlink. Clicking the *Find* hyperlink causes a form to appear under the field's properties that allows us to select a field.

To select a field, first select the section that contains the field. If the field is not in a section, select the name *General*. Next, select the field. Following this procedure, we select the *General* section and then the *Error Margin* field. Finish by clicking the *Accept* action.

Operators are selected from the drop-down list to the right of the field we just found. Select * (indicating multiplication) from the drop-down list.

Repeat the process of selecting a field to select the *EstimatedCost* field.

When the value of a field is computed by a formula, you may want the field's unit of measure to be determined by the one of the fields that the formula uses. In this case, we want this field to have the same unit of measure as the *EstimatedCost* field; set the value of the *UOM Source Attribute* property to *Estimated Cost*.

Chapter 4. Life cycles

In addition to having data, a record has something associated with it called its state. A record's state is used to associate the record with a phase in its life cycle.

Human beings have a life cycle that consists of such phases as infant, toddler, child, adolescent, and adult. Records also have phases in their life cycle. The phases that a particular record goes through in its lifetime depend on the business object from which it was created.

For many records, the life cycle is very simple. They have only one phase that is typically called "new" or "created".

Some records have a more complicated life cycle. For example, a record that describes a book might use these states to describe the phases of its life cycle:

Table 20. Sample states of a book's life cycle

State	Description
Work in Progress	The record has this state while the book is being written.
Prepublication	The state of the record after the book has been written and while the book is being prepared for publication.
Published	The book has been published.
Abandoned	Publication of the book has been abandoned.
Out of Print	The book was published but is no longer being published.

The state of a record changes in response to an action. We will illustrate this by continuing the example of a business object that describes a book.

When the book is created, its state is *Work in Progress*. When the book is complete, the state changes to *Prepublication*. While the book is in prepublication, its release may be delayed for marketing purposes. When the book is printed, the state changes to *Published*. If the book is abandoned when its state is *Work in Progress* or *Prepublication,* its state becomes *Abandoned*. If the book is abandoned when its state is *Published*, its state becomes *Out of Print*.

Life cycle diagrams

It is difficult to understand the relationships between actions and states by reading a description like the preceding one. The relationships are usually much easier to understand when they appear in a diagram. The following figure shows a type of diagram called a *life cycle diagram* that illustrates the relationship between the states and events discussed in the preceding paragraph.
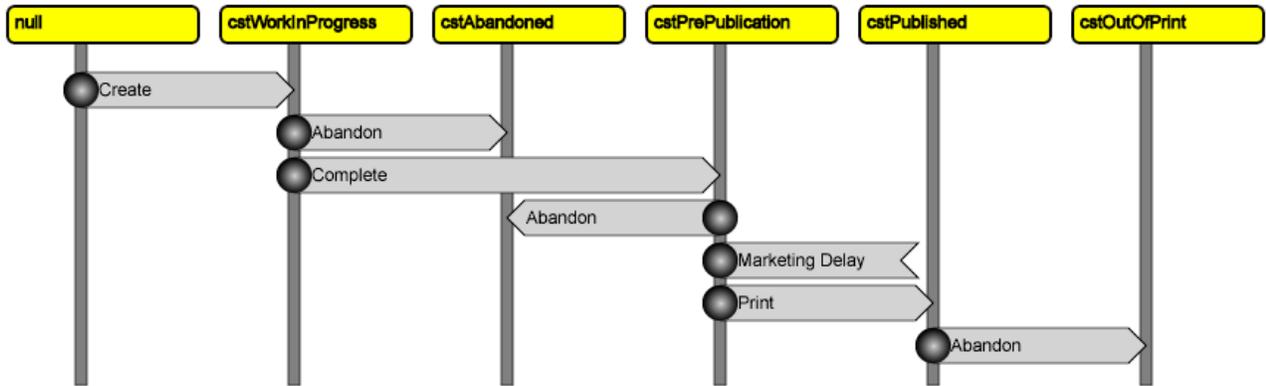
Figure 19. Sample life cycle diagram

The boxes in the figure represent the states of the record. The vertical lines under the boxes help show transitions between the states. The arrows between the lines represent transitions from one state to another; these transitions are called state transition actions. The words in the arrows are the labels users see for the station transition actions that cause the transitions.

When any state transition action is invoked, the system saves the record, regardless of the state transition action's name.

The odd shape that looks like the tail end of an arrow (such as the Marketing Delay transition in the figure) indicates a transition from a state to the same state. A state transition that does not go anywhere may seem pointless at first. The value of such a transition has to do with the fact that when a transition happens, it causes the record to be saved and may call a workflow and/or set the values of fields. Read more about workflows in "Overview of workflows".

Life cycle transitions

Every state in a life cycle has a set of state transition actions associated with it. When a record is being edited by a form, the state transition actions are included in the form's action buttons. When you see actions such as *Save* or *Save & Close* in a form's action buttons, it is because the record's current state has state transition actions that can be triggered by those action buttons.

State transition actions also can trigger events. See "Launch conditions" for more details.

Synchronous workflows can be triggered by a state transition action and asynchronous workflows can be triggered by the event generated in response to the state transition action. See "Synchronous versus asynchronous workflows" for a discussion.

The same state transition action name may have a different meaning depending on a record's state. In the previous figure, we see the action named *Abandon* meaning different things, depending on the current state of the business object. If the state is *Work in Progress* or *PrePublication*, the result of *Abandon* is that the state of the business object becomes *Abandoned*. If the current state of the business object is *Published*, the result of *Abandon* is that the state of the business object becomes *Out of Print*.

Example: Specifying state transition actions

The IBM TRIRIGA Application Platform uses state transition actions to model the life cycle of records created from a business object. Every business object has a defined set of state transition actions. Use the Data Modeler's *State Transitions* panel to specify a business object's state transition family. To access the *State Transitions* panel, click the *BO State Transition* menu item of the Data Modeler's *Tools* menu.

The *State Transitions* panel appears. For a new business object that does not yet have a state transition family, the *State Transitions* panel looks like the following figure.
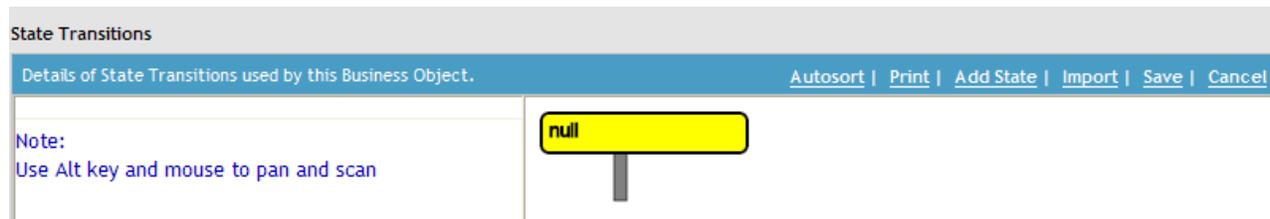
Figure 20. Initial state transition window

We will explain how to use the *State Transitions* panel by working through an example. The example is the creation of a state transition family for a *cstBook* business object that will model the life cycle shown in "Life cycle diagrams".

As you can see in the previous figure, when you start from nothing, a new business object's state transition family is created with a single state named *null*. The alternatives to starting from nothing are described in "State transition family reuse".

Null state

The *null* state has a special meaning. It has to do with the way records are created. A record is created in a special state named *null* to indicate that its fields have not yet been set to their proper values. While the record is in the *null* state, it does not fully exist. While the record is in the *null* state, queries will not find it and it is not stored in the database.

When the record transitions to a different state (not *null*), the transition tells the platform that the record's fields have been set to their proper values. When a record's state changes to something other than *null*, the record can be found by queries and is stored in the database.

The way to delete a record is to change its state to *null*.

A transition from *null* state to *null* state does not cause the control number to be calculated.

Example: Defining states and transitions

The first thing to do is add a state named *cstWorkInProgress*. To add a state, click the *Add State* action in the menu at the top of the State Transitions panel. The *State Transitions* panel now looks like the following figure.
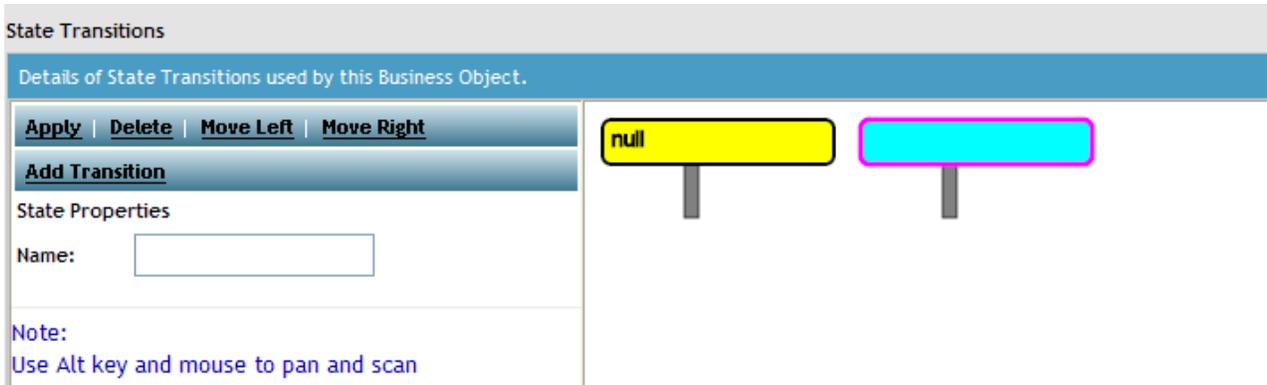
Figure 21. State Transitions panel with second state

The new state has no name. When new states are added, they automatically become the selected state. The form to edit the new state's properties appears in the left side of the panel. Set the name of the new state to *cstWorkInProgress*.

The next thing to do is to add a state transition action from the *null* state to the *cstWorkInProgress* state. To add this transition, first click the *null* state. Then click the *Add Transition* action in the left panel. Nothing happens right after you click the action. Then click the *cstWorkInProgress* state. After clicking the *cstWorkInProgress* state, a blank state transition action symbol appears from the *null* state to the *cstWorkInProgress* state. The *State Transitions* panel now looks like the following figure.
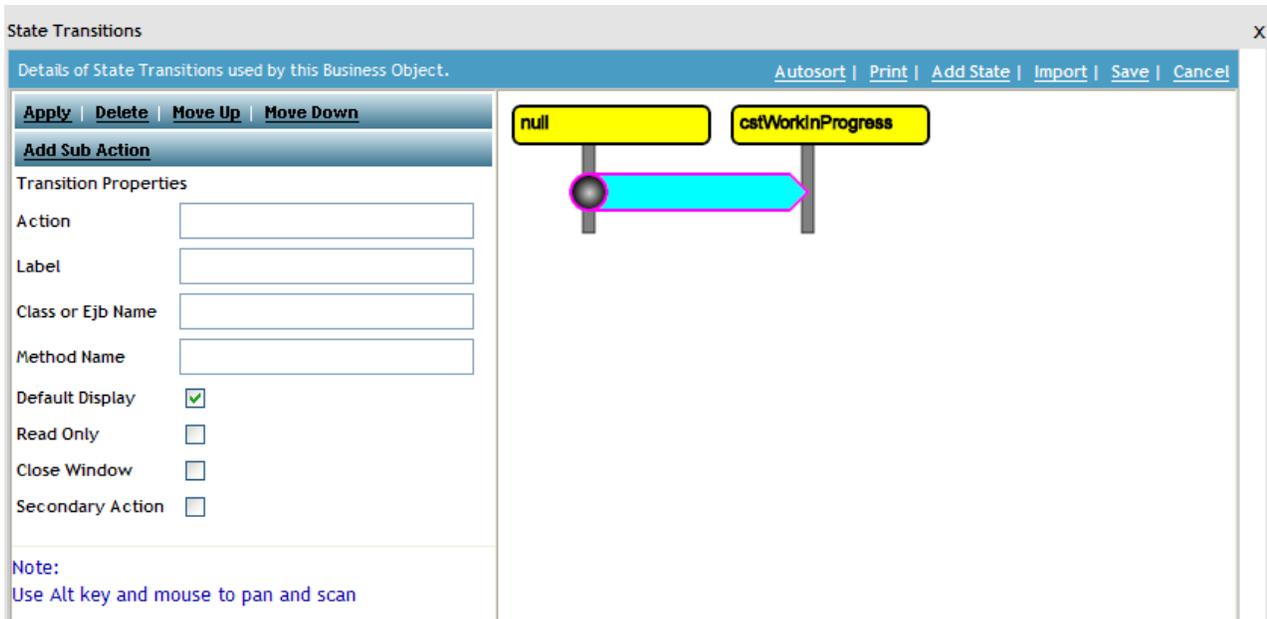
Figure 22. Work in Progress state and transition

Specify the properties of a state transition action with the following fields:

Action

> This is the name of the state transition action that will trigger this transition. The state transition action's name is used to identify the action from workflows.

Actions of the same name will trigger the same event. If you want the same labeled action to trigger different asynchronous workflows, use a different Action name in this field.

Use the naming conventions in "Naming conventions".

Label

This is the text that will be used to identify the state transition action in a menu that a user sees and in life cycle diagrams.

State transition action labels containing two or more words will not wrap/break between the words if the browser window is resized smaller. Instead, the wrap/break occurs between different labels.

Class or Ejb Name
This field is used to integrate a state transition action with a piece of the internal logic in the IBM TRIRIGA Application Platform. You should not put a value in this field unless specifically told to do so in the documentation or by your IBM TRIRIGA team.

Method Name
This field is used to integrate a state transition action with a piece of the internal logic in the IBM TRIRIGA Application Platform. You should not put a value in this field unless specifically told to do so in the documentation or by your IBM TRIRIGA team.

Default Display

The state transition family is part of the definition of a business object. When you define a form to edit records created from the business object, one of the things to specify is whether the state transition action to trigger this state transition will appear in the form's action buttons by default. As described in "Sub actions", a sub action can override this with its Inclusion Exclusion section.

The value of the Default Display property in the corresponding form overrides the value in the business object. See the discussion in "State-based actions". If you change the Default Display value in the business object after a form already exists, the Default Display property in the form still controls whether or not the state transition action displays to the user.

Read Only

When a state transition action is triggered, the record is read only if the state it is transitioning to is read only. To make a state read only, all of the state transition actions coming out of that state must have this check box checked.

At runtime, a state will be read only if all of the state transition actions visible on the form have the Read Only property checked.

Close Window
If this check box is checked, the window containing the record on which the state transition action appears closes when a user clicks the action button.

Secondary Action
If this check box is checked, the action is automatically put into the More menu for that form. If the check box is not checked, the action appears as a button on the form's menu bar. See "State-based actions" for more details on the effect of this property.

Fill in the fields shown in the previous figure, setting the *Action* field to *cstCreate* and the *Label* field to *Create*. Click the *Apply* action above *Transition Properties*. The name *Create* appears in the state transition action symbol. Click the *cstWorkInProgress* state. The life cycle diagram now looks like the following figure.
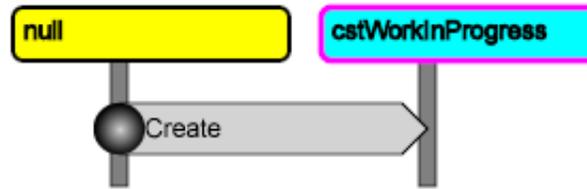
Figure 23. Life cycle diagram with new state added

Continue adding the rest of the states and state transition actions. Click the *Add State* action. This adds a new state. Set the name of the state to *cstAbandoned*. Then create a state transition action from the *cstWorkInProgress* state to the *cstAbandoned* state. Label this new state transition action *Abandon* and have it trigger an action named *cstAbandon*. At this point, the life cycle diagram looks like the following figure.
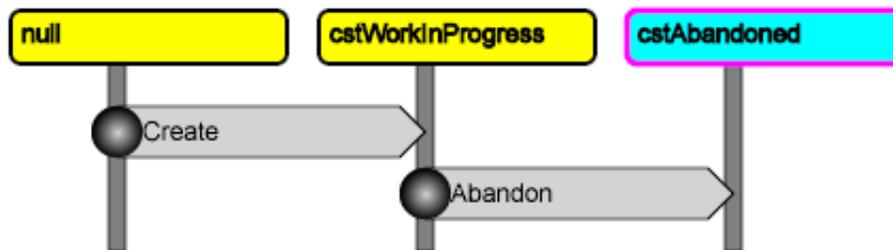
Figure 24. Life cycle diagram with abandoned state added

Build the rest of the state transition family. The completed life cycle diagram looks like the figure in "Life cycle diagrams".

When you are finished entering the states and state transition actions, click Save to save the diagram. Click Cancel to close the state transition diagram.

The following reviews the remaining actions in the upper right corner:

Print   This action allows you to print the contents of the state transition diagram.

Autosort

Prior to IBM TRIRIGA Application Platform 3.0, whenever you saved a state transition diagram, a built in algorithm rearranged the states for you. Unfortunately this algorithm did not let you use your own judgment in how to best place the states for optimal readability. Starting in IBM TRIRIGA Application Platform 3.0, the states retain their ordering when you click Save, and the previous algorithm is tied to the Autosort action. Note that the Autosort action does not commit to the database, so if you click Autosort and do not like the outcome, you can simply Cancel and reopen the diagram to restore the original ordering.

Import

This is described in "State transition family reuse". Note that, unlike Autosort, the system saves the results of this action to the database.

Life cycle diagram manipulation

A life cycle diagram can grow wider and/or taller than the Data Modeler window.

The *State Transitions* panel has no scroll bars. To see a different part of the diagram, drag it by putting the mouse over any part of the diagram.

State family editor tips

The state family editor in the Data Modeler is very sensitive to the order in which steps are taken when adding new states, state transition actions, sub actions, and the like.

The editor will not allow you to create a technically invalid state family. If an invalid state transition action is added for example, when you attempt to apply the changes, the editor will roll back the state family to its last saved state (or back to a blank state under some conditions). However, the editor usually does not inform you that a rule was violated; it just performs the roll back and your changes are lost.

The following tips, if followed, will ensure you do not lose changes that have been made when editing a state family:

- There are two actions that save the state of the diagram. The Apply on the left side saves individual states and state transition actions. The Save on the upper right saves the whole state transition family.
- You only can Save a valid state family (see definition below). If you attempt to Save an invalid state family, the editor reverts the state family to its last saved state.
- A valid state family consists of the following:
  - At least two states, the first of which is named null. Remember that null needs to be lower case.
  - At least one transition action between the two states.
  - Properly named states and transitions (i.e., no spaces or special characters in the names).
  - No "island" states, except in state family templates.
- When adding a sub action to a transition action, make sure the state family has been saved prior to adding the sub action if any state or transition actions have been newly created or modified.

State transition family reuse

In the preceding example, we built a state transition family from nothing. It is unusual to build a state transition family from nothing. Usually you will start out with a copy an existing state transition family. There are three ways to begin with an existing state transition family:

- If you are defining a business object that is not its module's base business object, the business object automatically starts out with a copy of the base business object's state transition family.
- If you want to start out with a copy of a different business object's state transition family, click the State Transition panel's Import action and select the *Import from Business Object* menu item.

- The IBM TRIRIGA Application Platform is shipped with a collection of generally useful state transition families that are not part of a business object. If you want to use one of these, click the *Import from State Family* menu item in the *State Transitions* panel's Import menu.

Importing a state transition family from another business object or from the State Family Manager will merge the states and transitions of the imported state transition family with the states and transitions already in the business object's state transition family.

Sub actions

It is possible to add additional information to any of the transitions to specify how they should behave with records created from this business object. These are called sub actions and are a primary method of triggering a workflow.

Transitions that have not had any information added are a single shade of gray, as shown in the following figure.
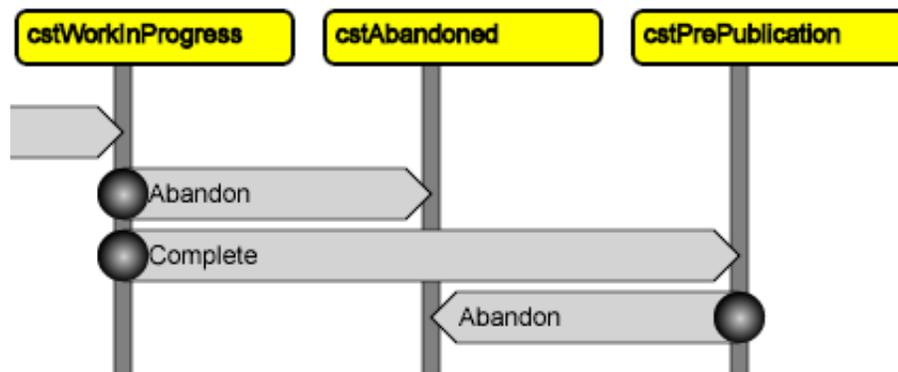
Figure 25. Transitions with no business-object-specific information

To add business object-specific information to a state transition action, first click the state transition action to select it. Then click the *Add Sub Action* action in the left panel. The Add Sub Action action causes a SubAction Label Details window to pop up. The window has three fields:

Label

This is the label text that will display in menus the user sees for the action that will trigger the state transition action. If just a single sub action is defined for a state transition action, this label text will be used instead of the label specified for the state transition action in the state transition family. If multiple sub actions are defined for a transition, they will appear in a drop-down menu under the action when the user clicks the action.

If a workflow triggers a state transition action, and that action has multiple sub actions, only the first sub action will be run, where the first sub action is determined by listing the sub action labels alphabetically.

State action labels containing two or more words will not wrap/break between the words if the browser window is resized smaller. Instead, the wrap/break occurs between different action labels.

Workflow Select

Use this field to associate a synchronous workflow with this state transition action. Workflows are discussed in "Overview of workflows" and "Workflow building". The specified workflow is launched when the state transition action is triggered.

Audit Action

If actions for the business object are being audited, this check box controls whether or not this particular state transition action will be audited.

The information entered into the window is saved when you click the *Apply* or *Ok* action. If you click the *Apply* action, you continue working in the window. After you click the *Apply* action (or click the *Ok* action and then edit the transition information) the window has more fields. After the initial business object-specific information has been saved, the SubAction Label window has two additional sections.

An action is visible on a record's form by default if the action's Default Display property is selected. The *Inclusion Exclusion* section provides the ability to toggle a given action's visibility. This section lists all state transition actions from the state that this transition comes from. Because the list shows the actions from the state from which the action originates, this function is most useful on a sub action that returns to the same state. If not returning to the same state, make sure the action for which you choose the Inclusion or Exclusion exists in the state to which the action is transitioning.

An action will be included on a form by default if the Default Display property is selected. At runtime you can override the Default Display value with the Include Exclude section. This override action occurs at runtime for each individual record. If the intent is to remove an action from the form when the user triggers the action, set the Exclusion radio button for the action(s) that you wish to exclude. If the intent is to include an action on the form when the user clicks this action, set the Inclusion radio button for each action you wish displayed.

If an action is included in a menu, the label text specified in the *Label* field is displayed in the menu to represent the action.

State transition actions that are not displayed in menus are called *hidden actions*. Hidden actions have their Default Display property not selected and typically use the term "hidden" in their action name and label.

Some actions are hidden by default but may be displayed by the use of an Inclusion set in a sub action. Some actions may be set to display (or not) by default but are displayed (or not) by inclusions or exclusions on the sub action.

For example, the state transition family for triContractLineItem in triCostItem, shown in the following figure, reflects a standard pattern used in the platform. In this Edit Hidden example, the ability to edit a record depends on the parent record. When it goes to an editable state, it triggers Edit Hidden, which changes the sub actions to those with Inclusion set (Save, Save & Close, Delete, and Copy).
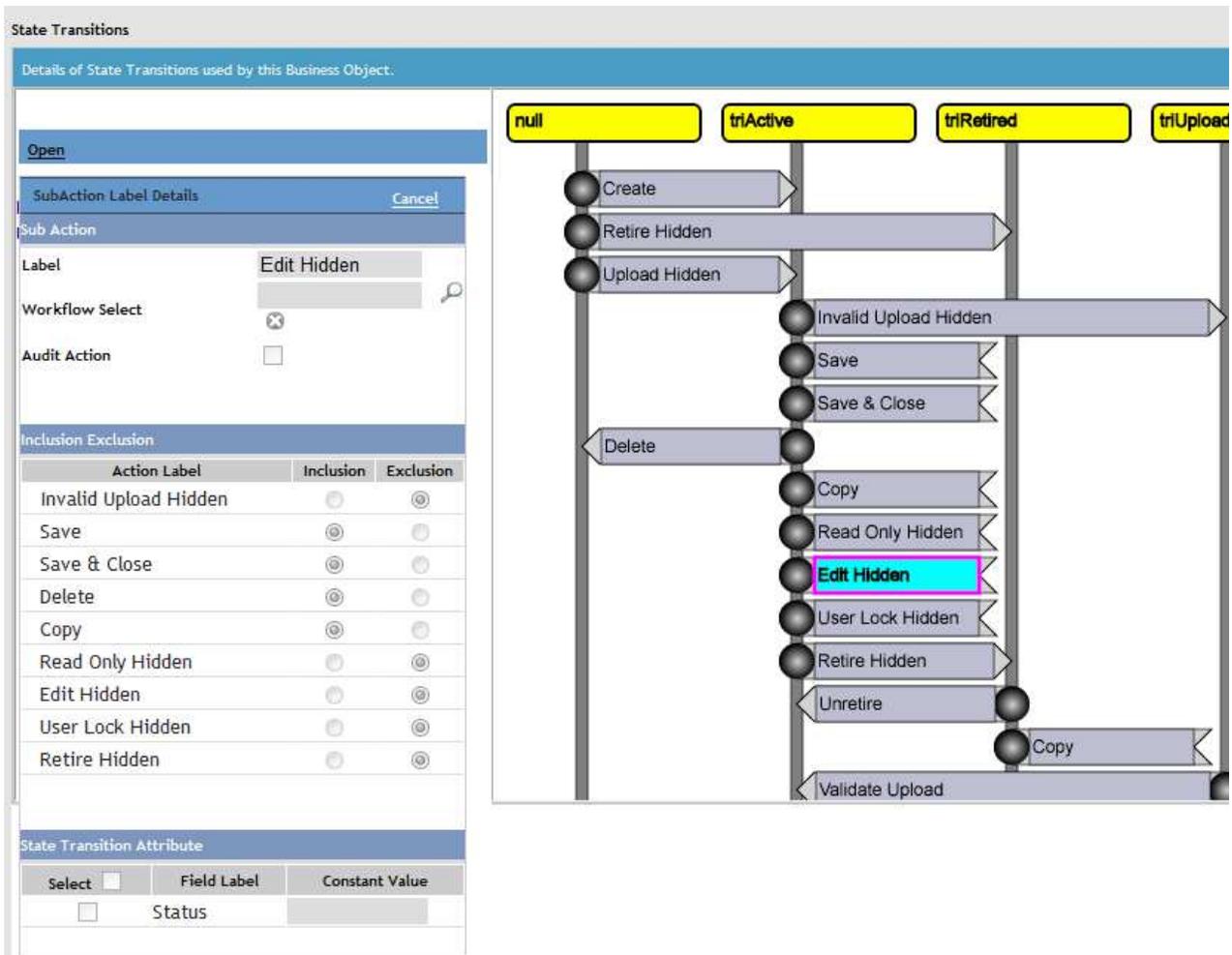
Figure 26. State Transitions panel for hidden sub action

Hidden actions cannot be initiated from a menu. However, a hidden action can be initiated from a workflow. Actions in general are discussed in more detail in "Actions".

Use the *State Transition Attribute* section to specify that a record's field should be set to a value when the state transition action is triggered. A typical scenario where you might use this would be to set status values for triStatusCL and triPreviousStatusCL as a record moves from one state to another.

To specify that a field should be set to a particular value when the state transition action is triggered, click the *Add* action. Clicking the *Add* action causes a State Transition Attribute window to pop up.

The State Transition Attribute window contains the names of all the fields in the business object. There is a check box to the left of each name. Select a check box to indicate that the state transition action should set a value in the field when it is triggered. After clicking *Ok*, the State Transition Attribute window goes away and the selected field(s) appear in the State Transition Attribute section of the SubAction Label Details window.

To specify the value to put in the field when this state transition action is triggered, enter the value in the *Constant Value* column.

In order for this to work in classification fields, the Constant Value must be one of the statuses in the classification hierarchy.

If you want the state transition action to set the value of more fields, add more fields to the *State Transition Attribute* section.

When finished entering business object-specific information, click the *Ok* action on the SubAction Label Details window. The SubAction Label Details window disappears.

After business object-specific information for a state transition action has been saved, the appearance of the state transition action in the life cycle diagram changes. State transition actions that have business object-specific information associated with them are a different shade of gray.

Chapter 5. Overview of workflows

The IBM TRIRIGA Application Platform allows you to add your business logic to your applications by creating a *workflow*. Workflows can be created to define any business process associated with the system or the business objects created in the system. A number of pre-defined workflows are delivered with the IBM TRIRIGA applications. This overview covers the major concepts involved in creating workflows. There is some discussion about how to connect workflows to other parts of an application. Most other details about workflows are discussed in "Workflow building".

Common uses for workflows

The first thing you should understand about workflows is what they are used for. It is not possible to present you with a complete list, because there is an almost limitless range of things that can be done. Here are some common uses for workflows:

- A workflow can be used to set the values in the fields of a new record. Formulas (discussed in "Life cycles") are a simpler and easier way to initialize the values of fields, but there are computations a workflow can do that a formula cannot. There is no guarantee of the order in which formulas are computed. If you need initial values for fields computed in a certain order, you must use a workflow.
- You can use workflows to vary the appearance of a form based on the contents of underlying records or a user's actions.
- You can use workflows to validate the contents of a form before saving its contents.
- You can use workflows to perform computations to set the values of fields in a record. You can also use workflows to perform computations that affect just the fields in a form.
- You can use workflows to create or manipulate records without requiring any interaction with a person.
- You can use workflows to route work to people. This can be done entirely within the IBM TRIRIGA Application Platform by having workflows put action items in a person's portal. (Portals are described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.) Workflows can also route work to people working outside the platform by copying data from records into a spreadsheet and e-mailing it to someone. A workflow can then receive an e-mailed spreadsheet that a person has worked on and copy its data into records.

Synchronous versus asynchronous workflows

One of the key properties of a workflow is whether it is *synchronous* or *asynchronous*. If a workflow is synchronous, it means that if the workflow is launched in response to something that a user did in the user interface, the workflow will begin executing immediately; the user has to wait for the workflow to finish before doing anything else.

If a workflow is asynchronous, it means that the workflow is launched in response to an event that has occurred in the system. When it launches depends on its

position in the workflow queue. For example, if the workflow is launched in response to an event that a user triggered in the user interface, the workflow may not start executing immediately; the user can do something else immediately without waiting for the workflow to finish.

Whether a workflow is synchronous or asynchronous is determined by the value of its *Concurrence* property (described in "Start task"). In addition to synchronous and asynchronous, the Concurrence property includes a value named subflow. A subflow workflow is a special type of synchronous workflow that allows required parameters. A subflow is special because the only way it can be used is from a Call Workflow task.

Although workflows marked asynchronous are run asynchronously (from the event that triggered it), workflows that are marked synchronous can be run a number of ways. Synchronous workflows are run in-line with whatever caused it to start, and the process causing it will wait until the workflow has finished before it continues. In the case of a form action causing the workflow to run, it means that control will not be returned to the user until the workflow completes.

A workflow that is marked synchronous can also be called. If it is called from an asynchronous workflow, it will be run as part of that asynchronous process. So workflows that are marked asynchronous are pretty clear, but synchronous really means "run in-line with the process invoking it".

Launch conditions

There are a number of ways a workflow can be launched. The ways that a workflow can be launched are determined by whether it is synchronous, subflow, or asynchronous. The distinction between synchronous and asynchronous is explained in "Start task".

Asynchronous workflows are launched in response to an event occurring on a business object. A workflow is registered for a given event/business object combination using the Event property in the workflow Start task. There are two types of events: events generated by state transition actions, and system events, such as user sign in, scheduled event start, or a DataConnect Job being run.

The workflow Event property and the Start task are discussed in "Start task". State transition actions are discussed in "Life cycles". DataConnect is discussed in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

Asynchronous workflows are not directly connected to a state transition action. Instead, they may be connected to events that correspond to the name of the state transition actions. When a state transition action is triggered on a record an event of the same name is registered in the system event queue for that record. If an asynchronous workflow exists that corresponds to the business object of the record and event in the queue, the system launches the workflow.

The following diagram illustrates the order in which things occur when state transition actions that invoke workflows are triggered.
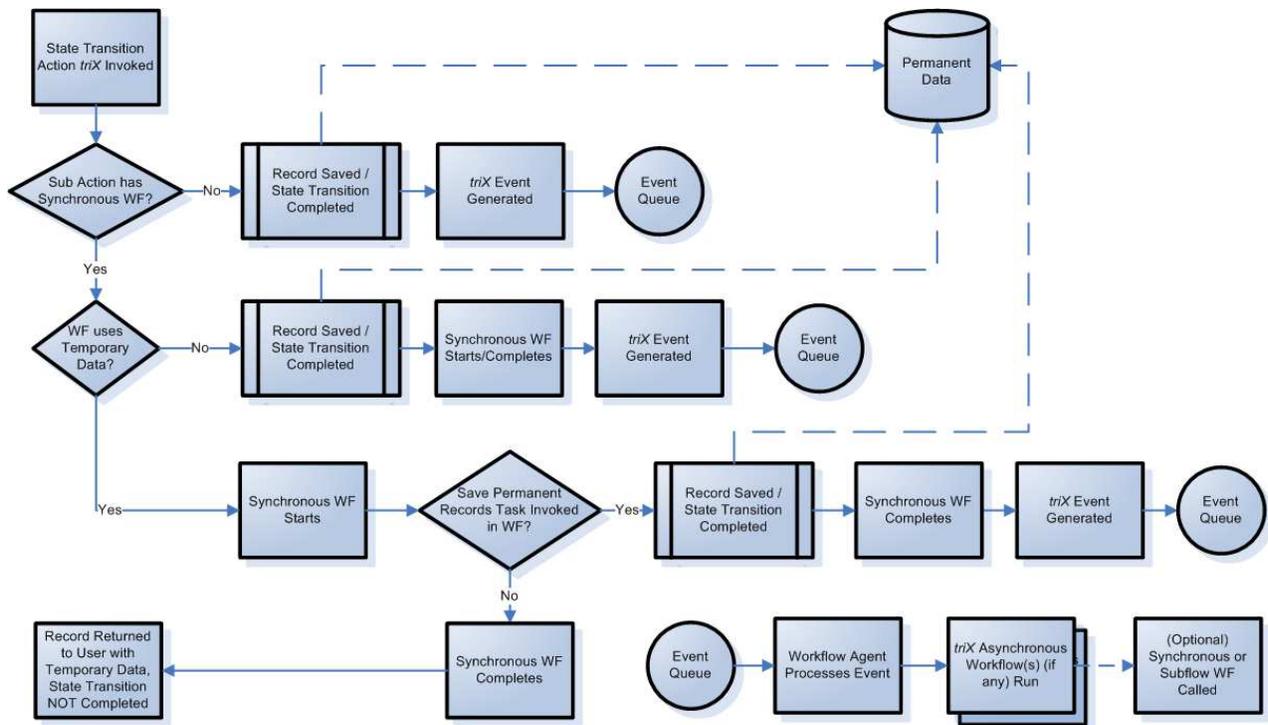
Figure 27. When state transitions that invoke workflows are triggered

An asynchronous workflow also can be launched by a system event. System events are things that happen to records that may not be the direct result of an action by a user. For example, when a record becomes de-associated from another record there is a system event. System events are discussed more fully in "System events that trigger workflows".

These are the ways a synchronous workflow can be launched:

- A synchronous workflow can be launched from a sub action attached to a state transition action. Sub actions are discussed in "Sub actions".
- A synchronous workflow can be launched from an action a user performed on a part of a form. Form actions include actions that appear at the top of a section, actions that are triggered when a button is pressed, and actions that are triggered when the value of a field changes. Form actions are discussed in "Form building".
- A synchronous workflow can be launched when a record is created. The way to specify that a workflow should be launched in this way is by the business object's Pre-Create Workflow property, which is described in "Viewing business object properties".
- A synchronous workflow can be launched prior to popping up a form to edit a record. The way to specify that a workflow should be launched in this way is described in "Form properties".
- A synchronous workflow can be launched from another workflow by using a Call Workflow task. Call Workflow tasks are described in "Call workflow task".
- A synchronous workflow can be launched from certain navigation items. See the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.
- A synchronous workflow can be launched from a report or query. Configuring actions to be launched from a report or query is discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

You can make the launch of a workflow conditional on such things as the contents of its record, the state of the record, or the current time. These conditions that you can impose on the launch of a workflow are called the workflow's start conditions. The details of how to do this are discussed in "Start task".

Launching more than one way

Although there are a number of ways that each workflow can be launched, the IBM TRIRIGA Application Platform allows each workflow to be launched only in one way. If you want a workflow to be launched in more than one way, there is a workaround to this limitation.

If you want to launch a workflow in two different ways, write two other workflows that will be launched in each of the ways that you want the real workflow to be launched. All that these other workflows need to do is call the desired workflow.

Workflow task organization

Workflows are organized into simple pieces called tasks. There are different kinds of workflow tasks. Each kind of workflow task does something different. For example, there are workflow tasks to create a record, change the contents of a record and create associations between records.

Workflow tasks are described in "Workflow building".

By arranging different types of workflow tasks into a suitable order and specifying the correct properties for each task, you can create a workflow that is capable of performing whatever business logic is needed.

Each kind of workflow task has a properties form associated with it that allows you to specify details of what the task will do. These properties typically include as how to find the record(s) that the task will work with, computations it should perform with the found record(s), or how it should modify the contents of the record(s).

There are a few fundamental techniques for arranging the order of workflow tasks. The most basic technique is to organize tasks into a simple sequence where after each task there is one other task that will be the next task performed.

Sometimes you want more than one task that can immediately follow a particular task. There are two kinds of workflow tasks you can use this way.
- A Switch workflow task causes one of two sequences of tasks to be performed, based on whether a condition is true or false. Switch workflow tasks are described in "Switch task".
- A Fork workflow task defines two or more sequences that can be performed at the same time. Fork workflow tasks are described in "Fork task".

Sometimes you will want a sequence of tasks to be performed multiple times. There are three kinds of workflow tasks you can use to cause a sequence of tasks to be repeated multiple times.
- The Loop task is a general purpose task for repeating a sequence of tasks for any reason. Loop tasks are described in "Loop task".

- The Iterator task is a simpler and more specialized kind of loop. It is used to repeat a sequence of tasks once for each record associated with a preceding task. Iterator tasks are described in "Iterator task".

- A DataConnect workflow task gets a set of records from the designated staging table and acts as an iterator to create or update a record for each row in the staging table and to run the body of the task for each row. DataConnect workflow tasks are described in "DataConnect task".

Sometimes you want to reuse some business logic or have some business logic be shared by workflows. You can organize workflows to do these things by having one workflow call another workflow. There are two kinds of workflow tasks available for this purpose:

- Workflows can use the Call Workflow task to launch synchronous workflows. The Call Workflow task is discussed in "Call workflow task".

- The Trigger Action workflow task triggers actions which in turn cause events that can cause asynchronous workflows to be launched. The Trigger Action workflow task is discussed in "Trigger action task".

Temporary versus permanent data

The data that workflows normally deal with is considered to be permanent data. Permanent data is the values in records that are kept in the database indefinitely. While a record is being edited with a form, a temporary version of a record's permanent data is used. As a person edits the fields of a form, the temporary data is updated. The permanent data is not affected by the edits until the edits are saved.

A workflow can access a record's temporary data if the workflow is synchronous and the value of the workflow's *Temporary Data* property is *Temporary*. A workflow can use a Get Temp Record task to access a record's temporary data. The Get Temp Record task is described in "Get temp record task".

Once a workflow has access to a record's temporary data, it can modify the temporary data. Some other possibilities are for the workflow to perform calculations based on the temporary data or to change the appearance of the form based on the values in the temporary data.

A workflow uses a Modify Metadata task to modify the appearance of a form. The Modify Metadata task is described in "Modify metadata task".

A workflow can use its ability to change the appearance of a form to tell a person about any problems with data entered into the form. Also, if there are no problems with data, the workflow can save the temporary data. A Save Permanent task updates the contents of a record with the values in its temporary data. The Save Permanent task is described in "Save permanent record task".

Start conditions, temporary data, and workflows

Sometimes you will have a synchronous workflow that is launched from an action or button on a form whose job it is to save temporary data to permanent. Such buttons or actions typically have a label such as Save or Ok.

When people click an action with such a label, they expect that the data will be saved. There may be legitimate reasons for the workflow not to save the data. For example, some fields may have inconsistent values. If such a workflow decides not

to save data, it should do something to inform the user that the data was not saved. It should never be a surprise that the data was not saved.

For these reasons, putting start conditions on this sort of workflow is a bad thing to do in this situation. If the workflow does not run because its start conditions are not satisfied, a user will have no way of knowing. To handle this, let the workflow run, have it detect a reason it should not save the data and then set a visible status indicator so that the user will be aware of the problem and that the data was not saved.

System events that trigger workflows

As mentioned earlier, one of the ways that an asynchronous workflow can be launched is for a system event to happen to a record. When a system event happens to a record, if there are any workflows associated with the system event and the business object that was used to create the record, then those workflows are launched.

Here is a list of the different kinds of system events and an explanation of what they mean:

Associate

An *Associate* system event happens to a record when it becomes associated with another record. When a workflow is launched as a result of an Associate system event, the other record that the record is becoming associated with is accessible from the workflow's tasks as the *Secondary BO* of the workflow's Start task.

De-Associate

A *De-Associate* system event happens to a record when its association with a record is removed. When a workflow is launched as a result of a De-Associate system event happening to a record, the record that was on the other end of the association is accessible from the workflow's tasks as the *Secondary BO* of the workflow's Start task.

There are two other situations in which a *De-Associate* system event happens to a record:

- If a record is removed from a smart section, that triggers a *De-Associate* system event even though its association with the record that contains the section still exists.

- An association may indicate that a record's existence is dependant on another record. If a record is deleted because a record it depends on deleted, a *De-Associate* system event happens to the dependant record.

SCHEVENTCREATE

A *SCHEVENTCREATE* system event happens to a record when a time-based event is scheduled to happen to the record. Time-based events are discussed in "Calendar and time-based events".

SCHEVENTEND

A *SCHEVENTEND* system event happens to a record at the end of a time based event that happens to a record. Time-based events are discussed in "Calendar and time-based events".

SCHEVENTSTART

A *SCHEVENTSTART* system event happens to a record at the start of a

time-based event that happens to a record. Time-based events are discussed in "Calendar and time-based events".

SYSTEM DC PROCESS JOB

This event is used by the DataConnect Agent when a DataConnect Job is ready to run. The agent triggers the workflow associated to the corresponding DataConnect Job record and the SYSTEM DC PROCESS JOB event.

WF Q Accept

A *WF Q Accept* system event happens to a record when a recipient accepts an action item associated with the record.

WF DELETE FROM MGR

A *DELETE FROM MGR* system event happens to a record when it is deleted using a system delete from a navigation item or a report action. Navigation items are discussed in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*. Reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

WF USER LOGIN

For every login ID in the IBM TRIRIGA Application Platform, there is a corresponding *My Profile* record to describe the login ID. When a person signs in, a *WF USER LOGIN* system event happens to the person's *My Profile* record.

WF USER LOGOUT

For every login ID in the IBM TRIRIGA Application Platform, there is a corresponding *My Profile* record to describe the login ID. When a person signs out, a *WF USER LOGOUT* system event happens to the person's *My Profile* record.

WF WIZARD CANCEL

A *WF WIZARD CANCEL* system event happens to a record when a user editing the record closes the form. This event triggers an asynchronous workflow on the *Cancel* or *X* action of a record. However, this event will not fire on a new record in the null state.

Transaction scope and processing

In a workflow, most task steps are processed in their own scope. If there are no problems affecting successful processing, a task step performs its database operations and the workflow's processing moves to the next task step and that task step processes as a unit. This continues through the entire workflow and provides data integrity within the task step, but it does not allow transactions that span task steps. Note that for Iterator and Loop tasks there are no transactions that span any or all of the iterations of the tasks within a given Iterator or Loop. Also, for tasks such as Associate Records or a task that works on a set of records and causes changes, where there may be more than one record modified by a single task step, each record that is modified is processed individually within its own transaction.

The DataConnect, Create Record, and Modify Records tasks can be configured to use transactions during the processing of the task. The DataConnect task includes a Transaction parameter that is used to enable transactions. Within a DataConnect task, the Transaction parameter determines how the business object record is committed. The Transaction parameter values are part of the description of the DataConnect workflow task in "DataConnect task".

When nested DataConnect tasks exist, the outermost task is in control of the transaction for that task and all nested tasks, regardless of how the nested tasks are configured. That outermost task is known as the Controlling Block. Transaction setting is valid only on the outermost DataConnect task with a Transaction setting (All Iterations, Per X Iterations). If the Transaction parameter is set to None, the DataConnect task processes the same as other workflow tasks.

The Create Record and Modify Records workflow tasks, like the DataConnect task, have Transaction properties available that can be used to enable transactions. Within a Create Record task or Modify Records task, the Transaction parameter determines how the business object record is created. The Transaction parameter values are part of the description of the Create Record task in "Create record task" and of the Modify Records task in "Modify records task".

When the Transaction parameter is set to None, the DataConnect, Create Record, or Modify Records task processes the same as other workflow tasks. It is only when the Transaction parameter has another value that a transaction is used across multiple tasks (DataConnect) or multiple records (Create Record and Modify Records).

A transaction is committed if everything processes correctly within the controlling block and is rolled back if an error that would compromise data integrity is encountered.

There are two ways a transaction can be committed:
*   The controlling block is exited successfully.
*   A looping type controlling block successfully reaches the end of an iteration that meets the transaction criteria (number of iterations) configured for the transaction block.

A block is successfully exited if it:
*   Meets the exit criteria without encountering a fatal error or a processing condition (task) that rolls back the transaction.
*   Encounters an End task within the block before encountering a condition causes a rollback.
*   Breaks out of a controlling block with a state of Success.

A transaction is rolled back if the controlling block does not complete successfully. The following cause unsuccessful completion of a block:
*   Encountering a non-recoverable error within the block (this includes non-recoverable errors anywhere within the block or anywhere within nested blocks).
*   Encountering a Stop task within the block (this includes Stop tasks within nested blocks).
*   Breaking out of, or Continuing, a controlling block with state of Error.

Variables, parameters, and return values

Task steps within a workflow can refer to other task steps for data values to be used in processing. Workflow variables provide another way to access these data values. Once a workflow variable is assigned a value from a task step another task step can reference the variable and use the data just as it would have referenced

the task step. Referring to a variable rather than directly to a task step can sometimes allow for better workflow structure and a reduction in repeated logic.

The following figure shows a simplistic example of using a variable.
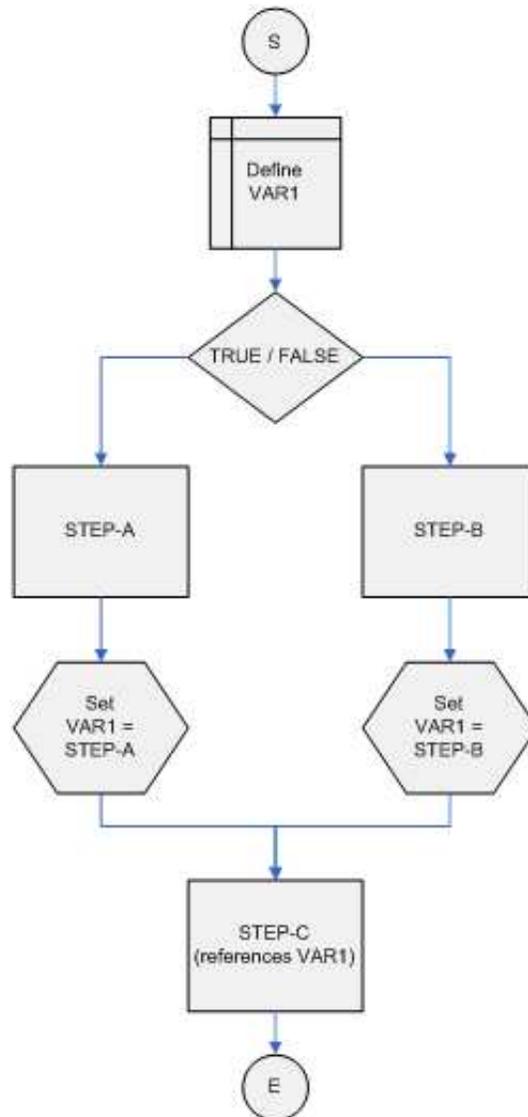
Figure 28. Simple example using a variable

In this example, before the condition step, a workflow variable VAR1 is defined. After STEP-A and STEP-B, variable VAR1 is set to be a copy of their result. STEP-C is defined to reference variable VAR1 rather than the actual task step, allowing STEP-C to exist a single time within the workflow while being able to use the result of either STEP-A or STEP-B depending on which was run.

In many business processes there is logic that is the same or very similar. To avoid duplicating the steps needed to perform the logic in each of the workflows that make up the process, those common steps can be broken out into a shared workflow that each of the others can call. Duplicating common logic makes it harder to maintain the processing logic and can lead to subtle differences in the way two similar business processes behave.

To allow these workflows to have access to all of the information needed they can be defined to accept one or more parameters (as workflow variables). They can also be defined to return one or more results to the calling workflow (as workflow variables). This makes it possible to break out common business logic into a shared workflow that can be called with a set of workflow variable parameters that have been set within the calling workflow. Upon return from the called workflow, a set of workflow variables can be set containing the return information. The calling workflow can then access the returned values as variables.

Workflow parameters and return values are one or more workflow variables that can be selected in the Start task to identify them as being parameters and/or return values. Asynchronous workflows cannot use parameters and return values. Synchronous workflows allow optional parameters while Subflow workflows allow either optional or required parameters.

More information can be found in "Start task", "Call workflow task", "Variable definition task", and "Variable assignment task".

Chapter 6. Form building

The IBM TRIRIGA Application Platform provides a tool named Form Builder for defining forms to create, view, and edit the contents of records. To get to the Form Builder, navigate to Tools > Builder Tools > Form Builder.

Each form is associated with a business object. The Form Builder organizes forms by the module that contains their associated business object. The left side of the Form Builder is a list of modules with a radio button next to each module. Selecting a radio button selects the corresponding module. The right side of the Form Builder lists the forms associated with the selected module.

The list of forms on the right side of the Form Builder is organized into these columns:

Name This is the name that is used in workflows to uniquely identify the form. This name is unique within a module.

Label This is the string that identifies the form in the user interface.

Business Object
The name of the business object the form is associated with.

Status This is the status of the form. For an explanation, see "Create-Publish-Revise cycle".

Object Label
This is the current object label identifier for the form.

The actions that appear at the top of the Form Builder when it first appears are always displayed at the top of the Form Builder:

Style Sheet Editor
Clicking this action pops up the Style Sheet Editor, which is a tool for editing style attributes that may be common to many forms. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Alternate Forms

You use the Alternate Forms action to specify an alternate form for a record when that record is opened in a query section, graphics section, locator field, availability section, or calendar section. At runtime, if an alternate form has been defined for a module, business object, and form, the record displays in the alternate form.

Clicking the Alternate Form action opens the Alternate Form List Manager, which is different from the List Manager. You use the Alternate Form List Manager to identify alternate forms or to delete them. To identify alternate forms, click the New action. In each row in the Alternate Form List, define the module, business object, form, and alternate form.

After the list is created, you can use it in a query section, graphics section, availability section, calendar section, or locator form field.

Do not use alternate forms for add or create functionality. Only use forms in the Alternate Form List framework to open records after they have been created.

At runtime, if an Alternate Form List is defined for an operation and more than one row in the Alternate Form List satisfies the record, the most specific is used. For example, if there is an alternate form for the record's form and a different alternate form for the record's module, the specific form alternate is used.

Where Used

If you select the radio button for a form and then click this action, a window pops up showing what references or uses the selected form. The fields displayed are: Name, Type (e.g., Query, Workflow, Nav Item), Module, Object, Form, Action, and Additional Information.

If you want to export the information in the Where Used window, click the Export Usage action at the top of the window. You will be offered the choice of saving or opening a csv file.

See "Deleting fields" for a list of references reported by Where Used.

New Clicking this action pops up a window that allows you to specify the properties of a new form. The new form will be associated with the currently selected module.

Open

If you select the radio button for a form and then click this action, a window pops up to allow you to edit the properties of the selected form.

You also can click the hyperlinked name of the form, which causes the same window to pop up.

Copy If you select the radio button for a form and then click this action, a copy of the selected form is created.

Delete If you select the radio button for a form and then click this action, the selected form is deleted.

Data Modeler
Clicking this action opens the Data Modeler tool. The Data Modeler is described in "Data modeling".

Additional actions appear at the top of the Form Builder when you select the radio button to the left of the name of a form. The actions that appear vary based on the status of the form.

If you select the radio button for a form in *Created* or *Revision In Progress* status, one additional action appears to the right of the *Delete* action. The label of this action is *Publish*. Clicking the *Publish* action causes the selected form to be published. Publishing is explained in "Create-Publish-Revise cycle".

If you select the radio button for a form in *Published* status, additional actions appear at the top of the Form Builder:

Report Header
This action is used to create a raw template for a form report. Form reports and this action are described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

ADO XML v2
This action is used for the integration between the IBM TRIRIGA Application Platform and SAP Crystal Reports to help configure SAP

Crystal Reports to produce a form report. This is explained in the *IBM TRIRIGA Application Platform 3 Connector for SAP BusinessObjects User Guide*.

ADO XMLv1

This action is used for the integration between the IBM TRIRIGA Application Platform and SAP Crystal Reports to help configure SAP Crystal Reports to produce a form report. This is explained in the *IBM TRIRIGA Application Platform 3 Connector for SAP BusinessObjects User Guide*.

Revise

This action appears to the right of the *Delete* action.

Clicking this action changes the state of the selected form to *Revision in Progress* and pops up a window that allows you to edit the properties of the selected form. Revision is explained in "Create-Publish-Revise cycle".

Form organization

A form is organized in a hierarchy. Each part of the hierarchy has its own properties. When you create a form, you generally specify the structure and properties of its hierarchy starting at the top and working down to the lowest details.

These are the types of components that appear in the structure of a form:

Form The form contains one or more tabs. Forms and their properties are described in greater detail in "Form properties".

Tab

Top-level tabs are directly contained by a form. Other tabs can be contained by a multi tab section (described in "Stacking sections"). More than one tab can be under the same form or section. However, the contents of only one tab under the same form or section is visible at one time. When you select a tab, its contents become visible and the contents of other tabs under the same form or section are not visible.

A tab contains one or more sections.

Tabs are described in greater detail in "Form tab properties".

Section

A tab contains one or more sections. Each section in a tab usually displays different information, though this is not required. The sections of a tab are different in their purpose than sections in a record. The basic purpose of sections in a record is to contain data. The basic purpose of sections in a tab is the presentation of data, even though it is possible to use the sections of a tab as containers for temporary data. Temporary data is described in "Temporary data".

Different kinds of sections are used to display different kinds of information in different ways. A summary of the different types of sections appears in "Overview of form sections".

Field and buttons

Some types of sections can contain fields or buttons. Fields and buttons are at the bottom of the hierarchy. A field contains an individual piece of information. A button does not contain any information but can trigger an action when clicked.

Fields are described in "Form field properties". Buttons are described in "Buttons".

The structure described in the preceding paragraphs is the internal presentation structure of the form. For forms that are used to present records that are part of a hierarchy, there is also an external presentation structure used to determine what forms will be used to look at other records in the hierarchy. This external presentation structure is discussed in "Includes/Forms tab".

Form Builder tabs and actions

When you create a new form, the window to configure its properties pops up. This window is called a form wizard. Initially, several tabs are visible:

Layout
> The *Layout* tab is used to specify the internal presentation structure of the form.

Includes/Forms

> The *Includes/Forms* tab is used for two separate purposes. It is used to describe the external presentation structure of forms for records that are part of a hierarchy, for example Classifications. This use of the *Includes/Forms* tab is described in "Includes/Forms tab".

> The *Includes/Forms* tab is also used to specify which form reports may be used in a *Reports* tab in the form. Reports tabs are described in "Form properties". Form reports are described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

State Transition
> The *State Transition* tab is used to manage the state-based actions. This is described in "State-based actions".

Revisions
> The *Revisions* tab lists all revisions of the form in reverse chronological order. A revision of a form is created every time the form is modified and published. You can compare two revisions of a form by selecting them and clicking **Compare Revisions**. Or click **Compare Objects** to compare the form with another form of a different name. The results display in the Comparison Result section. To view the results in a new window, click the button next to the Compare menu. You can also export the comparison results to a text file by clicking **Export**. On the Revisions tab, you can revert to a previous revision of the form by selecting it and clicking **Publish**.

The following additional actions appear at the top of the Layout tab:

Reset   Changes the form back to what it looked like the last time it was saved, including window size.

View   Presents a drop-down list of Form Builder sections. Click one to add it to the Form Builder display.

Preview

Renders the form in its current state.

Validate

When selected, the platform checks the section for the following and displays any errors.

- Duplicate fields on a tab.
- Duplicate sections on a tab.

If it is a Group By section, the platform also checks the following:

- Group By field set on the section.
- Only one Group By field set on the section.
- Only one Group By order field set on the section.
- Group By field is visible in the form.
- Group By field is either a text field or a locator field.
- Group By section has a query defined.
- Group By section query has a primary business object set.
- Group By section fields are all Display Columns in the query. (WARNING)
- Group By section columns are the same fields in the same order as the query's Order By.
- Query used by Group By section does not have Group By columns set. (WARNING)
- If Group By field is a locator, the referenced business object has the Group By Columns on it. (ERROR)
- If the Group By Columns Start Columns are equally dispersed. (WARNING)
- If there are no Group By Columns.

Form layout

In the Form Builder, the *Layout* tab is used to specify the internal presentation structure of the form. Its *Navigation* panel shows the organization of the form's internal presentation structure. When a form is first being created, all that is in the *Navigation* panel is an icon that represents the root of the hierarchy that is the internal structure.

The operation of the *Navigation* panel is similar to panels in other platform tools that display a tree. Each part of the tree or *node* is displayed with a name on the right and an icon on the left. If a node has other nodes under it, clicking the node's icon will alternately expose and hide the nodes under it. Clicking the name of the node will select it so you can work with its properties or other aspects.

The label displayed next to the root icon is initially *Form*. The label *Form* is just a placeholder that is used until you specify what the name of the form will be. After you specify that, the name of the form replaces the label *Form*.

The *Layout* panel displays a mock-up of the form being specified. It will help you see how the internal structure you specify lays out on the screen.

When you are just beginning to create a form, the *Layout* panel is blank. As you fill in the structure of the form, you will see the *Layout* panel filled in.

The *Layout* panel shows a mock-up of the form based solely on its internal presentation structure. Some aspects of a form's appearance are not based on its internal structure, such as the presence of an Associations tab. Clicking the *Preview* action at the top of the window causes a preview window to pop up that presents a mock-up of the form that includes things like the Associations tab that are not part of the form's internal presentation structure. In preview mode, the system does not render a report in a report section; instead it displays "Report Section".

The *Properties* panel shows the properties of whatever component of the form's internal structure is selected in the *Navigation* panel. Initially the form is the selected component, so initially the *Properties* panel shows the properties of the form.

The *Components* panel displays a list of fields or sections from business objects that is available for inclusion in a section.

When you create a new form, first specify the properties for the form. You will not be able to save a new form until you have specified values for its *Business Object* and *Name* properties.

After you have saved the form for the first time, it will have a *Publish* action at the top of the *Layout* tab. Clicking the *Publish* action will publish the form. While the state of the form is published, the top of the *Layout* tab will not have a *Publish* action, but it will have a *Revise* action that you will be able to click to revise the form. Publication and revision are described in "Create-Publish-Revise cycle".

Form properties

When you select the form in the Form Wizard's *Navigation* panel, the *Properties* panel opens.

Here are descriptions of the properties for a form:

Business Object

> The value of this property is the name of the business object that the form is associated with.

> When you are creating a new form, this property has no value. Select its value from a drop-down list of the business objects in the module with which the new form is associated.

> You cannot save a new form without first specifying the value of this property. After the first time the new form is saved, this property is grayed out and you cannot change it.

Name

> The value of this property is the name that uniquely identifies the form. People using applications do not see this name.

> When you are creating a new form this property has no value. You must enter a name.

> You cannot save a new form without first specifying the value of this property.

Label The value of this property is the text displayed to identify this form to people using applications.

Description

Enter a description of the form.

Default Form

Since multiple forms can be associated with the same business object, it is convenient to identify one form as the default for a business object. Checking the check box for this property identifies this form as the default for its associated business object.

This check box has some special behavior. If this check box is checked when you save the form properties, it becomes grayed out. You cannot uncheck it in the usual way. The way to uncheck this check box is to make a different form associated with the same business object the default form.

Single Tab

If this check box is not checked and the form has multiple top-level tabs, a portion of the window that the form is displayed in will be used to display things a person can click to navigate from one tab to another.

If this check box is checked, nothing is displayed to allow a person to explicitly navigate from one tab to another. It is up to you, the person who is specifying the internal presentation structure of the form, to insure that appropriate actions are available in each tab to navigate to other tabs in the form.

When you save a form with the Single Tab property checked and then view the form state transition diagram, there will be an additional attribute, Show Tab. This allows an application builder to define multiple tabs on the form and use specific state transitions to determine which tab will be visible to the user.

This check box will not be visible if the specified business object was specified with its Show Single Tab check box not checked. A business object's Show Single Tab check box is described in "Business object properties".

Pre-Load Workflow

If this property has a value, it is the name of a synchronous workflow. The synchronous workflow named by this property is launched just before a form described by this form becomes visible, unless the record is being created for the very first time. Such workflows can be used to perform last minute bookkeeping or fix-ups of records before they are rendered or to adjust the security privileges for the form to be displayed.

You can adjust security privileges on a record based on the current state of the record that the user is viewing. When the pre-load workflow fires, it is pre-populated with a workflow variable. This workflow variable is a record in the triPreloadVariable business object in the System module. The triPreloadVariable record includes a field named triOverrideSecurityGroupIdTX. To override the security of a record, set the value of this field to a security group ID. The record renders only if the user is in the specified security group or in the Admin Group.

The basic process is as follows:

* The platform creates a system record.
* The platform fires the pre-load workflow with that record as an input variable.

- The workflow can then update the triOverrideSecurityGroupIdTX field with the ID of a security group.
- After the pre-load workflow is complete, the platform looks to see if there is a valid security group ID on the input variable record. If there is, then the form is rendered based on that group.

This security override affects this record only. If a child record is opened, the user's default security group is used. If the security needs to work for child records, then another pre-load workflow needs to be configured for them.

If a pre-load workflow does not have the triPreloadVariable configured, then no change is made to the security of the record.

If the triOverrideSecurityGroupIdTX field is blank or not a valid security group record ID, then no change is made to the security of the record.

Synchronous workflows are discussed in "Synchronous versus asynchronous workflows".

Allow to Bookmark to create record
If this check box is checked, this form will be bookmarkable. See "Form bookmarks" for more details on this property.

Allow to Bookmark to specific record
If this check box is checked, specific records created from this form will be bookmarkable. See "Form bookmarks" for more details on this property. Note that in order to have records from this form show up in a Last Visited portal section, this property must be checked.

Calculate Excel

If this check box is not checked, then if this form contains any Excel sections (described in "Excel sections"), the calculations in each Excel section will be performed only if someone clicks the action at the top of the Excel section.

If this check box is checked, clicking any state transition action in the form that saves data will cause the Excel calculations to be done first.

Show Association

If this check box is checked, the form will include an *Associations* tab. An *Associations* tab graphically shows a record's associations with other records. It also allows people to create and remove associations between records.

The *Associations* tab will be present at runtime but will not be part of the form's internal presentation structure. It will not be visible in the *Navigation* panel, but it will be visible in the preview window that pops up when you click the *Preview* action at the top of the Form Wizard's *Layout* tab.

Show Workflow Instance

If this check box is checked, the form will include a *Work Flow Instance* tab. A *Work Flow Instance* tab shows workflow instances that are associated with a record. *Work Flow Instance* tabs are discussed in greater detail in "Workflow instances".

The *Work Flow Instance* tab will be present at runtime but will not be part of the form's internal presentation structure. It will not be visible in the *Navigation* panel, but it will be visible in the preview window that pops up when you click the *Preview* action above the *Properties* panel.

Show Reports

If this check box is checked, the form will include a *Reports* tab. A *Reports* tab allows users to view the content of a record as a form report. A form report can be in any format that can be created using Microsoft Word or Excel or an external report. Form reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

The *Reports* tab will be present at runtime but will not be part of the form's internal presentation structure. It will not be visible in the *Navigation* panel, but it will be visible in the preview window that pops up when you click the *Preview* action above the *Properties* panel.

Show Audit

This property is visible only if the business object associated with the form has the check box checked in its *Audit Access* or *Audit Data Changes* property.

If the check box in this property is checked, the form will have an automatically generated *Audit* tab. Auditing records and the *Audit* tab are discussed in "Audit trails".

Show Audit Actions

This property is visible only if the business object associated with the form has the check box checked in its *Audit Actions* property.

If the check box in this property is checked, the form will have an automatically generated *Audit Actions* tab. Auditing records and the *Audit Actions* tab are discussed in "Audit trails".

Popup Height

This property only applies if the form opens in a pop-up window from a query, the Add action on a query, a custom action on a query section, the Add action on a query section, a chart, a portal section, an action on a portal section, section actions, and field actions.

The values are Small (40% of the screen size), Medium (60% of the screen size), Large (80% of the screen size), Full Screen (100% of the screen size), and blank (the default).

Popup Width

This property only applies if the form opens in a pop-up window from a query, the Add action on a query, a custom action on a query section, the Add action on a query section, a chart, a portal section, or an action on a portal section.

The values are Small (40% of the screen size), Medium (60% of the screen size), Large (80% of the screen size), Full Screen (100% of the screen size), and blank (the default).

Alternate Print Form

If an alternate print form is specified for a form that is being printed, the alternate form is used in the print page.

Sometimes you want the record data to print in a format that is different from the platform default. To specify a form to be used when the user selects the form Print action, select from the forms in this property. The forms in the drop-down list are other forms defined for the same business object.

Revision
>   This property displays the latest revision number of the form. A revision is created every time a form is modified and published.

Object Label
>   This property displays an identifier for the form.

Modified By
>   This property displays the last person who modified the form.

Modified Date
>   This property displays the last time that the form was modified.

Form bookmarks

The properties for *Allow to Bookmark to create record* and *Allow to Bookmark to specific record* in a particular form allow you to control whether or not it makes sense to allow bookmarks for these scenarios.

Note that the *Bookmark this record* option corresponds to the specific record scenario, and it allows the user to bookmark the Andrew Admin employee record. The *Bookmark this form* option allows the user to bookmark the form itself. Selecting the form to bookmark means that the user can create a new employee through a bookmark link in their My Bookmarks.

A user's access to the bookmark capabilities depends on security and license access.

Form tab properties

A tab is a part of a form. A form contains one or more top-level tabs. Top-level tabs take up all the available area in the window. Each tab contains one or more sections. One kind of section a tab can contain is a multi tab section (described in "Multi tab sections"). A multi tab section contains one or more tabs. Tabs in multi tab sections contain exactly one section.

More than one tab can be in the same form or section. However, the contents of only one tab in the same form or section are visible at one time. When you select a tab, its contents become visible and the contents of other tabs in the same form or section are not visible.

Top-level tabs have their own properties. Tabs in a multi tab section do not have their own properties; they are controlled by the properties of the multi tab section that contains them. For a description of multi tab sections, see "Multi tab sections".

To add a top-level tab to a form, first select the form in the *Navigation* panel and then click the *Add Tab* action in the Form Wizard's *Layout* tab. To delete a top-level tab, first select the tab in the *Navigation* panel and then click the *Delete* action in the Form Wizard's *Layout* tab.

While a top-level tab is selected in the *Navigation* panel, its properties appear in the *Properties* panel.

Here are descriptions of a top-level tab's properties:

Name

The value of this property is the name used by workflows to identify the tab. This name must be unique within the form.

You cannot save the properties of a new tab until you set its name. Once you have saved a tab, its *Name* property becomes read-only and its value is grayed out.

Label The value of this property appears as the tab's label.

Tab Information

The value of this property appears as bold text in parentheses in the area under the tab followed by a colon. Common values for this field are Required, Optional, or Summary.

If the contents of the Tab Information field exceed 200 characters when the form is saved, a warning message displays.

Instruction
This property's value appears in the area under the tab after the colon. It should briefly explain the tab's intended use.

Visible

If this check box *is* selected, this tab is visible in the form. If the check box is *not* selected, this tab is hidden. The default for this property is for the check box to be selected.

Workflows can change the value of this property at runtime, making the tab visible or hidden.

Access Key

The value of this property identifies a way to navigate directly to this tab from the keyboard. The value of this property should be a single character. Pressing a key or a key chord and the Access Key character navigates the user directly to the tab. Different browsers use different keys or key chords, as follows:

- For Microsoft Internet Explorer -> Alt+AccessKey+Enter
- For Mozilla Firefox -> Alt+Shift+AccessKey
- For Google Chrome -> Alt+AccessKey
- For Apple Safari -> Alt+Control+AccessKey

Style Class

If you do not specify a value for this property, the appearance of this tab is determined by default settings in the Style Manager.

If you specify a value for this property, it must be the name of a tab style sheet defined by the Style Sheet Editor. The style sheet determines the appearance of the label. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Custom
If this check box *is* selected, the IBM TRIRIGA Application Platform does not generate the contents of this tab in the normal way. Instead, the tab's contents are taken from the URL that is the value of the *URL* property.

URL This property is visible only if the check box in the *Custom* property is selected. If the check box in the *Custom* property is selected, the contents of the tab are taken from the URL that is the value of this property.

Copying tabs

About this task

You can copy a tab from the current form to another place in the current form, to another form in the same business object, or to another form in a different business object in the same module.

Procedure

1. Select the tab in the Navigation panel. While the tab is selected, there is an action labeled **Copy Tab** at the top of the Form Wizard's Layout tab. The purpose of the **Copy Tab** action is to allow you to copy the tab and paste it.

2. When you click the **Copy Tab** action, the Properties panel changes to allow you to specify the properties of the tab when it is pasted. The properties are as follows:

   Name   This is the name of the tab in the target form. The default is the name of the source tab. You cannot use the same name unless you are copying the tab to a different form. The name must be unique in the target form.

   Target Business Object
   This is the business object of the target form. The default is the business object of the current form. You can copy a tab to a form in the same business object or in a different business object in the same module. Select the Target Business Object from the drop-down list. The list shows all business objects in the same module as the current form.

   Target Form
   This is the form into which the tab will be copied. The default is the current form. You can copy a tab into any form in the selected Target Business Object. Select the target form from the drop-down list. The list shows all forms in the target business object.

3. Click the **Apply** action in the Properties panel. The system revises the target form if necessary and pastes the copied tab as the last tab in the target form. Be sure to publish the target form.

   - If the target business object is not the same as the source business object, the system removes fields, smart section fields, and smart sections that are not valid in the target business object.
   - You will see an error message if the tab is not uniquely named within the target form.

Sorting tabs

Procedure

1. To specify the order in which top-level tabs appear in a form, begin by selecting the form in the *Navigation* panel.

2. While the form is selected, there is an action labeled *Sort Tab* at the top of the form Wizard's *Layout* tab. The purpose of the *Sort Tab* action is to allow you to rearrange the order of top-level tabs.

3. When you click the *Sort Tab* action, a list of the form's top-level tabs appears in the *Properties* panel.

4. The top-level tabs in the list appear in the same order they will appear in the form. Click the arrow heads in the first column to move the tabs up or down in the list.

5. Remember to save the form before you move on to something else, otherwise the modified tab order will be lost. Also remember to see what your tabs look like in the window that pops up when you click the *Preview* action. The preview window is a more accurate indication of what tabs will look like at runtime.

   Note: It is sometimes convenient to include the same record field in more than one tab. The Form Builder allows you to include the same field in multiple tabs. However, it will not allow you to have the same field on the same tab. If that occurs, you will not be able to publish the form. The **Validate** action can help you find duplicate fields.

Overview of form sections

You can add a section to a tab by clicking the tab's name in the *Navigation* panel and then clicking the *Add Section* action.

Sections of a form are used to present information of some sort. These kinds of sections are used to present different kinds of information:

Form Section

A form section can contain fields and buttons. Each field may be explicitly based on a field in the business object associated with this form.

Form sections are discussed in greater detail in "Form sections".

Smart Section

A smart section is connected with a smart section in the business object associated with the form. A smart section can contain fields explicitly connected with fields in the business object's smart section.

Smart sections are discussed in greater detail in "Smart sections".

Multi Tab Section

A multi tab section cannot directly contain any fields. It can only contain other sections. A tab is created for each section in a multi tab section.

Multi tab sections are discussed in greater detail in "Multi tab sections".

Query Section

A query section displays the results of a query. The query may be produced by the IBM TRIRIGA Application Platform internal report generator or by an external report generator. See the *IBM TRIRIGA Application Platform 3 Reporting User Guide* for details about building queries.

Query sections are discussed in greater detail in "Query sections".

Report Section

A report section displays the results of a form report.

Report sections are discussed in greater detail in "Report sections".

Graphics Section

Graphics sections exist to display a graphic image that is associated with an organization, geography or location.

Graphics sections are discussed in greater detail in "Graphics sections".

Excel Section

> Excel sections are used to display an Excel spreadsheet and use the calculations of the spreadsheet to update fields in other sections.

> Excel sections are discussed in greater detail in "Excel sections".

Gantt Section

> Gantt sections are used to display Gantt charts.

> Gantt sections are discussed in greater detail in "Gantt sections".

Availability Legacy Section

> Availability Legacy sections are used to display the availability of resources.

> Availability Legacy sections are discussed in greater detail in "Availability legacy sections".

Stacking Section

> Stacking sections are used to view current or planned assignments of demand to supply and allow the user to make changes by moving some of the demand from one location to another.

> Stacking sections are discussed in greater detail in "Stacking sections".

Group By Section

> Group By sections show data in a tabular grid-like format grouped by the values of a specified field. The data in a Group By section is sourced from a query.

> Group By sections are discussed in greater detail in "Group by sections".

Calendar Section

> Calendar sections show resource calendars. The data in a calendar section is sourced from a calendar set.

> Calendar sections are discussed in greater detail in "Calendar sections".

Availability Section

> Availability sections display resource availability charts. The data in an availability section is sourced from a query.

> Availability sections are discussed in greater detail in "Availability sections".

When you create a new section, the first thing you need to do is specify the type of section it will be.

After you set the value of the *Type* property, the set of properties visible below the *Type* property changes to properties that are appropriate for the selected type of section.

After the first time you save a section's properties, the *Type* property becomes grayed out and you are no longer able to change its value.

If the type of section you want to create is a smart section, there is a shortcut that can save you a few steps. While you are editing the properties of a tab, the *Components* panel contains a list of smart sections in the form's associated business

object. This list of smart sections looks like the following figure.
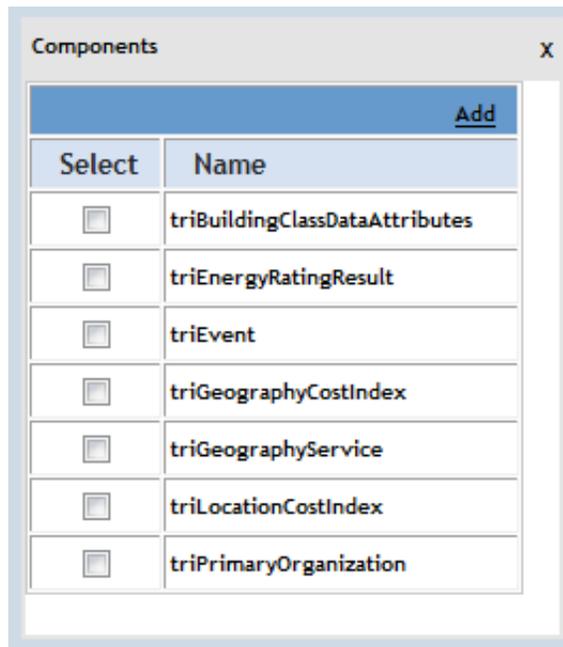
Figure 29. List of sections in Components panel

To create smart sections in the currently selected tab that correspond to sections in the associated business object, check the check box next to each section's name, then click the *Add* action. A smart section will be added to the currently selected tab for each section selected in the *Components* panel.

A word about single-record smart sections: To specify that a user can click a hyperlink to the linked record in a single-record smart section, check the smart section's Show Embedded Link check box.

Section copies

You can copy a section from the current tab to another place in the current form. Begin by selecting the section in the Navigation panel. While the section is selected, there is an action labeled Copy Section at the top of the Form Wizard's Layout tab. The purpose of the Copy Section action is to allow you to copy a section and paste it.

When you click the Copy Section action, the Properties panel changes to allow you to specify the properties of the section when it is pasted. The properties are as follows:

Name This is the name of the section in the target tab. The default is the name of the source section. You cannot use the same name unless you are copying the section to a different tab. The name must be unique in the target tab.

Target Tab
This is the tab into which the section will be copied. The default is the current tab. You can copy a section into any tab in the current form. Select the target tab from the drop-down list. The list shows all tabs in the current form.

Click the Apply action in the Properties panel. The system pastes the copied section as the last section in the target tab. You will see an error message if the section is not uniquely named within the target tab.

The Target MultiTab property appears if the source section is not a multi tab section and the target tab you picked has a multi tab section. The Target MultiTab property allows you to place a section into a multi tab section. You can copy a section from a multi tab section, copy a section into a multi tab section, or copy an entire multi tabsection. However, you cannot copy a multi tab section into a multi tab section.

The system displays an error message if you try to publish a form that has form fields or form smart sections duplicated on a given tab. Use the Validate action to help work through fixing those issues.

Tip: To move a section, use a Copy Section and then a Delete.

Tip: To copy a section to another form, use the Copy Tab action to copy the tab containing the section to the other form and proceed from there.

Row insertion

You can insert a row before a section or a field. This provides an easy way to insert fields/tabs into the middle of an existing layout. Begin by selecting the section or field in the Navigation panel. While the section or field is selected, there is an action labeled Insert Row at the top of the Form Wizard's Layout tab.

When you select a section and click the Insert Row action, the system increments the Start Row property on all sections in the tab from that section on and refreshes the selection.

When you select a field and click the Insert Row action, the system increments the Start Row property on all fields in the section from that field on and refreshes the selection.

Common section properties

Many of the properties of a section are the same regardless of the form type. This section discusses these common properties. The information is not repeated in other parts of this user guide except to highlight a difference from the common property or to add additional information.

Type   The type of section this will be. Select the value from the drop-down list: Form Section, Smart Section, Multi Tab Section, Query Section, Report Section, Graphics Section, Excel Section, Gantt Section, Availability Legacy Section, Stacking Section, Group By Section, Calendar Section, or Availability Section.

Name   The value of this property is the name that workflows use to identify the section. It must be unique within the tab that contains it.

Label   The value of this property is the text that will appear in the title bar at the top of the section.

Height

The number that is the value of this property determines the height of the area used to display the section. The height is measured in lines of text. This sets the boundary in which an overflow of data causes scroll bars to appear.

The default value is 0 (zero), which translates to an initial display of 10 rows of data with a height of 23 pixels for each row.

Title Bar Color

This property usually does not have a value. If this property does not have a value, the background color of the title bar for this section will be the default color specified in the Style Manager.

To specify a color, click the search icon. A palette of colors will pop up. After you have selected a color by clicking it, the color palette disappears and a rectangle of the selected color appears to the left of the search icon. The Advanced color picker also is available. See "Color fields" for its usage.

Clicking the X icon causes the color selection to be cleared, so that the default color will be used for the title bar.

Visible

If this property's check box *is* checked, this section will be visible. If the check box is *not* checked, this section will be hidden. The default for this property is for the check box to be checked.

Workflows can change the value of this property at runtime, making the section visible or hidden as appropriate.

Expand Section
This is checked by default. If this check box *is* checked, the first time a user uses this form, this section will be expanded by default. If *not* checked, the section will be collapsed by default. Note that for subsequent use of this form, IBM TRIRIGA remembers the state of each section for each user.

Read Only
If this check box is checked, all fields in this section will be read-only. People will not be allowed to modify the contents of the fields using the form or to drag or resize events.

Show Title Bar

This check box is usually checked. If this check box *is* checked, a title bar will be displayed above the section. If the check box is *not* checked, no title bar will be displayed.

A common reason to leave this check box unchecked is so that what is internally organized as two sections appears to be one section.

Border Size
The number that is the value of this property determines the width of the border around this section.

Border Color

If this property does not have a value, the border around this section is the default color specified in the Style Manager.

To specify a color, click the search icon. A palette of colors will pop up. After you have selected a color by clicking it, the color palette disappears and a rectangle of the selected color appears to the left of the search icon. The Advanced color picker also is available. See "Color fields" for its usage.

Clicking the X icon causes the color selection to be cleared, so that the default color will be used for the title bar.

Style Class

If you do not specify a value for this property, the appearance of this section will be determined by default settings in the Style Manager.

If you do specify a value for this property, it will be the name of a section style sheet defined by the Style Sheet Editor. The style sheet determines the appearance of the label. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Start Row

The value of the *Start Row* property specifies in which of the tab's rows this section will appear. The rows in tabs are counted from the top to the bottom, so row 1 is above row 2. If a section is more than one row tall, this property specifies in which row of the tab the top of the section will appear.

The platform uses the absolute value of the number entered.

Row Span

This property specifies the height of the section in rows. For most sections, the *Row Span* value is 1.

The *Start Column* and *Col Span* properties determine a section's horizontal position and width within its tab. A tab is laid out in twelve columns. Most tabs do not look like they are laid out in twelve columns because they do not use all twelve columns. If a tab has no sections in a column, the column has no width and it looks like the column is not there. The following figure is a visual explanation of how the values of the *Start Row*, *Row Span*, *Start Column* and *Col Span* properties combine to determine a section's size and position within a tab.
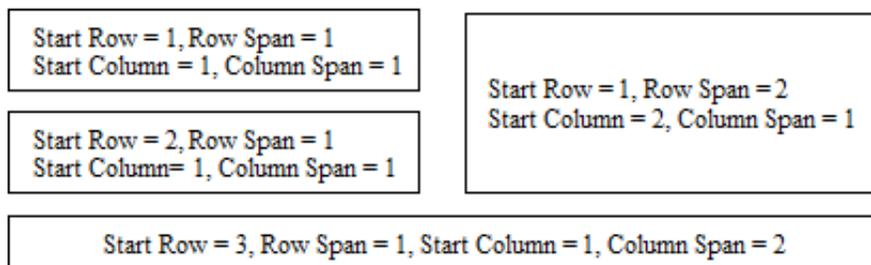
Figure 30. Start Row, Row Span, Start Column, and Column Span combinations

Form sections

A Form section can contain fields and buttons. Each field may be explicitly based on a field in the associated business object.

If you want fields in the form that are connected to fields in the associated business object that *are* part of a smart section, you must put those fields in a smart section. Smart sections are discussed in "Smart sections".

When you are editing a Form section, there are two aspects of the section you may be concerned with. As with all components, a Form section has properties you will

want to give appropriate values. You also will want a Form section to contain the appropriate fields and buttons laid out in the appropriate way.

Fields are discussed in "Form field properties". Buttons are discussed in "Buttons"

See "Common section properties" for information about the following properties in a Form section: Type, Name, Label, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span.

Once you have set a Form section's properties to appropriate values, the usual next step is to add fields to the section and give them an appropriate layout.

Once this section's fields are set up, you may want to specify what actions will be available in the section. Actions for sections are discussed in "Section actions".

Form field properties

There are two ways you can edit the properties of a field: You can click the name of the field in the *Navigation* panel or add a new field to a section.

To add a field to a form section or smart section, first click the section's name in the *Navigation* panel so the section properties appear in the *Properties* panel. Then click the *Add Field* action at the top of the *Layout* tab.

There are three kinds of fields you can add to a section:
- A Data field is a field that is explicitly connected to a field in the business object with which the form is associated.
- A Form Field is a field that is not explicitly connected to any field in a business object.
- A Form Action is not really a field. It is a button. It is something you can put in a section that does not contain data. You can arrange for a workflow to run when someone clicks a button. Buttons are described in greater detail in "Buttons".

If the kind of field you want to add to a section is a *Data* field, there is a shortcut that may be less work than clicking the *Add Field* action. If you are looking at properties of a section you can add a field to, the *Components* panel contains a list of the fields in the form's associated business object that you can use to create a *Data* field. After you select check boxes in the list and then click the *Components* panel's *Add* action, Data fields that correspond to record fields whose check box you checked will be added to the currently selected section in the form.

When you are editing the properties of a field, the field's properties will appear in the *Properties* panel.

Additional properties may become visible as you select values of some of the initially shown properties. Here are descriptions of the properties:

Type

A field has two properties named *Type*. This is a description of the first.

The value of this property determines whether the field is a form field, a data field, or a button. The value for this field is chosen from a drop-down list. The possible values are:

Data   If the value of the *Field Type* property is *Data*, the field will be a data field connected to the business object's field specified by the *Data Field* and *Data Section* properties.

Form Field

If the value of the *Field Type* property is *Form Field,* the field will be a form field not connected to any business object field. This is the equivalent of a label and does not let the user input data.

Form fields are usually used to present instructions or messages.

Form Action

If the value of the *Field Type* property is *Form Action,* the field will be a button. Buttons are described in "Buttons".

After a new field is saved for the first time, the Type property becomes read-only and is not displayed in the *Properties* panel.

Data Field

This property is visible only if the value of the *Type* property is *Data*. Otherwise, this property is hidden and does not appear in the *Properties* panel.

The value of this property is the name of the business object field that this data field is connected to. The names of available business object fields appear in a drop-down list.

When you select a field from this list, the name of the smart section it appears in becomes the value of this field's *Data Section* property. If the business object field is not part of a section, the value of this field's *Data Section* property is set to *General*.

After a new field is saved for the first time, this property becomes read-only and its value is grayed out.

Data Section

This property is visible only if the value of the *Type* property is *Data*. Otherwise, this property is hidden and does not appear in the *Properties* panel.

The value of this property cannot be set directly, rather it is set when you select a value for the *Data Field* property. When you select a value for the *Data Field* property, the value of this property is set to the name of the smart section that contains the business object field that is named by the value of the *Data Field* property. If the business object field that is named by the value of the *Data Field* property is not part of a smart section, the value of this field is set to *General*.

After a new field is saved for the first time, this property becomes read-only and its value is grayed out.

Name

The value of this property is the name that workflows can use to refer to this field. The name must be unique within the section that contains this field.

After a new field is saved for the first time, this property becomes read-only and its value is grayed out.

Label The text that is the value of this property is used as the field's label. The property whose value determines the appearance of the label is the *Label Style Class property.*

Type

A field has two properties named *Type*. This is a description of the second.

If this field is a Data field, this property is read-only and the value of this property is the data type of the business object field that this field is connected to. For example, Smart Object.

If this field is a Form Field, the value of this property is always *Label*.

If this field is a Form Action, the value of this property is always Form Action. Refer to "Buttons" for information about the properties of a Form Action field.

Root Classification
This property is visible only if the value of the second Type field is Classification. Click the search icon to select the value.

Lookup Query

This property is visible only if the value of the second Type field is Classification.

When blank, at runtime the system uses the hierarchy tree to determine the value.

If a query is present, at runtime the system runs the specified query to select a value. For the tabout functionality to work, the query must contain the field triNameTX in TRIRIGA 9.x or IBM TRIRIGA 10.x, or ClassName in TRIRIGA 8x.

Display Type

This property is visible only if the value of the *Data Type* property is *Binary*. The value of this property can be chosen from two values in a drop-down list:

Content
If the value of this property is *Content*, the scrollable area this field takes up will be used to display whatever rendering of the binary data the browser is configured to use for the type of binary data this field has for its value.

Link If the value of this property is *Link*, this field is rendered as a hyperlink. Clicking the hyperlink causes the browser to pop up a separate window to display whatever rendering of the binary data the browser is configured to use for the type of binary data this field has for its value.

Icon Position

This property is visible only if the value of the *Data Type* property is a type of field that has icons next to its data area that can be clicked to set the field's value, such as *Business Object* or *Classification*. This field may be narrower than the width of the column it is in. In addition to this field's label and data areas, it also has an icon a person can click to set the field's value.

The value of this property can be chosen from two values in a drop-down list:

Next to Field

>> If this property's value is *Next to Field*, the icon will be positioned next to the right side of the field's data area.

Right Align

>> If this property's value is *Right Align*, the icon will be positioned next to the right side of the column.

Display Fields

> This property is visible only if the value of the first Type property is Data and the value of the second Type property is Duration. Select the check boxes for the duration choices to be available to users; for example, Year(s) and Month(s).

Enable Locator

> This property is visible only if the field is a locator field in a dependent single-record smart section. At runtime, this field is a locator field.

Locator Query

> This property is visible only if the Enable Locator property is selected. It identifies the query that provides the locator.

Alternate Form List

> This property is visible only if the Enable Locator property is selected. It identifies the alternate form list to be used to display a record when that record is opened.

Icon Position

> This property is visible only if the Enable Locator property is selected. The value of this property determines where the locator icon is positioned on the form.

Minimum Size

> The value of this property determines the minimum number of characters that must be entered into this field for the value in the field to be considered valid.

Maximum Size

> The value of this property determines the number of visible characters when this field is displayed to the user. This prevents the displayed size of the field from expanding to be larger than the size to contain this number of characters. If the value of this property is 0, the field's display expands to show all characters. Note, the Size property of the field from the Data Modeler overrides this value if it is less then the Maximum Size.

Score Type

> This property is visible only if the value of the Type field is Number, Financial Rollup, or Classification Rollup and the field is defined in the Data Modeler as a scored number field (as described in "Number fields"). The value of this property can be chosen from a drop-down list containing the following:

>> **Number Value Only**

>>> Display the number value of the field only and not the score image.

>> **Score Image Only**

>>> Display the score image only and not the number value of the field.

>> **Both** Display both the score image and the number value of the field.

Delta Type

This property is visible only if the value of the second Type field is Number, Financial Rollup, or Classification Rollup and the field belongs to an enabled vertical comparison section (as described in "Smart sections").

The delta value is the difference between this field's value in this record and its value in the compare to record. This value is not stored; it is only a runtime calculation for viewing the data.

The values of the Delta Type property are as follows:

No Delta
The delta value is not displayed in the vertical comparison section.

Numeric Delta
The delta value displayed will be numeric and is calculated by subtracting the value of the record being compared from the compare to record's value.

Percent Delta
The delta value displayed will be numeric and is calculated by subtracting the compare to record's value from the value of the record being compared and dividing the result by the compare to record's value.

Delta Precision

This property is visible only if the value of the Type field is Number, Financial Rollup, or Classification Rollup and the field belongs to an enabled vertical comparison section (as described in "Smart sections").

The Delta Precision property defines how many decimal places the platform is to use when computing and displaying the delta value for this field. Enter the number of digits to the right of the decimal point. The maximum value is 5.

Use Custom Display Mask

This property specifies whether or not this field's display mask uses the Display Mask property defined for the business object or the UOM.

When Use Custom Display Mask is not checked, the platform uses the display mask defined on the business object or UOM.

When the Use Custom Display Mask property is checked, the Display Mask property is visible and set to the value defined on the business object or UOM. Changing the Display Mask property only affects this field. See the definition of Display Mask in "Number fields".

The chart in "Number field display and storage" describes the Display Mask property override relationship.

If a value has more digits to the right of the decimal place than shown in the Display Mask property, the platform uses Round Half Up to round the value in the display.

Input Width

The value of this property determines whether this field will be displayed with its natural width or the full width of the column it occupies.

A text field with its *Maximum Size* property set to 0 has no natural width. However, many fields do have a natural width. For example, a text field with its *Maximum Size* property set to 15 has a natural width wide enough

to hold 15 characters. Some fields have a natural width based on the type of data they contain. A field that contains a date will have a natural width based on the date format that is in the user's *My Profile* record.

Each column of a section is at least as wide as the widest combination of field and label the column contains. A column may be noticeably wider than the natural width of some of the fields it contains.

This property has two possible values that can be chosen from a drop-down list. These are the possible values:

Default
> If the value of this property is *Default*, then the field will be displayed only as wide as its natural width.

Full Column
> If the value of this property is *Full Column*, then the field will be displayed as wide as the width of the column that contains it.

Label Style Class

> If you do not specify a value for this property, the appearance of this field's label will be determined by default settings in the Style Manager.

> If you do specify a value for this property, it will be the name of a field style sheet defined by the Style Sheet Editor. The style sheet determines the appearance of the label. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Data Style Class

> If you specify a value for this field, it will be the name of a data style defined by the IBM TRIRIGA Application Platform Style Manager tool. The named style is used to specify the appearance of the field's data instead of the default.

> If you do not specify a value for this property, then the appearance of this field's data will be governed entirely by the default specified in the IBM TRIRIGA Application Platform Style Manager tool. This tool is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Data Alignment
> The value of this property determines the alignment of data in this field when it is displayed. The possible values for this property are:
> - Left
> - Center
> - Right

Hierarchical Parent

> This property is visible only for fields in smart sections with the Hierarchical Fields property selected. Such sections are backed by a Vertical Table business object section. Vertical sections are described in "Record organization".

> To indicate that this field should be displayed in a Vertical section as the child of another field, select the parent field in the drop-down list.

> If this property is blank, this field does not have a parent.

> If a field is set to be not visible, it and its child records will not be visible.

Hierarchical State

This property is visible only for fields in smart sections with the Hierarchical Fields property selected and when a value is not selected in the Hierarchical Parent property.

If this field turns out to have child fields, this property determines whether the parent / child relationship is initially rendered as expanded or collapsed.

Start Row

The following description of this property does not apply if the field is in a smart section that is connected to a multiple-record smart section. The interpretation of this property in a smart section that is connected to a multiple-record smart section is discussed in "Multiple-record smart sections".

The *Start Row* property specifies in which of the section's rows the field will appear. The rows in sections are counted from the top to the bottom, so row 1 is above row 2. If a field is more than one row tall, this property specifies which row of the section the top of the field will appear in.

See "Common section properties" for a visual description of this property.

Row Span

The following description of this property does not apply if the field is in a smart section that is connected to a multiple-record smart section. The interpretation of this property in a smart section that is connected to a multiple-record smart section is discussed in "Multiple-record smart sections".

The value of this property specifies the height of the field in rows of a section. For most fields, the value of this property is 1. Descriptions and other lengthy text fields usually take up multiple lines, so the value of their *Row Span* property is usually at least 3.

See "Common section properties" for a visual description of this property.

Start Column

The following description of this property does not apply if the field is in a smart section that is connected to a multiple-record smart section. The interpretation of this property in a smart section that is connected to a multiple-record smart section is discussed in "Multiple-record smart sections".

The value of the Start Column property determines the field's horizontal position. The value of this property specifies in which of the section's columns this field will appear. If the field occupies more than one column, the value of this property is the leftmost column that this field occupies.

A section is laid out in twelve columns. Most sections do not look like they are laid out in twelve columns because they do not use all twelve columns. If a section has nothing in a column, the column has no width and it looks like the column is not there.

You can choose a value from 1 to 12 for this property in a drop-down list.

See "Common section properties" for a visual description of this property.

Col Span

The following description of this property does not apply if the field is in a smart section that is connected to a multiple-record smart section. The

interpretation of this property in a smart section that is connected to a multiple-record smart section is discussed in "Multiple-record smart sections".

The value of the Col Span property specifies the width of the field in columns of a section. For most fields, the value of this property is 1. Descriptions and other lengthy text fields usually have a value for *Col Span* of at least 2.

See "Common section properties" for a visual description of this property.

Group Seq

It is possible to have more than one field in the same rectangle that is specified by the *Start Row*, *Row Span*, *Start Column* and *Col Span* properties. If more than one field in a section is specified with these properties having the same value, all the fields with the same values for these properties will appear in the same rectangle. The order in which the fields appear in the rectangle is determined by the value of this property.

Prefix If the Group Seq property is greater then zero, the Prefix property will be displayed. The value of this property defines the text that will be displayed between grouped fields. This text will be displayed instead of the Label.

Tab Order

A person using this section can navigate between its fields and buttons by pressing the tab key. The order in which the tab key navigates to the fields and buttons in a section is determined by their *Tab Order* property. The tab key will navigate to fields and buttons in a section in the same order as the relative magnitude of the value of their *Tab Order* property. If the value of this property is 0 for all fields in the section, the tab order is left to right, top to bottom of the fields in the section.

Do not use this feature. Custom tab ordering can lead to confusion for screen readers and can break accessibility rules. For applications with custom tab ordering, the tabbing traverses the fields with custom ordering first, then continues with the remaining objects starting with the top-most tab-enabled object.

Required

If this property's check box is checked, a user will not be able to save the record until he or she enters a value into this field. If the business object's field is set as required in the Form Builder, the Form Field required flag will be set and the check box is read only and cannot be changed.

Show UOM

This property is visible only if the value of the *Data Type* property is *Number*. If the property's check box is checked, the unit of measure (UOM) associated with the field will be displayed. If the field is not read-only and there is no Source UOM for the field, a user will be able to select a different UOM.

Units of measure are discussed in "Units of measure".

Hide Label

If the check box for this property is checked, this field is displayed without a label.

Position

The value of this property is normally *Both*. A value of *Both* means that both the label and data portions of the column will be used to display the field.

There are two other possible values for this property that you can select from a drop-down list. The other values for this property are primarily intended to be used when the check box for the *Hide Label* property is checked.

If the value of this property is *Label*, only the label portion of the column will be used to display this field.

If the value of this property is *Data*, only the data portion of the column will be used to display this field.

If the value of this field's *Col Span* property is greater than 1, the width of the multiple columns this field spans is first broken up into label and data areas. The field is positioned into one area, the other area, or both areas, depending on the value of this property.

Visible

If this check box is checked, this field will be visible. If the check box is not checked, this field will be hidden. The default is for the check box to be checked.

Workflows can change the value of this property at runtime, making the field visible or hidden as appropriate.

Read Only

If the check box for this property is checked, it will not be possible for a user to interactively change the value of this field by directly editing the field.

If a field is read-only, it does not have a border. If a field is editable (not read only), it has a border.

The next three fields are intended for situations in which you want a field of records created from the same business object to contain different kinds of data. An example of when this is useful is a survey for which the answers to different kinds questions may be different kinds of data. You want to use one kind of record to contain responses to questions. Define the field that will contain the answers as a text field. Then set the following three properties so that the field will be presented by the form as text, a number, a list, a check box, or whatever is appropriate for the specific question. The idea is that other fields in the same record will control the presentation of the field that contains the answers.

Type Field

This property is visible only if the value of the *Data Type* property is *Text*. Even though the value of a text field is always stored as text, it is possible for the field's value to be presented in forms and reports as if it were a date, number, duration, or some other type of data. The way that this field's data is presented can vary by record and over time. The purpose of this property is to control the presentation of this field as if it contained a different data type.

If this property has no value, this field will be presented as a normal text field.

If this property does have a value, even though this field's value will be stored as text, the field will be presented by the form and in reports as whatever type is named by the field whose name is the value of this property.

This property's value is chosen from a drop-down list of other text fields in the same section. Though it is not required, the field named by this property will usually have its *Visible* check box unchecked.

The field you select to drive this property should be a list field, so users can only select a valid data type. There is a list in the List Manager named triDynamicFieldType that has all valid field types defined.

Content Field

This property is visible only if the value of the *Data Type* property is *Text*.

If the *Type Field* property has a value that is the name of another field in this section and the text value of the named field is "List", the value of the field named by this property will be used as the name of the list of values to be presented by the field.

Lists of values are managed by the List Manager, which is discussed in "List management".

The value of this property is chosen from a drop-down list of other text fields in the same section. Though it is not required, the field named by this property will usually have its *Visible* check box unchecked.

Required Field

This property is visible only if the value of the *Data Type* property is *Text*.

If this property has a value, this field's *Required* property will be ignored. Instead, the value of the field whose name is the value of this property will determine whether this field is required or not.

The value of this property is chosen from a drop-down list of other text fields in the same section. Though it is not required, the field named by this property will usually have its *Visible* check box unchecked.

If the text value of the named field is "true" or "yes", this field will be required. If the text value of the named field is "false" or "no", this field will not be required.

Do not use this property. Similar functionality can be achieved using a workflow to change the required metadata property. See "Modify metadata task" for more details on how to do this.

After you have finished setting up the properties of a section, you may want to set up actions that can be performed within the section. Section actions are described in "Section actions".

Example: Form fields can receive different data types

In order to illustrate the use of these three properties, we will now go through a simple tutorial.

In the Data Modeler, create/revise a business object and add the following fields:
- Create a new text field cstComboTX
- Create a new text field cstListNameTX

- Create a new list field cstFieldTypeLI. For the Module, select System, and for the List, select triDynamicFieldType
- Create a new text field cstRequiredTX.

Publish the business object (create a Publish Name and a data-revise state transition diagram if necessary).

In the Form Builder, create a new form for this business object. Add each of the fields to the form. For the cstComboTX field, set the Type Field to cstFieldTypeTX, the Content Field to cstListNameTX, and the Required Field to cstRequiredTX, as shown in the following figure.
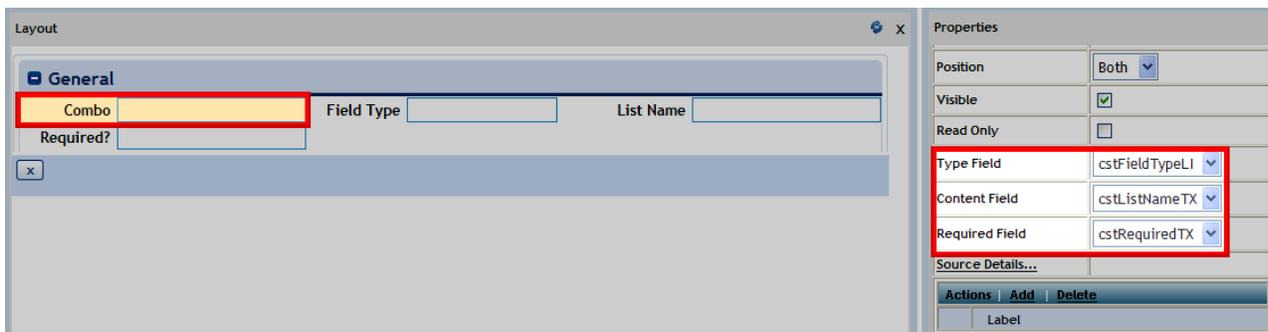
Figure 31. Tutorial - Form Builder

Now, create a new form of this type (you may need to create a navigation item for this new form first, see *IBM TRIRIGA Application Platform 3 User Experience User Guide* for details on how to do this).

To demonstrate the Required Field property, assume the Combo field was originally set to not required in the Form Builder. Now type true or yes in the Required? field and save the record. Notice the Combo field is now required. Similarly, if the Combo field was set to required in the Form Builder, you can override it by typing in false or no in the Required? field.

To demonstrate the Type Field property, open the record for this form. Change the Field Type to Color and save. Notice the Combo field literally becomes a color field, as shown in the following figure. Similarly you can change the Field type to other values and the type of the Combo field changes accordingly. Note that the underlying representation of the field is saved as text.
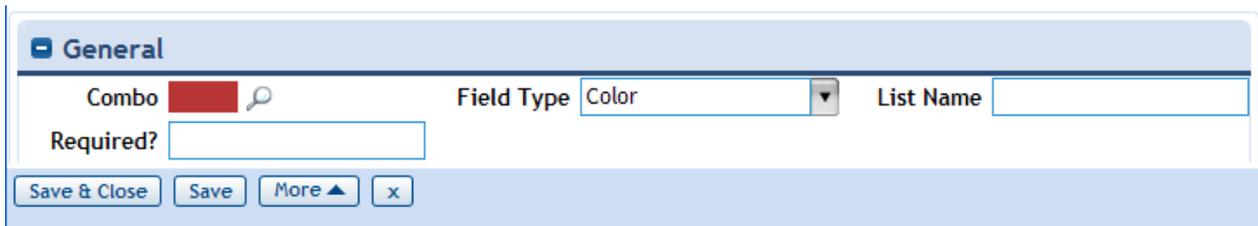
Figure 32. Tutorial - Field Type = Color

To demonstrate the Content Field property, first set the corresponding Type Field property, Field Type, to List (alternatively you could have used a hidden, read-only text field set to List). Now set the Content Field, List Name, to the name of a list in the system. For example, type in Language and save. Now the Combo field has a

list drop down of the Languages in the system, as shown in the following figure. Similarly, if you change the List Name to Year, the Combo field will have a list drop down of years, and so forth. Note that if the List Name field does not contain a valid list, the drop down will have an empty list. In addition, if the Field Type is not List, the List Name field will be ignored.
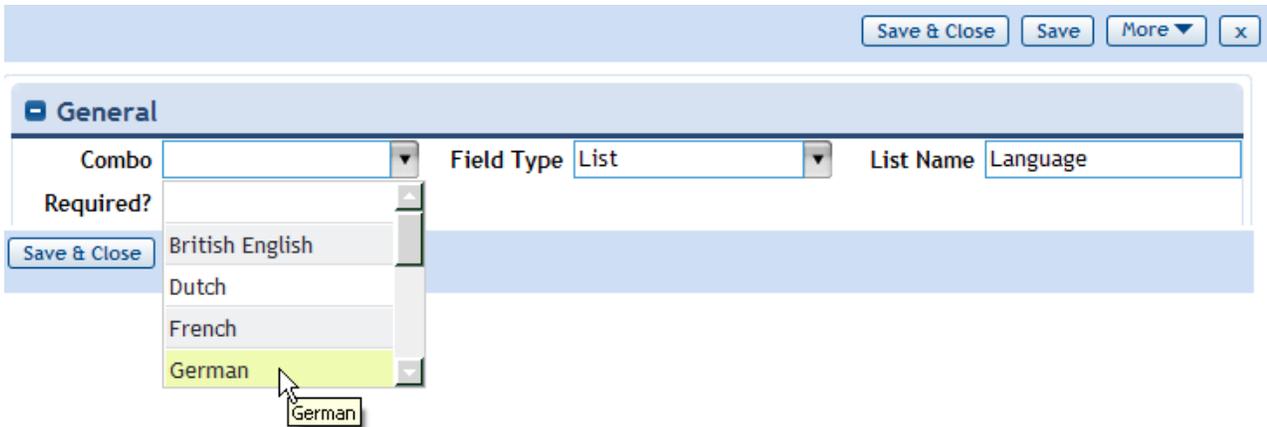
Figure 33. Tutorial - Field Type = List

Despite the flexibility these properties provide, this comes at the cost of difficulty--it is very easy to configure these properties improperly, so you should explore other options, e.g., using workflows and swapping in hidden fields, if doing so can accomplish the same thing. Needless to say, target fields like List Name should be hidden to prevent the casual user from being able to search for sensitive list data values with this setup.

Buttons

The way you edit a button or add a button to a smart section or form section is the same as the way you edit a field. This is because the Form Wizard treats buttons as a kind of field. Fields are described in "Form field properties".

Buttons have a fundamentally different purpose than fields. The purpose of a field is to allow someone to view or edit a piece of data. The purpose of a button is to trigger an action when someone clicks it. Actions are discussed in "Actions".

Buttons have some of the same properties as fields and some additional properties. Buttons always have property Type of Form Action.

Button's properties that were not discussed for fields are described below:

Display Type

The value of this property determines what sort of button this will be displayed as. These are the choices:

Button

If the value of this property is *Button*, this button will be displayed as a push button.

Image Only

If the value of this property is *Image Only*, this button will be displayed as a hyperlink just using the images specified by the *Display Image* and *Press Image* properties. No label text is displayed.

Image and Text

If the value of this property is *Image and Text*, this button will be displayed as a hyperlink using the label text specified by the *Label* property and the images specified by the *Display Image* and *Press Image* properties.

Text Only

If the value of this property is *Text Only*, then this button will be displayed as a hyperlink using just the label text specified by the *Label* property.

Display Image

If the value of the *Display Type* property is *Image Only* or *Image and Text*, the image that is the value of this property will be the image that is normally displayed for this button.

To specify an image to be the value of this property, click the search icon. Select the file that contains the image you want. A small version of the image appears to the left of the search icon.

Clicking the X icon causes the image selection to be cleared, so that there will be no image to display for the button.

If the value of the *Display Type* property is *Button* or *Text Only*, the value of this property is irrelevant and this property is ignored.

Image Tooltip

If the value of the *Display Type* property is *Image Only* or *Image and Text*, the Image Tooltip field appears. Enter the text of the tooltip you wish to appear when a user moves their mouse over the image.

Press Image

If the value of the *Display Type* property is *Image Only* or *Image and Text*, the image that is the value of the Display Image property will be the image that is normally displayed for this button. When the mouse pointer is over this button and the mouse button is pressed, you may want an alternate image to be displayed for this button to give the appearance that the button is being pressed. If the Press Image property has a value that is an image, the image that is the value of this property is the alternate image.

To specify an image to be the value of this property, click the search icon. Select the file that contains the image you want. A small version of the image appears to the left of the search icon.

Clicking the X icon causes the image selection to be cleared, so that there will be no image to display for the button.

If the value of the *Display Type* property is *Button* or *Text Only*, the value of this property is irrelevant and this property is ignored.

Active If this check box is checked, this button has its normal appearance and clicking this button will cause an action to be triggered. If this check box is not checked, this button will appear to be grayed out and clicking it will have no effect.

Smart sections

Smart sections are similar to form sections in that they both contain fields. Smart sections are different from form sections because they are connected to a specific

smart section of the associated business object. A smart section can have actions a person can click to control what records are referenced by the connected smart section.

The specific actions that a smart section can offer and the way it displays the contents of the smart section depends on whether it is a single-record smart section or a multiple-record smart section. The basic setup and properties for smart sections are the same whether they are connected to a single-record smart section or a multiple-record smart section. The aspects of smart sections that are independent of the kind of connected record smart section are discussed in the following paragraphs. The details of laying out fields in a single-record smart section are discussed in "Form field properties". The details of laying out fields in a multiple-record smart section are discussed in "Multiple-record smart sections".

Like other kinds of sections, you can create a smart section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Smart Section*.

The next thing to do is select the smart section you want connected to this section. The names of smart sections in the associated business object appear in the drop-down list for the *Data Section* property. When you select the value for the *Data Section* property, the value of the *Name* and *Label* properties are set to the name of the smart section.

Tip: Ensure that the business object for the smart section has a default form specified. You can specify the default form by using the Form Properties box in the Form Builder.

There is a shortcut that can get you to this point in the creation of a smart section in fewer steps. When you select a tab in the *Navigation* panel, the *Components* panel contains a list of the smart sections in the business object that have not been connected to a smart section in this form.

You can use the list of smart sections in the *Components* panel to create all the smart sections you want in a single step. Check the check box of each smart section you want to use to create a smart section, then click the *Add* action at the top of the *Components* panel. That's all there is to it.

When you create a smart section you may not need to add any fields to it. A smart section starts out with a data field for each of the fields in the smart section it is connected to. You can delete any of these fields. If you change your mind, you can add them back in.

"Common section properties" includes information about the following properties in a smart section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a smart section that are not described in "Common section properties" follow:

Fixed Data Cols

If you would like some of the left columns in the display to be fixed and not move as the display is scrolled horizontally, enter the number of columns in Fixed Data Cols. You can identify the fixed columns because they are to the left of a thicker grey demarcation line. The default value is 0 (zero). Performance degrades as the number of fixed columns increases. Set Fixed Data Cols to no more than 5.

Hierarchical Fields

> This property is visible only for smart sections that are backed by a Vertical Table business object section. Vertical sections are described in "Record organization".

> When selected, fields in vertical table sections have + and - icons that allow the user to expand parent fields to show child fields below or to collapse the display. A field has a parent when its Hierarchical Parent property identifies a field, as described in "Form field properties".

Comparison

> This property is visible only for smart sections that are backed by a Vertical Table business object section. Vertical sections are described in "Record organization".

> When selected, the Comparison property indicates that the section will compare numeric field values of the records in the section to the same field in a compare to record. The number type fields that have the compare ability are Number, Financial Rollup, and Classification Rollup fields. The compare to record also must be a record in the section and is designated by an association in the Comparison Association property.

> A comparison vertical section displays all records that are compared to the compare to record with a delta value next to the actual value of the field. The delta value is configured in the Delta Type property as either a numeric difference, a percent difference, or as not displayed. If the compare to value has a different UOM than the compare from value and a conversion is available between the two UOMs, the compare from value is converted first.

> If there are fewer than two records in the section, the platform displays the section without the comparison functionality.

> The user will be able to change the compare to record during runtime. When the user changes the compare to record, the platform refreshes the comparison vertical section showing updated delta values and updates the association. The order of the records in the section does not change.

> The compare to record can be changed during workflow by removing the previous association and associating a new compare to record.

Comparison Association

> This property is visible only when the Comparison property is selected. The association designates the compare to record in a comparison vertical section. This association is a temporary association.

> The drop-down list contains all associations that associate to the same business object that is referenced by the section and that are defined on the business object of the form. The list does not include the association that the section uses.

> The compare to record must be in the smart section and there should be at most one associated record at a time.

> If no association is found, the platform uses the first record in the section as the compare to record. If more than one record is associated with the Comparison Association, the platform sorts the records alphabetically by their names and uses the first record from the sorted list.

Print Preview Link

This property is visible only on vertical table sections. When selected, a Print Preview link shows on the vertical table section bar.

At runtime, when a user selects the Print Preview link, the platform displays the vertical table section in a new window. The display is read-only and retains the expand/collapse state of the fields in the original section.

After you have finished setting up the properties of a section, you may want to set up actions that can be performed within the section. Section actions are described in "Section actions".

Source details hyperlink

Below a smart section's real properties is what looks like a property with no value and the label *Source Details...* This label is a hyperlink. Clicking this hyperlink causes a window to pop up that displays information about the underlying smart section.

Autocomplete in smart sections

Autocomplete functionality is available for all single-record smart section text fields. Specify whether or not your users will have this feature in the USE_AUTO_COMPLETE_IN_SMART_SECTION property in the TRIRIGAWEB.properties file. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*. The default is for autocomplete to be enabled. The autocomplete functionality is available on locator fields, as described in "Autocomplete in locator fields".

Multiple-record smart sections

Laying out the fields in a smart section connected to a multiple-record smart section is different from laying out a single-record smart section. Instead of laying out individual fields you lay out columns of fields.

Laying out a multiple-record smart section is simpler than laying out a single-record smart section because you do not have to specify the position of each field. All you have to do is specify the order in which the columns will appear. The order in which the columns appear is determined by the value of each field's *Start Row* property. The values of the *Row Span*, *Start Column* and *Col Span* properties are ignored.

In the properties of the connected business object, if the check box for the *Vertical Section* property is checked, you will be laying out rows of fields instead of columns. The *Vertical Section* property of a business object is discussed in "Record organization".

One other difference between a single-record smart section and a multiple-record smart section is that you cannot add form fields to a multiple-record smart section. A multiple-record smart section can only contain data fields.

In a multiple-record smart section, the properties of fields not related to their layout are interpreted the same as in other kinds of sections. For descriptions of these properties, see "Form field properties".

Multi tab sections

Multi Tab sections contain tabs. The tabs in a Multi Tab section are different from top-level tabs in that they contain exactly one section.

Like other kinds of sections, you can create a Multi Tab section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Multi Tab Section*.

"Common section properties" includes information about the following properties in a Multi Tab section: Type, Name, Label, Title Bar Color, Visible, Expand Section, Read Only, Style Class, Start Row, Row Span, Start Column, and Col Span.

Once you have saved a Multi Tab section for the first time, you will see an *Add Section* action. The way that you add a tab to a Multi Tab section is to add a section. When you add a section to a Multi Tab section, it shows up in its own tab that has the same label as the section.

Since you can add any section to a Multi Tab section (except for another Multi Tab section), the properties of the section that is displayed override the Multi Tab section's properties. The only exception is the Visible property. If the Visible property of a Multi Tab section is unchecked, the entire Multi Tab section will be hidden.

Query sections

A Query section looks similar to a multiple-record smart section. A query section contains the records found by a query. A query section has no direct connection with a business object. Queries are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Like other kinds of sections, you can create a query section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Query Section*.

After you have set the value of the *Type, Name, and Label* properties, the next property to set the value of is the *Query* property. The value of the *Query* property is used to determine which query will be run to generate the contents of this query section. What is run to generate the contents of this section does not actually have to be a query. It can be any type of report managed by the Report Manager (discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*).

A query section is always viewed within a parent record. If the record is a capital project, the active project for that query section is the record itself. If the record is not a capital project, the active project for the query section is the project that the parent record is in. Whether or not the active project is used in the query section results is dictated by the data scope on the query definition. If the data scope of the query definition is Active Project, the records returned by the query are restricted to the active project.

Queries attached to query sections are not run in the Form Builder. Instead the layout displays a label identifying the section.

When a graphics section is associated to a query section, the only items that can be selected from the graphics section are those associated items that are derived from the query results.

Query sections are resized when the number of rows returned by the query is greater than the height defined. The platform produces scroll bars so the user can scroll the results. For hierarchical queries, the number of rows refers to the top-level (check box) rows and expanding the top level causes the query section height to expand accordingly. For queries with Group By or Sum, the number of rows refers to data (check box) rows, not the sum rows. If a query section has a Sum column, all rows display, including the sum row, and the user does not see the Show drop-down list from which to select the number of rows to view. A query section with Height = 0 defaults to Height = 10. A query section with Height = -1 defaults to Height = 10.

At runtime, if an alternate form is defined for this query section and the user selects a record in this query, the alternate form displays.

In a form, when a user sorts query results, the data is sorted based on Unicode sorting for all languages. The result may not match the expected behavior for non-US English-language users.

Query section properties

To select a query, click the search icon in the *Query* property. Clicking the search icon causes a *Select Item(s)* panel to pop up over the *Properties* panel.

Initially, no queries appear in the *Select Item(s)* panel because no module is selected. Select the module of the query you want from the drop-down list to the right of the *Module* label. Once the module is selected, the system displays all queries for that module from the Report Manager. If what you are looking for is a query, you will see it in a list.

If what you are looking for is not a query but some other kind of report, you will not see what you are looking for because the list initially includes only queries. You will need to select the type of report you want to see.

To select the type of report you want to see, click the filter icon to the right of the *Reports List* label. From the menu, you can choose the type of report you want to see a list of or you can choose *All* to see a list that contains every kind of report.

The query or report you use in a query section should have an association filter that uses *$$RECORDID$$* or *$$PARENTID$$* (discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*) so that its results only include records associated with the record that contains the Query section.

"Common section properties" includes information about the following properties in a query section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a query section that are not described in "Common section properties" follow:

Query See the explanation of this property in the preceding paragraphs.

Threshold Record Field

This property is visible only when the query specified in the Query property is a metric query. Use this property to provide context sensitive behavior for a metric query displayed in a form.

The Threshold Record Field drop-down list contains locator fields anchored to the Threshold business object. At runtime, the platform uses the threshold record from this locator instead of the threshold record specified in the query within the Report Manager. If the locator does not specify a threshold record, the platform uses the threshold record specified in the query.

Auto Refresh

If you wish the query section to refresh automatically, select the Auto Refresh property.

Refresh Time in secs.

This property is visible only when the Auto Refresh property is selected. Enter the number of seconds the platform should wait between automatic refreshes of the query section.

Workflow

The workflow specified by the value of this property is run when, as a result of a *Find* action, a person selects records to be associated with the record that contains the Query section. *Find* actions are discussed in "Query section actions".

This workflow is typically used to update totals or other statistics about associated records in the record that contains the query section.

If this property has no value, no workflow will be run when a find action is used to associate additional records with the record that contains the query section.

To specify the value of this property, click the search icon. Clicking the search icon causes a list of synchronous workflows to appear. The synchronous workflows in the list will be workflows launched from records created from the business object associated with the form that the query section is part of. If you select one of the workflows in the list and click the *OK* action at the top of the list, the workflow you selected becomes the value of this property.

If you want to clear the value of this property so that it has no value, click the X icon.

Association Type

The value of this property is the name of the association this section will be based on. If you use the *Find* or *Add* action that the IBM TRIRIGA Application Platform may provide for this query section, the found records or newly created record will become associated with the record that contains this section using the specified association name.

These platform-provided actions are described in "Query section actions".

Select Association Type

Workflows may be run as a result of someone clicking a form state transition action (described in "State-based actions"). It may be helpful for the workflow to be able to recognize if the person who clicked the action that triggered the workflow first selected some records displayed in the query section by checking the check box to the left of the record.

If an association name is selected for the value of this property and someone clicks an action you defined that triggers a workflow, the platform will create an association from the record associated with the form to the any records shown in this section that have their check box checked.

There is a different mechanism that allows a workflow run from an action in a query section to navigate to selected records. This is discussed in "Query section actions".

Having the same Select Association Type on multiple query sections in the same form may result in undesired select behavior as those sections are sharing the same select associated records. Use a different Select Association Type for each query section to properly maintain each select association.

Temp Select Association
When selected, the query section association is temporary. When not selected, the query section association is permanent.

Alternate Form List
This property identifies the alternate form list to be used to display a record when that record is opened from this query section. The alternate form used to display the record depends on the module, business object, and form defined for the record.

Single Select

If this property's check box is *not* checked, it means that a user may select the check box for any or all of the records in the Query section.

If this property's check box *is* checked, it means that a user may select only one of the records in the query section.

Select Workflow

This property is only displayed if the Single Select property is checked. The workflow specified by the value of this property is run when a user selects the radio button of a record contained in the query section.

If this property has no value, then no workflow will be run when a radio button is clicked.

To specify the value of this property, click the search icon. Clicking the search icon causes a list of synchronous workflows to appear. The synchronous workflows in the list will be workflows launched from records created from the business object associated with the form that the query section is part of. If you select one of the workflows in the list and click the *OK* action at the top of the list, the workflow you selected becomes the value of this property.

If you want to clear the value of this property so that it has no value, click the X icon.

Note that the Select association created when the user clicked the radio button will be deleted after the workflow(s) complete.

Show Add
If this property's check box is checked, then there will be an *Add* action for people to click at the top of this query section. The *Add* action is described in "Query section actions".

Show Query Header

If this property's check box is checked, then column headings for the query results will be displayed in the query section. This check box is usually checked. If this check box is not checked then no column headings will be displayed in this query section.

Reserve query sections

To implement a Reserve query section in a form, several fields are required in the parent business object and record.

For non-recurring scenarios, the following fields are required:
- triStartDT
- triEndDT
- triTimeZonesCL

For recurring scenarios, the following fields are required:
- triStartDT
- triEndDT
- triTimeZonesCL
- triRecurrenceRuleTX
- triRecurrenceIdTX
- triRecurrenceExceptionTX
- triRecurrenceRuleTimeZoneCL

For more information about recurring events, see "Calendar and time-based events".

To enhance performance in either scenario, the following fields are optional:
- triAvailabilityThresholdNU
- triAvailabilityResultsLimitNU

For more information, see "Availability sections".

Report sections

A Report section allows a form report to be presented as a section in a form. Form reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Like other kinds of sections, you can create a Report section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Report Section*.

After you have set the value of the *Type* property, the next property you will probably want to set the value of is the *Document* property. The value of the *Document* property is used to determine which form report template will be used to generate the contents of this report section.

To select a form report template, click the search icon in the *Document* property. Clicking the search icon causes a *Select Item(s)* panel to pop up over the *Properties* panel containing a list of the documents in the Document Manager.

To specify which document to use as the form report template, select the document's radio button and then click the *OK* action at the top of the list.

After you have set the value of the *Document* property, you may want to set the value of some of the other properties. Here are descriptions of the properties of a report section:

"Common section properties" includes information about the following properties in a Report section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a Report section that are not described in "Common section properties" follow:

Document
> See the explanation of this property in the preceding paragraphs.

Financial Report
> If the values in this report are or depend on Financial Rollup tokens, you should check this property's check box. Financial Rollup tokens are discussed in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

Graphics sections

Graphics sections are a specialized kind of section used to support CAD-related and other graphics for IBM TRIRIGA applications. The details of how to put content in a graphics section are not described here. However, the following paragraphs explain how to create a graphics section.

Like other kinds of sections, to create a graphics section, select the tab that will contain the section and click the *Add Section* action. Then set the value of the *Type* property to *Graphics Section*.

At runtime, if an alternate form is defined for this graphics section and the user selects a record in this section, the alternate form displays.

"Common section properties" includes information about the following properties in a graphics section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a graphics section that are not described in "Common section properties" follow:

Query Select the query that defines the data to display in this graphics section.

Module
> For the value of this property you may select either *Geography*, *Location*, or *Organization*. This determines to which of these hierarchies the images are connected.

Association Type
> The value of this property is the name of the association that connects the record this form is editing to the record the graphic is associated with.

Alternate Form List
> This property identifies the alternate form list to be used to display a record when that record is opened from this graphics section. The alternate form used to display the record depends on the module, business object, and form defined for the record.

Default Theme

Every graphics section must have a theme. The Default Theme determines which theme is used when a graphics section is rendered for the first time in a form. If this property does not have a value, IBM TRIRIGA will use the Global Default Theme as the theme of this section when the section is rendered initially. The Global Default Theme is named triDefault and is defined in Tools > Administration > Graphics > Theme.

You may set this property to specify a different Default Theme for this section. For more information on themes, see the *IBM TRIRIGA Application Platform 3 Graphics User Guide*.

Lock Theme

When this property is selected, the theme and associated label and report on the graphics section cannot be changed by the user. The graphics section is displayed to the user with the default theme specified in the section properties, or, if none is specified, with the Default field in the theme list. When a user opens a form containing a graphics section, the values for Theme, Labels, and Report are greyed out and the label type cannot be changed.

Use this property to avoid the high data traffic caused when the user changes the theme or related fields.

Auto Zoom

This property enables the automatic zoom of the graphic when the graphic is rendered. Note that the related Zoom (%) and Zoom Delay properties are always displayed in the Form Builder properties for the Graphics Section type.

Note: The automatic zoom is only applied on the initial load of the graphic for a record session. For example, the zoom is not reapplied when you refresh the form tab.

Note: The automatic zoom is only applied when a single highlighted entity exists in the drawing. For example, the zoom is applied when configuring a graphics section by association, or linking to a query section that has one result. The zoom is not applied if the graphics section is linked to a query section with multiple results.

Zoom (%)

If Auto Zoom is enabled, this property sets the automatic zoom of the graphic. The default is 100%, but the value can be edited to change the scale factor of the zoom. For example, 50% will be twice as far, and 200% will be twice as near. The value must be between 1 and 400.

Note: The Zoom percentage is based on the bounding box of the entity that is being zoomed, where 100% equals the exact size of the bounding box of that shape.

Zoom Delay (seconds)

If Auto Zoom is enabled, specify the amount of time in seconds to wait before Auto Zoom occurs. This property allows the user to configure the time to see the full context of the graphic before the zoom occurs. The value must be between 0 and 99.

Note: The Zoom Delay will recognize how long a graphic has taken to load and count that time as part of the delay. For example, if Zoom Delay = 3 but the graphic takes 10 seconds to load, the Auto Zoom will occur immediately after loading is completed.

Expand Section

This is selected by default. If this check box *is* selected, the first time a user uses this form, this section will be expanded by default. If *not* selected, the section will be collapsed by default. Note that for subsequent use of this form, IBM TRIRIGA remembers the state of each section for each user.

Tip: Graphics sections do not load if the section is collapsed, which can have a significant positive impact on the time to load the section.

When a graphics section is associated to a query section, the only items that can be selected from the graphics section are those assigned items that are derived from the query results.

After filling in these properties, click the Apply action at the top of the Layout tab. Additional properties are now available.

Display

This property indicates the source of the data displayed in the graphics section. The values in the drop-down list are as follows:

Self This value means that the data is from the business object of this form.

By Association

This value means that the source is by association. The Association Type and the Associated Query Section properties define the association.

Associated Query/Availability Section

This property is used to link a query section or an availability section to the graphics section. Select the query section or the availability section from the drop-down list. One of the choices is -None-.

Actions Section

Graphics sections can have section actions. Read about section actions in "Actions".

Graphics sections highlight data in query and availability sections

You can to link a query section, an availability section, or both to a graphics section. In order to do this, you will need to first add a query section, or an availability section, or both to the form. Once this is done, those sections are available in the Associated Query/Availability Section property.

When a graphics section is linked to a query section, the spaces returned in the query are highlighted in the graphic. When the user selects records in the query section, those records are highlighted in the graphic, and vice versa.

When an availability section is linked to a query section, the spaces returned by the query in the availability section are highlighted in the graphic. The spaces are highlighted with the value of the Selectable property in the theme record. The user can only select these selectable spaces, either from the graphic or from the availability section.

After an availability section is linked to a graphics section, when the user selects spaces in the graphics section, the spaces are also selected in the availability section, and vice versa.

If either the graphics section or the availability section is configured for single-select, the other section respects this restriction even if it is not set as single-select.

When a query section is linked to a graphics section, the graphics section reflects runtime filters applied by the user to the query section. However, when an availability section is linked to a graphics section, the availability section ignores runtime filters applied by the user to the query section.

The primary use of this feature is to associate a floor plan and the availability of the resources on the floor plan to a query that returns spaces. The spaces on the floor plan and resources in the availability section that are also in the linked query section are highlighted.

Using this feature requires defining the context of the query results, which is used to determine the floor plans and resource availabilities associated with the query results. In order to define the context, you set the Display property to By Association and set the Module and Association Type properties appropriately in the graphics section. For example, if you want the floor plan from the Parent Floor of the spaces in the query results, set the Module to Location and Association Type to Parent Floor. Note that this association determines which floors appear in the menu in the graphics section. If the query had results of spaces from 10 different floors, then all 10 floors would appear in the menu.

Tip: The primary business object of the query determines the record type of the results. In this example, we used Spaces, but potentially any attached entity (e.g., Assets, People, Organizations) that corresponds to the query results could be used in a linked query or availability section. In order to define the context, an association needs to be available from that record to a record with a floor plan, i.e., Floor.

Tip: When running a graphic report on a graphics section that is linked to a query section or to an availability section, or both, that report only applies colors to the linked query results. Only entities that are in both the linked query results, or the linked availability section, or both, and the graphic report results will be selectable. For more information on graphic reports, see the *IBM TRIRIGA Application Platform 3 Graphics User Guide*.

Excel sections

An Excel section is like a report section in that it displays values from a record using the formatting and context of a Microsoft Excel spreadsheet. It is different from a report section in that you can use it to perform calculations. Values computed by the spreadsheet can be used to update fields in the record associated with the form. Excel sections can also update fields in records that are directly or indirectly associated with the underlying record.

Excel sections are only supported when the user is running Internet Explorer. Excel sections are not supported for any other browser.

The main use for an Excel section is to perform computations that are not easily done within the platform. However, there is a major restriction.

Values computed by the spreadsheet can only be used to update the underlying record when a record's form is being used interactively by a person. Changes to a record that happen outside the interactive environment will not be processed by the Excel spreadsheet. In particular, the spreadsheet is not used to update fields when a workflow creates or modifies a record. Similarly, when records are created or modified by an application that runs outside the IBM TRIRIGA Application Platform, the Excel spreadsheet is not run to compute values.

The values used and updated by an Excel section come from the permanent data in records, not the temporary data in a form. So, when a user clicks the Calculate on an Excel section, the platform permanently saves the data and when it is calculated based on a state transition action, it only does so after the data is permanently saved. See "Temporary data" for an explanation of the distinction between temporary and permanent data.

Note: According to Microsoft's website on http://technet.microsoft.com/en-us/library/cc178954.aspx, the OWC library that IBM TRIRIGA uses for Excel sections is not available in Office 2007. That article has a link to where you can download the OWC library and install it as a separate component. If you upgrade Office 2003 or Office 2000 to Office 2007, the OWC library is available. If you have a clean install of Office 2007, the library is missing.

Excel form template

Before you create an Excel section, you must already have created an Excel Form Template like those used for form reports. Form reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

There are some requirements for the form report template you create:
- The form report template must be created using Excel.
- The form report template must have cells that contain formulas to make whatever computations are needed.
- You must have Excel save the form report template as HTML with interactivity enabled. Check the *Add Interactivity* check box is checked. If you do not check this, no computations will be performed by the Excel section.

Excel section properties

Like other kinds of sections, you can create an Excel section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Excel Section*.

After you have set the value of the *Type* property, the next property you will probably want to set the value of is the Excel (.htm) *Document* property. The value of the Excel (.htm) *Document* property is used to determine which form report template will be used to generate the contents of this Excel section.

To select a form report template, click the search icon in the Excel (.htm) *Document* property. Clicking the X icon causes a panel to pop up over the *Properties* panel containing a list of the documents in the Document Manager.

To specify which document to use as the form report template, select the document's radio button and then click the *OK* action at the top of the list.

After you have set the value of the Excel (.htm) *Document* property, you may want to set the value of some of the other properties.

"Common section properties" includes information about the following properties in an Excel section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span.

After you have saved an Excel section for the first time, under the properties is a form section that contains a spreadsheet. Initially the spreadsheet is blank. After you have specified the Excel (.htm) *Document*, the spreadsheet you are using as a template should appear in place of the blank spreadsheet.

Linking Excel results to fields

Procedure

1. After you have specified the Excel section's properties and clicked the *Apply* action to save the properties, the Excel section exists. However, the Excel section is not yet finished. One major step still remains.
2. All that we have done so far is to arrange for values in a record's fields to be presented in a spreadsheet. We still need to arrange for computed result values in the spreadsheet to be copied to fields in the record being edited by the form.
3. Below the properties and above the spreadsheet is an action labeled *Link Cell*. It is used to link spreadsheet cells to fields. If a spreadsheet cell is linked to a field, it means that the value in the spreadsheet cell will be copied to the field(s) it is linked to. If a spreadsheet cell is linked to more than one field, then the value in the spreadsheet cell is copied to all the fields that the spreadsheet cell is linked to. The field must be present in the form for it to be updated.
4. The way we link a spreadsheet cell to fields is to first click the spreadsheet cell we want to link. When we click a spreadsheet cell, the spreadsheet cell becomes highlighted.
5. After we have highlighted a spreadsheet cell, we click the *Link Cell* action to link the spreadsheet cell to field(s). Clicking the *Link Cell* action causes an Excel Field List window to pop up.
6. Initially the spreadsheet cell is not linked to any fields, so the Excel Field List window is empty. We can add a link to a field by clicking the *Add* action. Clicking the Excel Field List window's *Add* action brings up a Formula Tree window that allows you to select a field in the business object that contains the Excel section. The Formula Tree window can also be used to select fields in business objects that are associated, directly or indirectly, with the business object that contains the Excel section. Formula trees are discussed in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.
7. The record number field defaults to 1, which is always the correct value unless the field is in a multiple-record smart section. If the field is in a multiple-record smart section, the *Record Number* value is used to determine which record in the multiple-record smart section contains the field that will be updated. The value 1 indicates the first record.
8. Be careful when linking a spreadsheet cell to a field on an associated business object, as shown in the following figure. When someone is using the spreadsheet to update data, the IBM TRIRIGA Application Platform will follow all specified associations to find the record that contains the field a spreadsheet cell is linked to. If it finds multiple records through the association, then it will update the field in all of the records it finds with the spreadsheet cell value.
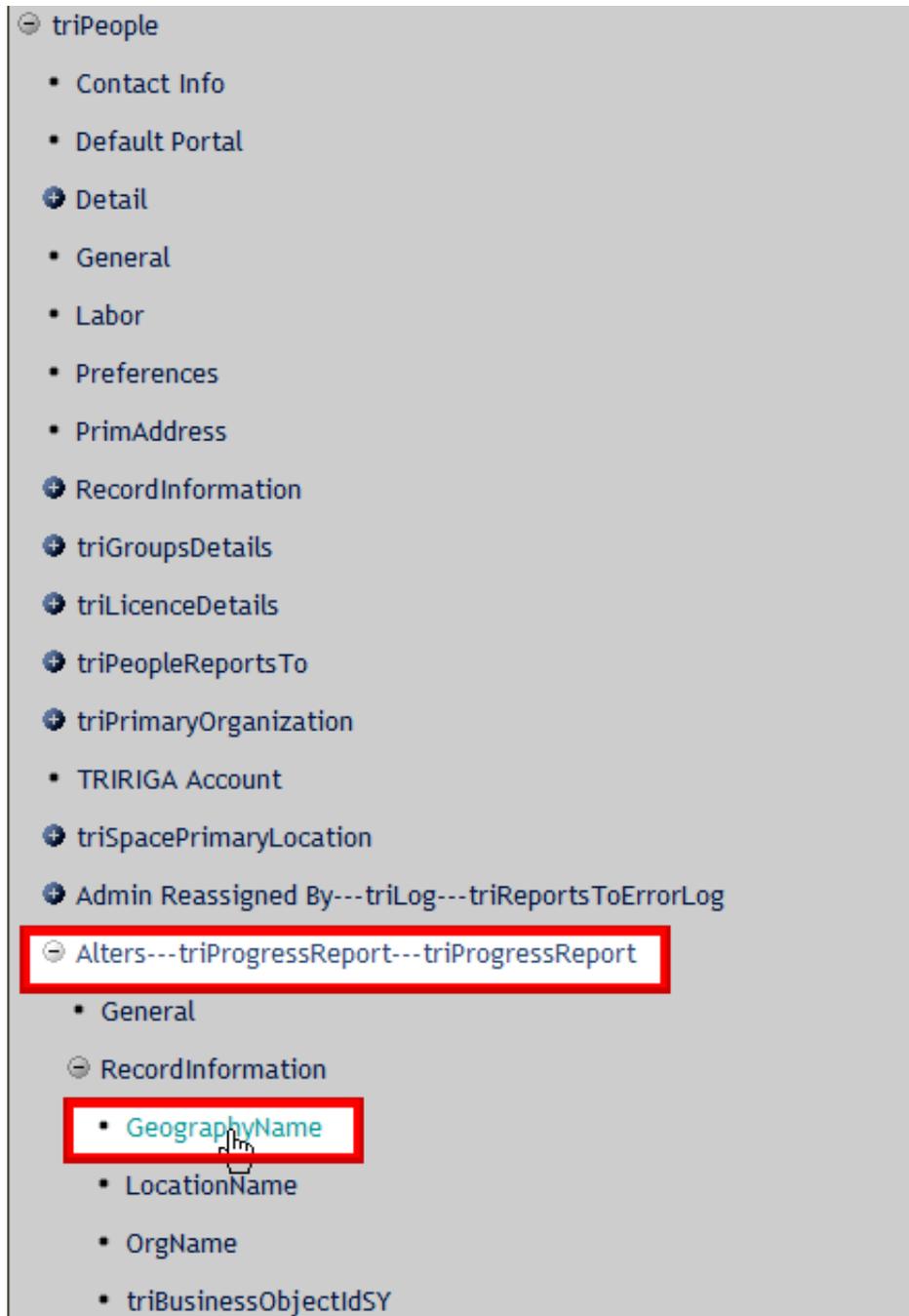
Figure 34. Result when cell is linked to associated business object field

Gantt sections

A Gantt section is used to display a Gantt chart. Gantt sections have a scheduling engine that can be used to set scheduling information in records that represent tasks. This scheduling information includes earliest start dates, latest start dates, constraints, percentage complete, critical path, associated resources, container or umbrella tasks, baseline, and dependent tasks.

The Planned Start and Planned End dates in the Gantt display using the value in the Date Time Format field from the user's My Profile record. The time scale in the hourly view displays a 24 hour time format, and above the hours, the date, time, and day display based on the value in the Date Format field of the user's My Profile record.

The task table in the Gantt section shows specified fields in records that represent the tasks displayed by the Gantt chart. The fields that display are defined by the query that is specified for the Gantt section. The Gantt chart in the Gantt section displays a Calendar view of the task records; the tasks display as bars on a timeline. If the Gantt section is not read only, you can use the Gantt section to interactively edit and create records that represent tasks in a project.

You can add a tab in the project form to provide more room for the Gantt section, which is beneficial for larger projects. The Open Gantt in New Window button that was part of the Java applet-based Gantt section in previous versions is not supported. It has no function in the current JavaScript based DHTML component new Gantt section and has therefore been disabled. To add a Gantt only tab, do the following steps:

1. Navigate to the form for the record type that you want to modify under Tools > Form Builder. Select the module, then select the form, and click Revise. For example, to add a tab for Capital projects, select triProject for the module and select triCapitalProject for the form.
2. Select the tab that contains a Gantt section. For example, the Gantt section in the Capital Project form is on the Schedule tab, so select the triSchedule tab.
3. Select the Copy Tab action.
4. Enter a name for the new tab, such as 'GanttOnly' and click Apply. No changes are needed for the Target Business Object or Target Form.
5. After clicking Apply, you might need to refresh the form by clicking the name of the form in the Navigation panel.
6. Expand the new tab in the Navigation panel.
7. Select each section that you do not want to display in the new tab and delete them one at a time by using the Delete action.
8. Adjust the height of the Gantt section by selecting the section and editing the Height parameter.

The Form Builder will allow you to put a Gantt section in any form. However, for the Gantt section to work properly, the business object associated with the form must contain these fields in its General section:

triProjectPlanStartDA
> This is the project's planned start date.

triProjectPlanEndDA
> This is the project's planned end date.

triProjectCalcStartDA
> This is the project's calculated start date. This should be a read only field because the scheduling engine calculates the value.

triProjectCalcEndDA
> This is the project's calculated end date. This should be a read only field because the scheduling engine calculates the value.

triTimeZonesCL
> This is the time zone of the project.

triProjectCalcFromLI

> This establishes scheduling engine calculation settings. The values in the drop-down list are `Start` and `End`. The `Both` value is deprecated.

Tip: Prior to opening the Gantt chart, any unsaved changes to the following fields should be saved so that the project can be scheduled correctly: Plan Start, Plan End, and Calculate Project From. This applies only to the fields listed.

In addition to needing a business object that contains these fields, a Gantt section also needs a query that determines which records and which of their fields will be displayed in the Gantt section. The records include but are not limited to contract review tasks, facility assessment work tasks, inspection tasks, punchlist tasks, schedule tasks, submittal tasks, and work tasks.

A query section will automatically update based on changes saved in the Gantt section. If the query section is a standalone query section, the name of the section must contain the word "task." If the query section is within a multi tab section, the query section name must contain the word "taskMilestone."

One last thing needed to set up a Gantt section is an association name to use for records that are interactively created through the Gantt section.

"Common section properties" includes information about the following properties in a Gantt section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a Gantt section that are not described in "Common section properties" follow:

Query

> The value of this property is the query used by this section to determine which records and which of their fields to display. This also determines which forms can be selected while creating tasks from within the Gantt section.
>
> The details of how to specify the query that will be the value of this property are described in "Query sections".
>
> **Tip:** Once tasks are added to a Query section and loaded in the Gantt, the Gantt scheduling engine schedules (or arranges) the tasks. After the tasks have been scheduled, the start and end dates on the Gantt may be different from those on the Query section. These dates will coincide once the user either clicks the Save button on the Gantt or saves the record. The save causes the dates from the Gantt section to be propagated to the Query section. The start and end dates on the Gantt should be the Calculated Start and the Calculated End.

Workflow

> The workflow specified by the value of this property is run when a person selects records to be associated with the record that contains the Gantt section.
>
> If this property has no value, no workflow will be run.
>
> To specify the value of this property, click the search icon. Clicking the search icon causes a list of synchronous workflows to appear. The synchronous workflows in the list will be workflows launched from records created from the business object associated with the form that the

Query section is part of. If you select one of the workflows in the list and click the *OK* action at the top of the list, the workflow you selected becomes the value of this property.

If you want to clear the value of this property so that it has no value, click the X icon.

Association Type

When this section is used to interactively create records, an association is created from the record being edited by the form to the new record. The value of this property is the name that is used for the association.

Select Association Type

Workflows may be run as a result of someone clicking a form state transition action (described in "State-based actions"). It may be helpful for the workflow to be able to recognize if the person who clicked the action that triggered the workflow first selected some records displayed in the Gantt section.

If an association name is selected for the value of this property and someone clicks an action you defined that triggers a workflow, the platform will create an association from the record associated with the form to the any records shown in this section.

Allow Delete

The check box for this property is selected by default. When the property is enabled, users can delete tasks in the Gantt section. To disable user ability to delete tasks from the Gantt section, clear the check box.

Note: The Open Gantt in New Window button that was part of the Java applet-based Gantt section in previous versions is not supported. It has no function in the current JavaScript based DHTML component new Gantt section and has therefore been disabled.

Single select Gantt sections

IBM TRIRIGA treats single select Gantt sections differently from other Gantt sections. Checking the Single Select check box makes some additional properties relevant.

The primary purpose of these properties is to support putting a parent Gantt section and a child Gantt section on the same form, and have the parent Gantt chart drive what is displayed in the child Gantt chart. By having a single select Gantt chart be able to run workflows, you can make whatever modifications to the form are needed to enable the child Gantt section to display the appropriate data.

Note: Multi select Gantt sections are not supported for the current JavaScript based DHTML component Gantt functionality. Only single select Gantt sections are supported.

Single Select

If this property's check box is checked, it means that a user may select only one of the rows in the Gantt section. When this property is checked, the form will refresh automatically when the user selects a row.

If this property's check box is not checked, it means that a user may select any or all of the rows in the Gantt section.

Single Select Always Available

This property only displays if the Single Select property is checked.

If this property's check box is checked, it means that a user may select rows in the Gantt section even when the form is read-only.

If this property's check box is not checked, it means that a user may not select rows when the form is read-only.

Select Workflow

This property only displays if the Single Select property is checked. The workflow specified by the value of this property is run when a user selects a row in the Gantt section.

If this property has no value, no workflow will be run when the user selects a row in the Gantt section.

To specify the value of this property, click the search icon. Clicking the search icon causes a list of synchronous workflows to appear. The synchronous workflows in the list will be workflows launched from records created from the business object associated with the form that the Gantt section is part of. If you select one of the workflows in the list and click the OK action at the top of the list, the workflow you selected becomes the value of this property.

If you want to clear the value of this property so that it has no value, click the X icon.

Note that the system deletes the association created when the user clicked the radio button after the workflow(s) complete.

Select Association Type
With Single Select, you can specify a workflow to run when a user selects a row in the Gantt section. If an association name is selected for the value of this property and someone selects a row in the Gantt chart, the platform will create an association between the form containing the Gantt section and the record associated with the selected row in the Gantt. This association will be available to the select workflow when it runs.

Requirements for records in Gantt sections

The types of tasks that may be created in a Gantt section are determined by the query specified for the Gantt section. Only forms that are allowed for the query can be used to create records in a Gantt section.

The forms that can be used with a query are determined by the value of a field named *Form* on the *Description* tab of its specification.

In order to work properly with the Gantt section and its scheduling engine, certain requirements must be satisfied by records used to represent project tasks.

All records that represent project tasks must have these fields:

triNameTX
The name of the task.

triPlannedStartDT
The planned start date and time of the task.

triPlannedEndDT
The planned end date and time of the task.

triPlannedLateStartDT

The planned latest start date and time of the task. This is calculated by the scheduling engine.

triPlannedLateFinishDT

The planned latest end date and time of the task. This is calculated by the scheduling engine.

triBaselineStartDT

The baseline start date and time of the task.

triBaselineEndDT

Planned early end date and time of the task.

triActualStartDT

The actual start date and time of task.

triActualEndDT

The actual end date of the task.

triActualPercentCompleteNU

The task's actual percent complete.

triIdTX

The task's work breakdown structure (WBS) code. This is used for sorting the tasks in the Gantt section.

triFreeFloatDU

The free float duration of the task. This is calculated by the scheduling engine.

triTotalFloatDU

The total float duration of the task. This is calculated by the scheduling engine.

triPlannedWorkingInputHrsNU

The planned number of hours of the task.

triPlannedWorkingInputDaysNU

The planned number of days of the task.

triHoursPerDayNU

The number of hours in a work day of the task. The number of working hours in a day is given by the calendar. If the task has a calendar, the Calendar record has a field named triHoursPerDayNU that gives the number of working hours per day. If the task does not have a calendar, the value defaults to 24 working hours in a day.

Tip: As mentioned in "Gantt sections", the left panel of the Gantt section has fields that correspond to the query set in the Gantt section. Most fields that appear here will be read only. However, the following fields will be directly editable in the Gantt section if included in the query: triNameTX, triSequenceNU, triIdTX, triPlannedStartDT, triPlannedEndDT, triPlannedWorkingHoursNU, triActualPercentCompleteNU.

Tip: If you are experiencing issues with date and time fields displaying the time incorrectly, the following steps may help resolve them:

1. Make sure your operating system is updated and includes the most current time zone patches.
2. Upgrade the Application Server's Java with the appropriate time zone patches.
3. When applicable, make sure the TZ environment variable has DST set.

Container or Umbrella tasks

Some project tasks are called container tasks or umbrella tasks. This means that they are composed of one or more tasks. The tasks inside a container task can run in series or in parallel (be overlaying). A container task is used to break a project into major phases; each container task contains a group of tasks or activities.

If you want a record to represent a container task, create an association between the record that represents the container task and the records that represent the smaller tasks that are in the container task. On the container task side of the association, the name must be *Includes*. On the other side of the association, the name must be *Included In*.

It is possible to interactively make some tasks container tasks by using the Gantt section's indent and outdent features. To create an umbrella task, you indent a task under it or drag a task under it. The task that is indented becomes the child task of the umbrella task. You use the indent feature by selecting a task in the task table of the Gantt section and then clicking the *Indent* button on the Gantt section toolbar. To remove a task from under the umbrella task, you click the *Outdent* button or delete the task.

For this feature of the Gantt section to work properly, there must be a workflow called *triTask - Synchronous - Promote Task* that creates the *Includes / Included In* association.

Another way to create the parent child relationship for the container task is to use the Rollup To Task (triTaskRollupTo) section from the task form. Populate this section with the ID of the task that is the parent of the current task.

Calendars

When scheduling a task, the scheduling engine takes into account the Calendar of the task. The calendar includes information on Working Hours and non-Working Events. Calendars such as these are represented using records created from the business object named *triCalendar* in the *triSetup* module.

The *triCalendar* record must also have an association with the record that describes the underlying calendar. The association on the side of the underlying calendar must have the name *Calendar For*. On the task side of the association, the name must be *Has Calendar*.

Note: If the project has a calendar, all tasks in the project have that calendar when you create them. If you add a calendar to a project after you create tasks, those tasks do not contain the project calendar. You can set a calendar for those tasks individually in each task record.

Dependent tasks

It is common for there to be dependencies between tasks. Some common dependencies are that one task must start after another finishes or that two tasks must start at the same time. Dependencies such as these are represented using records created from the business object named *triDependency* in the *triIntermediate* module.

Two associations from a *triDependency* record to records that represent tasks are used to identify the tasks involved in a dependency relationship. One of the tasks

involved in the dependency relationship is considered to be the predecessor task; the other task is considered to be the dependent task.

The predecessor task must have an association with the *triDependency* record. The name of the association on the predecessor task side must be *Is Predecessor*. On the *triDependency* side, the name of the association must be *Has Predecessor*.

The dependent task must have an association with the *triDependency* record. The name of the association on the dependent task side must be *Has Dependency*. On the *triDependency* side, the name of the association must be *Defines Dependency*.

Some fields of the *triDependency* record must be set to describe the dependency. The required fields are:

triDependencyRelationshipLI
> The value of this field determines the type of dependency that this record represents. The choices available are:

> **Finish-to-Finish**
>> This value means that the finish time of the dependent task depends on the finish time of the predecessor task.

> **Finish-to-Start**
>> This value means that the start time of the dependent task depends on the finish time of the predecessor task. This is the most common type of dependency.

> **Start-to-Finish**
>> This value means that the finish time of the dependent task depends on the start time of the predecessor task.

> **Start-to-Start**
>> This value means that the start time of the dependent task depends or the start time of the predecessor task.

triLagInputDaysNU
> If the value of this field is negative, it is a lead, which means that the successor task can start sooner. If the value is positive, it is a lag, which means the successor task has to start later. Whether this is measured from the start or finish of the predecessor task to the start or finish of the dependent task is determined by the value of the *triDependencyRelationshipLI* field.

triLagInputHoursNU
> If the value of this field is negative, it is a lead, which means that the successor task can start sooner. If the value is positive, it is a lag, which means the successor task has to start later. Whether this is measured from the start or finish of the predecessor task to the start or finish of the dependent task is determined by the value of the *triDependencyRelationshipLI* field.

triLagWorkingHoursNU
> If the value of this field is negative, it is a lead, which means that the successor task can start sooner. If the value is positive, it is a lag, which means the successor task has to start later. Whether this is measured from the start or finish of the predecessor task to the start or finish of the dependent task is determined by the value of the *triDependencyRelationshipLI* field.

Dependencies can be created from within the Gantt section by dragging a line between two tasks, provided there is workflow called *triTask - Synchronous - Create Dependancy from Selected Task* that creates a *triDependency* record with the necessary associations.

Resources

It is often the case that you want to associate resources with tasks. Resources that can be associated with tasks are represented by records created from the *triResource* business object in the *triIntermediate* module. A *triResource* record has a number of fields that can be used to describe the resource it represents. It also has a number of sections that can be used to reference other records that describe the resource being represented. Applications and people have a great deal of leeway in how they use these fields and sections, since neither the Gantt section nor the scheduling engine rely on them.

triResource records have one field that must be filled in. The value of the *triNameTX* field is used as the name of the resource.

To indicate that a resource is needed by a task, a *triResource* record must have an association with the task record. The name of the association on the task side of the association must be *Assigned*. The name of the association on the *triResource* side of the association must be *Assigned To*.

The *triResource* record also must have an association with the record that describes the underlying resource. The association on the side of the underlying resource must have the name *Is Resource For*. On the task side of the association, the name must be *Has Resource*.

Workflow custom tasks that adjust dates in Gantt system

Some of the features employed by the Gantt scheduling system also can be accessed via workflow. For example, there are places in the IBM TRIRIGA applications (tasks, for example) where the ability to adjust dates based on calendar availability is desired. The same logic employed by the Gantt scheduling system can be made available to workflows via Custom tasks.

The following section describes the available classes used by the Gantt scheduling system that also can be called using Custom tasks in a workflow to adjust dates, hours, and durations based on calendar availability for virtually any type of object. In order for these classes to work, a calendar needs to be appropriately associated to the record(s) that will be processed by the workflow that uses these classes. For more information on using calendars see "Calendar and time-based events". For more information on using Custom tasks, see "Custom task". For examples of how these classes have been used by the IBM TRIRIGA applications you may refer to many of the Synchronous On Change workflows in the triTask module that are used to adjust task dates, hours, and durations using the Gantt classes defined below.

The names of the Custom task workflow calls for Gantt sections all start with com.tririga.architecture.web.process.gantt.workflow. The following table lists each Custom workflow call, gives a description of the call, and shows each field and whether it is input or output. A value of Input in a cell means that for that particular custom task, that particular field is used to calculate other fields, labeled Output.

It is important to note that the field names listed below need to be named exactly as listed in the business object that you are using the classes for or the classes will not work correctly. Ignore any hyphens you see in a class name or a field name; the class names and field names in the system do not contain hyphens. Also, remember that a field that is used as an input for one class may be used as an output for another.

CalcActualDuration

Calculates actual duration, working days and hours, total working hours based on actual start and actual end date.

- triActualStartDT = Input
- triActualEndDT = Input
- triActualDU = Output
- triActualWorkingHoursNU = Output
- triActualWorkingInputHrsNU = Output
- triActualWorkingInputDaysNU = Output

-

CalcActualEndDateFromDuration

Calculates actual end date, working hours and days, total working hours based on actual start date and actual duration.

- triActualStartDT = Input
- triActualEndDT = Output
- triActualDU = Input
- triActualWorkingHoursNU = Output
- triActualWorkingInputHrsNU = Output
- triActualWorkingInputDaysNU = Output

CalcActualEndDateFromWkgdays

Calculates actual end date, actual duration, actual total working hours based on actual start date and actual working days and hours.

- triActualStartDT = Input
- triActualEndDT = Output
- triActualDU = Output
- triActualWorkingHoursNU = Output
- triActualWorkingInputHrsNU = Input
- triActualWorkingInputDaysNU = Input

CalcActualEndDateFromWkghrs

Calculates actual duration, actual end date, actual working days and hours based on actual start date and actual total working hours.

- triActualStartDT = Input
- triActualEndDT = Output
- triActualDU = Output
- triActualWorkingHoursNU = Input
- triActualWorkingInputHrsNU = Output
- triActualWorkingInputDaysNU = Output

CalcActualStartDateFromDuration

Calculates actual start date, actual working days and hours, actual total working hours based on actual end date and actual duration.

- triActualStartDT = Output

- triActualEndDT = Input
- triActualDU = Input
- triActualWorkingHoursNU = Output
- triActualWorkingInputHrsNU = Output
- triActualWorkingInputDaysNU = Output

CalcActualStartDateFromWkgdays
    Calculates actual total working hours, actual start date, actual duration based on actual working days and hours and actual end date.
- triActualStartDT = Output
- triActualEndDT = Input
- triActualDU = Output
- triActualWorkingHoursNU = Output
- triActualWorkingInputHrsNU = Input
- triActualWorkingInputDaysNU = Input

CalcActualStartDateFromWkghrs
    Calculates actual start date, actual working days and hours based on actual end date and actual total working hours.
- triActualStartDT = Output
- triActualEndDT = Input
- triActualDU = Output
- triActualWorkingHoursNU = Input
- triActualWorkingInputHrsNU = Output
- triActualWorkingInputDaysNU = Output

CalcBaselineDuration
    Calculates baseline duration, baseline working days and hours, baseline total working hours based on baseline start and baseline end date.
- triBaselineStartDT = Input
- triBaselineEndDT = Input
- triBaselineDU = Output
- triBaselineWorkingHoursNU = Output
- triBaselineWorkingInputHrsNU = Output
- triBaselineWorkingInputDaysNU = Output

CalcBaselineEndDateFromDuration
    Calculates baseline end date, baseline working hours and days, baseline total working hours based on baseline start date and baseline duration.
- triBaselineStartDT = Input
- triBaselineEndDT = Output
- triBaselineDU = Input
- triBaselineWorkingHoursNU = Output
- triBaselineWorkingInputHrsNU = Output
- triBaselineWorkingInputDaysNU = Output

CalcBaselineEndDateFromWkgdays
    Calculates baseline duration, baseline working days and hours, baseline total working hours based on baseline start and baseline end date.
- triBaselineStartDT = Input
- triBaselineEndDT = Output

- triBaselineDU = Output
- triBaselineWorkingHoursNU = Output
- triBaselineWorkingInputHrsNU = Input
- triBaselineWorkingInputDaysNU = Input

CalcBaselineEndDateFromWkghrs
Calculates baseline duration, baseline end date, baseline working days and hours based on baseline start date and baseline total working hours.
- triBaselineStartDT = Input
- triBaselineEndDT = Output
- triBaselineDU = Output
- triBaselineWorkingHoursNU = Input
- triBaselineWorkingInputHrsNU = Output
- triBaselineWorkingInputDaysNU = Output

CalcBaselineStartDateFromDuration
Calculates baseline start date, baseline working days and hours, baseline total working hours based on baseline end date and baseline duration.
- triBaselineStartDT = Output
- triBaselineEndDT = Input
- triBaselineDU = Input
- triBaselineWorkingHoursNU = Output
- triBaselineWorkingInputHrsNU = Output
- triBaselineWorkingInputDaysNU = Output

CalcBaselineStartDateFromWkgdays
Calculates baseline total working hours, baseline start date, baseline duration based on baseline working days and hours and baseline end date.
- triBaselineStartDT = Output
- triBaselineEndDT = Input
- triBaselineDU = Output
- triBaselineWorkingHoursNU = Output
- triBaselineWorkingInputHrsNU = Input
- triBaselineWorkingInputDaysNU = Input

CalcBaselineStartDateFromWkghrs
Calculates baseline start date, baseline working days and hours based on baseline end date and baseline total working hours.
- triBaselineStartDT = Output
- triBaselineEndDT = Input
- triBaselineDU = Output
- triBaselineWorkingHoursNU = Input
- triBaselineWorkingInputHrsNU = Output
- triBaselineWorkingInputDaysNU = Output

CalcLagInputDaysHoursFromWorkingHours
Calculates the lag working input hours and lag working input days based on the total lag working hours.
- triLagWorkingHoursNU = Input
- triLagInputDaysNU = Output
- triLagInputHoursNU = Output

CalcLagWorkingHoursFromInputDaysHours

Calculates the total lag working hours based on the lag working input hours and lag working input days.

- triLagWorkingHoursNU = Output
- triLagInputDaysNU = Input
- triLagInputHoursNU = Input

CalcPlannedDuration

Calculates planned duration, planned working days and hours, planned total working hours based on planned start and planned end date.

- triPlannedStartDT = Input
- triPlannedEndDT = Input
- triPlannedDU = Output
- triPlannedWorkingHoursNU = Output
- triPlannedWorkingInputHrsNU = Output
- triPlannedWorkingInputDaysNU = Output

CalcPlannedEndDateFromDuration

Calculates planned end date, planned working hours and days, planned total working hours based on planned start date and planned duration.

- triPlannedStartDT = Input
- triPlannedEndDT = Output
- triPlannedDU = Input
- triPlannedWorkingHoursNU = Output
- triPlannedWorkingInputHrsNU = Output
- triPlannedWorkingInputDaysNU = Output

CalcPlannedEndDateFromWkgdays

Calculates planned duration, planned working days and hours, planned total working hours based on planned start and planned end date.

- triPlannedStartDT = Input
- triPlannedEndDT = Output
- triPlannedDU = Output
- triPlannedWorkingHoursNU = Output
- triPlannedWorkingInputHrsNU = Input
- triPlannedWorkingInputDaysNU = Input

CalcPlannedEndDateFromWkghrs

Calculates planned duration, planned end date, planned working days and hours based on planned start date and planned total working hours.

- triPlannedStartDT = Input
- triPlannedEndDT = Output
- triPlannedDU = Output
- triPlannedWorkingHoursNU = Input
- triPlannedWorkingInputHrsNU = Output
- triPlannedWorkingInputDaysNU = Output

CalcPlannedStartDateFromDuration

Calculates planned start date, planned working days and hours, planned total working hours based on planned end date and planned duration.

- triPlannedStartDT = Output

- triPlannedEndDT = Input
- triPlannedDU = Input
- triPlannedWorkingHoursNU = Output
- triPlannedWorkingInputHrsNU = Output
- triPlannedWorkingInputDaysNU = Output

CalcPlannedStartDateFromWkgdays

Calculates planned total working hours, planned start date, planned duration based on planned working days and hours and planned end date.

- triPlannedStartDT = Output
- triPlannedEndDT = Input
- triPlannedDU = Output
- triPlannedWorkingHoursNU = Output
- triPlannedWorkingInputHrsNU = Input
- triPlannedWorkingInputDaysNU = Input

CalcPlannedStartDateFromWkghrs

Calculates planned start date, planned working days and hours based on planned end date and planned total working hours.

- triPlannedStartDT = Output
- triPlannedEndDT = Input
- triPlannedDU = Output
- triPlannedWorkingHoursNU = Input
- triPlannedWorkingInputHrsNU = Output
- triPlannedWorkingInputDaysNU = Output

Tip: If you are using the Gantt scheduling Custom workflow tasks on a server with no graphics card, start the Application Server with the following arguments to avoid java.awt.HeadlessExceptions when calling the tasks: -Djava.awt.headless=true

Availability Legacy sections

The Availability Legacy section uses a Java applet. Note that many browsers no longer support Java applets. For more information, see Java Applet Support on the IBM TRIRIGA wiki.

An Availability Legacy section is used to show the availability of resources over a period of time.

The left side of an availability legacy section shows attributes of some resources. Which resources are shown in the left side and which of their attributes appear in the left side is determined entirely by the query that is used by the availability legacy section.

The right side of the availability legacy section shows the availability of the resources over time. The availability of the resources is determined by a free/busy status table that is internal to the IBM TRIRIGA Application Platform. There is no way to directly manipulate the information in the free/busy status table by editing records. Instead, the contents of the free/busy status table are manipulated through custom Java workflow tasks that are described in "Workflow custom tasks that adjust free status".

The right side of the availability legacy section contains a green and a red line that you can drag left or right. The green line is called the start line and the red line is called the end line. These lines are intended to allow a person to interactively indicate that they want to work with a particular range of time. However, there is no significance for them built into the availability legacy section. It is up to workflows to use the position of these lines in a way that is useful.

Requirements for the form and underlying business object

For an availability legacy section to work properly, the business object that the form is based on must have two Date and Time fields named *triPlannedStartDT* and *triPlannedEndDT*.

These two fields are used by the availability legacy section's internal logic. Their temporary values correspond to the current positions of the green start and the red end lines. Applications should not attempt to use these fields for other purposes.

Because of the way availability legacy sections use *triPlannedStartDT* and *triPlannedEndDT* fields, forms should not contain more than one availability legacy section. If a form contains more than one availability legacy section, their use of these fields will cause them to interfere with each other.

Availability legacy section properties

Like other kinds of sections, you can create an availability legacy section by selecting the tab that will contain the section and clicking the *Add Section* action. You then set the value of the *Type* property to *Availability Legacy Section*.

"Common section properties" includes information about the following properties in an availability legacy section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in an availability legacy section that are not described in "Common section properties" follow:

Query

>
> The value of this property is the name of the query that will be used by this section to determine which records will be displayed. The direct results of this query are displayed in the left side of the availability legacy section.
>
> The information in the free/busy status table related to the records retrieved by the query is displayed on the right side of the availability legacy section, unless a value is selected for the *Association Type* property. See the description of the *Association Type* property for more information.
>
> The details of how to specify the query that will be the value of this property are described in "Query sections".

Workflow

>
> The value of this property is the name of a synchronous workflow that will be run when a person interactively moves the availability legacy section's green start line or its red end line. Workflows run this way will usually examine the temporary value of the *triPlannedStartDT* and *triPlannedEndDT* fields to determine the current setting of the green start line and the red end line.

The synchronous workflows in the list will be workflows launched from records created from the business object associated with the form that the availability legacy section is part of.

Association Type

If no value is specified for the *Association Type* property, the free/busy status information displayed in the right side of the availability legacy section is for the records returned by the query specified by the *Query* property.

If a value is selected for the *Association Type* property, it is the name of an association. If an association name is specified, then the availability legacy section expects each record returned by the query to have the named associated with exactly one record. The right side of the availability legacy section shows the free/busy status information for the associated records.

The usefulness of the *Association Type* property is described below.

Free/busy status for different kinds of resources

It is common to use an availability legacy section to display the free/busy status of different kinds of resources. For example, you may need to display the free/busy status of people, conference rooms, and equipment in the same availability legacy section.

Though records a query returns need not be created from the same business object, they all must be from the same module. To get around this limitation, use the availability legacy section's *Association Type* property.

Have the query return records that summarize real resource records. If each of these summary records has the association specified by the *Association Type* property to a real resource record, then the availability legacy section will display the free/busy status for resource records without caring what kind of records they are.

Interactive edits of availability legacy sections

An availability legacy section does not provide any built-in way to edit any of the information it displays. It is the responsibility of application developers to provide appropriate actions and workflows to perform any appropriate editing.

Workflow custom tasks that adjust free status

Workflows are able to manipulate the free/busy status table through Custom workflow tasks. A Custom workflow task allows a workflow to access a part of the IBM TRIRIGA Application Platform internal logic. There is a general discussion of Custom workflow tasks in "Custom task".

Something common to all Custom workflow tasks is that they are given a single record to work with. Using the record's fields and associations, the Custom workflow task must be able to do whatever it needs to do.

The pieces of internal logic that a Custom workflow task uses to manipulate the free/busy status table all expect the same organization for the record that is passed to them. The expected organization is shown in the following figure.
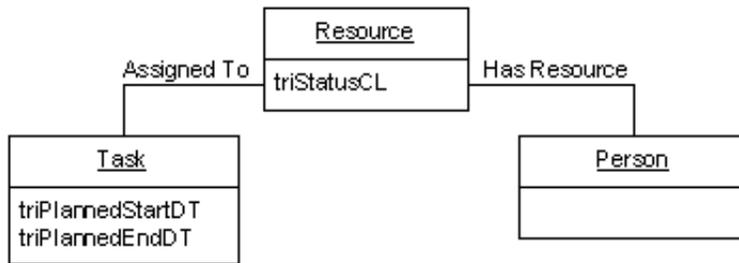
Figure 35. Organization of records manipulating free/busy status

The names shown for the three records in the above figure are not record types. The names just indicate the role that the record plays in manipulating the free/busy status table. The three records can be created from any business object, so long as they satisfy the following requirements.

The record labeled *Resource* is the record that is passed to the Custom workflow tasks. The *Resource* record must have a Classification field named *Resource*. The root classification of the *triStatusCL* field should be *Status*.

The *Resource* record must have an association named *Has Resource* with exactly one other record. In the above figure, this associated record is labeled *Person*. This *Person* record should be the record that the availability legacy section will use for its right side.

The *Resource* record must also have an association named *Assigned To* with exactly one other record. In the above figure, this associated record is labeled *Task*. This *Task* record must have Date and Time fields named *triPlannedStartDT* and *triPlannedEndDT*. The internal pieces of logic used by the Custom workflow task use the *Task* record to identify a single chunk of time that the *Person* record is associated with a particular free/busy status. *Task* records correspond to the colored rectangles that appear on the right side of an availability legacy section.

The piece of internal logic that a Custom workflow task will use is determined by the value of its *Class Name* field. Here are descriptions for the pieces of internal logic a Custom workflow task can use to manipulate the free/busy status table:

com.tririga.architecture.web.process.gantt.workflow.AddResourceFreebusy
> Adds the free/busy information corresponding to the *Task* record for the *Person* record. The beginning and end of the chunk of time will come from the *Task* record's *triPlannedStartDT* and *triPlannedEndDT* fields. The status of this chunk of time in the free/busy status table will be *Tentative*.

com.tririga.architecture.web.process.gantt.workflow.RemoveResourceFreebusy
> Removes the free/busy information corresponding to the *Task* record for the *Resource* record.

com.tririga.architecture.web.process.gantt.workflow.UpdateResourceFreebusy
> Changes the start and end time for the free/busy information corresponding to the *Task* record for the *Person* record. The new beginning and end of the chunk of time will come from the *Task* record's *triPlannedStartDT* and *triPlannedEndDT* fields.

com.tririga.architecture.web.process.gantt.workflow.SetResourceFreebusy
> Changes the status for the free/busy information corresponding to the *Task*

record for the *Person* record. The new status will come from the *Resource* record's *triStatusCL* field. The value must be *Accepted*, *Tentative*, *Declined*, or *Cancelled*.

Stacking sections

The Stacking section uses a Java applet. Note that many browsers no longer support Java applets. For more information, see Java Applet Support on the IBM TRIRIGA wiki.

A Stacking section enables viewing of current or planned assignments of demand (for example, organizations) to supply (for example, floor area and capacity) and allows changes to be made by moving some of the demand from one location to another. A stacking section graphically displays supply and demand data in a usable way and allows a user to manipulate it by moving demand (for example, organization allocations) from one location to another (supply capacity) by picking one or more demand blocks and placing them in a different supply location.

IBM TRIRIGA uses stacking sections in the IBM TRIRIGA Strategic Facility Planning product. The following figure shows a stacking section from IBM TRIRIGA Strategic Facility Planning. In IBM TRIRIGA Strategic Facility Planning, demand is required area or capacity, supply is assignable area or capacity, and capacity is headcount per location. Using this stacking section is described in the *IBM TRIRIGA 10 Strategic Facility Planning User Guide*.
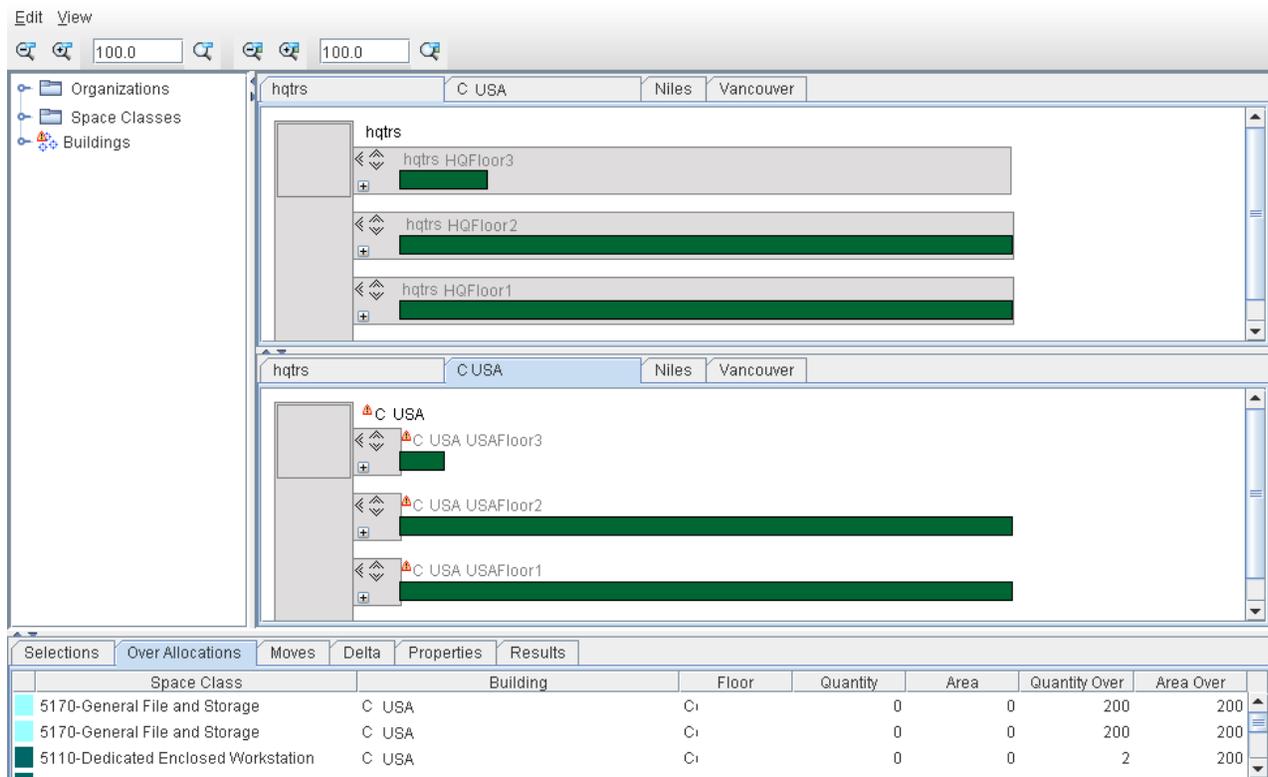

Figure 36. Example of a stacking section

To create a Stacking section, select the tab that will contain the section and click the Add Section action. In the Properties panel for the new section, set the value of the Type property to Stacking Section.

When you finish defining the properties of the stacking section, click the Apply action. To improve performance, the platform displays a tan panel with the words "-Stacking Section-" in the middle.

"Common section properties" includes information about the following properties in a Stacking section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a Stacking section that are not described in "Common section properties" follow:

Pre-Move Workflow

> When the user performs a place operation, the pre-move workflow identified here runs, if one is configured. This workflow determines whether the move is allowed. If an operation is denied, a message is displayed to the user and the demand block is not moved. The primary record for the workflow is the record for the form containing the Stacking section.

> To specify a workflow, click the search icon and choose from the list presented.

Association Type
> Select the association the section should use to access the triStackControlData record from the primary record.

Stack implementation

The stack data provides the information to the control about what is being stacked, the current allocations, and the current capacity.

The main stack data business object is triStackControlData. The stack data must be associated to the scenario record (the record for the form the stacking control is contained in) by the application before the stacking control can function. The stack data provides the UOM to display with the area measurements (it is assumed that all area values in the stacking data use the same UOM) and has a number of associations to records that contain specific fields to provide the other stacking information. To provide the flexibility to stack different types of things, specific business objects are not required beyond the triStackControlData business object. Instead the control requires specific fields to exist on the records and the records to be associated with specific system-defined associations.

The associations and record field requirements are in the following table.

Table 21. Stacking section associations and record field requirements

Association	Information for	From / To record	Required fields on record
<Configured>	*Top-level stacking data*	Primary / triStackControlData	Must be triStackControlData record
Stack Supply Category	*Supply categories*	triStackControlData / Supply-Category records [1-n]	triIdTX, triNameTX, triReportingNameTX, triSequenceNU, triColorCO
Stack Demand Category	*Demand categories*	triStackControlData / Demand Category records [1-n]	triIdTX, triNameTX, triReportingNameTX, triSequenceNU, triColorCO

Association	Information for	From / To record	Required fields on record
Stack Container	*Containers*	triStackControlData / Container records [1-n]	triIdTX, triNameTX, triReportingNameTX, triSequenceNU, triAreaNU, triCapacityNU, triImageIM
Stack Tray	Trays	Container records / Tray records [1-n]	triIdTX, triNameTX, triReportingNameTX, triSequenceNU, triAreaNU, triCapacityNU
Stack Supply Block	*Supply blocks for trays*	Tray records / Supply Block records [0-n]	triSupplyCategoryTX, triSequenceNU, triAreaNU, triCapacityNU, triUnassignableBL
Stack Demand Block	*Demand blocks for trays*	Tray records / Demand Block records [0-n]	triSupplyCategoryTX, triDemandCategory TX, triSequenceNU, triAreaNU, triQuantityNU, triUnmovableBL

Group by sections

A Group By section shows data in a tabular grid-like format grouped by the values of a specified field. The following figure shows an example of a Group By section used in a Space Forecast Survey form.
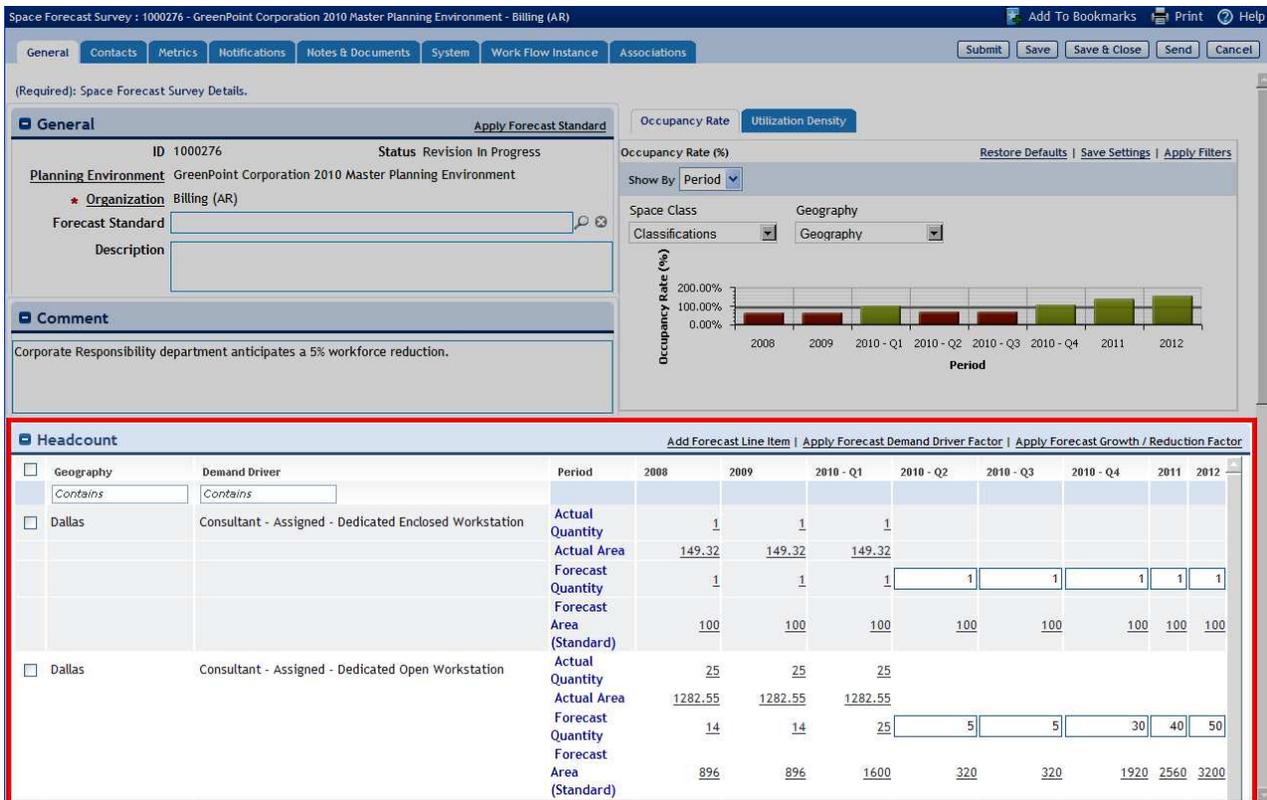
Figure 37. Example of Group By section

The data in a Group By section is sourced from a query. The platform takes sorted records from a query and displays them in a tabular format, showing data grouped by the values of one specific field. The format shows more than one field for the record in the grouped by section and allows dynamic runtime properties (Visible, Read-Only, and Label) for these columns.

This section type allows a lot of data to be shown in a small amount of real estate in a cross tab-like format. It condenses data, taking all records with like "column" values and displaying them in a grid.

For example, if your query returned the 15 rows in the following figure.

Figure 38. Example of query source for Group By section

The result could be displayed in a Group By section as shown in the following figure. The first five rows in the query result, circled in red, correspond to the first row in the Group By section, also circled in red.

| | | | 02 - 2009 | | | 01 - 2009 | | | 03 - 2009 | | 04 - 2009 | | 05 - 2009 | |
	* Location	* Space Class	ACTUAL	Actual Area	FORECAST	Forecast Area	ACTUAL	Actual Area	FORECAST	Forecast Area	FORECAST	Forecast Area	FORECAST	Forecast Area	FORECAST	Forecast Area
	Contains	Contains														
	17355 Weimar Cross Roads	Auditorium	33	10	0	6666	2	223	5	55	66	333	2201	44	500.1	52
	739 Ridgecrest Drive	Antenna Farm	2	20	2	20					301	30	401	444		
	739 Ridgecrest Drive	Auditorium	2	20	2	20	1	10	1	10	300	32	400	40	5	50

Figure 39. Group By section based on example query

Fields in a group by section can be edited. A border indicates to the user which fields can be edited. The platform can save the changes.

The organization of a Group By section includes three major parts, as illustrated in the following figure.

Figure 40. Organization of Group By section

Group By Field
> Field whose values the section will be grouped by. There should be a discrete number of values for this field as its values determine the number of columns in the section. This is Period in the above figure and is indicated by A.

Group By Columns
> Fields to show in the Group By section. These are the fields that show for each record in the row/column. This is B in the above figure, and the fields are: Actual, Actual Area, Forecast, and Forecast Area. Notice not all are visible for each Group By field value. In this example, the system uses runtime metadata to hide/show based on the Group By Field value.

Columns
> These fields determine the contents of the row. All records with like valued columns make up a row. This is C in the above figure, and the fields are Location and Space Class. Raw data is sorted first on Location, then on Space Class. All records with the same Location/Space Class combination make up a row.

Each group of cells under a Group By Field value represents a record.

Group by section properties

To create a Group By section, select the tab that will contain the section and click the Add Section action. In the Properties panel for the new section, set the value of the Type property to Group By Section.

When you finish defining the properties of the Group By section, click the Apply action. To improve performance, the platform renders the Group By section but does not run the queries.

"Common section properties" includes information about the following properties in a Group By section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a Group By section that are not described in "Common section properties" follow:

Query Click the search icon to identify the query that defines the Group By section. See "Query section properties".

Group By
This radio button is always selected and grayed out for Group By sections.

Orientation
Select either Horizontal or Vertical from the drop-down list to indicate whether the grid will be horizontal or vertical in appearance. It determines whether the Group By Columns and their respective values are laid out horizontally or vertically.

The following figure shows an example of horizontal orientation.

Figure 41. Horizontal Group By section

The following figure shows an example of a vertical orientation.

Figure 42. Vertical Group By section

Although a Group By section has no automatically generated actions, you may want to define section actions. See "Actions".

On section save, when not in create mode, the platform checks to make sure there is one and only one Group By field and at most one Group By order field. If either is not true, the platform displays a message and saves the section. On section publish, the platform makes the same validation; however, if either is not true, the

platform does not publish the section. On section publish, if no query is specified, the publish fails. Use the Validate action to help find issues with the Group By section prior to section publish. The Validate action is described in "Form Builder tabs and actions".

Group by section field properties

The fields for a Group By section come from the query as Display Columns for the primary business object. Use the Components panel to add fields to the control. The following figure shows this relationship. All display columns in the Group By query can be added to the Group By section. For example, triActualAreaNU has already been added to the Group By section, and triForcastAreaActualNU has not yet been added but is available in the Components section.
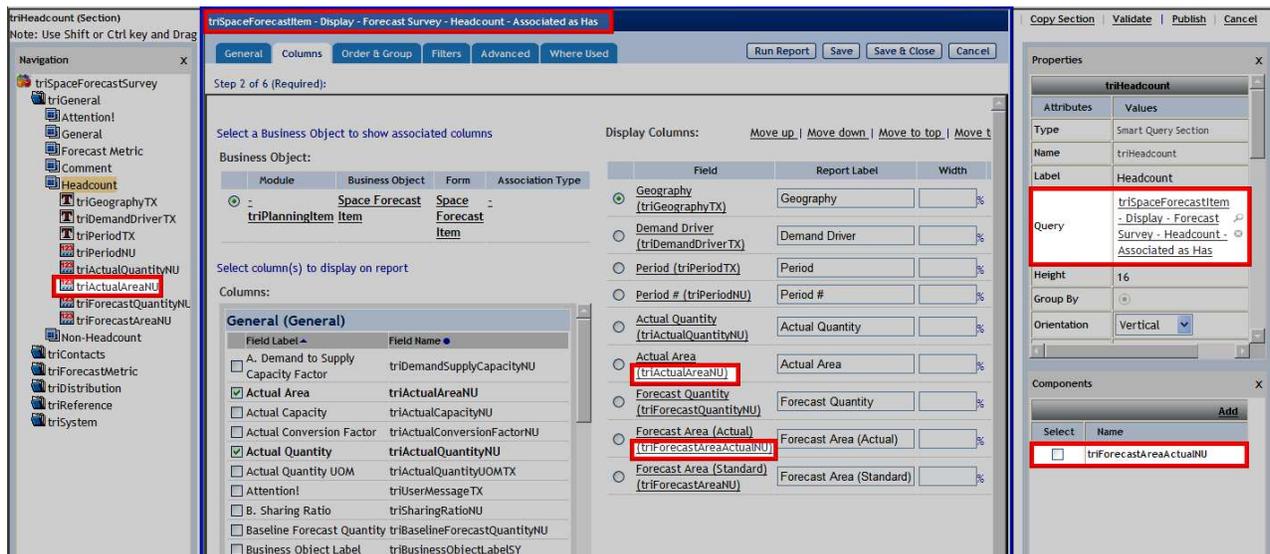
Figure 43. Example showing fields and display columns

Select what the fields represent with the Column Use property. Order the fields left to right with the Start Row. The fields of type Column must be first and in the sequence indicated by the query Order By settings. Place the other fields in the sequence you want them to appear in the Group By section.

Most are as described in "Form field properties". The properties unique to a Group By section's field are as follows:

Label Style Class

If you do not specify a value for this property, the appearance of Column labels and Group By value labels will be determined by default settings in the Style Manager.

If you do specify a value for this property, it will be the name of a field style sheet defined by the Style Sheet Editor. The style sheet determines the appearance of the Column and Group By value labels. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Sub Label Style Class

If you do not specify a value for this property, the appearance of this field's Group By Column labels will be determined by default settings in the Style Manager.

If you do specify a value for this property, it will be the name of a field style sheet defined by the Style Sheet Editor. The style sheet determines the appearance of the Group By Column label (these are the circled labels in the figures for "Group by section properties"). The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

Column Use

The value of this property determines how the column is used when the Group By section is rendered. The values are Group By field, Group By order field, Group By Column, and Column.

Group By field is the area in the Group By section described in "Group by sections". During publish, the platform validates that a Group By field is defined.

Group By order field is the field supplying the values used to order the Group By section.

Group By Column is the area in the Group By section described in "Group by sections".

Column is the area in the Group By section described in "Group by sections".

Only one field can be designated as either the Group By field or the Group By order field. The platform validates this when you click Apply.

If you do not set a Group By order field, the Group By field values are sorted alphabetically.

Start Row

The sequence in which columns are displayed comes from the Start Row property. This applies to Group By Columns and Columns. It is extremely important that the Column's Start Row is the same order as the Order By in the defined query as this determines the layout order of the section.

Start Column

Use the Start Column property to indicate that the Group By Columns should use two rows instead of one. For Group By Columns, valid values are 1 and 2. The Hide Label property hides the label on the second row if set.

The following figure shows an example of a horizontal orientation with Start Column = 2 for Actual Area and Forecast Area, and with Hide Label checked for Actual Area and Forecast Area:

Figure 44. Horizontal orientation, Start Column = 2

If the Hide Label property had not been checked for Actual Area and Forecast Area, the second row of labels would appear immediately below the first row of labels, and would look like the following figure:

Figure 45. Horizontal orientation with second row of labels

The following figure shows an example of a vertical orientation with Start Column = 2 for Actual Area and Forecast Area, and with Hide Label checked for Actual Area and Forecast Area:

Location	Space Class		01-2009		02-2009		03-2009		04-2009		2010	
Building A	Office Space	Forecast	10	3000	10	3000	10	3000	10	3000	10	3000
		Actual	15	4500	15	4500	15	4500	15	4500	15	4500
Building A	Conference Space	Forecast	10	3000	10	3000	10	3000	10	3000	10	3000
		Actual	15	4500	15	4500	15	4500	15	4500	15	4500
Building A	Storage Space	Forecast	10	3000	10	3000	10	3000	10	3000	10	3000
		Actual	15	4500	15	4500	15	4500	15	4500	15	4500
Building B	Office Space	Forecast	10	3000	10	3000	10	3000	10	3000	10	3000
		Actual	15	4500	15	4500	15	4500	15	4500	15	4500
Building B	Conference Space	Forecast	10	3000	10	3000	10	3000	10	3000	10	3000
		Actual	15	4500	15	4500	15	4500	15	4500	15	4500

Figure 46. Vertical orientation, Start Column = 2

If there is an issue when the platform is building a multi-row / multi-column Group By section, the platform attempts to use a single-row layout.

Workflow metadata

If you want to use metadata to influence the Visible, Read Only, or Label property for the Group By Columns, the field specified as the Group By field must be a reference field (i.e., a locator, classification, or smart object). The metadata defined on its reference instances will be used to determine the Visible, Read Only, or Label properties for the Group By Columns. The data definition for the referenced object must include the Group By Column fields. Use a Modify Metadata task in a workflow to change the Group By Column metadata on the Group By Field value records.

Labels changed via instance data are not translatable.

Data

Data in the Group By section that is editable will have borders turned on automatically. When the user clicks the Save or Save & Close action, the platform updates the records with any changes made by the user.

If there is more than one record for a given cell in the Group By section, or there is no record for the cell, the platform writes a log entry. If there is more than one record, the platform places one of them in the cell. If there is no record, the platform places an empty value in the cell.

The user will be able to click a read-only value in the Group By section and see the linked record.

If the query reaches a threshold where the query results are large enough that the platform cannot build a Group By section, the platform displays a message to the user. This threshold is defined in the TRIRIGAWEB.properties file.

Column filters

Group By sections support column filters on the Columns type fields. The operator and filter columns come from the query. After a user enters a filter string, the platform runs the query and displays the Group By section filtered by the filter string.

Query

The query and the section are intertwined. The query determines the fields available in the section by which fields are defined as the Display Columns. The query Order By determines how the data is ordered and these must be designated as the Column fields in the Group By section. Do not use the Group By feature of the query. The Filter Columns in the query determine which Columns in the Group By section have filters.

Calendar sections

A Calendar section displays the calendar for the record.

To create a calendar section, select the tab to contain the section and click the Add Section action. Then set the value of the Type property to Calendar Section.

Calendar sections are not executed in the Form Builder. Instead, the Layout displays a label identifying the section.

"Common section properties" includes information about the following properties in a calendar section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The properties in a calendar section that are not described in "Common section properties" follow:

Calendar Set
> The value for this property is the calendar set that is the source of events to be displayed in the calendar section. For more information about calendar sets, go to the *IBM TRIRIGA 10 Application Administration User Guide*.

Context Override

By default, the context for a calendar section is the record the form is displaying. You can use the Context Override property to specify a different context in terms similar to $$ query filters. For example, $$PARENT::General::triSpaceTX or $$USERID$$. For more information about these filters, see the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Select Workflow

This workflow handles user interactions with the calendar section. All user actions trigger this workflow.

The workflow is passed two business objects as workflow parameters: a business object representing the action taken by the user and a business object that conveys the details for the calendar event on which the user performed the action.

These business objects are system business objects. The workflow detects what action was taken by the user as defined in the first business object and performs the appropriate tasks with the information on the calendar event defined in the second business object. The workflow returns failure and an error message that can be displayed to the user.

The first business object contains the action taken by the user. The name of this system business object is triUserAction and the fields in this business object are as follows:

triActionTX

The action performed. For example, RescheduleEvent or ChangeTimeWindowSelector.

If the action is to delete or cancel, the workflow must provide the processing to support the cancellation. For example, if a reservation is cancelled, the workflow must not only remove the reservation from the resource, but also any associated reservations for meal service and equipment, and ensure any cancellation charges are taken care of.

triSuccessBL

Returned by the workflow to tell whether the user action was a success or a failure.

triUserMessageTX

Returned by the workflow when the action fails. It contains a message to display to the user.

triRefreshStrategyTX

Returned by the workflow to identify what to refresh as a result of the action. The value returned can be one of the following:

- none: Refreshes the data in the right side of the calendar section.
- self: Refreshes the calendar section.
- parent: Refreshes the tab that contains the calendar section under the control of the value of the Session only property in the Modify Metadata task in the workflow. For more information about the Session only property, see "Modify metadata task".

triReturnRecordIdTX

Returned by the workflow to identify what record to open.

The second business object contains the details about the calendar event affected by the action. The name of this system business object is triCalendarEvent and the fields in this business object are as follows:

triEventIdTX
>The SpecId of the event acted upon. This is blank if the action does not affect any event or for an occurrence of a recurrence that is not an exception to the recurrence.

triRecurrenceIdTX
>The SpecId of the event acted upon when the action affects an instance of a recurring event. This is blank if the action does not affect any event or for a single occurrence event.

triResourceIdTX
>The SpecId of the resource affected by the event. This is filled only when the user selects or deselects a resource in the list of resources.

triStartDT
>The start date of the event acted upon. If this is empty, triEndDT is also empty.

triEndDT
>The end date of the event acted upon. If this has a value, triStartDT also has a value.

For example, the section workflow is triggered when the user changes the time for a resource. The value of triActionTX in triUserAction is ChangeTimeWindow. triCalendarEvent contains the new start date and time and end date and time. The time window selector updates to the new dates and times when the workflow runs a field onchange workflow.

If this property has no value, then no workflow runs when the user interacts with the section.

To specify the value of this property, select from the list of synchronous workflows. The synchronous workflows in the list are workflows launched from records created from the business object associated with the form that the calendar section is part of.

Note that the Select association created when the user interacts with the calendar section records is deleted after the workflow completes.

Available View Modes
>A view mode describes the way days and events are displayed to the user. Select the check boxes for the options users are to have when they view the data in this calendar section.

Default View Mode
>Select how this calendar section first appears to the user. After initial display, the user can change the way days and events are displayed from this initial value to another that is specified in the Available View Modes property.

Alternate Form List
>This property identifies the alternate form list to be used to display a record when that record is opened from this calendar section. The alternate form used to display the record depends on the module, business object, and form defined for the record.

Read Only

If this check box is selected, a user cannot modify an event, nor move or drag an event, nor delete an event. A user can open an event record but not modify or delete any fields.

Availability sections

An Availability section displays the availability of resources over time.

The left side of an availability section shows resources. The query in the availability section properties defines which resources and columns are displayed. All columns returned by the query display on the left side.

The right side of an availability section shows the availability of the resources over time. The availability of each resource is determined by its calendar set. For more information, see the *IBM TRIRIGA 10 Application Administration User Guide*.

The colors within an availability section are set in the Style Manager. The Style Manager is discussed in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*. For details on styling the availability section, see Availability section on the IBM TRIRIGA wiki.

Print preview is not supported for availability sections.

Several properties in TRIRIGAWEB.properties control the Availability section.

AVAILABILITY_SECTION_ROW_LIMIT

When rendering the Availability section, this property limits the maximum number of results to the number specified. If there are too many results, a warning about exceeding the row size is displayed.

The default value is set to 50 rows. If set to 0, -1, or other invalid value, it will use the default.

WARNING: A large number of returned rows may cause memory issues. It is recommended that you update the backing reserve queries to reduce results or design filters that keep the number of record results under this value. Any value above the max value of 500 will be set to 500.

ENABLE_CONCURRENT_AVAILABILITY

Determines whether to process Reserve Queries concurrently. Set to True if you want to process Reserve Queries concurrently. The default is True.

CONCURRENT_AVAILABILITY_POOL_SIZE

If ENABLE_CONCURRENT_AVAILABILITY is true, this property determines the maximum number of system wide threads that will be used in processing availability. The default is 50.

CONCURRENT_AVAILABILITY_REQUEST_BATCH_SIZE

If ENABLE_CONCURRENT_AVAILABILITY is true, this property determines the maximum number of threads that each availability request can use to process availability. The default is 10.

When the record in an availability section is a recurring event, the user sees a Ratio column to the left of the data displayed about the record in the left side of the availability section. This column shows how available the resource is for all occurrences of the recurring event within the view of the right side of the

availability section. For example, if the record is an event that is to happen once a day for five days and the resource is available for each of those days, the Ratio column contains 5/5. If the resource is available for three of those days, the Ratio column contains 3/5.

The maximum number of occurrences that will be displayed and created in the Availability section defaults to 20. This means that if you create, for example, a recurring event that occurs each week for a year, only 20 occurrences, not 52, are displayed and created in the Availability section. To change this limitation, you can change the following property in the TRIRIGAWEB.properties file.

CALENDAR_EVENT_MAX_OCCURRENCES

> Sets the maximum number of occurrences when creating a recurring booking. The greater the number of occurrences set for this variable, the greater the chance that performance will degrade. The default is 20.

Availability section properties

To create an availability section, select the tab that is to contain the section and click the Add Section action. Then set the value of the Type property to Availability Section.

Availability sections are not executed in the Form Builder. Instead, the Layout displays a label identifying the section.

"Common section properties" includes information about the following properties in an availability section: Type, Name, Label, Height, Title Bar Color, Visible, Expand Section, Read Only, Show Title Bar, Style Class, Start Row, Row Span, Start Column, and Col Span. The other properties in an availability section follow:

Query

> Select the query that defines the resources to display in the availability section. This query identifies which records and which of their fields display in the left side of the availability section. The events for these records are retrieved by using each record's calendar set and displayed in the right side of the section.

Select Workflow

> This workflow handles user interactions with the availability section. All user actions trigger this workflow. The workflow is passed two business objects as workflow parameters: a business object presenting the action taken by the user and a business object that conveys the details for the calendar event affected by the action.

> If the Create Association property is set, then the Select Workflow will not run when the user selects a resource. Instead, the system will create an association explicitly.

> These business objects are system business objects. The workflow detects what action was taken by the user as defined in the first business object and performs the appropriate tasks with the information on the calendar event defined in the second business object. The workflow returns failure and an error message that can be displayed to the user.

> The first business object contains the action taken by the user. The name of this system business object is triUserAction and the fields in this business object are as follows:

triActionTX

>   The action performed. For example, RescheduleEvent or ChangeTimeWindowSelector.

>   If the action is to delete or cancel, the workflow must provide the processing to support the cancellation. For example, if a reservation is cancelled, the workflow must not only remove the reservation from the resource, but also any associated reservations for meal service and equipment, and ensure any cancellation charges are taken care of.

triSuccessBL
>   Returned by the workflow to tell whether the user action was a success or a failure.

triUserMessageTX
>   Returned by the workflow when the action fails. It contains a message to display to the user.

triRefreshStrategyTX
>   Returned by the workflow to identify what to refresh as a result of the action. The value returned can be one of the following:
>   - none: Refreshes the data in the right side of the availability section.
>   - self: Refreshes the availability section.
>   - parent: Refreshes the tab that contains the availability section under the control of the value of the Session only property in the Modify Metadata task in the workflow. For more information about the Session only property, see "Modify metadata task".

triReturnRecordIdTX
>   Returned by the workflow to identify what record to open.

The second business object contains the details about the calendar event affected by the action. The name of this system business object is triCalendarEvent and the fields in this business object are as follows:

triEventIdTX
>   The SpecId of the event acted upon. This is blank if the action does not affect any event or for an occurrence of a recurrence that is not an exception to the recurrence.

triRecurrenceIdTX
>   The SpecId of the event acted upon when the action affects an instance of a recurring event. This is blank if the action does not affect any event or for a single occurrence event.

triResourceIdTX
>   The SpecId of the resource affected by the event. This is filled only when the user selects or deselects a resource in the list of resources.

triStartDT
>   The start date of the event acted upon. If this is empty, triEndDT is also empty.

triEndDT
>   The end date of the event acted upon. If this has a value, triStartDT also has a value.

For example, the section workflow is triggered when the user changes the time for a resource. The value of triActionTX in triUserAction is ChangeTimeWindow. triCalendarEvent contains the new start date and time and end date and time. The time window selector updates to the new dates and times when the workflow runs a field onchange workflow.

Maximum View Mode

A view mode describes the way days and events are displayed to the user. Select the widest view users are to have when they view the data in this availability section.

The purpose of this property is to control how much data the system has to generate and display to the user. The more data that is generated, the heavier the data load. Month is the default value.

Default View Mode

Select how this availability section first appears to the user. After initial display, the user can change the way events are displayed.

Display time window selector

Select this check box to enable the time window selector in the availability section. The time window selector is a vertical bar that displays a visual indication of the time range for a reservation. The time range is based on the date and time that the user selects in the Start Date and End Date fields in the Reservation Details section of the reservation.

The colors that are used for the Time window selector and reservations are configured in the Style Manager. The style is from the Style Class section property. For details on styling the availability section, see Availability section on the IBM TRIRIGA wiki.

Enable time window selector change

This property is visible only when the Display time window selector property is selected.

Note: The **Enable time window selector change** property has no function in this release. In previous releases, when the property is selected, the user can drag or resize the time window selector to change the time and length of the reservation.

Enable resource select

When this property is selected, the user can select resources in the resources table on the left side of the availability section.

Enable resource single select

This property is visible only when the Enable resource select property is selected. When the Enable resource single select property is selected, the user can select only a single resource at a time in the resources table on the left side of the availability section. When this property is not selected, the user can select multiple resources.

Create association

This property is visible only when the Enable resource select property is selected. When the Create association property is selected, an association is created or removed between the context record and the resources selected or deselected.

Select Association Type

This property is visible only when the Create association property is selected. You use this property to identify the association to create or remove between the context record and the resources selected or deselected.

Temporary association

This property is visible only when the Create association property is selected. You select this property when the association to be created or removed is temporary. A temporary association no longer exists after the workflow completes.

Alternate Form List

This property identifies the alternate form list to be used to display a record when that record is opened from this availability section. The alternate form used to display the record depends on the module, business object, and form defined for the record.

Actions

Actions can happen to elements in the platform. We are interested in actions that can happen to forms or to parts of a form.

You can specify a specific response or set of responses to form actions. Actions may be triggered by clicking something or by changing the value of a field. Actions can be associated with most parts of a form.

Actions can be associated with the form itself. Hyperlinks and buttons to trigger such actions appear at the top of the form. Actions associated with the form are managed through the state transition family of the form and underlying record. State transition families are discussed in "Life cycles". The way that they control the actions of a form is discussed in "State-based actions".

Actions can be associated with sections. There are some actions that the platform automatically creates and associates with some kinds of sections. Section actions are discussed in "Section actions".

Actions can be associated with fields. Field actions are not triggered through a hyperlink or a button, but through changes to the contents of a field. Field actions are discussed in "Field and button actions".

Buttons can and usually do have associated actions. Button actions are triggered by a user clicking the button. The process of configuring a button action is the same as the process of configuring a field action. Button actions are also discussed in "Field and button actions".

When a user performs an action, the platform detects whether any Modify Metadata workflow tasks were run for that action.

If a Modify Metadata task has been run, the tab is re-rendered. If no Modify Metadata task was run, field values are updated and graphics sections, calendar sections, query sections, and availability sections are refreshed by updating any information shown to the user.

When an action happens, the things you can specify to happen in a sequence of responses depends on what the action is associated with. Here are some of the possibilities:

- Some types of actions may allow you to specify that a record should be created from a business object that you specify. If you specify that the action should create a record, then the next thing that will happen is that a window will pop up for the user to edit the record's fields.
- If the type of action does not allow creating a record, it will allow specifying that a window should pop up that displays the content from a URL that you specify.
- You may specify that a synchronous workflow should be run. If you specify that the action should run a synchronous workflow, there may be the option of specifying an association that should be created to allow the workflow to navigate to other records.

Section actions

You can explicitly specify actions for a section in a form. You can also request that the platform automatically generate standard actions. Except for properties used to request the automatic generation of standard actions, you cannot specify actions for a section until after the first time you save the section.

If a section is a type of section for which you can explicitly specify actions, after the first time you save the section, a list of actions will appear in the Actions section below the properties. This list of actions is initially empty because it has no automatically generated actions. You can add an action by clicking the *Add* action on the section bar. You can delete an action, if it is not automatically generated, by selecting the check box and clicking the *Delete* action.

You cannot delete automatically generated actions. To make an automatically generated action not appear at the top of a section, edit the section properties and clear the *Visible* property check box.

To edit the properties of an action, click its name.

The details of what actions and responses are available vary with the type of section. There are separate descriptions for each type of section:

Form Section
　　　　See "Form section actions".

Smart Section
　　　　See "Smart section actions".

Query Section
　　　　See "Query section actions".

Report Section
　　　　No action can be associated with a Report section.

Excel Section
　　　　You cannot explicitly specify actions for an Excel section.

Gantt Section
　　　　Only Form section Execute Workflow actions (see "Form section execute workflow actions") can be associated with a Gantt section.

Multi Tab Section
No actions can be directly associated with a Multi Tab section.

Common section action properties

Many of the properties of a section action are the same regardless of the type of section on which the section action appears. The information is not repeated in other areas except to highlight a difference from the common property or to add additional information.

Label This is the text of a hyperlink or button that appears at the top the section.

Display Sequence

The value of this property is a number.

The order of the actions displayed at the top of the section depends on the value of the *Display Sequence* property. The actions are displayed from lowest Display Sequence value to highest Display Sequence value.

Popup If the check box in this property is selected, a window pops up when a user clicks this action. The nature of the window that pops up depends on the value of the *Action Type* property.

Visible

If this check box *is* selected, the action is visible. If the check box is *not* selected, the action is hidden. The default for this property is for the check box to be selected.

Active

If the check boxes for both the *Visible* property and this property are selected, then this action is displayed in its normal way. Clicking it triggers the action.

If the check box for the *Visible* property is selected, but the check box for this property is not selected, then it is grayed out. Clicking it does not trigger this action.

Access Key

The value of this property identifies how a user can cause this action to run from the keyboard. The value of this property should be a single character. The user then presses a key or a key chord and the Access Key character while working in the form to run this action. Different browsers use different keys or key chords, as follows:

- For Microsoft Internet Explorer: Alt+AccessKey+Enter
- For Mozilla Firefox: Alt+Shift+AccessKey
- For Google Chrome: Alt+AccessKey
- For Apple Safari: Alt+Control+AccessKey

Action Type

The value of this property determines the type of response that will be given to this action. The values you can select for this property are determined by whether the check box for the *Popup* property is selected.

If the check box for the *Popup* property is *not* selected, the only value available for this property is *Execute Workflow*. The meaning of this value is discussed in "Form section execute workflow actions".

If the check box for the *Popup* property *is* selected, then the value for this property may be one of the following:

Custom

> The meaning of this value is discussed in "Form section custom actions".

Form The meaning of this value is discussed in "Form section form actions".

Query The meaning of this value is discussed in "Form section query actions".

URL The meaning of this value is discussed in "Form section URL actions".

Association

> The value of this property is the name of the association to be created by this action from the record being edited by this form to the records created by this action. The workflow that is run by this action can use this association to navigate to the newly created records.
>
> The names in the drop-down list are the names in the List Manager's *Association Types* list.

Permanent Association

> If the check box for this property is *not* selected, then the association specified by the *Association* property is considered a temporary association. If the association is a temporary association, then the association is automatically deleted after the workflows have completed.
>
> If the check box for this property *is* selected, then the association specified by the *Association* property is considered a permanent association. Permanent associations are not automatically deleted.

View Type

> You use this property to specify the appearance of the section action.

Link Link is the default value. The section action appears as a hyperlink on the section bar.

Dark Button

> The section action appears as a button with a dark background and light letters on the section bar. The button colors are set on the Style Manager.

Light Button

> The section action appears as a button with a light background and dark letters on the section bar. The button colors are set in the Style Manager.

Form section actions

When you click the *Add* action, you create a new action for a Form section.

"Common section action properties" includes information about the following properties of a Form section action: Label, Display Sequence, Popup, Visible, Active, Access Key, Action Type, Association, Permanent Association, and View Type.

Form section execute workflow actions

If the value of the *Action Type* property is *Execute Workflow*, the response to the action is to run a synchronous workflow. This option is available only if the check box for the *Popup* property is *not* selected.

The additional action property when the value of the *Action Type* property is *Execute Workflow* follows:

Workflow

> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Form section custom actions

If the value of the *Action Type* property is *Custom*, the response to the action is to pop up a window and run synchronous workflows. This option is available only if the check box for the Popup property is selected.

The additional action properties when the value of *Action Type* is *Custom*:

URL   The content of the window that pops up comes from the URL specified as the value of this property.

Workflow

> The workflow can navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

> If this property does not have a value, then no workflow runs after the newly created record has been edited.

> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Note the following behavior for section custom actions:

- When you create a section custom action that performs associations, only the association workflow event for that section custom action will fire. No other association workflow event will fire unless they are specified in the *Pre Form Workflow* or *Workflow* properties for the section custom action. This behavior is true whether or not the check box for the *Permanent Association* property is selected for the section custom action.
- The reason for this behavior is because the association is made during the creation of the child record (i.e., the form that opens when you click the section custom action). When the child record is created, it is not yet clear whether the user will discard or continue the process of filling in the popup form information. Therefore, the association must remain temporary until the user saves and closes, or activates the child record.

- Also note that the check box for the *Permanent Association* property determines whether or not the specified association (not the section custom action association) will remain after the child record is closed. This property does not affect the execution of workflow events that are specified in the *Pre Form Workflow* or *Workflow* properties.
- If you want to specify another association workflow event to fire, that is different from the section custom action workflow, you can specify that workflow for the section custom action. The *Pre Form Workflow* or *Workflow* properties specify which workflow events to fire. The *Pre Form Workflow* fires its workflow when the child record is created, while the *Workflow* fires its workflow after closing the child record. If specified, these workflows will fire regardless of whether the user discards, save and closes, or activates the child record.

Form section form actions

If the value of the *Action Type* property is *Form*, the response to the action is to create a record and run synchronous workflows. This option is available only if the check box for the *Popup* property is selected.

The additional action properties when the value of the *Action Type* property is *Form* follow:

Popup Module
> The value of this property is the name of the module that contains the form to be used for editing records created by this action. The drop-down list for this property contains the names of all modules.

Popup Form

> The value of this property is the name of the form to be used for editing records created by this action. This also determines the type of record created by this action. The business object used by this action to create records is the business object associated with the form named by this property.

> The drop-down list in this property contains the names of forms in the module named by the value of the *Popup Module* property.

Pre Form Workflow

> The value of this property is the name of a synchronous workflow that runs before the user can edit the record created by this action. The Pre Form Workflow runs after the record is physically created.

> The workflow can navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

> If this property does not have a value, then no workflow runs before the window pops up.

> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with the form selected in the Popup Form property. The workflow you select from this list is the workflow that runs when this action is triggered. The selected workflow runs instead of the workflow defined for the business object as the Pre-Create Workflow property in Data Modeler. Pre-Create workflows are discussed in "Viewing business object properties".

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Workflow

> The value of this property is the name of a synchronous workflow that runs after the person using the form has finished editing the record this action created. More precisely, this workflow runs after the person editing the newly created record initiates an action that changes the state of the newly created record to a state other than null and then closes the window that the record was in.

> The workflow can navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

> If this property does not have a value, then no workflow runs after the newly created record has been edited.

> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Save On Form Popup

> If the check box for this property is *not* selected, the underlying parent record is not saved after the popup renders.

> If the check box for this property *is* selected, the system saves the underlying parent record data as soon as the popup is rendered.

Form section query actions

If the value of the *Action Type* property is *Query*, the response to the action is to run a query, pop up a window containing the results of the query, wait for a user to select some of the query results, and then run a synchronous workflow. This option is available only if the check box for the *Popup* property *is* selected.

The additional action properties when the value of the *Action Type* property is *Query* follow:

Popup Module
> The value of this property is the name of the module that contains the query to be run. The drop-down list for this property contains the names of all modules.

Popup Query

> The value of this property is the name of the query to be run.

> The drop-down list in this property contains the names of queries and reports in the module specified by the value of the *Popup Module* property.

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected query or report.

Single Record Select

> If this check box in this property *is* selected, the pop-up query only allows a user to pick *one* of the query results.

> If this check box in this property is *not* selected, the pop-up query allows a user to pick any number of the query results.

> The value of this property is the name of a synchronous workflow that runs after the person using the form has finished selecting query results. The workflow can navigate from the record being edited by this form to the records that correspond to the selected query results by using the association specified by the *Association* property.
>
> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.
>
> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Form section URL actions

If the value of the *Action Type* property is *URL*, the response to the action is to pop up a window. This option is available only if the check box for the *Popup* property is selected.

The additional action when the value of the *Action Type* property is *URL* follow:

URL   The content of the window that pops up comes from the URL specified as the value of this property.

Smart section actions

When you click the Add action, you create a new action for a Smart section.

The actions you can explicitly specify for a Smart section are similar to those you specify for a Form section. See "Form section actions" for a description of those actions.

Smart section find action

The IBM TRIRIGA Application Platform automatically generates a *Find* action for Smart sections that reference a stand alone or link business object. This *Find* action makes it possible for a user to use a query to find a record and add the found record to the Smart section.

When the user clicks the *Find* action, they see the results of a query that finds records of the type referenced by the underlying smart section. The user selects records in the panel that he or she wants to be referenced by the smart section. The user then clicks the *Accept* action to close the panel. This causes the smart section to reference the selected records.

There are some slight differences in what the *Find* action does for multiple-record smart sections versus single-record smart sections. For multiple-record smart sections, it is possible for the person who clicks the *Find* action to select multiple records in the panel that pops up. Also, the multiple-record smart section references the selected records in addition to any records it previously referenced.

For single-record smart sections, the user can select only one record. When the single-record smart section is made to reference the selected record, it no longer references the record it previously referenced.

The *Find* action is not generated if the underlying smart section references an embedded business object. Embedded records exist only in smart sections, so they cannot be found using a query.

When you set up a Smart section, the *Find* action is generated when you first save the section.

Edit the properties of a *Find* action, at least to specify the query to use for finding records. The default query finds every record that the smart section's underlying association would allow to be in the Smart section. It displays only the name of the record. You generally want to be more selective about which records are displayed. Also, you generally want more than just the names of the records to be displayed. Thus, you generally want to create a query that is more appropriate for the *Find* action and edit the *Find* action's properties so that the *Find* action uses the query.

To edit the properties of the *Find* action, click the *Find* hyperlink that is its name.

The property you that is most usually set is the *URL* property. The value of this property identifies the query used to find records.

The details of how to select a query with the *Select Items* panel are discussed in "Query section properties".

The *Find* action has the following properties in addition to the properties described in "Common section action properties":

Popup The check box in this property is always selected and grayed out. You cannot change it.

Action Type
The value of this property is always *System Action*.

Smart section delete or clear action

The IBM TRIRIGA Application Platform automatically generates an action for smart sections to cause them to stop referencing records. For single-record smart sections, the name of the action is *Clear*. This action causes the underlying smart section not to reference a record.

For multiple-record Smart sections, the name of the action is *Delete*. In a multiple-record Smart section there is a check box next to the display of each record. This action causes the underlying smart section to stop referencing records whose check box in the Smart section is checked.

When you are setting up a Smart section, the *Clear* or *Delete* action is generated when you first save the section.

If you click the name of the action, you will see the action's properties.

The Delete or Clear actions have the following properties in addition to the properties described in "Common section action properties":

Popup The check box in this property is always grayed out and not selected. You cannot change it.

Action Type
The value of this property is always *System Action*.

Smart section add action

The IBM TRIRIGA Application Platform can automatically generate an action to create new records and add them to a smart section's underlying smart section. The default label for this action is *Add*. Whether the platform actually generates the action depends on two things:

- If the smart section references embedded records, the action is always generated. However, if the smart section references embedded records, the smart section has a property labeled *Quick Add*. If the check box in the *Quick Add* property is selected, the response to the action is different. The response that is given if the *Quick Add* property is selected is described in "Smart section quick add action".

- If the smart section references stand alone or link records, the section has a property labeled *Show Add*. If the check box in the *Show Add* property is selected, the action is generated.

When you are setting up a Smart section, the *Add* action is generated when you first save the section. If you click the name of the action, you will see the action's properties.

The Add action has the following properties in addition to the properties described in "Common section action properties":

Popup The check box in this property is always grayed out and selected. You cannot change it.

Action Type
The value of this property is always *System Action*.

Smart section quick add action

If the smart section underlying a smart section references embedded records, then the smart section has a property labeled *Quick Add*. If the check box in the *Quick Add* property is checked then the response to an *Add* action is different. It creates a new record without popping up a window to edit it. There are a few special considerations to think about since the record is created without human interaction:

- Like all other records, records created by quick add must have a unique name. Since the record is embedded, the name only needs to be unique within the records in the smart section that reference it. If the name is not at least unique within its smart section, then the record is not created. Usually the uniqueness of a record's name comes from a control number field or a workflow.

- Fields that are not part of the name but are supposed to be required should have a default value.

- Consider setting default values for fields that are not required.

Query section actions

After you create a query section, two actions are automatically generated for the query section.

Find The *Find* action pops up a window to find records that are not in the Query section because they do not have the required association with the record that contains the query section. The *Find* action is discussed in "Query section find actions".

DeAssociate

The *DeAssociate* action removes the association between selected records in the query section and the record that contains the query section. The *DeAssociate* action is discussed in "Query section deassociate actions".

When you click the *Add* action, you create a new action for a query section.

"Common section action properties" includes information about the following properties: Label, Display Sequence, Popup, Visible, Active, Access Key, Action Type, Association, Permanent Association, and View Type.

Query section execute workflow actions

If the value of the *Action Type* property is *Execute Workflow*, the response to the action is to run a synchronous workflow. This option is available only if the check box for the *Popup* property is *not* selected.

The additional action property when the value of the *Action Type* property is *Execute Workflow* follows:

Workflow

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Query section custom actions

Query section custom actions are defined in the same way as Form section custom actions. See "Form section custom actions".

Query section form actions

If the value of the *Action Type* property is *Form*, the response to the action is to create a record and run synchronous workflows. This option is available only if the check box for the *Popup* property is selected.

The additional action properties when the value of the *Action Type* property is *Form* follow:

Popup Module

The value of this property is the name of the module that contains the form to be used for editing records created by this action. The drop-down list for this property contains the names of all modules.

Popup Form

The value of this property is the name of the form to be used for editing records created by this action. This also determines the type of record created by this action. The business object used by this action to create records is the business object associated with the form named by this property.

The drop-down list in this property contains the names of forms in the module named by the value of the *Popup Module* property.

Pre Form Workflow

The value of this property is the name of a synchronous workflow that runs before the user can edit the record created by this action. The Pre Form Workflow runs after the record is physically created.

The workflow can navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

If this property does not have a value, then no workflow runs before the window pops up.

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with the form selected in the Popup Form property. The workflow you select from this list is the workflow that runs when this action is triggered. The selected workflow runs instead of the workflow defined for the business object as the Pre-Create Workflow property in Data Modeler. Pre-Create workflows are discussed in "Viewing business object properties".

The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Workflow

The value of this property is the name of a synchronous workflow that runs after the person using the form has finished editing the record this action created. More precisely, this workflow runs after the person editing the newly created record initiates an action that changes the state of the newly created record to a state other than null and then closes the window that the record was in.

The workflow can navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

If this property does not have a value, then no workflow runs after the newly created record has been edited.

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Save On Form Popup

If the check box for this property is *not* selected, the underlying parent record is not saved after the popup renders.

If the check box for this property *is* selected, the system saves the underlying parent record data as soon as the popup is rendered.

Query section query actions

If the value of the *Action Type* property is *Query*, the response to the action is to run a query, pop up a window containing the results of the query, wait for a user to select some of the query results, and then run a synchronous workflow. This option is available only if the check box for the *Popup* property *is* selected.

The additional action properties when the value of the *Action Type* property is *Query* follow:

Popup Module

The value of this property is the name of the module that contains the query to be run. The drop-down list for this property contains the names of all modules.

Popup Query

The value of this property is the name of the query to be run.

The drop-down list in this property contains the names of queries and reports in the module specified by the value of the *Popup Module* property.

The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected query or report.

Single Record Select

If this check box in this property *is* selected, the pop-up query only allows a user to pick *one* of the query results.

If this check box in this property is *not* selected, the pop-up query allows a user to pick any number of the query results.

Workflow

The value of this property is the name of a synchronous workflow that runs after the person using the form has finished selecting query results. The workflow can navigate from the record being edited by this form to the records that correspond to the selected query results by using the association specified by the *Association* property.

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that runs when this action is triggered.

The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Query section URL actions

If the value of the *Action Type* property is *URL*, the response to the action is to pop up a window. This option is available only if the check box for the *Popup* property is selected.

The additional action when the value of the *Action Type* property is *URL* follow:

URL   The content of the window that pops up comes from the URL specified as the value of this property.

Query section find actions

A query section's *Find* action pops up a panel to find records that are not in the query section because they do not have the required association with the record that contains the query section. For the contents of the *Find* pop-up to be consistent with the contents of the query section you need to provide a query specifically for this purpose.

When a user reviews the query results popped up by a Find action, the user can select the check boxes next to records to be added to the query section. The Find action does not add the records to the query section but does something else that should accomplish the same thing if all is properly configured.

When a user clicks the *Accept* action at the top of the popped up panel, an association is created from the record associated with the form to the selected records. The name of the association is the name specified as the value of the query section's *Association Type* property.

The Find action returns records only from the business object that the query section is defined against, unless multiple business objects are used in the query.

To edit the properties of the *Find* action, click the Find hyperlink that is its name. The most important property is the *URL* property. The value of this property identifies the query used to find records. The details of how to select a query with the *Select Items* panel are discussed in "Query section properties".

The *Find* action has the following properties in addition to the properties that are described in "Common section action properties":

Action Type
> The value of this property is always *System Action*.

Query section deassociate actions

This action removes records from a query section if the check box that corresponds to the record is selected by removing an association from the record associated with the form to the records whose check boxes were selected. The name of the association removed is the name specified as the value of the query section's *Association Type* property, which is described in "Query section properties".

The *DeAssociate* action may have other consequences, depending on what else relies on the association.

The properties of a *DeAssociate* action are the same as for the *Find* action, except that it does not have a *URL* property.

Field and button actions

Field actions are triggered when the IBM TRIRIGA Application Platform detects that a user has finished changing the value in a field. Button actions are triggered when a user clicks a button. To create a new field action, select a field in the Navigation panel. Scroll to the bottom of the field's properties, and click Add in the Actions section. To create a new button action, add a new Form Action field, as described in "Buttons" and save the button. Now scroll to the bottom of the properties, and click Add in the Actions section. The way to set up field and button actions is very similar.

For both a field action and a button action, the value of the *Label* property is read-only. For a field action the value of the *Label* property is always *onChange*. For a button action the value of the *Label* property is always *onClick*. That is the only difference in the setup for the two kinds of actions.

Here are the things that you can arrange for in a response to a field or button action:
- You can arrange for a window to pop up with the results of a query. This will allow the user to select records that correspond to query results.
- You can arrange for a workflow to run. If you also had the query pop up in a window, the workflow can be aware of which records were selected in the query.

- You can arrange for a record to be created and a form to pop up so the person using the form can edit the new record.

Here are descriptions of the other properties for these actions:

Popup Module

> The value of this property is the name of the module that contains the query to be run. The drop-down list for this property contains the names of all modules in the IBM TRIRIGA Application Platform environment.

> If no value is specified for this property, no query will be run.

Popup Query

> The value of this property is the name of the query to be run. If no value is specified for this property, no query will be run.

> The drop-down list in this property contains the names of queries and reports in the module specified by the value of the *Popup Module* property.

> The name of this property is a hyperlink. Click the name of this property to view or edit the definition of the selected query or report.

Single Record Select

> If the check box in this property *is* checked, the person using the form will only be able to pick one of the query results.

> If the check box in this property is *not* checked, the person using the form will be able to pick any number of query results.

Workflow

> The value of this property is the name of a synchronous workflow that will run after the person using the form is finished selecting query results. If no query is run, the workflow is run immediately.

> The workflow is able to navigate from the record being edited by this form to the records that correspond to selected query results by using the association specified by the *Association* property.

> If this property does not have a value, no workflow will run after the newly created record has been edited.

> The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that will run when this action is triggered.

> The name of this property is a hyperlink. Clicking the name of this property opens the definition of the selected workflow.

Association

> The value of this property is the name of an association that may be created by this action. If a query is specified and a user selects some of the query results, an association with the specified name is created from the record being edited by this form to the records that correspond to the selected query results. The workflow that runs after the query results are selected can use the association to retrieve the selected records.

> If it is specified that this action will create a record, an association with the specified name is created from the record being edited by this form to the newly created record. The workflows specified by the *Pre Form Workflow*

property or the *Post Form Workflow* property can use this association to navigate from the record being edited by this form to the newly created record.

The names in this property's drop-down list are the names in the List Manager's *Association Types* list.

Permanent Association

If the check box for this property is *not* checked, the associations created by this action are considered temporary. If the associations are temporary, they are automatically deleted after the workflows have finished.

If the check box for this property *is* checked, the associations created by this action are considered permanent and are not automatically deleted.

Save On Form Popup

If the check box for this property is not checked, the underlying parent record is not saved after the popup renders.

If the check box for this property is checked, the system saves the underlying parent record data as soon as the popup is rendered.

Popup Form Module
The value of this property is the name of the module that contains the form to use for editing records that will be created by this action. The drop-down list for this property contains the names of all the modules in the IBM TRIRIGA Application Platform environment.

Popup Form

The value of this property is the name of the form to use for editing records that will be created by this action. This also determines the type of record created by this action. The business object used by this action to create records will be the business object associated with the form named by this property.

The drop-down list in this property contains the names of forms in the module specified by the value of the *Popup Module* property.

Pre Form Workflow

The value of this property is the name of a synchronous workflow that will be run before the window pops up that allows the user to edit the record created by this action. The Pre Form workflow runs after the record is physically created.

The workflow is able to navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

If this property does not have a value, no workflow will be run before the window pops up.

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with the form selected in the Popup Form property. The workflow you select from this list is the workflow that will run when this action is triggered. The selected workflow will run instead of the workflow defined for the business object as the Pre-Create Workflow property in Data Modeler. Pre-Create workflows are discussed in "Viewing business object properties".

The name of this property is a hyperlink. Click the name of this property to view or edit the definition of the selected workflow.

Post Form Workflow

The value of this property is the name of a synchronous workflow that will run after the person using the form is finished editing the record this action created. More precisely, this workflow is run after the person editing the newly created record initiates an action that changes the state of the newly created record to a state other than null and the user closes the window he was using to edit the record.

The workflow is able to navigate from the record being edited by this form to the newly created record by using the association specified by the *Association* property.

If this property does not have a value, no workflow will run after the newly created record has been edited. If no value is specified for the *Popup Form* property, no record will be created and the workflow specified by this property will not run.

The drop-down list for this property contains the names of synchronous workflows associated with the business object associated with this form. The workflow you select from this list is the workflow that will run when this action is triggered.

The name of this property is a hyperlink. Click the name of this property to view or edit the definition of the selected workflow.

Includes/Forms tab

The Includes/Forms tab of a Form Wizard window is used for two separate purposes. The top section is labeled *Includes*. It is used to control how records that are part of a hierarchy are displayed in a manager and which form to use for them, based on the type of parent they have. There is an explanation of how the *Includes* section is used in "Includes".

The bottom section is labeled *Forms*. It is used to specify what form report templates will be available for use in the form's *Reports* tab. Form reports are described in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

If the form's *Show Reports* check box (described in "Form properties") *is* checked, it means that the form will have an automatically generated *Reports* tab. If the form's *Show Reports* check box is *not* checked, it means that the form will not have a *Reports* tab and the contents of the Form Wizard's *Forms* section will be ignored.

Style sheets

For some applications, you may want certain kinds of fields, sections, or tabs to be presented differently from others. You may want some fields to present data with a different font or another kind of field to have a label that is a different color. You may want some sections to have a different colored title bar. If you do want to do such things, you will want to do them not just in one form, but consistently throughout your application.

The IBM TRIRIGA Application Platform provides a way to do this in the form of style sheets. Fields have a property named *Label Style Class* (described in "Form field properties") that can determine what the field's label looks like. Fields have a

property named *Data Style Class* that determines what the field's data looks like. Sections and tabs have a property named *Style Class* that determines their appearance.

The value for these properties is a name of a style sheet that has its own properties associated with it. To view, edit or create style sheets, you use a tool named the Style Sheet Editor. The Style Sheet Editor is described in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

State-based actions

Actions that apply to an entire form are managed through the state transition family of the business object associated with the form. State transition families are described in "Life cycles".

You can use the *State Transition* tab of the Form Wizard to control how the state transition family of the business object associated with the form is used by the form. You can use the *State Transition* tab to:
- Make only part of the state transition family available through the form.
- Cause a state transition to make the underlying record read only.
- Cause a state transition to close the window that the form is in.
- Associate a keyboard shortcut with a state transition.
- Cause a state transition to create a new record.
- Cause a workflow to run prior to the opening of a pop-up form.

The sequence in which actions display on a form is determined by the sequence in which the actions appear in the State Transition tab. Top-to-bottom in the tab corresponds on the form to left-to-right for normal actions and top-to-bottom in the More menu. The value of the MAX_FORM_ACTION_NUMBER property in TRIRIGAWEB.properties impacts this behavior. This property sets the number of action buttons that can appear on the form menu, not including the More button or the X. For example, if the form has 7 primary actions and MAX_FORM_ACTION_NUMBER is 4, the first 4 actions will appear on the form as buttons and the last 3 will be in the More menu.

When you first look at a form's *State Transition* tab, it shows all the states and transitions in the state transition family of the business object associated with the form. It will look something like the following figure.
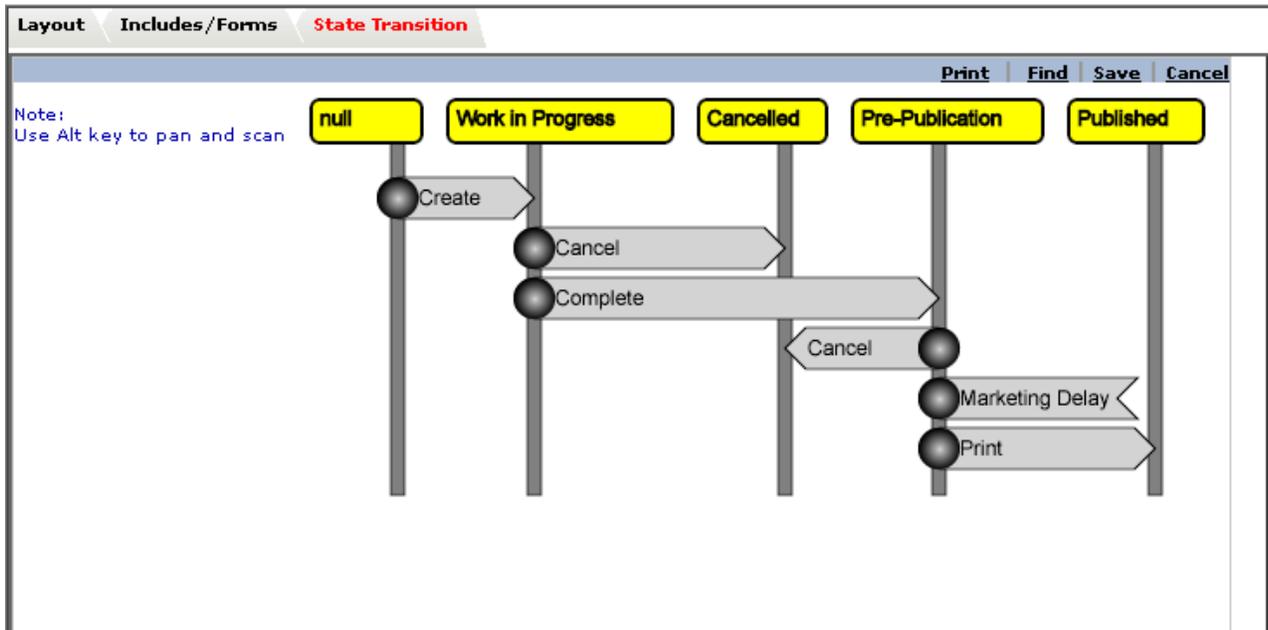
Figure 47. Form wizard State Transition tab

If you do not want this form to have access to the *Published* state or the state transition that leads to the *Published* state, you can delete the *Published* state from the form's view of the state transition family.

To delete a state from this form's view of the state transition family, begin by selecting the state you want to delete. When you click a state it becomes selected. You can tell that a state is selected because its color is cyan instead of yellow, like in the following figure.
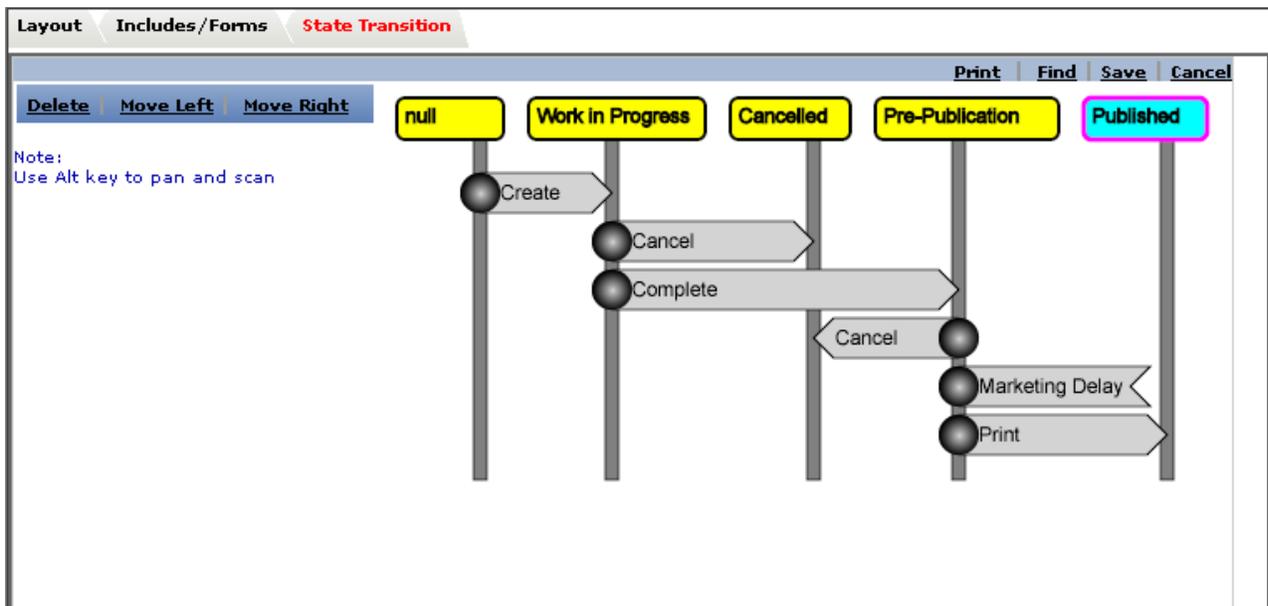
Figure 48. State Transition tab with selected state

Click the *Delete* action to delete the selected state and all state transitions connected to it from the form's view of the state transition family.

When a state is selected, actions to manipulate the state appear on the left side of the *State Transition* tab. You can use the *Move Left* or *Move Right* action to change the order in which the states appear across the *State Transition* tab.

If you decide that you want to delete a state transition but not a state, you can do that too. Begin by clicking the state transition you want to delete. Clicking a state transition selects it. A selected state transition looks like the following figure.
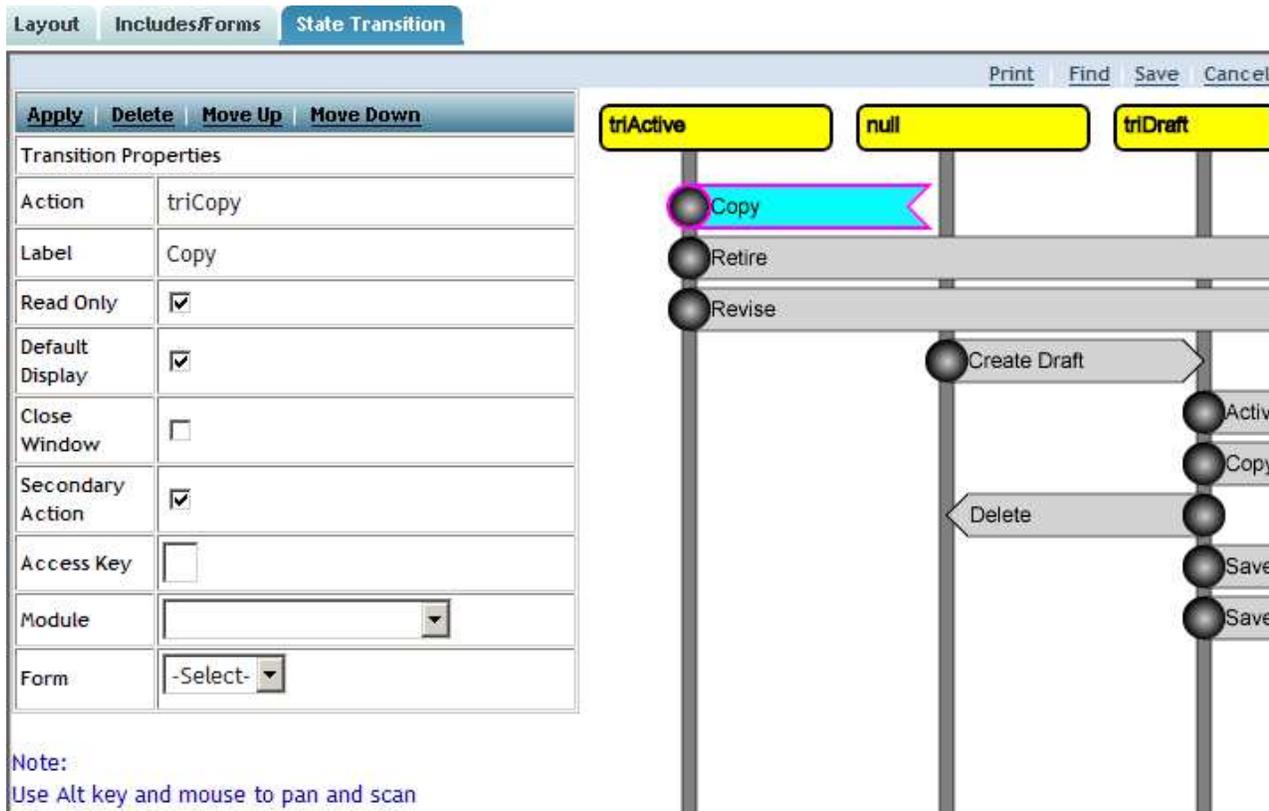
Figure 49. State Transition tab with selected transition

After you have selected a transition, you can delete it by clicking the *Delete* action that appears on the left side of the Form Wizard.

There may be transitions or states defined in the state transition family of the underlying business object that are not included in the form's *State Transition* tab. This can happen because they were previously deleted from the form's *State Transition* tab or because they were added to the state transition family of the underlying business object after the form was created.

If there are any transitions or states defined in the state transition family of the underlying business object that are not included in the form's *State Transition* tab, they can be added to the form's *State Transition* tab by clicking its *Find* action. When you click the *Find* action, a window pops up. The window contains a list of the transitions in the state transition family of the underlying business object that are not shown in the form's *State Transition* tab.

You can add state transitions that appear in this list to the form's *State Transition* tab by checking the check box for each state transition you want to add and then

clicking the *Ok* action. If you add any state transitions that are connected to a state that is not in the form's *State Transition* tab, that state is also added to the form's *State Transition* tab.

When a state transition is selected, there are properties you can specify for it that appear on the left side of the Form Wizard:

Read Only

When a state transition is triggered, the record's read-only property is set to true or false, depending on if the state it is transitioning to is read only. To make a state read only all of the transitions coming out of that state must have this check box checked. If a record's read-only attribute is set, then a form displaying the contents of the record will not allow the value of any of its fields to be changed.

Default Display

Determines whether the state transition action appears on the form as an action button by default. In other words, if you want this action to be selectable by the user on the form, check this box. If you want the action to be hidden, leave it unchecked. Hidden actions are still useful as they can be triggered by workflows or by IBM TRIRIGA Connector for Business Applications. Workflows are discussed in "Workflow building". IBM TRIRIGA Connector for Business Applications is described in the *IBM TRIRIGA Connector for Business Applications 3 Technical Specification*. Note that a sub-action can use its Inclusion Exclusion section to override the Default Display. See "Sub actions" for details about using Inclusion Exclusion to change the visibility of an action.

Close Window

If this check box is selected, then an action that triggers this state transition also closes the window that contains the Form Wizard.

Secondary Action

If this check box is selected, the action shows in the More menu for that form.

If this check box is not selected, the action appears as a button on the form's menu bar (with an exception, see below).

For example, if the set up is like that of the following figure,

Figure 50. Example of secondary action setup

where each action named Secondary has its Secondary Action property check box checked and each action named Primary has its Secondary Action property unchecked, the actions on the form will appear as shown in the following figure.
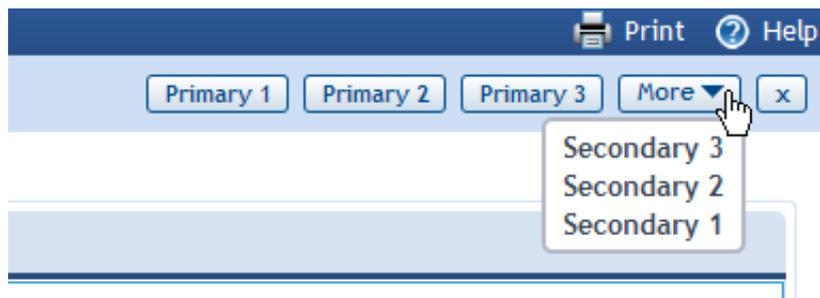
Figure 51. Result of secondary action example setup

Aside from allowing you to limit the number of actions that appear on the form's menu, this property also makes it possible for you to put rarely used or destructive actions, such as Copy or Retire, into the More menu so they are not as evident to the user and he or she is less likely to accidentally click one of them.

Note that there is a property in the TRIRIGAWEB.properties file that impacts this behavior, MAX_FORM_ACTION_NUMBER. This property

limits the number of buttons that can appear on the form menu, not including the More button or the X. For example, if we have a set up with 7 primary actions and MAX_FORM_ACTION_NUMBER is 4, the first 4 actions will appear on the form as buttons, and the last 3 will be in the More menu.

Access Key

The value of this property should be a single character. At runtime, pressing a key or a key chord and the Access Key character will be treated the same as clicking the action for the state transition. Different browsers use different keys or key chords, as follows:

- For Microsoft Internet Explorer: Alt+AccessKey+Enter
- For Mozilla Firefox: Alt+Shift+AccessKey
- For Google Chrome: Alt+AccessKey
- For Apple Safari: Alt+Control+AccessKey

Module

The value of this property is the module that contains the form used to edit the new record created if someone clicks an action that triggers this state transition.

If this property has no value, then triggering this state transition will not cause any records to be created.

The drop-down list in this property contains a list of the modules that are defined in the IBM TRIRIGA Application Platform environment.

Form

The value of this property is the name of the form used to edit the new record created if someone clicks an action that triggers this state transition. The business object associated with the named form is used to create the new record.

If this property has no value, then triggering this state transition will not cause any records to be created.

Pre-Popup Workflow

This property is only available if a pop-up form is defined for the selected state transition. Select from the synchronous workflows in the drop-down list. This workflow executes prior to the opening of the pop-up form.

A workflow can change which form pops up or cause no form to pop up. The technique uses the triPopupForm workflow parameter. The triPopupForm business object contains these fields:

triPopupFormModuleNameTX
The name of the module the pop-up form is defined in.

triPopupFormBoNameTX
The name of the business object the pop-up form is defined in.

triPopupFormNameTX
The name of the pop-up form.

You can change the values of these fields to change which form pops up, or you can clear the three fields to cause no form to pop up. If the workflow does not modify any of the field values, the form specified on the state transition pops up.

When a state transition is selected, there are *Move Up* and *Move Down* actions that you can click to move the selected state transition up or down in the form's *State Transition* tab. The sequence in which the transitions from a state appear from top to bottom in the form's State Transition tab corresponds to the left to right order in the form's action buttons, and corresponds to the top to bottom order in the More action button for Secondary Actions, as illustrated in the previous two figures.

State action labels containing two or more words will not wrap/break between the words if the browser window is resized smaller. Instead, the wrap/break occurs between different action labels.

Attention: Be sure to click the *Apply* action on the left side of the Form Wizard after you have finished editing the properties of the selected state transition. If you move on to something else without first clicking that *Apply* action, then any changes you made to the state transition's properties will be lost.

User messages

There are any number of messages that a form may be required to display to a user to explain a problem or to alert the user to a noteworthy situation. It is a common situation for multiple forms to be required to display some of the same messages.

Many of these messages have a fixed part that is always the same and a variable part that may be different each time the message is generated. To clarify this distinction between the fixed and variable parts of a message, consider this example. Suppose you want to generate a message that looks like this:

* Your start date 03/08/2014 must be before your end date 02/15/2014.

The fixed part of this notification would probably be:

* Your start date _____ must be before your end date _____.

The variable part of the message would be the pieces of text that get put in the blanks. In the case of the preceding example, these would be:
* 03/08/2014
* 02/15/2014

The mechanism that satisfies these needs involves two kinds of records that play different roles in the message creation process:

triUserMessage
> *triUserMessage* records are templates for messages. The fixed portion of a message's content is specified by a *User Message* record.

triUserMessageHelper
> *triUserMessageHelper* records supply the variable part of a message's content.

User message properties

triUserMessage records are templates for messages. They provide the fixed portions of a message's content. These records are normally created interactively by a person. *triUserMessage* records are found in Tools > System Setup > General > User Messages.

The value of the *ID* field is used by workflows to identify the *triUserMessage* record that is to be used for creating a message.

The value of the *Message Content* field is used to create the body of the message. Notice that the text in the *Message Content* field contains numbers in curly braces like *{2}*.

Each of these numbers in curly braces is a place holder for some text that can be supplied from a *triUserMessage* record. The contents of the *Description* field should describe the purpose of each of the place holders.

Tip: If you want to display a message containing a single quote (for example, I'm {1}), you need to escape the single quote by adding another single quote (for example, I''m {1}).

The value of the *Language* field identifies the language in which the value of the *Message Content* field is written.

Tip: By creating User Message records with the same ID but different languages, you can configure messages that are automatically translated for your users.

triUserMessageHelper

To create finished message text, a workflow creates a *triUserMessageHelper* record from the *triUserMessageHelper* business object in the *triHelper* module. After creating the record, the workflow sets the values of its fields and then finishes the message by triggering a *Calculate* action on the *triUserMessageHelper* record. After the message is finished, the workflow that triggered the *Calculate* action can copy the finished message text to the appropriate field in a form.

Here are descriptions of the relevant fields in a *triUserMessageHelper* record:

triIdTX
> Workflows must set the value of the *triIdTX* field to the ID of the *triUserMessage* record that will provide the fixed portion of the message's content.

triLanguageLI
> Workflows must set the value of the *triIdTX* field to the Language of the *triUserMessage* record that will provide the fixed portion of the message's content. For a given ID there could be text for multiple languages. This field determines which message to retrieve. If the language selected is not found, the helper looks for a message in "US English".

triInput1TX, triInput2TX, ..., triInput9TX
> These fields correspond to the place holders *{1}*, *{2}*, ..., *{9}* that may appear in the fixed content provided in the *triUserMessage* record. The value in each of these fields will be substituted for the corresponding place holder in the content.

triUserMessageTX
> Triggering a *Calculate* action on the *triUserMessageHelper* record launches a workflow that creates a finished message with the value of the appropriate field substituted for each place holder. After the workflow is done, the finished message is in the *triUserMessageTX* field.

Chapter 7. Workflow building

Before you actually create or modify any workflows, you must first navigate to a part of the IBM TRIRIGA Application Platform called the *Workflow Builder*. The Workflow Builder is used to create new workflows and to modify existing workflows.

To navigate to the Workflow Builder, click the *Workflow Builder* item in the *Administration* menu. The Workflow Builder will appear.

Workflow Builder

The Workflow Builder organizes workflows by the module with which they are associated. It shows a list of the workflows associated with whichever module is selected. You can select a module by clicking its name on the left side of the Workflow Builder. Clicking the name of a module causes workflows associated with the module or one of the business objects in the module to be listed on the right side of the Workflow Builder.

Most workflows are associated with a particular business object in a module. If many business objects in a module have associated workflows, you may prefer to select just one business object and see just the workflows associated with that business object.

If you do not see the names of business objects in the relevant module, click the plus sign (+) next to the module's name to expand the list. The business objects in the module now show under the module's name. Also the plus sign (+) next to the module name changes to a minus sign (-). To contract the list of business objects, click the minus sign (-).

The Workflow Builder has a menu of actions that can be performed on workflows:

New   Click this action to create a new workflow.

Copy

> Sometimes, you want to create a new workflow that is similar to an existing workflow. It is easier to create a similar workflow by starting with a copy of the selected workflow and modifying it.
>
> Clicking the *Copy* action creates a new workflow that is a copy of the currently selected workflow.

Publish

> Click this action to make the selected workflow available for use. New workflows are not available for use until they are published.
>
> When you modify an existing workflow, the modified workflow is not automatically put into use. The modified revision of the workflow does not replace the revision of the workflow currently in use until the modified revision of the workflow is published.
>
> Publishing is described in "Create-Publish-Revise cycle".

Revise

The *Publish* action sets the state of a workflow to *Published*. As a safeguard against unintentional modification, while the state of a workflow is *Published* you cannot modify the workflow.

If you want to revise a published workflow, click the *Revise* action. This changes the state of the selected workflow to *Revision In Progress*. While the state of the workflow is *Revision In Progress* you are allowed to modify the workflow.

For more information workflow revisions, see "Workflow revisions." Revising is also described in "Create-Publish-Revise cycle."

Retire

Clicking the Retire action takes the currently selected workflow out of use. It can be put back in use by publishing the workflow again.

If the workflow to be retired is referenced by a Call Workflow task in another workflow, the system displays a warning. Retiring is allowed after confirmation. To see the list of workflows calling the selected workflow, open the workflow's Start task and click the Callers action on the Workflow Properties section bar.

Delete

Clicking the Delete action deletes the selected workflow.

If the workflow to be deleted is referenced by a Call Workflow task in another workflow, the system displays a warning and will not allow the deletion. To see the list of workflows that call the selected workflow, open the workflow's Start task and click the Callers action on the Workflow Properties section bar.

The Workflow Builder has a menu of actions that give you information about a workflow. First select the radio button next to the name of the workflow. Then select one of the following actions:

List Active Instances

Shows the record of the selected workflow being run.

List All Instances

Shows instances of this workflow that are still running.

List All Revisions

Shows all revisions of this workflow. The system creates a history record whenever you change a workflow. For more information, see "Workflow Revisions."

Where Used

A window pops up showing what references or uses the selected workflow. The fields displayed are: Name, Type, Module, Object, Form, Action, and Additional Information. The Type could be other workflows, business object pre-create workflows, smart section workflows to initialize a record, form pre-load workflows, form query section workflows, form Gantt section workflows, form availability legacy section workflows, form stacking section pre-move workflows, form availability section workflows, form calendar section workflows, form action workflows and pre-form workflows, query action pre-create workflows, state transitions, and manager action pre-create workflows.

If you want to export the information in the Where Used window, click the Export Usage action at the top of the window. You will be offered the

choice of saving or opening a csv file. Clicking the linked object opens the builder for the reference for most object types.

Refer to "Deleting fields" for a list of references reported by Where Used.

The *workflow editor* allows you to view or modify a workflow. Clicking the *New* action causes the workflow editor to pop up so you can edit the new workflow. To modify or view an existing workflow in the workflow editor, click the workflow's name in the Workflow Builder.

Workflow Editor

The Workflow Editor allows you view and modify workflows. The Workflow Editor pops up in its own window and shows the tasks of a workflow as shapes. Each kind of task has a different color and shape.

The Workflow Editor shows arrows connecting tasks. The purpose of the arrows is to show the order in which the tasks will be performed.

When you create a new workflow, the workflow has two tasks: a Start task and an End task. The arrow in a newly created workflow shows that after the Start task is performed, the next task to be performed is the End task.

The Workflow Editor is organized into three sections called:
* diagram
* properties
* task palette

The diagram section is the only section of the workflow editor that is always visible.

The properties section of the workflow becomes visible when you click one of a workflow's tasks. The properties section always appears at the bottom of the window. It shows the properties of the task that was clicked. You can tell which task's properties the properties section is showing because the task is highlighted in the diagram section.

Each kind of task has a different set of properties. The properties of each kind of workflow task are described as part of the discussion of each kind of task.

If the properties section is already visible and you click a different task, that task's properties are displayed in the properties section. To make the workflow editor stop displaying any properties section, click the background of the diagram section.

The task palette section appears when you move the mouse pointer over the New Task bar on the left side of the workflow editor. The shapes in the task palette correspond to the different kinds of tasks that can be in a workflow.

Depending on the context and properties set in the workflow's Start task, some kinds of tasks may not be visible in the task palette. If a kind of task is visible in the task palette only under certain conditions, those conditions are noted in the description of the individual workflow task.

Use the task palette to add a task to the workflow. To add a task to the workflow, move the mouse pointer over the *New Task* bar. After the task palette appears, click the task's shape in the task palette. The task palette disappears except for the shape that you clicked. As you move the mouse pointer, the shape follows the mouse. Use the mouse to move the task to the end of the arrow that should lead to the task.

When you move the shape to the end of an arrow, the workflow editor shows a preview of what the workflow will look like if you leave the task where it is. At this point you can click the task shape to leave it where it is or move the mouse pointer to take the task shape somewhere else.

To delete a task from a workflow, click the task to make its properties visible in the properties section. If the task can be deleted (some tasks cannot be deleted under certain circumstances), there will be a *Delete* action in the menu of the properties section. Clicking the *Delete* action deletes the task.

Workflow changes take effect immediately

You may have noticed that nowhere on the workflow editor is there an *Ok*, *Save,* or *Cancel* action to click. The workflow editor works differently in this respect from the rest of the IBM TRIRIGA Application Platform. All changes you make to a workflow or to the properties of a task take effect as you make them and click elsewhere, on another task or in the background. They are not saved if you close your browser without clicking elsewhere.

Workflow state transitions

If a workflow triggers a state transition action and that action has multiple sub actions, only the first sub action runs. The first sub action is determined by listing the sub action labels alphabetically. Use actions that have at most one sub action instead of actions with multiple sub actions.

Workflow naming conventions

When you design or develop new workflows, they are easier to understand if you use a naming convention for your tasks and workflows. This convention allows others who review the workflow to know your intentions for the workflow as a whole and for each task in the workflow without interpretation. Use the standard naming conventions for each task type, and add a description to each task that further clarifies its intent.

For business object names, use the business object upon which the task is performing work. If the task could be doing work on any business object in the module, use the module name. You might need to further qualify the business object name to give it a more descriptive context. For example, if multiple employee records could be used in the workflow, you could use "Supervisor" and "Employee" to distinguish them. If you are referencing a business object that is associated to another business object, indicate both business objects and the association relationship.

Name all tasks in a workflow a unique name. Remember to add context where required.

If a workflow task is completely custom, prefix the task name with cst. For example, cstIterate for each triAsset.

If a workflow task is an as-delivered task that has at least one modification, do not change the task name. The object revisioning capabilities eliminate the need to rename and modify as-shipped IBM TRIRIGA objects to suit your business needs.

A best practice for a completely custom workflow is to prefix the Start task and all tasks within the workflow with cst. When the business object is also custom and is in a completely custom workflow, use cst only one time. For example, cstWidget - triSave - Trigger Approval Process.

Tip: Microsoft Word often changes the minus sign in a workflow task name or a query name to a hyphen. If you copy workflow task names or query names from a document that you prepared in Word or from a PDF of a Word document, you need to change the hyphen to a minus sign. The Object Migration tool does not recognize the hyphen and does not import your work.

The following table contains naming standards for completely custom workflow and tasks that you create.

Attention: When you modify as-shipped IBM TRIRIGA objects to suit your business needs as opposed to creating your own objects, do not use a cst name prefix or follow the naming conventions. With he new object revisioning and object labeling features in IBM TRIRIGA Application Platform, you can modify as-shipped objects without the need for naming conventions. Your objects are saved in revisions each time you revise them.

For more information on object revisioning capabilities, the new object conversion process, and the upgrade best practices, see Object Labels and Revisions on the IBM TRIRIGA wiki.

Attention: Do not use the ', <, >, =, or % special characters in workflow names or workflow task names. Do not put a space after the "cst" or the "cst-".

Table 22. Naming conventions for completely custom workflows

Task	Naming Convention	Example
Start Task Asynchronous Workflow Workflow is completely custom and is against an as-delivered object	cst + Business Object Name - Event that triggers the workflow - Description or cst + Business Object Name - Associate - Secondary Business Object Name - Description or cst + Module Name - Event that triggers the workflow - Module Level Description	cstTriSpace - triSave - Trigger Approval Process or cstTriPeople - Associate - triSpace - Primary Location or cstTriContract - triCreateDraft - Module Level Call Update Dependent Records

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Start Task Asynchronous Workflow Workflow is completely custom and is against a custom object	Business Object Name - Event that triggers the workflow - Description or Business Object Name - Associate - Secondary Business Object Name - Description or Module Name - Event that triggers the workflow - Module Level Description	cstWidget - triSave - Trigger Approval Process or cstWidget - Associate - triSpace - Primary Location or cstWidget - triCreateDraft - Module Level Call Update Dependent Records
Start Task Asynchronous Workflow Workflow is a modification of an as-delivered workflow	cst- + Business Object Name - Event that triggers the workflow - Description or cst- + Business Object Name - Associate - Secondary Business Object Name - Description or cst- + Module Name - Event that triggers the workflow - Module Level Description	cst-triLand - Cancel - Reset Meta-data or cst-triPeople - Associate - triSpace - Primary Location or cst-triPayment - triComplete - Module Level Action Item to get Response
Start Task Subflow Workflow Workflow is completely custom and is against an as-delivered object	cst + Business Object Name - Subflow - Description	cstTriAsset - Subflow - Count asset inventory
Start Task Subflow Workflow Workflow is completely custom and is against a custom object	Business Object Name - Subflow - Description	cstWidget - Subflow - Count asset inventory
Start Task Subflow Workflow Workflow is a modification of an as-delivered workflow	cst- Business Object Name - Subflow - Description	cst-triAsset - Subflow - Count asset inventory

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Start Task Synchronous Workflow Workflow is completely custom and is against an as-delivered object	cst + Business Object Name - Synchronous - Description	cstTriLocation - Synchronous - Populate parent fields on create
Start Task Synchronous Workflow Workflow is completely custom and is against an a custom object	Business Object Name - Synchronous - Description	cstWidget - Synchronous - Populate parent fields on create
Start Task Synchronous Workflow Workflow is a modification of an as-delivered workflow	cst- + Business Object Name - Synchronous - Description	cst-triLocation - Synchronous - Populate parent fields on create
Create Record Task	If the task is completely custom: cst + Create Business Object Name from Business Object Name (source) or If the task is a modification of an as-delivered task: cst- + Create Business Object Name from Business Object Name (source)	cstCreate triAddress from Building Primary Address or cst-Create triAddress from Building Primary Address
Modify Records Task	If the task is completely custom: cst + Set Business Object Name (target) from Business Object Name (source) or If the task is a modification of an as-delivered task: cst- + Set Business Object Name (target) from Business Object Name (source)	cstSet triProperty from Location or cst-Set triProperty from Location

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Retrieve Records Task	If the task is completely custom: cst + Get Business Object Name (target) from Business Object Name (source) or If the task is a modification of an as-delivered task: cst- + Get Business Object Name (target) from Business Object Name (source)	cstGet triProfile from triGroup associated to triBuilding or cst-Get triProfile from triGroup associated to triBuilding
Associate Records Task	If the task is completely custom: cst + Assoc Business Object Name (source) to Business Object Name (target) or cst + DeAssoc Associate Business Object Name (source) from Business Object Name (target) or If the task is a modification of an as-delivered task: cst- + Assoc Business Object Name (source) to Business Object Name (target) or cst- + DeAssoc Associate Business Object Name (source) from Business Object Name (target)	cstAssoc triNotification to triLocation or cstDeAssoc triPeople from triLocation or cst-Assoc triNotification to triLocation or cst-DeAssoc triPeople from triLocation
Trigger Action Task	If the task is completely custom: cst + Action [space] Business Object Name or If the task is a modification of an as-delivered task: cst- + Action [space] Business Object Name	cstRemove Temp triAddress; Issue triPurchaseOrder or cst-Remove Temp triAddress; Issue triPurchaseOrder

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Delete Reference Task	If the task is completely custom: cst + Delete Reference to Business Object Name (source) from Business Object Name (target) or If the task is a modification of an as-delivered task: cst- + Delete Reference to Business Object Name (source) from Business Object Name (target)	cstDelete Reference to triProfile from triGroup Associated to triBuilding or cst-Delete Reference to triProfile from triGroup Associated to triBuilding
Add Child Task	If the task is completely custom: cst + Add Child Business Object Name (source) to Business Object Name (target) or If the task is a modification of an as-delivered task: cst- + Add Child Business Object Name (source) to Business Object Name (target)	cstAdd Child triSpace to triFloor or cst-Add Child triSpace to triFloor
Set Project Task	If the task is completely custom: cst + Set Project of Business Object Name (target) from Business Object Name (source) or If the task is a modification of an as-delivered task: cst- + Set Project of Business Object Name (target) from Business Object Name (source)	cstSet Project of triInvoiceItem from triContract or cst-Set Project of triInvoiceItem from triContract

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Attach Format File Task	If the task is completely custom: cst + Attach File Name to Business Object Name or If the task is a modification of an as-delivered task: cst- + Attach File Name to Business Object Name	cstAttach Deleted Cost Code.rpt to triNotification or cst-Attach Deleted Cost Code.rpt to triNotification
Query Task	If the task is completely custom: cst + Get query or If the task is a modification of an as-delivered task: cst- + Get query	cstGet triPeople Associated with triBuildings or cst-Get triPeople Associated with triBuildings
User Action Task	If the task is completely custom: cst + Assign User or Group Name to Action Business Object Name or If the task is a modification of an as-delivered task: cst- + Assign User or Group Name to Action Business Object Name	cstAssign Admin Group to Remove triContactRoles Associated to Retired triPeople or cst-Assign Admin Group to Remove triContactRoles Associated to Retired triPeople
Approval Task	If the task is completely custom: cst + Assign User or Group Name to Approve Business Object Name or If the task is a modification of an as-delivered task: cst- + Assign User or Group Name to Approve Business Object Name	cstAssign At Bat triPeople to Approve triPurchaseOrder or cst-Assign At Bat triPeople to Approve triPurchaseOrder

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Schedule Task	If the task is completely custom: cst + Schedule Action for Business Object Name or If the task is a modification of an as-delivered task: cst- + Schedule Action for Business Object Name	cstSchedule triFinalIssue for triBidResponse or cst-Schedule triFinalIssue for triBidResponse
Get Temp Record Task	If the task is completely custom: cst + Get Temp Business Object Name **Note:** When performing other tasks to the results of a Get Temp Record task, use the modifier "Temp" for the Business Object Name. For example, if you are using a Modify Records task, cstAdd Temp triProperty:triAddressOther with triAddress"	cstGet Temp triPeople
Modify Metadata Task	If the task is completely custom: cst + Set Context for Business Object Name, Form Name (if to a particular form) or If the task is a modification of an as-delivered task: cst- + Set Context for Business Object Name, Form Name (if to a particular form)	cstSet ID to Read Only for triPeople or cst-Set ID to Read Only for triPeople
Save Permanent Record Task	If the task is completely custom: cst + Commit Temp Business Object Name or If the task is a modification of an as-delivered task: cst- + Commit Temp Business Object Name	cstCommit Temp triPeople or cst-Commit Temp triPeople

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Iterator Task	If the task is completely custom: cst + Iterate for each Business Object Name or If the task is a modification of an as-delivered task: cst- + Iterate for each Business Object Name	cstIterate for each triPeople or cst-Iterate for each triPeople
Custom Task	If the task is completely custom: cst + Call Context or If the task is a modification of an as-delivered task: cst- + Call Context	cstCall Update triTask triAcutalEndDT or cst-Call Update triTask triAcutalEndDT
Populate File Task	If the task is completely custom: cst + Populate Context with Business Object Name or If the task is a modification of an as-delivered task: cst- + Populate Context with Business Object Name	cstPopulate bid.xls with triBid or cst-Populate bid.xls with triBid
Distill File Task	If the task is completely custom: cst + Distill Context into Business Object Name or If the task is a modification of an as-delivered task: cst- + Distill Context into Business Object Name	cstDistill bid.xls into triBid or cst-Distill bid.xls into triBid

Table 22. Naming conventions for completely custom workflows (continued)

Task	Naming Convention	Example
Call Workflow Task	If the task is completely custom: cst + Call Business Object Name - Context or If the task is a modification of an as-delivered task: cst- + Call Business Object Name - Context	cstCall triTask - Permanent Save Validation (this is the synchronous workflow name with the " - Synchronous" removed) or cst-Call triTask - Permanent Save Validation (this is the synchronous workflow name with the " - Synchronous" removed)
DataConnect Task	If the task is completely custom: cst + Process Business Object Name or If the task is a modification of an as-delivered task: cst- + Process Business Object Name	cstProcess triPOBody or cst-Process triPOBody
Variable Definition Task	If the task is completely custom: cst + Define Variable Name or If the task is a modification of an as-delivered task: cst- + Define Variable Name	cstDefine triInvoiceCount or cst-Define triInvoiceCount
Variable Assignment Task	If the task is completely custom: cst + Assign Variable Name or If the task is a modification of an as-delivered task: cst- + Assign Variable Name	cstAssign triInvoiceCount or cst-Assign triInvoiceCount

Workflow tasks

Here are detailed descriptions of the tasks that can be included in a workflow.

Start task

A Start task is always the first task in a workflow. It cannot be deleted or moved. The properties of a Start task are in fact properties for the entire workflow.

Because you cannot delete a Start task, it does not have a *Delete* action in the menu for its properties as do most other kinds of tasks. The following actions may appear in the menu for a Start task's properties:

Publish

> This action appears only if the workflow is new or under revision. Clicking this action makes a new workflow available for use or puts revisions to an existing workflow into use.
>
> Publishing is described in "Create-Publish-Revise cycle".

Revise

> This action appears only if the workflow has been published. Clicking this action creates a new version of the workflow that you can revise without affecting the version of the workflow that is currently in use.
>
> Revising is described in "Create-Publish-Revise cycle".

Retire

> Clicking this action takes the existing version of the workflow out of use. It can be put back into use by publishing it again.
>
> If the workflow to be retired is referenced by a Call Workflow task in another workflow, the system displays a warning. Retiring is allowed after confirmation. To see the list of workflows calling the selected workflow, click the Callers action on the Workflow Properties section bar.
>
> Retiring is described in "Create-Publish-Revise cycle".

Callers
> Click this action to see a list of the workflows that call this workflow.

Parameters
> Clicking this action causes a list of the parameters and return values in this workflow to appear. If the workflow is synchronous or subflow you can define input parameters and return values. More information about workflow parameters and return values can be found in "Example: Workflow parameters and return values (start)".

The Start task always refers to the permanent version of the record in the database. Any workflow tasks that refer back to the business object of the Start task are referring to the last saved version of the record. This means that any changes that workflow tasks make to the Start task record are made immediately to the record in the database. The exception would be if the workflow is marked as a temporary workflow or using DataConnect with Temporary setting.

The properties in a Start task's properties are actually properties for the entire workflow. A Start task's Workflow Properties section has these fields:

Name

> This is the name of the workflow. This is the name that will appear in the Workflow Builder and the name you will use to identify the workflow.

The name of the workflow is not the Start task's label. Start tasks do not have a label that you can change. The label that appears on the Start task in the workflow editor's diagram section is always *Start*.

Workflow naming conventions are discussed in "Workflow naming conventions".

Description

This is a description of the whole workflow.

Concurrence

This property is used to determine whether the workflow will be synchronous, subflow, or asynchronous. The distinction between synchronous and asynchronous is explained in "Synchronous versus asynchronous workflows".

If you select the *Synchronous* radio button, the workflow will be synchronous. As a synchronous workflow, it will have access to temporary data. However, there will be no way to associate the workflow with events, since synchronous workflows cannot be associated with events. Nor can a synchronous workflow be used to migrate data from staging tables into IBM TRIRIGA records because the Integration property is not available to synchronous workflows.

Synchronous workflows that are triggered by actions on IBM TRIRIGA records' forms are processed to completion before allowing the user to continue. This makes them good for performing validations or automating processes that need to complete before the user performs their next task.

If you select the Subflow radio button, the workflow allows required parameters. Standard synchronous workflows only allow optional parameters. A Subflow workflow can be selected only in a Call Workflow task. A Subflow workflow cannot be selected as the workflow to run for a form action, a state action, or the like.

If you select the *Asynchronous* radio button, the workflow will be asynchronous. As an asynchronous workflow, it will not have access to temporary data. However, you will be able to associate the workflow with an event, since asynchronous workflows are launched in response to an event. An asynchronous workflow can be used to migrate data from staging tables into IBM TRIRIGA records because the Integration property is available to asynchronous workflows.

Temporary Data

This property is only visible if the value of Concurrence is Synchronous.

The value of this property determines whether the workflow will be accessing temporary or permanent data. This distinction is described in "Temporary versus permanent data".

If the *Permanent* radio button is selected, the workflow only can access permanent data that is stored in the record used to launch this workflow.

If the *Temporary* radio button is selected, the workflow can access temporary or permanent data that is stored in the record used to launch this workflow.

Module

The value of this property is the module with which the workflow is associated. This association between workflows and modules serves two purposes:

- The Workflow Builder uses the association between workflows and modules to organize workflows by module.
- If the workflow is associated with a particular business object, the module is used to help identify the business object with which the workflow will be associated.

Set the value of this property by clicking the Module icon. Clicking the Module icon causes a list of modules to pop up. Click the module name you want to be the value of this property.

Object Type

Workflows are launched as a result of an action or system event being performed on a record.

The value of this property is either the name of a business object in the specified module or *-Any-*. If the value of this field is the name of a business object, the workflow can be launched only as a result of something happening that is related to a record created from the named business object.

If the value of this property is *-Any-*, the workflow can be launched by something happening related to a record created from any business object in the specified module. The workflow will be restricted to accessing only the fields in a record it is launched for that have the same name as fields in the module's base business object.

Save Workflow Instances

If this check box is checked, a record will be kept of every time this workflow is run. The record includes the execution path that the workflow followed.

To see the record of this workflow being run, click the *List All Instances* action at the top of the Workflow Builder. To see just instances of this workflow that are still running, click the *List Active Instances* action at the top of the Workflow Builder.

The statuses in the list of workflow instances are described in "Workflow instances".

It is also possible to see a list of workflow instances that have run from a particular record. This is discussed in "Workflow instances".

If this check box is not checked (the default for new workflows), no record will be kept of this workflow running. This can produce a noticeable speed improvement.

Lock Record For Other Users

The forms for User Action tasks and Approval tasks have a check box with this same label. The value of this check box is used as the default value for the like-named check boxes for User Action and Approval tasks.

The User Action task is documented in "Approval task".

Propagate Integration Status

The Propagate Integration Status check box indicates whether the workflow should consider the Integration property. If the workflow is

called by an Integration workflow, the Integration status is propagated if the Propagate Integration Status property is checked. If the workflow is called by an Integration workflow and the Propagate Integration Status property is not checked, the Integration status of the calling workflow is not propagated. The default is for this property to be checked.

Integration is a property available to asynchronous workflows. The Integration property is discussed below.

The Start task properties if the workflow is subflow are in the following figure.
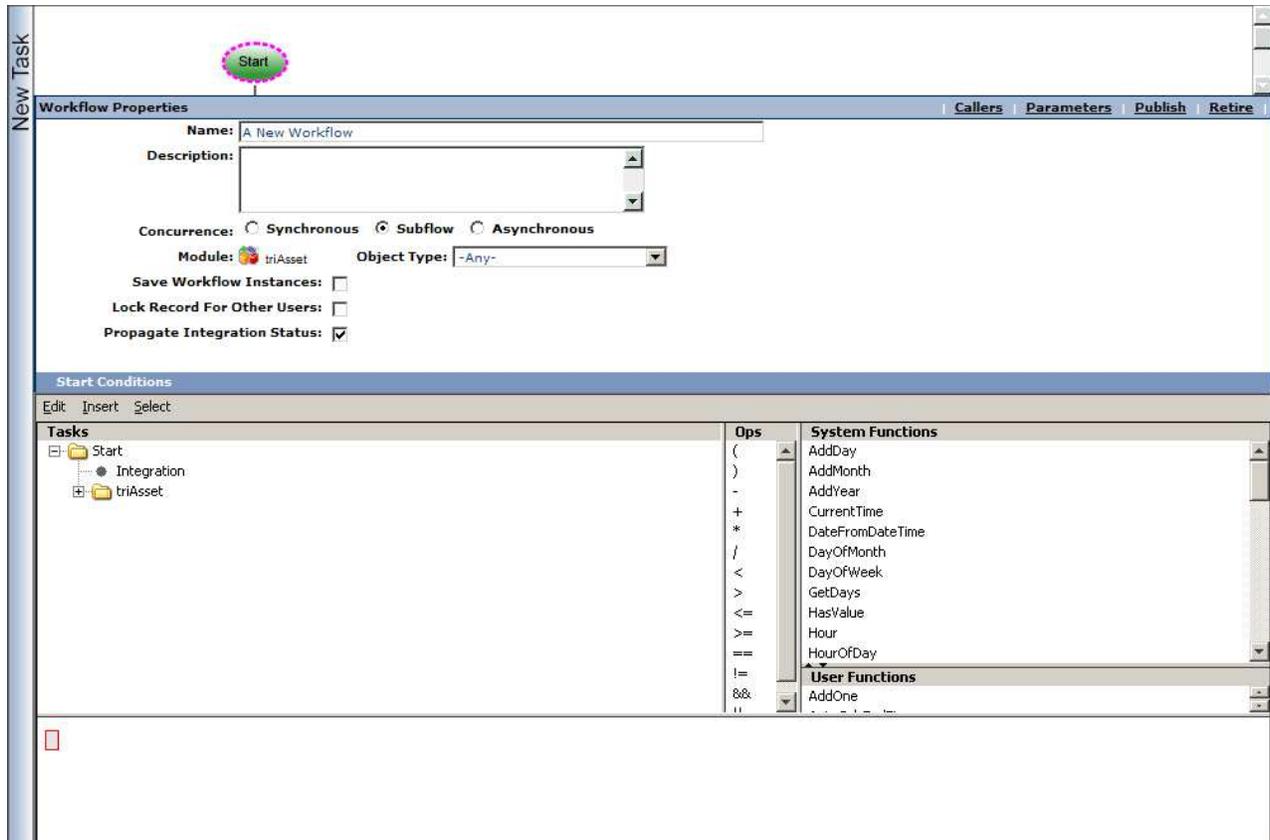

Figure 52. Start Task properties for subflow workflow

The following additional properties may be visible only if the workflow is asynchronous.
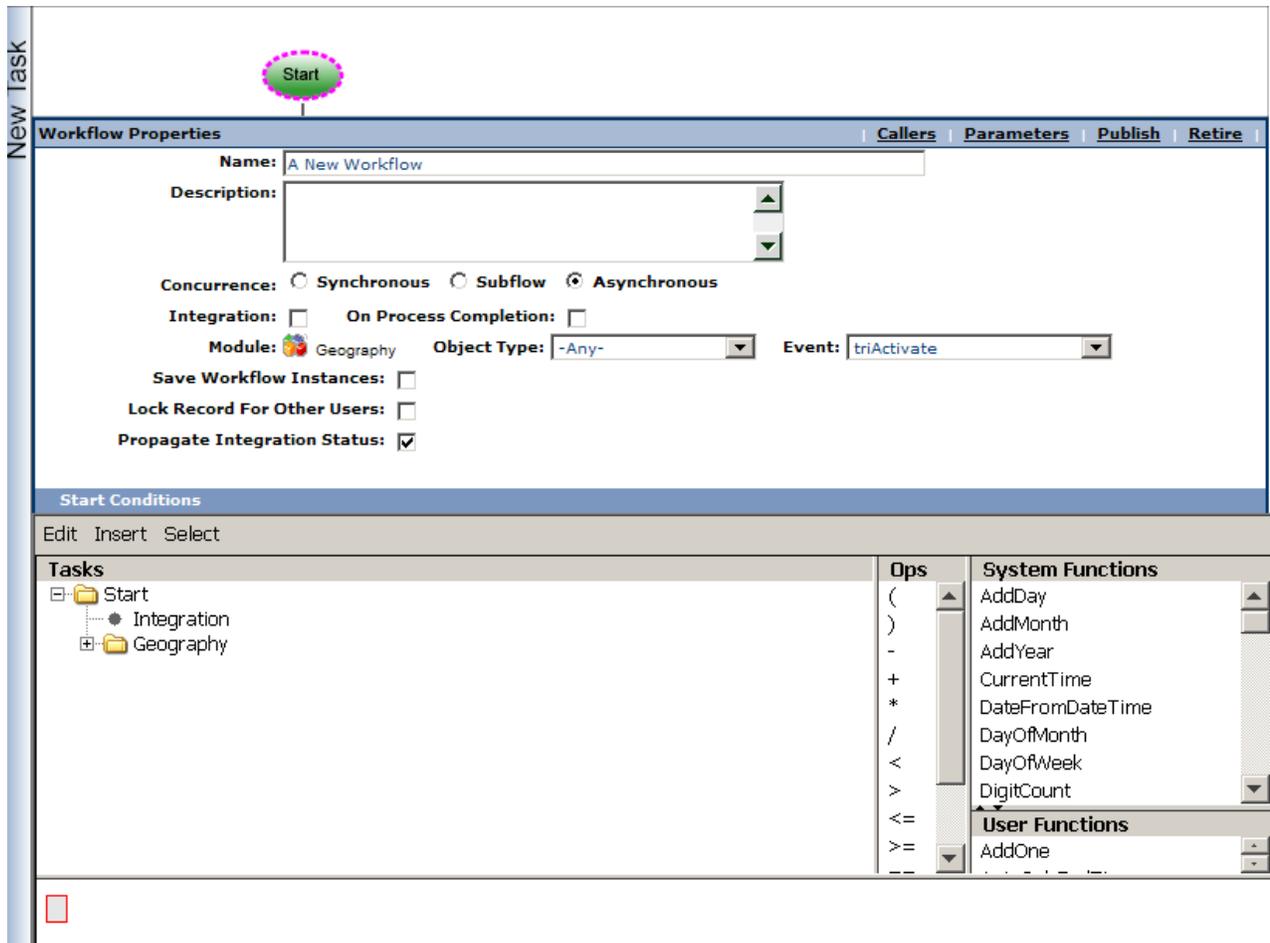
Figure 53. Start Task properties for asynchronous workflow

Integration

This property is always visible if the workflow is asynchronous.

When the property is selected, the workflow is used to migrate data from staging tables into IBM TRIRIGA records. This type of workflow is used extensively in IBM TRIRIGA DataConnect. More information about DataConnect can be found in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

Tasks can check the Integration status to control flow through the workflow.

If an event is generated while the workflow has Integration status on and the Propagate Integration Status flag on, when that event is processed the workflow context for the workflows run will have the Integration status set on (true).

The Integration status can be included in a condition expression in a Start task condition or a Switch task condition. For example, if there are steps in a workflow that should not be called if run from an Integration workflow, check the Integration setting with a Switch Condition or in the Start Condition.

On Process Completion

This property is always visible if the workflow is asynchronous.

When this property is selected, the workflow only runs when all other asynchronous workflows for the primary event (the initiating event) have been run. For example, assume the On Process Completion check box is checked for workflow W1 that fires on a Save event. Workflows W2, W3, and W4 also fire on the Save event. A user clicks Save. W1 does not start until W2, W3, and W4 have finished.

The initiating event is specified by the values of the Module, Object Type, and Event properties.

When the workflow agent detects that a process has completed, it posts a special event. The event is constructed by taking the event name and the record from the initiating event information. The workflow lookup process uses the initiating event information to identify the event as being for process completion and restricts the workflow lookup to be for On Process Completion type workflows for the given module, business object, and event name of the event.

Only one process completion event is posted for a given process.

Use an On Process Completion workflow for final processing and clean up. Do not use it to initiate another batch of involved processing.

To aid in debugging, the event name used in the agent thread name has (pc) appended to it, for example, triActivate(pc).

Event

This property is always visible if the workflow is asynchronous.

The value of this property is the action or system event that can launch this workflow. The launching of workflows is discussed in "Launch conditions".

This field is a drop-down list that contains system events and the actions that are used by the state families available to business objects in the specified module.

If an action from the business object's or module's state transition family is chosen, it means that the workflow will be asynchronous. The workflow will be launched when the specified action is performed on a record created from the specified business object. If the specified business object is -*Any*-, the workflow will be launched when the specified action is performed on a record created from any business object in the specified module.

If a system event is selected from the drop-down, the workflow is launched when the system event happens to a record created from the specified kind of business object. System events are discussed in "System events that trigger workflows".

If the selection from the drop-down is the system event *Associate* or the system event *De-Associate*, three additional fields appear to allow the affected association to be specified: Associated Module, Object Type, and Association.

Associated Module

This field is visible only if the selected value for Event is *Associate* or *De-Associate*.

The value of this property is the name of the module in which the business object used to create records at the other end of the association is defined.

Object Type

> This property is visible only if the selected value for Event is *Associate* or *De-Associate*.

> The value of this property is the name of the business object used to create records at the other end of the association.

Association

> This property is visible only if the selected value for Event is *Associate* or *De-Associate*.

> This is the name of the affected association.

At the bottom of each of the example Start task figures above is a section labeled *Start Conditions*. You can use the *Start Conditions* section to specify that the workflow should only be launched under certain conditions.

You can specify more than one start condition. If more than one start condition is specified, the workflow will launch only if all the start conditions are true.

The mechanics of specifying start conditions are described in "Example: Workflow condition builder".

Example: Workflow parameters and return values (start)

The Workflow Parameters and Return Values screen appears whenever you click the Parameters action. It shows the parameters and return values for the workflow.

An example best explains how to use the Workflow Parameters and Return Values screen. The following figure starts this example with a subflow workflow containing variables to hold the input parameters and the value to be returned. The example could also be a synchronous workflow, except the parameters must be optional. The following figure shows the Start task of the example workflow; the user is about to click the Parameters action.
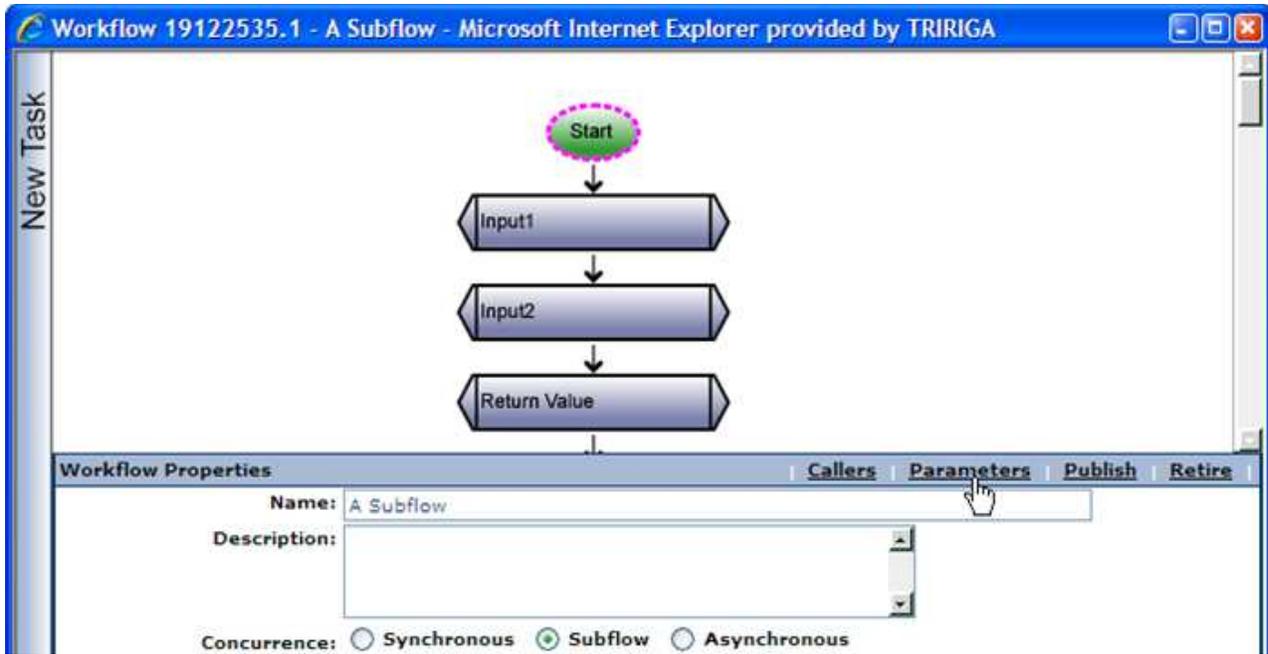
Figure 54. Start task example

> The Workflow Parameters and Return Values screen appears as shown in the following figure. Note there are two sections: the Parameters section and the Return Values section. Clicking Add in the Parameters section allows you to define input parameters. Clicking Add in the Return Values section allows you to define return values. In order to have either of these the workflow must contain some variables. Variables are defined with the Variable Definition task, which is described in "Variable definition task".

Figure 55. Initial Workflow Parameters and Return Values screen

> Click Add in the Parameters section. The Workflow Parameter Definition screen appears as shown in the following figure.
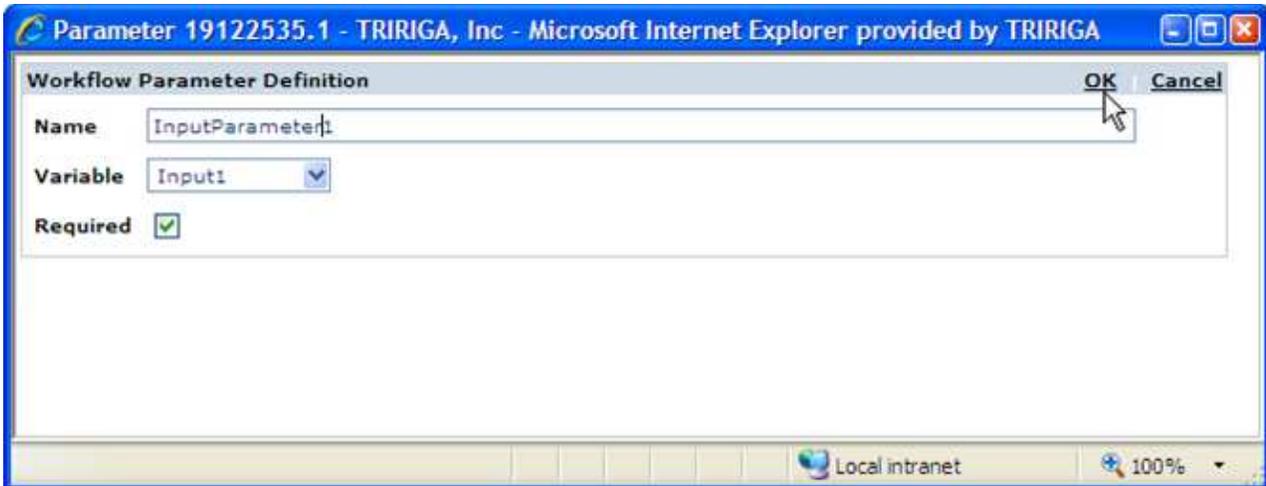
Figure 56. Adding a required parameter

Give the parameter a Name. The choices in the Variable drop down are the variables defined in the workflow. Check the Required check box if the parameter is required. In a synchronous workflow, this option is not available because all synchronous workflow parameters are optional. The following figure shows adding an optional parameter.
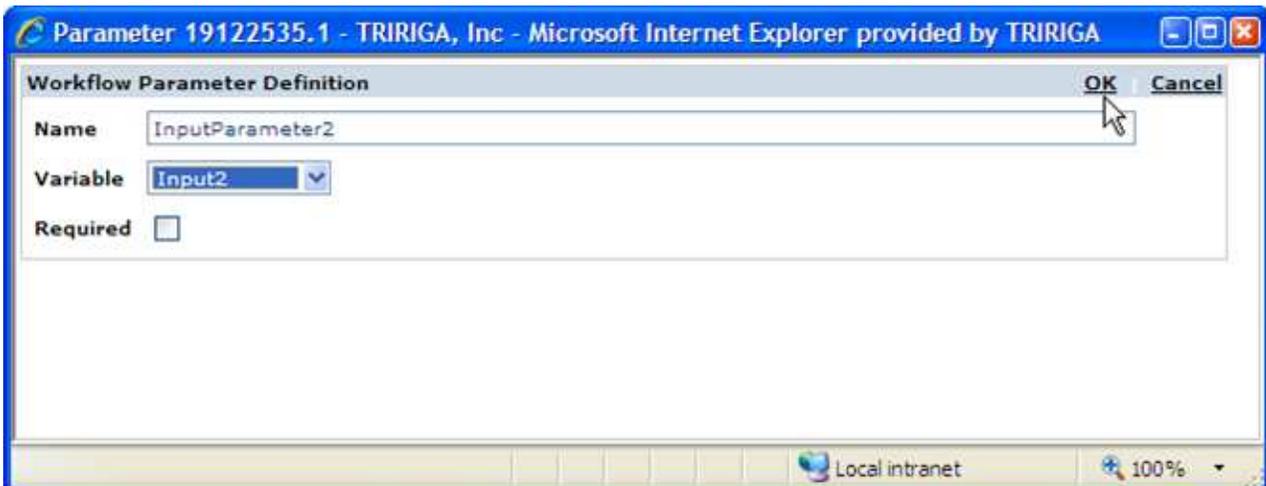
Figure 57. Adding an optional parameter

Click the Add action in the Return Values section. When adding a Return Value, specify its Name and select its source from the Variable drop-down list, as shown in the following figure.
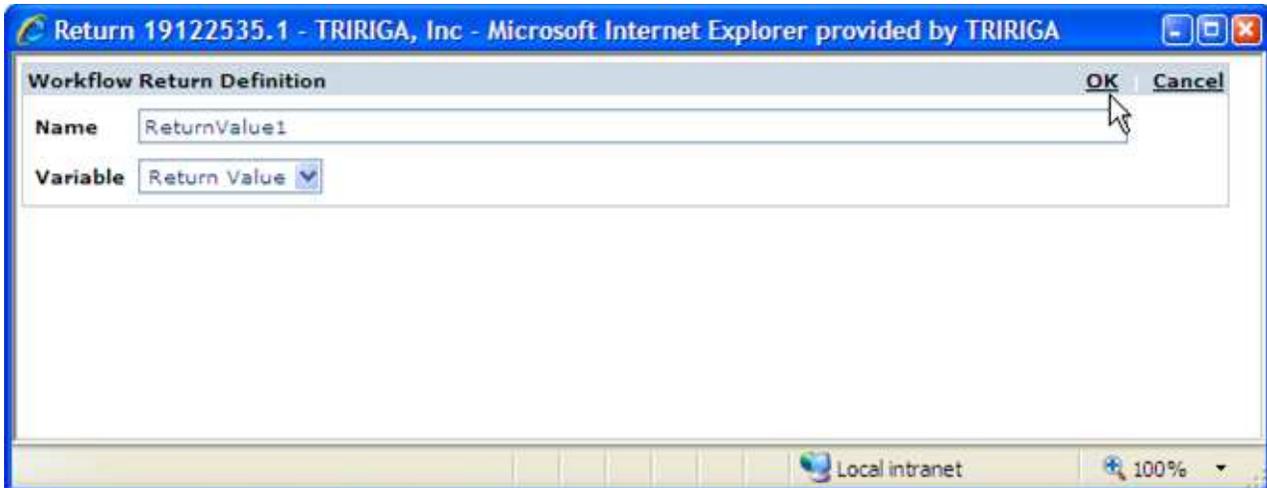
Figure 58. Adding a return value

Now the Workflow Parameters and Return Values screen looks like the following figure.
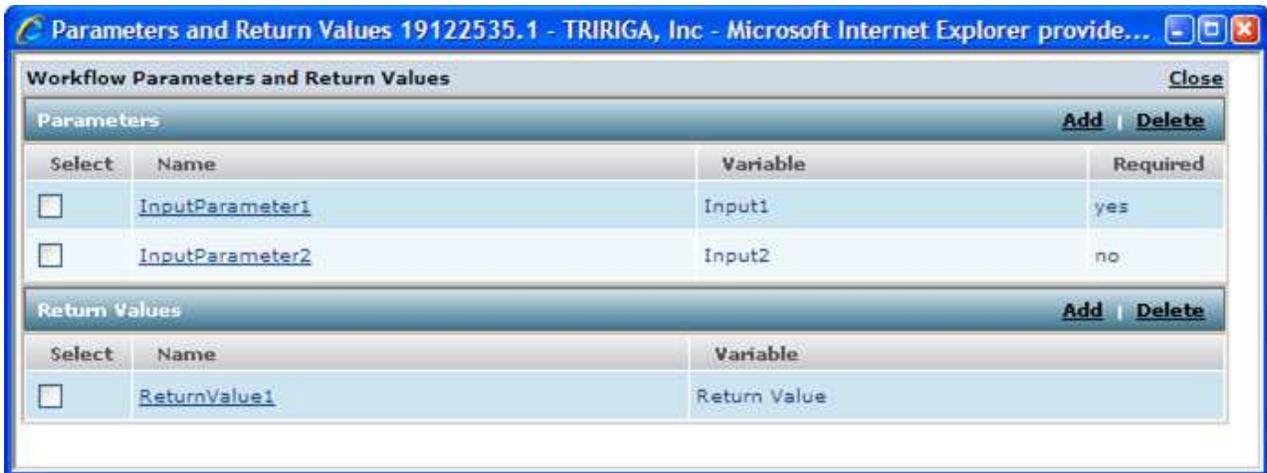
Figure 59. Example of workflow parameters and return values

To see how this example Start task is used with a calling workflow, see the Call Workflow task discussion in "Example: Workflow parameters and return values".

User action task

A User Action task creates an action item. The action item can be assigned to a specific person or to a group.

The User Action shape is in the task palette only if the workflow is asynchronous. This aspect of a workflow is specified by its Concurrence property which is described in "Start task".

The form for a User Action task is shown in the following figure. It is organized into five sections.

Figure 60. User action properties

Here are descriptions of the fields that appear in the first section:

Label

>This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

>This label is also what the person who gets the action item as a result of this task sees as a description of what to do in his Action Items portal section.

Description
>A description of this task goes in this field.

Lock Record For Other Users

If this check box is checked, the record identified by this task is locked for the assignee(s) identified by this task from the time that the action item is created until the time that the action item is completed. If any other user tries to access the record while it is locked, they will not be able to perform any actions on the record.

Note that a user lock does not prevent other workflows from modifying the same record. Also, users with administrator privileges will still be able to perform actions on the record.

Action

This field contains a drop-down list to select the specific action that will complete action items created by this workflow task. Action items will only be considered complete when the specified action is performed on the action item's record.

Use record locking carefully

It is important to lock records when more than one person at a time may be changing the contents of the same record. Otherwise, changing the contents of a record can produce confusion.

Be careful how you manage locked records, or the locks can produce their own sort of confusion. If a record is locked for use by a user who has gone on vacation, others may be inconvenienced because they cannot modify the record until the person on vacation comes back.

The usual way to get a record unstuck in this situation is for a user who is a member of the Admin Group to perform the action that will end the user action.

Multiple actions

The Action field allows you to specify a single action that will complete an action item. Occasionally you have a situation where you want to have two or more ways to complete an action item.

There is an indirect way of creating action items that can be completed with more than one action. The way to do it is make the action specified in the Action field a hidden action. For each action that you want to complete the action item, attach a workflow that performs the hidden action. The mechanism for performing an action on a record from a workflow is discussed in "Trigger action task".

Records section (user action)

The second section of the form for User Action task properties is labeled *Records*. The purpose of the *Records* section is to identify the record that will be in the action item.

There are fields at the top of the *Records* section that are used to identify a target record. The target record is used to determine the record that will be used for an action item created by this task.

There are radio buttons below these fields. The way that the target record is used to determine the record that will be the action item depends on which one of the radio buttons is selected.

Here are descriptions of the fields that appear at the top of the *Records* section:

Take the

This is a drop-down list that can have one of three possible values:

Business Object

If *Business Object* is selected, the record associated with the task specified by the field to the right of this one will be the target record.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the system event is the target record.

Assignee

If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record used by an action item.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of each radio button. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record is used for the action item.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target record will be used for the action item created by this task. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

If the record referenced by the smart section or locator field is a link, the record used for the action item will be the link, not the underlying record. Do not do this because link records do not have a form to display to the person receiving the action.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, a record associated with the target record will be used for the action item created by this task. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Records* section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

This is similar to the *Use its Association* radio button.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target record's parent will be the record to use for the action item.

When you select this radio button, a window pops up for you to select the business object to assume was used to create the parent record. This selection of a business object is not used for filtering.

The selection of a business object represents an assumption about what kind of record the parent will be. Because of this assumption, subsequent tasks will be able to access the parent record's fields. If the actual parent was not created from the assumed business object, the task may fail if the actual record does not have an expected field.

One of the choices for the business object that created the parent is *-Any-*. Choosing this is the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the *Records* section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be used as an action item. If the record can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

Assign to section

The third section of the form for user action task properties is labeled *Assign To*. The purpose of the *Assign To* section is to specify three things:

- The person or people who should receive action items created by this task. There is a group of radio buttons at the top of the section for this purpose.
- How long it should take for the action item to be completed. There are three fields in the middle of the section for this purpose.
- The start time to use when determining the actual time it has taken for the action to be completed. There is a group of radio buttons at the bottom of the section for this purpose.

Both groups of radio buttons in the *Assign To* section have a down-arrow button to the left of them. Initially only the currently selected radio button is visible in each group. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

Here are descriptions of the radio buttons at the top of the *Assign To* section that specify who should receive action items created by this task:

User

> If *User* is selected, action items created by this task will be assigned to a specified person.
>
> When this radio button is selected, a window pops up allowing you to select a person from among the people who have a user ID to sign into the IBM TRIRIGA Application Platform.

Group

> If *Group* is selected, then anyone in a specified group will be able to accept and complete the action item created by this task.
>
> When this radio button is selected, a window pops up allowing you to select a group.
>
> After someone in the group accepts the action item, the action item will appear only in that person's action item queue.

Assignee from task _____

> If this radio button is selected, then the action item created by this task will be assigned to the same user as was assigned to a previous task.
>
> The field to the right of the radio button has a drop-down list that you use to specify a task. The action item created by this task will be assigned to the same user or group that was the assignee of the specified previous task.
>
> The assignee of a workflow's start task is considered to be the currently logged in user.

People BO from task ____
> If *People BO* is selected, then the action item created by this task will be assigned to the user whose *My Profile* record is associated with a *triPeople* record associated with the specified task. The field to the right of the radio button has a drop-down list that you use to specify the task.

To the right of the label *Expected Time to Complete* are three fields that allow you to specify the expected amount of time that action items created by this task will take to complete. These three fields allow you to specify the expected amount of time in days, hours and minutes.

The group of radio buttons at the bottom of the *Assign To* section is used to specify the start time that will be used to compute the actual amount of time it has taken to complete an action item. There are two choices:

Estimate Start Date from System Date
> If this radio button is selected, the actual time to complete an action item is computed based on the system time when the action item was created. Usually you will choose this option.

Estimated Start Date from Field
> If this is selected, use the value of a specified Date or Date and Time field as the start time to compute actual time to complete. The field to the right of the radio button contains a drop-down list that allows you to specify the task associated with the record that contains the start time. When you select the radio button, a window pops up to allow you to select a field from the specified record.

Reminder section

The fourth section of the form for user action task properties is labeled *Reminder*. The purpose of the *Reminder* section is to specify whether the recipient of an action item should receive a reminder to complete the action item.

The field *Send Reminder to Assignee* has a drop-down list with these items in it:

Never If this is selected, no reminder will be sent.

Before Estimated Due Date
> If this is selected, a reminder will be sent a specified amount of time before the action item is due.

After Estimated Due Date
> If this is selected, a reminder will be sent a specified amount of time after the action item is due.

Under the *Send Reminder to Assignee* field is a field labeled *Expected Time to Complete*. This is *not what it sounds like*. It is the amount of time before or after the action item is due that a reminder should be sent if the action item is not completed.

Note section

Arbitrary notes about the task can be put here. They are copied into the action item record when the user action task runs.

Approval task

An Approval task is very similar to a User Action task. The difference is how the action item completes. The actions to complete the action item are always approve and reject. The Approval task understands acceptance and rejection, which it translates into success and failure.

For action items created by an Approval task to work properly, the target of the Approval task must include the approval state transitions. If it does not, records associated with the action items will not have accept or reject actions. It will be impossible to complete the action item.

The Approval shape is in the task palette only if the workflow is asynchronous. This aspect of a workflow is specified by its Concurrence property which is described in "Start task".

The form for an Approval task is very similar to the form for a User Action task (described in "User action task"). It differs from the User Action task form in only these ways:

- The heading says Approval Task Properties instead of *User Task Properties*.
- There is no *Lock Record for Other Users* check box. It is not needed because Approval tasks always lock the record to be approved.
- There is no *Action* field. It is not needed because the actions that complete an Approval task are always *Approve* and *Reject*.

In all other ways, the properties form for an Approval task is the same as for a User Action task and the same descriptions apply.

You may notice that in the IBM TRIRIGA 10 applications an approval engine has been implemented that expands the options to the user to include other actions than Approve and Reject. For more information on approvals, see the *IBM TRIRIGA 10 Application Administration User Guide*.

Create record task

The Create Record task creates new records. The properties form for a Create Record task is organized into three sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Module

The value of this property is the module that contains the business object this task will use to create new records. The value of this property appears to the right of the Module icon.

You set the value of this property by clicking the Module icon. Clicking the Module icon causes a list of modules to pop up. Click the module name you want to be the value of this property.

Object

This field contains a drop-down list of the business objects in the module specified in the *Module* field. Select the business object that this task will use to create new records.

Form The value of this property is the name of the form that will be used to edit or view records created by this task.

In Memory Only

Select this property if the record is needed for this workflow only. The record is not persisted, is not transitioned, cannot be used on either side of an association, and header fields (such as control number, name, cost, quantity, and path) are not updated as they do not exist.

In memory records can be passed to other workflows and can be used like other records, with the restrictions in the previous paragraph.

They can provide a significant performance improvement as no data is written to the database.

If an operation is attempted on an in memory record that is not supported, an UnsupportedOperationException exception is thrown to the server.log.

If you use an in memory smart object, you cannot suspend the workflow. When you publish a workflow that contains a Create Record task with the In Memory Only property selected, the platform displays a warning if this task is before a User Action task. The warning is because there is a suspend and resume within the User Action task while waiting for the user to respond. Also, the workflow cannot contain a Trigger Action task or an Attach Format File task.

By default, this property is not selected.

The default values for in memory record fields are supported for the following field types only: Date, Date and Time, Duration, Number, Text, and Time. The default values are not processed for system fields or any other field types. The default values that include formulas are processed with the available field values.

Note: In memory smart object (IMSO) records will not be found with either a Retrieve Records task or Query task. Since IMSO records appear in memory only, they do not reside in the database. Although IMSO records can be used in the *Right Side Data* section of the Task Filter window, IMSO records will not be returned in the result set from a Retrieve Records task or Query task.

Action

This task creates records in three steps. First, it creates an empty record. Next, it sets the value of the record's fields. Finally, it performs an action on the new record to change its state from *null* to something else.

When a record is first created it is in a special state called null. When a record's state is null, it is not stored in the database. Records in the null state disappear after the operation currently using the record is done. It is important for a task that creates a record to do something to take it out of the null state. Firing an action to take the record out of the null state is the way this is done.

The purpose of this field is to select the action this task will perform on new records after it has finished setting the values of their fields. Select an

action from the drop-down list in the *Action* field to specify the action that this task will perform on new records. Note that you need to make sure the proper values are set to create a unique name for this record or this task will fail. See "Naming records".

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed

Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Initialize from section

The second section of the form for a Create Record task's properties is labeled *Initialize From*. After a Create Record task has created an empty record, it sets the values of the new record's fields to their initial values. Setting the initial values of the fields happens in two steps.

First the fields are set to their default values. Then specified values are copied from a task or a record. The way the values are specified is described in "Object mapping".

The *Initialize From* section has two radio buttons to specify where data to copy to a new record will come from. These radio buttons are labeled:

Workflow Activity

> If this radio button is selected, it means that values will be copied from a preceding workflow task or a record associated with a preceding workflow task.

Existing Record

> If this radio button is selected, it means that values will be copied from a specified record that exists now, at the time you are editing this workflow.

The selection of one of these radio buttons determines what appears in the *Initialize From* section.

When the *Workflow Activity* radio button is selected, the *Initialize From* section has fields to select a workflow task from which values can be copied. The *Initialize From* section also has a field and radio buttons to identify a record that is associated with the specified workflow task. Values from this record may be copied to a newly created record.

Under the *Workflow Activity* radio button there are fields used to identify a target record that is used to determine the record from which values may be copied.

There are radio buttons below these fields. The way that the target record is used to determine the existing record depends on which one of the radio buttons is selected.

Here are descriptions of the fields used to determine the target record:

Take the

> This is a drop-down list that can have one of three possible values:

> **Business Object**

>> If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record. If a specified task has multiple records associated with it, then there are multiple target records. If there are multiple target records, there will be multiple existing records to copy values from. This task will create as many records as it has existing records to copy values from.

> **Secondary BO**

>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the system event is the target record.

> **Assignee**

>> If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine from which existing record values may be copied.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are descriptions of the radio buttons:

Use it If this is selected, the target record will be the existing record that values may be copied from.

Use its Reference

If this is selected, values may be copied from an existing record referenced by a smart section or locator field of the target record. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

If the record at the other end of the reference used by the smart section or locator field is a link, then values will be copied from the link and not its underlying record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, values may be coped from a record associated with the target record. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Initialize from* section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to

the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the values may be copied from the target record's parent.

When you select this radio button, a window pops up for you to select the business object that is assumed to have been used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

The selection of a business object represents an assumption about what kind of record the parent will be. Because of this assumption, subsequent tasks will be able to access the parent record's fields. If the actual parent was not created from the assumed business object, the task may fail if the actual record does not have an expected field.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the *Initialize from* section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that values may be copied from. If the record can have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

It is possible for there to be multiple records to copy values from. This happens if there are multiple target records or if a target record is associated with multiple records or references multiple records that fit the specification. This task will create one new record for each record it has to copy values from.

If you want to initialize records with information that is determined by the application's configuration, select the option of using an existing record in the *Intialize from* section. When *Existing Record* is selected, the *Intialize from* section looks like the one in the following figure.

Figure 61. Create Record properties (existing)

Specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record you want this task to use.

Transaction section (create record)

The third section of the Create Record task properties form is labeled *Transaction*. The purpose of the *Transaction* section is to specify the scope of the transaction being processed. The Transaction setting only applies to the creation of the object record prior to post-create and record transition processes. The Transaction setting is not applied to post-create and record transition of the newly created records, which means that pre-create and transition workflows called within a Create Record task will not be wrapped in the currently controlling transaction. The choices for Transaction are as follows:

None Indicates that the record is committed right after it is created. No transactions are created; the processing of the task follows standard workflow rules.

Per X Source Records
A new context is created for each X number of source records to be created from and committed when that number of source records is reached. The default value is 1, which means the records are committed after each creation

When a Create Record task is within a DataConnect task, the outermost DataConnect task is in control of the transaction for that Create Record task, regardless of setting in the Transaction section of the nested Create Record task. The Create Record task only can apply its Transaction setting if the controlling DataConnect task's Transaction setting is None.

At this point we have a newly created record to initialize with data and an existing record to copy data from. The next step is to specify what data is to be copied. This is done by clicking the *Edit Map* action on the *Initialize From* section bar. Clicking the *Edit Map* action opens the *Object Mapping* form so you can specify what values should be copied to the newly created record.

If you want to reset the object mapping to its defaults, click the *Reset Map* action.

Object mapping

The *Object Mapping* form is accessed from the properties of a Create Record task to specify how values will be copied to new records to initialize them. It is also accessed from the properties of a Modify Records task to specify how values will be copied to existing records to modify them.

In this explanation of the Object Mapping form, the record that values will be copied from is referred to as the source record.

The first section of an Object Mapping form is labeled with the name of the business object that is expected to have been used to create the record being modified or created. In the middle of the first section's top bar is the name of the business object used to create the source record.

In the first section is a drop-down field labeled *Association Type*. If an association is selected in this field, the selected type of association will be created between the record being initialized or modified and the source record. The drop-down list of Association Types is defined by the object-level associations defined for the object record being created in the Association Manager or Data Modeler.

The Association Type setting in the Object Mapping form is used to create a record-level association from the record being created to the record listed in the Initialize From section of the Create Record task.

If the Object Mapping form is being used to specify how to initialize new records, two check boxes appear under *Association Type*. These check boxes are not in the Object Mapping form when it is used to specify how to modify an existing record.

Use Source Project

> If this is checked, records created by this task will be part of the same project as the source record.

> If this is not checked, records created by this task will be part of whatever project is the current project. For a top-level record, its project will be that of the user's project context. For a new child record, its project will be that of its parent record. For an existing record that is made to be the child of another record, its project will be that of its parent record.

Include Child Records

> If the source record is part of a hierarchy module, copies of all of the source record's children will be made. The copies will be children of the newly created record.

> The copy of child records includes all children in the hierarchy, the values of their fields, any embedded records, and any associations. Note that even though the associations are copied, the associated records are not copied.

The Object Mapping form always has a section labeled *General*. The section labeled *General* is different from the other sections because it contains a list of fields that are not in any smart section.

The rest of the sections in the Object Mapping form are labeled with the names of smart sections in the record to be initialized or modified. In each of these sections is a list of the fields in the named smart section.

To the right of each field name and section name is text that describes what the mapping operation does to the field. When there is no text to the right of the field name it means that the mapping operation does nothing to the field.

The area to the right of the field name may contain text like:

- *General::PeopleFillName*

This names the section and field in the source record. If a string appears to the right of the field name, the new value for the field is copied from the specified field. You do not type in this text yourself. Instead, the text is put there as a result of a selection you made elsewhere.

If the text is in a text field, it means that the text is constant text that will be used as-is. This means that the mapping operation will always set the field to that value. To enter constant text into a text box, select the check box to the right. When the check box is selected, a box appears and you can type text into it.

Workflow constant mapping for a locator field is case sensitive.

Text mapping should not be used to populate locator fields. Instead, use a mechanism to identify the record to be mapped into a locator via the task's Map From or Initialize From section and use the Source mapping option. It may require an extra Retrieve Records task or Query task to identify the record, but doing so ensures that the current record is being mapped into the locator.

A circumstance where text mapping can be used safely is when the value being mapped is either the full publish name for non-hierarchical objects, or the full hierarchy path for hierarchical objects. Only do so if the locator field has those respective fields set in the locator field's business object.

There is a special constant you can specify that means to copy the current time into a field. If you map a field to the constant $$CURRENTTIME$$, the mapping operation will copy the current time and date to the field. If mapping $$CURRENTTIME$$ to a text field, it maps a formatted date time value. If mapping to a number field, it maps a millisecond value.

To set a duration to a constant value, express the duration as a number of milliseconds. A duration of one hour is 3600000 milliseconds. A duration of one week is 604800000 milliseconds. A duration of one day is 86400000 milliseconds. Because a duration of one day is relatively common, there is a special way to represent a duration of one day that is easier to remember and easier to read. Instead of entering 86400000, you can enter $$DAY$$.

If you click the search icon to the right of the check box, a small window pops up. This window contains radio buttons for the fields and sections of the source record.

This small window is called an attribute picker. If you want the mapping operation to copy data from a field in the source record, click the attribute picker icon for the field you want the data copied to and select the radio button for the source field. The procedure is similar if you want to copy from a smart section in the source.

There are two radio buttons in an attribute picker that do not correspond to a field or a section. These radio buttons are labeled *None* and *Source*.

If you select the radio button labeled *None*, it means that the mapping operation will not do anything to the affected field. In the Object Mapping form, the text to the right of the field's name will be blank.

If you select the radio button labeled *Source*, it means that you want to use the entire source record. This is useful, for example, if you want to point a smart section or locator field at a particular record. When populating a locator field or smart section via the workflow Object Mapping form, always choose the Source option from the attribute picker. This ensures that a valid reference gets created to the record being placed in the locator field or smart section.

If you use the *Source* radio button to put a record in a smart section, what actually happens depends on the kind of record that is the source for the task.
- If the source is a stand alone record, then a reference is created from the smart section to the source record.
- If the source is an embedded or link record, then a copy is made of the source record and a reference is created from the smart section to the copy.

There is another source of data for a mapping operation to copy. Some tasks perform computations. The results of these computations are associated with the task that performed them rather than with a record.

If you want a mapping operation to copy the results of one of these computations to a field, click the task data picker icon (it looks like binoculars). The task data picker shows each of the tasks that precede the current task.

Sometimes you want a mapping operation to use a formula to compute a value. You can use an extended formula as the source of data for a field. Extended formulas are described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

To specify an extended formula as the source of data for a field, click the formula picker icon. Clicking the formula picker icon causes an Extended Formula window to pop up.

This Extended Formula window differs in three ways from the one discussed in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.
- First, extended formulas are available for fields of any data type in the Workflow Builder; they are not limited to numbers and dates.
- Second, the extended formula does not have an *Outputs* section. An extended formula used in object mapping always has exactly one output, which is the field it is being used to provide data for.
- Third, extended formulas can access fields in the source record of the object map only; fields from associated objects cannot be accessed.

With these exceptions, specifying an extended formula for use in an object mapping operation is exactly the same as using an extended formula anywhere else.

Note that when the name of a field is the same in the source record and the destination record and the destination field has formula mapping, the target field value is not mapped. Instead, it is calculated from its Data Modeler formula mapping. If a formula field is grayed out, it means that the formula map defined in the Data Modeler is used as the source value for this field.

The Object Mapping form also allows makes it possible for you to map associations from the source object to the target object. The Association Mapping section of the object map displays a list of the associations defined for the target business object. When you click the search icon to the right of one of the associations, the association picker contains radio buttons for the association definitions of the source record.

You will notice that the association definition of the target is used to filter which associations you see from the source to those of the same business object. If you want the mapping operation to copy associations from the source record, click the search icon for the association definition you want copied to and select the radio button for the source association definition.

At runtime if an association is found from the source to the selected business object with the selected string, a new association will be made from the target using the definition selected. If the target association is flagged as dependent, a copy of the record associated to the source will be made and associated to the target.

To clear the mapping for a field, section, or association click the clear data icon.

Modify records task

The Modify Records task modifies values in existing records. The properties form for a Modify Records task is organized into four sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed
Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities. This is the default value.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option provides performance optimization.

An example of the use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Map to records section

The second section of the properties form for a modify records task is labeled *Map To Records*. The purpose of this section is to identify the record(s) that the task will modify.

At the top of the *Map To Records* section are fields used to identify a target record. The target record is used to determine the record that will be modified.

There are radio buttons below these fields. The way that the target record is used to determine the record that will be modified depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

This is a drop-down list that can have one of three possible values:

Business Object

If *Business Object* is selected, the record associated with the task specified by the field to the right of this one will be the target record. If multiple records are associated with the task, there will be multiple target records.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or

SCHEVENTEND system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the system event is the target record.

Assignee

If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that will be modified.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record that is modified.

Use its Reference

If this is selected, records referenced by a smart section or locator field of the target record will be modified. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will be modified. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Map To Records* section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module

that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If -*Any*- appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click -*Any*- which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the target record's parent will be modified.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

One of the choices for the business object that created the parent is -*Any*-. Choosing this is the equivalent of selecting the module's base business object (the one with the same name as the module).

Under the radio buttons is a check box labeled *Only if association is*. This can be useful if there may be more than one record to be modified. Checking this check box enables filtering of records to be modified. The filtering is based on associations that these records may have with the other records. If the check box is checked and there is a name to the right of the Association icon, only records having an association with the specified name will be modified. If the check box is checked and there is nothing to the right of the Association icon, only records that have an association of any sort will be modified. For the purpose of this check box, it does not matter what kind of record is at the other end of the association.

To specify what appears to the right of the Association icon, click the Association icon. A list of association names pops up. Click the association name that you want to appear to the right of the Association icon. If you want nothing to appear after the Association icon, click -*Any*- which appears at the top of the list.

At the bottom of the *Map To Records* section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be modified. If the record can have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Map from records section

The third section of the properties form for a modify records task is labeled *Map From Records*. The purpose of this section is to identify the record that values will be copied from (source).

The *Map From Records* section has two radio buttons to specify where data to copy to the record being modified will come from. These radio buttons are labeled:

Workflow Activity
Copy values from a preceding workflow task or a record associated with a preceding workflow task.

Existing Record
Copy values from a specified record that exists now.

The selection of one of these radio buttons determines what appears in the *Map From Records* section.

When the *Workflow Activity* radio button is selected, the *Map From Records* section has all the fields that are in the *Map To Records* section, except for the *Only if association is* check box. The difference between the fields in these sections is that the fields in the *Map To Records* section are used to select records to modify and the fields in the *Map From Records* section are used to select records from which to copy values.

If you want to modify records with information that is determined by the application's configuration, you should select the option of using an existing record in the *Map From Records* section. When *Existing Record* is selected, the *Map From Records* section looks like the one in the following figure.

Figure 62. Modify Records (existing)

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record you want this task to use.

Transaction section (modify records)

The fourth section of the Modify Records task properties form is labeled *Transaction*. The purpose of the *Transaction* section is to specify the scope of the transaction being processed. The choices for Transaction are as follows:

None   Indicates that the record is committed right after it is modified. No transactions are created; the processing of the task follows standard workflow rules.

Per X Source Records
A new context is created for each X number of records to be modified and committed when that number of records is reached. The default value is 1, which means the records are committed after each modification.

When a Modify Records task is within a DataConnect task, the outermost DataConnect task is in control of the transaction for that Modify Records task, regardless of setting in the Transaction section of the nested Modify Records task. The Modify Records task only can apply its Transaction setting if the controlling DataConnect task's Transaction setting is None.

At this point, you are ready to specify how the records specified in the *Map To Records* section will be modified by data copied from records specified in the *Map From Records* section. You specify this by using the Object Mapping form.

Access the Object Mapping form by clicking the *Edit Map* action on the *Map To Records* section bar. The Object Mapping form is described in "Object mapping".

If you want to reset the object mapping to its defaults without popping up the Object Mapping form, click the *Reset Map* action.

Retrieve records task

A Retrieve Records task retrieves a list of records, filters out unwanted records, and performs a computation on the filtered records. The result of the computation can be used by another task.

The Retrieve Records and the Query workflow tasks are similar in that the essential purpose of both workflow tasks is to find records. They differ in the way that they find records. The Retrieve Records task finds record through their association with a record already associated with a workflow task. A Query workflow task does not use associations. Instead it runs a query and uses the records returned by the query.

Note: In memory smart object (IMSO) records will not be found with either a Retrieve Records task or Query task. Since IMSO records appear in memory only, they do not reside in the database. Although IMSO records can be used in the *Right Side Data* section of the Task Filter window, IMSO records will not be returned in the result set from a Retrieve Records task or Query task.

The properties form for a Retrieve Records task is organized into five sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Retrieve section

The second section of the properties form for a Retrieve Records task is labeled *Retrieve*. The purpose of this section is to specify the computation that is to be performed on the records that are retrieved. The computation is determined by which one of the radio buttons in this section is selected.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button they appear next to.

Here are the descriptions:

A List If this option is selected, the task will take the retrieved and filtered records and make them the result list that is associated with this task.

____ **Record(s) with the Greatest value In CHOOSE FIELD**

If this option is selected, this task will take the specified number of retrieved and filtered records that contain the greatest values in the specified field and make them the result list associated with this task. Note there is no unit of measure considerations, it is a straight comparison.

Records with a null value will be returned last. You can add a static filter in the Filter Using section if you want to eliminate null returns. See information about the Filter Using section in "Filter using section (retrieve records)".

When you click this radio button, a window pops up for you to specify which field this task should use. After you have specified which field to use, the name of the field appears as the text of the hyperlink in place of the previous text. Clicking the hyperlink will pop up the window again to you can change the specified field.

The result list can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

____ **Record(s) with the Least value In CHOOSE FIELD**

If this option is selected, this task will take the specified number of retrieved and filtered records that contain the least values in the specified field and make them the result list associated with this task. Note there is no unit of measure considerations when taking this sum, it is a straight comparison.

Records with a null value will be returned first. You can add a static filter in the Filter Using section if you want to eliminate null returns. See information about the Filter Using section in "Filter using section (retrieve records)".

When you click this radio button, a window pops up for you to specify which field this task should use. After you have specified which field to use, the name of the field appears as the text of the hyperlink in place of the previous text. Clicking the hyperlink will pop up the window again to you can change the specified field.

The result list can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

The Sum of CHOOSE FIELD

If this option is selected, this task will take the retrieved and filtered records, add the numbers in the specified field and make the sum the result sum that is associated with this task. Note there is no unit of measure considerations when taking this sum, it is a straight addition.

When you click this radio button, a window pops up for you to specify which field this task should use. After you have specified which field to use, the name of the field appears as the text of the hyperlink in place of the previous text. Clicking the hyperlink will pop up the window again to you can change the specified field.

The result sum can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

Financial Data from Token CHOOSE TOKEN

If this option is selected, this task will filter the retrieved financial records so that only those that have a Financial Rollup field that uses the specified financial token will be included in the result list associated with this task.

When you click this radio button, a window pops up for you to specify which token this task should use. After you have specified which token to use, the name of the token appears as the text of the hyperlink in place of the previous text. Clicking the hyperlink will pop up the window again to you can change the specified token.

The result list can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

The description of Financial Rollup fields and financial tokens is in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

All Descendants

If the retrieved records are created from a business object that is part of a hierarchy module and this option is selected, then this task takes the descendants of the retrieved and filtered records and makes them the result list associated with this task.

If the *Including Self* check box to the right of this radio button is checked, then the result list will include the retrieved and filtered records along with their descendants.

The result list can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

All Ancestors

If the retrieved records are created from a business object that is part of a hierarchy module and this option is selected, then this task takes the retrieved and filtered records along with their ancestors and makes them the result list associated with this task.

If the *Including Self* check box to the right of this radio button is checked, then the result list will include the retrieved and filtered records along with their ancestors.

The result list can be used by subsequent tasks to initialize or modify records. This is described in "Object mapping".

From records section

The purpose of the *From Records* section is to specify the records that this task will retrieve.

At the top of the *From Records* section are fields used to identify target records. The target records are used to determine the records that will be retrieved.

There are radio buttons below the fields. The way that the target records are used to determine the records that will be retrieved depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

This is a drop-down list that can have one of three possible values:

Business Object

> If *Business Object* is selected, then records associated with the task specified by the field to the right of this one will be the target records.

Secondary BO

> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the records at the other end of the association are the target records.

> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee

> If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

> The value of this field is the label of the task that the target records will be associated with.

The radio buttons under these fields determine how the target records will be used to determine the records that will be retrieved.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target records will be the records that are retrieved.

Use its Reference

> If this is selected, records referenced by a smart section or locator field of the target record will be retrieved. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

> If this is selected, records associated with the target record will be retrieved. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Map To Records* section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target records are created from a business object that is part of a hierarchy module and this option is selected, the target records' parent will be retrieved.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the *Map To Records* section is a read-only field labeled *Object Type*. The value displayed in this field is the type of records that will be retrieved. If the records can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

Filter using section (retrieve records)

The purpose of the *Filter Using* section is to specify filter conditions that will be used to filter the retrieved records. A filter condition is something about a retrieved record that can be either true or false. This task's computation will only use

retrieved records for which all the filter conditions are true. If any filter conditions are false for a retrieved record, the retrieved record is discarded and not used.

To add a filter condition to the *Filter Using* section, click the *Add Filter* action on the *Filter Using* section bar. Once you have added filter conditions to the *Filter Using* section, you will see a check box for each filter condition. To remove a filter condition from the *Filter Using* section, check the filter condition's check box and click the *Delete Filter* action on the section bar.

When you click the *Add Filter* action on the *Filter Using* section bar, a Task Filter window pops up so you can specify the details of a filter condition. When you click the link in the filter condition's *Left Side Data* field a Filter Window pops up to allow you to modify the details of an existing filter condition.

A Task Filter window has two sections. The purpose of the *Left Side Data* section is to specify how retrieved records will be used in the filter condition. The purpose of the *Right Side Data* section is to specify what the retrieved records will be compared to.

When the *List* check box in the *Left Side Data* section is not checked, it means that the value of a field in each retrieved record will be compared to one other value. The field in the retrieved records to use for the comparison is specified by the *Section Name* and *Field Name* fields.

The comparison to use is specified by the value of the *Operator* field. The *Operator* field is a drop-down list that allows you to choose from the following:
* Equals
* Not Equals
* Less Than
* Less Than or Equals
* More Than
* More Than or Equals
* Contains
* Contains - Case Sensitive
* Start With
* Start With - Case Sensitive
* End With
* End With - Case Sensitive
* Before
* After
* In
* Not In
* Does Not Contain
* Does Not Contain - Case Sensitive

At the end of this section, we will explain what each of these comparisons are for.

The value to compare the *Left Side Data* to is specified by the *Right Side Data* section. The three radio buttons at the top of the *Right Side Data* section determine which of three places the value will come from. Fields underneath the three radio buttons provide additional details about the value.

If the *Source* radio button is selected, the other value will come from a record specified by the *Filter Record* section of the Retrieve Records task properties. It will use the field in the record specified by the *Section Name* and *Field Name* fields of the *Right Side Data* section. If the *Target* radio button is selected, then the other value for the comparison will come from the field of the retrieved record specified by the *Section Name* and *Field Name* fields of the *Right Side Data* section.

If the source or target is a list of records, one of the records in the list will be chosen to do the comparison. Instead, ensure the source or target for filtering is a single record when the List property is not checked.

If the *Constant* radio button is selected, then the *Section Name* and *Field Name* fields of the *Right Side Data* section are not visible. Instead, there is another field labeled *Value* where you can type the actual other value to be used for the comparison. If you are checking to see if the field is empty, compare it to a constant value of Null.

When the *List* check box in the *Left Side Data* section is checked it means that the set of retrieved records will be compared to the set of records specified by the *Filter Using* section of the Retrieve Records task's properties form. Since you are comparing lists of records, both the From Records and the Filter records should be created from the same module.

The comparison to use is specified by the value of the Operator field. The Operator field is a drop-down list that allows you to choose from the following values: In or Not In.

Check the *List* check box in the *Right Side Data* section.

We conclude this description of the Task Filter window by describing what the comparison operators are for:

Equals, Not Equals

> These are useful for comparing individual values. If both values are the same, then Equals is true. If the values are different then Not Equals is true.

> To test for the existence of a date value, use *Equals* or *Not Equals* for the operator and Null for the value; do not use *More Than* or *Less Than* zero to test for the existence of a date value.

Less Than, Less Than or Equals, More Than, More Than or Equals

> These are used to compare individual number values. *Less Than* is true if the value specified by the *Left Side Data* section is less than the value specified by the *Right Side Data* section. The rest of these comparisons work in a similar way.

> To test for the existence of a date value, use *Equals* or *Not Equals* for the operator and Null for the value; do not use *More Than* or *Less Than* zero to test for the existence of a date value.

Contains, Start With, End With, Does Not Contain

> These are used to compare individual text values. *Contains* is true if the text value specified by the *Left Side Data* section contains the text value specified by the *Right Side Data* section. For example, *abcdefg* contains *bcd*; *abcdefg* does not contain *bug*.

Start With is true if the text value specified by the *Left Side Data* section starts with the text value specified by the *Right Side Data* section. For example *abcdefg* starts with *abcd*; *abcdefg* does not start with *xyz*.

End With is true if the text value specified by the *Left Side Data* section ends with the text value specified by the *Right Side Data* section. For example *abcdefg* ends with *fg*; *abcdefg* does not end with *abc*.

Does Not Contain is true if the text value specified by the *Left Side Data* section does not contain the text value specified by the *Right Side Data* section. For example, *abcdefg* contains *bcd*; *abcdefg* does not contain *bug*.

Contains, *Start With*, *End With*, and *Does Not Contain* are not case sensitive. *Contains - Case Sensitive*, *Start With - Case Sensitive*, *End With - Case Sensitive*, and *Does Not Contain - Case Sensitive* are case sensitive.

Before, After

These are used to compare individual date, time and date and time values. *Before* is true if the value specified by the *Left Side Data* section is before the value specified by the *Right Side Data* section. *After* is true if the value specified by the *Left Side Data* section is after the value specified by the *Right Side Data* section.

In, Not In

These are used to compare sets of records. These are the only comparisons that should be used if the *List* check box is checked. If the *List* check box is not checked then one of the other comparison operations should be used.

In is true for retrieved records that are in the set of records specified by the *Filter Records* section of the Retrieve Records task's properties form. *Not In* is true for retrieved records that are not in the set of records specified by the *Filter Records* section.

After you click OK on the Task Filter, the platform displays the task filter in the Filter Using section.

To remove a task filter from the list in the Filter Using section, select the check box next to a filter and click Delete Filter.

Filter records section (retrieve records)

The purpose of the *Filter Records* section is to identify the records that this task will use to filter retrieved records.

The *Filter Records* section has two radio buttons to specify where records for filtering will come from. These radio buttons are labeled:

Workflow Activity

If this radio button is selected then the records for filtering will be associated with a previously performed workflow task.

Existing Record

If this radio button is selected, then the record used for filtering will be a specified record that exists now.

The selection of one of these radio buttons determines what appears in the *Filter Records* section.

When the *Workflow Activity* radio button is selected, the *Filter Records* section has all the fields that are in the *From Records* section. The difference between the fields

in these sections is that the fields in the *Filter Records* section are used to select the records to use for filter conditions that have their *Source* radio button selected.

If you want to filter records with information that is determined by the application's configuration, then you should select the option of using an existing record in the *Filter Records* section. When the *Existing Record* radio button is selected, the *Filter Records* section looks like the following figure.

Figure 63. Retrieve Records Filter Records section (existing filter)

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record you want this task to use.

Query task

A Query task retrieves the list of records found by a query. The result list can be used by another task.

The Query and the Retrieve Records workflow tasks are similar in that the essential purpose of both workflow tasks is to find records. They differ in the way that they find records. A Query workflow task runs a query and uses the records returned by the query. The Retrieve Records task does not use queries. Instead it finds records through their association with a record already associated with a workflow task.

Note: In memory smart object (IMSO) records will not be found with either a Retrieve Records task or Query task. Since IMSO records appear in memory only, they do not reside in the database. Although IMSO records can be used in the *Right Side Data* section of the Task Filter window, IMSO records will not be returned in the result set from a Retrieve Records task or Query task.

When a query is set to Active Project and run from the Report Manager, the results depend on whether or not the record is in a capital project. If the record is in a capital project, the results are based on the scope of the project the user is in. If the record is not in a capital project, the results are based on the scope of the project the parent record is in. When the same query is run in a workflow, the results are based on all projects, including the Company Level. To ensure that a query run in a workflow returns the same results as that query returns when run from the Report Manager, use a Set Project task at the start of the workflow.

The properties form for a Query task is organized into four sections. Here are descriptions of the fields in the first section:

Label This is the caption used to identify this task. The text in this field will be used as the caption for the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Query, Query Object Type

Clicking the *Query* link causes a window to pop up that allows you to select the query that this task will run. The name of the selected query appears in a read-only field to the right of the *Query* link. The object type of the selected query appears in a read-only field to the right of the *Query Object Type* link.

The window that pops up to select the query looks like the following figure. Select the module of the query you want from the drop-down list to the right of the *Module* label.

Figure 64. Select Item(s) panel showing queries

If what you are looking for is not a query, but some other kind of report, you will not see what you are looking for because the list initially includes only queries. You will need to select the type of report you want to see it.

To select the type of report you want to see, click the filter icon to the right of the *Reports List* label. A menu will pop up that looks like the following figure. From this menu, you can choose the type of report you want to see a list of or you can choose the *All* menu item to see a list that contains every kind of report.
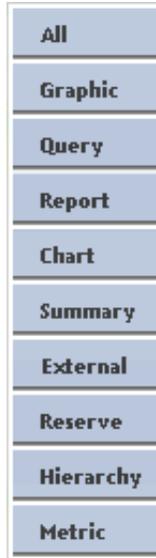
Figure 65. Report Type menu

Query Security Enabled

>To enable the security group settings of the logged-in user on records retrieved by the Query task, select the *Query Security Enabled* check box.

Some queries have filter values that can be specified interactively ($$RUNTIME$$ filters). A Query task does not have any way to provide a value for interactive filters. However, a Query task has other filtering mechanisms that it can use. The purpose of the other sections of a Query task's properties are to control the filtering of query results.

Context records section

A query can have association filters or field filters specified to filter out records in context of a particular record. This context is based on the relationship to the record and is specified to filter records that are not associated in a specified way or do not have the required values in particular fields. These are often used in query sections or actions that the user interacts with based on the record the user is currently viewing. In a workflow, the way to specify the records that are used for this type of filter is called context records. Association filters and field filters are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

At the top of the *Context Records* section are fields used to identify target records. The target records are used to determine the records that will be used as context records.

There are radio buttons below the fields. The way that the target records are used to determine the records that will be used as context records depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

>This is a drop-down list that can have one of three possible values:

Business Object

If *Business Object* is selected, records associated with the task specified by the field to the right of this one will be the target records.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the records at the other end of the association are the target records.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target records will be associated with.

The radio buttons under these fields determine how the target records will be used to determine the records that will be the context records.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it   If this is selected, the target records will be the context records.

Use its Reference

If this is selected, records referenced by a smart section or locator field of the target record will be the context records. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will be the context records. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Map To Records* section in the *Context Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target records are created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the context record.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

The Query task does not actually care about the fields of the context record. The selection of a business object is just for consistency with the way this mechanism works for other kinds of workflow tasks.

At the bottom of the *Map To Records* section is a read-only field labeled *Context Object Type*. The value displayed in this field is the type of records that will be the context record. If the records can have been created from any business object in a particular module, then the name of the module appears in the *Context Object Type* field.

Filter using section (query)

The purpose of the *Filter Using* section is to specify filter conditions that will be used to filter the query results. A filter condition is something about a record in the query results that can be either true or false. This task will only use records in the query results for which all the filter conditions are true. If any filter conditions are false for a query result record, the record is discarded and not used.

To add a filter condition to the *Filter Using* section, click the *Add Filter* action on the *Filter Using* section bar. Once you have added filter conditions to the *Filter Using* section, you will see a check box for each filter condition. To remove a filter condition from the *Filter Using* section, check the filter condition's check box and click the *Delete Filter* action on the section bar.

When you click the *Add Filter* action on the *Filter Using* section bar, a Task Filter window pops up so you can specify the details of a filter condition. When you click the link in the filter condition's *Left Side Data* field, a Filter Window pops up to allow you to modify the details of an existing filter condition.

A Task Filter window has two sections. The purpose of the *Left Side Data* section is to specify how query result records will be used in the filter condition. The purpose of the *Right Side Data* section is to specify what the query result records will be compared to.

When the *List* check box in the *Left Side Data* section is not checked, it means that the value of a field in each query result record will be compared to one other value. The field in the query result records to use for the comparison is specified by the *Section Name* and *Field Name* fields.

The comparison to use is specified by the value of the *Operator* field. The *Operator* field is a drop-down list that allows you to choose from the following:
* Equals
* Not Equals
* Less Than
* Less Than or Equals
* More Than
* More Than or Equals
* Contains
* Contains - Case Sensitive
* Start With
* Start With - Case Sensitive
* End With
* End With - Case Sensitive
* Before
* After
* In
* Not In
* Does Not Contain
* Does Not Contain - Case Sensitive

At the end of this section, we will explain what each of these comparisons are for.

The value to compare the *Left Side Data* to is specified by the *Right Side Data* section. The three radio buttons at the top of the *Right Side Data* section determine which of three places the value will come from. Fields underneath the three radio buttons provide additional details about the value.

If the *Source* radio button is selected, then the other value will come from a record specified by the *Filter Records* section of the Query task properties. It will use the field in the record specified by the *Section Name* and *Field Name* fields of the *Right Side Data* section.

If the *Target* radio button is selected, then the other value for the comparison will come from the field of the query result record specified by the *Section Name* and *Field Name* fields of the *Right Side Data* section.

If the source or target is a list of records, one of the records in the list will be chosen to do the comparison. Instead, ensure the source or target for filtering is a single record when the List property is not checked.

If the *Constant* radio button is selected, then the *Section Name* and *Field Name* fields of the *Right Side Data* section are not visible. Instead, there is another field labeled *Value* where you can type the actual other value to be used for the comparison. If you are checking to see if the field is empty, compare it to a constant value of Null.

When the *List* check box in the *Left Side Data* section is checked it means that the set of query result records will be compared to the set of records specified by the *Filter Using* section of the Query task's properties form. Since you are comparing lists of records, both the From Records and the Filter records should be created from the same module.

The comparison to use is specified by the value of the Operator field. The Operator field is a drop-down list that allows you to choose from the following values: In, Not In, or Associated To.

Check the *List* check box in the *Right Side Data* section.

We conclude this description of the Task Filter window by describing what the comparison operators are for:

Equals, Not Equals

> These are useful for comparing individual values. If both values are the same, then *Equals* is true. If the values are different then *Not Equals* is true.

> To test for the existence of a date value, use *Equals* or *Not Equals* for the operator and Null for the value; do not use *More Than* or *Less Than* zero to test for the existence of a date value.

Less Than, Less Than or Equals, More Than, More Than or Equals

> These are used to compare individual number values. *Less Than* is true if the value specified by the *Left Side Data* section is less than the value specified by the *Right Side Data* section. The rest of these comparisons work in a similar way.

> To test for the existence of a date value, use *Equals* or *Not Equals* for the operator and Null for the value; do not use *More Than* or *Less Than* zero to test for the existence of a date value.

Contains, Start With, End With, Does Not Contain

These are used to compare individual text values. *Contains* is true if the text value specified by the *Left Side Data* section contains the text value specified by the *Right Side Data* section. For example, *abcdefg* contains *bcd*; *abcdefg* does not contain *bug*.

Start With is true if the text value specified by the *Left Side Data* section starts with the text value specified by the *Right Side Data* section. For example *abcdefg* starts with *abcd*; *abcdefg* does not start with *xyz*.

End With is true if the text value specified by the *Left Side Data* section ends with the text value specified by the *Right Side Data* section. For example *abcdefg* ends with *fg*; *abcdefg* does not end with *abc*.

Does Not Contain is true if the text value specified by the *Left Side Data* section does not contain the text value specified by the *Right Side Data* section. For example, *abcdefg* contains *bcd*; *abcdefg* does not contain *bug*.

Contains, *Start With*, *End With*, and *Does Not Contain* are not case sensitive. *Contains - Case Sensitive*, *Start With - Case Sensitive*, *End With - Case Sensitive*, and *Does Not Contain - Case Sensitive* are case sensitive.

Before, After

These are used to compare individual date, time and date and time values. *Before* is true if the value specified by the *Left Side Data* section is before the value specified by the *Right Side Data* section. *After* is true if the value specified by the *Left Side Data* section is after the value specified by the *Right Side Data* section.

In, Not In

These are used to compare sets of records. These are the only comparisons that should be used if the *List* check box is checked. If the *List* check box is not checked then one of the other comparison operations should be used.

In is true for retrieved records that are in the set of record specified by the *Filter Records* section of the Query task's properties form. *Not In* is true for retrieved records that are not in the set of records specified by the *Filter Records* section.

Associated To

This comparison is available only if the *List* check box is checked.

When Associated To is the Operator in the *Left Side Data* section of a Task Filter, three properties show in the *Right Side Data* section: Record From Task, With Association Name, and Or Its Children.

In Record From Task, specify from which task to retrieve the associated record. If the task does not return a record, the platform skips this filter.

In Association Name, specify the association to use. -Any- is one of the values available.

Selecting the Or Its Children check box causes the association filter to also check the associated record's children.

After you click OK on the Task Filter, the platform displays the task filter in the Filter Using section.

To remove a task filter from the list in the Filter Using section, select the check box next to a filter and click Delete Filter.

Filter records section (query)

The purpose of the *Filter Records* section is to identify the records that this task will use to filter query result records.

The *Filter Records* section has two radio buttons to specify where records for filtering will come from. These radio buttons are labeled:

Workflow Activity
> If this radio button is selected then the records for filtering will be associated with a previously performed workflow task.

Existing Record
> If this radio button is selected, then the record used for filtering will be a specified record that exists now.

The selection of one of these radio buttons determines what appears in the *Filter Records* section.

When the *Workflow Activity* radio button is selected, the *Filter Records* section has all the fields that are in the *Context Records* section. The difference between the fields in these sections is that the fields in the *Filter Records* section are used to select the records to use for filter conditions that have their *Source* radio button selected.

If you want to filter records with information that is determined by the application's configuration, then you should select the option of using an existing record in the *Filter Records* section.

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record you want this task to use.

Associate records task

An Associate Records task can create an association between two records or remove an association between two records. Also, if defined in the Data Modeler, the reverse association is created or removed.

The properties form for an Associate Records task is organized into three sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
> A description of this task goes in this field.

The radio buttons under these fields determine whether this task will create or remove an association.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons:

Associate using _____

If the *Associate using* radio button is selected, an association having the specified name is created from the records identified by the *Create the association from* section to the record identified by the *To* section, unless such an association already exists.

Remove the association_____

If the *Remove the association* radio button is selected, an association having the specified name is removed from the records identified by the *Remove the association from* section to the record identified by the *Where the associated record is* section. The form for an Associate Records task when Remove the association is selected is shown in the following figure.

Figure 66. Associate Records properties for Remove the association

To the right of both radio buttons is an Association icon. The Association icon is used to specify the name of the association to be created or removed. The name of the specified association appears to the right of the Association icon.

To change the specified association name, click the Association icon. A list of association names appears. Click the association name you want to be the specified association name.

Formulas

Defines whether or not the changes made by this task could be put in the Extended Formula Queue for further consideration. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed

Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query tokens, association tokens, or both are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query tokens, association tokens, or both can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor or space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

The next two sections identify the records to use.

Create the association from / Remove the association from

The second section of the Associate Records task properties form is labeled *Create the association from* or *Remove the association from*, depending on whether the Associate using or the Remove the association radio button is selected. The purpose of this second section is to identify the first record in the association.

For the sake of brevity, we refer to the record in this section as the first record and refer to the record in the next section as the second record. You can think of this as Create an association from the first record to the second record, or Remove the association from the first record to the second record.

To / Where the associated record is

The third section of the Associate Records task properties form is labeled *To* or *Where the associated record is*, depending on whether the Associate

using or the Remove the association radio button is selected. The purpose of the third section is to identify the second record in the association.

This section also provides the option of using an existing record.

Options for record selection

The options for record selection in these two sections are the same except that the *To / Where the associated record is* section provides an option to select an existing record, which is discussed in "Existing record identification".

The *To / Where the associated record is* section has two radio buttons to specify where the second record will come from. These radio buttons are labeled:

Workflow Activity
> If this radio button is selected, the second record will come either directly or indirectly from a previously performed workflow task.

Existing Record
> If this radio button is selected, the second record will be a specified record that currently exists. More information is in "Existing record identification".

The following description of the fields for record selection apply to the Create the association from / Remove the association from section and the To / Where the associated record is section when its Workflow Activity ratio button is selected:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, a record associated with the workflow task specified by the field to the right of this one will be used to select the record for this section.

> **Secondary BO**

>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the secondary record in the Associate/De-Associate will be used to select the record for this section.

>> This option also is available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART,* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event will be used to select the record for this section.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be used to select the record for this section.

of Task
> The value of this field is the label of the workflow task to retrieve the record from.

The radio buttons under these fields determine how the record identified will be used to determine the record used for this section.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio

button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the record identified will be used for this section.

Use its Reference

>If this is selected, a record referenced by a smart section or locator field of the record identified will be used for this section. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the record identified in this section.

>After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

>If this is selected, a record associated with the record identified will be used for this section. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be in the second record and optionally the name of the association.

>After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

>This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

>When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the second record. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

>You also may specify that the second record must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the second record must have been created from the named business object. If *-Any-* appears in the drop-down list, the second record may have been created from any business object in the named module.

>To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-,* which appears at the top of the list.

Use its Parent

> If the record identified is created from a business object that is part of a hierarchy module and this option is selected, the record's parent will be used for this section.
>
> When this radio button is selected, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.
>
> One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be the second record. If the record can have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Existing record identification

The *To / Where the associated record is* section provides the option of using an existing record.

If the Existing Record radio button is selected, the second record will be a specified record that currently exists. When the *Existing Record* radio button is selected, the third section looks like the one in the following figure.

Figure 67. Associate Records (existing)

Specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want for the second record. Then click the *Record* link to find the specific record you want this task to use.

Trigger action task

A Trigger Action task performs a specified action on specified records.

If there are multiple records for a Trigger Action task to perform an action on, the action is performed on all of the records, one at a time. If the actions launch asynchronous workflows, depending on how the IBM TRIRIGA Application Platform and its environment are configured, some or all of the launched workflows may run at the same time.

The properties form for a Trigger Action task is organized into three sections. Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Action
This field is a drop-down list of the actions that can be performed on the kind of records specified in the *Records* section of this task's properties form. The action specified by this field is the action that this task will perform on its records.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed
Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Records section (trigger action)

The second section of the Trigger Action task properties form is labeled *Records*. The purpose of the *Records* section is to identify the records this task will perform its action on.

At the top of the *Records* section are fields used to identify a target record. The target record is used to determine the records that the action will be performed on.

There are radio buttons below the fields. The way that the target records are used to determine the records an action will be performed on depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, a record associated with the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the records on which to perform the action.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio

button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record on which the action is performed.

Use its Reference

> If this is selected, a record referenced by a smart section or locator field of the target record will be the record on which the action is performed. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

> If this is selected, a record associated with the target record will be the record on which the action is performed. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

> After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

> This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

> When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

> You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

> To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager

pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the target record's parent will be the record on which the action is performed.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It allows this task to know what action will be available and also allows other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the records on which the action will be performed. If the record can have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Trigger when section

The value selected in the Trigger When section specifies when a time-based action is performed.

Immediately

For each record returned from the Records section, the specified state action is performed on the record and the record is transitioned. The act of performing the state action also posts the corresponding event to the WF_EVENT table, where a workflow agent processes it and runs any asynchronous workflows registered for the combination of the business object and event.

Later

For each record returned from the Records section, performance of the specified state action is delayed until the specified time.

At runtime, the platform stores the action information in the WF_FUTURE_EVENT table. The workflow future agent picks it up at the specified time, posts it to the current WF_EVENT table, and tags it as an action. A workflow agent processes it as described above under "Immediately", and the record is transitioned.

After an action is posted to be performed in the future, it is no longer tied to the originating workflow. The originating workflow does not wait for a future action to complete. If the future action fails when it runs, the errors do not reflect in the results of the original workflow. Errors during the execution of future actions are written to the appropriate server log file.

Clear Existing Future Actions

When this property is selected, at runtime similar actions that exist in the WF_FUTURE_EVENT table based on the action name and record Id combinations are deleted before new future actions for the specified records are inserted.

From Task Result field and Date Time field

Together, the values in the From Task Result property and the Date Time field property determine when in the future the specified state action occurs. The From Task Result drop-down list shows tasks that are prior to this task in the workflow. The Date Time field only shows the Date and Time fields of the business object defined in the task identified in the From Task Result property.

Clear Existing Future Actions

For each record returned from the Records section, no state action is performed. Existing future trigger actions are removed from the WF_FUTURE_EVENT table.

Delete reference task

A Delete Reference task removes references to records from a smart section.

The properties form for a Delete Reference task is organized into three sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Delete Reference From Section

This field is a drop-down list of smart section names. The value of this field is the name of the smart section from which to remove a reference.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed

Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Delete reference from record section

The second section of the properties form for a Delete Reference task is labeled *Delete Reference From Record*. The purpose of this section is to specify the record that contains the smart section from which the reference to a record will be removed.

At the top of the *Delete Reference From Record* section are fields used to identify a target record. The target record is used to determine the record that contains the smart section from which a reference to a record will be removed.

There are radio buttons below the fields. The way that the target record is used to determine the record that contains the smart section depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Record**
>> If *Record* is selected, the record associated with the task specified by the field to the right of this one will be the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate action. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that contains the reference to be removed.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio

button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record that contains the smart section.

Use its Reference

If this is selected, records referenced by a smart section or locator field of the target record will contain the smart section. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will contain the smart section. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of

the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the target record's parent will contain the smart section.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields and smart sections.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will contain the smart section. If the record can have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

To record section

The third section of the properties form for a Delete Reference task is labeled *To Record*. The purpose of this section is to identify the record that is referred to by the reference that will be removed from the smart section.

The *To Record* section has two radio buttons to specify how to find the referenced record. These radio buttons are labeled:

Workflow Activity
The referenced record will be associated with a preceding workflow task.

Existing Record
The referenced record will be a record that exists now.

The selection of one of these radio buttons determines what appears in the *To Record* section. The following figure shows what the *To Record* section looks like when the *Workflow Activity* radio button is selected.

When the *Workflow Activity* radio button is selected, the *To Record* section has all the fields that are in the *Delete Reference From Record* section. The difference between the fields in these sections is that the fields in the *To Record* section are used to select the record to remove and the fields in the *Delete References From Record* section are used to select the record that contains the smart section.

If you want to remove a reference to record that is part of the application's configuration information, you should select the option of using an existing record in the *To Record* section. When *Existing Record* is selected, the *To Record* section looks like the following figure.

Figure 68. Delete Reference properties (existing)

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to remove a reference to. Then click the *Record* link to find the specific record you want this task to use.

Add child task

An Add Child task adds a child record to a record.

This will work only if the child records are allowed to be children of the candidate parent record. For this to be the case, the parent and child records must have been created from business objects that are part of the same hierarchy module. An *Is Parent of* association definition must exist from the business object used to create the parent record to the business object used to create the child records.

The properties form for an Add Child task is organized into three sections. Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed

Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Parent record section

The second section of the Add Child task properties form is labeled *Parent Record*. The purpose of the *Parent Record* section is to identify the record that will be the parent.

The *Parent Record* section has two radio buttons to specify where the primary record will come from. These radio buttons are labeled:

Workflow Activity

If this radio button is selected, the parent record will be associated with a previously performed workflow task.

Existing Record

If this radio button is selected, then the parent record will be a specified record that exists now.

The selection of one of these radio buttons determines what appears in the *Parent Record* section.

When the *Workflow Activity* radio button is selected, below the radio buttons, the *Parent Record* section has all the fields that are in the *Child Record* section. The difference between the fields in these sections is that the fields in the *Parent Record* section are used to select the parent record and the fields in the *Child Record* section are used to select child records. Because of these similarities, the description of these fields is part of the description of the *Child Record* section.

If you want the parent record to be part of the application's configuration, you should select the option of using an existing record. When you select the *Existing Record* radio button, the second section looks like the one in the following figure.

Figure 69. Add Child properties (existing)

Specify the value for the *Module* and *Object* properties so that this task knows what kind of record will be the parent. Then click the *Record* link to find the specific record you want this task to use.

Child record section

The third section of the association task properties form is labeled *Child Record*. The purpose of the *Child Record* section is to identify child records for this task.

The following description of the fields in the third section also applies to the fields in the second section that appear in the *Parent Record* section when its *Workflow Activity* radio button is selected. The fields in the *Parent Record* section serve a purpose similar to the corresponding fields in the *Child Record* section. The difference between the corresponding fields in the *Parent Record* section is that they are used to identify the parent record for this task and the fields in the *Child Record* section are used to identify child records for this task.

At the top of the *Child Record* section are fields used to identify target records. The target records are used to determine the child records.

There are radio buttons below the fields. The way that target records are used to determine the child records depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

This is a drop-down list that can have one of three possible values:

Business Object

If *Business Object* is selected, then records associated with the task specified by the field to the right of this one will be the target records.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the records at the other end of the association are the target records.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target records will be associated with.

The radio buttons under these fields determine how the target records will be used to determine the child records.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target records will be the child records.

Use its Reference

If this is selected, records referenced by a smart section or locator field of the target records will be the child records. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target records.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target records will be the child records. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target records are created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the child records.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the records that will be child records. If the records can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

Set project task

A Set Project task has two functions. It (1) sets the current project for the following tasks in this execution of this workflow, or (2) changes the project on a record to the specified project. Projects are described in "Projects".

The form for a Set Project task depends on selections made in the first section.

The properties form for a Set Project task is organized into one, two, or three sections, depending upon selections made in the first section. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Under these fields are four radio buttons:

Set Workflow Project to Company Level
If this radio button is selected, this task will cause the Company Level project to be the current project. The Company Level project is the top level project and includes all other projects.

Set Workflow Project from Record
If this radio button is selected, the current project will be set to the project associated with the record specified in the *Project Record* section of this form. If the record specified in the Project Record section is from the triProject module, the current project will be set to the actual record found.

Change Project for Records to Company Level

If this radio button is selected, the project on the records specified in the To Records section of this form is changed to the Company Level.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to Y, the children are put in the Company Level, and the children's children, and so on.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to N, the project for the children is not changed. Only the record's project is affected.

For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Change Project for Records from Record

If this radio button is selected, the project on the records specified in the To Records section of this form will be set to the project associated with the records specified in the From Record section of this form. If more than one record is pulled from the From Record section at runtime, the system uses the first record pulled.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to Y, the children are put in the Company Level, and the children's children, and so on.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to N, the project for the children is not changed. Only the record's project is affected.
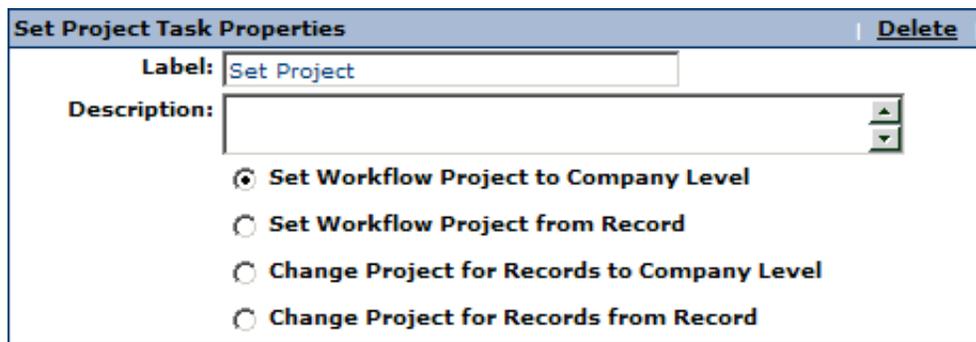
For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Set project to company level

Initially, the current project for a workflow is the project that was the current project for the user that launched the workflow. If the workflow was launched from a Call Workflow task, the initial project is the project context at the time the Call Workflow task ran.

When a query is set to Active Project and run from the Report Manager, the results are based on the scope of the project the user is in if the record is in a capital project or on the scope of the project the parent record is in if the record is not in a capital project. If the same query runs in a workflow, the results are based on all projects including the Company Level. To resolve this discrepancy, use a Set Project task at the start of the workflow. This ensures the workflow returns the same results as the query returns when run from the Report Manager.

Selecting the Set Workflow Project to Company Level radio button changes the form to a Set Project task to look like the following figure.

Figure 70. Set Project properties - Set Workflow Project to Company Level

With this setting, a Set Project task sets the current project for the following tasks in this execution of the workflow to Company Level.

The properties form for a Set Project task when Set Workflow Project to Company Level is selected is organized into one section.

Set project from record

Initially, the current project for a workflow is the project that was the current project for the user that launched the workflow. If the workflow was launched from a Call Workflow task, the initial project is the project context at the time the Call Workflow task ran.

When a query is set to Active Project and run from the Report Manager, the results are based on the scope of the project the user is in if the record is in a capital project or on the scope of the project the parent record is in if the record is not in a capital project. If the same query run in a workflow, the results are based on all projects including the Company Level. To resolve this discrepancy, use a Set Project task at the start of the workflow. This ensures the workflow returns the same results as the query returns when run from the Report Manager.

With this setting, a Set Project task sets the current project for the following tasks in this execution of the workflow to the project associated with the record specified in the Project Record section of the form.

The properties form for a Set Project task when Set Workflow Project from Record is selected is organized into two sections.

Project record section

The second section of the properties form for the Set Project task when Set Workflow Project from Record is selected is labeled *Project Record*. The purpose of the *Project Record* section is to identify the record whose associated project will become the current project or to select a record from the triProject module.

At the top of the *Project Record* section are fields used to identify a target record. The target record is used to determine the record that will be used to set the current project.

There are radio buttons below the fields. The way that the target record is used to determine the record that will be used to set the project depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, a record associated with the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, a record at the other end of the association is the target record.

>> This option also is available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART,* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that will be used to set the current project.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio

button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record used to set the current project.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target records will be the record used to set the current project. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, a record associated with the target record will be used to set the current project. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of

the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-*, which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the target record's parent will be used to set the current project.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of record that will be used to set the project. If the record could have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Set project for records to company level

If this radio button is selected, the project on the records specified in the To Records section of this form is changed to the Company Level.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to Y, the children are put in the Company Level, and the children's children, and so on.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to N, the project for the children is not changed. Only the record's project is affected.

For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Selecting the Change Project for Records to Company Level radio button changes the form to a Set Project task to look like the following figure.

Figure 71. Set Project properties - Change Project for Records to Company Level

The properties form for a Set Project task when Change Project for Records to Company Level is selected is organized into two sections. The second section, To Records, identifies the target records to be changed to the Company Level.

To records section

The second section of the properties form for the Set Project task when Change Project for Records to Company Level is selected is labeled *To Records*. The purpose of the *To Records* section is to identify the records whose associated project will become the Company Level.

At the top of the *To Records* section are fields used to identify target records whose associated project will be changed to the Company Level.

There are radio buttons below the fields. The way that the target records are determined depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, records associated with the task specified by the field to the right of this one will be the target records.

> **Secondary BO**

>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, records at the other end of the association are the target records.

This option also is available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART,* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target records will be determined.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target records will have their project changed to Company Level.

Use its Reference

If this is selected, records referenced by a smart section or locator field of the target records will be the records that have their project changed to Company Level. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target records.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will have their project changed to Company Level. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful

when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You also may specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If -Any- appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click -Any-, which appears at the top of the list.

Use its Parent

If the target records are created from a business object that is part of a hierarchy module and this option is selected, the target records' parent will have its project set to the Company Level.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is -Any-. Choosing this is the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of record that will have its project changed to Company Level. If the records could have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Set project for records from record

If this radio button is selected, the project on the records specified in the To Records section of this form will be set to the project associated with the records specified in the From Record section of this form. If more than one record is pulled from the From Record section at runtime, the system uses the first record pulled.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to Y, the children are put in the Company Level, and the children's children, and so on.

If an affected record has children and the RECORD_PROJECT_CONTAINMENT property in TRIRIGAWEB.properties is set to N, the project for the children is not changed. Only the record's project is affected.

Selecting the Change Project for Records from Record radio button changes the
form to a Set Project task to look like the following figure.

Figure 72. Set Project properties - Change Project for Records from Record

The properties form for a Set Project task when Change Project for Records from
Record is selected is organized into three sections. The second section, To Records,
identifies the target records to be changed. The third section, From Record,
identifies the source record that contains the project to which the target will be
changed.

To records section

The second section of the properties form for the Set Project task when Change
Project for Records from Record is selected is labeled *To Records*. The purpose of
the *To Record* s section is to identify the records whose associated project will be
changed.

At the top of the *To Records* section are fields used to identify target records. The
target records will have their project changed.

There are radio buttons below the fields. The way that the target records are determined depends on which radio button is selected.

Here are descriptions of the fields:

Take the

> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, records associated with the task specified by the field to the right of this one will be the target records.

> **Secondary BO**

>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, records at the other end of the association are the target records.

>> This option also is available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART*, or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target records will be determined.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target records will have their project changed.

Use its Reference

> If this is selected, records referenced by a smart section or locator field of the target records will be the records that have their project changed. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target records.

> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will have their project changed. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You also may specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target records are created from a business object that is part of a hierarchy module and this option is selected, the target records' parent will have its project changed.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of record that will have its project changed. If the records could have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

From record section

The third section of the properties form for the Set Project task when Change Project for Records from Record is selected is labeled *From Record*. The purpose of the *From Record* section is to identify the record whose associated project is the source for the project of the record identified in the To Record section or to select a record from the triProject module.

If more than one record is pulled from the From Record section at runtime, the system uses the first record pulled.

At the top of the *From Record* section are fields used to identify a target record. The target record is used to determine the record that will be used to set the project in the record identified in the To Record section.

There are radio buttons below the fields. The way that the target record is used to determine the record that will be used to set the project depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, a record associated with the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, a record at the other end of the association is the target record.

>> This option also is available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART,* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that will be used to change the project on the records identified in the To Records section.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record used to set the project in the records in the To Records section.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target record will be the record used to set the project in the records in the To Records section. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, a record associated with the target record will be used to set the project in the records in the To Records section. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of

the Association icon. To retrieve association records that are not restricted to a particular association name, click -*Any*-, which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, the target record's parent will be used to set the project in the records in the To Records section.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is -*Any*-. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of record that will be used to set the project. If the record could have been created from any business object in a particular module, the name of the module appears in the *Object Type* field.

Schedule task

Schedule tasks allow a mapping operation to be scheduled once or as something to be done on a recurring basis. Schedule tasks are described in "Calendar and time-based events".

Instead of designing applications to create scheduled events manually, use the Event form, as described in "Calendar and time-based events".

Modify metadata task

The purpose of the Modify Metadata task is to change the metadata that controls the presentation of a form for a particular record.

Using the Modify Metadata task, you can do things like make sections or fields hidden or visible, or change the color of text in a field or the text of a label.

For each form that can be used to edit a record, a set of metadata is kept with the record. This means that if there is more than one form for a record, you can use a Modify Metadata task to make a field appear in red text for one form and the same field appear in blue text for another form. If two people are using the same form to look at the same record, while they are seeing the same data, they may not be seeing the same metadata. If the Session only field is selected, the application designer may have changed the metadata.

The properties form for a Modify Metadata task is organized into two sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Form

The value of this field determines which form(s) will be affected by this task. This field's value is selected from its drop-down list.

The drop-down list includes the names of all forms associated with the business object shown in the *Object Type* field at the bottom of the Modify Metadata task properties. If you select one of these, the Modify Metadata task will alter just the named form for working with the record specified in the second section.

You may need to fill in the second section of this form before setting the value of this field, otherwise the drop-down list for this field may not contain the right form names.

In addition to the names of forms, there are two other items that appear in this field's drop-down list.

If the value you select for this field is *-Current-*, this task will modify the metadata for whichever form is currently being used with the record identified by the second section.

If the value you select for this field is *-All-*, the modifications this task makes will be for the metadata of all forms associated with the business object for the record identified by this task's second section of properties. This is not often used since there is always a single form associated with a record. Instead of using *-All-*, use the *-Current-* option, which has better performance if there are several forms associated to the business object.

Reset Before Modification

If this check box is checked, before this task makes its modifications to metadata, the form's metadata will be reset to the values specified for the form in the Form Builder.

Session only

When selected, changes made in the metadata are only in effect for displaying the current record to the user and are not permanent. When not selected, modifications to metadata are permanent and remain in effect until the next time the metadata is modified.

When Session only is selected, the Reset action only affects the session modified metadata. When Session only is not selected, the Reset action only affects the permanent modified metadata and not the session modified metadata.

You cannot modify the currently displayed form by using an action that calls a workflow containing a Modify Metadata task with the Session only check box selected.

Records section (modify metadata)

The second section of the Modify Metadata task properties form is labeled *Records*. The purpose of the *Records* section is to specify the record associated with the form whose metadata is to be modified.

At the top of the *Records* section are fields used to identify a target record. The target record is used to determine the record associated with the form(s) to be modified.

There are radio buttons below the fields. The way that target records are used to determine the record associated with the form(s) depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
>   This is a drop-down list that can have one of three possible values:

>   **Business Object**
>   >   If *Business Object* is selected, record associated with the task specified by the field to the right of this one will be the target record.

>   **Secondary BO**
>   >   This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

>   >   This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

>   **Assignee**
>   >   If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
>   The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record associated with the form.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record associated with the form(s).

Use its Reference

>   If this is selected, a record referenced by a smart section or locator field of the target records will be the record associated with the form(s). When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will be the record associated with the form(s). When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the record associated with the form.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be associated with the form(s).

You can access the Form Mapping window by clicking the *Edit Map* action at the top of the *Records* section.

If you want to reset the object mapping to its defaults without popping up the Object Mapping form, click the *Reset Map* action.

There are special attributes that can be set at the root level of the Form Map. The Change GUI To attribute allows you to change the form to any form defined for the business object of the record. Change GUI To is ignored in session. The Close Window attribute overrides the Close option of the state transitions for the form defined in "State-based actions" for this particular runtime instance of the workflow. This property is reset to that of the form state transition each time the record is opened. Typically, this attribute is used for the "fail" path of a validation workflow to ensure the form is not closed if the workflow did not save the record permanently.

All other attributes are described in "Form building". The navigation of the Form Mapping tree is also the same as in the Layout tab of the Form Wizard. When you click one of the nodes in this tree you will see attributes that correspond to your selection, they will be different if you select a tab, section, field or action.

You will notice that there are three ways to populate the map. You can either type directly into the Value column, select from a list in the Value column or for some attributes check the box in the Task column. When you check the box in the task column a binoculars icon is displayed. When you click the binoculars icon, another window is displayed which shows a list of all tasks in the workflow, you can select the result of one of these tasks to populate the attribute.

Note: If the task selected has a list of records, one of the records in the list will be chosen to do the map into the metadata. Instead, use a task for mapping that results in a single record.

Attention: Be sure to click the *Apply* action on the left side of the Form Mapping after you have finished editing the map. If you move on to something else without first clicking that *Apply* action, then any changes you made to the map will be lost.

End task

An End task is used to indicate the end of a workflow. When a workflow reaches an End task, the workflow is complete and there are no more tasks to be performed. The workflow is considered to have completed normally.

All workflows are created with an End task. The End task that a workflow is created with cannot be deleted. End tasks that are added to a workflow after it is created can be deleted.

End tasks do not have any properties.

Stop task

A Stop task is used to indicate the end of a workflow. When a workflow reaches a Stop task, no more of its tasks will be performed. The workflow is considered to have completed abnormally.

Stop tasks do not have any properties.

Switch task

A Switch task contains two sequences of tasks. Based on conditions specified as properties of a Switch task, it causes only one of its task sequences to be performed. After the task sequence has been performed, the task after the Switch task is performed. The Switch task is much like the traditional "If statement."

The following figure shows a simple example of a workflow that contains a Switch task.
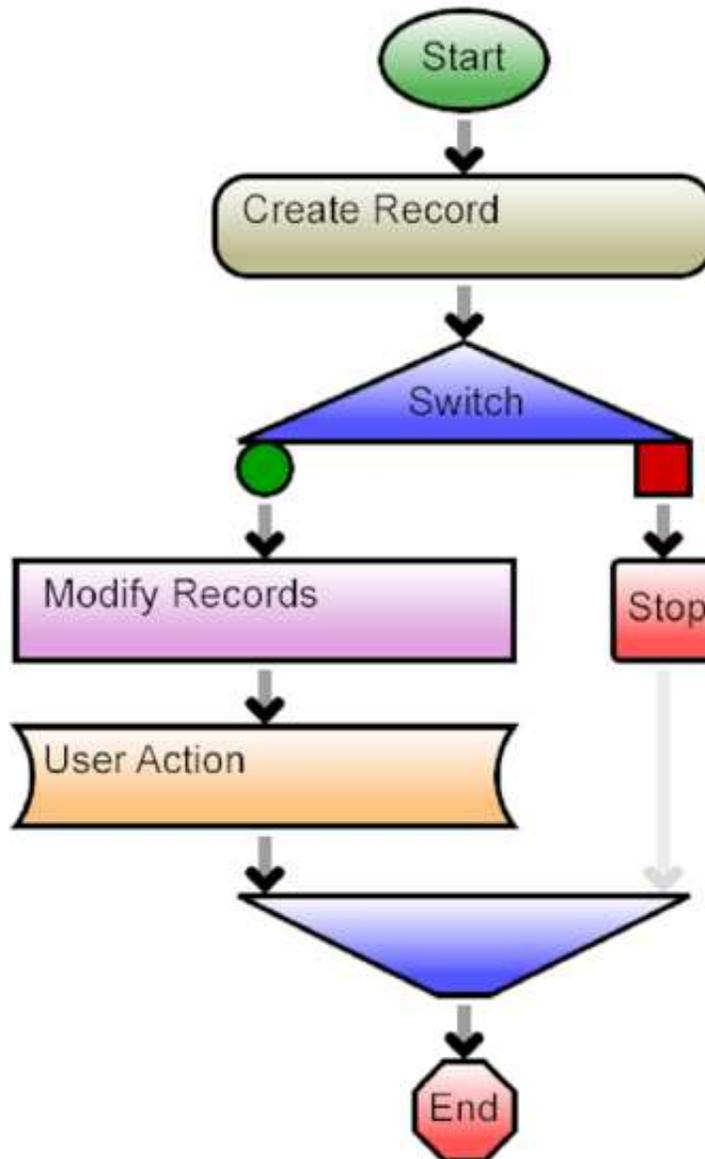
Figure 73. Switch example

The sample workflow in the figure begins by creating a record. If the create task is successful, the record is modified and is then used for an action item created by the User Action task. If the create task is not successful, the workflow ends in failure.

One thing to notice about the switch task as it appears in the figure is that there is more of it than just its palette shape. The bottom of the switch task is where the two alternate task sequences come together. You can click the top or the bottom of a switch task to see its properties.

The properties form for a switch task is shown in the following figure. The properties form for a switch task has more actions than we have seen in previous task forms. Like other task property forms, it has a *Delete* action to delete the task from the workflow. Be particularly careful when deleting a switch task from a workflow, because deleting a switch task also deletes the task sequences contained

in the switch task.
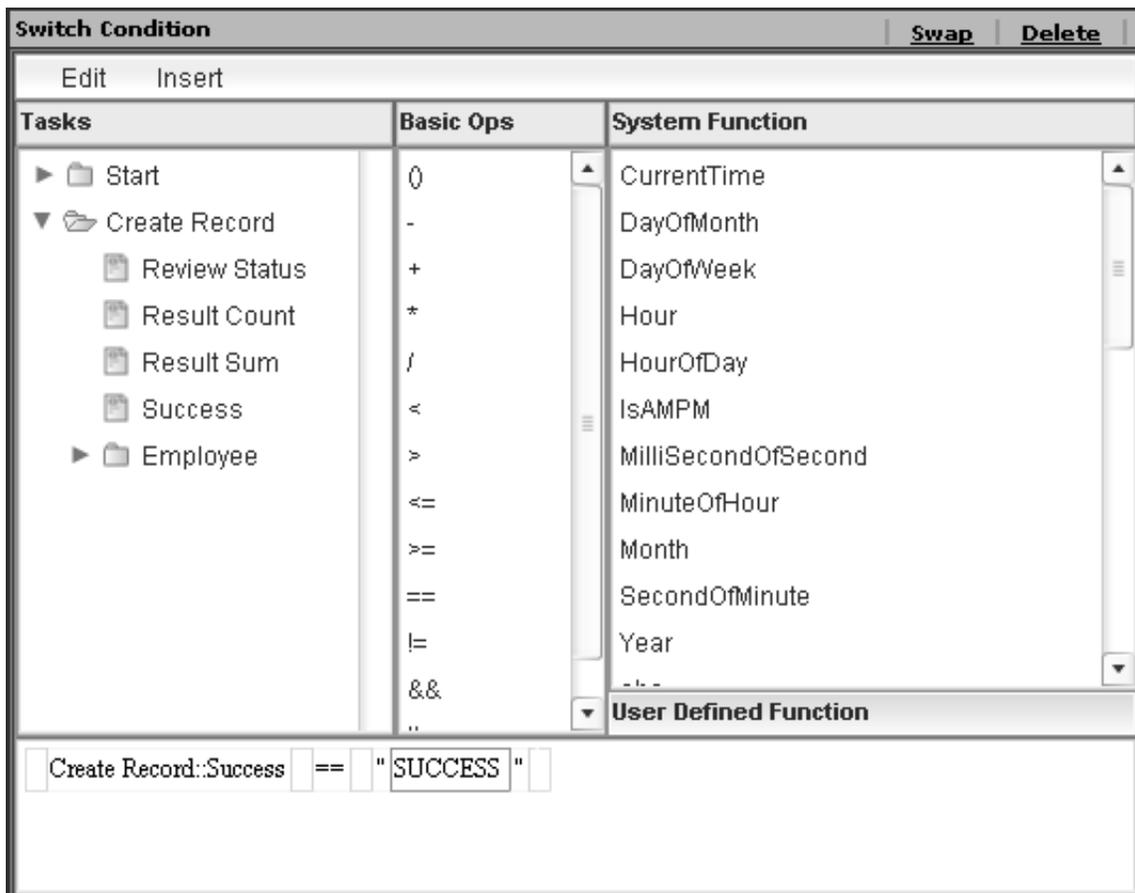
Figure 74. Switch properties

The contents of the Switch task form is a condition that is either true or false. If the condition is true when a switch task is performed, the task sequence attached to the green circle is performed. If the condition is false, the task sequence attached to the red square is performed.

The *Swap* action swaps the positions of the green circle and the red square, while leaving the task sequences where they are. After a *Swap* action, the task sequence that was attached to the green circle is attached to the red square and the task sequence that was attached to the red square is attached to the green circle.

Example: Workflow condition builder

Some types of tasks use a condition to decide what they should do. A condition is a comparison between values. A condition is either true or false. The properties forms for these tasks share a common organization to create and edit conditions.

The Condition Builder's purpose is to build a formula in its bottom panel that is either true or false. You build the formula using the *Edit* and *Insert* menus at the top of the condition builder, along with the *Tasks*, *Basic Ops* and *System Function* panels.

The *Tasks* panel allows you to insert task attributes and fields of records into the condition.

The *Basic Ops* panel allows you to insert operators into the condition.

The *System Function* panel allows you to insert system defined and user defined functions into a condition.

The following example shows the use of these panels and menus. After this example are more detailed explanations.

The example we will work through is for a workflow to be launched for an *Employee* record. The example will be to build a condition that is true if a preceding Create workflow task succeeded and if the current time is on or after the date that the employee was hired. When we are all finished, the condition will look like:

```
Create Record::Success=="SUCCESS" &&
   Start::Employee::Detail::DateHire<=CurrentTime()
```

When we first begin building a new condition, the bottom panel of the Condition builder looks like the following figure.

Figure 75. Empty condition

In the beginning there is no formula. There is just a small gray rectangle in the bottom panel. The Condition Builder puts a gray rectangle before each part of the formula.

When you add a part to the formula, it gets added to the right of the currently selected gray rectangle. You can select a gray rectangle by clicking it. You can tell that a gray rectangle has been selected because it will have a blinking cursor inside of it.

The gray rectangle in an empty Condition Builder is already selected. The first thing we will add to the condition is a reference to the Create Record task's success attribute. To do this, we use the *Tasks* panel.

We begin by clicking the folder icon next to the name of the *Create Record* task. Clicking the icon causes the attributes and business objects associated with the task to be displayed under the task. The folder icon changes to an open-folder icon.

The attributes now displayed under the *Create Record* task include an attribute named *Success*. Clicking the name *Success* causes the name of the attribute along with the name of the task to be copied to the condition. The Condition Builder now looks like the following figure.
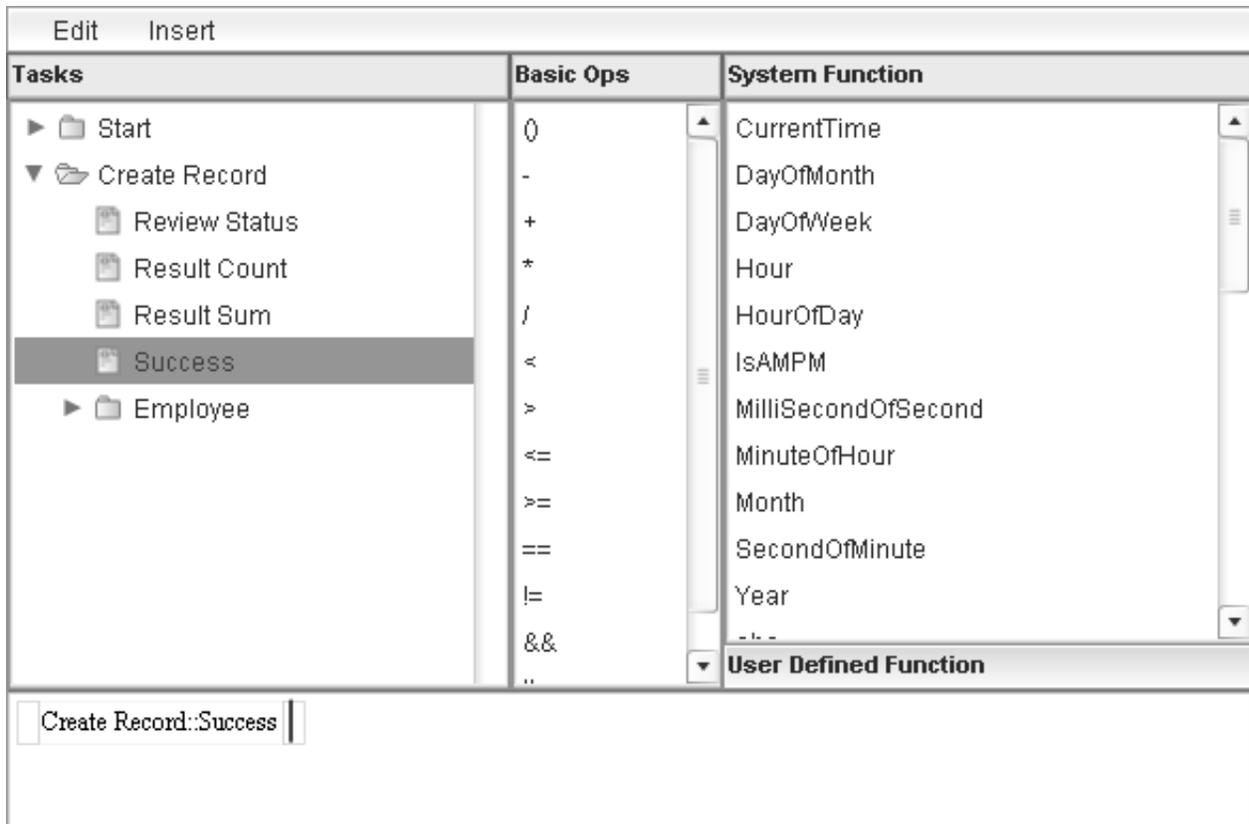
Figure 76. Condition Builder Success attribute

When we add the reference to the *Success* attribute, the Condition builder adds another gray rectangle after what we added. The new rectangle after what we added is automatically selected, so the next thing we add will be to the right of that.

The next thing to add to the condition is ==. The == will compare the value of the *Success* attribute to whatever we put to the right of the ==. The comparison will be true if the two have the same value or false if they do not.

To add the == click the == in the *Basic Ops* panel. For a complete list with explanations of the operators in the *Basic Ops* panel, see "Condition builder operators". Clicking an operator in the *Basic Ops* panel adds the operator to the condition. After you click the == the Condition Builder looks like the following figure.
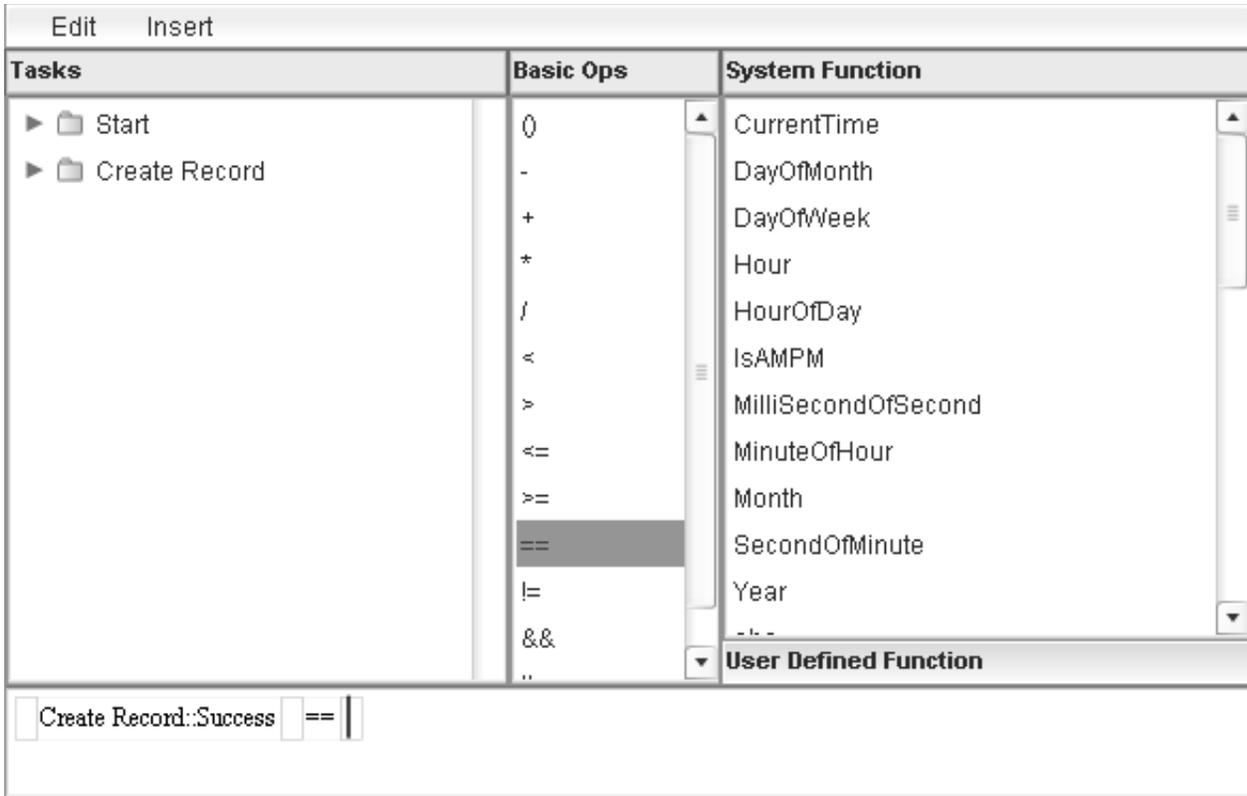
Figure 77. Condition Builder ==

The next thing to do is add the text "SUCCESS" so that the condition can be true if the value of the Create Record task's *Success* attribute is equal to "SUCCESS". To put a text or number value in a condition, we use the *Insert* menu at the top of the condition builder.

When you click the *Insert* menu, its menu items appear. They look like the following figure.
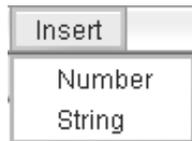

Figure 78. Insert menu

Clicking the *Number* menu item allows you to add a Number value to a condition. Clicking the *String* menu item allows you to add a text value to a condition. Clicking either of these menu items creates a box in the Condition where you can enter a number or text value as appropriate.

To enter a string value into the box that is added when you click the *String* menu item, click the box and then type the value. After you type the value "SUCCESS", the condition builder looks like the following figure.
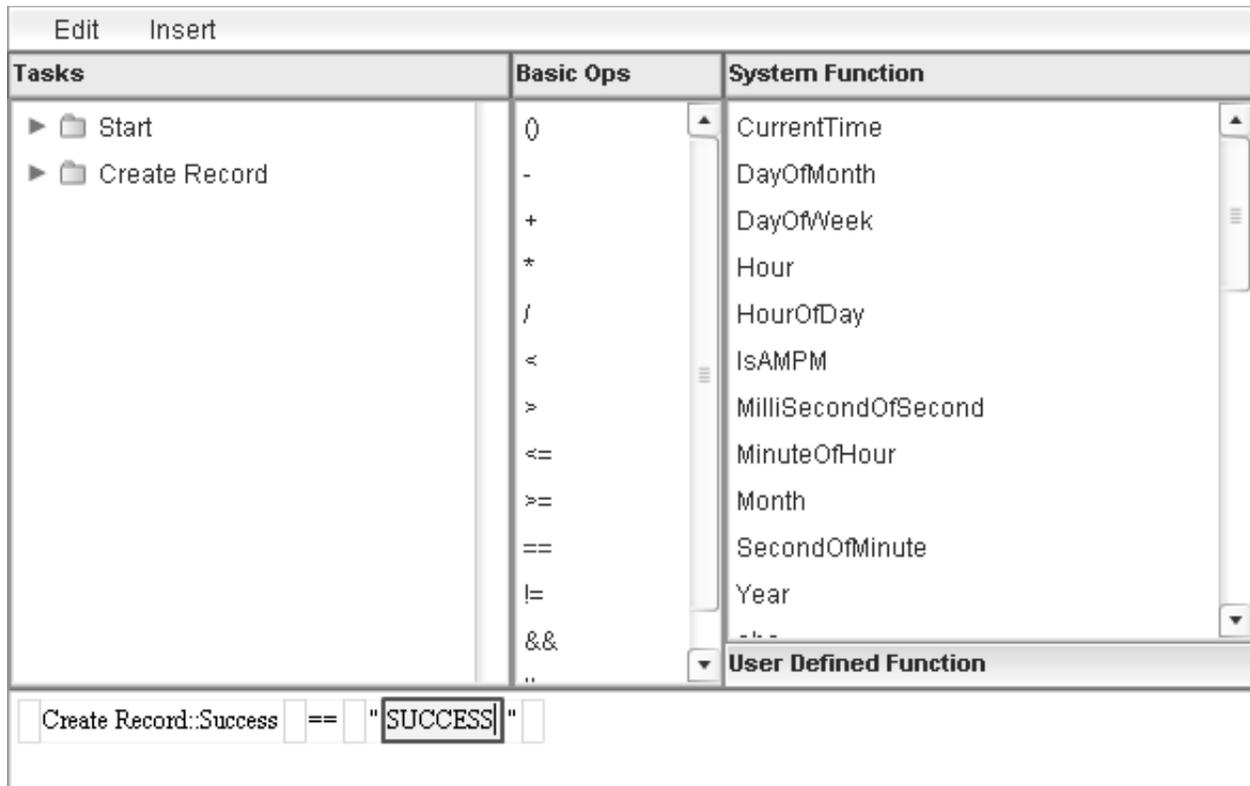
Figure 79. Condition Builder string

The next thing to add to the condition is the operator *&&*. The *&&* operator is a logical *&&* operator. The *and* operator is true if the conditions on both sides of the *&&* operator are true.

After we added *"SUCCESS"* to the condition, the part of the condition that was selected was the box that contains *"SUCCESS"*. The Condition Builder will not allow us to do anything to the condition unless one of the gray boxes is selected. Before we can add the *&&* operator to the end of the condition, we must select the gray box at the end of the condition by clicking it. We then add the *&&* operator by clicking *&&* in the *Basic Ops* panel.

The next thing we add to the condition is a reference to the *DateHire* field in the *Employee* record that was used to launch the workflow. Since the record that was used to launch the workflow is associated with the *Start* task, click the *Start* task's folder icon to see the list of things under the *Start* task. This list includes the *Employee* record that was used to launch the workflow.

To see what is in the *Employee* record we click the folder icon next to the name *Employee*. The Condition Builder now looks like the following figure.
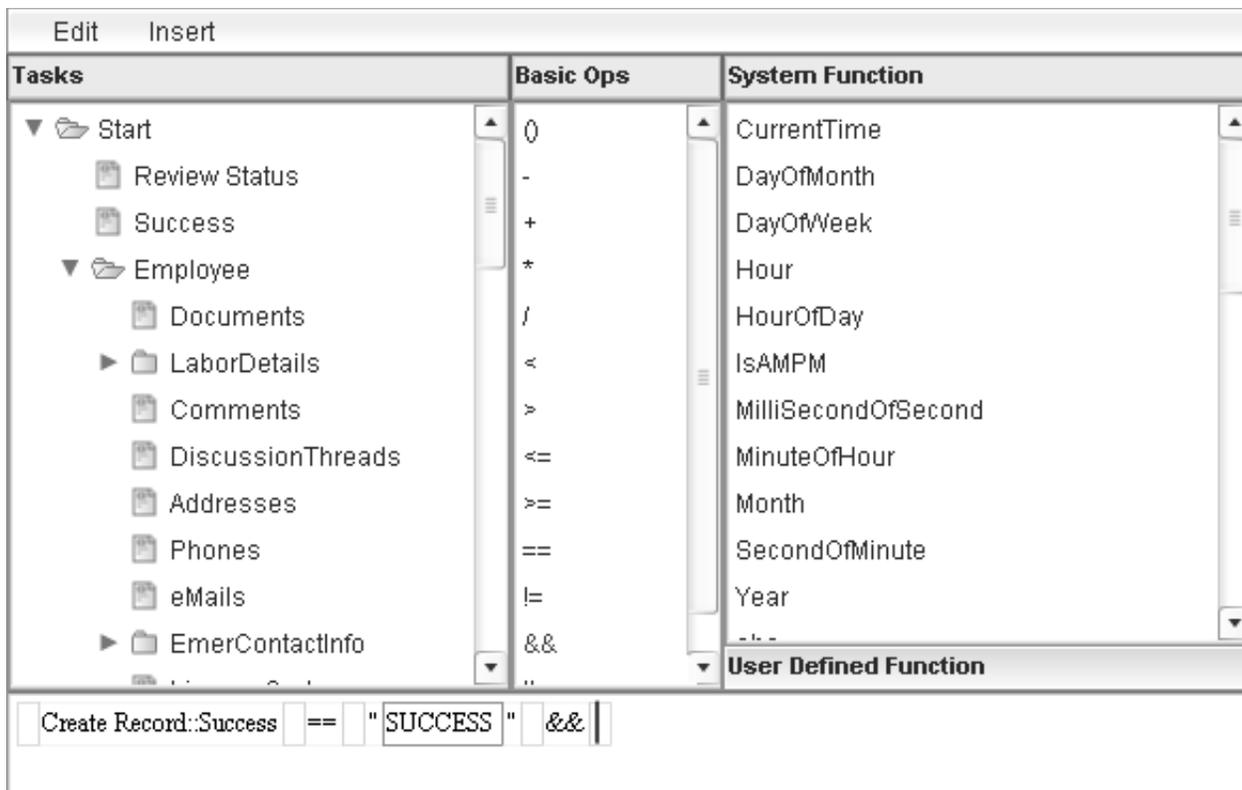
Figure 80. Condition Builder &&

The list that appears in the *Tasks* panel under the name of a kind of record is a list of section names. An icon appears next to each section name. If what appears next to a section's name is a document icon, the section is a type of section that the Condition Builder cannot use to access fields.

At the time of this writing, the Condition Builder is unable to access fields in multiple-record smart sections. Most of the sections in the figure that have a document icon are multiple-record smart sections.

If what appears next to a section's name is a folder icon or an open-folder icon, then the Condition Builder is able to access the section's fields. At the time of this writing, there are only two kinds of sections that have a folder icon or an open-folder icon next to their name. Single-record smart sections have one of these icons next to their name.

The other kind of section that has a folder icon or an open-folder icon next to its name really is not a section. The Condition Builder pretends that a business object's fields that are not in a smart section are in a section named *General*. This artificial *General* section has a folder icon or an open-folder icon next to its name.

An *Employee* record's *DateHire* field is in a section named *Detail*. To see the *Detail* section we scroll down to a lower part of the Tasks panel and click the folder icon next to the name of the *Detail* section. We can now see the name of the *DateHire* field, so we click it. The Condition Builder now looks like the following figure.
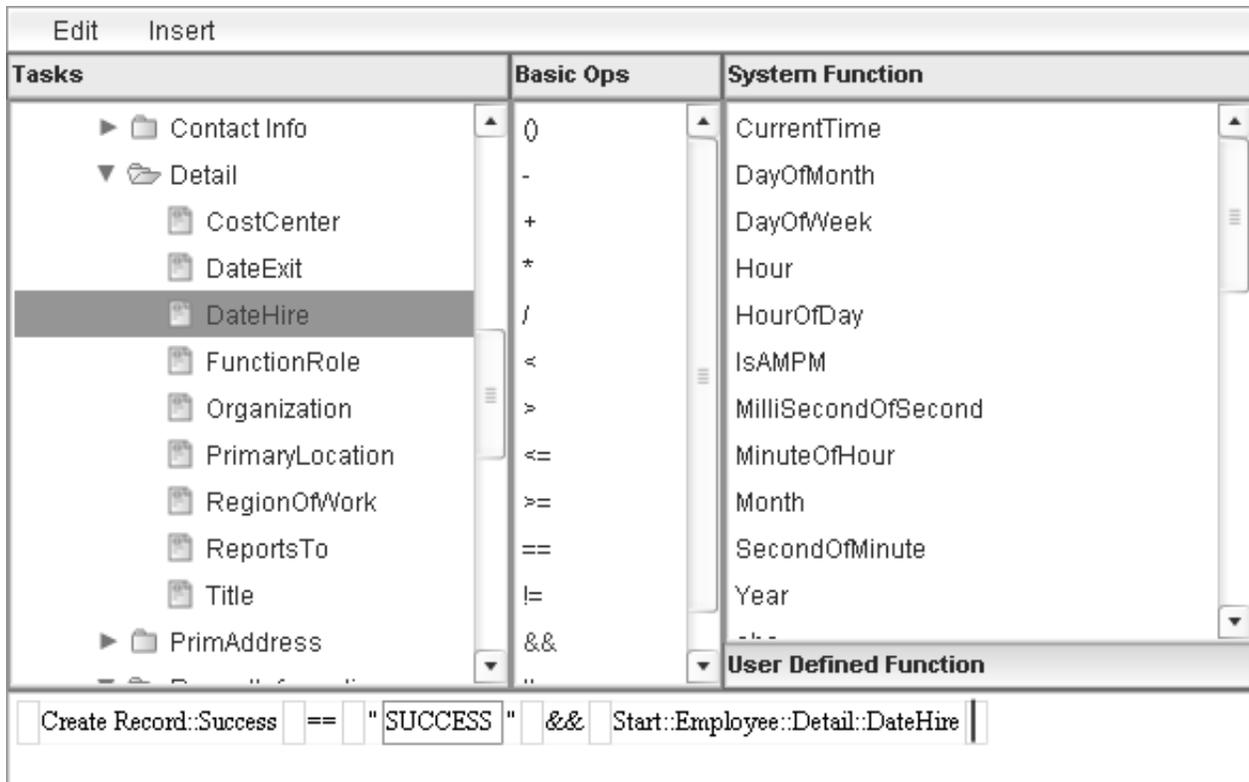
Figure 81. Condition Builder field

The next thing to add to the condition is a <= that will be used to determine if the value of the *DateHire* field is before or equal to the current time. We add it by clicking the <= in the *Basic Ops* panel.

We finish by clicking *CurrentTime* in the *System Function* panel. The Condition Builder now looks like the following figure.
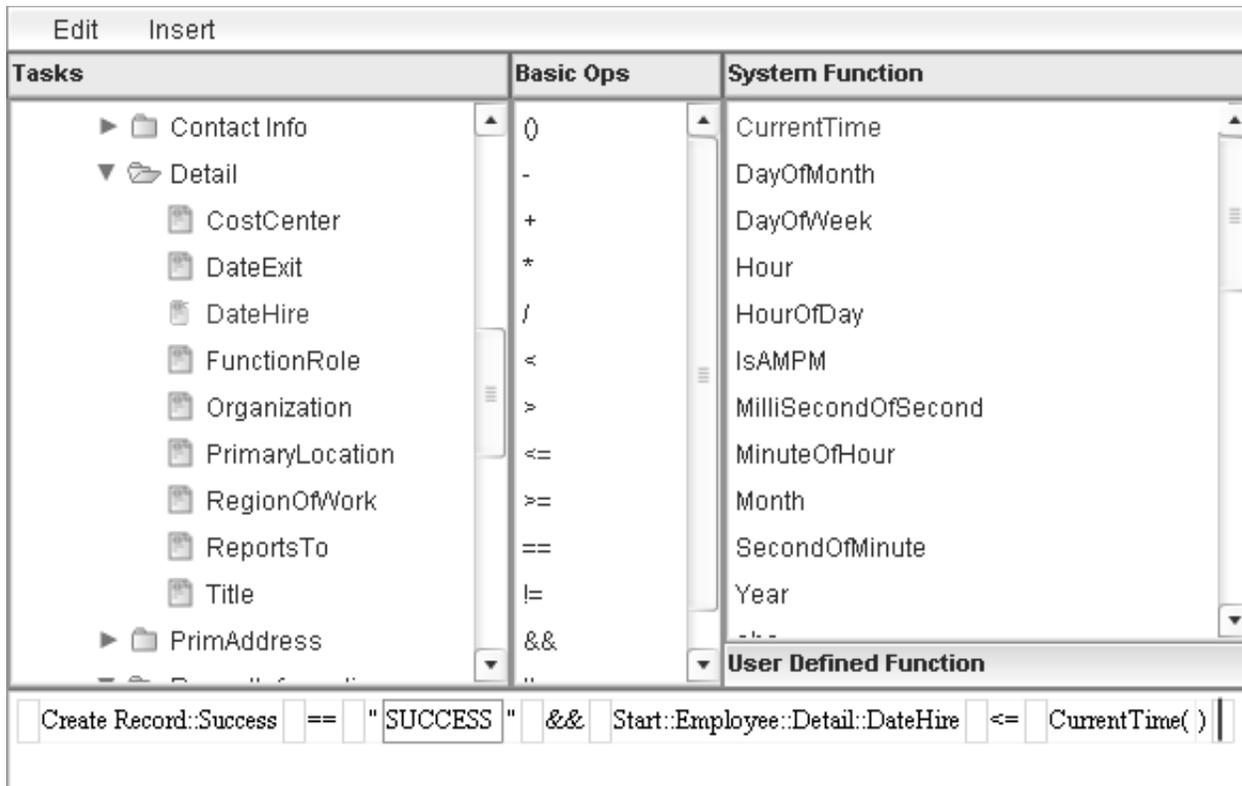
Figure 82. Condition Builder done

The system functions that you can click in the *System Function* panel are the same system functions that are used for extended formulas. For more information, see *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

You can access user defined functions by clicking the button labeled *User Defined Function*. These are the same user defined functions that are used for extended formulas. For more information, see *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

The operators in a condition are evaluated strictly from left to right, except for parentheses. For example, this is always true:

```
2 + 3 * 4 == 4 * (3 + 2)
```

For Integration, 0 is Not Selected and 1 is Selected. For more information on the Integration property, see "Start task".

Condition builder operators

The Basic Ops section of the Condition Builder contains the operators shown in the following table:

Table 23. Operators in the Basic Ops section

(	The left parenthesis is used with the right parenthesis to change the order in which operations in a formula are performed.
)	The right parenthesis is used with the left parenthesis to change the order in which operations in a formula are performed.
-	A minus sign is used for subtraction.

Table 23. Operators in the Basic Ops section (continued)

+	A plus sign is used for addition.
*	A star is used for multiplication.
/	A slash is used for division.
<	A less-than evaluates to true if the expression to its left evaluates to less than the expression to its right.
>	A greater-than evaluates to true if the expression to its left evaluates to greater than the expression to its right.
<=	A less-than-or-equals evaluates to true if the expression to its left evaluates to less than or equal to the expression on its right.
>=	A greater-than-or-equals evaluates to true if the expression to its left evaluates to greater than or equal to the expression to its right.
==	A double equals sign is used for equals. It evaluates to true if the expressions on both sides of it evaluate to the same value.
!=	Exclamation point equals is used for not-equals. It evaluates to true if the expressions on both sides of it evaluate to different values.
&&	A double ampersand is used for logical "and". It evaluates to true if the expressions on both sides of it evaluate to true.
||	A double vertical bar is used for logical "or". It evaluates to true if the expressions on either or both sides of it evaluate to true.

Fork task

A Fork task defines multiple sequences of tasks. A Fork task allows all its task sequences to be performed at the same time. After all the task sequences have been performed, the task after the Fork task is performed.

The following figure shows a simple example of a workflow that contains a Fork task.
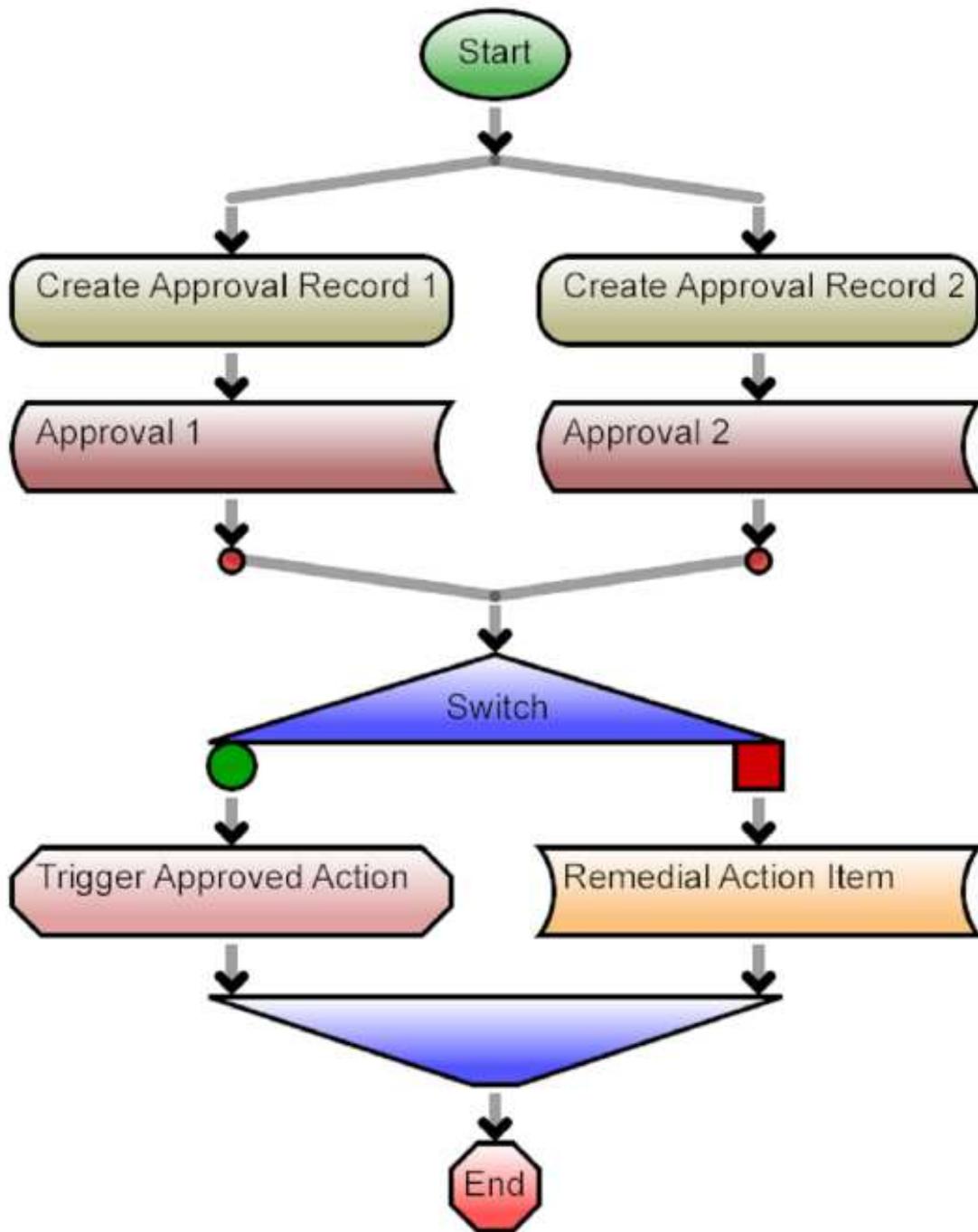
Figure 83. Fork example

One thing to notice about the Fork task as it appears in the figure is that there is more of it than just its palette shape. The bottom of the Fork task is where the task sequences come together.

The purpose of the sample workflow shown in the figure is to get two approvals for the record that caused the workflow to be launched. Rather than waiting for

one approval to be made before waiting for the second, the workflow waits for both approvals to happen at the same time. This way, the workflow can finish as soon as both approvals have been made, no matter which one happens first.

The workflow in the figure begins with a Fork task. The fork has two branches, one for each of the two approvals. The task after the fork is not performed until after all of the fork's branches are finished being performed, which is when both of the approvals have been done.

Each branch of the fork contains a similar sequence of tasks. It is not unusual for the branches of a fork to do different things. We chose an example in which the branches of a fork do similar things to keep the example simple. Each branch begins by creating a record for the kind of approval that the fork is responsible for. It then performs an approval task to get the appropriate approval.

When both approval tasks have completed, the fork is done. The task that follows the fork task is a switch task that causes the appropriate task to be performed, based on whether both approval tasks resulted in an approval or not.

You can click the top or bottom of a Fork task to see its properties form. A Fork task's properties form does not contain any properties. It does have two actions:
- The form for a fork has a *Delete* action for deleting the Fork task. This should be used with care, since deleting a Fork task also deletes all the tasks in all its branches.
- The form for a fork also has an *Add Branch* action. Clicking the *Add Branch* action has the effect of adding a new empty branch to the fork.

Clicking the circle at the bottom of a branch has the effect of selecting the branch and causing the properties form for the branch to be displayed. The properties form for a branch does not contain any properties, but it does have a *Delete* action for deleting the branch and all the tasks that the branch contains.

There are some benefits to allowing more than one thing to happen at the same time. The most obvious is that work can be completed sooner if some parts of the work can be done at the same time rather than completing the parts one after the other. If all the parts are done one after the other then the time required to finish the work will be the sum of the times it takes to perform each part. If all the parts are done at the same time, then the work takes only as long as the longest part.

When different parts of some work are being done at the same time, it is very important that the tasks being performed on the different parts do not interfere with each other. This one important consideration is the reason that the example in the figure is structured the way that it is structured.

Simply having two approval tasks try to approve the same record at the same time will not work because an approval task automatically locks the record to be approved from the time that the task starts until there is an action that ends the approval. This locking is not something that we want to try to bypass. The locking happens because it is always a bad idea to have one person modifying a record while someone else is modifying the record.

To avoid conflicts between whatever else is happening with the record being approved and the approvals, the example in the figure creates new records for the

approval. Because each approval works on a record that exists only for that approval, there can be no possible interference between the approvals and anything else.

Loop task

A Loop task contains a sequence of tasks that may be performed multiple times. The following figure shows an example of a workflow that contains a Loop task.
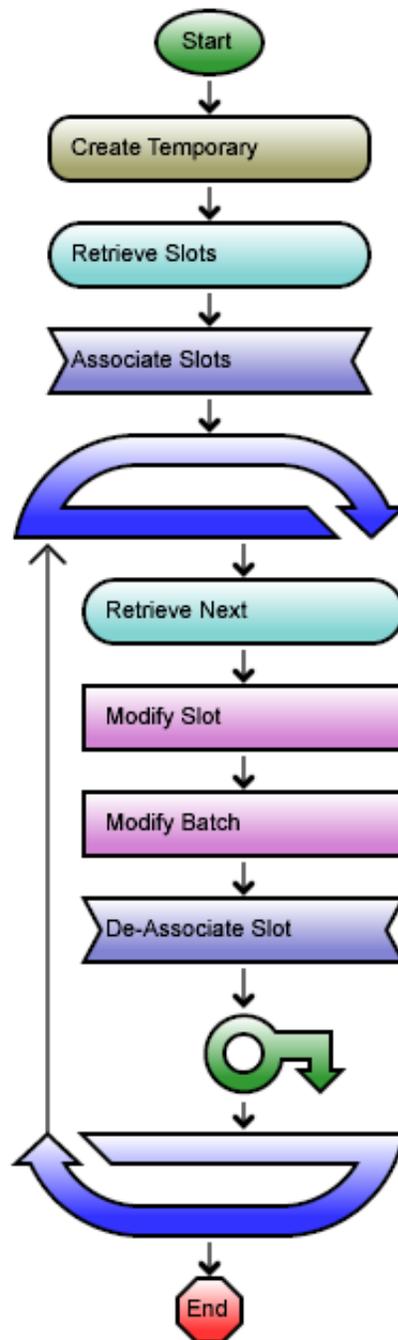

Figure 84. Loop example

One thing to notice about the Loop task as it appears in the figure is that there is more of it than just its palette shape. The bottom of the Loop task has an arrow that goes back to the top.

The purpose of the sample workflow shown in the figure is to find slots in a warehouse to put a new batch of products. The workflow begins by creating a temporary record that it will use to keep track of which slots remain to be checked. It retrieves the slots that may have sufficient storage space available. It then associates the slots with the temporary record. The next task is the loop task.

All that the Loop task does is perform the sequence of tasks it contains over and over. The first task in its sequence retrieves one of the slots associated with the temporary record.

The next task is a Break task. A Break task works with the Loop task that contains it. What a Break task does is decide if the next task to be performed should be the next task inside the loop or the task after the loop.

The Break task in the figure checks if there was a next slot for the preceding task to retrieve. If there was a slot to retrieve, the next task in the loop is the next task that will be performed. However, if there was no next slot, then the Break task breaks out of the loop, making the next task to be performed the end task after the loop.

Break tasks are described in "Break task".

If the loop continues after the Break task, it adds as much of the product batch as still remains and will fit to the slot. It then reduces the amount of product remaining to be stored. Finally, it removes the association between the slot it just checked and the temporary record.

After it finishes this, the Loop task has finished performing its sequence of tasks. The Loop task continues by executing its sequence of tasks again, beginning with the task labeled *Retrieve Next*.

The Loop task maintains a count of the current loop it is in. The loop count is exposed as the Result Count. The Loop task can be named so you can identify it in other tasks. The loop Result Count can be used in conditions, for example, in a Break task or Switch task condition.

The Loop task has a delay value. When the delay value is greater than 10, the loop will delay that many milliseconds before executing the next iteration.

Often, it is possible to arrange for a sequence of tasks to be performed on multiple records at once. When you can find such a sequence of tasks, it is always a better way to do things that way than using a loop. Loops are more complicated to set up, more difficult to get right and they take longer to run. Avoid loops when you can.

Iterator is simpler than loop

The Loop task is very flexible and can be used to create a great variety of loops. One of the most common kinds of loops is a loop where each time through the loop, the body of the loop works on a different record in a list of records.

Because this is a particularly common case, the platform has another workflow task named Iterator that makes it simpler to write this common type of loop. Use the Iterator task instead of the Loop task.

The Iterator task is discussed in "Iterator task".

Break task

A Break task works within an enclosing Loop task, Iterator task, or DataConnect task to decide whether the enclosing task should continue performing the sequence of tasks that it contains or exit the sequence, and whether the exit should be considered a success or an error. As you develop a workflow that contains a Break task, the Break task shape and color may change depending on the properties you select.

A Break task decides whether to continue with its enclosing task or to exit out of it based on the Break task properties and whether or not all of the conditions associated with them are true. The form for a Break task is used to set properties, to associate conditions with a Break task, and to manage conditions. The creation and management of conditions is discussed in "Example: Workflow condition builder".

The properties form for a Break task is organized into three sections. Here are descriptions of the fields in the first section:

Flow Type

> Loop, Iterator, and DataConnect tasks are tasks that define a block of tasks. Select Break if the workflow should exit from the block when the condition is met. Select Continue if the workflow should continue processing from the beginning of the block when the condition is met. The block to Break out of, or the block to Continue in, is determined by the Scope property.

> If a Break task is encountered, the next task is determined by the values in the Break Scope section. The Flow Type (Continue or Break) determines whether it continues on the next iteration or loop of the enclosing task or breaks out. The Task Status determines what the transaction management should do.

Task Status

> Indicates how the status of the enclosing block should be set on Break or Continue. In a DataConnect task the status is used as part of the transaction control. The DataConnect task is described in "DataConnect task".

> Selecting Success may cause a commit if the value of the block in the Break Scope section is a controlling block and it is time for the commit based on the transaction settings or if breaking out of the controlling block. Selecting Error causes a rollback if the value of the block in the Break Scope section is the controlling block or if breaking out of the controlling block.

Break scope section

The second section of the Break task properties form is labeled Break Scope. The purpose of the Break Scope section is to specify what the Break task is controlling. The values for Break Scope are as follows:

Current Block

When the workflow encounters this Break task, processing either continues or breaks from the current block.

Selected Block

When the workflow encounters this Break task, processing either continues or breaks at the block selected from the drop-down list. The tasks available in the Selected Block drop-down are above the current task.

DataConnect section

The third section of the Break task properties form is labeled DataConnect. The purpose of the DataConnect section is to specify what happens to the temporary data.

The platform ignores properties in the DataConnect section when the Break task is not used in conjunction with a DataConnect task or when temporary data is not enabled (Use Temporary Data not checked) for the DataConnect task.

The values for DataConnect are as follows:

Discard Temporary Data

When the workflow encounters this Break task:

- None = do not discard temporary data
- Current = discard the temporary data in the current iteration
- All = discard all temporary data.

Fail Staging Rows

This property is visible only when Discard Temporary Data is Current or All. Select the Fail Staging Rows check box to fail the DataConnect row that the DataConnect task was processing.

When the Fail Staging Rows check box is selected, the system displays Current and All properties for the Fail Staging Rows property. These properties are read-only and mirror the properties in Discard Temporary Data.

If the workflow has multiple DataConnect tasks, one within another, and the Break scope of an inner DataConnect task points to an outer DataConnect task, if Discard Temporary Data is Current, the platform discards the current iteration of the outer DataConnect task (and therefore also discards all the inner DataConnect task iterations).

The shape and color of the Break task changes depending on how it is configured. If there is a red dot inside the Break task, it means the Task Status will be Error if the condition is met. No dot inside the Break task means the Task Status will be Success if the condition is met.

A Break task with a round interior and green color indicates a regular Break or Continue. If the interior is square and the task shape is gold color, the condition is swapped.

The chart below shows the different shapes for the Break task:

Table 24. Break task shapes

Shape	Description
	Break
	Break with swapped conditions
	Break with error status
	Break with swapped conditions and error status
	Continue
	Continue with swapped conditions
	Continue with error status
	Continue with swapped conditions and error status

The normal color for a Break task is green. If a Break task is green, it is normal (not swapped) and will break (or continue at the top) when the condition is true. Clicking the *Swap* action in the task's properties form causes the Break task to turn gold. If the Break task is already gold, clicking the *Swap* action causes it to turn green.

Iterator task

An Iterator task contains a sequence of tasks that may be performed multiple times, once for each record associated with a specified workflow task. Iterators are used to loop through a finite set of records and perform one or more tasks individually on the records that the Iterator loops through.

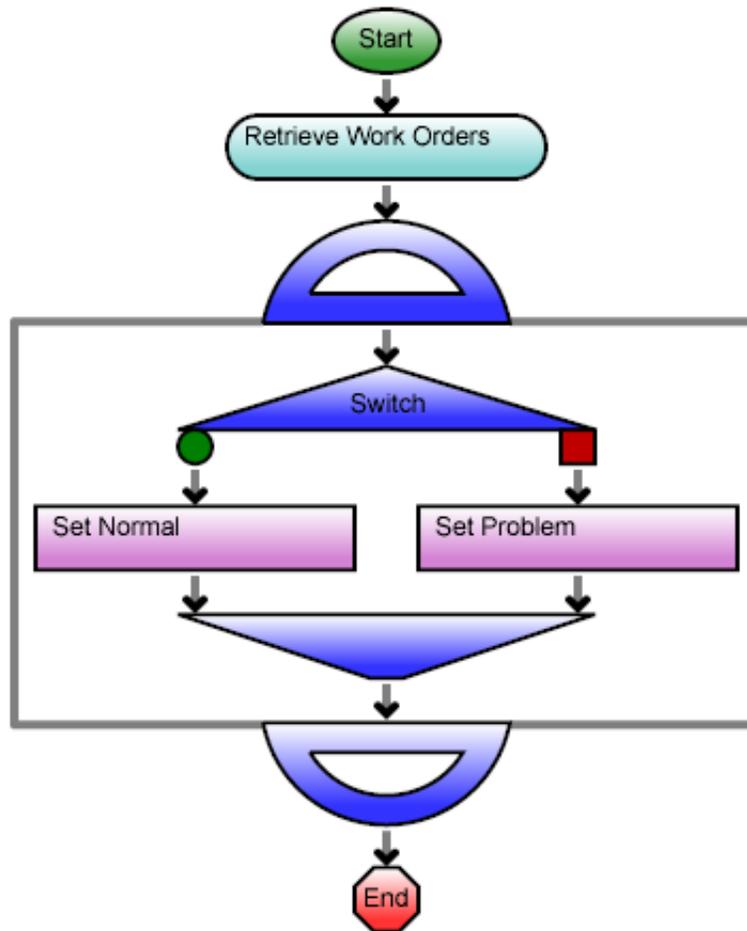The following figure shows an example of a workflow that contains an Iterator task.

Figure 85. Iterator example

One thing to notice about the Iterator task as it appears in the figure is that the end of the Iterator task is the Iterator palette shape upside down.

Here is a summary of what the workflow in the figure does:

- It retrieves work orders associated with a person.
- The Iterator task performs the workflow tasks inside of it, once for each retrieved work order.
- Using a complex condition, the Switch task determines if each work order is proceeding normally or is a problem.

The properties for an Iterator task specify which other task's associated records will be used for looping. An Iterator task performs the sequence of tasks it contains once for each record associated with the specified task. Each time it performs the sequence of tasks, the Iterator task becomes associated with a different one of the records that is associated with the other task.

The properties form for an Iterator task is organized into two sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Iterate records section

The second section of the Iterator task properties form is labeled *Iterate Records*. The purpose of the *Iterate Records* section is to specify the workflow task whose associated records will control the looping of this task.

At the top of the *Iterate Records* section are fields used to identify the record that will control this task's looping.

Here are descriptions of the fields:

Take the

This is a drop-down list. Though it is possible to select other values, the value of this field should be *Business Object*.

of Task

The value of this field is the label of the task that will control this task's looping.

At the bottom of the *Iterate Records* section is a read-only field labeled *Object Type*. The platform sets the value of this field to the type of record that is associated with the workflow task specified by the *of Task* field.

Call workflow task

The purpose of the Call Workflow task is to launch a synchronous or subflow workflow. Synchronous workflows are described in "Synchronous versus asynchronous workflows" and subflow workflows are described in "Start task".

The initial project for a Call Workflow task is the project context at the time the Call Workflow task runs.

The properties form for a Call Workflow task is organized into two sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Static Workflow

The value of this field is the name of the workflow that this task will launch. This field's value is chosen from its drop-down list. The names in the drop-down list will be the names of synchronous and subflow workflows that can be launched from the type of record identified in this form's *Records* section.

The Workflow list in a Call Workflow task is limited to those workflows that have the same business object as set in the Records section of the task. Module-level workflows for the module of the object specified are also listed.

You may need to set the value of the fields in the *Records* section before you can set this field to its proper value.

Clicking the View Static Workflow action causes the workflow identified in the Workflow drop down to appear in the workflow editor. This is useful for making sure the workflow you have identified is the one you want.

The Parameters action only appears if the workflow identified has parameters or return values. Clicking the Parameters action brings up the Workflow Parameters and Return Values screen. There is more information about this screen in "Example: Workflow parameters and return values (call workflow)".

Dynamic Workflow
The values of this field are used to dynamically define the name of the workflow that this task will launch. For the "Task with Workflow Name" drop-down list, select the task record that contains the workflow name. For the "Workflow Name Field", select the field on the task record that contains the workflow name. This record field must contain the workflow name that you want to launch.

Records section (call workflow)

The second section of the Call Workflow task properties form is labeled *Records*. The purpose of the *Records* section is to specify the record that will be used to launch the workflow.

At the top of the *Records* section are fields used to identify a target record. The target record is used to determine the record that will be used to launch the workflow.

There are radio buttons below the fields. The way that target records are used to determine the record that will be used to launch the workflow depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
This is a drop-down list that can have one of three possible values:

Business Object
If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee
If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that will be used to launch the workflow.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record used to launch the workflow.

Use its Reference

> If this is selected, a record referenced by a smart section or locator field of the target records will be the record used to launch the workflow. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

> If this is selected, records associated with the target record will be the record used to launch the workflow. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

> After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

> This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

> When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If -Any- appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click -Any- which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the record used to launch the workflow.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is -Any-. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be used to launch the workflow.

If multiple records are identified by the information in the Records section, a separate instance of the workflow will be launched for each of the records. The initiating workflow will not continue to the next task until the called workflow is complete for all records.

Example: Workflow parameters and return values (call workflow)

The Workflow Parameters and Return Values screen appears whenever you click the Parameters action. It shows the parameters and return values from the workflow being called.

An example best explains how to use the Workflow Parameters and Return Values screen. The following figure starts this example with a workflow containing variables to hold the input parameters and the return value for a call to a subflow workflow. The example could also be a synchronous workflow, except the parameters could not be required.
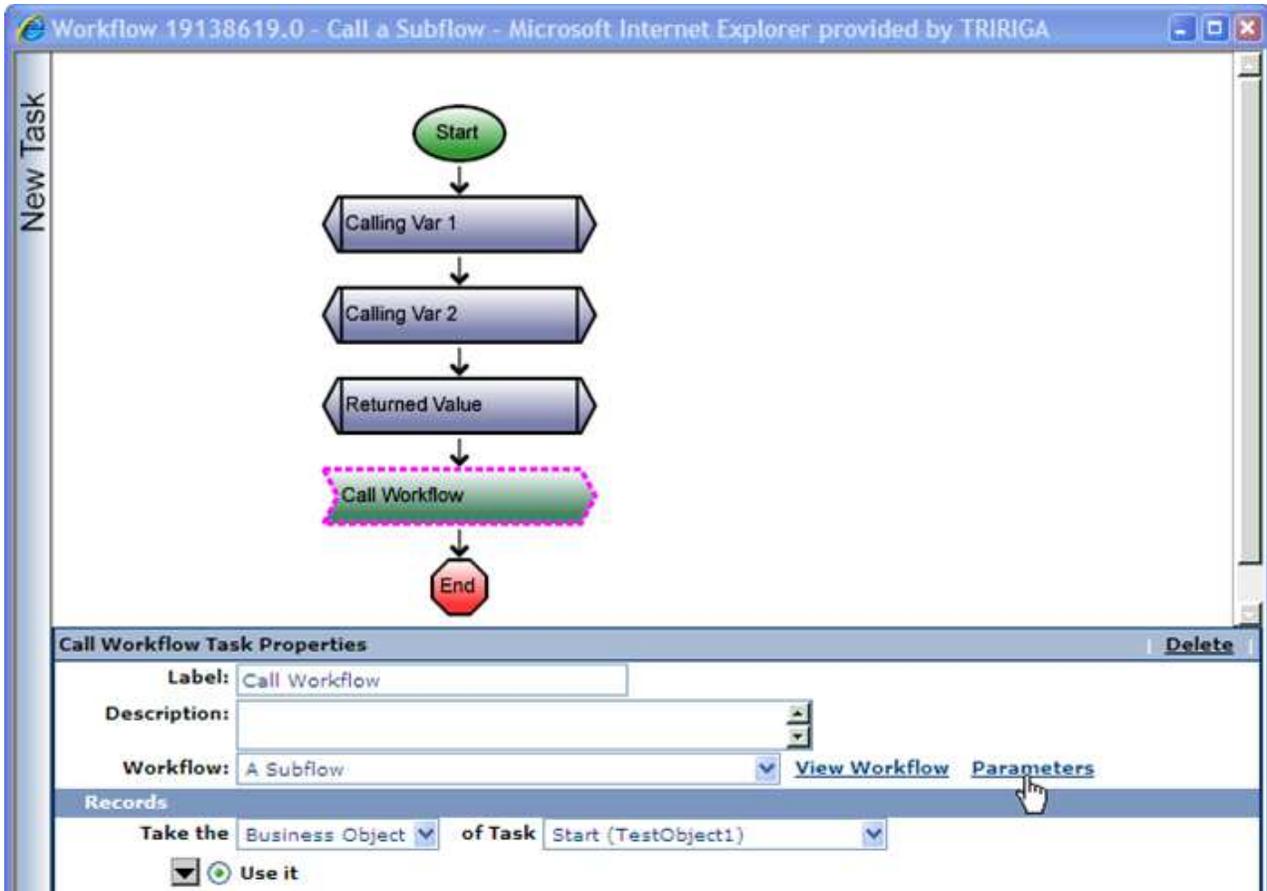
Figure 86. Call Workflow task example

Of course to really be of value this workflow would contain some tasks that would assign values to the variables we are using as the source for our input parameters.

Click Parameters. The Parameters link in the Workflow Parameters and Return Values screen appears. The first time you click the Parameters link the screen does not show any values.

Figure 87. Initial Workflow Parameters and Return Values screen

Click the Initialize action and the system fills in the initial values in both the Parameters section and the Return Values section based on the workflow being
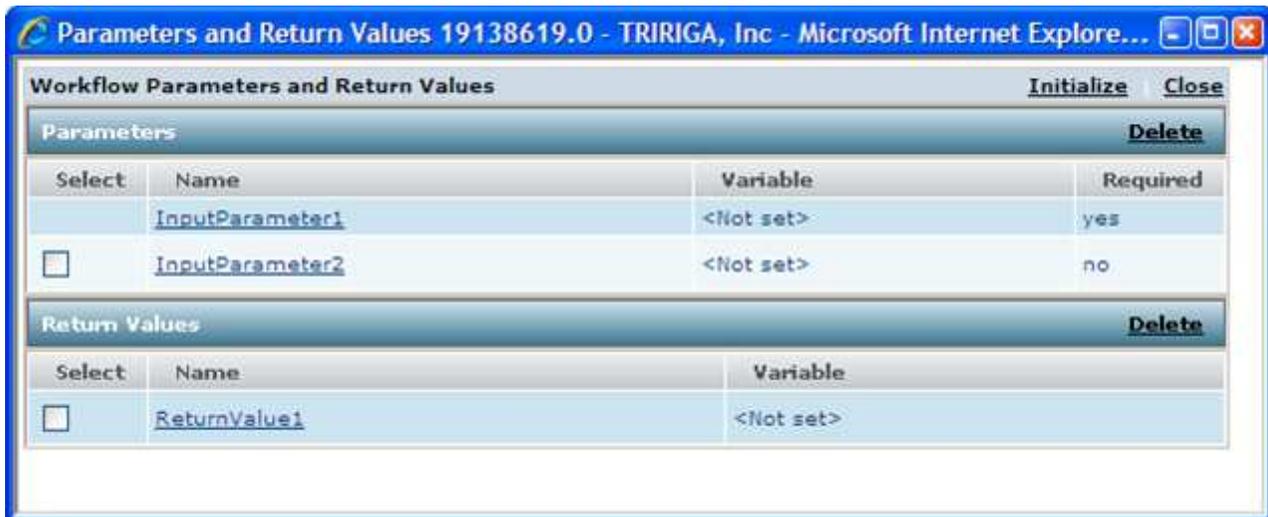
called.

Figure 88. After Initialize action

The parameter and return value names from the subflow are filled in. One of the parameters is required and therefore does not have a check box to allow selecting it to be deleted.

The variables to use within this workflow have not been assigned yet. Assign them by clicking on the Name. The Workflow Parameter Definition screen appears.
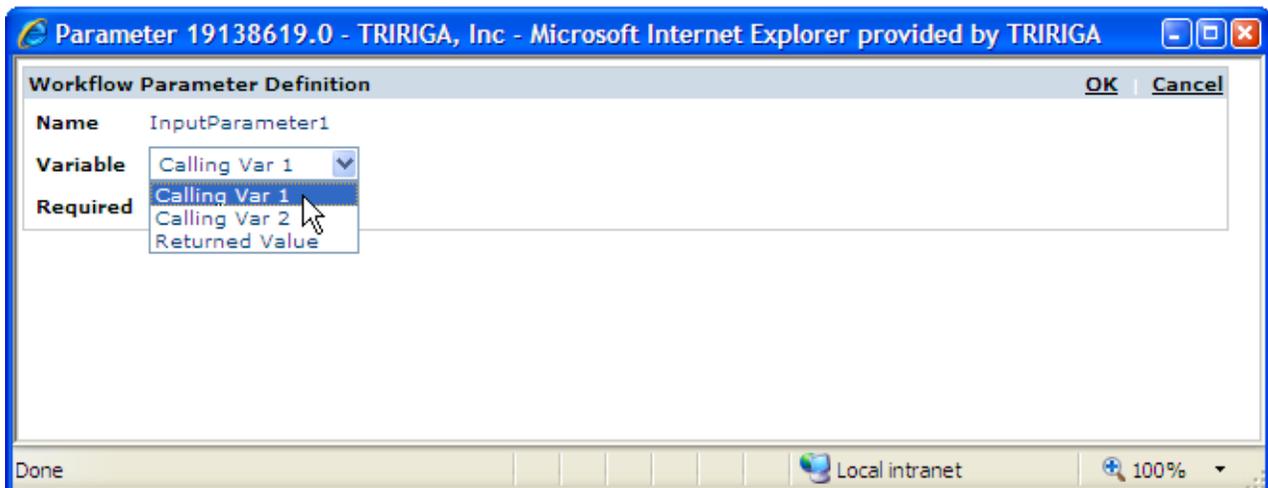

Figure 89. Workflow Parameter Definition screen

The choices in the Variable drop down are the variables defined at the top of the workflow. Click OK after selecting the Variable. The value displayed in the Required field reflects the value in the workflow being called.
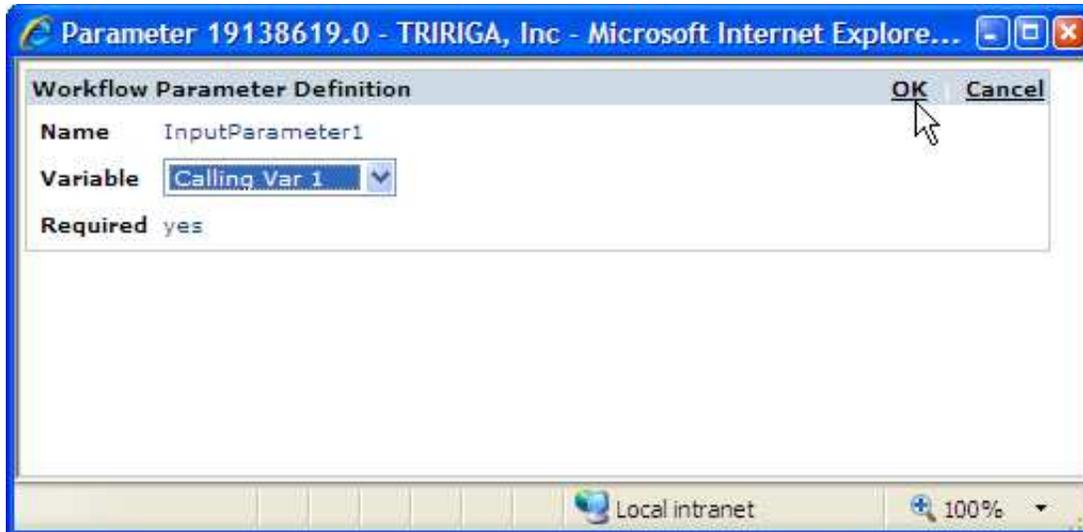
Figure 90. Workflow parameter defined

Repeat the definition process for the other parameter and the return value. Since the second parameter is optional it could be deleted rather than assigning it a value. You can always choose to ignore any returned values by deleting them. Any that are left in the Workflow Parameters and Return Values screen must be valid, which means the input parameters must be assigned to variables. Any left with <Not set> in the Variable column will prevent the workflow from publishing.

In the following figure, the second (optional) input parameter has been deleted and the return value assigned a variable.
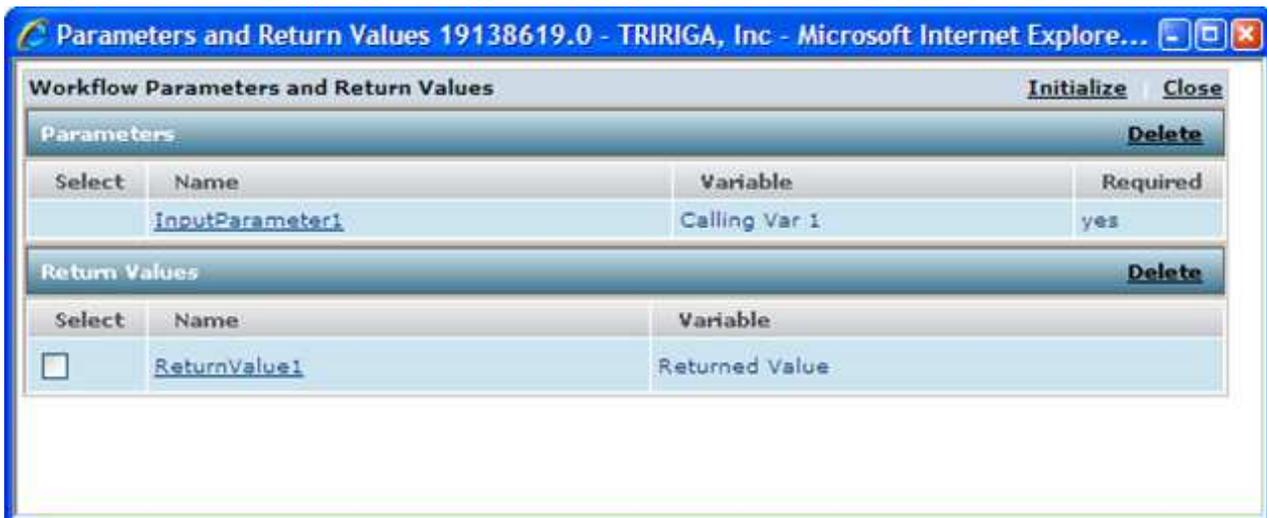

Figure 91. Optional parameter deleted

If after deleting a parameter or return value you decide you want it back, click Initialize and assign a variable to it.
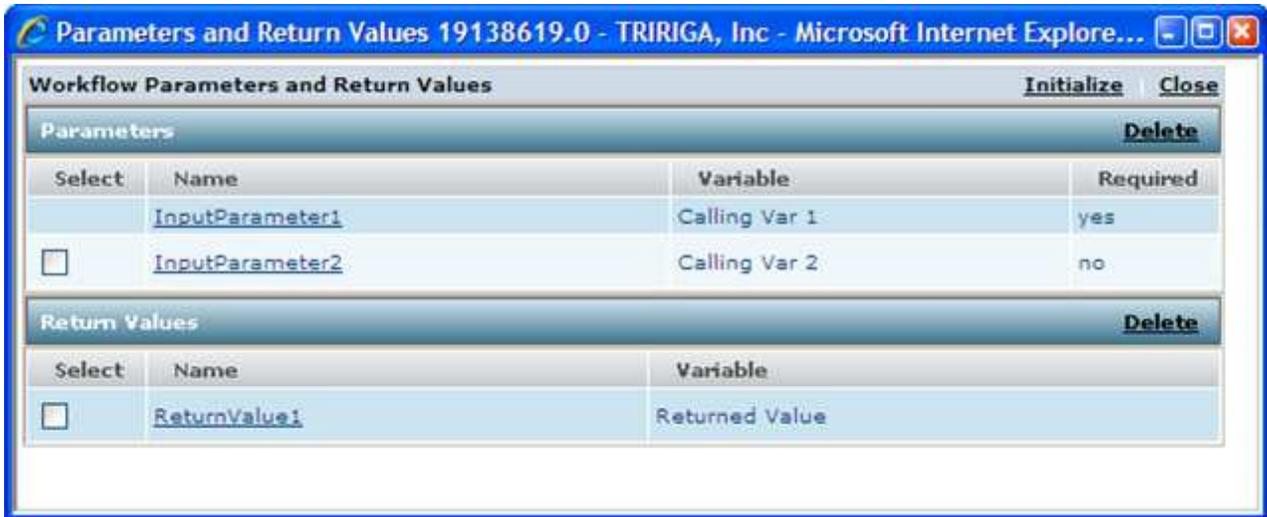
Figure 92. All parameters and return values defined

If the calling workflow has invalid parameters or returns (either required parameters missing or any that do not have a variable assigned), it cannot be published. Attempting to do so produces an error message.
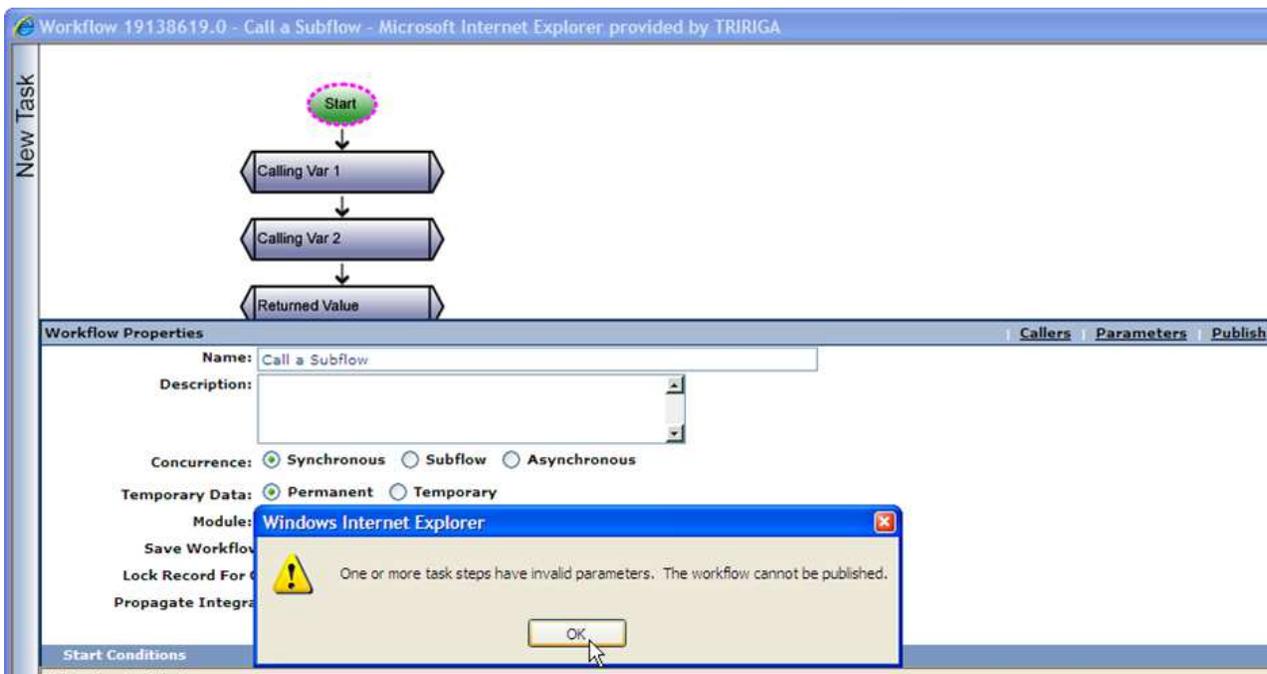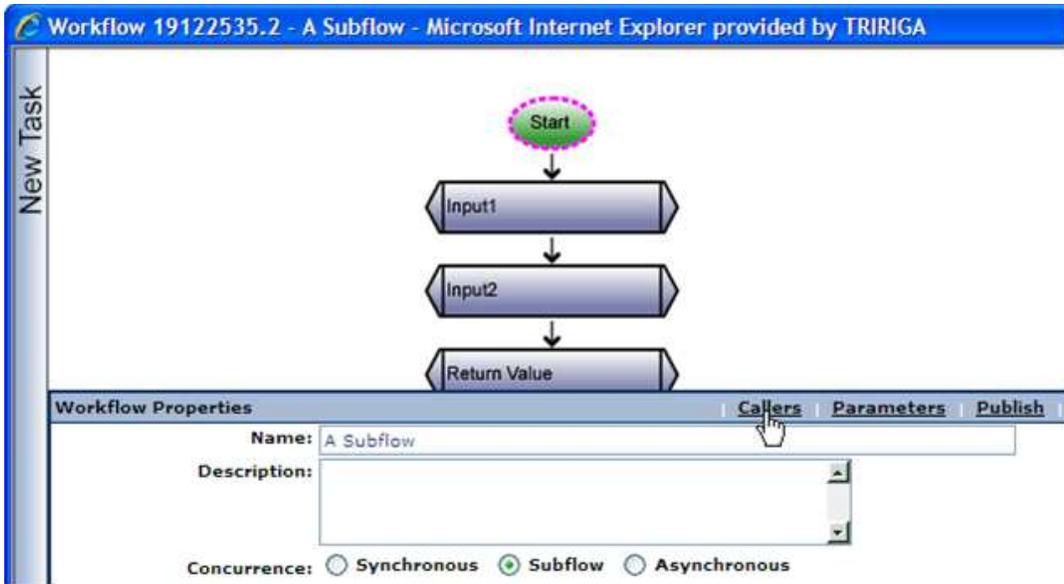

Figure 93. Error message when calling workflow has invalid parameter

If you were to try to publish the called workflow while there is a calling workflow with invalid parameters or returns, the system displays a warning. As this is a warning message, you can choose to publish it and then go fix the calling workflow(s) later or can cancel and look at the calling workflow(s) now. You can use the Callers link to view the workflows that have problems. Clicking Callers brings up the Calling Workflows Parameters Check screen. Click a hyperlinked Name to open the workflow so the problem can be resolved.

Figure 94. Calling Workflows Parameters Check screen

Note the Calling Workflows Parameters Check identifies the workflows with errors by showing Invalid in red in the Parameters column.

Within a workflow, Call Workflow tasks that call a workflow with parameters are checked to verify that the parameter and return value mapping is valid. If the mapping is not valid the Call Workflow task is outlined in red and the workflow will not publish. The following conditions make the mapping invalid:

- A required subflow parameter is not assigned a value.
- A parameter is assigned and the parameter does not exist on the called workflow (can happen due to modifications to the called workflow after the mapping was defined).
- A parameter is assigned and the type is not compatible (can happen due to modifications to the called workflow after the mapping was defined).
- A variable is assigned from a return value that does not exist on the called workflow (can happen due to modifications to the called workflow after the mapping was defined).

When a workflow with parameters and/or return values is published, the system checks calling workflows to verify that the parameter and return value mappings are valid. Any workflows that have invalid mappings an be viewed by using the Callers link.

Custom task

The purpose of the Custom task is to allow external code to run within the context of a firing workflow. This is something that no other platform-supplied workflow task can do. Among the possible uses for this are complex computations, communication with other applications, and interfacing with IBM TRIRIGA Connector for Business Applications.

To understand and successfully use a Custom task, you must have a thorough understanding of the IBM TRIRIGA Application Platform, the fundamental concepts required to build applications on this platform, and Java development.

The Custom task can work with any Java class that implements either the *com.tririga.pub.workflow.CustomTask* interface described in "CustomTask interface" or the *com.tririga.pub.workflow.CustomBusinessConnectTask* interface. The task creates an instance of the specified class using a zero argument constructor. The task then calls the instance's *execute* method, passing it the IDs of the records in the result list of a specified task. The called Java code can then use the record IDs to interact with the IBM TRIRIGA Application Platform using IBM TRIRIGA Connector for Business Applications. IBM TRIRIGA Connector for Business Applications is described in the *IBM TRIRIGA Connector for Business Applications 3 Technical Specifications*.

Each time the Custom task is performed, it determines which interface is being used, creates a new instance of the specified Java class, and calls its *execute* method.

The custom Java class must be specified in the *Class Name* field of the Custom Task object in the workflow. In order for the execution to occur properly, the specified class must be in the application classpath. For example, if you were using JBoss as your application server and you had a Custom task with the *Class Name* of *com.company.myFirstCustomTask*, you would place the compiled *com.company.myFirstCustomTask.class* file into the `jboss/server/all/lib` directory.

Custom task properties form

The properties form for a Custom task is organized into two sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Class Name
This is the name of a Java class that implements the interface.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed
Recalculates most formulas only if any input value of the formula

changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Records section (custom task)

The second section of the Custom task properties form is labeled *Records*. The purpose of the *Records* section is to specify the record(s) whose ID will be passed to the Java method.

At the top of the *Records* section are fields used to identify a target record. The target record is used to determine the record(s) whose ID will be passed to the Java method.

There are radio buttons below the fields. The way that target records are used to determine the record(s) whose ID will be passed to the Java method depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

This is a drop-down list that can have one of three possible values:

Business Object

If *Business Object* is selected, the record associated with the task specified by the field to the right of this one will be the target record.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART,* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record(s) whose ID will be passed to the Java method depends.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the ID of the target record(s) will be passed to the Java method.

Use its Reference

If this is selected, the ID of record(s) referenced by a smart section or locator field of the target records will be passed to the Java method. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, the ID of record(s) associated with the target record will be passed to the Java method. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

>This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

>When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, click the Module icon. A list of modules will pop up. Click the correct module.

>You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

>To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

>If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the ID of the target records' parent will be passed to the Java method.

>When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

>One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record whose ID will be passed to the Java method.

CustomTask interface

The Java class is *com.tririga.pub.workflow.CustomTask* shown below.

```
package com.tririga.pub.workflow;

/**
 * This interface must be implemented by any class specified in
 * the "Class Name" field of a Custom Workflow Task.
 * The implementing class must have a zero argument constructor.
 * The workflow runtime engine will get the Class and call the
 * zero argument constructor through reflection.
 * It will then call the execute method and set the success
 * attribute of the task based on the value returned.
```

```
 */
public interface CustomTask {
    /**
     * This method is called by a custom workflow task.
     *
     * @param recordIds
     *          An array of record IDs specified by the Record
     *          section of the task's properties.
     * @return true if successful, false if not
     */
    public boolean execute(long[] recordIds);
}
```

CustomBusinessConnectTask interface

The Java interface class CustomBusinessConnectTask contains one method called execute whose parameters are a preauthenticated TririgaWS Interface object, the userId of the person who triggered the workflow, and the array of Record objects.

```
public interface CustomBusinessConnectTask {
   public boolean execute(TririgaWS twsClient, long userId, Record [] records);
}
```

When implementing this class, the first thing that needs to be done in order to allow the twsClient to work properly is to call the register method. This will verify the context of the client in relation to the workflow that triggered it. The register method takes the userId as an argument. An example is provided below:

```
public boolean execute(TririgaWS tws, long userId, Record [] records) {
   try {
      //you must validate that the tws object has been preauthenticated
      //and that an existing active session user is being utilized
      tws.register(userId);
      //...
   } catch(Exception e) {
      e.printStackTrace();
      return false;
   }
}
```

Custom task uses

Custom tasks can be used to do things such as adjust the dates in Gantt sections, or adjust the free/busy status in Availability Legacy sections. Custom tasks can be used to retrieve field values dynamically from the setup record of a field, or execute an integration object record or integration definition. Custom tasks can also be used to turn on and off Data Load mode, and rebuild the hierarchy cache.

When you create a custom task, you can specify the following values in the *Class Name* field:

com.tririga.platform.admin.cache.web.CacheProcessingCustomTask $RefreshAllCache

> Triggering this custom task does not give an indication of any change. However, it produces the same effect as selecting *All Caches (Global)* in the Administrator Console Cache Manager. Validate there are no errors in the logs.

com.tririga.platform.admin.cache.web.CacheProcessingCustomTask $SetDataLoadMode

> Triggering this custom task shows a message in server.log indicating that you are now in *Data Load* mode. You can validate this in the Administrator Console Cache Manager, where it should also indicate *Data Load* mode.

**com.tririga.platform.admin.cache.web.CacheProcessingCustomTask
$SetNormalMode**
> Triggering this custom task shows a message in `server.log` indicating that
> you are now in *Normal* mode. You can validate this in the Administrator
> Console Cache Manager, where it should also indicate *Normal* mode.

**com.tririga.platform.admin.cache.web.CacheProcessingCustomTask
$ClearCacheAndRebuildHierarchyTree**
> Triggering this custom task does not give an indication of any change.
> However, it produces the same effect as selecting *Hierarchy Tree Data - with
> rebuild* in the Administrator Console Cache Manager. Validate there are no
> errors in the logs.

com.tririga.platform.util.customtasks.DynamicFieldValueCustomTask$ProcessFieldValue
> Triggering this custom task with the parameter
> triDynamicFieldValueHelpers retrieves field values dynamically from the
> setup record of a field. The workflow parameter can contain one or more
> helper records that are built from the triDynamicFieldValueHelper business
> object in the System module.
>
> In the triRecordIdNU field, enter the record ID of the IBM TRIRIGA record
> from which you are retrieving the value. In the triFieldNameTX field, enter
> the field name for the value that is being retrieved.
>
> When the CustomTask completes processing, the triFieldValueTX and the
> triFieldDisplayValueTX fields on each triDynamicFieldValueHelper record
> will contain the non-formatted value and the translated display value for
> the value that is being retrieved.

com.tririga.platform.util.customtasks.DynamicFileAttachCustomTask$ProcessNotificationQueryAt
> Triggering this custom task allows for dynamically attaching report results
> to notification emails. The report types Report, Query, and External are
> supported.
>
> You must pass a parameter called triDynamicFileAttachHelpers. It can
> contain one or more triDynamicFileAttachHelper records. The business
> object triDynamicFileAttachHelper resides in the System module. It
> contains the fields triQueryModuleNameTX, triQueryBoNameTX,
> triQueryNameTX, triQueryIdNU, and triFileAttachmentNameTX.
>
> Map the triDynamicFileAttachHelper record to either have a
> triQueryIDNU value to represent the query id of the report to attach. Or
> enter the appropriate values in the triQueryModuleNameTX,
> triQueryBoNameTX, triQueryNameTX fields for the custom task to look up
> the query id of the report to attach. The value mapped to the
> triFileAttachmentNameTX field is the name that you want give to the
> attachment file.
>
> The target record of the CustomTask workflow task should be a
> notification record.

For more information about custom tasks, see the following areas:

Gantt sections
> Custom tasks with Gantt sections are described in "Workflow custom tasks
> that adjust dates in Gantt system".

Availability Legacy sections
> Custom tasks with Availability Legacy sections are described in "Workflow
> custom tasks that adjust free status".

Integration Object records

Custom tasks with Integration Object records are discussed in the *IBM TRIRIGA Application Platform 3 Connector User Guide*.

DataConnect task

The purpose of the DataConnect task is to invoke IBM TRIRIGA DataConnect for moving data from IBM TRIRIGA DataConnect staging tables into IBM TRIRIGA native business object records. The properties of this task determine which record type is processed. For more information about DataConnect, see *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

The DataConnect task is in the task palette only if the workflow is asynchronous and has the Integration property checked. Asynchronous workflows are described in "Synchronous versus asynchronous workflows". The Integration property is described in "Start task".

The following figure shows an example of a workflow that contains DataConnect tasks.
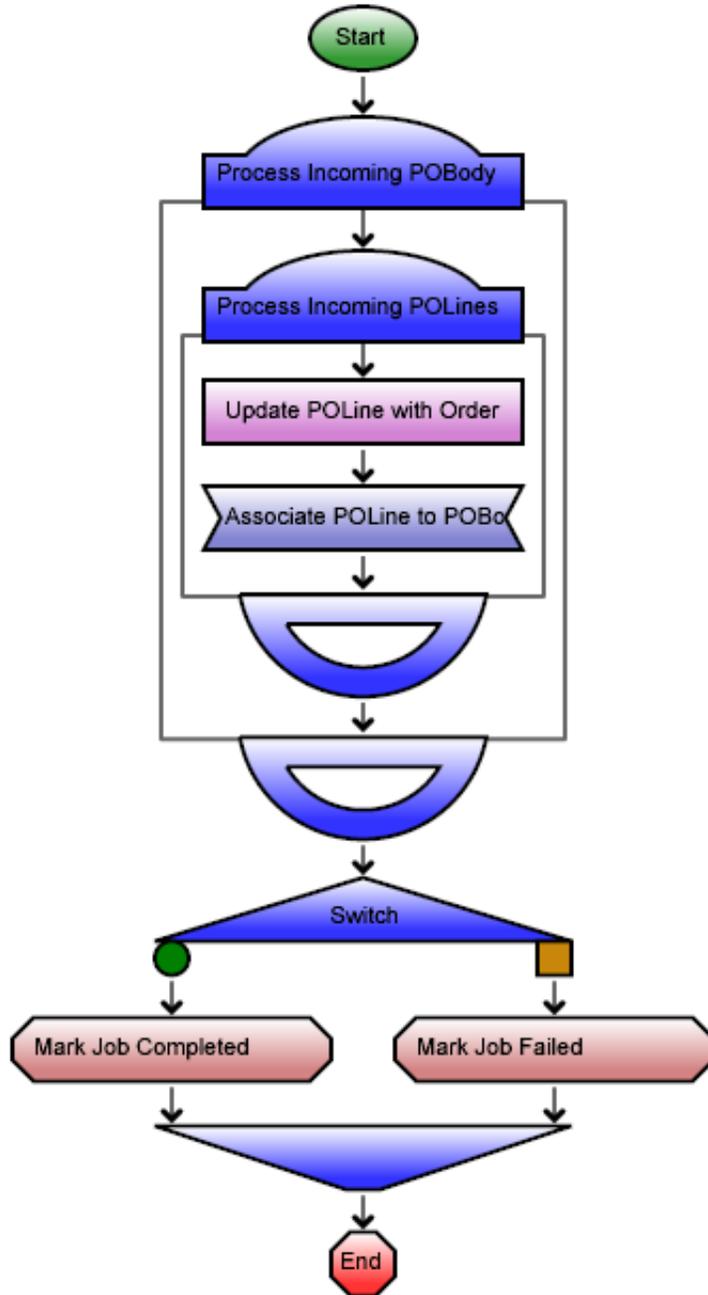
Figure 95. DataConnect example

One thing to notice about each DataConnect task as it appears in the figure is that there is more of it than just its palette shape. The bottom of the DataConnect task reflects the iterative nature of a DataConnect task.

The purpose of the sample workflow shown in the figure is to import a purchase order body and its purchase order line items. The workflow begins by using the first DataConnect task to import the purchase order body. The second DataConnect task imports the line items for each purchase order body. Within the second DataConnect task are tasks performing business logic on each purchase order line item.

The DataConnect task acts in two parts:

- Retrieves records to work on from the appropriate DataConnect staging table.
- As an iterator, creates a new record or updates an existing record for each row and runs the body of the DataConnect task for each row. Other task steps can be positioned within the body of the DataConnect task.

The properties form for a DataConnect task is organized into three sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Module

The value of this property is the module containing the business object DataConnect is to process.

Set the value of this property by clicking the Module icon. Clicking the Module icon causes a list of modules to pop up. Click the module name you want to be the value of this property.

Object

The value of this property is the business object DataConnect is to process. Select the appropriate business object from the drop-down list.

This indicates the DataConnect staging table to pull from and what IBM TRIRIGA business objects will be created or updated by the integration process.

Initial State

For the value of this field select the state in which the record for the business object should be created by DataConnect. The drop-down list displays the valid states for the business object configured in Object.

Keep in mind the record created or updated will not be transitioned to the state, it will be put into that state. If you want the record to be transitioned, use null in the Initial State and use a Trigger Action task within the DataConnect task body to transition the record to the desired state. On update, the Initial State property is ignored.

Use Temporary Data

This property determines whether the DataConnect task uses permanent or temporary data. When the DataConnect task applies the incoming data, if the Use Temporary Data check box is selected, the platform inserts or applies updates to temporary data rather than to the permanent record. In this scenario, within a DataConnect task, referring to the DataConnect task allows access to permanent values and referring to a Get Temp Record task allows access to the incoming values. This enables data validation and cleansing of incoming values within the workflow and even data mapping on update if fields are blank in the staging tables.

If you leave a DataConnect task that is processing temporary data either through a Break task or by running out of iterations, and have not used a Save Permanent Record task, the temporary data is no longer available to

the workflow. A Save Permanent task must be executed prior to leaving the DataConnect task to ensure the data is made permanent.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed

Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Correlation section

The second section of the DataConnect task properties form is labeled *Correlation*. The purpose of the *Correlation* section is to specify where the task should get the correlation number. The correlation number is a secondary key to help with sequencing or to signify child records for a specific parent. The choices for Correlation are as follows:

In-Sequence

Indicates that the correlation on the staging entry should be used in ordering the rows to process and will be used in the Order By in conjunction with the Sequence Number. This is the default and the most common scenario unless you have DataConnect tasks one within another.

Task Step

Indicates that the correlation should come from an enclosing DataConnect task and should be used in the Where clause to determine what entries to

process (Job Number + Correlation Number, ordered by Sequence Number). Using this allows the DataConnect task to get the correlation number from one of the enclosing DataConnect task's data. For example, if you are working on the fourth POBody in the job, you only want the POLines that go with that POBody. The Correlation setting along with the staging table data allow this type of relationship. The drop-down list contains the DataConnect task steps this step is contained within, if any. Only DataConnect task steps are valid for this reference. Select the task step from those listed in the drop-down list.

Transaction section

The third section of the DataConnect task properties form is labeled *Transaction*. The purpose of the *Transaction* section is to specify the scope of the transaction being processed. The Transaction setting only applies to the outermost DataConnect task. If a DataConnect task is nested within another DataConnect task, the first, outer task specifies the Transaction value. The choices for Transaction are as follows:

None Indicates that the record created or updated is committed before the DataConnect task processes the tasks inside the body of the DataConnect task, right after the record is created. No transactions are created; the processing of the block follows standard workflow rules.

Per X Iterations
A new context is created for each X iterations and committed when that number of iterations is complete. The default value is 1, which means the records are committed after each iteration.

All Iterations
A new context is created before the task starts any processing and is committed when all iterations have been completed.

The body of a DataConnect task should not contain User Action or Approval tasks. The User Action task and Approval task require user intervention. DataConnect tasks are intended to be run on non-peak hours to avoid impacting system performance and it is unlikely the required user would be present to respond to a User Action or Approval task.

Variable definition task

The purpose of the Variable Definition task is to define a variable for use in a workflow. Workflow variables can represent a task step instance within a workflow. This includes step values (e.g., sum, count, success), the result record, and the result list.

The properties form for a Variable Definition task is organized into two sections. Here are descriptions of the fields in the first section:

Label This is the label used to identify this task as well as the name of the variable. The label must be unique within the workflow. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Module
The value of this property is the Module with which this variable is

associated. Set the value of this property by clicking the Module icon.
Clicking the Module icon causes a list of Modules to pop up. Click the
Module name you want to be the value of this property.

Object
Set the value of this property to the Business Object within the Module
with which this variable is associated. Select the appropriate Business
Object from the drop-down list. The Business Objects in the list are those
defined for the module.

Value section (variable definition)

The second section of the Variable Definition task properties form is labeled *Value*.
The purpose of the Value section is to specify the source of the initial value for this
variable. The possible values are:

Clear   The initial value of the variable is empty.

From Task
When you select the From Task radio button, the system presents a
drop-down list containing the compatible preceding task steps whose
result type (module and business object) match that of the variable. Choose
a task step from those presented.

From Expression
When you select the From Expression radio button, you use the expression
editor to define how to derive the value for the variable. At runtime, the
system evaluates the expression and returns an array of record Ids. If an
expression does not evaluate to any record or if an error is encountered
when evaluating the expression, an empty array is returned.

Variable assignment task

The purpose of the Variable Assignment task is to assign a value to a variable
within a workflow. Workflow variables can represent a task step instance within a
workflow. This includes step values (e.g., sum, count, success), the result record,
and the result list.

The properties form for a Variable Assignment task is organized into three sections.
Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the
shape in the drawing that represents this workflow task. Use the standards
in "Workflow naming conventions".

Description
A description of this task goes in this field.

Assign variable section

The second section of the Variable Assignment task properties form is labeled
Assign Variable. The purpose of the Assign Variable section is to identify the
variable being assigned a value in this task.

Variable
The value of this property is the variable being assigned a value. Select
from the drop-down list. The variables listed are those defined above this
task in the workflow.

Object Type
>   After you select the variable, the system displays the variable's business
>   object.

Value section (variable assignment)

The third section of the Variable Assignment task properties form is labeled *Value*.
The purpose of the Value section is to specify the value for this variable. The
possible values are:

Clear   The value assigned to the variable is empty.

From Task

>   When you select the From Task radio button, the system presents a
>   drop-down list containing the compatible preceding task steps whose
>   result type (module and business object) match that of the variable. Choose
>   a task step from those presented.

>   The Append Results field makes it possible to build up a result from
>   multiple source tasks. Selecting this check box adds the result (record list)
>   from the source task to any existing values in the variable. If this check box
>   is not selected, the result replaces existing values in the variable.

From Expression

>   When you select the From Expression radio button, you use the expression
>   editor to set the expression to derive the value of the target variable
>   specified in the Assign Variable section. At runtime, the system evaluates
>   the expression and returns an array of record Ids. If an expression does not
>   evaluate to any records, an empty array is returned.

>   When checked, the Append Results field appends the results to the existing
>   variable value. If the Append Results check box is not selected, the result
>   of the expression replaces the existing variable value.

Workflow revisions

A revision of a workflow is automatically created when you select the workflow in
the Workflow Builder and click **Revise**.

You can view a list of revisions for a workflow by selecting the workflow in the
Workflow Builder and clicking **List All Revisions**. The current revision is
displayed at the beginning of the revision list. For each revision, the revision list
also shows the object label, modified date, modified by, and any details such as
whether the revision originated in an object migration (OM) package.

The current revision number, object label, modified by, and modified date for a
workflow are also displayed in the Workflow Properties panel.

When you revise a workflow, the object label is set to ObjectLabelPending until
you publish the revision, where the label changes to In Progress. For more
information on object labels, see *Tracking changes to objects and comparing object
revisions* in the Resources section of Object Labels and Revisions on the IBM
TRIRIGA wiki.

Reverting the current revision of a form to a previous revision
>   You can revert the current workflow revision to a previous revision. In the

list of revisions for the workflow, select the radio button of the revision whose data that you want to make current and select one of the following actions.

Publish
> Select this action to publish the previous revision, where the Object Label, Modified Date, and Modified By values of the selected revision are applied.

Publish with Object Label
> Select this action if you want to modify the previous revision when you publish it. With this action, the Modified Date field contains the date that you publish the previous revision. The Modified By field is your user name; that is, the logged in user who publishes the previous revision.

When you publish the workflow, a new revision is displayed in the revision list and it contains the data of the previous revision that you published. The **Detail** column indicates the origin of the data in the new revision, for example, `From revision 4`.

Comparing workflow revisions
> To compare revisions of a workflow, you must export each revision as text and use a third party comparison tool. See the section, *Exporting workflows as text* that follows.

Exporting workflows as text

A workflow can be exported from the Workflow Builder application to a text file. This text file contains a text-based model of the workflow logic.

About this task

You might find it useful to export two revisions of the same workflow to compare the changes that were made from one revision to the next. You can use a comparison tool to analyze the text files and identify differences between revisions. When you analyze a workflow that is exported as text, you can evaluate the entire workflow. If you analyze a workflow from the Workflow Builder user interface, you can access one task of the workflow at a time.

Procedure

1. Select a workflow from the main **List Template** view, or the **List All Revisions** view of the Workflow Builder application. Only one workflow can be exported at a time.
2. Click **Text Export**. Specify a location and file name to save the file to the system. Individual workflows are saved with a `.cld` file extension.
3. Use a text editor to examine the exported workflow.

Example

In this example, the **My Profile - Create** workflow is shown as it appears in the Workflow Builder application and then as it appears when exported as text.
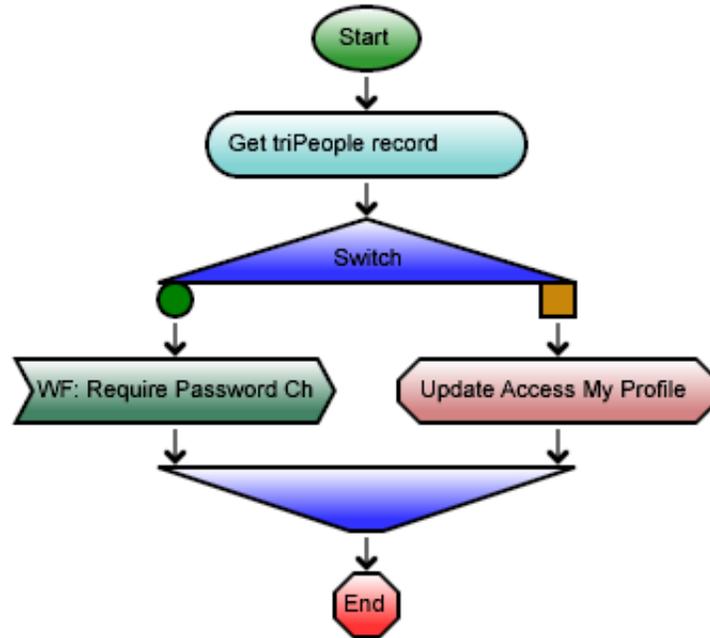
Figure 96. Application Builder workflow

Figure 97. Application Builder workflow properties

```
/*
 *  Generated from System: "My Profile - Create"
 *  Workflow Version: 10
 *  Workflow Status: "Published"
 *
 *  Generator Version: 1
 */

/**
 * Updates the Group and License Access sections
 */
Start ( )
[
    mbo("triPeople.My Profile");
    name("My Profile - Create");
    async("CREATE");
    propagateIntegrationStatus
]

/**
 *
 * @label Get triPeople record
 */
RetrieveRecords GetTriPeopleRecord
[
    mbo(triPeople.triPeople);
    from(Start.record, useAssociation("Associated To", triPeople.triPeople))
]

If (GetTriPeopleRecord.record.Detail.triInitialPwdResetBL == "TRUE" )  {

    /**
     *
     * @label WF: Require Password Change Portal Update
     */
    CallWorkflow     (triPeople.My Profile, "My Profile - Synchronous - Require
Password Change Portal Update")
    [
        mbo("triPeople.My Profile");
        records(Start.record)
    ]
}
Else  {

    /**
     *
     * @label Update Access My Profile
     */
    TriggerAction
    [
        mbo("triPeople.My Profile");
        formulaRecalc(asNeeded);
        action("UPDATEACCESS");
        target(Start.record);
        triggerActionWhen(immediately)
    ]
}

End
```

Figure 98. Application Builder Workflow exported as text

Chapter 8. Temporary data

Most of the data stored by the IBM TRIRIGA Application Platform is permanent data. Permanent data is the data stored in the fields of records that persists in the database until something happens to explicitly change the permanent data or delete the record that contains it.

While someone is editing a record with a form, the system keeps a temporary version of the record's data. Whenever a user switches tabs or does something that can cause a workflow to run, the system updates the temporary version of the record's data.

When someone saves the data they are editing with a form, what happens behind the scenes is that the system copies the record's temporary data and replaces the record's permanent data. If the user closes the form without saving, the system discards the temporary data.

The concept of temporary data is important for writing workflows that enrich the quality of interaction between the IBM TRIRIGA Application Platform forms and users. Some of the useful things a workflow can do with temporary data to enrich the interaction between forms and people are:

- Perform calculations on temporary data and put the results in a form field, so that you have computed values that appear in the form but are not part of the underlying permanent data.
- Based on values in the temporary data, change the colors, fonts, or other metadata used in fields to give people feedback about missing, wrong, or questionable values.
- Validate combinations of values in the temporary data, setting the contents of a form field to contain an explanation of any problems.
- Save the temporary data as permanent data.
- Allow people to explore what-if scenarios by performing calculations on temporary data, restoring the temporary data to a record's permanently stored values if a what-if is rejected, and then finally saving the temporary values as permanent when they are accepted.

Temporary data access by workflow

To process temporary data, you need a workflow (described in "Workflow building") that will do the actual processing and also a sub action (described in "Sub actions") or form component (described in "Form organization") that can be used to launch the workflow.

Workflows launched this way must be synchronous. Synchronous workflows are discussed in "Synchronous versus asynchronous workflows".

In order for a workflow to be able to access temporary data, some of its properties must be specified to have certain values. The properties of a workflow are discussed in "Start task".

To be able to access temporary data, a workflow must have the value of its Concurrence property specified to be *Synchronous* and the value of its Temporary Data property to be *Temporary*.

If a workflow has its properties set this way, then the workflow tasks that are used to get and save temporary data are included in the workflow editor's task palette. The names of these workflow tasks are *Get Temp Record Task* and *Save Permanent Record Task*.

Asynchronous workflows processing data from staging tables with DataConnect task(s) are exceptions to the above. Get Temp Record tasks and Save Permanent Record tasks can occur within a DataConnect task. More information about DataConnect tasks is discussed in "DataConnect task".

Get temp record task

The purpose of the Get Temp Record task is to access the temporary data associated with a record. Temporary data is data that the user has created or modified that has not yet been permanently saved.

This kind of task is in the workflow editor's task palette only if the workflow is synchronous and allowed to access temporary data or if within a DataConnect task.

After a Get Temp Record task has successfully completed, its result list contains a record that contains the temporary data associated with the record that was used to launch the current workflow. In the case of a Get Temp Record task used within a DataConnect task, the result list contains a record created or updated by the staging table input.

Note: If a workflow is defined as temporary and is called through a callable workflow task from an asynchronous workflow, there is no user interacting with the data and therefore there is no temporary data. In this case, all of the tasks defined to do work on the temporary data will do that same work on the permanent data.

Use a Modify Records task (described in "Modify records task") to change the values in the temporary data. Then use a Save Permanent Record task (described in "Save permanent record task") to update the record's permanent data from the temporary data.

If the start record is an in memory record, a Get Temp Record task is ignored.

The properties form for a Get Temp Record task is organized into two sections. Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Records section (get temp record)

The second section of the Get Temp Record task properties form is labeled *Records*. The purpose of the *Records* section is to specify the record whose temporary data is to be fetched.

At the top of the *Records* section are fields used to identify a target record. For the Get Temp Record task, the target record is the record whose temporary data is to be fetched.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record. Although the other two options below are available in the list, this is the option you want to select.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with. Although this looks similar to the like-named field used for the properties of other kinds of tasks, the value of this field is always forced to be the Start task, unless the Get Temp Record task is within a DataConnect task.

> When a Get Temp Record task is within a DataConnect task, the values in the of Task drop-down are the DataConnect tasks above it in the workflow.

Save permanent record task

The purpose of the Save Permanent Record task is to update a record's permanent data with temporary data that was previously fetched by a Get Temp Record task. The Save Permanent Record task is an all or nothing task; it saves everything that is in temporary data. Temporary data is described in "Temporary data". The Get Temp Record task is described in "Get temp record task".

This kind of task is in the workflow editor's task palette only if the workflow is synchronous and allowed to access temporary data or if within a DataConnect task.

The properties form for a Save Permanent Record task is organized into two sections. Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Formulas

Defines the formula recalculation behavior on this task when the record is saved. The default is Disable Auto Recalculation. To change the value, click the recalculate formulas icon.

The three possible values are as follows:

Recalculate as Needed
Recalculates most formulas only if any input value of the formula changed during the task's activities. Extended formulas that contain query and/or association tokens are recalculated on every save.

Disable Auto Recalculation

Extended formulas that contain query and/or association tokens are not recalculated. All other formulas are recalculated only if any input value of the formula changed during the task's activities.

Extended formulas that contain query and/or association tokens can take longer to calculate, so this option is an available performance optimization for isolated use cases.

An example of an appropriate use of Disable Auto Recalculation is within a DataConnect task that is importing a large number of space records. You may wish to delay any floor/space rollup recalculations until after all the space records have been imported.

Recalculate All

Forces every formula to be recalculated during the record's save.

Selecting Recalculate All may slow performance.

An example of an appropriate use of Recalculate All is in a data cleansing workflow that refreshes formula values in the database that are stale for whatever reason. For example, if the definition of a formula changes for a business object, existing records may contain incorrect values as a result of the change. You may wish to use Recalculate All in a maintenance workflow that runs through the records and updates their formula values.

Records section (save permanent record)

The second section of the Save Permanent Record task properties form is labeled *Records*. The purpose of the *Records* section is to specify the record that contains the temporary data.

There is no need to separately identify the related record that contains the permanent data. The connection between the record that contains the permanent data and the record that contains the temporary data is established by the Get Temp Record task when it gets the record that contains the temporary data.

At the top of the *Records* section are fields used to identify a target record. For the Save Permanent Record task, the target record is the record that contains temporary data.

Here are descriptions of the fields:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, the record associated with the task specified by the field to the right of this one will be the target record. Although the other two options below are available in the list, this is the option you want to select.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, the *Event* record that triggered the event is the target record.

> **Assignee**
>> If *Assignee* is selected, the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with. Although this looks similar to the like-named field used for the properties of other kinds of tasks, the value of this field is always forced to be a Get Temp Record task.

Chapter 9. Hierarchies

The IBM TRIRIGA Application Platform allows almost any data organization to be constructed using records and associations as building blocks. The platform provides special support for constructing hierarchies.

Hierarchy modules

A hierarchy is an organization of records that are connected to each other in a special way. One record is chosen to be the root of the hierarchy. Other records are connected directly to the root. Other records may be connected to those. The following figure is an example of a hierarchy organization.
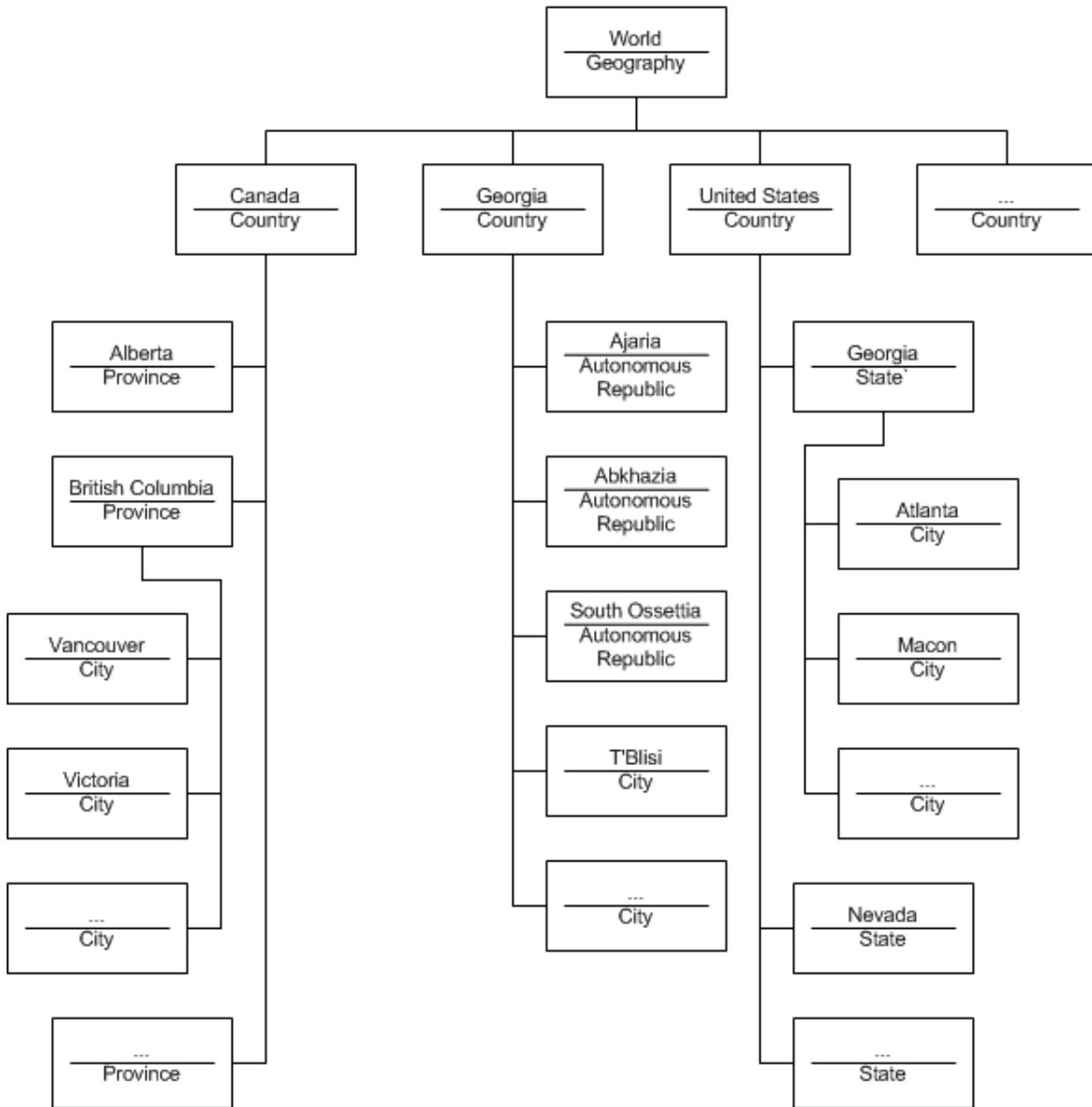
Figure 99. Geography hierarchy organization

There are a few points about the hierarchy that is shown in the figure that are important to notice.

- The hierarchy contains more than one kind of record, but all records must be from the same module. This is common. However, it is not necessary for a hierarchy to contain more than one kind of record.
- A record can appear in only one place in a hierarchy. The same record cannot appear in more than one place in a hierarchy.
- Except for the record at the root of a hierarchy, every record in a hierarchy has exactly one parent. The record at the root of the hierarchy has no parent.
- A record in a hierarchy can have any number of children.
- It may be possible for two records in the hierarchy to have the same name, but records with the same parent cannot have the same name. The hierarchy in the

figure has a country that is named Georgia and a state that is named Georgia. They are represented in the hierarchy by two different kinds of records with two different parents that have the same name.

The figure above is just for explaining what a hierarchy is. The IBM TRIRIGA Application Platform does not display a hierarchy that way. The following figure shows how the platform displays a hierarchy.

Figure 100. Geography as displayed

To improve the performance in hierarchy trees with many children, only the first 1000 children display along with an informational message "1000 results displayed. Use Manager Queries to view more." Child records to the nodes that are not displayed in the hierarchy tree can be added by opening the record from the Master/Detail Query and using the Includes tab. The Includes tab displays the tree from the record being shown.

Hierarchies in every business object

Four hierarchies are built into the IBM TRIRIGA Application Platform that are used to support specific data types of fields:

- Geography
- Location
- Organization
- Classification

The hierarchy that is shown in "Hierarchy modules" is part of the Geography hierarchy. The Location hierarchy is used to describe such things as the location of a building or a particular room in a building. The Organization hierarchy is used to describe organizations and sub-organizations.

A business object field can refer directly to a Geography, Location, or Organization record. Even though they do not appear in the Data Modeler, the IBM TRIRIGA Application Platform supplies one of each of these in the layout for every business object as a non-display field. You can use the relationship that records can have to hierarchical records to manage or organize records. This is discussed in greater detail in the *IBM TRIRIGA Application Platform 3 User Experience User Guide*.

The Geography and Organization hierarchies also can be used with the IBM TRIRIGA Application Platform's security features to control access to specific records. These security features are discussed in "Security".

The Classification data type is based on the classification hierarchy.

Hierarchy creation

It is more common to add new kinds of records to the Classification hierarchy than it is to create an entirely new hierarchy.

Hierarchies are built with business objects in a module that has its Hierarchy Module check box checked. Checking the module's Hierarchy Module check box means that records created from the module's business objects will be used in a hierarchy.

Checking the Hierarchy Module check box also causes the Show Quantity check box to appear. Leave the Show Quantity check box unchecked. You may see it checked in some IBM TRIRIGA modules.

After you finish editing a module's properties, you are ready to start defining business objects in the newly defined hierarchy module. Like any other module, the first business object you must define in a hierarchy module is the base business object that has the same name as the module. A record that is created from the module's base business object will serve as the root of the hierarchy.

All business objects that you use to create records in a hierarchy should be defined as Stand Alone. See "Creating business objects" for more information about stand alone records.

As you specify fields that will be in the records of a hierarchy, pay extra attention to the fields or combination of fields that will be used as the record's name. In addition to the requirement that each existing record created from the same business object has a different name, you also should be sure that the names will be suitable as labels in the hierarchy.

Business object association requirements

In addition to other steps you perform when you define new business objects, there is an extra step to perform if a business object will be used to create records for a hierarchy.

For each kind of record that can appear in a hierarchy, the platform restricts what kinds of records may appear under it in the hierarchy. The way this is controlled is through the business objects used to create the records. The kinds of child records that a record in a hierarchy can have is determined by associations that are defined for the business object that is used to create the record. If an association named Is Parent Of is defined from a business object, records created from the business

object can be in the hierarchy as the parent of a record that is created from a business object on the other side of the association.

On the parent side, the name of the association must be Is Parent Of. The association name Is Parent Of is treated specially. When you specify that the name on the near side an association is Is Parent Of, the Data Modeler forces the name on the far side of the association to be Is Parent Of. It also forces the module that contains the business object on the far side of the association to be the same as on the near side.

Due to the special significance of the Is Parent Of association, it is displayed specially in the Data Modeler's Association List with cyan color coding as shown in the following figure.
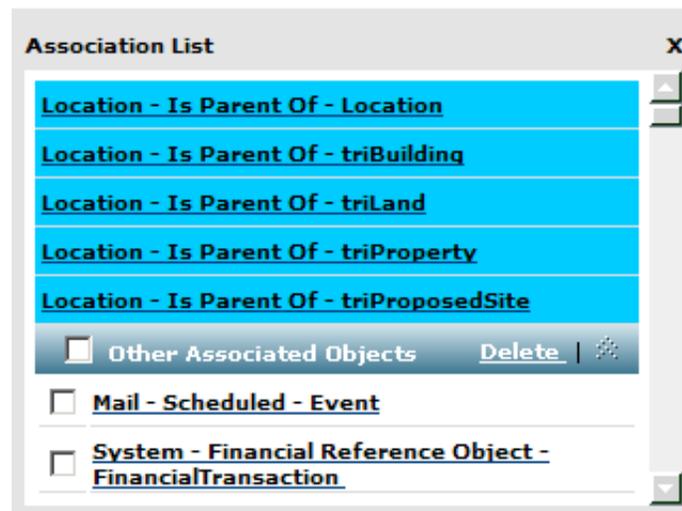

Figure 101. Is Parent Of association

Is Parent Of associations are used to determine the structure of a request central portal section.

Is Parent Of revises a business object

The platform treats associations that are named Is Parent Of specially. Associations with other names can be defined for published business objects because defining them does not change the business objects that the association references.

Because associations named Is Parent Of are treated specially, when you define one, the platform changes the business object on the Is Parent Of side of the association. The changes are needed to account for the additional structure that the new association imparts to a hierarchy.

Rather than preventing you from defining an association that is named Is Parent Of on its near side that also refers to a published business object on its near side, the platform changes the state of the business object from Published to Revision in Progress. After you define an association that is named Is Parent Of, the changes to the hierarchy are not visible outside the Data Modeler until you publish the business object on the near side of the association.

Form hierarchies

The IBM TRIRIGA Application Platform is organized to provide a clear separation between the design of a user interface and the internal organization of records. In keeping with this principle, you need to organize a hierarchy of forms to create a hierarchy of form records.

This hierarchy of forms does not need to precisely mirror the organization of the record hierarchy. If the same kind of record can appear under two different kinds of parents, you can choose to use the same or a different form in each context.

The way the Form Wizard organizes this hierarchy is that for each form that can be used in a hierarchy, you specify what form to use for each kind of child record. If you do not specify a form to create a particular kind of child record under a form, you are not able to create that kind of child from that form.

To specify what forms can be used to create child records from a parent form, use the Form Wizard's Includes/Forms tab, which is shown in the following figure.

Figure 102. Form wizard: Includes/Forms tab

For each business object that can be used to create a child of a record this form is used to edit, you can add a form to the Includes section of the Form Wizard's Includes/Forms tab.

Each row of the Includes section specifies a form that is to be used when you create a child record from a particular business object. You can have one row in the Includes section for each business object that may be used to create a child record under the type of record that the form edits.

In addition to specifying the form that will be used to create child records, the rows of the Includes section also specify the label that will be used in a hierarchy tree to indicate the type of record that a child is.

The other columns in the Includes section are intended to specify the appearance of a child record's label and icon image in a hierarchy tree. At the time of this writing, the platform does not use the values that are specified in these columns.

Example: Biological hierarchy

We explain how to build a hierarchy by working through an example. The example is to create a hierarchy to model the scientific classification of living things (biological taxonomy). The top three levels of this hierarchy are built with records created from business objects named Kingdom, Phylum, and Class.

Begin by creating a hierarchy module named Living Thing. The properties for the Living Thing module look like the following figure.
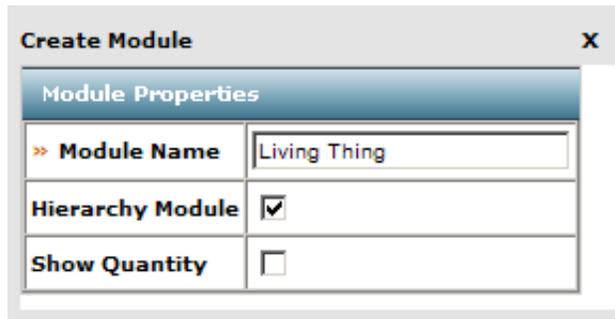
Figure 103. Living Thing module

In the Living Thing module, create stand alone business objects named Living Thing, Kingdom, Phylum, and Class. Each of these business objects has a Text field named Name. In each business object's mapping information, the Name field is identified as containing the name of records that are created from the business object.

The permissible structure of the hierarchy is that Kingdom records can be under a Living Thing record, Phylum records can be under a Kingdom record, and Class records can be under a Phylum record. To specify this structure, specify the associations that are shown in the following figure.
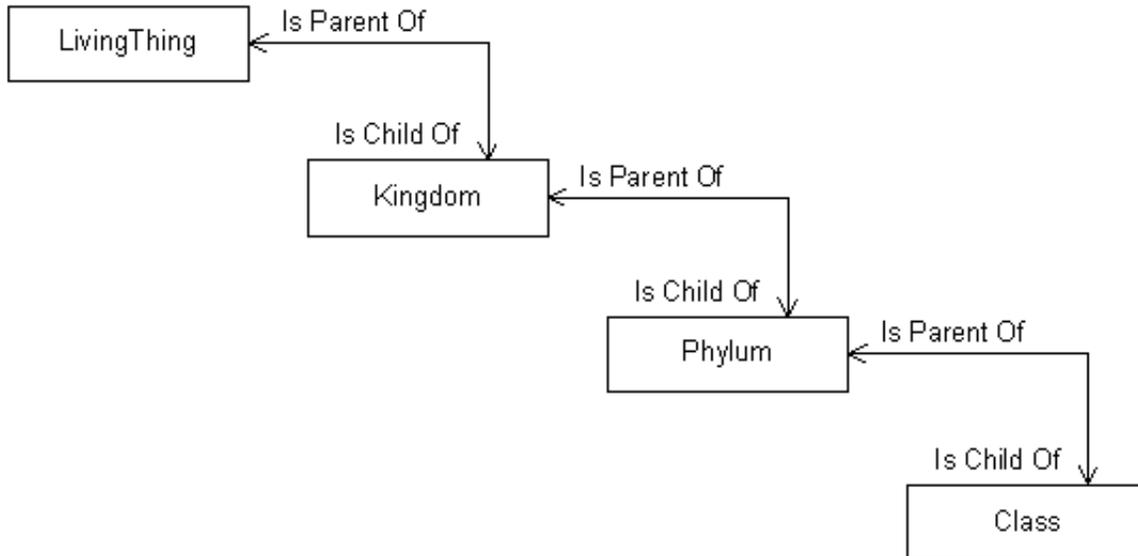
Figure 104. Associations defining structure of Living Thing hierarchy

As you are creating these associations, the Association Properties panel in the Data Modeler looks like the following figure.

Figure 105. Properties for Is Parent Of association

The form for specifying an association's properties treats an Is Parent Of association specially. If you specify that the name on the near side of the association is Is Parent Of, the module of the business object on the other side of the association is forced to be the same as the module on the near side of the association. Also, the name on the far side of the association is forced to be Is Child Of.

After you save an Is Parent Of association, the Associate Business Object form is replaced with an Includes in Hierarchy form that looks like the following figure.

Figure 106. Include properties

After you finish creating the business objects and associations, the next step is to create the forms you will use to edit records in the LivingThing hierarchy. Begin by creating a form to edit Class records, which is at the bottom of the hierarchy.

Phylum records will appear in the hierarchy as parents of Class records, so the next form to create is a form to edit Phylum records. After we have laid out the Phylum form in the Form Wizard's Layout tab, the next step is to specify what form will be used to create child records for a Phylum record.

To specify that the Class form that is used to edit Class records will be used to create and edit records under a Phylum record, navigate to the Includes/Forms tab of the Form Wizard. Initially, the Includes section of the Includes/Forms tab is empty and looks like the following figure.

Figure 107. Empty Includes/Forms tab

To add a form to the Includes section, click the **Add** action on the Includes section bar. When you click the **Add** action, a panel pops up that allows you to select existing forms that can edit records that are allowed to appear in the hierarchy under the kind of record this form edits. The panel looks like the following figure.
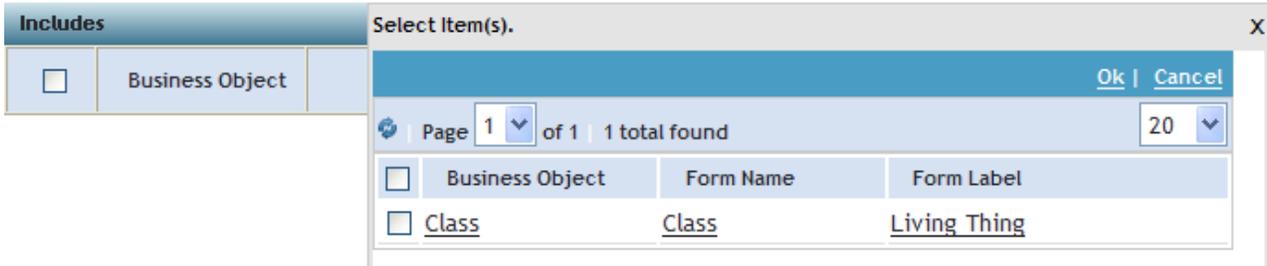
Figure 108. Select items for includes

In order to create records for this hierarchy, you will need to create a couple more items. First, you should create a manager query definition for a hierarchy using the Report Builder. Then you will need to create a navigation item for a master/detail hierarchy.

To create the manager query, navigate to My Reports. Select the System Reports sub-tab. Click New to create a new System Report. Click the Add Business Object link. Select the Living Thing Module, -All- Business Objects, and -All- Forms. Click Ok. Fill the remaining fields as shown in the following figure.
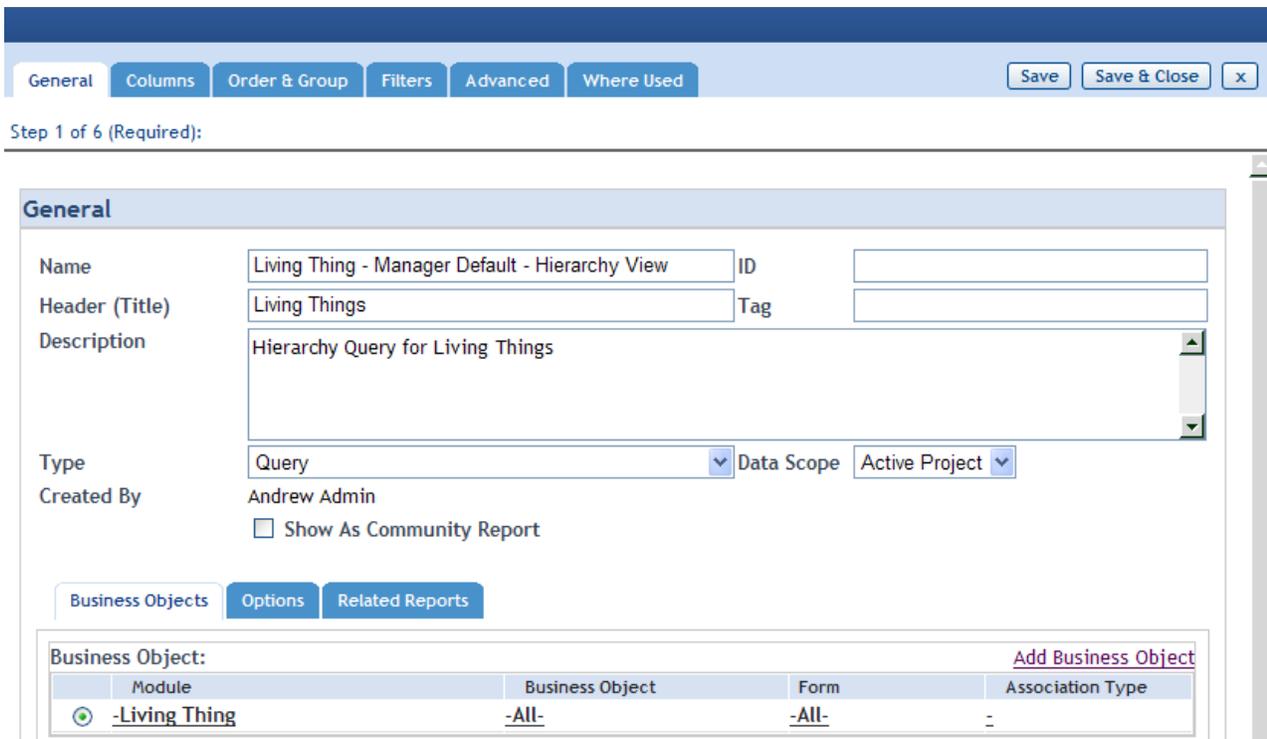
Figure 109. Create system report

Select the Columns tab. Add the Name (triNameTX) field to the Display Columns. Select the Advanced tab. Click the Add action in the Association Filters section. Set the Module to -Any-, Business Object to All, Association Type to Is Parent Of, Reverse Association to No, and Filter Type to Record. Then type $$RECORDID$$ in the Record/Query field, as shown in the following figure. Click OK. Click Save & Close to save the query.

Figure 110. Create association filters

The final step is to add a navigation item to your menu so that you can create new records in your hierarchy. Before you do this, check your menu by going to the People record > Profile tab, as shown in the following figure. This will be the menu, also known as a navigation item collection, to modify. Note: Any change you make to this navigation item collection will affect all other users who have the same navigation item collection set as their menu. Make sure you are working in your own environment, or undo all changes at the end, or create a custom navigation item collection for this example. See the *IBM TRIRIGA Application Platform 3 User Experience User Guide* for more details about navigation collections.

Figure 111. Menu in People record

Go to Tools > Navigation Builder. Select your menu and click Edit as shown in the following figure.

Figure 112. Navigation item collection

Now click Navigation Items Library on the bottom of the page. Click Add to create a new navigation item. Set the Name to Hierarchy - Living Things and Label to Living Things. Now set the Target Type to Master/Detail Hierarchy. Make sure to check the Modify Hierarchy check box. Then set the Hierarchy to Living Thing, the Query Module to Living Thing, and the Report to the query you just defined, as shown in the following figure. Click Save & Close.

Figure 113. Create navigation item

The last step is to add the navigation item to your menu. Select your navigation item on the left panel and Landing Page - Portfolio on the right panel. Click Add to collection. If you now expand Landing Page - Portfolio, the new navigation item is there. Now click Save and Close, as shown in the following figure (the left side of the Navigation Builder).
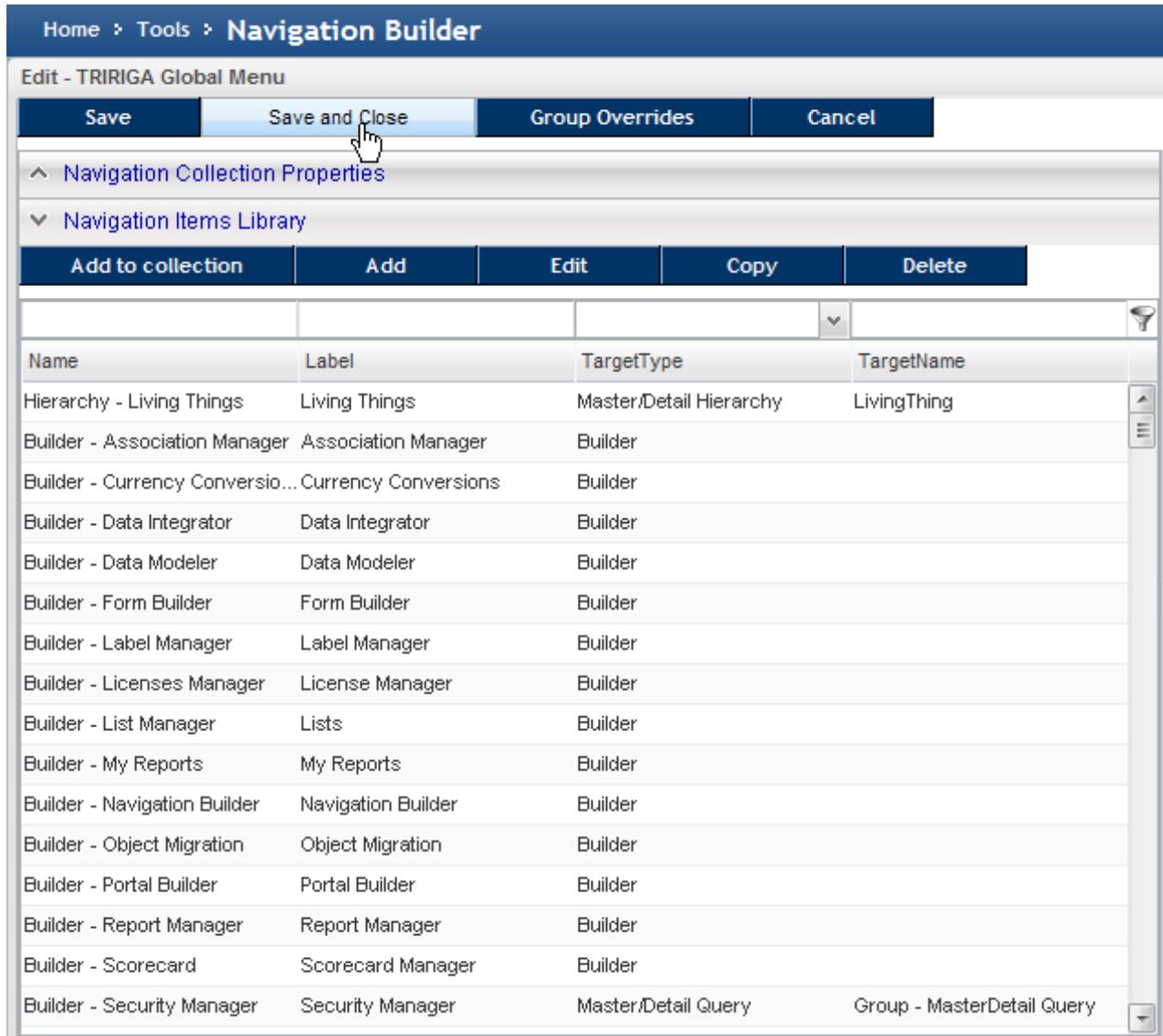
Figure 114. Left side of Navigation Builder

And as shown in the following figure (the right side of the Navigation Builder), to save the changes to your menu. To update your menu, either refresh the browser, or sign out and sign back in.
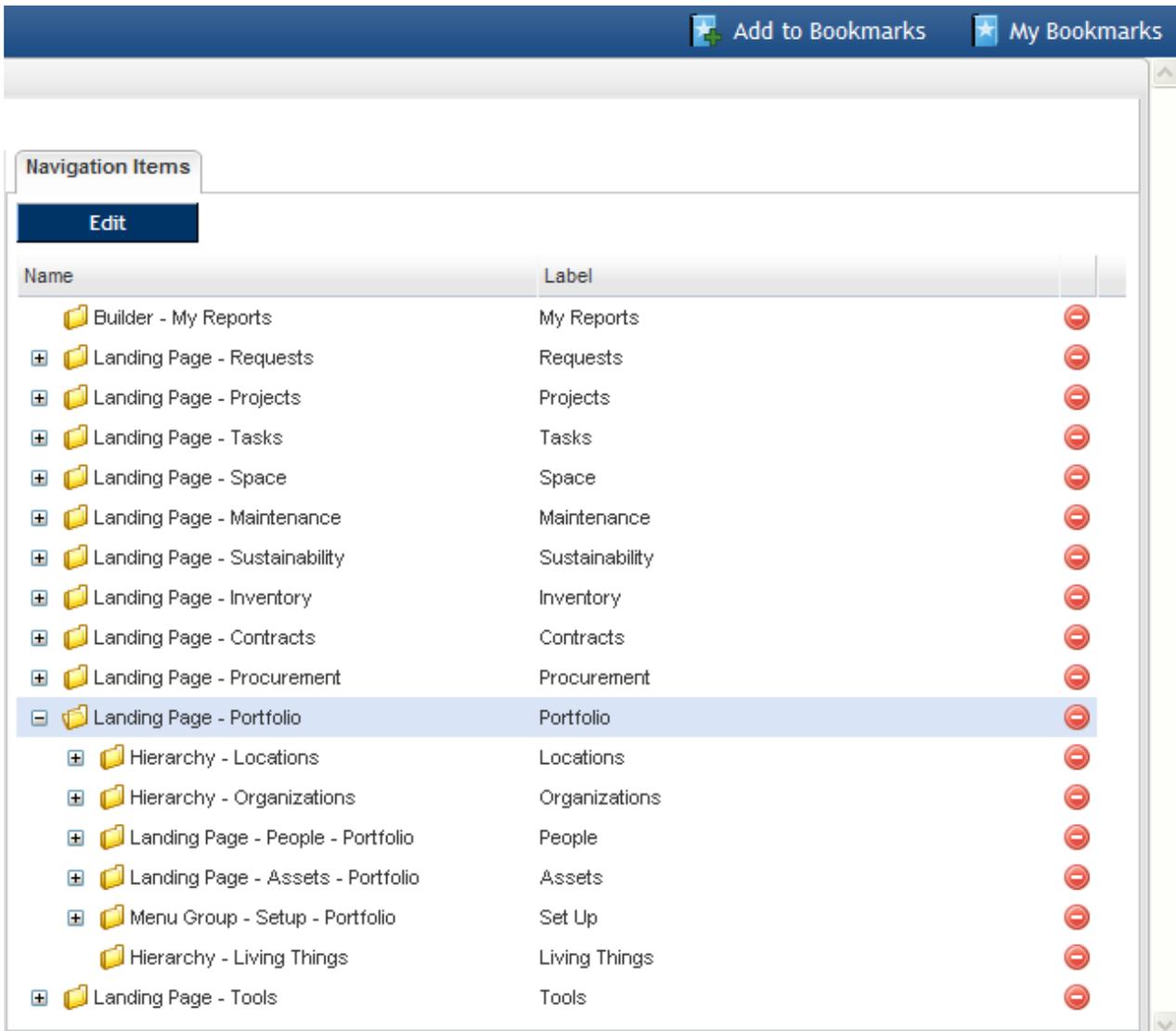
Figure 115. Right side of Navigation Builder

Navigate to Portfolio > Living Things. As shown in the following figure, the Hierarchy is displayed in the left panel. It only contains the record that is the root of the hierarchy. The right panel displays all the records that the root Living Thing is a Parent of, currently none. Note that if you had not defined the query earlier and set it as the query for the navigation item, the right panel would show "No queries defined"; you would still be able to use the hierarchy to create and open records, however.
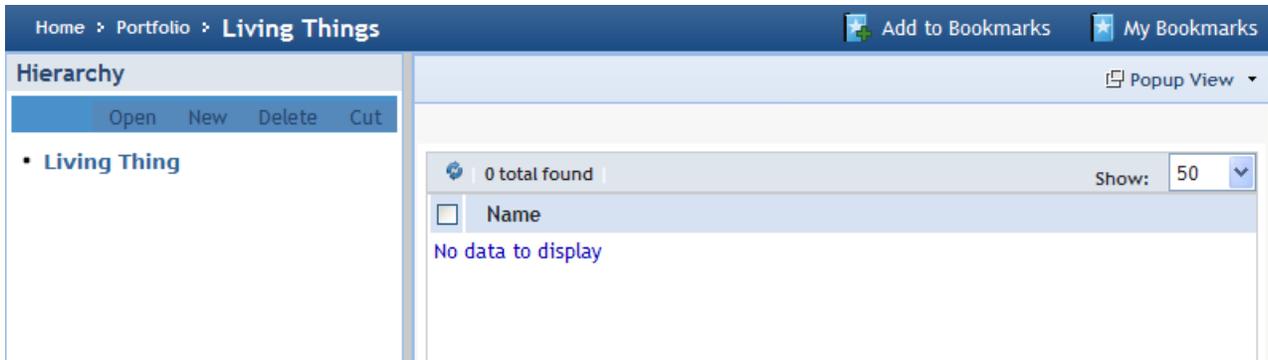
Figure 116. Living Thing hierarchy

The IBM TRIRIGA Application Platform automatically creates the root record of a hierarchy. The way that root records are created is unique. No pre-create workflow is run. Both the name and the state of the record is initially null, yet the record is saved in the database.

If you wanted to edit the record that is the root of the hierarchy, click the **Open** action. You should edit the root record of a new hierarchy if only to just set its name.

To add additional records to the hierarchy, click the **New** action. Clicking the **New** action causes a menu to appear that contains a list of the different kinds of records that can be added to the hierarchy. The menu looks like the following figure.
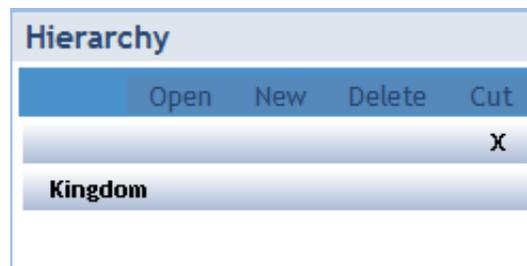
Figure 117. Select type of new record for Living Thing hierarchy

The kinds of records that appear in the menu are determined by the Is Parent Of associations you created in the Data Modeler and the Includes defined in the Form Builder. The LivingThings business object has an Is Parent Of association with only the Kingdom business object, so Kingdom is the only choice in the menu.

At this point, you can click Kingdom in the menu to create a new Kingdom record under the root of the hierarchy. If you change your mind about wanting to create a new Kingdom record under the hierarchy's root, click the x at the top of the list to get rid of the list.

When you click Kingdom, a form appears in the details view for you to enter in the values of the new Kingdom record. After you click the form's **Create** action, the new Kingdom record appears in the hierarchy. Repeat this four times, to create records for all five of the biological kingdoms. The hierarchy now looks like the following figure.
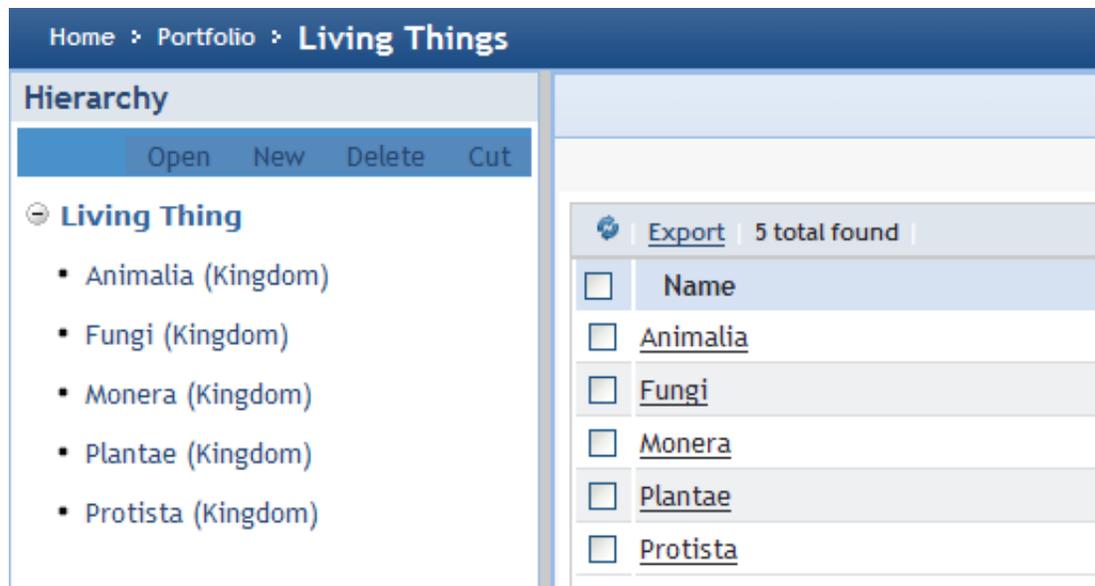
Figure 118. Kingdoms added

You can add Phylum records under the Kingdom records in the hierarchy. To do this, first click the name of the Kingdom record you want the Phylum record to appear under. This causes the Kingdom record to become the selected record. If any record other than the root is selected, its name becomes red.

When you click the **New** action in the Hierarchy panel, it means that you are trying to create a new record under the selected record.

If you make the mistake of putting a record under the wrong member of the hierarchy, it is easily to fix the mistake by following these steps:

- Select the record you want to move to a different place in the hierarchy.
- Click the **Cut** action.
- Select the member of the hierarchy that you want the record to appear under. This has the effect of replacing the **Cut** action with a **Paste** action.
- Click the **Paste** action.

Names in hierarchies

The IBM TRIRIGA Application Platform requires every record created from the same business object to have a different name. There is an additional naming constraint for records in a hierarchy.

The children of a parent record in a hierarchy are required to have different names, even if they were created from different business objects. This constraint ensures that there is never any confusion about which record is being referred to in a hierarchy when it is identified by its name and path.

Hierarchy data auto-population

In the Data Modeler, among the properties of each field in a business object is a check box that is labeled Do not Auto Populate. If this check box is checked, it disables a feature of the IBM TRIRIGA Application Platform called auto-populate.

The auto-populate feature supplies initial values for fields in new records that do not have a specified default value. The auto-populate feature uses as a source of default values the triPeople or other kind of record that describes the logged in person. If a field in a new record has the same name as a field in record that describes the logged in person, the value of the like-named field is copied to the field in the new record.

If a new record is created as part of a hierarchy, its parent is used as a second source of data for auto-populate. If the record that describes the logged in person does not have a field with the same as a field in a new record in a hierarchy but its parent has a field with the same name as a field in the new record, the value is copied from the field in the parent to the corresponding field in the new record.

If you do not want the auto-populate feature to set the initial value of a field, when you are defining the field be sure to check the Do not Auto Populate check box.

Report hierarchies

The Report Manager does not allow explicitly defining reports on types of records that have a parent-child relationship in a hierarchy. However, it does allow creating such a report by treating the connection between a parent and its children as an association named Is Parent Of.

Chapter 10. Calendar and time-based events

You can make an application do things based on dates and the passage of time. You can also use a calendar to track things that have happened and will happen to records.

Events

The discussion of calendar and time based events begins with definitions of event and of calendar. An event is a period with a start time and an end time that serves a practical purpose. A meeting is an example of an event, as is a work task. A calendar is any logical grouping of events.

Event interfaces

Any business object with specific fields can be considered an implementation of an event interface. The IBM TRIRIGA Application Platform offers several services for any business object satisfying the event interface.

An event implementation is any business object that has a specific set of business object fields. All other aspects of the business object can be application-specific, for example control numbers, other attributes of the event, any state transitions, and forms. The business object fields needed for an event implementation are the following:

Table 25. Business object fields needed for event implementation

Name	Business Object Field Name	Description
START_DT	triStartDT	This is the start time of the event. In the case of a recurring event, this is the start time of the first occurrence of the event.
END_DT	triEndDT	This is the end time of the event. In the case of a recurring event, this is the end time of the first occurrence of the event.
QUERY_END _DT	triEventQuery EndDT	When querying events for any application, the platform filters the result's date range based on this end time. This is especially important for recurring events to avoid the platform pulling recurring events into memory that have long since ended.
RRULE	triRecurrence RuleTX	For recurring events, this holds the iCal-compliant recurrence rule (for example, RRULE:FREQ=WEEKLY;BYDAY=WE). iCal or iCalendar is Internet Calendaring and Scheduling Core Object Specification, an industry-standard way of representing a calendar or portion of a calendar such as a calendar event.

Table 25. Business object fields needed for event implementation (continued)

Name	Business Object Field Name	Description
TZNAME	triRecurrence RuleTimeZoneCL	For recurring events, this identifies the time zone behavior the pattern should follow. This is critically important in the case the event contains an RRULE that generates dates on both sides of a time zone shift (for example, both in Standard Time and Daylight Saving Time). This field should be modeled like the triTimeZoneCL field, a Classification field referencing values from the \Classifications\Time Zones part of the Classification tree.
RECURRENCE _ID	triRecurrence IdTX	For exceptions to a recurring event, this holds the ID to the record that defined the recurring event.
EXDATE	triRecurrence ExceptionTX	For recurring events, this holds the iCal-compliant exception dates.
NAME	triReporting NameTX	This holds the name to be presented to the user in user interfaces. No platform business logic is based on this value. This value is used for display purposes only.
ALL_DAY_FLAG	triAllDayBL	This indicates whether or not the event should be represented as an all day event in user interfaces. No platform business logic is based on this value. This value is used for display purposes only.
FREE_BUSY _TYPE	triFreeBusy TypeLI	This indicates how the event should be rendered in user interfaces. No platform business logic is based on this value. This value is used for display purposes only.
BACKING _EVENT_ID (Optional)	triBackingEvent IdTX	This is a locator field that references the event that this record represents. This can be useful if, for example, the user interface (UI) is working with a transient record that is backed by a different record for persistence. In that case, the backing record would be referenced by this field. This field is optional, meaning that the platform assumes there is no backing event in the case that the business object does not have this field or the field exists but does not have a value.

To help explain how the data is persisted in these fields, look at the sample data in "Business object properties needed to configure a record's calendar". In this example, Pacific represents (GMT -8) Pacific Time (US, Canada); Tijuana [US/Pacific]. The date-times represented in the figure are not exactly how they are persisted. In this representation of the sample data, they are in a more readable format.

SPEC_ID	START_DT	END_DT	QUERY_END_DT	RRULE / TZNAME	RECURRENCE_ID	EXDATE	Description
1	3/2/2011 9AM	3/2/2011 10AM	3/2/2011 10AM				Single-occurrence event
2	3/3/2011 12PM	3/3/2011 1PM	3/5/2011 1PM	RRULE: FREQ=DAILY; COUNT=3 / Pacific			Recurring daily event that occurs 3 times
3	3/9/2011 4PM	3/9/2011 5PM	3/13/2011 5PM	RRULE: FREQ=DAILY; COUNT=5/ Pacific		EXDATE: 3/10/2011 4PM; 3/11/2011 4PM	Recurring daily event. The recurrence rule says it will occur 5 times, but there are two exception dates that will cause the 2nd and 3rd occurrences to not occur.
4	3/14/2011 12PM	3/14/2011 1PM	3/16/2011 1PM	RRULE: FREQ=DAILY; COUNT=3/ Pacific		EXDATE: 3/15/2011 12PM;	Recurring daily event. The value in EXDATE along with the data in record #5 means that the second occurrence will occur an hour late.
5	3/15/2011 1PM	3/15/2011 2PM	3/15/2011 2PM		4		This is an exception to the second occurrence of the recurring event defined in record #4.
6	3/23/2011 8AM	3/23/2011 9AM		RRULE: FREQ=WEEKLY; BYDAY=WE/ Pacific			Recurring weekly event, occurring every Wednesday with no end date.

Figure 119. Event sample data

Platform persistence support

In the calendar server, the platform understands recurrence rules and recurrence exceptions. The platform includes formula functions that application builders can use to generate the appropriate data for the event fields.

Table 26. Platform persistence support

Name	Platform Persistence Support	Description
START_DT		
END_DT		
QUERY_END_DT	createQueryEndDateFromEvent (eventRecordId)	Utilizing the START_DT, END_DT, and RRULE of an event record, the platform returns the date-time to be populated in the QUERY_END_DT field.

Table 26. Platform persistence support (continued)

Name	Platform Persistence Support	Description
RRULE	getRruleFromEventUi (eventUiRecordId) applyRruleToEventUi (rRule,eventUiRecordId)	Although RRULE is the format understood by the platform at runtime, an application can expose recurrence management in a more structured way. The getRruleFromEventUi() and applyRruleToEventUi() functions get the RRULE in to and out of the more structured, user friendly data structure. This is discussed in "Recurrence rules for user interface".
TZNAME		
RECURRENCE _ID		
EXDATE	createRecurrenceExDate (contextRecordId,associationName, dateTimeFieldName)	Although EXDATE is the format understood by the platform at runtime, an application builder can expose exception management in a more structured, usable way. The createRecurrenceExDate() function constructs an EXDATE from the more structured representation. This is discussed in "Exception dates in recurring events".
NAME		

Platform formula functions are described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

Recurrence rules for user interface

An application exposes recurrence management in a much different way than it is persisted on an event record. The user inputs are structured as independent fields, yet the persistence format holds the entire recurrence rule in a single string. The following figure shows an example of an application-developed user interface presenting a weekly, every Monday with no end, recurrence rule for an event:

Figure 120. Recurrence rule example

To move data in to and out of this RRULE string, the platform uses formula functions that have knowledge about the structured fields that an applications can present in a recurrence rule form. The following is a list of fields the platform reads to, writes from, or both, in the createRruleFromEventUi() and applyRruleToEventUi() functions.

Table 27. Business object fields related to the recurrence rule

Business Object Field Name	Description of valid values
RecurrencePatternType	DAILY/WEEKLY/MONTHLY/YEARLY
DailyRecurrenceDays	Number representing the number of days between occurrences of a daily event.
WeeklySunday WeeklyMonday WeeklyTuesday WeeklyWednesday WeeklyThursday WeeklyFriday WeeklySaturday	Boolean representing whether the weekly event should fall on this day.
WeeklyRecurrenceWeeks	Number representing the number of weeks between occurrences of a weekly event.
MonthlyWeekOfMonth	Number representing the week of a month on which an occurrence should occur for a monthly event. Negative numbers represent, for example, the second-to-last week of a month.
MonthlyDayOfMonth	Number representing the day of a month on which an occurrence should occur for a monthly event. Negative numbers represent, for example, the second-to-last day of a month.
MonthlyRecurrence Months	Number representing the number of months between occurrences of a monthly event.
YearlyDayOfMonth	Number representing the day of the month on which an occurrence should occur for a yearly event. Negative numbers represent, for example, the second-to-last day of a month.
YearlyWeekOfMonth	Number representing the week of a month on which an occurrence should occur for a yearly event. Negative numbers represent, for example, the second-to-last week of a month.
YearlyMonth	Number representing the month of a year on which an occurrence should occur for a yearly event.
YearlyDayOfWeek	Number representing the day of a week on which an occurrence should occur for a yearly event. Negative numbers represent, for example, the second-to-last day of a week.
EndDate	The end date beyond which no more occurrences should occur.
NoOfOccurrencesBefore End	Number indicating the number of occurrences that should occur before the recurring event is complete.
NoEndDate	Boolean indicating whether this event has no end date.

When a record is passed in to the createRruleFromEventUi() function, the platform reads the recurrence data from the input fields in order to construct an iCal-compliant RRULE string.

When an RRULE string and a record is passed into the applyRruleToEventUi() function. the platform deconstructs the RRULE and applies it appropriately to the input fields to enable an application to render the RRULE in a structured format.

Exception dates in recurring events

Exceptions are occurrences of a recurring event that vary from the originally defined recurrence rule in any way. To give the platform the visibility it needs into recurring event exceptions, the an application builder must do the following:

- Along with each recurring event record, maintain an associated set of records where each record has an exception date for the recurring event. The association string between the records must be Has Recurrence Exception and the field on the associated record must be named triEventExceptionDT. An exception date is the original start time of the occurrence that was modified or deleted from the series.
- At the save of a recurring event, populate the event's triRecurrenceExceptionTX field with the result of the createRecurrenceExDate() function.

At runtime, when createRecurrenceExDate() is called, the event's record ID is passed in. The platform then iterates through the associated records and generates an iCal-compliant EXDATE string representing the start time of each recurring event exception. If the resulting string is greater than 1000 characters, then the platform instead returns a single character, `+'. The plus sign indicates to the platform at runtime that the exception dates for this recurring event cannot be found in the triRecurrenceExceptionTX field. Instead, the platform must query the associated exception date records. The iCal-compliant EXDATE is generated at runtime in this case. This design supports an unlimited number of exception dates.

In terms of application data modeling, the recurring event exception can be the record associated to the recurring event with Has Recurrence Exception. However, this may only be sufficient to model the occurrences of the series that were modified. It is not a good way to model the occurrences of the series that were deleted. To handle those occurrences, use a business object to store just the deleted occurrences. For the purposes of generating an EXDATE, it does not matter what business objects are associated to the recurring event as long as 1) they are associated via the `Has Recurrence Exception association string and 2) the original start time of the occurrence modified or deleted exists in a field named triEventExceptionDT.

Availability of recurring events to users

No background agent is needed to roll occurrences of a recurring pattern in order for users to interact with those occurrences. The calendar server makes any occurrence immediately usable for UI interactions and resource availability interactions.

The following figure shows an example of a recurring event created to start in 1920:

SPEC_ ID	START_DT	END_DT	QUERY_ START_DT	QUERY_ END_DT	RRULE	RECURRENCE _ID	EXDATE	Description
6	3/2/1920 10AM	3/2/1920 11AM			RRULE: FREQ=WEEKLY ; BYDAY=WE			Recurring weekly event, occurring every Wednesday with no end date.

Figure 121. Example of recurring event starting in 1920

Immediately after a user creates this event, that user or any other user can begin viewing and interacting with occurrences of the event in 2011 or whenever. And if this event is designed to make a resource busy, the system is immediately able to calculate availability in 2011 or whenever.

The following figure shows a Calendar view of the system derived occurrences in April 2011 for the recurring event in the previous figure that started in 1920:
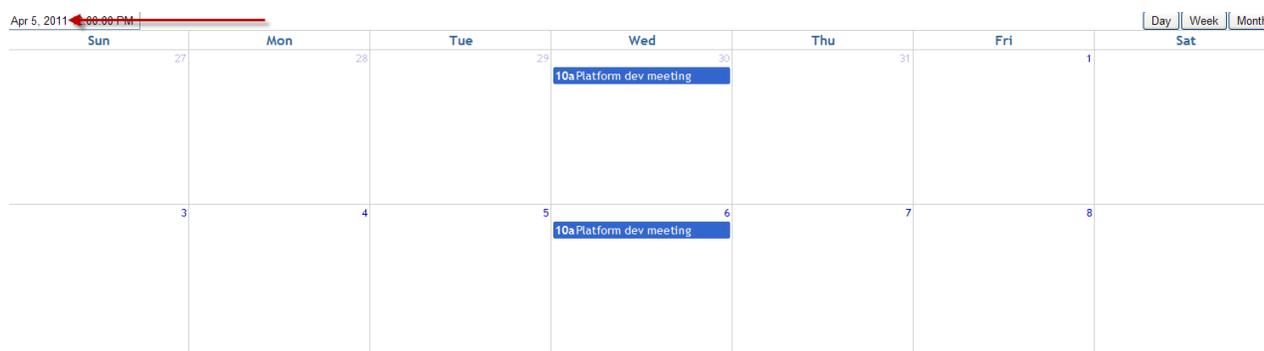
Figure 122. Example of recurring event calendar

Recurrence exceptions

When a recurring event does not conform to the RRULE, you use the getFirstDateTimeMatchingRrule(startDateTime, rRule) function. This function returns the first date time matching a particular recurrence rule and start date-time.

Example: Platform and application communication for recurring events

Occurrences of a recurring event do not necessarily exist in the database. However, applications need to have a handle to those occurrences for any number of business processes.

Perhaps this need is best described by example. Consider the recurring event described in the prior section. It was a recurring weekly event starting in 1920 and having no end date. When the user views the April 2011 calendar UI, they need to be able to remove the occurrence on April 6th, 2011. Since the platform only understands event at an abstract level, then it is the application that handles the business logic around reacting to that user action. For example, there may be food service to be cancelled and notifications to be sent.

An application can see the event details with the triCalendarEvent platform business object. The platform uses the triCalendarEvent business object to pass

event occurrence data to applications. The triCalendarEvent business object resides in the System module. The following table shows the fields in triCalendarEvent and what they mean:

Table 28. Business object fields in triCalendarEvent

Business Object Field Name	Description
triEventIdTX	This is the ID of the affected event. If the event is for a non-exception occurrence of a recurring event, this field has no value. In all other cases, this ID is the smart object ID of the event record.
triRecurrenceIdTX	In the case the affected event is an occurrence (exception or non-exception) of a recurring event, this is the smart object ID of the original recurring event.
triStartDT	This is the start date-time of the affected occurrence.
triEndDT	This is the end date-time of the affected occurrence.

Platform features use this business object to communicate occurrence information from the platform to an application. Here are some examples:

- If a user takes action on an occurrence in a Calendar UI, the platform communicates that user action to an application by way of workflow input parameters and a triCalendarEvent record. The following figure shows a workflow variable being assigned from an expression:
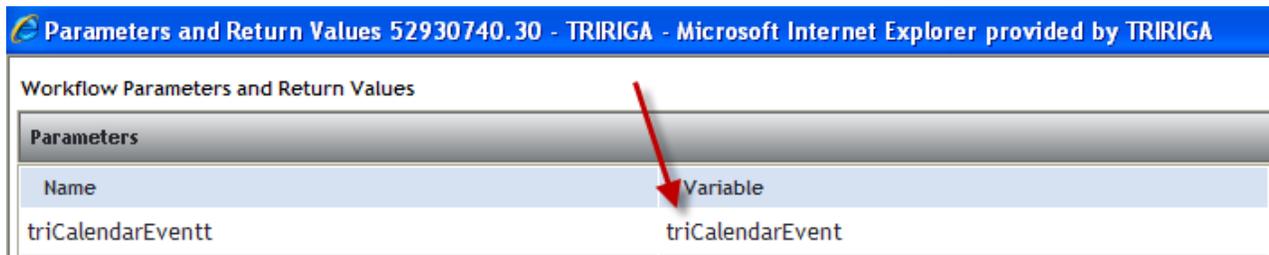
Figure 123. Example of using parameters and return values

- Applications can use formula functions to request occurrence information for events that do not necessarily exist in the database. One such example is the getEventsForResource() function which, given a resourceId and a few other parameters, returns a collection of triCalendarEvent records that can then be processed in a workflow.

Legacy calendar and time-based events

The IBM TRIRIGA Application Platform has the ability to do things based solely on the passage of time. It is possible to schedule an event to happen at any future point in time. When an event happens, it can cause a workflow to be launched.

Because events and other things can happen to records at different points in time, the IBM TRIRIGA Application Platform associates a calendar with records to help keep track of what has happened or will happen to a record and when.

Business object properties needed to configure a record's calendar

Although every record has a calendar associated with it, to see a record's calendar in the user interface, the business object it was created from must be configured with its *Has Calendar* check box checked. The properties of business objects, including the *Has Calendar* check box, are discussed in "Business object properties".

If the *Has Calendar* check box for a business object is checked and the Show Calendar check box is checked for a form, the form will be displayed with a tab named Calendar that otherwise would not appear.

You should also build a tab named triCalendarDetails in your form. The purpose of a *Calendar Details* tab is to specify details that are needed to schedule events for the resource that a record represents. The *Calendar Details* tab looks like the following figure.

Figure 124. Calendar Details tab

The system does not depend on this tab existing in your form to function properly. Instead there are two things you must add to your business object in the Data Modeler. For the scheduling engine to work properly you should add a field to your business object named triTimeZonesCL. You also should make an association to the *triCalendar* business object in the triSetup module with the association name Has Calendar and reverse string of Calendar For. This field and association are not added automatically.

The *Time Zone* field (triTimeZonesCL) is used to specify the time zone that is used to schedule the resource represented by the record.

Another field used if working in conjunction with the Reserve product is the Reservable flag. If the *Reservable* (triReservableBL) check box is checked it means that the resource can be reserved for an event. This is normally checked for things like conference rooms and audio-visual equipment, but not for people.

If you want the calendar to include events from an external Microsoft Exchange server or a Lotus Notes server that the IBM TRIRIGA Application Platform environment has been configured to work with, specify the appropriate login information in the *Mail Server Login* field (triExternalLoginTX).

Note: If using Microsoft Exchange 2007, be sure to configure Plain text logon (Basic authentication) in Microsoft Exchange > Server Configuration > Client Access > IMAP4 Properties > Authentication tab.

The *Availability Calendar* section is used to specify when the resource is available. This section can reference multiple *Calendar* records. If there are multiple *triCalendar* records, the resource is available at any time that any of the calendars say it is available. The Query section here is based on the association named Has Calendar.

The information about when a resource is available is in a separate record because it is usually the case that availability of multiple resources follows the same pattern. For example, if the resource is a person, it will usually be the case that multiple people work the same hours. Putting this common information in a shared place makes it easier to maintain.

Calendar records are located in Tools > System Setup > General > Calendars.

The Scheduling Assumptions section identifies the working hours per day, working days per week, and working days per month for resources using this calendar.

The *Working Hours* section is used to specify ranges of time that resources are available during days of the week. When you select the Quick Add action to add an entry to this section, an open line appears at the top of the section; enter the new day, start time, and end time. Click the down-arrow, up-arrow, or circle icon in the column header to sort the information shown.

The *Non-Working Events* section is used to specify individual days on which resources are not available. In the example shown, it is used to indicate non-working holidays for people. When you select the Quick Add action to add an entry to this section, an open line appears at the top of the section; enter the date and the name of the new event. Click the down-arrow, up-arrow, or circle icon in the column header to sort the information shown.

The *Calendar* tab is shown automatically in your form if the Show Calendar option is selected. In addition to just showing dates, it shows scheduled events and three numbers. The three numbers are labeled P, S, and A. This is what they mean:

P   The planned number of hours of availability.

S   The overall number of hours that at least one event is scheduled. If events overlap in time, the overlapped period of time is counted just once, no matter how many events are scheduled at the same time.

A   The remaining available time.

The three numbers have this relationship: A = P - S

There are four buttons to the right of the year that allow you to select the view that you want. The buttons are:

Y   Press this for a year view.

Q   Press this for a quarterly view.

M   Press this for a monthly view.

W   Press this for a weekly view.

Everything we have looked at doing with a calendar so far is done manually by a person interacting with the user interface. However, there is no entirely manual way to add events to a calendar. At least one step involved in putting an event in a calendar must be done by a workflow.

Legacy scheduled events

Before we discuss the details of putting events in calendars, it is helpful to explain just what we mean by an event. When we are talking about events that can appear in a calendar, we are talking about scheduled events. A scheduled event begins at a particular time. It has a duration. After a scheduled event begins and its duration has passed, the scheduled event ends.

A *Scheduled Event* record is used to represent a scheduled event in the IBM TRIRIGA Application Platform. To be useful, *Scheduled Event* records should not be created directly. Instead, they should be created indirectly by a *SCHEDULE* action performed on an Event record. The form for editing Event records does not have an action in its menu bar for performing a *SCHEDULE* action on an Event record.

At least three workflow tasks are needed for a workflow to create *Scheduled Event* records. The first task is a *Schedule* task. A *Schedule* task does not create *Scheduled Event* records. What it does is create an *Event* record. The *Event* record describes one or more scheduled events.

After the *Event* record is created, one or more workflow tasks are needed to set the fields in the *Event* record.

The last workflow task needed is a *Trigger Action* task to perform a *SCHEDULE* action on the *Event* record. This causes the *Scheduled Event* records to be created.

The Schedule workflow task is described in "Schedule workflow task".

The *Event* and *Scheduled Events* business objects are in the *Mail* module.

Setting the fields of the Event record is described in "Fields in legacy event records".

Trigger Action workflow tasks are described in "Trigger action task".

What happens after a *SCHEDULE* action is performed on an *Event* record is described in "Scheduled events".

Manual event scheduling with workflows

It is possible to manually enter the information needed to create *Scheduled Event* records. This works by having people use the *Event* form to create an *Event* record and then using an action on another form to launch a workflow that performs a *SCHEDULE* action on the *Event* record.

Instead of designing applications to create *Scheduled Event* records this way, use the *Event* form because it offers more options for the scheduling and recurrence of events. Some of the options may be inappropriate or confusing for some applications.

To avoid having to train people about event scheduling options they do not need and to avoid people choosing event scheduling options that are inappropriate for an application, provide an application-specific form.

Here is an outline of the things involved in manually entering event scheduling information:

At design time:

Define an association between the *Event* business object and the business object used to create the records for which you want to create *Scheduled Event* records. Both sides of the association should be named *Scheduled*.

To the form used to edit the records that events should be scheduled for, you must add a way to create and edit an associated *Event* record. The simplest way to do this is to add a single-record smart section to the business object that underlies the form.

You will need a way to perform a *SCHEDULE* action on the associated *Event* record. The simplest way to do this is to write a one task synchronous workflow that is launched from a form and performs a *SCHEDULE* action on the associated Event record.

At runtime:

The person who wants to associate *Scheduled Event* records with another record creates and edits the associated *Event* record, specifying when the event should happen, the event's duration, if the event should be repeated and when the event should be repeated.

When the scheduling information is correctly entered, the person must click whatever has been provided to perform a *SCHEDULE* action on the *Event* record. Performing the *SCHEDULE* action on the *Event* record causes the *Scheduled Event* records to be created and associated with the same record associated with the *Event* record.

Having outlined these activities, we describe some of them in greater detail.

Use the Association Manager to define the association named *Scheduled* between the *Event* business object and the other business object. The associations and the Association Manager are described in "Association definition".

Smart sections are described in "Record organization".

If the *Event* record is accessed through a single-record smart section, the simplest way to provide an action for performing a *SCHEDULE* action on *Event* records is to add the action to the smart section in the form that is used to access the single-record smart section.

It is easiest to first create the simple workflow that will perform a *SCHEDULE* action on the associated *Event* record before adding an action to a form to launch the workflow. In the Workflow Editor, the workflow could look like the following figure. Like all workflows, this workflow has a Start task and an End task which do not do anything. Between them is a Trigger Action workflow task that triggers the *SCHEDULE* action.
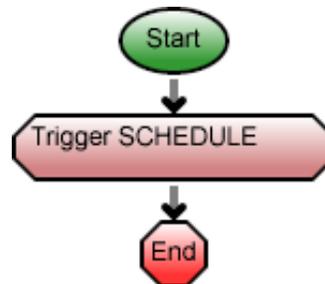
Figure 125. SCHEDULE workflow

The properties for the workflow's Start task will look like the following figure. The most important thing to notice about the settings in the following figure is that the *Module* and *Object Type* fields specify the business object used to create records that contain the single-record smart section.
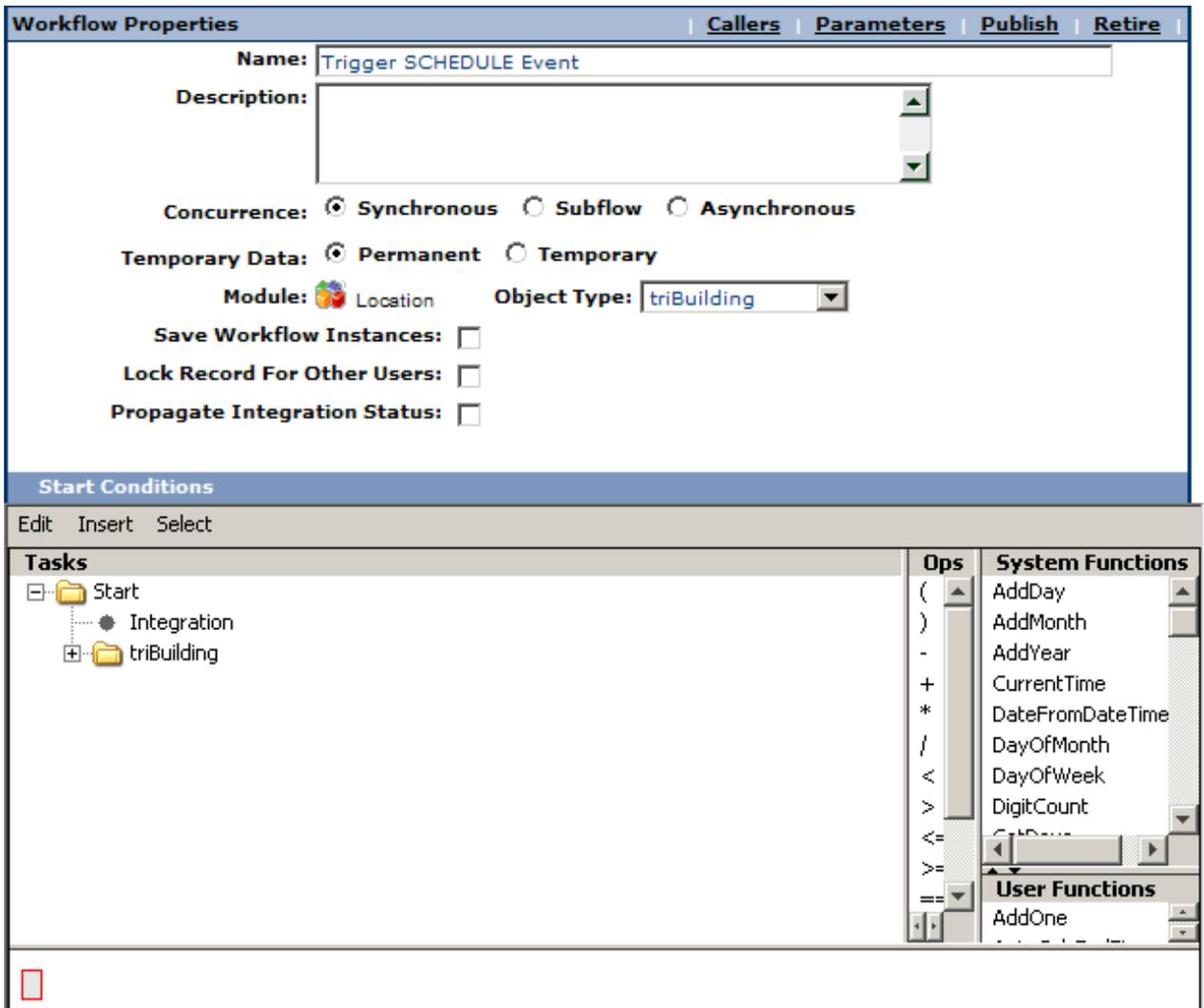
Figure 126. SCHEDULE workflow properties

The properties for the *Trigger Action* task labeled *Trigger SCHEDULE* are shown in the following figure. Notice the following things about the settings:

- The action that the task will perform is specified in the *Action* field as *SCHEDULE*. It will perform the *SCHEDULE* action on the *Event* record referenced by the *Event* single-record smart section.

- The *Records* section of the workflow task's properties specifies that to find the record to perform the action on, the task should begin with the record associated with the *Start* task. This is the record that contains the smart section. It is the target record for the *Records* section.

Figure 127. SCHEDULE workflow Trigger Action properties

The *Records* section also specifies that the action should be performed on the record referenced by the target record's *Event* record section.

After the workflow is created, the next thing to do is add an action to the form's smart section to launch the workflow. Adding an action to a form's smart section is discussed in "Smart section actions".

Having discussed these design-time activities, we are now ready to discuss what a user needs to do at runtime to schedule an event using the *Event* form.

The form for a new *Event* record initially looks like the following figure.

Figure 128. Event form, Event tab

Begin filling out this form by supplying values for the fields in the *Event Info* section.

The *Subject* field can be used to enter a brief comment about the purpose of the event.

The *Event Type* field allows a person to select a value that workflows may use to distinguish between different kinds of events. This field's drop-down list may contain values that are appropriate for some applications but not others. The items in the drop-down list are managed by the List Manager under the name *Event Type*.

A person should set the value of the *Event Start Date* field to the time and date when the event will be scheduled to start. If the event will be recurring, the date portion of this field will be used to determine the earliest start date, not the actual start date. The actual date(s) when the event will occur will be determined by the recurrence information instead.

The default value for this field is the current time and date. For most events, this is not a useful default.

The *Event Duration* field may be filled in with the duration of the event.

After you have finished entering values into the fields of the *Event Info* section, the next thing to do is select one of the *Recurrence Pattern Type* radio buttons. The portion of the form below the *Recurrence Type* section varies, depending on which of these radio buttons you select.

Manually Scheduling a Single Occurrence

Selecting the *Single Occurrence* radio button means that the event will occur just at the one date and time specified by the *Event Start Date* field.

Manually Scheduling a Daily Occurrence

Selecting the *DAILY* radio button means that the event will recur with a frequency measured in days. It will recur after the number of days specified in the *Recur Every ___ Day(s)* field has elapsed. For example, if the value of the *Recur Every ___ Day(s)* field is 2 then the event will recur every other day.

Use the Weekday and Weekend Day check boxes to define whether the event will recur on just weekdays (i.e., Monday through Friday), weekend days (i.e., Saturday and Sunday), or both.

The event will continue to recur until it has completed the number of occurrences specified in the *End After ___ Occurrences* field. If no value is specified in the *End After ___ Occurrences* field, the event will continue recurring forever.

If an event recurs, a *Scheduled Event* record will be created for each recurrence. However, these *Scheduled Event* records may not be generated all at the same time. Instead, just the next few *Scheduled Event* records that are needed are generated. The exact number that will be generated is determined by the value of the *Generate __ Day(s) of Events (rolling)* field.

After the first of the scheduled events happens, another *Scheduled Event* record is created. This keeps the number of pre-generated *Scheduled Event* records at the number specified by the *Generate __ Day(s) of Events (rolling)* field. Once the number of remaining scheduled events for the recurrence specified falls below the *Generate ___ Day(s) of Events (rolling)* field value, the system stops generating additional *Scheduled Event* records.

In the *Exclude Dates* section, specify ranges of dates. During these ranges of dates, the event will not recur.

Manually Scheduling a Weekly Occurrence

Selecting the *WEEKLY* radio button means that the event will recur with a frequency measured in weeks. It will recur after the number of weeks specified in the *Recur Every ___ Week(s) On* field has elapsed. For example, if the value of the *Recur Every ___ Week(s) On* field is 2 then the event will recur every other week.

On weeks when the event occurs, it will happen on the days of the week that correspond to the checked check boxes in the *Weekly Recurrence* section.

The event will continue to recur until it has completed the number of occurrences specified in the *End After ___ Occurrences* field. If no value is specified in the *End After ___ Occurrences* field, the event will continue recurring forever.

If an event recurs, a *Scheduled Event* record will be created for each recurrence. However, these *Scheduled Event* records may not be generated all at the same time. Instead, just the next few *Scheduled Event* records that are needed are generated. The exact number that will be generated is determined by the value of the *Generate __ Week(s) of Events (rolling)* field.

After the first of the scheduled events happens, another *Scheduled Event* record is created. This keeps the number of pre-generated *Scheduled Event* records at the number specified by the *Generate __ Week(s) of Events (rolling)* field. Once the number of remaining scheduled events for the recurrence specified falls below the Generate ___ Week(s) of Events (rolling) field value, the system stops generating additional Scheduled Event records.

In the *Exclude Dates* section you can specify ranges of dates. During these ranges of dates, the event will not recur.

Manually Scheduling a Monthly Occurrence

Selecting the *MONTHLY* radio button means that the event will recur with a frequency measured in months. It will recur after the number of months specified in the *Recur Every ___ Month(s) On* field has elapsed. For example, if the value of the *Recur Every ___ Month(s) On* field is 2 then the event will recur every other month.

The day of the month on which the event happens can be determined by one of three methods. If it is desired to have the event occur on a specific numbered day of the month, the *Day of month (1-31)* field value can be set. For example, if the value of the *Day of month (1-31)* field is 2, the event will happen on the second day of the month.

If the value in the *Day of month (1-31)* field is 29, 30, or 31 and a recurrence falls in a month that does not have that many days, the event happens on the last day of the month. So, if you want an event to be repeated on the last day of every month, specify 31 for the value of the *Day of month (1-31)* field.

Alternately, the day of the month on which the event happens can be determined by setting the Week of month or Day of Week value. Setting a value for Week of month instructs the system to have the event happen on the week number selected. The Day of Week value defines which day of the week the event happens for the week selected in the Week of month field.

The event will continue to recur until the date specified in the *End After ___ Occurrences* field. If there is no value specified in the *End After ___ Occurrences* field, the event will continue recurring forever.

If an event recurs, a *Scheduled Event* record will be created for each recurrence. However, these *Scheduled Event* records may not be generated all at the same time. Instead, just the next few *Scheduled Event* records that are needed are generated. The exact number that will be generated is determined by the value of the *Generate __ Month(s) of Events (rolling)* field.

After the first of the scheduled events happens, another *Scheduled Event* record is created. This keeps the number of pre-generated *Scheduled Event* records at the number specified by the *Generate __ Month(s) of Events*

(rolling) field. Once the number of remaining scheduled events for the recurrence specified falls below the *Generate ___ Month(s) of Events (rolling)* field value, the system stops generating additional *Scheduled Event* records.

Specify ranges of dates in the *Exclude Dates* section. During these ranges of dates, the event will not recur.

Manually Scheduling a Yearly Occurrence

Selecting the *YEARLY* radio button means that the event will recur every year. It will recur in the month specified by the *Recur Every ___ Year(s) On* field on the day specified in the *Day of Month (1-31) field*.

The event will continue to recur until it has completed the number of occurrences specified in the *End After ___ Occurrences* field. If no value is specified in the *End After ___ Occurrences* field, the event will continue recurring forever.

If an event recurs, a *Scheduled Event* record will be created for each recurrence. However, these *Scheduled Event* records may not be generated all at the same time. Instead, just the next few *Scheduled Event* records that are needed are generated. The exact number that will be generated is determined by the value of the *Generate __ Year(s) of Events (rolling)* field.

After the first of the scheduled events happens, another *Scheduled Event* record is created. This keeps the number of pre-generated *Scheduled Event* records at the number specified by the *Generate __ Year(s) of Events (rolling)* field. Once the number of remaining scheduled events for the recurrence specified falls below the *Generate ___ Year(s) of Events (rolling)* field value, the system stops generating additional *Scheduled Event* records.

In the *Exclude Dates* section you can specify ranges of dates. During these ranges of dates, the event will not recur.

Manually Scheduling an Ad Hoc Occurrence

Selecting the *Ad Hoc* radio button means that the event will recur but with no particular pattern. In addition to the event happening on the date specified by the *Event Start Date* field, it will also happen on the dates you specify in the *Also Schedule On* section.

Sometimes you may have multiple events fall on the same day and want just one of the events to happen, not all of them. For example, there may be a weekly event for shampooing the carpet in a hallway. There also may be an annual event to paint the hallway. You do not want the rug shampooing to happen when it would coincide with the painting. There is a feature called event shadowing that controls things like this.

There are two steps to make one event shadow another. The first is to put the event that has priority in the *Shadowing Events* section of the other. In the above example, you would put the painting event in the *Shadowing Events* section of the rug shampooing event. The other step is to supply a value for the *OffsetDate Duration (for Shadowing)* field.

An event is shadowed if another event referenced in its *Shadowing Events* section occurs within the number of days specified by the value of the *OffsetDate Duration (for Shadowing)* field. If you did not want the rug shampooing to happen if it is scheduled within two days of a painting event, then you would set the value of the rug shampooing event's *OffsetDate Duration (for Shadowing)* field to 2.

Schedule workflow task

Instead of designing applications to create scheduled events manually, use the Event form.

To use a workflow to automate the entire process of scheduling events, write a workflow that uses the Schedule workflow task. The Schedule task handles a number of details that would otherwise have to be done by a few other tasks.

A Schedule workflow task creates an *Event* record and associates it with the record on which events will be performed. A Schedule task also creates a *Recurrence* record, sets the value of its fields, and associates the *Recurrence* record with the *Event* record.

A Schedule task does not schedule an event to happen. A Trigger Action workflow task is needed to do that. The Trigger Action workflow task is discussed in "Trigger action task".

The properties form for a Schedule task is organized into three sections. Here are descriptions of the fields in the first section:

Label   This is the label used to identify this workflow task. The text in this field is used as the label for the shape that represents this kind of workflow task in drawings.

Description
A description of this workflow task goes in this field.

Records section (schedule workflow)

The second section of the form for a *Schedule* task's properties is labeled *Records*. The purpose of this section is to specify which record or records the event(s) will happen to.

The top of the *Records* section has two radio buttons to specify how the *Schedule* task will find the record(s). These radio buttons are labeled:

Workflow Activity
If this radio button is selected, the event(s) will happen to a record or records associated directly or indirectly with a preceding workflow task.

Existing Record
If this radio button is selected, the event(s) will happen to a record that exists at the time the workflow is being created.

The selection of one of these radio buttons determines what appears in the *Records* section.

After we finish describing the way that the *Records* section works when the *Workflow Activity* radio button is selected, we describe what happens when the *Existing Record* radio button is selected.

The fields and radio buttons that appear in the *Records* section under the *Workflow Activity* radio button are visible only if the *Workflow Activity* radio button is selected. Under the *Workflow Activity* radio button are two fields used to identify a target record(s). The target record is used to determine the record(s) that events will happen to.

There are radio buttons below the two fields. The way that the target record(s) will be used to determine the record(s) that events will happen to depends on which one of the radio buttons is selected.

Here are descriptions of the two fields:

Take the

> This drop-down list can have one of three possible values:
>
> **Record**
>
> > If *Record* is selected, then the record associated with the task specified by the field to the right of this one will be the target record. If multiple records are associated with the task, there will be multiple target records.
>
> **Assignee**
>
> > If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right will be the target record.

of Task

> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under the two fields determine how the target record(s) will be used to determine the record(s) that events will happen to.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record that the event(s) happen to.

Use its Reference

> If this is selected, a record referenced by a smart section or locator field of the target record will be the record(s) that the event(s) happen to. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.
>
> If the record referenced by the smart section or locator field is a link, the record used for the action item will be the link, not the underlying record.
>
> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

> If this is selected, records associated with the target record will be the records that the event(s) happen to. When you select this radio button, a window pops up that allows you to specify the type of association to use.

It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Records* section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

This is similar to the *Use its Association* radio button.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the events will happen to the parent of the target record.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this is the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the *Records* section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that the event(s) will happen to. If the record can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

If you want event(s) to happen to record(s) that are part of the application's configuration, then you should select the option of using an existing record in the *Records* section. When the *Existing Record* radio button is selected, the *Records* section looks like the one in the following figure.

Figure 129. Schedule properties, existing record selected

You use the *Module* and *Object* drop-downs to specify a kind of record and then click the *Record* link to find the specific record that the event(s) should happen to.

Recurrence section

The third section of the form for a *Schedule* task's properties is labeled *Recurrence*. The purpose of this section is to specify how often and how many times the event should be repeated, if at all.

The Recurrence section contains a group of radio buttons that allows you to select how often the event will be repeated. Under these radio buttons is a field that

contains a date used to calculate when the event will first happen. Under this field is another set of radio buttons used to select when the repetitions of the event will end.

Here are explanations of the radio buttons that control how often the event will be repeated:

None If this radio button is selected, the event will not be repeated.

Every _____ Days

If this radio button is selected, the event will happen after a specified number of days have elapsed. The event will first happen on the date specified in the *Start* field. After that, the event will happen every time the specified number of days has elapsed. The number of days between recurrences of the event is specified in the field between the words Every and Days. For example, *Every 3 Days* means that beginning on the date specified in the *Start* field, the event will happen every third day.

Every Weekday

If this radio button is selected, the event will happen on every Monday, Tuesday, Wednesday, Thursday and Friday on or after the date specified in the *Start* field.

Every Weekend Day

If this radio button is selected, the event will happen every Saturday and Sunday on or after the date specified in the *Start* field.

_____ Days After Each Task is Complete

If this radio button is selected, the event will first happen on the date specified in the *Start* field. It will be repeated the specified number of days after the end of the event. The number of days is specified in the field to the right of the radio button.

The end of an event is determined by the time that the event starts and the value in the *Event* record's *EventDuration* field. The value of the *EventDuration* field must be set after the *Event* record is created by the *Schedule* workflow task. The *EventDuration* field is described in "Fields in legacy event records".

Every _____ Weeks on _____

If this radio button is selected, the event will first happen on or after the date specified in the *Start* field, the first time that the current day is the specified day of the week. After the first repetition, the event is repeated every n weeks. n is specified by the field between *Every* and *Weeks*. The day of the week is specified by its selection in the drop-down list to the right of *Weeks on*. For example, *Every 3 Weeks on Monday* means that the event will happen on the first Monday on or after the date in the *Start* field and then repeat every three weeks.

_____ Weeks After Each Task is Complete

If this radio button is selected, the event will first happen on the date specified in the *Start* field. It will be repeated the specified number of weeks after the end of the event. The number of weeks is specified in the field to the right of the radio button.

The end of an event is determined by the time that the event starts and the value in the *Event* record's *EventDuration* field. The value of the

EventDuration field must be set after the *Event* record is created by the *Schedule* workflow task. The *EventDuration* field is described in "Fields in legacy event records".

On Day _____ Every _____ Months

If this radio button is selected, the event will first happen on or after the date specified in the *Start* field, on the first day that is the specified day of the month. The event will be repeated after the specified number of months, on the specified day of the month. The day of the month is specified by the field between *Day* and *Every*. The number of months is specified in the field between *Every* and *Months*.

For example, suppose that the value of the *Start* field is April 12, 2004 and the recurrence setting is *On Day 1 Every 3 Months*. The event would first happen on May 1, 2004 because that is the first day on or after the start date that is the first day of a month. The event would be repeated on August 1, 2004, November 1, 2004 and so on.

If the specified day of the month is 29, 30 or 31 and a repetition falls in a month that does not have that many days, then the repetition happens on the last day of the month. For example, if you want an event to be repeated on the last day of every month, you should specify *On Day 31 Every 1 Months*.

On the _____ _____ Every _____ Months

If this radio button is selected, the event will first happen on or after the date specified in the *Start* field, on the first day that is the specified day of the month. The event will be repeated after the specified number of months has elapsed on the specified day of the month.

The number of months is specified in the field between *Every* and *Months*.

The day of the month is specified as either the first, second, third, fourth or last occurrence of a particular day of the week in the month. The selection of first, second, third, fourth or last is specified in the drop-down list to the right of *On the*. The day of the week is specified in the drop-down list to the left of *Every*.

Selecting the last occurrence of a day of the week means that the event will happen on the fifth occurrence in a month of the specified day of the week if there is a fifth occurrence; otherwise the event will happen on the fourth occurrence in a month of the specified day of the week.

_____ Months After Each Task is Complete

If this radio button is selected, the event will first happen on the date specified in the *Start* field. It will be repeated the specified number of months after the end of the event. The number of months is specified in the field to the right of the radio button.

The end of an event is determined by the time that the event starts and the value in the *Event* record's *EventDuration* field. The value of the *EventDuration* field must be set after the *Event* record is created by the *Schedule* workflow task. The *EventDuration* field is described in "Fields in legacy event records".

Every _____ _____

If this radio button is selected, the event will first happen on or after the date in the *Start* field on the first day that is the specified day of the year. The event will be repeated every year after that on the specified day of the

specified month. The month is specified by selecting it in the drop-down list to the immediate right of *Every*. The day of the month is specified in the field to the right of the drop-down list.

If February 29 is specified, on years that are not leap years the event will happen on February 28.

On the _____ _____ of _____

If this radio button is selected, the event will first happen on or after the date in the *Start* field on the first day that is the specified day of the specified month. The event will repeat once a year on the specified day of the specified month. The month is specified by selecting it in the drop-down list to the right of *of*.

The day of the month is specified as either the first, second, third, fourth or last occurrence of a particular day of the week in the month. The selection of first, second, third, fourth or last is specified in the drop-down list to the immediate right of *On the*. The day of the week is specified in the drop-down list to the left of *of*.

Selecting the last occurrence of a day of the week means that the event will happen on the fifth occurrence in a month of the specified day of the week if there is a fifth occurrence; otherwise the event will happen on the fourth occurrence in a month of the specified day of the week.

_____ Years After Each Task is Complete

If this radio button is selected, the event will first happen on the date specified in the *Start* field. It will be repeated the specified number of years after the end of the event. The number of years is specified in the field to the right of the radio button.

The end of an event is determined by the time that the event starts and the value in the *Event* record's *EventDuration* field. The value of the *EventDuration* field must be set after the *Event* record is created by the *Schedule* workflow task. The *EventDuration* field is described in "Fields in legacy event records".

The *Start* field is under the set of radio buttons used to select how often the event will happen. The *Start* field contains the date used to determine on what day the event will first happen. The exact way this date is used depends on which one of the above radio buttons is selected. The description of each radio button contains more specifics.

The time that the event will happen is the time portion of the value in the *Event* record's *EventStartDate* field. This is discussed in "Fields in legacy event records".

Under the *Start* field are radio buttons for selecting how long the event will continue to be repeated. Here are descriptions of the radio buttons:

End Never
If this is selected, the event will never stop repeating.

End After _____ Occurrences
If this is selected, the event will stop repeating after it has happened the specified number of times. The number of times is specified in the field between *After* and *Occurrences*.

End by _____

> If this is selected, there will be no repetitions of the event on or after the specified date. The date is specified in the field to the right of *End by*.

Fields in legacy event records

Once an *Event* record has been created, the values of its fields must be set. A Schedule workflow task does not set the values of an *Event* record's fields, except for *EventRefId*. This is what a *Schedule* task does:

- A Schedule workflow task creates an *Event* record.
- A Schedule workflow task creates a *Recurrence* record.
- A Schedule task sets the values of the *Recurrence* record's fields with what is specified in the Schedule task's *Recurrence* section.
- A Schedule task associates the *Recurrence* record with the *Event* record.

An *Event* record has these fields:

EventStartDate

> This is the date and time when the event will happen.
>
> If the *Event* record has an associated *Recurrence* record and the selection of recurrence is not *None*, the date portion of this field will be used to determine the earliest start date, not the actual start date. In this case, the information in the associated *Recurrence* record determines the actual dates on which the event happens.
>
> The time of day an event starts is always the time of day in this field.
>
> The value of this field must be specified.

EventDuration

> This is the duration of the event.
>
> The value of this field must be specified, but can be zero.

EventEndDate

> The value of this field is when the event ends.
>
> This is a read-only field. Its value is calculated from the values of *EventStartDate* and *EventDuration* fields. The date portion of this value is meaningful only if the date portion of the *EventStartDate* field is meaningful.

Subject

> A person can type a few words into this field to describe what the event is about. The value of this field is copied into the *Subject* field of *Scheduled Event* records created from the *Event* record. The text in a *Scheduled Event* record's *Subject* field is used in the label for the *Scheduled Event* record that is displayed in a calendar tab.

EventType

> This is a list field. A workflow may use the value of this field to determine something about the nature of an event.
>
> The value of this field will be an item from a list named *Event Type*. The *Event Type* list contains values that are used by multiple applications. Feel

free to add items to the *Event Type* list for other applications. However, it is not safe to remove items in this list or change them unless you are certain that it will not hurt any applications.

EventInstruction
This is a read-only text field that is used to present some text in the *Event* form.

EventRefId
This is a read-only control number that is used as an *Event* record's name.

Recurrence Id
This is a read-only field that is used internally by the IBM TRIRIGA Application Platform. It must never be modified by a workflow or the user.

When the values of all the fields of an *Event* record and its associated *Recurrence* record have been set to the correct values, you can cause the event they describe to be scheduled by performing a *Schedule* action on the *Event* record.

Performing a *Schedule* action on an *Event* record causes a *Scheduled Event* record to be created for occurrences of the event that are scheduled. Another consequence of performing a *Schedule* action on an Event record is that a *SCHEVENTCREATE* system event happens to the record that the event is scheduled for.

The way that the life cycle for an *Event* record is organized, a *Schedule* action can be performed only once on an *Event* record.

After a *Schedule* action is performed on an *Event* record, its scheduling effect can be undone by performing an *Unschedule* action on the *Event* record. Performing an *Unschedule* action on the *Event* record does not allow another *Schedule* action to be performed on the *Event* record.

Scheduled events

A *Scheduled Event* record is used to represent each time that an event has been scheduled. If the form for a record includes a *Calendar* tab, you can see links to *Scheduled Event* records in the calendar.

Scheduled Event records are involved in at least these two associations:

- There is an association between a *Scheduled Event* record and the *Event* record that it was created from. The name on both sides of the association is *RECURRENCE*.
- There is an association from a *Scheduled Event* record to the record for which the event is scheduled. Its name is also *RECURRENCE*.

At the time a *Scheduled Event* record says that an event is supposed to start, a *SCHEVENTSTART* system event happens to the record that the event is scheduled for. At the time a *Scheduled Event* record says that an event is supposed to end, a *SCHEVENTEND* system event happens to the record that the event is scheduled for.

Chapter 11. Integration with external applications

Applications that run on the IBM TRIRIGA Application Platform can be made to work with applications that run outside the IBM TRIRIGA Application Platform in different ways. There are a variety of mechanisms available for this purpose. These mechanisms are intended for different sorts of applications and data. They differ from each other in a number of ways.

One of the ways integration mechanisms differ from each other is the extent to which they require the services of a professional programmer. The following is a list of integration mechanisms in order of how much programmer involvement is required.

Table 29. Integration mechanisms

Integration Mechanism	Description	Programmer Needed
Email	An email message is sent to an external application that can interpret the message.	Only if the external application is not already able to process email messages.
Form Report	Form reports provide a mechanism of presenting data in an Excel spreadsheet, Word document, or external report through IBM TRIRIGA. The application developer is responsible for creating the spreadsheet, document, or external report.	Not usually.
DataConnect	The DataConnect solution employs staging tables and workflow tasks, allowing an external source to write data directly into the IBM TRIRIGA staging tables and have IBM TRIRIGA workflow process this data for insertion into native IBM TRIRIGA business objects.	DataConnect may require a programmer to move the data into the staging tables. The workflows require a person that knows the business processes of the integration.
DataConnect for Fact Tables	The DataConnect for Fact Tables solution employs staging tables, allowing an external source to write data directly into the IBM TRIRIGA staging tables where ETL can be used to process this data for insertion into IBM TRIRIGA fact tables.	DataConnect for Fact Tables may require a programmer to move the data into the staging tables and a programmer to move the data from the staging tables to the fact tables.
Data Integrator	Reads data from simple spreadsheets or comma-separated files into IBM TRIRIGA Application Platform records.	Extracting data from another source into a comma-separated file or simple spreadsheet may require the assistance of a programmer.

Table 29. Integration mechanisms (continued)

Integration Mechanism	Description	Programmer Needed
Financial Transactions	The IBM TRIRIGA Application Platform processes financial transactions in a way that makes individual journal entries available to an external accounting program.	No programmer needed to make data available. Some applications will require programmer involvement to get the application to access the data.
IBM TRIRIGA Connector for Business Applications	IBM TRIRIGA Connector for Business Applications allows programmers to write programs that work directly with the IBM TRIRIGA Application Platform, solving most integration problems that cannot be solved with other integration mechanisms.	This requires a programmer who is knowledgeable about SOAP technology and this part of the IBM TRIRIGA Application Platform or the services of IBM TRIRIGA consultants.
Java Objects	This is a custom piece of software that can be invoked to communicate with the platform and other software. This is the most general possible integration mechanism.	This requires a programmer who is skilled at Java and is able to use IBM TRIRIGA Connector for Business Applications.

Email

Some applications are able to process email messages if they are in a specific format. You can create a workflow in the IBM TRIRIGA Application Platform to send email messages. The details of writing a workflow to send an email are discussed in "Notifications". The details of sending a form report as part of an email are discussed in "Attach format file task".

Form reports are a way to display IBM TRIRIGA application data using HTML to format the data. You can create an HTML form report to format the data that is sent via email. The details of creating a form report are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

A form report can be presented as the body of an email message. This is a convenient way to send a form report to someone who may not be logged into the IBM TRIRIGA Application Platform environment.

If you want to send your data through email and email is the application with which you are integrating, the form report can be sent to applications that run outside the IBM TRIRIGA Application Platform through email.

For form reports using Eclipse Business Intelligence Reporting Tools (BIRT) and SAP Crystal Reports, attach the report to the email as a PDF file.

If the recipient will be using the data in a Microsoft Excel spreadsheet, you can use Excel to generate the HTML. That makes it easier for the recipient to import into Excel.

IBM TRIRIGA DataConnect

The IBM TRIRIGA DataConnect imports data from external systems into the IBM TRIRIGA system. It is useful for initial loads of data and for batch insert/update of data on a recurring basis. The DataConnect solution employs staging tables and workflow tasks, allowing an external source to write data directly into the IBM TRIRIGA staging tables and using IBM TRIRIGA workflow to process this data for insertion into native IBM TRIRIGA business objects.

The basic steps in the DataConnect process are as follows:
- Create a staging table for each business object that will be populated or updated from the external source.
- Make the job business object. This business object controls the instance of the integration.
- Move data from the external source into the appropriate business object staging table and the DataConnect Job Control table.
- The DataConnect Agent will initiate a process (a synchronous workflow) within IBM TRIRIGA that reads records from the business object staging tables and either inserts new instances or updates existing instances while validating the input data and executing associated business logic.

More information about DataConnect can be found in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

IBM TRIRIGA DataConnect for fact tables

The IBM TRIRIGA DataConnect for Fact Tables imports data from external systems into the IBM TRIRIGA system. It is useful for initial loads of data and for batch insert/update of data on a recurring basis. The DataConnect for Fact Tables solution employs staging tables and ETL, allowing an external source to write data directly into the IBM TRIRIGA staging tables and using ETL to process this data for insertion into IBM TRIRIGA fact tables.

The basic steps in the DataConnect for Fact Tables process are as follows:
- Create a staging table for each business object that will be populated or updated from the external source.
- Move data from the external source into the appropriate business object staging table.
- Create an ETL that process the data from the staging table and move it into the fact table. If a recurring move is desired then create an ETLJobItem. Prune the staging table as appropriate. Additional processing, such as combining data from other tables, deriving new values, and validation the data should also be considered.

More information about DataConnect for Fact Tables can be found in *Application Building for the IBM TRIRIGA Application Platform 3: Data Management*.

Data Integrator

The Data Integrator is a tool provided by the IBM TRIRIGA Application Platform to allow records to be created or updated from data in a kind of file called a tab-delimited file. If you want an application that runs in the IBM TRIRIGA Application Platform to receive data from an external application that runs outside of the IBM TRIRIGA Application Platform, the Data Integrator is the simplest way to get the data into the IBM TRIRIGA Application Platform.

Application Building for the IBM TRIRIGA Application Platform 3: Data Management describes the Data Integrator in detail.

Tab-delimited files can be created from spreadsheets. There are a number of accounting packages and other kinds of programs that can produce tab-delimited files.

The simplest way to run the Data Integrator is for it to be run interactively by a person. If you run the Data Integrator this way, no programmer involvement is needed.

In some cases, it is not practical to have a person run the Data Integrator every time there is data to be received into the IBM TRIRIGA Application Platform from a tab-delimited file. For these cases, it is possible to schedule data to be read from a tab-delimited file periodically without the involvement of a person.

Financial transactions

Among the integration requirements for an application may be requirements for the application to record transactions that capture the impact of numbers recorded by an application. For example, when a customer buys something, an application may generate a transaction to bill the customer and decrease the value of the inventory by the amount that was sold.

Requirements to record numbers as Atomic, Consistent, Isolated, and Durable (ACID) transactions and feed them to an accounting package are rather common. They are so common that the IBM TRIRIGA Application Platform has a special utility for generating financial transactions. This is described in *Application Building for the IBM TRIRIGA Application Platform 3: Calculations*.

IBM TRIRIGA Connector for Business Applications

IBM TRIRIGA Connector for Business Applications allows programmers to write external programs that interact directly with IBM TRIRIGA Application Platform using a technology called SOAP. This mechanism allows external programs to manipulate records in the platform environment.

There are two ways a programmer can use the SOAP APIs. One is to modify an existing external application to use the APIs directly. The other is to write a separate program that uses the APIs and acts as an intermediary between the IBM TRIRIGA Application Platform and the external programs with which it needs to be integrated.

IBM TRIRIGA Connector for Business Applications is described in *IBM TRIRIGA Connector for Business Applications 3 Technical Specification*.

Java objects

A method of a custom Java object can be invoked to perform any custom logic or solve any integration problem. The mechanism that makes this work is described in "Custom task".

Chapter 12. Notifications

Sending a notification that appears in a user's portal or as an email is done by creating a Notification record in the Mail module; populating it with required information, such as recipients, content, and subject; and using workflow to trigger an action on that record to send the notification.

Message content creation

Message content is managed centrally and is independent of workflows. There is a clear distinction between the fixed part of a notification that is always the same when a particular notification is generated and the variable part of a notification that may be different each time a particular notification is generated.

To clarify this distinction between the fixed and variable parts of a notification, consider this example. Suppose you want to generate a notification that looks like this:

```
Michael Smith requested a snack cart in room W319 at 03/19/2014 10:30 AM.
```

The fixed part of this notification would probably be:

```
_____ requested a _____ in _____ at _____.
```

The variable part of the message would be the pieces of text that get put in the blanks. In the case of the preceding example, these would be: Michael Smith, snack cart, room W319, and 03/19/2014 10:30 AM.

Three kinds of records play different roles in the notification process:

- Notification Content records are templates for notifications and specify the fixed portion of a notification's content. You use Notification Content records to predefine the content of notification messages. Notification Content records are described in "Notification content records".
- Notification Helper records supply the variable part of a notification's content and also may be used to associate a specific record with a notification. Notification Helper records are described in "Notification helper records".
- *triPeople* records specify the recipients of notifications. Associations from a Notification Helper record to *triPeople* records specify to whom the generated notifications are sent. *triPeople* records are described in "triPeople records".

Notification content records

Notification Content records are templates for notifications. They provide the fixed portions of a notification's content. To manage Notification Content records, navigate to Tools > Approvals & Notifications > Notifications > Notification Content.

Workflows use a combination of the value of the *ID* field and the Language field to identify the Notification Content record to be used to create a notification.

The value of the *Notification Subject* field is used to create the subject of a notification and the value of the *Notification Content* field is used to create the body of the notification.

Notice that the text in the *Notification Subject* field and in the *Notification Content* field contains numbers in curly braces, for example, *{7}*. Each number in curly braces is a place holder for text to be supplied by a Notification Helper record. The number inside the curly braces is not converted. The *Description* field should describe the purpose of each of these place holders.

The value of the *Language* field identifies the language in which the values of the *Notification Subject* field and the *Notification Content* field are written. For more information about language support, see the *IBM TRIRIGA Application Platform 3 Globalization User Guide*.

For more information about notifications, see the *IBM TRIRIGA 10 Application Administration User Guide*.

If a Notification Content record references any form report templates, Notification Helper records that use the Notification Content record must identify the record to provide the variable content for the form report.

Notification helper records

To send a notification, a workflow creates a Notification Helper record from the Notification Helper business object in the *triHelper* module. After creating the record, the workflow sets the values of its fields, associates the record with *triPeople* records to identify the notification's recipients, and then sends the notification by triggering a Calculate action on the Notification Helper record.

The fields in a Notification Helper record are as follows:

triIdTX
> Workflows must set the value of the *triIdTX* field to the ID of the Notification Content record that will provide the fixed portion of the notification's content.

triInput1TX, triInput2TX, ..., triInput9TX
> These fields correspond to the place holders *{1}, {2}, ..., {9}* that may appear in the fixed content provided by the *triNotificationContent* record. The value in each of these fields will be substituted for the corresponding place holder in the content.

triLanguageLI
> Workflows must set the value of the *triLanguageLI* field to the Language of the Notification Content record that will provide the fixed portion of the notification's content.

triLinkedRecordTX
> If the notification should be associated with a record, this locator field should refer to the appropriate record.

triLinkedRecordIdTX
> If the notification should be associated with a record, this text field should contain the system's internal ID of the record referenced by the triLinkedRecordTX field for the appropriate record. This is the field that identifies the record that will provide the variable content for the form report(s).

triLinkedBusinessObjectLI
> The value of this field should be the label text of the business object that was used to create the record referenced by the *triLinkedRecordTX* field.

triLinkedFormLI

The value of this field should be the label text of the form associated with the record referenced by the *triLinkedRecordTX* field.

triLinkedRecordStateTX

The value of this field should be the state of the record referenced by the *triLinkedRecordTX* field.

triLinkedRecordStatusCL

The value of this field should be the status of the record referenced by the *triLinkedRecordTX* field.

triSkipLinkedRecordLinkCreationBL

The associations needed to identify the recipients of a notification are discussed in "triPeople records".

Additional associations can be made to records in the Document Manager that will attach specific documents to the resulting notification that are not included in the Notification Content. To attach additional documents, associate the document record to the Notification Helper using the association Has Notification Attachment. To attach additional form reports, associate the document record to the Notification Helper using the association Has Notification Report.

To attach a binary field to an email as an attachment, associate a record of type EmailAttachment in the Mail module to the Notification record before triggering the notification action.

After the above fields are all set and the associations made, the workflow can trigger a Calculate action on the Notification Helper record to send the notification. After the Calculate action, the *triSubjectTX* field will contain the subject of the notification that was sent and the *triContentTX* field will contain the body of the notification.

triPeople records

The workflow triggered by the Calculate action knows where to send the notification it creates because of associations named *Notify* that the Notification Helper record has with *triPeople* records.

If an associated *triPeople* record identifies an active user, the notification is put in the notification section of the user's portal. If an associated *triPeople* record identifies an e-mail address, the notification is e-mailed to the specified address.

Workflows for notifications

The Attach Format File workflow task is intended specifically for preparing notifications to be sent to users.

Attach format file task

An Attach Format File task attaches the output of a form report to a *Notification* record in a way that causes the form report output to be attached to messages sent using the *Notification* record. Form reports are discussed in the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

The properties form for an Attach Format File task is organized into three sections. Here are descriptions of the fields in the first section:

Label This label identifies this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description

A description of this task goes in this field.

Format File

The *Object Type* field of this form's *Attach Format File From* section specifies the business object that must have been used to create the record for which the form report will be produced.

The value of this field is the name of form report template file that this task will use to produce form reports. To select a file, click the hyperlink to the right of the *Format File* label.

The task checks whether the file type supplied is a supported format, and if not, it attaches the file as-is (note that this does not work if Embed File is selected). The supported report formats are HTML documents, BIRT reports, and RPT files.

After you select a file, the name of the selected file replaces the previous text of the hyperlink.

Embed File

A form report can be included in an e-mail in two different ways:

- The form report can be embedded in the message, meaning that it will appear as part of the message's text. The advantage of sending a message this way is that the person reading the message need not do anything extra to see the report. To send the form report this way, select the *Embed File* check box.

- The form report can be sent as an attached PDF file or Microsoft Word or Excel file. When sent this way, the person reading the message generally needs to perform an extra mouse click to see the message in Word, Excel, or PDF. The advantage is that the report can be used in Word, Excel, or PDF. To send the form report this way, clear the *Embed File* check box.

Attach format file from section

The second section of an Attach Format File task properties form is labeled *Attach Format File From*. The purpose of this section is to specify the record whose data will be used to generate the form report.

The *Attach Format File From* section has two radio buttons to specify the record used to generate the form report. These radio buttons are labeled:

Workflow Activity

If this radio button is selected, then the record that contains the data to generate the form report will be associated with a previously performed workflow task.

Existing Record

If this radio button is selected, then the record that contains the data used to generate the form report will be a specified record that exists now.

The selection of one of these radio buttons determines what appears in the *Attach Format File From* section.

When the *Workflow Activity* radio button is selected, below the radio buttons the *Attach Format File From* section has all the fields that are in the *To Notification* section. The difference between the fields in these sections is that the fields in the *Attach Format File From* section are used to select the record that contains the data that will be used to generate the form report and the fields in the *To Notification* section are used to specify the *Notification* record that will be used to send the form report. Because of these similarities, the description of these fields is part of the description of the *To Notification* section.

If you want the form report to be generated from a record that is part of the application's configuration, then you select the option of using an existing record. Specify the value for the *Module* and *Object* properties so that this task knows what kind of record will be the parent. Then click the *Record* link to find the specific record you want this task to use.

To notification section

The third section of the Attach Format File task properties form is labeled *To Notification*. The purpose of this section is to specify the *Notification* record that will be used to send the form report.

The following description of the fields in the third section also applies to the fields in the second section that appear in the *Attach Format File From* section when its *Workflow Activity* radio button is selected. The fields in the *Attach Format File From* section serve a purpose similar to the corresponding fields in the *To Notification* section. The difference between the corresponding fields in the *Attach Format File From* section is that they are used to identify the record that contains the data that will be used to generate the form report and the fields in the *To Notification* section identify the *Notification* record used to send the form report.

At the top of the *To Notification* section are fields used to identify a target record. The target record is used to determine the *Notification* record that will be used to send the form report.

There are radio buttons below the fields. The way that target records are used to determine the record used to send the report depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the

> This is a drop-down list that can have one of three possible values:

> **Business Object**
>
> > If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>
> > This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record that will be used to send the report.

When the properties form is first displayed, only the currently selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it   If this is selected, the target record will be the *Notification* record used to send the report.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target records will be the *Notification* record used to send the report. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, records associated with the target record will be the *Notification* record used to send the report. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful

when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the child record.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be used to send the notification. For this third section the value of this field should be *Notification*.

Example: Notification workflow

The workflow diagram in the following figure is a simple example of a workflow that always sends the same message to the person who is currently logged in.
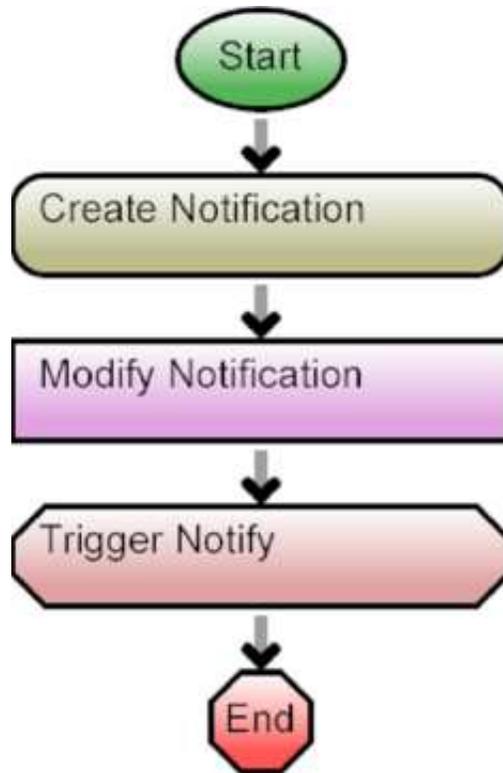
Figure 130. Example of a notification sending workflow

The first task creates the *Notification* record, sets the message content and makes the *RefObject* field refer to the record that was used to launch the workflow.

The second task sets the recipient of the notification.

The third task triggers the *Notify* action on the *Notification* record.

Chapter 13. Security

Security controls access to application features and business objects.

Security elements

In its most general sense, security is something that prevents bad things from happening while allowing good things to happen. As it applies to the IBM TRIRIGA Application Platform, security is what allows users to look at or manipulate data they should be looking at or manipulating, while preventing users from looking at or manipulating data they should not be.

The following lists the main facets to security on the IBM TRIRIGA Application Platform:

Authentication
> This involves users identifying themselves to the IBM TRIRIGA Application Platform and then doing something else to prove that they really are who they claim to be.

Licenses
> A license allows a user to access specific features of an application or to use tools that are provided by the IBM TRIRIGA Application Platform. A license is useful for controlling access to applications and tools. It is not helpful for controlling access to records.

Groups
> A group is a list of users and of other groups. A user can belong to more than one group.

Access Permissions
> Access permissions determine what kinds of records the members of a group can access and what they can do with the records.

Organization and Geography
> You can use a record's organization and geography to control which users can access the record.

Project
> The project that is associated with a record can restrict who is able to view or modify the record.

The basic rule of IBM TRIRIGA Application Platform security is that a user can look at something or perform an action only if the user belongs to a group that has permission to do so.

Authentication

Authentication requires users to identify themselves to the IBM TRIRIGA Application Platform and then do something else to prove that they really are who they claim to be.

Setting up authentication for a user involves two steps:
- Create a record that describes the user.

- Create an embedded My Profile record that describes the user's user ID and how the user interacts with the IBM TRIRIGA Application Platform.

The typical procedure for creating a record that describes a user is a manual procedure. Navigate to Portfolio > People. Here you can create triPeople records by using the Employee, Consultant, or External Contact forms. You also may configure other types of records to describe users.

The forms for these records have a number of tabs. After you have filled in the required fields on the General tab, you are ready to fill in the details of the My Profile record that describes the user ID to the IBM TRIRIGA Application Platform and the way the user interacts with the platform.

You can create the My Profile record indirectly by going to the Profile tab of the record you just created. The fields on the Profile tab have these meanings:

User Name

> This is the user ID that the user uses to sign in to IBM TRIRIGA. This field must be filled in even though the form does not indicate that it is required.

> There is no field to set the user's initial password. Unless modified in your implementation, the password for a new user is password.

Active TRIRIGA User?
> Select this check box if you want the user to be able to sign in.

Initial Pwd Reset
> When this check box is selected, the first time this user signs in to IBM TRIRIGA the user is prompted to change their password.

Home Page
> Defines the main window, or portal, the user sees when the user first signs into IBM TRIRIGA when at the Company Level. This also displays whenever the user clicks the Home menu item. If this value is blank, IBM TRIRIGA renders a default portal. See the *IBM TRIRIGA Application Platform 3 User Experience User Guide* for more information about portals.

Menu A navigation collection that defines the structure and content of the menu bar at the top of page when at the Company Level. See the *IBM TRIRIGA Application Platform 3 User Experience User Guide* for more information about navigation collections and menus.

Project Home Page
> Similar to Home Page, but only applies when the user is working in a Project context.

Project Menu
> Similar to Menu, but only applies when the user is working in a Project context.

Sitemap?
> If this check box is selected, a Sitemap link is available for this user. The sitemap displays a site index that contains all of the selectable navigation items based on the user's menu. Note that navigation items can be configured to only appear on the sitemap, and hence would be unavailable to users that do not have the Sitemap check box selected. See the *IBM TRIRIGA Application Platform 3 User Experience User Guide* for more information about sitemap and navigation items.

Disable Company Level?
>Normally you leave this cleared. If this check box is selected, the Clear Project button is disabled for the user. This works in conjunction with application features that populate the last project used. The project selector enables the user to move between projects, but without the Clear Project button, the user cannot exit a project and go to the Company Level.

User Language

>The value of this field is the language that the user prefers to use, such as US English, Spanish, French. The value of this field determines which set of labels and other text the user sees.

>The list of languages that you can choose from in this field is determined by a list named Language.

>For more information, see the *IBM TRIRIGA Application Platform 3 Globalization User Guide*.

Currency
>Use this field to specify the default currency for the user. For more information, see the *IBM TRIRIGA Application Platform 3 Globalization User Guide*.

Time Zone
>The time zone that the user is normally in. If blank, the time zone is from the time zone in which the application server is running.

Delegate To
>The purpose of this field is to allow another user to handle this user's approval action items when this user is not able to do it. If this field has a value, it is the user allowed to approve action items on behalf of this user.

Approval Amount
>The maximum amount that this user is authorized to approve on a single transaction.

Enable Accessibility Mode

>When selected, metric reports are displayed in tabular format by default. The user can switch to chart format.

>Also when selected, schedule assumptions in a capital project form are shown in a separate Schedule Assumptions tab instead of in the Schedule tab.

Enable Skip Navigation

>When selected, the Skip Navigation link is displayed at the top of portals and forms. After the user tabs to the Skip Navigation link and presses Enter, the user is taken directly to the main content area.

Group Details
>The groups the user is in.

License Details
>The licenses the user has.

Having filled in the fields of the Profile tab, you are ready to create the user's My Profile record by clicking the Activate action. After the My Profile record is created, the user can sign in to IBM TRIRIGA.

By default, the platform logs successful and failed logins and user account changes in the security.log file. You can turn off this logging by editing the log4j.xml file located in the `<IBM TRIRIGA installation>\config` folder (for example, `C:\Tririga\config\log4j.xml`).

Being able to sign in is not useful by itself. If no licenses are granted to a user, the user cannot access any application features or tools provided by the platform. User IDs that have not been granted any licenses are not allowed to sign in.

Having licenses without membership in a group is not useful either. In order to be able to access any record, a user must have permission to do so. To perform an action on a record, a user must have permission to perform the action on the record. A user gets permissions by being a member of a group that has the permissions.

Other triPeople business objects

If using a record created from a triPeople business object is not appropriate for your application, you can create and use another business object in the triPeople module.

The business object must have an association with the My Profile business object. The association must be named Associated To on both ends.

The form should have a Profile tab similar to the one in the Consultant, Employee, or External Contact form. The Profile tab should have an action on it that allows a My Profile record to be created that describes the user's user ID.

The easiest way to accomplish these things is to copy what already exists. Create the new business object by copying the triPeople business object. Copy the corresponding form to create the new form.

Passwords

IBM TRIRIGA supports strong passwords. Using strong passwords lowers the overall risk of a security breach. Define your company's implementation in Tools > System Setup > System > Password Setup.

The system only uses the rules defined in the rest of the Password Setup form when the Enforce Password Rules field is selected. When Enforce Password Rules is not selected, which is the default value, the system uses the standard IBM TRIRIGA password rules and ignores the rest of the values in the Password Setup form.

The system begins to follow the rules defined in the Password Setup as soon as the record is in Active status.

Single sign-on

The IBM TRIRIGA Application Platform's authentication mechanism is designed to work with most single sign-on mechanisms. There is more than one way to integrate the IBM TRIRIGA Application Platform's authentication mechanism with single sign-on mechanisms. The essence of how it works is that the IBM TRIRIGA Application Platform learns from an external trusted source the user ID of the user

to be logged in to the platform. If the user ID from the external source matches a user ID specified by a My Profile record, then the platform accepts the user ID as authenticated and logs the user in.

The details of how to integrate the IBM TRIRIGA Application Platform authentication mechanism with a single sign-on mechanism are described in the *IBM TRIRIGA Application Platform 3 Single Sign-On Setup User Guide*.

License management

A license allows access to business objects, tools, and their corresponding menu items in the IBM TRIRIGA Application Platform. When an IBM TRIRIGA product is purchased, one or more license files are made available to give access to the combination of objects and tools provided by each license.

You obtain licenses by buying them. You buy the kinds of licenses you need in the quantity that you need and install them in your IBM TRIRIGA Application Platform environment. After you install the licenses, they are in your `<IBM TRIRIGA installation>\config\licenses` folder.

The licenses are listed in **Tools** > **System Setup** > **System** > **Licenses**. For each license, you can see whether it is active in the system.

Users cannot take advantage of a license unless it is assigned to them.

To assign licenses to users for existing users, use the License Manager in Tools > Administration. You can also assign licenses individually to new or existing users when you create or edit their people records. For more information, see *Creating people records* in the IBM Knowledge Center. The rest of this section describes how to assign licenses to existing users in the License Manager.

Three kinds of licenses may appear in the License Manager.
- Product licenses that allow users to access features of IBM TRIRIGA application products. A user ID cannot sign in to the IBM TRIRIGA Application Platform environment unless it has been assigned at least one product license.
- Platform licenses that determine the features or environment for which a IBM TRIRIGA Application Platform environment is configured. These affect the platform environment as a whole. They do not have any direct relationship to what an individual user ID may do.
- An IBM TRIRIGA Custom Application license is a special kind of product license that does not provide access to any application features. However, assigning a custom application license to a user allows that user to sign in and use applications that were not created by IBM TRIRIGA.

The List View tab in the License Manager contains a list of available licenses. When you select a license in the List View tab, the user IDs to which the license has been assigned are displayed.

You can add user IDs to the list by clicking the **Add Users** action. You can remove user IDs from the list by selecting the check box next to a user ID and clicking the **Delete Users** action.

The query on the Add User action can be customized. To have the system run your query, name your query "Security License List" in the Report Manager.

The licenses shown in the License Manager are synchronized with the licenses shown in user profiles. Adding a user to the License Manager triggers a workflow to update the user's profile record. Deleting a user from the License Manager triggers a workflow to remove the license from the user's profile record.

When you need to change the licenses or security groups assigned to more than a few users, you can use the Bulk Security Utility to update many users at once. The Bulk Security Utility is in **Tools** > **Administration**.

The Matrix View tab in the License Manager shows the list of available licenses and the applications and modules.

The license matrix is IBM PROPRIETARY and CONFIDENTIAL and its disclosure to you shall be governed by the terms of the IBM Agreement for the Exchange of Confidential Information found at http://www-05.ibm.com/support/operations/files/pdf/aeci_ca_en.pdf, or the applicable confidentiality agreement signed between IBM and your company. IBM TRIRIGA is providing the license matrix as a courtesy to you under the condition that you treat the information with the utmost confidence. Under no circumstances may you distribute or allow use of the license matrix outside of your organization without first obtaining IBM's express written consent. The license matrix should not be construed as a definitive list of functionality licensed by the you; it is simply a proxy for the limited functionality that you have actually licensed, but your actual licenses are set forth in separate agreements between IBM and your company. The license matrix is provided for informational purposes only without warranty of any kind, does not contain contractual obligations, does not constitute a software, product or services warranty, and is subject to change without notice.

When you select a user on the Matrix View tab, the applications and modules that the user has access to are highlighted.

IBM TRIRIGA Application Platform license

You need an IBM TRIRIGA Application Platform license to create or modify business objects or to create or modify a workflow.

Attention: Be extremely careful to whom you assign an IBM TRIRIGA Application Platform license. An IBM TRIRIGA Application Platform license allows a user to circumvent all platform security features.

Groups

A group identifies a set of user IDs and the access that the group has to records and what can be done with the records.

Groups are managed by the IBM TRIRIGA Application Platform's Security Manager. To access the Security Manager, navigate to Tools > Administration > Security Manager.

IBM TRIRIGA provides a number of predefined security groups corresponding to roles our customers commonly have. Use these security groups as a starting point to meet your company's specific security needs.

You can create a new group by clicking the **Add** action. Click the hyperlinked name of a group to edit it. To delete a group, select the check box next to the group's Name and click the **Remove** action.

After you add, update, or delete a group, clear the Security Scope cache in the Administrator Console. For more information about the Administrator Console, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

When you use a group other than the Admin Group to control access to the Document Manager, you must include the group in the ROOT document record and the group must have at least edit access.

The form for a group has three tabs that are used for security purposes:
- General
- Members
- Access

General tab

The **General** tab has two sections that are labeled General and Data Access.

Here are descriptions of the fields in the General section.

Name   The text in this field is the name of the group.

Description
> This field contains a description of the group.

Personalize Portal

> If Yes, users in this group can select **Personalize** on their portal, which allows them to customize their Home page. If No or empty, the Personalize button is not available to users in this group.

> For information about how users can personalize portals, see the Getting Started videos in the Media Library at https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM+TRIRIGA1/page/Media+Library.

Personalize Bookmarks

> If Yes, users in this group can select **Add to Bookmarks** and personalize their Bookmarks menu. If No or empty, bookmarking is not available to users in this group.

> For information about how users can personalize bookmarks, see the Getting Started videos in the Media Library at https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM+TRIRIGA1/page/Media+Library

Ignorable For Nav. Overrides

> If Yes, the group is excluded from consideration in Group Overrides in a Navigation Collection.

> If No or empty, the group is included in Group Overrides in a Navigation Collection.

> See "Navigation Collections" in the *IBM TRIRIGA Application Platform 3 User Experience User Guide* for more information about Group Overrides.

Access All Profiles

If Yes, users in this group can access their own My Profile record or another user's My Profile record.

If No or empty, users in this group can access their own My Profile record but cannot access another user's My profile record. This is the default value.

The ENABLE_PROFILE_ROW_LEVEL_SECURITY property in TRIRIGAWEB.properties must be set to Y for the value of Access All Profiles to control user security access to My Profile records. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Here are descriptions of the fields in the General tab's Data Access section.

System Organization

If the value of the group System Organization field is blank, members of the group have access to records with blank System Organization fields. And, members of the group do not have access to records with values in their System Organization fields.

If the value of the group System Organization field is not blank, members of the group have access to records with values in their System Organization fields at the same level or at a lower level in the Organization hierarchy. If the value of the group System Organization field is \Organizations, members of the group have access to records with blank System Organization fields. If the value of the group System Organization field is not blank and not \Organizations, and the record's System Organization field is blank, members of the group have access to records in queries but do not have access to records in forms.

If you do not intend to use groups to restrict access to or visibility of records, and records have values in their System Organization fields, then set the value of the group System Organization field to \Organizations.

This is discussed in greater detail in "Organization and geography".

System Geography

If the value of the group System Geography field is blank, members of the group have access to records with blank System Geography fields. And, members of the group do not have access to records with values in their System Geography fields.

If the value of the group System Geography field is not blank, members of the group have access to records with values in their System Geography fields at the same level or at a lower level in the Geography hierarchy. If the value of the group System Geography field is \Geography, members of the group have access to records with blank System Geography fields. If the value of the group System Geography field is not blank and not \Geography, and the record's System Geography field is blank, members of the group have access to records in queries but do not have access to records in forms.

If you do not intend to use groups to restrict access to or visibility of records, and records have values in their System Geography fields, then set the value of the group System Geography field to \Geography.

This is discussed in greater detail in "Organization and geography".

Members tab

The Members tab is used to control which users and other groups are members of a group.

You can add users directly to the group by clicking the **Add Users** action.

You can add users to a group indirectly by clicking the **Add Groups** action. Adding a group as a member of another group causes all user IDs and groups that are members of the added group to be members of the group to which they were added. It also gives the Access Permissions in the added group to the group.

Access tab

The Access tab determines how much access membership in the group grants to records created from specified forms. It also determines what actions the members of the group are allowed to perform on the records. In addition, the Access tab determines the access that the members of the group have to the tabs and sections of the form. The Access tab also determines the access to models and their associated actions.

The Access tab contains two sub-tabs: an Access Configuration sub-tab and an Access Summary sub-tab. The Access Configuration sub-tab has an Object panel and a Permission panel. The Object panel shows a tree that contains a list of forms that are organized by module. The tree also contains all of the tools that the IBM TRIRIGA Application Platform provides, such as the Data Modeler and the Report Manager. The only exception is the Security Manager. To set access for the Security Manager, select **Group** in the Object panel. The tree also contains a list of all the models available in the environment.

The content of the Permission panel varies based on the selection in the Object panel.

Modules that contain published forms are represented by a plus sign icon. You can see the forms under a module by clicking the module's plus sign icon. Tools and modules that do not contain any published forms are represented by a dot.

Models appear in the Object panel under the Models node. Only systems with models have the Models node.

Changes made outside of the Models node do not affect UX applications. The only place to change security access to a UX application is in the Models node.

You can select a module, tool, or anything else that is in the Object panel by clicking its name. When something is selected in the Object panel, relevant choices for permissions appear in the Permission panel.

The permissions vary with the type of object that is selected in the Object panel. The meaning of the permission also varies with the type of object that is selected.

Review the access permissions in any group that you create

A new custom group by default provides no access to any tool or form object. With a new group, all access is prohibited until you explicitly grant it. Users in an unmodified, newly created group do not have access to anything.

On the Access Configuration sub-tab of the group, select each object in the Object panel for which you want to set permissions, then select the permissions and click **Save** in the Permission panel.

The Access Summary sub-tab contains a list of all available permissions for the group. The Access Controls panel contains a list of permissions for objects such as tools, modules, and forms. The UX Model Access Controls panel contains a list of permissions for UX models.

If you do not set specific permissions when you create a group, the Access Summary sub-tab will not contain any data. You must save the permissions for each object that you want to be associated with the group. If the permission that you want is already selected, then you do not need to click **Save**.

Objects such as forms, tabs, and sections that have Inherit from Parent selected for permissions do not display on the Access Summary sub-tab. Objects such as tools and modules that have No Access selected for permissions do not display on the Access Summary sub-tab unless you explicitly save the permissions on the Access Configuration sub-tab.

You can export the contents of the Access Summary sub-tab to a tab-delimited text file, which can be easily copied and pasted into a spreadsheet for convenient viewing. The file is saved on the server in a folder called `\Groups` under `\userfiles` in the IBM TRIRIGA server installation directory, for example, `C:\Tririga\userfiles\Groups`. The syntax of the file name is `ACL_<group_name>.` A notification is set to the user's Notifications portal when the export is completed.

Module access

When a module is selected in the Object panel, two kinds of permissions are shown in the Permission panel: Data Access permissions and Application Access permissions. Data Access uses CRUD credentials to govern how users can access the record as a whole. Application Access governs what users can do to a record and the actions that can be performed on a record.

You may select one Data Access permission. Initially, No Access is selected. No Access means just what it sounds like. The other levels of data access permissions allow progressively more access to records, all the way up to Read, Update, Create, and Delete, which allows members of the group to look at existing records, change the contents of records, create new records, and delete records.

The forms listed in a group are organized by module. The data access permission that you select for a module is used as the default data access permission for the forms in the module. If you do not select a data access permission for a form, then the form inherits its permissions from the module.

The Application Access permissions govern which actions can be performed on a record. Each permission corresponds to an action in a state transition family. If you select the check box next to an action permission at the module level, members of the group can perform that action using any form in the module.

Tool access

When a tool is selected in the Object panel, the members of the group can either have full access to a tool or no access. Tools include the Data Modeler and Workflow Builder. The Report Manager has more permission choices.

Security Manager is not listed in the Object panel. To set permissions for the Security Manager, select **Group** in the Object panel. If you do not want a user or a group to have access to the Security Manager, select `No Access` in the Permission panel.

Form access

When you click the plus sign icon to the left of a module name, it shows the forms in the module.

When a form is selected in the Object panel, the Permission panel contains Data Access, Application Access, and Form Action Access permissions.

You may select one Data Access permission. Initially, *Inherit from Parent* is selected. The *Inherit from Parent* access permission sets the access level of the form, tabs, or sections. In a situation where the user is a member of multiple security groups, *Inherit from Parent* means that this group inherits permissions from its group parent, not from the collection of multiple groups.

No Access means just what it sounds like. The other levels of data access permissions allow progressively more access to records, all the way to *Read, Update, Create, and Delete*, which allows members of the group to look at existing records, change the contents of records, create new records, and delete records.

The Application Access permissions shown for a form only include actions from the state transitions that the form uses. Members of the group can perform an action on a form's record only if the action's check box is selected in the list for the form or its module.

The Form Action Access permissions shown for a form include all actions added in the form to sections as well as any action fields. Members of the group can perform a form action on a form's record only if the action's check box is selected in the list for the form and if they have *Read, Update, Create, and Delete* access to the form.

Tabs and sections access

When you click the plus sign icon next to the name of a form, you see the tabs that the form contains. If a tab contains sections, it has by a plus sign icon you can click to see the sections.

You can select one of the Data Access permissions. Initially, Inherit from Parent is selected. If the members of a group have No Access to a tab or a section, they cannot see the tab or section.

Model access

You can open the Models tree to assign models to a security group, and assign permissions to a model.

When a model is selected in the Object panel, the Permission panel contains the following Model Access permissions: No Access; Read; Read and Update; Read, Update, and Create; and Read, Update, Create, and Delete. You can select one Model Access permission.

No Access means just what it sounds like. The other levels of data access permissions allow progressively more access to models, all the way to *Read, Update,*

Create, and Delete, which allows members of the group to look at existing models, change the contents of models, create new models, and delete models.

In addition, the actions that are available through the model are shown. The Model Action Access permissions shown for a model include all actions added in the model to sections as well as any action fields. Members of the group can perform an action only if the action's check box is selected and the members have the appropriate access to the model.

Model security data is cached for better performance at runtime. The cache is automatically refreshed when security assignment changes are made in Security Manager. If needed, you can manually refresh the cache in the Administrator Console. Select the **Cache Manager**, then in the Flush a Cache section, select either **Security Scope** or **All Caches**.

Admin group

The Admin Group is a very special group for the IBM TRIRIGA Application Platform. When they have license access, members of the Admin Group are automatically allowed access to all records, models, and to most of the tools in the platform environment whether they have been granted explicit permission to do so or not.

The only tool access that membership in the Admin Group does not confer to its members is the tool access that is granted to user IDs that have an IBM TRIRIGA Application Builder license.

To ensure the security of records in the IBM TRIRIGA Application Platform environment, it is very important to limit the users that are members of the Admin Group to a small number of trusted users.

The Admin Group cannot be copied or deleted.

Organization and geography

You can restrict user access to records based on the relationship between individual records and Organization and Geography. The definition of a group in the Security Manager includes the System Organization and System Geography fields in the Data Access section of the General tab.

Most business objects have a field that is named OrgName and a field that is named Geography Name. These fields are in the business object's General section. These fields are automatically supplied by the IBM TRIRIGA Application Platform. Because the platform automatically adds these fields, they do not appear in the Data Modeler. They do appear in the Form Wizard as part of the layout.

The OrgName field can have as its value any Organization record. The GeographyName field can have as its value any Geography record. The Geography hierarchy and the Organization hierarchy can be accessed in the Portfolio menu.

A new record inherits the System Organization and System Geography values of the currently logged in user as default values. For example, if Sam is logged in and has the values ZetaBank and US for these fields in his My Profile record, then most new records he creates have these values by default.

By default, many dependent child records inherit their System Organization and System Geography values from their parent records. For example, a new clause in a real estate contract inherits from the parent contract.

If a record's System Organization field has a value, the value of the field may restrict the users that can access the record. Users can access a record if they are a member of at least one security group that contains a System Organization value that is the same as or higher in the hierarchy than the organization contained in the record's System Organization field.

If a record's System Geography field has a value, the value of the field may restrict the users that can access the record. Users can access a record if they are a member of at least one security group that contains a System Geography value that is the same as or higher in the hierarchy than the geography contained in the record's System Geography field.

Attention: The logged in user's System Organization and System Geography values do not control any access rights. It is the security groups that the user is a member of that control access rights.

It is possible for a record to not have a value for the System Organization or System Geography fields. If a record's System Organization field has no value, the record is treated as though the value is \Organizations. If a record's System Geography field has no value, the record is treated as though the value is \Geography.

The following table summarizes the relationship between a record's System Organization field and a group's System Organization field.

Table 30. Relationship between a record's System Organization field and a group's System Organization field

	Record System Organization is blank	Record System Organization is \Organizations	Record System Organization is NOT blank
Group System Organization is blank	User in group DOES see record in queries and forms	User in group DOES NOT see record in queries or forms	User in group DOES NOT see record in queries or forms
Group System Organization is \Organizations	User in group DOES see record in queries and forms	User in group DOES see record in queries and forms	User in group DOES see record in queries and forms
Group System Organization is not blank	User in group DOES see record in queries, but does NOT see record in forms **Note:** In UX apps, records shown to users are records in queries	User in group DOES NOT see record in queries or forms	User in group DOES see record in queries and forms if the value in the group System Organization is at the same hierarchy level as or at a higher level than the value in the record System Organization

The following table summarizes the relationship between a record's System Geography field and a group's System Geography field.

Table 31. Relationship between a record's System Geography field and a group's System Geography field

	Record System Geography is blank	Record System Geography is \Geography	Record System Geography is NOT blank
Group System Geography is blank	User in group DOES see record in queries and forms	User in group DOES NOT see record in queries or forms	User in group DOES NOT see record in queries or forms
Group System Geography is \Geography	User in group DOES see record in queries and forms	User in group DOES see record in queries and forms	User in group DOES see record in queries and forms
Group System Geography is not blank	User in group DOES see record in queries, but does NOT see record in forms **Note:** In UX apps, records shown to users are records in queries	User in group DOES NOT see record in queries or forms	User in group DOES see record in queries and forms if the value in the group System Geography is at the same hierarchy level as or at a higher level than the value in the record System Geography

After you add, update, or delete a System Organization or a System Geography, clear the Security Scope cache in the Administrator Console. For more information about the Administrator Console, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

Group best practice

Your group structure can be difficult to manage if your groups combine System Organization, System Geography, and application security in the same group. The best practice is to use multiple groups and layer groups for each user.

For example, Group 1 defines System Organization security as \Organizations\Greenpoint. Group 2 defines System Geography security as \Geography\North America\United States. Group 3 defines a level of application security as Read access to triBudget. You assign a user to Group 1, Group 2, and Group 3, and the user has the combined security of all groups.

Forms, reports, and queries

The ENFORCE_GUI_LEVEL_QUERY_SECURITY property in the TRIRIGAWEB.properties file controls form-level security in reports and queries. When this property is N, the default, queries and reports do not consider form-level security.

When this property is Y, the user's group controls whether the user can access records in reports and queries. If a user does not have access to records being queried, no results are returned. If at least one of a user's groups has at least read

access, the data is returned to the form. For module queries, only data for the forms in the module to which the user has access are returned.

Projects

One of the business objects that is standard with the IBM TRIRIGA Application Platform is triCapitalProject in the triProject module. triCapitalProject records are used to track a set of related activities and their associated financial data. The triCapitalProject business object is useful for applications that manage projects.

triCapitalProject records have a security-related purpose that is independent of project management. triCapitalProject records can be used to restrict who can access records. They can be used to filter which records are visible to the user.

Access to records associated with a triCapitalProject record is limited to a specified set of users. You can allow some users to view but not change records that are associated with a Capital Project record.

Company projects and active projects

The IBM TRIRIGA Application Platform comes with one standard triCapitalProject record that has the name Company Level. There are a few things about the Company Level project that make it special.

- You may not edit or delete the Company Level project.
- Users are not denied access to a record because the record is associated with the Company Level project.
- A user must have at least one projects license to change their active triCapitalProject.

When a user signs on to the IBM TRIRIGA Application Platform, the user's session with the platform is associated with a triCapitalProject record. This triCapitalProject record is referred to as the active project. The name of the active project is shown above the menu bar.

When a user clicks the Find Project icon, if the user has permission to change the active project to a triCapitalProject record other than Company Level, a list of other triCapitalProject records pops up. The current project can be changed to one of those capital project records. To change the appearance of the list of capital projects that is popped up, update the query named Portal Project Search in the Report Manager.

Clicking the Company button sets the current project to the Company Level project.

When a user signs out, the system saves their active project. The next time that user signs in, the user starts in that project.

When projects are displayed in a grid in the user's portal, the user can change the active project by clicking the project switching icon to the left of the project name. The project switching icon is only available when the record corresponds to a capital project record and the user is granted project security privileges.

When the user clicks the project switching icon for a record in a non-default project, the user's active project is switched to that record's project. When the user

clicks the project switching icon for a record in a default project, the user's active project is switched to the Company Level, if not already there. If a user's project is changed, the current project name at the top of the home portal is updated and the portal is refreshed.

The active project is significant in two ways.

* Every record has an internal association with a triCapitalProject record. When a user interactively creates a record, it is internally associated with a triCapitalProject record. When a workflow creates a record, the workflow can either explicitly specify a triCapitalProject record for the new record or use the triCapitalProject record that is active for the current user.
* The data that is returned by reports and queries can be limited to the active project. To restrict the data that is returned by reports and queries to only projects to which a user has security access, set the USE_PROJECT_SECURITY property in the TRIRIGAWEB.properties file to Y. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

When a user creates a new record, the Capital Project that is associated with that record depends on the following factors:

* The project for a top-level record is the user's current active project.
* If the record is a new child record and the Project Containment Disabled property in the association definition for the dependent business object is off (the default), the record's project is that of its parent record. If the Project Containment Disabled property is on (selected), the record's project is the user's current active project.
* The project for an existing record that is made to be the child of another record is the project of that other record.

A record is considered to be a child if one of the following is true:

* It is associated to a record via a dependent association.
* It is a hierarchical child through a platform hierarchy. An exception is the child of root, which is treated as a new record.
* It is a record in a dependent section.

When requests come to the server, the user's active project is relevant because the active project drives among other things what the project context is when synchronous workflows run. To have the user's current active project always be the active project no matter what, set the RECORD_PROJECT_CONTAINMENT property in the TRIRIGAWEB.properties file to N. For information about the properties files, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

When using projects, be aware that when a user has access to multiple projects, that user may create records that should be part of one project while in a different project. This can result in the wrong users having access to the records, or in users who need access being denied it.

Project security setup

triCapitalProject records are managed using the Capital Projects page. To access the Capital Projects page, go to Projects > Capital.

To create a new capital project, click **New Project**. To edit a capital project record, click the hyperlinked name. To delete a capital project record, select the check box next to it and click **Delete Project**.

When you are creating a capital project record only for security purposes, the only field in the General tab that you need to complete is the Name field.

All of the security-related fields in the capital project form are in the Security tab, which is divided into sections that control access to records associated with the capital project record.

Users that are members of groups listed in the Group Access section have access to records internally associated with the capital project record. If a user ID's only access is as a member of groups that have their Read Only check box selected, the user can view but not change records that are internally associated with the capital project. Use the Find action to add a group to the Group Access section.

User IDs can also have access to a record internally associated with a capital project record if they are listed in the User Access section. If a user ID's Read Only check box in the User Access section is selected and the user does not belong to any groups listed in the Group Access section that do not have their Read Only check box selected, the user can view but not change records that are internally associated with that capital project record. Use the Find action to add a user to the User Access section.

For this security to be enforced, you must click the Activate action on the capital project form. Depending on the approval requirements for capital projects, the project may need to be approved. After the project is in Active status, the security is enforced. For more information about creating approval processes, see the *IBM TRIRIGA 10 Application Administration User Guide*.

Workflows and projects

When a synchronous or subflow workflow starts, its active project is always the current project. If initiated by a user, the current project is the active project of the user who initiated the workflow. If the workflow was launched from a Call Workflow task, the initial project is the project context at the time the Call Workflow task ran. Note that asynchronous workflows run with a setting of `no project'.

The active project does not affect the ability of a workflow to access records, because workflows can access all records without being affected by project-related restrictions. However, Query tasks can filter results based on the active project. See "Set Project Task" for details.

Initially, new records are not assigned to any project. When a workflow task creates a new record, you can specify that the new record should be associated with a triCapitalProject record in the following ways:

- A workflow task can set the project on a record to a specified project. The specified project can be either the Company Level or provided by the project of another record.
- A workflow task can specify that a new record can be associated with the same triCapitalProject record as another record.
- A workflow task can specify that a new record can be associated with a triCapitalProject found through a Query task or another task.

Use a Set Project task to change the workflow's current project context or to change a record's project.

Reports

There are three types of reports: System Reports, Community Reports, and My Reports. You control access to each report type in the Access tab of each group. Select Report Manager in the Object panel. These permissions control access to the tabs in the Report Manager.

If a group has No Access to system reports, users in that group do not see the System Reports tab in the Report Manager. If a group has Full Access to system reports, users in that group can manage (create, edit, copy, delete) and share any system report.

Attention: Be careful to which groups you give system report Full Access because users with such access can potentially break the application with this access level.

If a group has No Access to community reports, users in that group do not see the Community tab in the Report Manager. If a group has View Access to community reports, users in that group can run community reports and copy those reports into their own My Reports tab. Note that community reports are system reports that have been determined to be useful for end users to be able to copy and modify for their own use as My Reports. To manage a community report, a user must belong to a group with access to system reports.

If a group has No Access to My Reports, users in that group do not see the My Reports tab in the Report Manager, and users in that group cannot create or manage their own personal reports. If a group has Manage permission to My Reports, users in that group can create, edit, copy, and delete their own personal reports. If a group has Manage and Share permission to My Reports, users in that group also can share their reports with other groups.

The Copy as Community Report action, on the My Reports tab in the Report Manager, is available only to users belonging to a group with Full Access to system reports. The Share Report action is only available to users belonging to a group with Manage and Share permission for My Reports.

In addition to controlling the access to the tabs in the Report Manager through the Security Manager, you also can specify an additional granularity of security for individual reports, based on the type of report, through the Security sub-tab in the report's definition.

For system reports, there is no Security tab. Groups either have full access or no access to system reports.

For community reports, the Security tab in the General tab can be used to restrict view access to a community report to the specified groups. If the Security tab is empty, the community report is available to all groups that have access to the Community tab.

My Reports also has a Security tab. If a user belongs to a group with Manage and Share permission for My Reports, that user can share personal reports with members of other groups. By default, shared groups have View Access, i.e., they can run the report. However, the owner of the report also can specify more

granular levels of access to shared reports. If the user chooses, the user can give one of the groups All access, which effectively grants that group the same privileges as the user has as the owner. The owner of a My Report always has view, edit, delete, and copy access to that report.

The Report Manager Administration tab displays only for members of the Admin Group. The Administration tab lists all My Reports for all users. In this tab, an administrator can change the owner of a report, or copy, delete, or share My Reports. In addition, an administrator can open individual My Reports and update the Security sub-tab as needed.

For information about the Report Manager, see the *IBM TRIRIGA Application Platform 3 Reporting User Guide*.

Audit trails

Some records may be considered so sensitive that you want to keep a record of all changes made to them. A record of all changes that were made to a record is called an audit trail.

These are the most common reasons for wanting an audit trail.
- Users with legitimate access to records may be tempted to do something dishonest. This often happens when the records in question are used to represent money or other things that are worth money.
- There is an undesirable legal consequence if an improper change to, or action on, a record is performed.
- There is a regulation or standard that requires your company to track changes to records.

If any of these concerns is relevant to a record, it is important to have an audit trail that tells who did what to each such record and when they did it. Having an audit trail is a disincentive for dishonesty, because it makes it easier to discover dishonest actions. An audit trail removes speculation about what actually happened.

An audit trail is an historical record of everything that happened to a record. Because it is an historical record, the information that is in an audit trail cannot be altered or deleted.

There are two main reasons for not wanting an audit trail:
- Maintaining an audit trail increases the amount of time that an operation takes. Saving the audit records takes additional time.
- Maintaining an audit trail can greatly increase the amount of storage that an application uses. In addition to needing space to store records, you need space to store the audit trail. The space that is needed to store the history of everything that ever happened to a record can be significantly larger than the record itself.

Support for audit trails is built directly into the IBM TRIRIGA Application Platform. Though it is possible to implement audit trails as part of an application's business logic, IBM TRIRIGA strongly suggests that you use the audit trail support that is built into the platform. There are at least three reasons for this:
- An audit trail that is maintained by the platform is more secure than an audit trail maintained by the application. An audit trail based on an application's

business logic is only as reliable as the application itself. If there are bugs or mistakes in the application, an audit trail that is maintained by the application may not be accurate. The accuracy of an audit trail that is maintained by the built-in audit trail mechanism is unaffected by bugs or mistakes in an application.

- It takes a lot more work to build the logic for an audit trail into an application than it does to use the audit trail support that is built into the platform.
- Although all audit trail mechanisms increase the amount of time it takes an application to do things, an audit trail that is maintained by the platform's built-in mechanisms has less impact on performance than an audit trail that is maintained by an application.

You set the options for maintaining an audit trail for records that are created from a business object by setting the relevant properties of the business object with the Data Modeler.

The following properties of a business object control the generation of an audit trail for records that are created from the business object.

Audit Actions

> If the Audit Actions check box is selected, when a user clicks a sub action in a form that is editing a record that is created from this business object, an audit record is created.

> Selecting the Audit Actions check box enables the creation of audit records that identify the sub action that was performed, the user ID of the user that clicked the sub action, and when the sub action was clicked. Selecting the Audit Actions check box only enables the generation of audit records for sub actions that you specify should be audited by selecting their Log check box. For audit records to be generated for a particular sub action, the Log check box in the sub action's properties must be selected.

> Select the Audit Actions check box for the Request for Information (RFI) metrics and for reports in IBM TRIRIGA Workplace Performance Management products.

Audit Access

> If the Audit Access check box is checked, every time a user utilizes a form to view the contents of a record that is created from this business object, an audit record is created.

Audit Interactive Data

> If the Audit Interactive Data radio button is selected, every time a user uses a form to change the contents of a record that is created from this business object, an audit record is made. The audit record contains the user ID that made the change and the time of the change. The audit record also contains the old value and new value of every changed field.

Audit All Data

> If the Audit All Data radio button is selected, an audit record is made for every change to a record created from this business object. This includes changes that are made by a user in a form, due to a formula calculation or rollup calculation, or by a workflow. This does not include changes that are made if the record is modified by an editable query.

> Selecting the Audit All Data property can affect performance.

Require Explanation

If the Require Explanation check box is selected, when a user uses a form to change the contents of a record that is created from this business object, the user is asked to enter a reason. If the Require Explanation check box is selected, the Audit Interactive Data property is automatically selected.

The reason that is given for the changes to the record is included in the audit record along with the other information about the change.

Whether you are creating a new sub action or editing it, after you click the appropriate action you check the Log check box to enable auditing of the sub action.

Configuring the generation of audit records is only half of the IBM TRIRIGA Application Platform's built-in mechanism for maintaining an audit trail. The other half is a mechanism for viewing audit records.

If a form is used to edit records for which audit records are generated for data access or changes, you need to update that form to show the Audit tab. To do so, revise the properties of the form for the business object where you would like to see the audit data and check Show Audit Actions.

After the form is published, the form has a tab that is labeled Audit. The purpose of the Audit tab is to display all the audit records that contain data changes that were made to the record.

Three columns display in an Audit tab.

Modified By
The user ID that made the changes.

Comment
The reason the user entered if required to enter a reason for making changes to a record through a form.

Modified On
When the data was changed.

Tip: You may not want all users to see the information in the Audit tab or the Audit Actions tab. Access to those tabs is controlled through group permissions, just like any other tab.

To see details of the data changes that were made, click the hyperlinked audit record. The Audit Details form shows the user that made the change, the name of the field that was changed, the value of the field before it was changed, and the value of the field after it was changed. If changes were made while the record was created, the fields had no previous value, and the old value is blank.

If a form is used to edit records for which audit records are generated as a result of clicking sub actions, the form has an Audit Actions tab. The purpose of the Audit Actions tab is to display the audit records generated to record actions that were performed on the record.

The Audit Actions tab shows every recorded sub action that was clicked in the record. The Audit Actions tab shows the label of the sub action, the user that clicked the sub action, and when the sub action was clicked. To allow a user to view this data, you need to update that form to show the Audit Actions tab. To do so, revise the properties of the form for the business object where you would like to see the Audit data and check Show Audit Actions.

Workflow instances

Depending on your auditing requirements, it may be good enough to know what sub actions were clicked for records, when those sub actions were clicked, and who clicked them. If you need to know what workflows were started as a result of the sub actions and what the workflows did to records, you need more information than is in the Audit Actions tab.

The IBM TRIRIGA Application Platform provides a way for you to examine the actual workflows that ran in response to clicking sub actions and to see how the workflows ran. This feature is based on the fact that workflows run only in response to an action or system event happening to a record. You can see the workflows that ran in response to something happening to specific records.

To be able to see workflows that were started for a record, you need to:
- Decide which workflows are of interest.
- Specify that information about each running of those workflows should be saved.
- Include a tab in appropriate forms for viewing the records of the workflows that have run.

In order for information to be saved each time that a workflow is run, the Save Workflow Instances check box in the workflow's properties must be selected. The Save Workflow Instances check box is discussed in "Start Task".

If a workflow's Save Workflow Instances check box is selected, every time the workflow runs, a copy of the workflow is kept with information about how it ran. This saved information about a single running of a workflow is called a workflow instance.

You can view all saved workflow instances for a workflow by using the Workflow Builder. To do this, select the workflow of interest and then click the List All Instances action.

For researching what a workflow did when it was run a particular time for a particular record, you include a Work Flow Instance tab in the forms that are used to access the type of records that are of interest. To configure a form to include a Work Flow Instance tab, edit the form's properties so that its Show Workflow Instance check box is selected. The Show Workflow Instance check box is discussed in "Form Properties".

The Work Flow Instance tab shows the name of each workflow that ran, the current status of the workflow, and when the workflow started. There is no indication of what started the workflow. You need to use the times sub actions were clicked and the times that workflows were started to match them up.

The statuses showing in the Work Flow Instance tab are as follows:

For synchronous workflows:
- Active (the workflow is currently running)
- Completed (the workflow has completed)

For asynchronous workflows:

- Aborted, Aborted-Warn (the workflow was aborted using the Stop capability in the Administrator Console)
- Active (the workflow is currently running)
- Completed, Completed-Warn (the workflow has completed)
- Failed (the workflow encountered an error and could not continue; information about the problem was written to the System Log)
- Skipped (the workflow would have run, but the Start Conditions were not satisfied)
- Stopped, Stopped-Warn (the workflow ran a Stop task)
- Waiting, Waiting-Warn (the workflow is waiting for a user (user action or approval task))

If Warn is on the status it means that a problem was encountered, information was written to the System Log, and the workflow continued processing.

The workflow names in the Work Flow Instance tab are hyperlinks. If you click a workflow name, the workflow editor shows the actual workflow that ran. Even if the workflow changed after this run, you see the actual version of the workflow that ran.

The workflow editor displays the workflow differently from when you access it through the Workflow Builder. Workflow tasks that were performed are highlighted with a green dashed border. Looking at the highlighting, you can see the path that was taken through the workflow when it ran.

Tip: The workflow trace that you see in the Work Flow Instance tab can be useful in debugging a workflow, since dashed green highlighting shows the path that the workflow followed.

If a workflow step was started but has not finished, it is highlighted with a red dashed border.

Workflow instances subject to cleanup agent

The IBM TRIRIGA Application Platform does not keep workflow instances indefinitely. The Cleanup Agent discards workflow instances after a specified number of days. The number of days that the platform keeps workflow instances is determined by the platform's configuration properties.

The IBM TRIRIGA Application Platform configuration properties are discussed in the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Audit trail integrity

By auditing the actions that users perform on a record and the changes that users make to its data, it is possible to have a detailed and accurate audit trail with which you can determine exactly how the record came to have its current data and state.

There are some rules that you must observe to ensure the integrity of audit trails that are produced by the IBM TRIRIGA Application Platform:

Make changes to an application only when users are not using it.

It is a good idea to keep users from using an application while it is being changed, just to be sure of getting consistent results. Another reason to keep users from using an application while it is being changed has to do with accurately interpreting audit trails.

It is important to know whether the consequences of clicking a sub action happened before or after a change to the way an application works. If users are using an application at the same time it is being changed, it may become difficult to know if an action and its consequences happened before or after a change. If the change affected which workflows are started by the action, it may not be possible to know which workflows were started by an action.

Restrict access to the Data Modeler.

Even if you are only auditing data changes, it is still important to restrict access to the Data Modeler.

Through the Data Modeler, it is possible to turn off generation of audit records for all records that are created from a business object and then later turn the generation of audit records back on. While the generation of audit records is turned off, a user can change data in records without the change being recorded in an audit trail. Also, there is no record in the audit trail that generation of audit records was turned off or on.

Security settings should prevent users from deleting records for which you are keeping an audit trail.

When a record is deleted, the audit records associated with the record are also deleted.

Instead of designing audited records to be deleted, design them to have a retired state that keeps them from being seen or used in most contexts.

Avoid designing records that are likely to accumulate many audit records.

There is no way for workflows to access audit records. They must be interpreted manually. If a record accumulates a large number of audit records, it may not be practical to interpret the audit trail.

To avoid this problem, try to design records so that they do not accumulate a large number of audit records.

Changes to records that have an audit trail should only be initiated from a form.

Data audit records are only created by changes that are made to a record through a form.

Audit records are not created when records are created or modified by the Data Integrator or by other means.

If you cannot follow these rules, and you must have a secure audit trail, you can use your database manager to keep an audit trail. The platform creates a database table for each business object. Records that are created from a business object are stored in the database table that corresponds to the business object used to create the record. The details of auditing database tables vary with the database software used. Consult your database administrator or database documentation for the specifics.

Queries control visible records

Queries with filtering can be used to control the records that are visible in navigation items, query sections, and through Find and Add actions. You can craft queries so that users only see the records that you want them to see.

Limiting user access to records this way has one important advantage. It gives you a great deal of flexibility in how you filter a user's view of records. However, it has enough drawbacks that you should only use this technique as a last resort. Here are some of the problems with this technique:

- Because multiple queries must be made consistent to present a consistent view of the data, errors are more likely when you use this security technique than with other security techniques.
- Because the policy that controls the records that a user is allowed to see may be replicated in multiple queries, this technique makes it difficult to keep the policy consistent when you change it.

Workflows bypass security

Workflows are not subject to any security restrictions. If you want filtering that is based on organization or geography, you must explicitly specify those restrictions in the workflow or in queries that the workflow uses.

Chapter 14. Offline Microsoft Excel spreadsheets

An IBM TRIRIGA Offline form is a Microsoft Excel spreadsheet with fields on the sheet mapped to fields in an IBM TRIRIGA business object. Through the mappings on this sheet, data in the IBM TRIRIGA application can be exported into an Excel sheet for review and updates, and data can be collected into the spreadsheet for import into the IBM TRIRIGA application. After the fields on the sheet are mapped to IBM TRIRIGA business object fields, workflow tasks can process this data in the application. This is useful for people who want to work with data in spreadsheets outside of the platform environment.

In order for the IBM TRIRIGA Application Platform to copy data into or out of an Excel spreadsheet, the spreadsheet must be in a binary field of a record. There are specialized workflow tasks for copying data into or out of a spreadsheet. In order for the specialized workflow tasks to work with an Excel spreadsheet, the spreadsheet is required to have a special organization.

Offline forms are best suited for handling small or moderate amounts of data. For large amounts of data, use one of the IBM TRIRIGA integration tools.

Note: If you are using the IBM TRIRIGA Connector for Offline Forms via Microsoft Exchange 2007, be sure to configure Plain text logon (Basic authentication) in Microsoft Exchange > Server Configuration > Client Access > IMAP4 Properties > Authentication tab.

The process flow in the following figure describes the outbound process.
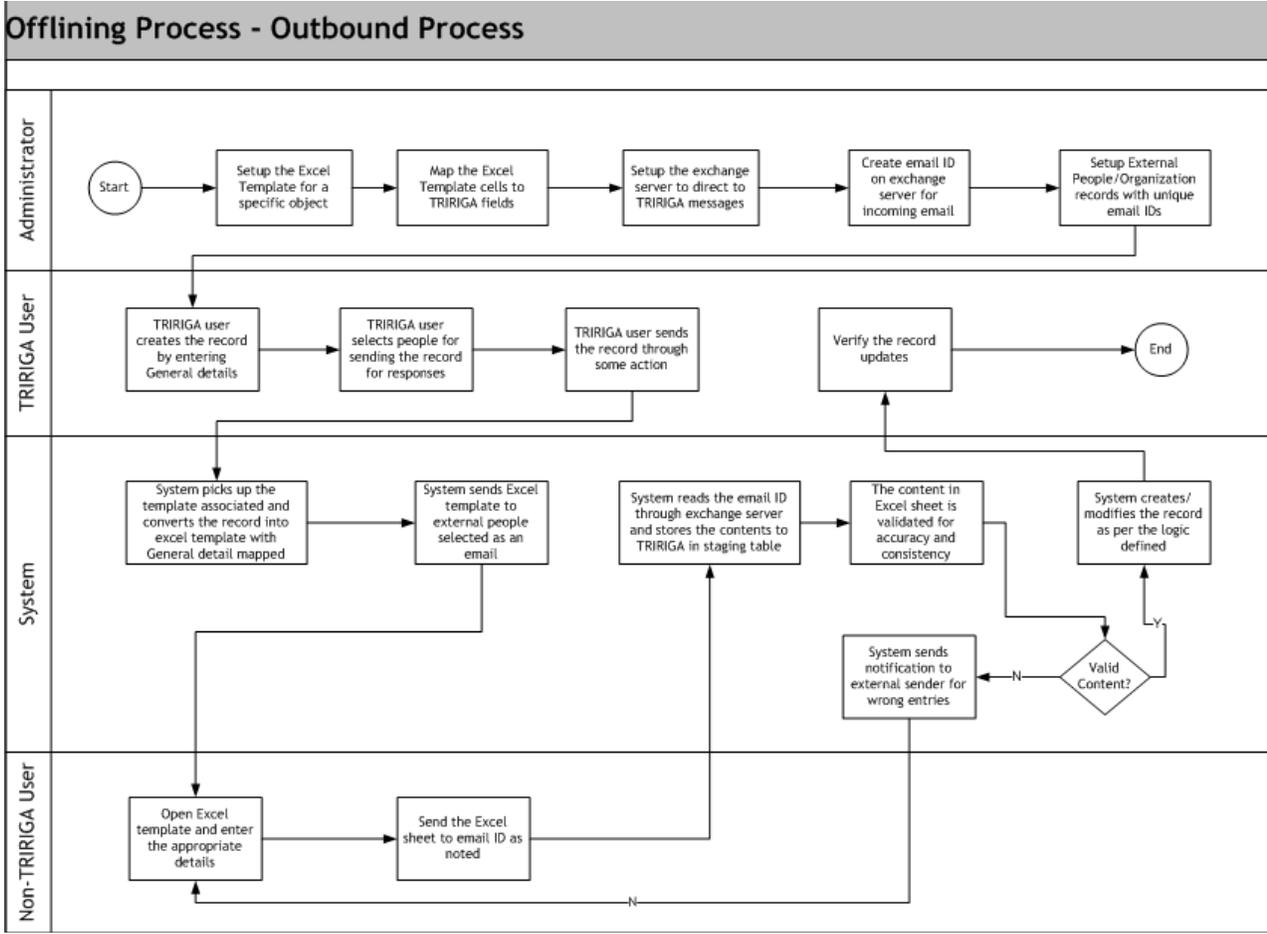
Figure 131. Outbound Process Flow - Round trip

The process flow in the following figure describes the inbound process.

Offlining Process - Inbound Process
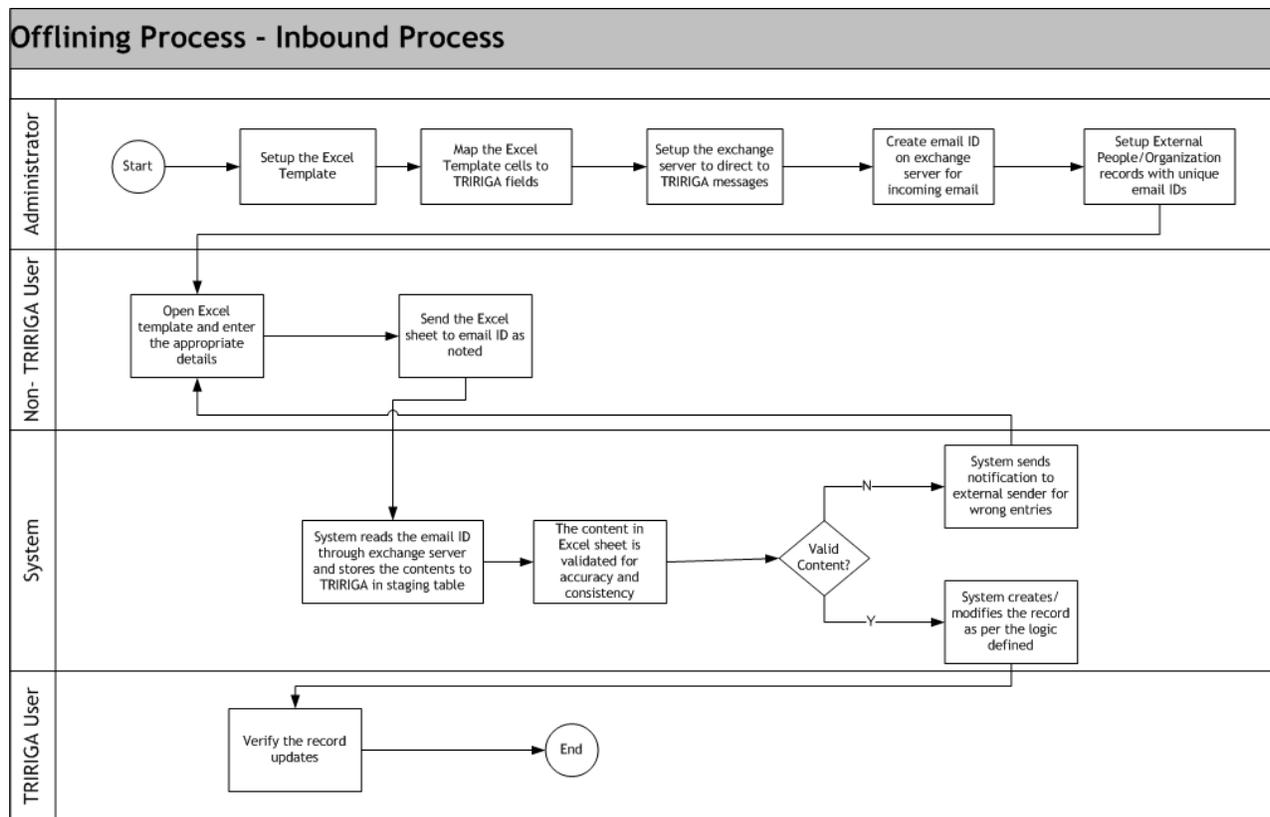
Figure 132. Inbound Process Flow - Staging table validation

Several components make up an IBM TRIRIGA Offline form. The steps to create an IBM TRIRIGA Offline form include the following:

- Create the Excel template for a specific business object.
- Create a Binary field definition in that business object. This is where the offline form instances will be stored a run time. See "Binary fields".
- Define the IBM TRIRIGA Object Map to map the Excel template cells to IBM TRIRIGA form fields. See "IBM TRIRIGA object map".
- Upload the Excel template with the IBM TRIRIGA Object Map into IBM TRIRIGA. The Excel template records for the standard IBM TRIRIGA applications are in the triOfflineContent business object.
- Use the Populate File workflow task to populate data from IBM TRIRIGA into the Excel sheet. See "Populate file task".
- Use the Distill File workflow task to distill the Excel data into the IBM TRIRIGA application. See "Distill file task".

IBM TRIRIGA object map

To use a Microsoft Excel spreadsheet with workflow tasks that copy data into or out of it, the spreadsheet must contain a worksheet named *Tririga Object Map*. The rows of this worksheet specify how to map values from the Excel spreadsheet into records or vice versa. The relationships specified in this worksheet between data in Excel spreadsheets and data in records are collectively called a mapping. The following figure shows an example of an IBM TRIRIGA object map.

E7　='Portfolio Budget'!B6

	A	B	C	D	E	F	G
1	Tririga to Excel						
2		Field	Record Information	triIdTX			
3		Field	Record Information	triForecastYearCL			
4	Selector	Organization Serviced by				Organization	Organization
5		Field	General	triParentLevel0TX			
6	End Selector						
7		Field	Record Information	triGeographyLookupTX			
8		Field	Record Information	triDescriptionTX			
9	Selector	Has Planning Item				triPortfolioPlanning	triPlanningItems
10		Field	Record Information	triBudgetItemCL			
11		Field	Record Information	triUnitsUO			
12		Field	Record Information	triPeriod1ActualNU			
13	End Selector						
14	Selector	Has Expense Item				triPortfolioPlanning	triExpenseItems
15		Field	Record Information	triBudgetItemCL			
16		Field	Record Information	triCurrencyUO			
17		Field	Record Information	triPeriod1ForecastNU			
18	End Selector						
19	Selector	Has Savings Item				triPortfolioPlanning	triSavingsItems
20		Field	Record Information	triBudgetItemCL			
21		Field	Record Information	triCurrencyUO			
22		Field	Record Information	triPeriod1ForecastNU			
23	End Selector						
24	End						
25							
26	Excel to Tririga						
27		Field	Record Information	triDescriptionTX			
28	Association	Has Planning Item	triPortfolioPlanning	triPlanningItems	triCreate		
29		Field	Record Information	triBudgetItemCL			
30		Field	Record Information	triUnitsUO			
31		Field	Record Information	triPeriod1ActualNU			
32	End Association						
33	Association	Has Expense Item	triPortfolioPlanning	triExpenseItems	triCreate		
34		Field	Record Information	triBudgetItemCL			
35		Field	Record Information	triCurrencyUO			
36		Field	Record Information	triPeriod1ForecastNU			

Portfolio Budget　Planning Items　Expense Items　Savings Items　**Tririga Object Map**

Figure 133. IBM TRIRIGA object map

The first non-empty column of each row determines the purpose served by the rest of the columns in the row. For example, if the first column in a mapping row contains the value *Field*, then the row contains information on how the value of a particular cell or cells in the Excel spreadsheet map to a field in a record. This value in the first non-empty value of a mapping row is called the row's *tag*. In the previous example, one could just refer to the *Field* tag.

Empty rows in the worksheet are ignored, as are empty cells before the tag on each row. On a given row, the meaning of cells after the tag depends on the type of tag for that row. The cells to the right of a row's tag are called *property* cells, because they refer to the properties that define a particular tag. Each tag defines some number of property cells, whose position is relative to the tag cell. For example, if the tag that defines three properties is in column C, the properties for the tag will be in columns D, E, and F. If the tag were in column A, the properties would be in columns B, C, and D.

Some tags in the mapping worksheet have a corresponding end tag. This mechanism serves to enclose other tags between the given tag and its corresponding end tag, giving the mapping a hierarchical structure. Mapping in

either direction starts with a root row. Various tags in the mapping let the mapping engine navigate to other objects related to the root through associations. Nesting these tags allows navigation through multiple association hops. This document refers to parent and child rows to describe these relationships. For example: if we follow an association from the root row to row *A*, the root row is the parent of row *A* and row *A* is a child of the root row. If we then follow an association from the row *A* to row *B*, *A* is the parent of *B* and *B* is the child of *A*.

All tag, object, field, association names, and the like are case sensitive.

Records to Excel section

This refers to the Tririga to Excel section in the Tririga Object Map on the offline form. This section defines field mapping from IBM TRIRIGA form fields to the offline form. The following describes the tags used for mapping from records to an Excel spreadsheet:

Tririga to Excel

>This tag is the root of the record to Excel spreadsheet mapping and encloses all other tags in the map.
>
>**Properties:**
>>None
>
>**End tag:**
>>End

Selector

>The *Selector* tag finds existing records associated to the parent so that their data can be copied to the Excel spreadsheet.
>
>Optionally, the *Selector* can specify a filter that compares a field value in the records to a value in the map. It also can filter by record type.
>
>Sub-tags of this tag are applied to the records found.
>
>**Properties:**
>>- *Association Name* - The name on the parent side of the association.
>>- *Field Name* - The name of the field in the child record to use in the comparison.
>
>**Optional Properties:**
>>- *Module Name* - If this is specified, only records created by business objects in the named module are included. If present, the value must be in the fifth cell to the right of the *Selector* tag.
>>- *Object Type Name* - Only records created by the specified business object are included. If present, the value must be in the sixth cell to the right of the *Selector* tag.
>>- *Query Section Name* - Specifies the query section from which to populate data. If present, the value must be in the seventh cell to the right of the *Selector* tag, or the next to last column of the Selector row. If specified, you must provide a *Tab Name* property. *Query Section Name* and *Tab Name* maintain the order of data in a query section. If specified, the populate retrieves from the query defined in the query section using the current record as the context record.

- *Tab Name* - Specifies the name of the tab on the form that contains the query section identified in *Query Section Name*. If present, the value must be in the eighth cell to the right of the *Selector* tag, or the last column of the Selector row. *Query Section Name* and *Tab Name* maintain the order of data in a query section. If specified, the populate retrieves from the query defined in the query section using the current record as the context record.

End tag:
> End Selector

Field

This tag defines how fields in records are mapped to cells in the Excel spreadsheet. This tag can appear anywhere in the mapping hierarchy. The field mapping is applied to the records that correspond to that level in the hierarchy.

If there is more than one current record because the previous *Selector* tag found multiple records, this tag will duplicate the row of the referenced cell for each record and map the field from each record into a cell in the copied row.

This action is coordinated with the sibling *Field* tags in the mapping. For example: Suppose the previous *Selector* tag found three records. Also suppose there are two *Field* tags below the *Selector* that reference cells in the same row of the document. The first *Field* tag maps field *x* to cell *G4* and the second *Field* tag maps field *y* to cell *H4*.

The mapping engine will make two copies of row *4* (plus the original row makes three rows total) and insert them into the document as rows *5* and *6*. It will then map fields *x* and *y* from the first record into cells *G4* and *H4*, fields *x* and *y* from the second record into cells *G5* and *H5*, and fields *x* and *y* from the third record into cells *G6* and *H6*. Use this mechanism to create line items in the Excel spreadsheets from line items in records.

Properties:
- *Section Name* - The name of the section that contains the field to be mapped. For smart section fields, specify the BO smart section name, for example, triParkingClause.
- *Field Name* - The name of the field in the current record(s) from which data will be mapped.
- *Cell Reference 1* - A reference to the cell to which the data will be copied.

End tag:
> None

Excel to records section

This refers to the Excel to Tririga section in the Tririga Object Map on the offline form. This section defines field mapping from the offline form to IBM TRIRIGA form fields. The following describes the tags used for mapping from an Excel spreadsheet to records:

Excel to Tririga
> This tag is the root of the Excel spreadsheet to record mapping and encloses all other tags in the map.

Properties:
>None

End tag:
>End

Association

>This tag tells the mapping engine to create new child records from a specified business object, trigger a specified action on the new records, and associate them with the parent record. *Field* tags within the *Association* tag define how data in the Excel spreadsheet will be mapped into the newly-created child records.

>If *Field* tags contained within the *Association* tag reference multiple values, then the *Association* tag will create multiple records. The number of records created is equal to the maximum number of values referenced by the enclosed *Field* tags.

>**Properties:**
>- *Association Name* - The name to put on the parent side of the association.
>- *Module Name* - The name of the module that contains the business object that will be used to create records.
>- *Object Type Name* - The name of the business object that will be used to create records.
>- *Action* - The name of the action to trigger on the newly- created records.

>**End tag:**
>>End Association

Selector

>This tag is similar to the *Association* tag except that no new child records are created. Instead, the *Selector* finds existing records associated to the parent so they can be modified. *Field* tags within the *Selector* tag define how data in the Excel spreadsheet will be mapped into the child records found by the *Selector*.

>Optionally, the *Selector* can specify a filter that compares a field value in the child records to a value in the map. It also can filter by record type. Use this mechanism to find a single associated child in a list of children.

>**Properties:**
>- *Association Name* - The name to put on the parent side of the association.

>**Optional Properties:**
>- *Field Name* - The name of the field in the child record to use in the comparison.
>- *Module Name* - If this is specified, only records created by business objects in the named module are included.
>- *Object Type Name* - Only records created by the specified business object are included.

>**End tag:**
>>End Selector

Field

This tag defines how values in the Excel spreadsheet are mapped to a single field in a record or records. This tag can appear anywhere in the mapping hierarchy. The field mapping is applied to records that correspond to that level in the hierarchy.

If only one reference property is specified, a single value is retrieved from the referenced cell in the spreadsheet and put into the specified field in the records.

If a second reference is specified, the mapping engine collects all values in the spreadsheet from the first reference (inclusive) to the second reference (exclusive). The collected values will be set to the given field in the records sequentially. This mechanism is most useful inside an *Association* tag.

If a *Field* tag inside an *Association* tag references a range of three values, the *Association* tag creates three child records and sets the first value to the given field in the first created record, second value to the second record, and third to the third. Use this mechanism to extract line items from the spreadsheet to create line item records.

Excel formatted Time fields are correctly distilled into IBM TRIRIGA.

Duration fields can be distilled when they are entered into the Excel spreadsheet in the following format where each element is optional: y Year(s) m Month(s) w Week(s) d Day(s) h Hour(s) m Minute(s) s Second(s). For example, 2 Year(s) 3 Month(s) 4 Day(s) and 5 Month(s) 2 Week(s) 7 Hour(s) 15 Minute(s).

Properties:
- *Section Name* - The name of the section that contains the field to be mapped. For smart section fields, specify the BO smart section name, for example, triParkingClause.
- *Field Name* - The name of the field to which data will be mapped.
- *Cell Reference 1* - A reference to the cell that contains the data to be put into the field.

Optional Properties:
- *Cell Reference 2* - A reference to a second cell in the document. Specifying a second reference indicates that this *Field* tag is working with a range of values from the document.

End tag:
>   None

For a referenced smart section, such as a live link or reference only section, you can have the distill look up an existing record to use as the association in the section. Instead of the Field name, use the keyword OfflineFindById or they keyword OfflineFindByName and then reference the record Id or the published name of an existing record for the distill to use as the associated record for the section. If you do not use either OfflineFindById or OfflineFindByName, the distill creates a new associated record based on the information in the Excel spreadsheet and uses that for the section.

For a single row smart section when a row already exists on the smart object being distilled to, the distill updates the existing smart section row with the data in the Excel spreadsheet.

For a multiple row smart section when rows already exist on the smart object being distilled to, the distill deletes all existing rows and replaces them with the rows specified in the Excel spreadsheet.

Populate file task

The purpose of the Populate File task is to copy record data into an Excel spreadsheet as defined in the Tririga Object Map. The required organization of the Excel spreadsheet is described in "IBM TRIRIGA object map". This task requires a binary field to store the Excel spreadsheet that contains the mapping and which is the target of the populate.

The Populate File task always uses UTF-8 encoding.

The properties form for a Populate File task is organized into four sections that have these purposes:
- Name and describe the workflow task.
- Specify the record that the data will come from.
- Specify the binary field in a record that will contain a template for the Excel spreadsheet.
- Specify the binary field in a record that will contain the Excel spreadsheet that is created from the data in the record and the template in the other binary field.

Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
>   A description of this task goes in this field.

Populate from record section

The second section of the Populate File task properties form is labeled *Populate from Record*. The purpose of the *Populate from Record* section is to specify the record that data will be copied from.

At the top of the *Populate from Record* section are fields used to identify a target record. The target record is used to determine the record from which data will be copied.

There are radio buttons below the fields. The way that the target record is used to determine the record from which data will be copied depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
>   This is a drop-down list that can have one of these possible values:

>   **Business Object**
>   >   If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee

If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record from which data will be copied.

When the properties form is first displayed, only the currently-selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record from which data will be copied.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target record will be the record from which data will be copied. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, a record associated with the target record will be the record from which data will be copied. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of

the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the record from which data will be copied.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be the record from which data will be copied.

From binary field section

The third section of the form for a Populate File task's properties is labeled *From Binary Field*. This section specifies a binary field in a record that contains the Excel spreadsheet. This workflow task creates a copy of the spreadsheet in the specified field and copies data into the copy. The Excel spreadsheet in the specified field is not modified.

The *From Binary Field* section has two radio buttons to specify the record that contains the template Excel spreadsheet. These radio buttons are labeled:

Workflow Activity
> If this radio button is selected, it means that the record that contains the binary field that contains the Excel spreadsheet will be associated with a preceding workflow task.

Existing Record
> If this radio button is selected, it means that the binary field that contains the Excel spreadsheet is contained in a record that exists now, at the time you are editing this workflow.

The selection of one of these radio buttons determines what appears in the *From Binary Field* section.

When the *Workflow Activity* radio button is selected, the *From Binary Field* section has fields to select a workflow task to which a record that is associated that contains the Excel Spreadsheet.

Under the *Workflow Activity* radio button there are fields used to identify a target record that is used to determine the record that contains the Excel spreadsheet.

There are radio buttons below these fields. The way that the target record is used to determine the record that contains the Excel spreadsheet depends on which one of the radio buttons is selected.

Here are descriptions of the fields used to determine the target record:

Take the
> This is a drop-down list that can have one of three possible values:

> **Business Object**
>> If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.

> **Secondary BO**
>> This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

>> This option is also available if the workflow is launched in response to a *SCHEVENTCREATE*, *SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the system event is the target record.

> **Assignee**
>> If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under the these fields determine how the target record will be used to determine the record that contains the Excel spreadsheet.

When the properties form is first displayed, only the currently-selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are descriptions of the radio buttons:

Use it If this is selected, the target record will be the record that contains the Excel spreadsheet.

Use its Reference

If this is selected, the record that contains the Excel spreadsheet will be referenced by a smart section or locator field of the target record. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, the record that contains the Excel spreadsheet will be associated with the target record. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Initialize from* section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If

-Any- appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the record that contains the Excel spreadsheet will be the target record's parent.

When you select this radio button, a window pops up for you to select the business object that is assumed to have been used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

The selection of a business object represents an assumption about what kind of record the parent will be. Because of this assumption, subsequent tasks will be able to access the parent record's fields. If the actual parent was not created from the assumed business object, the task may fail if the actual record does not have an expected field.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

Below the radio buttons is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that values may be copied from. If the record can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

At the bottom of the *From Binary Field* is a record labeled *Field*. This field has a drop-down list that contains the names of the binary fields in the type of record shown in the *Object Type* field. The binary field selected as the value of the *Field* field is the field that will be expected to contain the Excel spreadsheet.

If you want to initialize records with information that is determined by the application's configuration, you should select the option of using an existing record in the *From Binary Field* section.

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record that contains the template Excel spreadsheet.

Once you have specified the record that contains the Excel spreadsheet, you then specify the binary field in the record that contains the spreadsheet by selecting its name as the value of the *Field* field.

To binary field section

The fourth section of the form for a Populate File task's properties is labeled *To Binary Field*. This section specifies a binary field in a record in which this workflow task will put a copy of the template Excel spreadsheet that has had data copied to it.

Except for the fact that this section is used to specify the binary field that will contain this workflow task's result, the fields in this section function identically to the corresponding fields in the *From Binary Field* section.

Distill file task

The purpose of the Distill File task is to copy offline data into IBM TRIRIGA records as defined in the Tririga Object Map spreadsheet. The required organization of the Excel spreadsheet is described in "IBM TRIRIGA object map". This task requires a binary field to store the Excel spreadsheet that contains the mapping and which is the source of the distill.

The Distill File task always uses UTF-8 encoding.

The form for a Distill File task is shown in the following figure.

Figure 134. Distill File properties

The properties form for a Distill File task is organized into three sections that have these purposes:

- Name and describe the workflow task.
- Specify the record that the data will be copied to.
- Specify the binary field in a record that will contain the Excel spreadsheet.

Here are descriptions of the fields in the first section:

Label   This is the label used to identify this task. This field's text appears on the shape in the drawing that represents this workflow task. Use the standards in "Workflow naming conventions".

Description
A description of this task goes in this field.

Distill to record section

The second section of the Distill File task properties form is labeled *Distill to Record*. The purpose of the *Distill to Record* section is to specify the record that data will be copied to.

At the top of the *Distill to Record* section are fields used to identify a target record. The target record is used to determine the record to which data will be copied.

There are radio buttons below the fields. The way that the target record is used to determine the record to which data will be copied depends on which one of the radio buttons is selected.

Here are descriptions of the fields:

Take the
This is a drop-down list that can have one of these possible values:

Business Object
If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.

Secondary BO

This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO*, then the record at the other end of the association is the target record.

This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO*, then the *Event* record that triggered the event is the target record.

Assignee
If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task
The value of this field is the label of the task that the target record will be associated with.

The radio buttons under these fields determine how the target record will be used to determine the record to which data will be copied.

When the properties form is first displayed, only the currently-selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are the descriptions:

Use it If this is selected, the target record will be the record to which data will be copied.

Use its Reference

If this is selected, a record referenced by a smart section or locator field of the target record will be the record to which data will be copied. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.

After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, a record associated with the target record will be the record to which data will be copied. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of this section in the *Object Type* field.

Use any Associated BO from Module ____ of type ____

This option is useful when you want to specify an associated record without specifying which association to use. This option is also useful when the association defined in the Data Modeler was to the base business object and you do not know which type of business object in a module is selected at runtime.

When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears in the drop-down list to the right of the module name, then the record on the other end of the association must have been created from the named business object. If *-Any-* appears in the drop-down list, then the record on the other end of the association may have been created from any business object in the named module.

To specify that a particular association name is required, click the Association icon. A list of the association types defined in the List Manager pops up. Click the association name that you want to appear to the right of the Association icon. To retrieve association records that are not restricted to a particular association name, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the target records' parent will be the record to which data will be copied.

When you select this radio button, a window pops up for you to select the business object that was used to create the parent record. This selection of a business object is not used for filtering. It is used to allow other tasks to access the parent's fields.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

At the bottom of the section is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that will be the record to which data will be copied.

From binary field section

The third section of the form for a Distill File task's properties is labeled *From Binary Field*. This section specifies a binary field in a record that contains the Excel spreadsheet from which data will be copied. The Excel spreadsheet in the specified field is not modified.

The *From Binary Field* section has two radio buttons to specify the record that contains the Excel spreadsheet. These radio buttons are labeled:

Workflow Activity

If this radio button is selected, it means that the record that contains the binary field that contains the Excel spreadsheet will be associated with a preceding workflow task.

Existing Record

If this radio button is selected, it means that the binary field that contains the Excel spreadsheet is contained in a record that exists now, at the time you are editing this workflow.

The selection of one of these radio buttons determines what appears in the *From Binary Field* section.

When the *Workflow Activity* radio button is selected, the *From Binary Field* section has fields to select a workflow task to which a record is associated that contains the Excel spreadsheet.

Under the *Workflow Activity* radio button there are fields used to identify a target record that is used to determine the record that contains the Excel spreadsheet.

There are radio buttons below these fields. The way that the target record is used to determine the record that contains the Excel spreadsheet depends on which one of the radio buttons is selected.

Here are descriptions of the fields used to determine the target record:

Take the

> This is a drop-down list that can have one of three possible values:
>
> **Business Object**
>
> > If *Business Object* is selected, then the record associated with the task specified by the field to the right of this one will be the target record.
>
> **Secondary BO**
>
> > This option is available if the workflow is launched in response to an Associate or De-Associate system event. If the value of this field is *Secondary BO,* then the record at the other end of the association is the target record.
> >
> > This option is also available if the workflow is launched in response to a *SCHEVENTCREATE, SCHEVENTSTART* or *SCHEVENTEND* system event. If the value of this field is *Secondary BO,* then the *Event* record that triggered the system event is the target record.
>
> **Assignee**
>
> > If *Assignee* is selected, then the *My Profile* record of the user assigned to the task specified by the field to the right of this one will be the target record.

of Task

> The value of this field is the label of the task that the target record will be associated with.

The radio buttons under the these fields determine how the target record will be used to determine the record that contains the Excel spreadsheet.

When the properties form is first displayed, only the currently-selected radio button is visible. There is a down-arrow button to the left of the visible radio button. Clicking the down-arrow button alternately makes all the radio buttons visible or just the selected radio button visible.

The following descriptions explain the meaning of the radio buttons. There are fields that appear to the right of some of the radio buttons. These fields contain additional information needed for the choice represented by the radio button.

Here are descriptions of the radio buttons:

Use it If this is selected, the target record will be the record that contains the Excel spreadsheet.

Use its Reference

> If this is selected, the record that contains the Excel spreadsheet will be referenced by a smart section or locator field of the target record. When you select this radio button, a window pops up that allows you to choose from the smart sections and locator fields in the target record.
>
> After you have selected this radio button, the name of the selected smart section or locator field is displayed in a read-only field to the right of the radio button.

Use its Association

If this is selected, the record that contains the Excel spreadsheet will be associated with the target record. When you select this radio button, a window pops up that allows you to specify the type of association to use. It allows you to identify the association by the type of record that must be on the other end of the association and optionally the name of the association.

After you have selected this radio button, if you specified an association name, the association name is displayed in a read-only field to the right of the radio button. The type of record that was selected appears at the bottom of the *Initialize from* section in the *Object Type* field.

Use any Associated BO from Module _____ of type _____

This option is useful when you want to specify an associated record without specifying which association to use. When you select this radio button, you must specify a module, unless the module whose name appears to the right of the Module icon is the module that contains the business object used to create the record on the other end of the association. If that is not the correct module, then click the Module icon. A list of modules will pop up. Click the correct module.

You may also specify that the record on the other end of the association must have been created by a particular business object in the specified module. If the name of a business object appears to the right of the Business Object icon, then the record on the other end of the association must have been created from the named business object. If nothing appears to the right of the Business Object icon, then the record on the other end of the association may have been created from any business object in the named module.

To specify what appears to the right of the Business Object icon, click the Business Object icon. A list of the business objects in the named module pops up. Click the business object that you want to appear to the right of the Business Object icon. If you want nothing to appear after the Business Object icon, click *-Any-* which appears at the top of the list.

Use its Parent

If the target record is created from a business object that is part of a hierarchy module and this option is selected, then the record that contains the Excel spreadsheet will be the target record's parent.

When you select this radio button, a window pops up for you to select the business object that is assumed to have been used to create the parent record. This selection of a business object is not used for filtering. It is used to allow this task to access the parent's fields.

The selection of a business object represents an assumption about what kind of record the parent will be. Because of this assumption, subsequent tasks will be able to access the parent record's fields. If the actual parent was not created from the assumed business object, the task may fail if the actual record does not have an expected field.

One of the choices for the business object that created the parent is *-Any-*. Choosing this will be the equivalent of selecting the module's base business object (the one with the same name as the module).

Below the radio buttons is a read-only field labeled *Object Type*. The value displayed in this field is the type of the record that values may be copied from. If

the record can have been created from any business object in a particular module, then the name of the module appears in the *Object Type* field.

At the bottom of the *From Binary Field* is a record labeled *Field*. This field has a drop-down list that contains the names of the binary fields in the type of record shown in the *Object Type* field. The binary field selected as the value of the *Field* field is the field that will be expected to contain the Excel spreadsheet.

If you want to initialize records with information that is determined by the application's configuration, then you should select the option of using an existing record in the *From Binary Field* section.

You specify the value for the *Module* and *Object* properties so that this task knows what kind of record you want to copy values from. Then click the *Record* link to find the specific record that contains the template Excel spreadsheet.

Once you have specified the record that contains the Excel spreadsheet, you then specify the binary field in the record that contains the spreadsheet by selecting its name as the value of the *Field* field.

Receiving offline Microsoft Excel spreadsheets by email

About this task

Typically when using the IBM TRIRIGA Connector for Offline Forms processing functionality, you will want to use email to send populated Microsoft Excel files. The system can then use the populated Excel file to update records.

The email capabilities used for offline processing are not the same as those used for notifications and do not support the same features as available for notifications.

Procedure

1. To configure the IBM TRIRIGA Application Platform to receive incoming email, you must first set up the SMTP server for your installation. Details on how to set up the SMTP server can be found in the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*. Next, you must set up at least one incoming email address for the system to listen for.
2. To create or modify incoming email settings, navigate to Tools > System Setup > System > Incoming Mail Config.
3. You can use the *Add* or *Edit* actions to create or edit Incoming Mail Config records.
4. If you add an Incoming Mail Config record, you will see the Incoming Mail Config form. If you wish to have the IBM TRIRIGA Application Platform listen for email on multiple user account / folder combinations, set up an Incoming Mail Config record for each combination. The following explains the sections and fields in an *Incoming Mail Config* record:

   UID   This field stores the control number for this record assigned by the system.

   Host   This field stores the name of the host server that will receive the incoming email.

   Username
   This field stores the user name that will receive the email on the host server.

Password

> This field stores the password for the user that will receive the email on the host server.

Action

> This field stores the name of the action that will be called on the EmailMessage when a message is received for this user. Often this field's value is CREATE.

MailServerType

> Select the mail server type from the drop-down list. pop3 is the default.

5. When an email is received for this account, a new EmailMessage record is created. Additionally, new records are created for each attachment to the message and for each email address associated with the message. The EmailAddress and EmailAttachment records are associated to the EmailMessage record.

6. After these records are created and the action specified is triggered, the EmailMessage record is moved from the null state. Attach a workflow to the Action triggered to process this email and distill any attachments.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp. Sample Programs.
© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and IBM's Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy/.

IBM ®

Printed in USA