

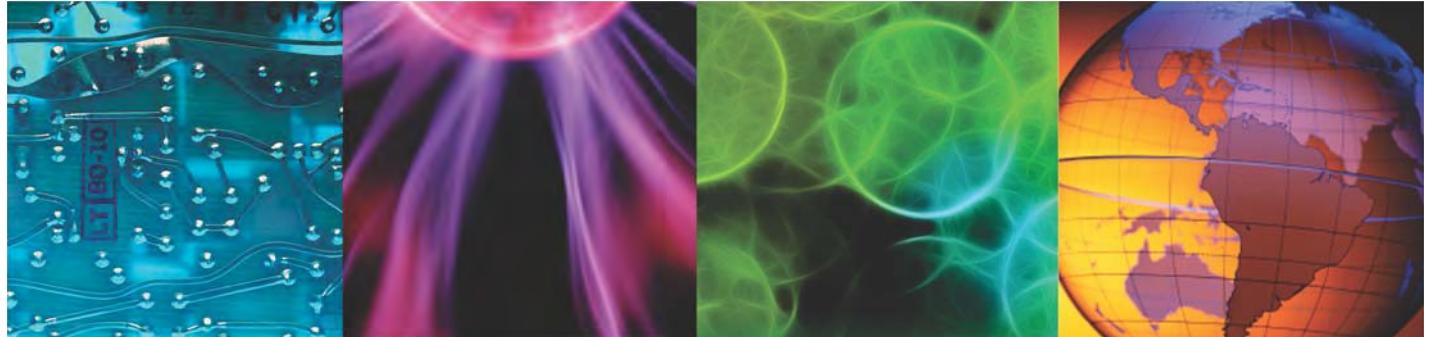


# IBM Training

**IBM TRIRIGA 10.3  
Application Platform I SPVC  
Student Notebook**

Course code TOD18 ERC 1.0

November 2014



**Cloud & Smarter Infrastructure**

[ibm.com/training](http://ibm.com/training)

Authorized  
**IBM | Training**

All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I SPVC) are IBM copyright property covered by the following copyright notice.

© Copyright IBM Corp. 2014. All Rights Reserved.

US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, and ibm.com are 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](http://www.ibm.com/legal/copytrade.shtml).

Trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered 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.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this publication to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results.



# Contents

<b>1 Architectural overview . . . . .</b>	<b>1</b>
Objectives . . . . .	2
Lesson 1. Architectural overview . . . . .	3
Architecture: Applications . . . . .	4
Architecture: Platform . . . . .	5
Applications and platform . . . . .	7
Custom applications in IBM TRIRIGA . . . . .	9
Data model . . . . .	10
User interface . . . . .	12
Reports and queries . . . . .	13
Custom logic . . . . .	14
Records . . . . .	15
Security . . . . .	16
Integration . . . . .	18
Instructor demonstration . . . . .	19
Student exercises . . . . .	20
Summary . . . . .	21
<b>2 Business scenario . . . . .</b>	<b>23</b>
Objectives . . . . .	24
Lesson 1. Creating an online catalog . . . . .	25
Assignment overview . . . . .	26
Assignment details . . . . .	27
Association diagram . . . . .	28
Instructor demonstration . . . . .	29
Student exercises . . . . .	30
Summary . . . . .	31
<b>3 Data models and the Data Modeler . . . . .</b>	<b>33</b>
Objectives . . . . .	34
Lesson 1. Using data models . . . . .	35
Definition of a data model . . . . .	36
Data Modeler tool . . . . .	38
Data Modeler interface . . . . .	39
Actions in the Data Modeler . . . . .	41
Instructor demonstration . . . . .	42
Modules . . . . .	43
Business objects . . . . .	44
Base business object . . . . .	46

Fields .....	47
Adding fields to a business object .....	48
Field properties .....	50
Defining the key to the records .....	52
Publishing a business object .....	54
Student exercises .....	55
Checkpoint questions .....	56
Checkpoint answers .....	57
Summary .....	58
<b>4 State families.....</b>	<b>59</b>
Objectives .....	60
Lesson 1. Using state families .....	61
Life cycle of a record .....	62
State transition family .....	63
Tools for creating and modifying a state family .....	65
State Transition editor interface .....	66
Creating a state family in a business object .....	68
Saving the State family .....	70
Adding a state .....	71
Adding a transition .....	72
Status .....	74
Sub actions .....	75
Adding a sub action .....	76
Capturing status in a sub action .....	77
Student exercises .....	78
Checkpoint questions .....	79
Checkpoint answers .....	80
Summary .....	81
<b>5 Forms and the Form Builder .....</b>	<b>83</b>
Objectives .....	84
Lesson 1. Using forms and the Form Builder .....	85
Forms .....	86
Example of a record in a form .....	88
Form Builder .....	89
Form wizard interface .....	91
Creating a form .....	93
Adding a tab to a form .....	95
Adding a section to a tab .....	97
Adding fields to a Form section .....	99
Data field properties .....	101
Positioning fields in a section .....	103
Other actions .....	105
Multi-tab sections .....	107
Includes/Forms tab .....	108
State Transition tab .....	109
Student exercises .....	111
Checkpoint questions .....	112

Checkpoint answers .....	113
Summary .....	114
<b>6 Navigation .....</b>	<b>115</b>
Objectives .....	116
Lesson 1. Navigating the user interface .....	117
Navigation items and navigation collections .....	118
Navigation Builder tool .....	119
Navigation collections .....	121
Navigation Items Library .....	123
Creating a navigation item .....	124
Sample Master/Detail Query navigation item .....	125
Adding a quick link to a portal .....	126
Student exercises .....	127
Checkpoint questions .....	128
Checkpoint answers .....	129
Summary .....	130
<b>7 Portals .....</b>	<b>131</b>
Objectives .....	132
Lesson 1. Portals .....	133
Portals and portal sections .....	134
Portal Builder tool .....	135
Portal Properties panel .....	136
Portal Layout panel .....	138
Portal Sections Library .....	139
Creating a portal section .....	140
Adding a quick link to a portal .....	142
Student exercises .....	143
Checkpoint questions .....	144
Checkpoint answers .....	145
Summary .....	146
<b>8 Basic reporting .....</b>	<b>147</b>
Objectives .....	148
Lesson 1. Creating reports .....	149
Reports in IBM TRIRIGA .....	150
Report Manager .....	151
Report Builder, General tab .....	153
Report Builder, Columns tab .....	155
Report Builder, Order & Group tab .....	157
Report Builder, Filters tab .....	158
Student exercises .....	160
Checkpoint questions .....	161
Checkpoint answers .....	162
Summary .....	163
<b>9 Lists and classifications .....</b>	<b>165</b>
Objectives .....	166

<b>Lesson 1. Lists . . . . .</b>	<b>167</b>
Introduction to lists . . . . .	168
List Manager . . . . .	169
Creating a manual list . . . . .	171
Editing a manual list . . . . .	173
Creating a dependent list . . . . .	174
Dependent list example . . . . .	175
Dynamic lists . . . . .	176
Creating a dynamic list . . . . .	177
Editing a dynamic list . . . . .	178
Attaching a list to a field . . . . .	179
Student exercises . . . . .	180
<b>Lesson 2. Classifications . . . . .</b>	<b>181</b>
Definition of classifications . . . . .	182
Comparison of lists and classifications . . . . .	183
Classification hierarchy . . . . .	184
Classification fields and associations . . . . .	186
Instructor demonstration . . . . .	187
Checkpoint questions . . . . .	188
Checkpoint answers . . . . .	189
Summary . . . . .	190
<b>10 Data Integrator . . . . .</b>	<b>191</b>
Objectives . . . . .	192
<b>Lesson 1. Using the Data Integrator . . . . .</b>	<b>193</b>
Defining the Data Integrator . . . . .	195
Requirements for Data Integrator files . . . . .	196
Creating a header file . . . . .	198
Creating a header file (continued) . . . . .	199
Uploading a data file . . . . .	201
Adding to a hierarchy . . . . .	203
Creating an association . . . . .	205
Student exercises . . . . .	207
Checkpoint questions . . . . .	208
Checkpoint answers . . . . .	209
Checkpoint answers, continued . . . . .	210
Summary . . . . .	211
<b>11 Object migration: Import . . . . .</b>	<b>213</b>
Objectives . . . . .	214
<b>Lesson 1. Importing an object migration package . . . . .</b>	<b>215</b>
Object migration . . . . .	217
Object migration, continued . . . . .	218
Importing . . . . .	220
Upload the Object Migration (OM) package . . . . .	222
Object Migration window for an import . . . . .	224
Summary report . . . . .	226
Comparison report . . . . .	227
Import . . . . .	229

Instructor demonstration .....	231
Student exercises .....	232
Checkpoint questions .....	233
Checkpoint answers .....	234
Summary .....	235
<b>12 Associations .....</b>	<b>237</b>
Objectives .....	238
Lesson 1. Creating and using associations .....	239
Associations .....	241
Examples of associations .....	243
Types of associations in IBM TRIRIGA .....	244
Association Types list .....	246
Viewing associations in the Association Manager .....	247
Creating associations in the Association Manager .....	249
Viewing associations in the Data Modeler .....	252
Creating associations in the Data Modeler .....	253
Record-level associations .....	255
Associations tab .....	256
Student exercises .....	257
Checkpoint questions .....	258
Checkpoint answers .....	259
Checkpoint answers (continued) .....	260
Summary .....	261
<b>13 Smart sections and locator fields .....</b>	<b>263</b>
Objectives .....	264
Lesson 1. Using smart sections and locator fields .....	265
Locating records .....	267
Smart sections .....	268
Creating a smart section .....	270
Smart section properties .....	272
Selecting smart section fields .....	274
Results of saving a smart section .....	275
Viewing the smart section field list .....	276
Using the chevron .....	277
Opening the section in the business object field list .....	278
Using the Association List panel .....	279
Single-record smart section on a form .....	280
Locator fields .....	281
Creating a locator field .....	282
Locator field mapping .....	284
Locator field on a form .....	285
Instructor demonstration .....	286
Student exercises .....	287
Checkpoint questions .....	288
Checkpoint answers .....	289
Checkpoint answers, continued .....	290
Summary .....	291

<b>14 Formulas . . . . .</b>	<b>293</b>
Objectives . . . . .	294
Lesson 1. Using formulas . . . . .	295
Two types of formulas . . . . .	297
Regular formulas . . . . .	298
Defining a regular formula in Data Modeler . . . . .	299
Extended formula . . . . .	301
Defining an extended formula in Data Modeler . . . . .	303
Defining extended formula in workflow object map . . . . .	305
Using an extended formula as a workflow condition . . . . .	306
Student exercises . . . . .	308
Checkpoint questions . . . . .	309
Checkpoint answers . . . . .	310
Checkpoint answers (continued) . . . . .	311
Summary . . . . .	312
<b>15 Workflows . . . . .</b>	<b>313</b>
Objectives . . . . .	314
Lesson 1. Building workflows . . . . .	315
Definition of a workflow . . . . .	316
Starting a workflow . . . . .	318
Workflow Builder tool . . . . .	320
Right side of the Workflow Builder . . . . .	321
Workflow Editor . . . . .	323
New Task palette . . . . .	325
Adding a task from the New Task palette . . . . .	326
Moving and removing a task . . . . .	327
Instructor demonstration . . . . .	328
Two types of workflows . . . . .	329
Two types of data . . . . .	331
Start task of a synchronous workflow . . . . .	332
Start task of an asynchronous workflow . . . . .	335
Basic workflow tasks . . . . .	337
Modify records task . . . . .	339
Specifying the record to work with . . . . .	340
Other choices for the record . . . . .	342
Object Mapping window . . . . .	343
Modify metadata task properties . . . . .	345
Form Mapping window . . . . .	347
Switch task properties . . . . .	349
Get Temp Record task properties . . . . .	351
Save Permanent Record task properties . . . . .	352
State transitions and temporary data . . . . .	353
State transitions and temporary data flowchart . . . . .	354
Message handling in workflows . . . . .	356
Publishing a workflow . . . . .	358
Student exercises . . . . .	359
Lesson 2. Triggering workflows . . . . .	360

Triggering asynchronous workflows .....	361
Triggering synchronous workflows .....	363
Attaching a workflow to a sub action .....	365
Setting a pre-create workflow in a business object .....	366
Setting a pre-load workflow in a form .....	367
Attaching a workflow to a field .....	368
Workflow instances .....	369
Execution path .....	371
Revising a workflow .....	372
Workflow version control .....	374
Student exercises .....	375
Checkpoint questions .....	376
Checkpoint answers .....	377
Checkpoint answers, continued .....	378
Summary .....	379
<b>16 Advanced queries.....</b>	<b>381</b>
Objectives .....	382
Lesson 1. Using advanced queries .....	383
Using Find queries .....	385
Replacing a default Find query .....	386
Editable queries .....	387
Creating an editable query .....	388
Related reports .....	389
Defining related reports .....	390
Association filters .....	392
Defining an association filter .....	393
Association filter keywords for Record filters .....	395
Student exercises .....	397
Checkpoint questions .....	398
Checkpoint answers .....	399
Summary .....	400
<b>17 Object migration: Export .....</b>	<b>401</b>
Objectives .....	402
Lesson 1. Exporting objects .....	403
Object migration .....	404
Object migration (continued) .....	405
Exporting .....	407
Creating an export package .....	408
Object Migration export window .....	410
Selecting objects to be exported .....	412
Adding objects to export package .....	414
Adding dependent objects to export package .....	415
Exporting the objects to a file .....	416
Instructor demonstration .....	418
Student exercises .....	419
Checkpoint questions .....	420

## Contents

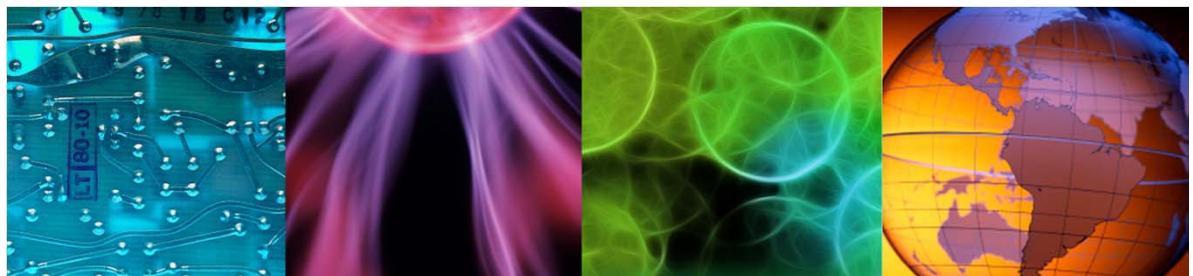
Checkpoint answers .....	421
Summary .....	422
<b>Appendix A Design .....</b>	<b>423</b>
Design elements .....	423
Field suffixes .....	427
Publish name .....	428
Associations .....	428
Workflows .....	429



# 1 Architectural overview



## 1 Architectural overview



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

IBM TRIRIGA is both a set of applications and a development platform. This unit is an overview of those components and an introduction to development of new applications.

### How you check your progress

You can check your progress by performing the lab exercises

### References

*Application Development for the IBM TRIRIGA Application Platform*



## **Objectives**

After completing this unit, you should have a basic understanding of the following topics:

- Application development in IBM® TRIRIGA®
- Reports and queries
- Records
- Security
- Integration

# Lesson 1. Architectural overview



## Lesson 1: Architectural overview



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is an overview of the TRIRIGA components and an introduction to development of new applications.

### What you should be able to do

After completing this lesson, you should have a basic understanding of the following topics:

- Application development in IBM TRIRIGA
- Reports and queries
- Records
- Security
- Integration

### References

*Application Development for the IBM TRIRIGA Application Platform*

## Architecture: Applications

- The IBM TRIRIGA Application includes the TRIRIGA version 10.x applications and the Integrated Workplace Management Solution (IWMS)
- The IWMS consists of the following items:
  - TRIRIGA 10 Real Estate
  - TRIRIGA 10 Facilities
  - TRIRIGA 10 Operations and Maintenance
  - TRIRIGA 10 Projects
  - TRIRIGA 10 TREES™
  - TRIRIGA Workplace Performance Management (WPM)

© Copyright IBM Corporation 2013

4

### *Architecture: Applications*

IBM TRIRIGA is both a set of applications and a development platform. The application part of IBM TRIRIGA includes the Release 10 applications and the Integrated Workplace Management Solution (IWMS).

The IWMS consists of the following items:

- TRIRIGA 10 Real Estate
- TRIRIGA 10 Facilities
- TRIRIGA 10 Operations and Maintenance
- TRIRIGA 10 Projects
- TRIRIGA 10 TREES™
- TRIRIGA Workplace Performance Management

## Architecture: Platform

- The platform is where the structure of applications is defined
- The IBM TRIRIGA platform includes the following items:
  - The TRIRIGA version 3.x Application Platform, a self-contained environment for running business applications
  - Tools for configuring and extending the applications
  - Security
  - Integration

© Copyright IBM Corporation 2013

5

### *Architecture: Platform*

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

The tool set within the IBM TRIRIGA Application Platform is the same tool set that was used for developing the IBM TRIRIGA 10 applications. Each tool in the tool set has a particular function. By using them together, you can build or configure an application to the specific requirements of your business.

The unique IBM TRIRIGA Application Platform has numerous features that are not in other development environments. These features include security measures to protect the data, and integration with other systems.

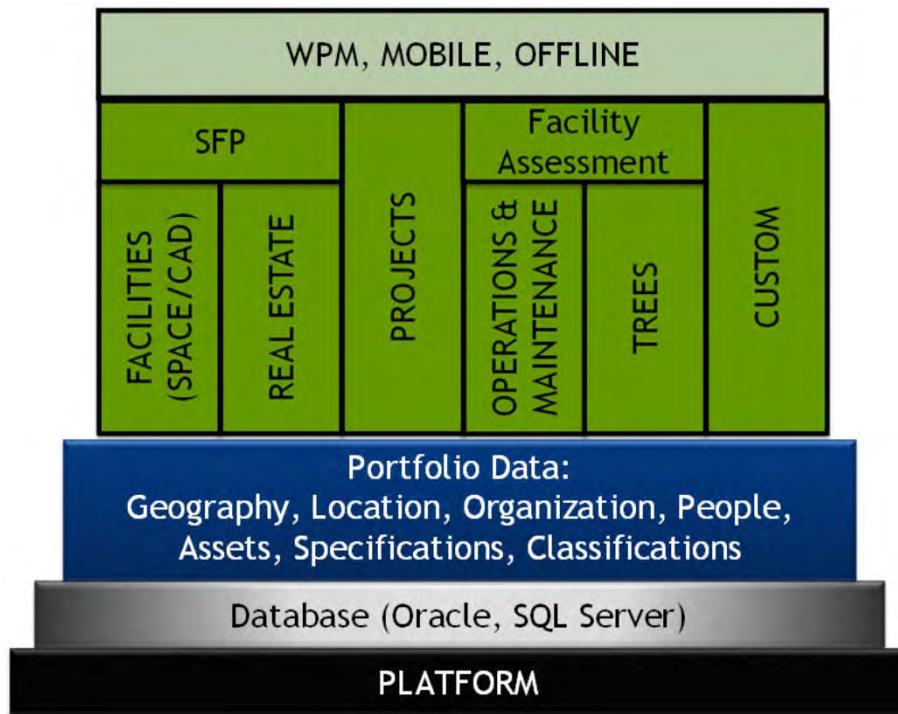
Two attributes of the IBM TRIRIGA Application Platform are significant:

- In many traditional technologies, a developer must have specific programming skills to build an application. The focus is on how an application works instead of what it does. The IBM TRIRIGA platform reduces the skills required to build an application. You can focus on what the application is supposed to do, and the result is typically a good fit for your business needs.

- The IBM TRIRIGA Application Platform includes predefined business objects to represent, for example, people, currencies, and schedules. These business objects can help save time and money that might be spent creating these business objects independently.

Building applications with these common predefined business objects has another, more important, benefit. Applications that use the same 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 must be entered only once.

## Applications and platform



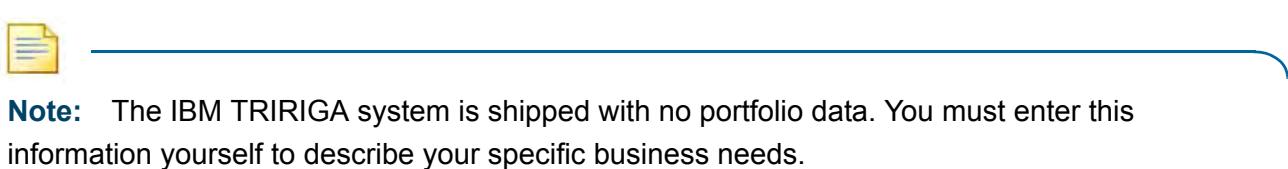
© Copyright IBM Corporation 2013

6

### Applications and platform

This slide shows the logical organization of the IBM TRIRIGA Application Platform and the suite of applications that are built on it. The following is an overview of each of the components:

- **Platform:** The Platform is where the structure of the applications is defined. The remainder of this course is devoted to development on the platform.
- **Database:** The database is where the data is stored. The database can be either Oracle or SQL Server, depending on your installation.
- **Portfolio Data:** The portfolio data is the core data of the IBM TRIRIGA system. It describes those things common to all applications. The categories of portfolio data are Geography, Location, Organization, People, Assets, Specifications, and Classifications.



- **Facilities:** Facilities is also known as Space Management, which means defining the spaces that you have and controlling their use by allocating space to people or organizations.
- **Real Estate:** This application has transactions to control the acquisition and release of properties or spaces. It also has contracts to manage current properties and includes a feature to send automated reminders.
- **SFP:** Strategic Facilities Planning (SFP) is for forecasting the need for future space by surveying management and compiling the results.
- **(Capital) Projects:** The Capital Projects application is for the construction of buildings, especially large ones like casinos.
- **Operations & Maintenance:** The focus of this application is to keep the equipment and machinery running smoothly. Service requests can be triggered on schedule, by events, or manually.
- **TREES:** TRIRIGA Environmental Sustainability (TREES) is an application that is used to calculate your company's carbon footprint. This footprint is based on many factors, including energy consumption, the source of the energy, and waste production. Knowing your carbon footprint is becoming increasingly important in a *green* world.
- **Facility Assessment:** The Facility Assessment application is for assessing the condition of the buildings. This information is especially important for new acquisitions, so that you know about and can prioritize necessary repairs.
- **Custom:** The Custom application is for everything that is custom-built or modified in the platform.
- **WPM, Mobile, Offline:**
  - WPM stands for WorkPlace Management. It is a tool to assist management with operations.
  - Mobile is used to provide access to IBM TRIRIGA on mobile devices.
  - Offline is a means to gather information from people while it keeps them out of your system. Requests for information are emailed to external entities, such as contractors and vendors. The responses are received and processed back into TRIRIGA.

## Custom applications in IBM TRIRIGA

To build a simple custom IBM TRIRIGA application, you must add the following four items to the platform:

- A description of how the data in the application is organized
- A description of what the user interface in the application looks like
- Descriptions of the reports and queries that the application supports
- Custom logic that is needed for the business processes that the application supports

© Copyright IBM Corporation 2013

7

### *Custom applications in IBM TRIRIGA*

The IBM TRIRIGA Application Platform is a self-contained environment for building and running business applications. Already 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 to add only the following four items:

- A description of how the application's data is organized
- A description of what the application's user interface looks like
- Descriptions of the reports and queries that the application supports
- Custom logic that is needed for the business processes that the application supports

## Data model

The description of application data is called a data model, which has these main parts:

- **Module:** A container for business objects
- **Business object:** A collection of fields that represents a real-world object
- **Field:** Container for a piece of information
- **State Family:** Defines the lifecycle of records that are created from a business object
- **Association:** A relationship between records

© Copyright IBM Corporation 2013

8

### *Data model*

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

The data model is organized into five main parts:

- **Field definitions:** Describe individual pieces of data.
- **Business objects:** Are a collection of attributes of a real-world items and the processes that act on that set of attributes. Business objects are used to create records. A record contains actual data for the fields that are described by the business object that is used to create the record.
- **Module:** Is a collection of one or more business objects. Each business object belongs to exactly one module.
- **State family:** Defines the lifecycle of records that are created from a business object. The lifecycle of most records is that they are created, modified, and possibly deleted. The lifecycle of a record is determined by the states and transitions in the state family of the business object that is used to create it.

- **Associations:** Describe how business objects are related to each other, which defines how records are related to each other.

When you use the IBM TRIRIGA Application Platform to build an application, begin by defining the data model. The other pieces of the application, the user interface, the reports, and the custom logic, all use the data model.

## User interface

The user interface has these essential pieces:

- **Portals:** Provide access to the major parts of the application
- **Navigation items:** Allow access to related types of records
- **Forms:** Facilitate creation, viewing, and editing of records

### *User interface*

The user interface of an application that runs on the IBM TRIRIGA Application Platform has three essential pieces:

- **Portal:** Use the portal to access the major parts of an application. A portal can include menus and more sophisticated graphical components.
- **Navigation items:** Use navigation items to display a collection of forms, the results of a query, and hierarchical data. Navigation items can also represent menus and portal quick link sections, run reports, and link to builder tools.
- **Forms:** Create, view, and edit records by using forms. Each form is associated with a business object. You can use different forms to access some of the same records to provide a different presentation of the contents of those records.

## Reports and queries

- You can use the TRIRIGA Application Platform to generate reports that reflect the contents of records and their associations
- In queries and reports that you generate, you can drill down into or click through to records
- External reporting tools also available; for example, BIRT or Crystal

© Copyright IBM Corporation 2013

10

### Reports and queries

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

One mechanism for generating internal reports is the **Report Manager**. You can use the Report Manager to generate tabular reports or graphs.

Three features of reports that are generated by the Report Manager are as follows:

- 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 usually called **queries**.
- You can drill down into underlying records of the reports. After you select data in a report, a form is displayed for you to view or edit the record that the report data came from.
- You can edit the values directly in the report, if the report is configured for it. With this feature, you can edit the values in several records at once.

You can use the IBM TRIRIGA Advanced Reporting tool to generate reports externally. You have more flexibility in formatting external reports than you do in reports that are generated internally.

## Custom logic

Custom logic for an application is provided by workflows and formulas

- **Workflows:** Specified system tasks
- **Formulas:** Used for calculating values and controlling conditional logic

### *Custom logic*

You can add business logic to your applications in the IBM TRIRIGA Application Platform by creating a workflow. You can create workflows to define any business process that is associated with the system or the business objects in the system. Several predefined workflows are delivered with the IBM TRIRIGA applications. You can also use formulas to add custom logic to an application, by calculating field values and by controlling the path through a workflow.

## Records

A record is a unique instance of real-world data

- Has fields as specified by the business object
- Has appearance as specified by the form
- Is typically identified by the BO or form
  - A triPeople record or an Employee record
- Can be created, grouped, modified, queried, and deleted
- Is hierarchical or nonhierarchical

© Copyright IBM Corporation 2013

12

### Records

Records are instances of a collection of data that represent information from business objects. A record contains one or more data elements, or fields. You create records from forms and associate records with other records. Most records have fields that you can use to associate them with a particular organization, location, and geography.

You identify types of records by using the names of business objects. For example, you create triPeople records from the triPeople business object. In some cases, where multiple forms are associated with a business object, the name of the form is used to identify a record. For example, an Employee record is a record that you create from an Employee form.

You can organize records into hierarchies, where each record can have child records. These records are commonly used to represent real-world data that is hierarchical, such as locations, geographies, and organizations.

## Security

- Record-level access restrictions are based on a matrix of organization and geography, and by project
- Field and tool access are restricted by membership in groups  
You have no access unless a group grants it to you
- Platform and application access are restricted by licenses

© Copyright IBM Corporation 2013

13

### Security

In the IBM TRIRIGA Application Platform, security controls the access to data and to applications. The main facets of security in the IBM TRIRIGA Application Platform are as follows:

- **Licenses:** With a license, you can access specific features of an application or 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 people and other groups. A person can belong to more than one group. A person's access is the sum of the accesses that are granted by all of the groups that the person belongs to.
- **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 association with an organization or geographic area to control which users can access the record. Record-level access is

hierarchical. Users can see all records that were created at or below the same level of their geography and organization, but cannot see any other records.

- **Project:** The project that is associated with a record can restrict who can access the record.

The basic rule of platform security is that a person has no access to something unless they belong to a group that gives them access.

## Integration

You can integrate data in the following ways:

- Import data by using the DataConnect™ integration utility
- Import data from flat files by using Data Integrator
- Connect to other systems by using TRIRIGA BusinessConnect™, a SOAP application programming interface

### *Integration*

There are various mechanisms to enable applications that run on the IBM TRIRIGA Application Platform to work with applications that run outside of the platform. Some of the mechanisms for bringing data into the IBM TRIRIGA Application Platform are as follows:

- **Data Integrator:** The Data Integrator reads data from tab-delimited text files and uses it to create or update IBM TRIRIGA Application Platform records.
- **DataConnect:** DataConnect ensures that an external source can write data directly into a staging area in IBM TRIRIGA. The data can then be processed for insertion into IBM TRIRIGA business objects.
- **IBM TRIRIGA Connector for Business Applications:** Programmers can write programs that work directly with the IBM TRIRIGA Application Platform. This mechanism is the only one of the listed mechanisms that can be used to transfer data in both directions.

Using this mechanism solves most integration problems that cannot be solved with other integration mechanisms. To use it, a programmer must be knowledgeable about SOAP technology. This technology is part of the IBM TRIRIGA Application Platform.

## Instructor demonstration



Take a tour of IBM TRIRIGA

© Copyright IBM Corporation 2013

15

*Instructor demonstration*

---

## Student exercises

---



© Copyright IBM Corporation 2013

16

### *Student exercises*

Perform the exercises for this unit.

## Summary

---

After completing this chapter, you should have a basic understanding of the following topics:

- Application development in IBM TRIRIGA
- Reports and queries
- Records
- Security
- Integration

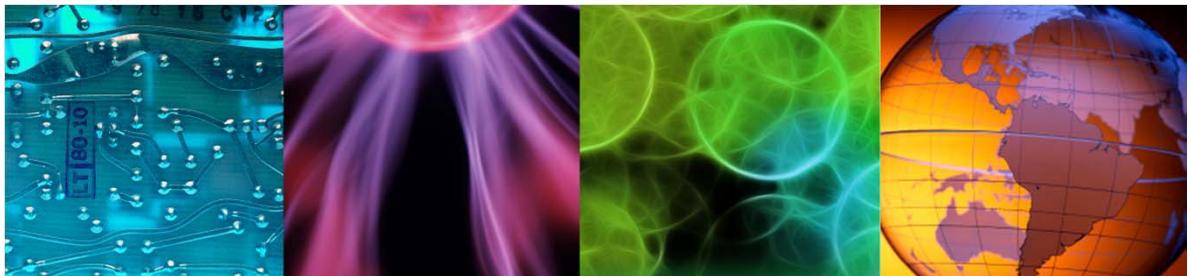




## 2 Business scenario



## 2 Business scenario



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

The application that you create during the class is described and demonstrated in this unit.

### How you check your progress

Not applicable for this unit



## **Objectives**

After completing this unit, you should be able to create an online course catalog according to the business scenario in this unit

# Lesson 1. Creating an online catalog



## Lesson 1: Creating an online catalog



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson provides a description of the assignment.

### What you should be able to do

After completing this lesson, you should be able to create an online course catalog according to the business scenario in this unit.



## Assignment overview

---

- You work for the local university, the University of U
- The university uses IBM TRIRIGA and wants to create an online course catalog to track their course offerings and enrollments
- You are assigned to the project, and you must undergo the training necessary to successfully implement this solution

---

© Copyright IBM Corporation 2013

4

### Assignment overview

You are employed by the local university, the University of U. The university uses IBM TRIRIGA and wants to create an online course catalog to track their course offerings and enrollments. You are now assigned to the project and you must undergo the training necessary to successfully implement this solution.

## Assignment details

The application must be able to create and manage the following types of records:

- **Courses:** What courses the university offers
- **Students:** Who attends classes
- **Instructors:** Who can teach the classes
- **Classrooms:** Where classes can be taught
- **Enrollments:** Which class a particular student takes
- **Course Sections:** Where all other information is gathered in one place

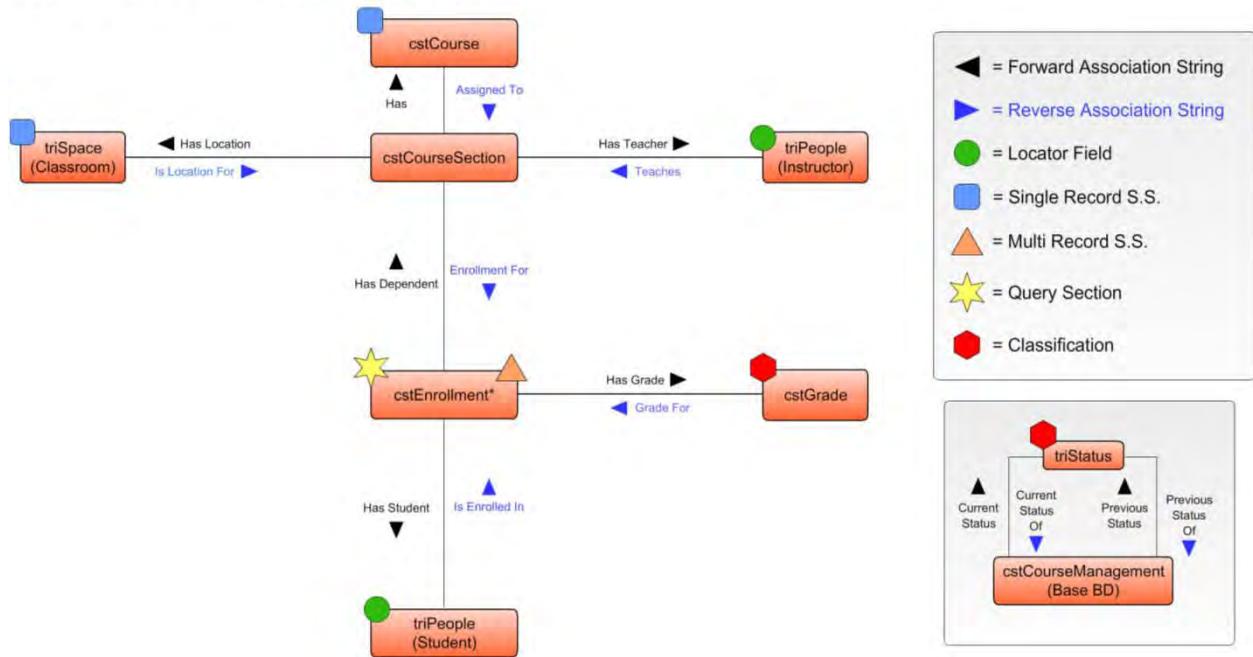
### Assignment details

The application must be able to create and manage the following types of records:

- **Courses:** What courses does the University offer? A business object of this type does not exist in the platform and must be created.
- **Students:** Who can attend classes? The existing business object triPeople can be used for this information. Students are considered *external contacts*.
- **Instructors:** Who can teach the classes? The existing business object triPeople can be used for this information. Students are considered *employees*.
- **Classrooms:** Where can classes be taught? The existing business object triSpace can be used for this information.
- **Enrollments:** Which class is a particular student enrolled in? This business object is an *intermediate object* that connects a student record to a course section record. A business object of this type does not exist in the platform and must be created.
- **Course Sections:** This business object is where all of the other information is gathered in one place. A business object of this type does not exist in the platform and must be created.

## Association diagram

Course Management Association Diagram



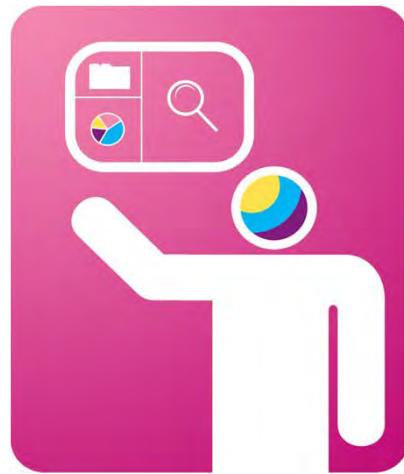
© Copyright IBM Corporation 2013

6

### Association diagram

The diagram in this slide shows the relationships between the business objects of the application. It also names the strings to be used for the forward and reverse associations. For example, from **cstEnrollment** to **cstGrade** is a *Has Grade* association. In the reverse direction, it is a *Grade For* association.

## Instructor demonstration



Demonstrate the application to be built

© Copyright IBM Corporation 2013

7

*Instructor demonstration*

---

## Student exercises

---



© Copyright IBM Corporation 2013

8

*Student exercises*

## Summary

---

Now that you have completed this unit, you should be able to create an online course catalog according to the business scenario in this unit





## 3 Data models and the Data Modeler



## 3 Data models and the Data Modeler



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

IBM TRIRIGA applications are built around the data model. In this unit, you learn how to use the Data Modeler tool to define the data model. You also learn how to publish business objects.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building for the IBM TRIRIGA Application Platform*



## **Objectives**

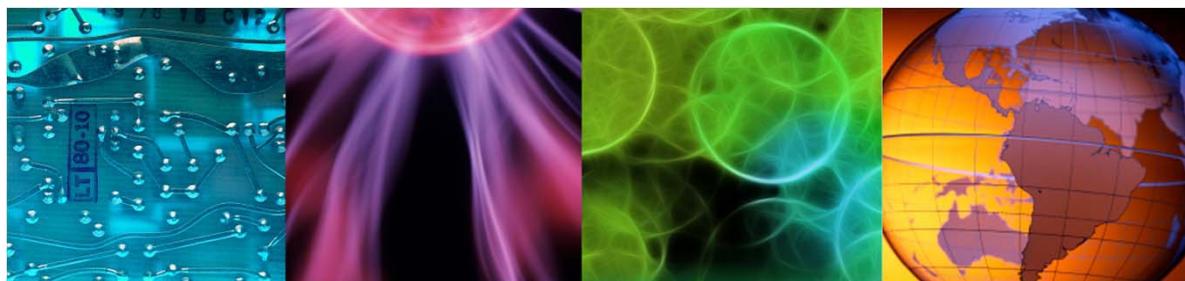
After completing this unit, you should be able to perform the following tasks:

- Describe the components of a data model in IBM® TRIRIGA®
- Use the Data Modeler to create and edit a data model
- Publish and revise business objects

# Lesson 1. Using data models



## Lesson 1: Using data models



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013. US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

In this lesson, you learn how to use the Data Modeler tool and publish business objects.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the components of a data model in IBM TRIRIGA
- Use the Data Modeler to create and edit a data model
- Publish and revise business objects

### References

*Application Building for the IBM TRIRIGA Application Platform*



## Definition of a data model

- A data model is a description of the data that an application uses
- It is organized into the following five main parts:
  - **Field:** Container for a piece of information
  - **Business object:** A collection of fields and business logic
  - **Module:** A container for business objects
  - **State Family:** Lifecycle of records that are created from a BO
  - **Association:** A relationship between records

© Copyright IBM Corporation 2013

4

### *Definition of a data model*

The IBM TRIRIGA Application Platform maintains a description of the data that applications use. This description is called the **data model**. The data model is organized into five main parts:

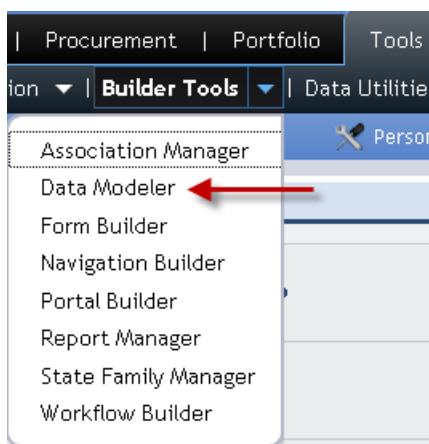
- **Field definitions:** Describe individual pieces of data. A field contains an individual piece of data.
- **Business objects:** Are a collection of attributes of a real-world items and the processes that act on that set of attributes. You use business objects to create records. A record contains actual data for the fields that are described by the business object that is used to create the record.
- **Module:** Is a collection of one or more business objects. Each business object belongs to exactly one module.
- **State transitions:** Control the lifecycle of records that are created from a business object. The lifecycle of most records is that it is created, modified, and possibly deleted. The lifecycle of a record is determined by the states and transitions in the business object that is used to create it.
- **Associations:** Describe how business objects are related to each other, which defines how records are related to each other.

Modules, business objects, and fields are **metadata**, which is data that describes data. For example, a record is data. The business object defines the record. The business object is metadata. This term is used throughout this course.

## Data Modeler tool

When you use the IBM TRIRIGA Application Platform to build an application, first define the data model

- Use the Data Modeler tool
- Click **Tools > Builder Tools > Data Modeler**



© Copyright IBM Corporation 2013

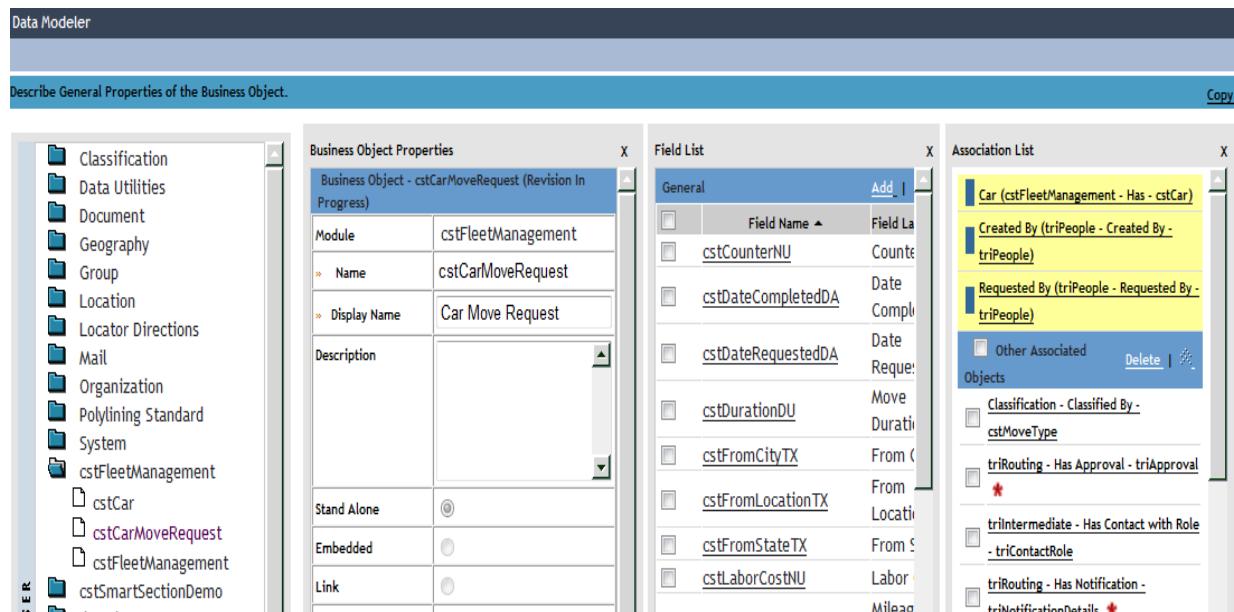
5

### Data Modeler tool

When you use the IBM TRIRIGA Application Platform to build an application, you begin by defining the data model. The other pieces of the application (the user interface, the reports, and the custom logic) all use the data model.

The data model is defined in the Data Modeler tool. You can access the Data Modeler by clicking **Tools > Builder Tools > Data Modeler**, or from the Application Builder home page, in the Builder Tools portal section.

# Data Modeler interface



© Copyright IBM Corporation 2013

6

## Data Modeler interface

The Data Modeler tool is where you define the components of the data model. Within the tool, they are organized by module and then by business object.

The following features of the Data Modeler interface are the primary ones:

- **Object Browser**
  - The Object Browser is the vertical bar on the left side of the Data Modeler window. You can expand the Object Browser to show modules and business objects.
  - Modules are shown as folders. Click the *name* of the module to see its properties. Click the *folder icon* to open it and see the business objects within the module.
  - In the example, the cstFleetManagement module is open, which is indicated by the open folder to the left of its name. The cstCarMoveRequest Business Object is selected.
- **Properties panel**
  - In this panel, you can see the properties of the object that you selected.
  - In the example, this panel is immediately to the right of the Object Browser. It shows the properties of the cstCarMoveRequest business object.

- **Field List panel**

- This panel has fields that are contained in the business object and the section that these fields are in.
- At the top of the General section, you can use the **Add**, **Find**, and **Delete** actions. Other section headings do not include these actions.
- In the example, this panel is to the right of the Properties panel.

- **Association List panel**

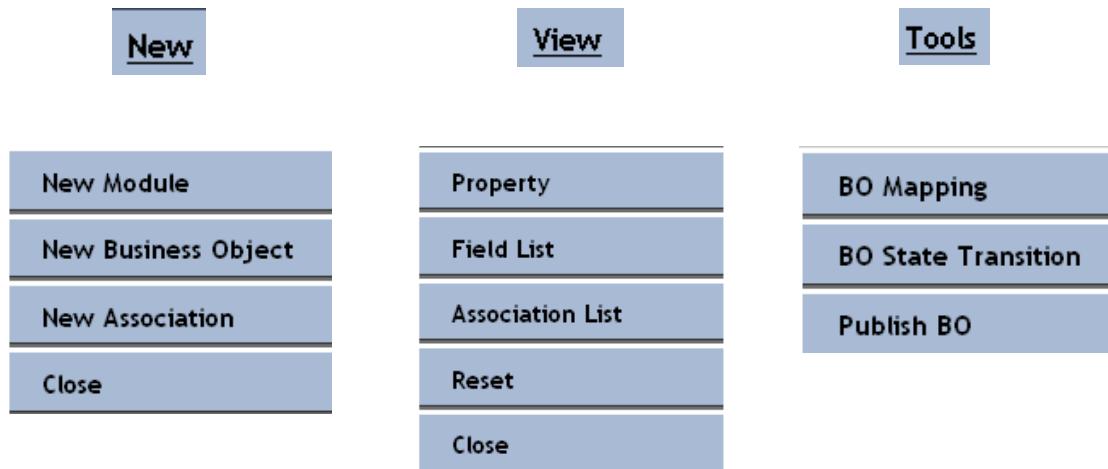
- This panel has a list of BO-level associations that originate from the current business object.
- In the example, this panel is on the far right.
- Associations that have a smart section are displayed with a yellow background. **Include** associations, which are used in hierarchies, are displayed with a light blue background.

## Manipulating panels in the Data Modeler

- To move a panel, click its header and drag it.
- To resize a panel, choose one of the following options:
  - Click the border of the panel and drag it.
  - Press and hold the Shift key, click the header of the panel, and drag it.

If you drag a panel up too far, click **View > Reset** to move the panel back to its default position.

## Commands in the Data Modeler



© Copyright IBM Corporation 2013

7

### *Actions in the Data Modeler*

At the upper right of the Data Modeler are the **New**, **View**, and **Tools** actions. Clicking one of these actions opens the following menu and choices:

- **New:** Displays a list of actions for creating objects: **New Module**, **New Business Object**, and **New Association**. Click the one you want to create.
- **View:** Displays a list of choices for opening panels that might not be visible, such as **Property**, **Field List**, **Association List**.

There is also a choice named **Reset**, which you can use to bring the panels back to their default position. You must use this action if a panel is too high in the display and you cannot reach the title bar to move it.

- **Tools:** Displays a list of tools that can be accessed: **BO Mapping**, **BO State Transition**, **Publish BO**. If the business object is published, the tool is **Revise BO**. There are other choices, but these tools are the most commonly used inside the Data Modeler.
- **Close:** Indicates that you are canceling a choice. The menus do not automatically close when you move the pointer outside of the list. You must choose an item from the list for the menu to close it.

## Instructor demonstration



## Data Modeler interface

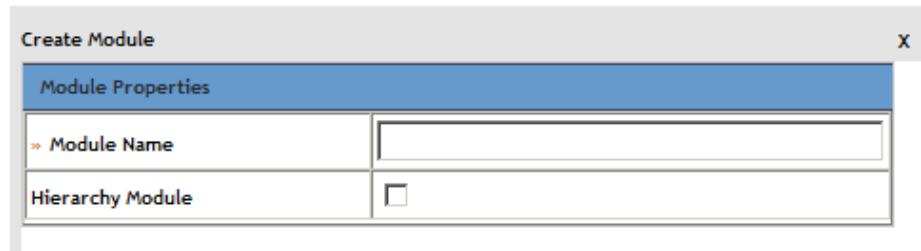
© Copyright IBM Corporation 2013

8

*Instructor demonstration*

## Modules

- A grouping of business objects with a similar purpose
- Used for organizing forms, workflows, reports, and queries
- Every module must have a unique name
- After you create a module, you cannot delete it
- To create a module, click **New > New Module**



© Copyright IBM Corporation 2013

9

### Modules

A module is a logical grouping of business objects that have a similar structure or purpose. For example, the Asset module contains business objects such as Building Equipment, Furniture, and Technology Assets.

Place multiple business objects in the same module by following these guidelines:

- Business objects that have common data fields
- Business objects that can have the same role in the same business process
- Business objects that have a similar business purpose

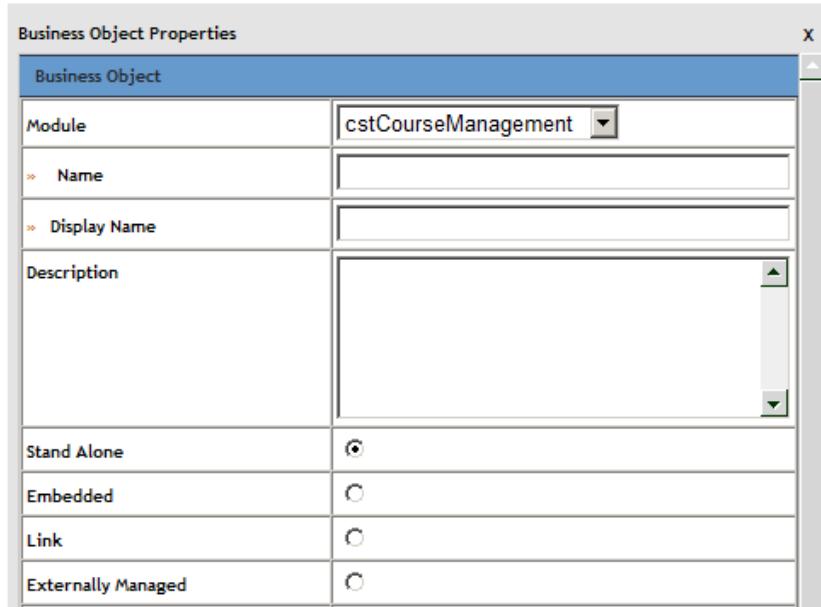
You create a module by clicking New > New Module. Every module must have a unique name.

A module has only two properties: Module Name and Hierarchy Module. Select Hierarchy Module only if the business objects in the module are for hierarchical data. Examples of hierarchical data are location, geography, and organization.

After a module contains a business object, you cannot delete it. Ensure that you create only the modules that are required.

## Business objects

- A collection of fields and business logic
- Create by clicking **New > New Business Object**



© Copyright IBM Corporation 2013

10

### Business objects

A business object is a collection of fields and business logic that represents something that exists in the real world. Business objects are used to create records that are instances of data.

Business objects in the same module cannot have the same name. It is good practice to give new business objects a unique name and label throughout all modules.

Create a business object by clicking **New > New Business Object**. This command opens a blank business object in the Properties panel. The module for a new business object defaults to the selected module.

The name that you use for the business object depends on whether it is the first business object in the module, with the following guidelines:

- If it is the first business object, it is a base business object and by convention it has the same name as the module.
- If it is not the first, then you can name it whatever you want, with two conditions:
  - The name has a prefix of cst.
  - The name cannot contain spaces or special characters.



**Note:** The label of a business object can contain spaces and special characters.

There are four types of business objects:

- Stand Alone

Records that you create from stand-alone business objects can exist independently of other records. You can use records from stand-alone business objects at the top-level organization of reports and queries. Most business objects are Stand Alone business objects.



**Important:** Stand Alone is the default business object type as of platform version 3.2. Exercise caution if you work on an IBM TRIRIGA installation that is before platform version 3.2. The default business object type is Embedded, and it is not displayed as such.

- Embedded

Records that are created from an embedded business object can exist only in a smart section or a query section. Embedded records cannot be shared between two different smart sections. Any such attempt causes a copy of the embedded record to be put in the second smart section. If an embedded record is created but not placed within a smart section, the record is considered an orphan and deleted by the Cleanup agent.

- Link

Links can exist only in smart sections (also known as containers). Current practice is to avoid the use of link business objects as much as possible. You can use link records to blur the distinctions between different types of records in a module. You can also use them to provide an annotated view of records with views of specific fields or lookup values.

- Externally managed

These business objects are used to create the fact tables that store WPM metric data.



**Note:** If possible, use Stand Alone for your business objects. Avoid using the other types unless you are certain that a different type is more suitable for a specific purpose.

## Base business object

- A base business object is the first business object that you create in a module
- By convention, it has the same name as the module
- After you create a business object, you cannot delete, rename, or copy it
- Other business objects that are created in the same module inherit the sections, fields, state family, and associations of the base business object after you create the base business object

© Copyright IBM Corporation 2013

11

### Base business object

The first business object that you create in a module is the *base* business object. Later business objects inherit a copy of the sections, fields, associations, and state transitions from the base business object when you create those objects.

By convention, the base business object has the same name as the module. This convention exists because there is no property to designate a business object as a base business object. It is simply the business object that is created first in the module. Unless you follow this naming convention, it can be difficult to determine which business object in a module is the base business object. To emphasize the point: a business object is *not* the base business object because it has the same name as the module, but because it is created first in the module.

After you create the base business object, you can modify it. These changes are inherited by any business objects that are created after you make the changes. The changes are not automatically applied to any business objects that are created before the change. If you want the changes to be applied to those business objects, you must do so manually.

You cannot copy, delete, or rename base business objects. Attempting to perform these actions results in an error.

## Fields

- A field contains an individual, stored piece of data
- Every field has a data type and a unique name within a BO  
The suffix indicates the data type
- View the fields by clicking **View > Field List**

Field List			
General		Add   Find   Delete	
<input type="checkbox"/>	Field Name	Field Label	Field Type
<input type="checkbox"/>	<a href="#">triActiveEndDA</a>	Actual Retirement	Date
<input type="checkbox"/>	<a href="#">triActiveStartDA</a>	In Service	Date
<input type="checkbox"/>	<a href="#">triAdjustedRemainingVacantNU</a>	Adjusted Remaining Vacant Number	
<input type="checkbox"/>	<a href="#">triAllocationCountNU</a>	Allocation Count	Number

© Copyright IBM Corporation 2013

12

### Fields

A field contains one piece of data of a particular type; for example, number or text. Every field has a unique name within a business object. By convention, the name of a field has a suffix that indicates the type of the field. For example, a Text field has a suffix of TX, while a Number field has a suffix of NU.

Fields in a business object are grouped by section. Every business object has a General section that contains its fields. You cannot rename or delete the General section, but you can change its label on a form. Other sections in a business object are smart sections, and they contain only those fields that are defined in the smart section.

You can view fields and sections in a business object in the **Field** list. Open the field list by clicking **View > Field List**.

In the Field list, you can view the name, label, and type property of each field in a business object organized by the section that the fields are in. By default, the field list is sorted by the field name. Click one of the other headings to sort by that property. The name of each field is a link. Click it to open the properties of the field. At the top of the Field List panel are actions to **Add**, **Find**, and **Delete** the fields. These actions apply only to the General section and not to any other section.

## Adding fields to a business object

From the Field List panel, make the following selections:

**Add:** Create new fields

The screenshot shows the 'Field Properties' dialog with the 'Field Details' tab selected. It contains fields for Section (General), Field Type (Text), Name, Label, Description, Purpose, Required, and Do not Auto Populate.

**Find:** Copy existing fields

The screenshot shows the 'Field Search' dialog with the 'Search' button highlighted. It lists fields found for the name 'Start': Start (Datetime), Start Time (Text), and Status (Text).

© Copyright IBM Corporation 2013

13

### Adding fields to a business object

At the top of the Field List panel, you can see actions to add fields to a business object or remove them from it, based on the following criteria:

- **Add:** You create a field and add it to the business object. Specify the field properties, and click **Save Field**.



**Important:** Field names must be unique within the platform. If you try to add a field with the same name as a field that exists anywhere in the platform, an error occurs. Use **Find** if the field exists in the platform and you want it added to this business object.

- **Find:** You can reuse field definitions to maintain consistency between business objects. Use the Find action to search for fields that are defined in the platform, and copy them into the business object. All properties of the field, except the name and type, can then be modified as necessary.

When you search for multiple fields of the same type, such as System Read Only, change the **Type** value in the Field Search window to match. This change makes searches more efficient.

- **Delete:** Use **Delete** to remove a field from the business object.



**Note:** After you delete a field, it is still defined in the platform. Use **Find** to add it to a business object.

These actions are present only for the General section.

## Field properties

The screenshot shows two panels side-by-side. On the left is the 'Field Properties' dialog box with the title 'Field Details'. It contains several configuration fields:

Section	Value
» Field Type	Text
» Name	[Empty]
» Label	[Empty]
Description	[Large text area]
Purpose	[Empty]
Required	<input type="checkbox"/>
Do not Auto Populate	<input type="checkbox"/>
Result Column	<input type="checkbox"/>
Mobile Field	<input type="checkbox"/>

On the right is the 'Properties' panel titled 'PROPERTIES'. It lists various field properties with their current values:

Property	Value
Staging Table Field	<input type="checkbox"/>
Staging Table Key	<input checked="" type="checkbox"/>
Do Audit	<input type="checkbox"/>
Read Only	<input type="checkbox"/>
Default Value	[Large text area]
Validation	[Empty]
Size	150
Locator Field	<input type="checkbox"/>
Formula	<a href="#">Find</a> <a href="#">Add</a> <a href="#">Delete</a>

© Copyright IBM Corporation 2013

14

### Field properties

When you click the name of a field in the Field List panel, you see its properties in the Properties panel. You also see the Properties panel when you use the **Add** action to create a field. The following properties are the key properties for a field:

- **Field Type:** The Field Type is the most important property to specify for a field, because some of the other properties depend on this property. This property defines the type of data that the field can hold. The default setting is **Text**. The different field types are listed in the [Appendix A](#) on page 423.
- **Name:** The name of a field is the name in the platform. It cannot contain spaces or special characters. The naming convention for fields is to begin with **cst** for custom fields, and to include a suffix that represents the field type. An example is **cstCurrentLocationTX** for a current location (text) field.

The name of a field must be unique within the platform. Trying to create a field with a name that is already used in the platform causes an error.

- **Label:** The label is the text that the user sees for this field. The label can contain spaces and special characters.

- **Required:** A required field must have a value before you can save the record.
- **Do Not Auto Populate:** When you create a record, fields in the record are automatically populated from fields with the same name in the user's My Profile record. This behavior is a useful way to put the identity of the creator on the record, such as the location and organization. However, it can also cause problems with other fields, especially if you were not expecting it. Use the **Do Not Auto Populate** field property to prevent fields from automatically getting a value.



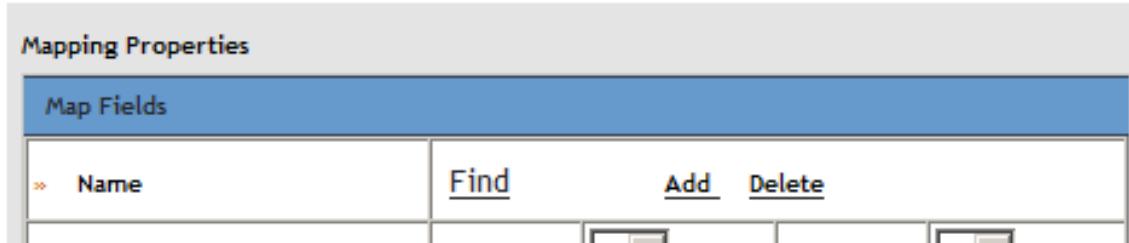
**Note:** It is good practice to always use this property unless you are certain that you want the field populated automatically. Do not assume that it is safe to ignore this property because the My Profile record does not have a field with the same name. The field might be added in the future, and the automatic populating would begin. Having a field that suddenly begins automatically populating can be a difficult problem to diagnose.

- **Result column:** You must select this property to use a field in a dynamic list. When you select this property, the field is automatically selected for any smart sections that are created from this business object.
- **Read Only:** When you select the Read Only check box, users cannot edit the field or set the value on forms. Other mechanisms, such as workflows, can change the value of this field.
- **Size:** The default size of a text field is 150 characters. The maximum is 1000.

## Defining the key to the records

Each record that is created from a BO must have a unique key:  
Name or Publish Name

- Use the **BO Mapping** tool to define these items:
  - Name for the BO
  - Starting point for the control number of the BO
- Click **Tools > BO Mapping** to open Mapping Properties panel



© Copyright IBM Corporation 2013

15

Defining the key to the records

## Defining the publish name

For every record, you must have a field or combination of fields whose value can uniquely identify that record. This value or combination of values is called the record's **name**. No two records that are created from the same business object can have the same record name.

The name is one of the mapping properties of a business object. These properties are defined in the BO Mapping tool, which is accessed by clicking **Tools > BO Mapping**.

The name is defined as a series of fields and separators. Start by clicking **Find**, which opens a selection window at the bottom of the panel. Choose the section and the field, and click OK. If you need only one field for the name, then you are done setting this property.

If the name contains more fields, you must next click **Add** and choose a separator. The last separator in the list, hyphen with spaces, provides the most visual separation. Append fields and separators until the name is complete.



**Note:** The platform compares only the first 100 characters of each name to determine whether records have the same name. If the first 100 characters match, then the names are considered to be the same, even if they differ after that point.

## **Defining the starting point for the control number**

Each business object has a control number. The control number is a counter that is incremented for each record that is created from that BO. The control number can have a text value as a prefix or suffix. The control number itself is stored as text.

Text is sorted from left to right, regardless of the length of the value. When the values in a text field are numeric, as the control number values are, they seem to be sorted incorrectly. For example, if the numbers 1, 2, and 10 are stored as text and sorted, the order is 1, 10, 2. People viewing this ordering of the numbers might report that the sorting is broken.

You cannot change this method of sorting, but you can avoid any confusion by beginning the control number at a higher value. This numbering method delays the onset of text-based sorting errors until many records are created.

After all of the mapping properties are set, click **Save Mapping**.

## Publishing a business object

- Business objects follow the Create-Publish-Revise lifecycle to make changes without affecting applications
- A business object must be published to be available for use
- To publish a BO, click **Tools > Publish BO**  
After you publish a business object, it is read-only
- To edit the BO, click **Tools > Revise BO**

© Copyright IBM Corporation 2013

16

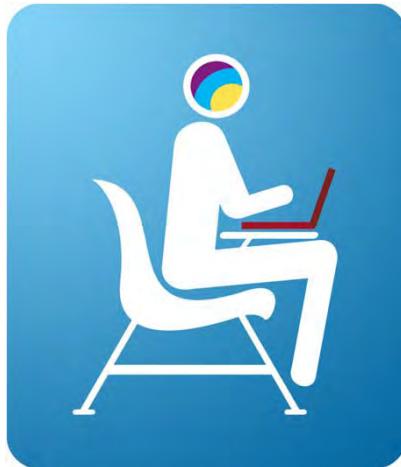
### *Publishing a business object*

Metadata is data that describes data. For example, records are data. Business objects describe records, so business objects are metadata. The pieces of metadata in an application, such as business objects, forms, and workflows, 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 prevent this problem, IBM TRIRIGA uses a **Create-Publish-Revise** lifecycle as a technique to allow metadata to be modified without breaking applications. The changes that you make to pieces of metadata do not actually take effect until you publish the metadata. No matter how many times you save the metadata that you edit, the previously published version is used until you publish the edited metadata.

After you publish something, it moves to a read-only state. You must use the Revise action to move it back to an editable state before you can change it. When you first create a piece of metadata, it is not available for use until you publish it.

## Student exercises



Perform the exercises for this unit

© Copyright IBM Corporation 2013

17

*Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

1. Name the five parts of a data model.
2. What business object type is almost always used?
3. How do you designate a base business object?
4. What do business objects inherit from the base business object? When does it happen?
5. If you move a panel header *off screen* in the Data Modeler, how do you regain control of it?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. Name the five parts of a data model.

*Fields; Business objects; Modules; State families; Associations*

2. What business object type is almost always used?

*Stand-alone*

3. How do you designate a base business object?

*It is the business object that is created first in the module*

4. What do business objects inherit from the base business object? When does it happen?

*Sections, Fields, State Family, and Associations. The inheritance occurs at the moment that the BO is created (using Save BO).*

5. If you move a panel off screen in the Data Modeler, how do you regain control of it?

*Use View > Reset to move the panels back to their default position*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

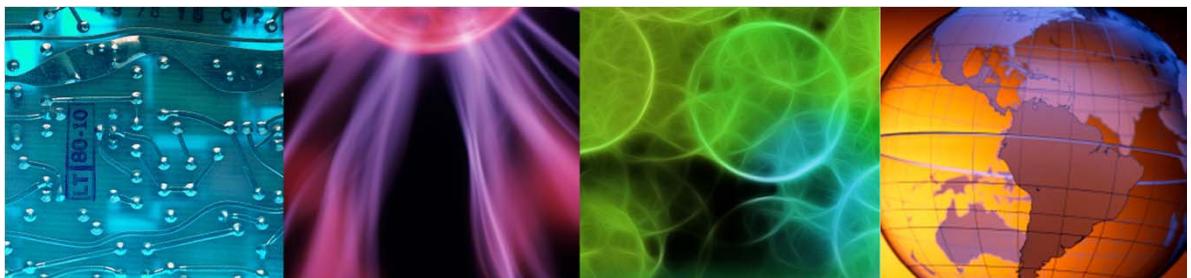
- Describe the components of a data model in IBM TRIRIGA
- Use the Data Modeler to create and edit a data model
- Publish and revise business objects



## 4 State families



## 4 State families



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

Data has a life cycle. In IBM TRIRIGA, this life cycle is called a State Transition family. This unit describes state transition families and shows you how to create and modify them.

**How you check your progress**

You can check your progress in the following ways:

- Review questions
- Lab exercises

**References**

*Application Building for the IBM TRIRIGA Application Platform*

---

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Describe a state family
- Add states and transitions to a state family
- Modify transition behavior by adding a sub action to it

# Lesson 1. Using state families



## Lesson 1: Using state families



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013. US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is about creating and modifying state transition families.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe a state family
- Add states and transitions to a state family
- Modify transition behavior by adding a sub action to it

### References

*Application Building for the IBM TRIRIGA Application Platform*



## Lifecycle of a record

- During the lifetime of a record, there is a sequence of states through which it can move
  - These states define the lifecycle of the record
- The state that a record is in changes in response to an action
  - Every state has a set of actions that are associated with it
- The collection of states and transition actions define the state transition family of the record

© Copyright IBM Corporation 2013

4

### *Life cycle of a record*

Human beings have a life cycle that consists of such phases as infant, toddler, child, adolescent, and adult. Records in IBM TRIRIGA also have phases in their life cycle. The phases, or states, that a particular record goes through in its lifetime depends on the business object from which it was created.

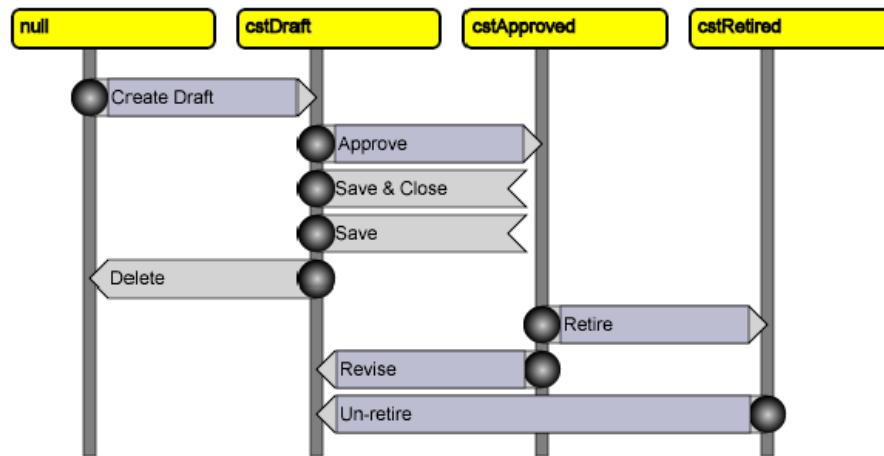
For example, a record that represents an asset might include the following states: null, Created, Assigned, Returned, Lost or Found. The states in this sequence define the life cycle of that record.

In IBM TRIRIGA, a new record always begins in the *null* state. The state of a record changes in response to an action. Every state in a life cycle has a set of state transition actions that are associated with it. These transition actions are the only actions that are valid at that point in the life cycle. Essentially, the transitions are the business rules of the state family, dictating where the record can go when it leaves a state.

The collection of states and transitions define the state transition family of the business object.

## State transition family

- A **state** identifies where a record is at in its lifecycle
- A **transition** is a way to get from one state to another
- A **state transition family** is a list of states and transitions that define the lifecycle of a business object



© Copyright IBM Corporation 2013

5

### State transition family

The state transition family identifies the states that exist in the life cycle of a business object and the means of moving from one state to another. All records that are created from a business object follow its life cycle.

A state family diagram is a representation of the state transition family. States are shown in diagrams as vertical lines with yellow boxes on top of them. Transitions are shown in a diagram as *arrows* or *flags*. The state that a transition originates from is indicated by a circular dot.

A transition that changes states is shaped like an arrow. It can point either to the left or right in the diagram, depending on the relative positions of the source and target states. Transitions that do not change states are shaped like a flag and always point to the right. These transitions include Copy, Save, and Save and Close.

The movement of a record in the state family is strictly controlled by the platform. Nothing, not even a workflow, can move a record from state to state in a way that is inconsistent with the state family. In other words, a record can only be moved directly from one state to another if there is a transition in the state family from the first state to the second state.

State transitions are the primary mechanism for saving records. A completed state transition saves the record, regardless of the name of the state transition or whether it changes states.

State transitions are also called actions. When you edit a record in a form, the state transitions are shown as action buttons on that form. When you see the Save, Activate, or Retire action buttons, the current state of the record has those state transition actions. Whether a state transition is visible to the user on a form is controlled by the properties of the transition.

## Tools for creating and modifying a state family

- State Family Manager for creating templates
  - Start from null family
  - Click **Tools > Builder Tools > State Family Manager**
- State Transition editor (in the Data Modeler) for business object-specific state family
  - Start from null family, or import from a BO or template
  - Click **Tools > BO State Transition**

© Copyright IBM Corporation 2013

6

### *Tools for creating and modifying a state family*

The platform has a tool to manage state transition family templates, called the **State Family Manager**. The different state transition family templates define common life cycles and processes that different types of business objects use. You can import state transition family templates into any business object.

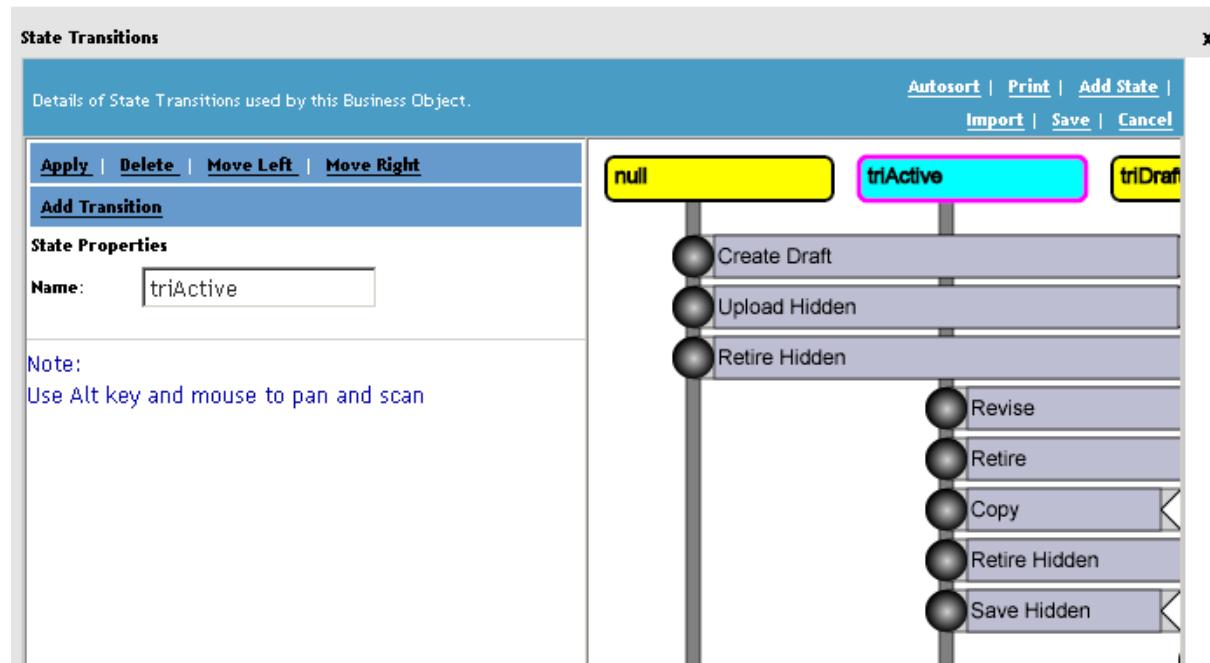
To access the State Family Manager tool, you click **Tools > Builder Tools > State Family Manager**.

The **State Transition** editor is a tool in the Data Modeler that you use to define and manipulate the state family of a business object, as follows:

- Create a new state family.
- Import a template and customize it
- Import a state family from another business object

States and transitions in an imported state family are merged with any existing states and transitions in the business object. You can access the State Transition editor from the Data Modeler by clicking **Tools > BO State Transition**.

## State Transition editor interface



© Copyright IBM Corporation 2013

7

### State Transition editor interface

From the Data Modeler, click **Tools > BO State Transition** to open the State Transition editor tool. This screen capture shows the interface of the State Transition editor.



**Note:** The screen capture of the interface was narrowed to fit the slide. The interface is not typically so compact.

The State Transition editor has two areas:

- The diagram and the actions that apply to it are on the right side of the window.
- The properties for the selected state, transition, or sub action are on the left side. This area also has actions that apply to the selected item.

States are shown in the diagram as a vertical bar with the name in a yellow box on top. Transitions, also called transition actions, or just actions, are shown as arrows between states. The origin of a transition is shown as a circle.

Moving around the diagram requires a technique that is known as **pan and scan**. When you press and hold the Alt key, the pointer changes from an arrow to a hand. Click and drag to move the diagram.

A selected state or transition has a blue background. In the slide, the triActive state is selected. Its properties are shown in the properties window.

To select items within the diagram, use the following guidelines:

- **State:** Click the yellow box.
- **Transition without a sub action:** Click anywhere in the transition.
- **Transition with a sub action:** Click the circle.
- **Sub action:** Click inside the box of the sub action.

You can move states to the left or right in the diagram by clicking the **Move Left** and **Move Right** links. These changes are not permanent until you use the Save action.

When states are moved, the actions automatically adjust position and appearance to preserve their connections between states. This movement might have unexpected results (an arrow that pointed to the right now points to the left), but its behavior is the same. A state at the far left of the diagram has all of its transitions at the top of the diagram and pointing to the right. Arranging the states in this way can be a convenient approach to working with transitions.

You can use **Autosort** to sort all the states alphabetically, except for the *null* state. It is positioned at the left of the diagram. These changes are not permanent until you use the Save action.

The **Apply** action applies changes to the diagram but does not save them permanently. There is no danger to using this button.

The **Save** action permanently saves changes, but *only* for a valid state family. See [Saving the State family](#). If you use the Save action when the state family is invalid, changes are discarded and the state family is rolled back to the last successful save.

You must close the State Transition editor window manually when you are done with it. Use the **Cancel** action or click the X in the upper right of the window to close it. Unsaved changes are discarded without notification when the window closes.



**Note:** Newly created records in the null state are not yet committed to the database. They might be deleted by the cleanup process unless they are moved out of the null state.

## Creating a state family in a business object

Start with the *null* state, or the state family that is inherited from the base business object

To Import a state family, click **Import** and select one of the following options:

[Import from State Family](#)

[Import from Business Object](#)

Add or remove states and transitions as needed

© Copyright IBM Corporation 2013

8

### *Creating a state family in a business object*

There are several starting points for creating a state family in a business object:

- Begin with the default state family, which consists of only the *null* state.
- Use the state family that is inherited from the base business object.
- Import a state family from another business object by clicking **Import > Import from Business Object**
- Import a state family template by clicking **Import > Import from State Family**

You must consider several important items when you import a state family:

- You are not limited to the current module when you import from another business object.
- The import process discards all sub actions from the state family that is being imported.
- Any existing states and transitions in the business object are merged with the imported state family. If you want only the imported state family in the business object, reduce the state family of the business object to *null* before you import.

To remove an existing state family, select a state and click **Delete**. Repeat for all of the states, even *null*. Ignore any residual transition actions. When you click **Save**, the diagram is refreshed with only the *null* state.

After your business object has a state family, modify it as needed by adding states and transitions, or by removing states or transitions.

## Saving the State family

**Important:** You can lose your work in the State Transition editor with no warnings or indications unless you perform the following steps:

- Ensure that you use the following rules for a valid state family:
  - At least two states, one of which is always *null*
  - At least one transition action between each state
  - No island states
  - No spaces or special characters in the name of states and transitions
- Click **Save** *only* when you have a valid state family
- Save the state family before adding a sub action

Use **Apply** to temporarily save changes

© Copyright IBM Corporation 2013

9

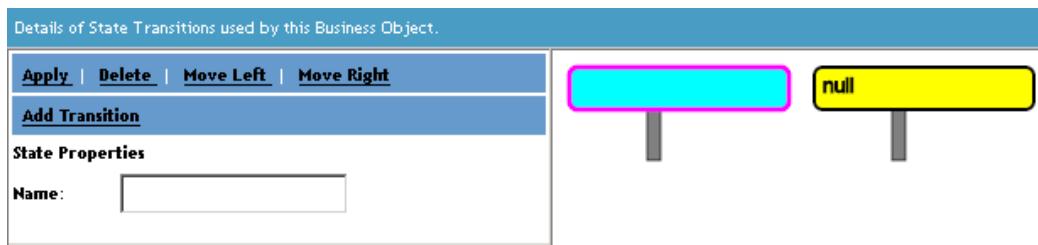
### Saving the State family

You can lose your work in the State Transition editor with no warnings or indications unless you perform the following steps:

- Use the following rules for a valid state family:
  - Have at least two states, one of which is always *null*.
  - Have at least one transition action between each state.
  - Have no island states, that is, no states that are not connected to other states by a transition.
  - Have no spaces or special characters in the name of states and transitions.
- Click **Save** *only* when you have a valid state family. The Save action validates the state family according to the rules. If the state family is not valid, it is rolled back to the last valid save. Use **Apply** to temporarily save changes. It does not validate the state family and does not discard changes. Use it as often as needed.
- Save the state family *before* you add a sub action. The process of adding a sub action to a state family also rolls the state family back to the last valid save.

## Adding a state

1. Click the **Add State** link 
2. Use the **Move Left** and **Move Right** links to position the state
3. Enter the name of the state in the **Name** field of the State Properties window
4. Click **Apply**



© Copyright IBM Corporation 2013

10

### Adding a state

You can add a state to a state family by clicking the **Add State** action. The new state is added to the state family. Enter its name in the **Name** field, without spaces or special characters, and click **Apply**.



**Important:** When you work with a state family in a business object, *never* click the Save action when you add a state. It does not have transitions and violates the *island state* rule, which results in the loss of your change.

You can move the state in the state family by using the **Move Left** and **Move Right** actions. You can also sort all of the states alphabetically by using the **Autosort** action.

## Adding a transition

1. Click the source state
2. Click **Add Transition**      **Add Transition**
3. Click the link for the destination state

The screenshot shows the 'Add Sub Action' dialog on the left and a state transition diagram on the right.

**Dialog Fields:**

- Action:** (empty text input field)
- Label:** (empty text input field)
- Class or Ejb Name:** (empty text input field)
- Method Name:** (empty text input field)
- Default Display:**
- Read Only:**
- Close Window:**
- Secondary Action:**

**State Transition Diagram:**

```

graph LR
    null((null)) --> cstNewState((cstNewState))
  
```

A yellow rounded rectangle labeled "null" is connected by a grey vertical line to a yellow rounded rectangle labeled "cstNewState". A thick pink arrow points from the "null" state to the "cstNewState" state.

11

### Adding a transition

You can add transitions to a state family by using the **Add Transition** action. The transition is created from the currently selected state to the state that you choose after you click Add Transition. Always follow these steps to add a transition to a state family:

1. Click the source state.
2. Click the **Add Transition** action. There is no change visible when you click this action.
3. Click the target state.

The transition is added to the state family, pointing from the source state to the target state. An example is shown in the slide. Enter the properties for the transition, and click Apply. Remember not to use the Save action unless you have a valid state family. If you want the transition to stay in the same state and have a *flag* appearance, use the same state for the source and target states.

The state transition properties are the same for either type of transition, as follows:

- **Action:** The name of the state transition action. The action name also indicates the name of the corresponding event that is triggered when the action is invoked. These events can be used to trigger asynchronous workflows. Actions that have the same name trigger the same event.

- **Label:** The label of the action button that is displayed on the form.
- **EJB Name:** This field is used to integrate a state transition action with a piece of the IBM TRIRIGA Application Platform internal logic. Do not put a value in this field unless told to do so by IBM TRIRIGA support personnel.
- **Method Name:** This field is used to integrate a state transition action with a piece of the IBM TRIRIGA Application Platform internal logic. Do not put a value in this field unless told to do so by IBM TRIRIGA support personnel.
- **Default Display:** Determines whether the action button automatically displays on the form.

State transition actions that are not displayed in menus are called *hidden actions*. Hidden actions do not have their Default Display property selected and typically use the term *hidden* in their name and label. Although not available to the user, they can still be used by workflows.

- **Read Only:** Determines whether the action is part of a read-only state.

You can use the Read Only attribute to make a record read-only when it is at a specific state. For a record to be read-only, all visible state transition actions that originate from the read-only state must have their Read Only property selected.

- **Close Window:** If this property is selected, the form closes when a user clicks the action button for this transition. This property can be overridden by a Modify Metadata workflow task.
- **Secondary Action:** Controls whether the actions associated with this transition are automatically put in the More menu for the form. Setting this property makes it difficult to accidentally trigger this action.

Some properties, such as Default Display, Read Only, Close Window, and Secondary Action, can be set differently in the state family of the form versus the business object. In those cases, the properties in the form are used.

## Status

Transitions that change states must be set to update the **Status** and **Previous Status** fields on the record

- The TRIRIGA applications use status for various purposes
- You can use status to help diagnose issues with records
- The **Status** field is typically updated in a sub action

Destination State	Status
null ( <i>Note: This is all lowercase</i> )	Deleted
triDraft	Draft
triReview	Review In Progress
triActive	Active
triRevision	Revision In Progress
triRetired	Retired

© Copyright IBM Corporation 2013

12

### Status

The IBM TRIRIGA Application Platform controls the state that a record is in. The names of these states might not be clear to users. By convention, each state has a corresponding status value that is more meaningful to the users. For example, the triReview state has a status of *Review In Progress*, and the triRevision state has a status of *Revision In Progress*.

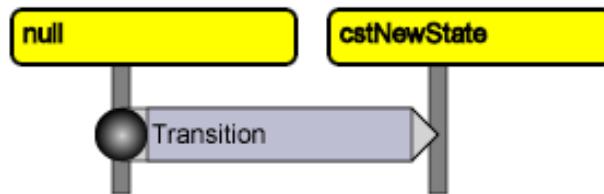
The status is used for various purposes; for example, diagnosing problems. Some state families have different paths to the same state and knowing the path it followed can be important.

Transitions that change states must be set to capture the status at the new state and store it in a field called **Status**. They must also capture the status at the previous state and store it in the field called **Previous Status**. The capture of these values is handled in a sub action.

Transitions that do not change state do not capture the status value. The status is the same before and after the action is performed. The states and status values in the slide are only the more commonly used ones. There are more than 100 status values in the Status classification.

## Sub actions

- A sub action is defined on an action
  - When the action is triggered, the sub action is also triggered
- You can use a sub action for these purposes:
  - Set the value of a field (for example, Status)
  - Run a workflow
  - Activate logging
- The label of a sub action is on the form instead of the action label



© Copyright IBM Corporation 2013

13

### Sub actions

A sub action is part of a business object. It is defined on an action and is triggered when the action is triggered. There can be more than one sub action defined on a single action. A sub action is shown as a box on an action. Click inside the box to see the properties of the sub action. Click the circle to see the properties of the action.

A sub action can perform any or all of these actions when it is triggered:

- Set constant values for fields in the business object; for example, the **Status** and **Previous Status** fields.
- Trigger a synchronous workflow.
- Activate logging for the action, if Audit Actions is specified.

A sub action can modify the presentation of an action in the menu in two ways:

- If only one sub action is associated with an action, then the label of the sub action is used instead of the label of the action. It is good practice to have the label of a single sub action the same as the label of the action.
- If multiple sub actions are associated with an action, their labels show in a menu.

## Adding a sub action

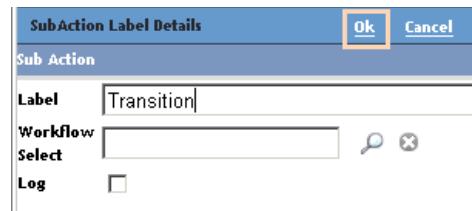
1. Select a transition



2. Click the Add Sub Action link

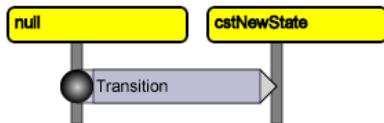


3. Set the label for the sub action



4. Set any other properties

5. Click Ok



### Adding a sub action

To add a sub action to a transition in the State Family Builder, follow these steps:

1. If you have unsaved changes in the state family, click **Save** before you add a sub action.
2. Select a transition by clicking it. The background color of the transition turns to blue in the diagram to show that it has focus.
3. Click the **Add Sub Action** link. The properties window for the sub action is displayed.
4. Set the properties for this sub action:
  - **Label:** Text that displays in menus for the action that triggers the state transition action.
  - **Workflow Select:** Select a synchronous workflow that begins when this action is clicked.
  - **Log:** If Audit actions are turned on for the business object, this property controls whether this state transition action is audited.

There are other properties that can be set in a sub action, but the sub action must be saved before those properties can be set.

5. After you set the properties of the sub action, click **OK**. The OK action saves the sub action and rolls the state family back to the last save point.

## Capturing status in a sub action

1. Click the sub action
2. Click **Open**
3. Click **Add** in the State Transition Attribute section
4. Select **Status and Previous Status**
5. Click **Ok**
6. Set the **Constant Value** fields
7. Click **Ok**



State Transition Attribute		Add   Delete
Select	Field Label	Constant Value
<input type="checkbox"/>	Status	Draft
<input type="checkbox"/>	Previous Status	

© Copyright IBM Corporation 2013

15

### Capturing status in a sub action

After you create a sub action, you can set other properties within it.

1. Open the sub action by clicking inside the box to select it.
2. Click the **Open** action.

Setting values in fields is performed in the State Transition Attribute section of a sub action. A typical scenario is to set values for Status and Previous Status as a record moves from one state to another.

3. Click the **Add** action to open a selection list of fields that are contained in the business object.
4. Select the check boxes next to the fields that you want to select.
5. Scroll to the top of the list, and click **OK**.  
The selected fields are added to the section.
6. Enter the value in the Constant Value column that the field is to receive when this state transition action is triggered. For classification fields, such as Status, the Constant Value must be one of the values in the classification hierarchy.

---

## Student exercises

---



© Copyright IBM Corporation 2013

16

### *Student exercises*

Perform the exercises for this unit.

## Checkpoint questions

1. What is a state family?
2. What are the four rules for a valid state family?
3. What happens if you break the rules?
4. Name the four properties of a transition that is selected by a check box.
5. Name three uses for a sub action.

© Copyright IBM Corporation 2013

17

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.



## Checkpoint answers

---

1. What is a state family?

*A state family defines the lifecycle of records that are created from a BO. It is composed of states and transitions (also called actions).*

2. What are the four rules for a valid state family?

*At least one state other than null; At least one transition between each state; No island states; all names follow naming convention.*

3. What happens if you break the rules?

*The state family is rolled back to the previous save with no warning.*

4. Name the four properties of a transition that is selected by check box.

*Default Display; Close Window; Read Only; Secondary Action.*

5. Name three uses for a sub action.

*Capturing field values; Running a workflow; Logging.*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Define a state family
- Add states and transitions to a state family
- Modify transition behavior by adding a sub action to it





## 5 Forms and the Form Builder

---



## 5 Forms and the Form Builder



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

### What this unit is about

IBM TRIRIGA separates the presentation of data from the definition of data. This unit shows you how to define the presentation of data that is contained in business objects.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building for the IBM TRIRIGA Application Platform*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Use the Form Builder tool
- Create a form that contains tabs, sections, and fields
- View and modify the state family of a form

# Lesson 1. Using forms and the Form Builder



## Lesson 1: Using forms and the Form Builder



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

In this lesson, you learn how to use the Form Builder tool and create and modify a form.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

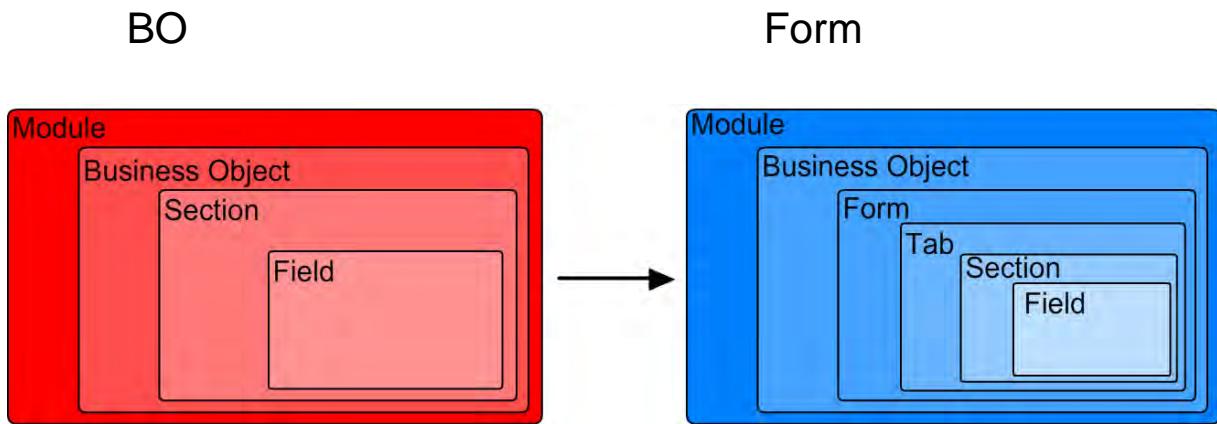
- Use the Form Builder tool
- Create a form that contains tabs, sections, and fields
- View and modify the state family of a form

### References

*Application Building for the IBM TRIRIGA Application Platform*

## Forms

- A form is the presentation layer of a business object
- You use forms to create and edit the contents of records
- Both BOs and forms have a similar layered structure



© Copyright IBM Corporation 2013

4

### Forms

The IBM TRIRIGA Application Platform separates the definition layer of a record from the presentation layer. The definition layer holds the field definitions while the presentation layer displays those definitions. A form is the presentation layer of a record. Forms are how the user can create, view, and edit the contents of records. The business object is the definition layer of a record.

Each form is associated with a single business object. A business object can be associated with multiple forms, allowing for different views of the same records.

A form is organized in a hierarchy. Each part of the hierarchy has its own properties. When you create a form, begin at the top and work down to the lowest details, as follows:

- **Form:** A form contains one or more top-level tabs.
- **Tabs:** A tab can contain one or more sections. Top-level tabs take up all of the available area in the window (as compared with multi-tab sections, which fit on a tab).
- **Sections:** The purpose of a section is the presentation of data. Sections come in various types, including form, smart, query, and multi-tab. A form section contains fields and buttons, as does a smart section. A multi-tab section can contain other sections.

- **Fields and 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.

## Example of a record in a form

(Required): Manage general information about the employee.

**General**

ID	AA - 0001	Status	Revision In Progress
* Last Name	Abstractor	* First Name	Abe
Full Name	Abe Abstractor		
Prefix	Mr.	Nick Name	

**Detail**

Date of Hire	RE Abstractor	25	X	D
Title	Staff			
Functional Role				
Assignment Type	Assigned			
Maintenance Priority				

Last Reported Problem Location

**Reports To**

Reports To
------------

© Copyright IBM Corporation 2013

5

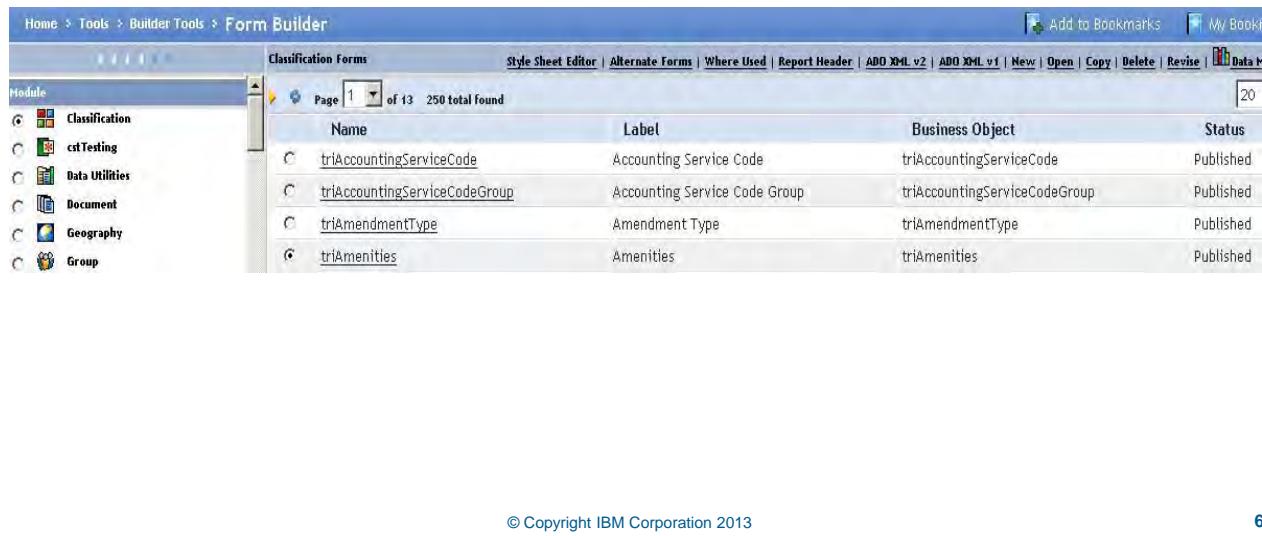
### Example of a record in a form

When you look at a record in a form, notice these features:

- Tabs:** Each of these tabs is a top-level tab that contains a page of information. Some forms have so many tabs that not all of them can be seen at the same time. To see other tabs than the ones shown, use the left and right arrows to scroll through the tabs. You can also click the chevron (the double-downward arrow) at the right of the tabs to see a selectable list of all tabs.
- Action buttons:** These buttons are created from the definitions in the state transition family.
- Instructions:** They are located below the tabs and above the first section. The instructions are defined in the tab properties and provide guidance to the user about the purpose of the tab and any important features. The instructions are useful for tabs that are rarely visited.
- Sections:** In the example in this slide, the General and Detail sections are Form sections, and the Reports To section is a smart section.
- Runtime features:** This example has a field named Date of Hire that has a calendar **picker**. You use pickers to select values for fields. You see them only when the form is used to display a record. You do not see them in the Form Builder tool.

## Form Builder

- You define forms in the Form Builder tool  
**Click Tools > Builder Tools > Form Builder**
- Forms are organized by module



The screenshot shows the 'Classification Forms' page in the Form Builder tool. The left sidebar lists modules: Classification, cstTexting, Data Utilities, Document, Geography, and Group. The main area displays a table of forms with columns: Name, Label, Business Object, and Status. The table contains four rows:

Name	Label	Business Object	Status
<a href="#">triAccountingServiceCode</a>	Accounting Service Code	triAccountingServiceCode	Published
<a href="#">triAccountingServiceCodeGroup</a>	Accounting Service Code Group	triAccountingServiceCodeGroup	Published
<a href="#">triAmendmentType</a>	Amendment Type	triAmendmentType	Published

Page 1 of 13 250 total found

© Copyright IBM Corporation 2013

6

### Form Builder

The Form Builder tool, commonly referred to as *Form Builder*, is used to manage forms. You access the Form Builder tool by clicking **Tools > Builder Tools > Form Builder**.

With the Form Builder, you organize forms by the module that contain their associated business object. The left side of Form Builder has a list of modules. When you select a module, the Form Builder displays a list of the forms that are associated with the selected module.

Each form must have a unique name within a module and must be based on a single business object. Multiple forms can be based on the same business object.

You typically use the following primary Form Builder actions:

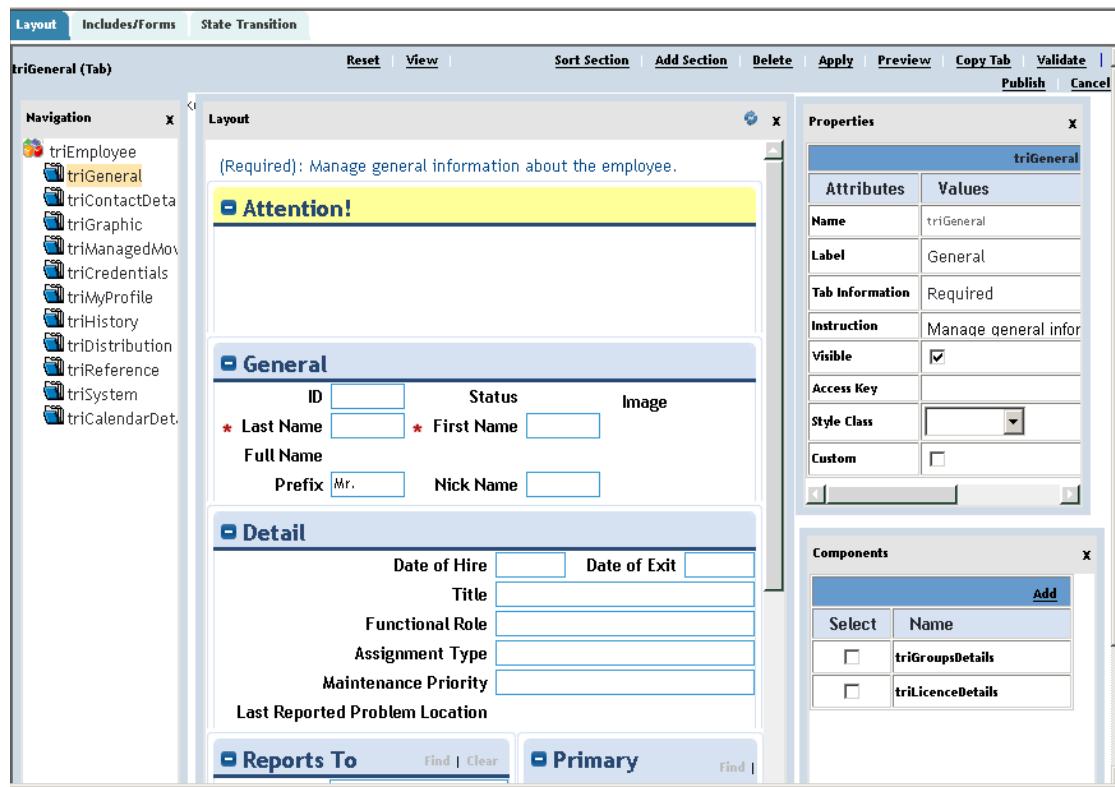
- **New:** Create a form that is associated with the selected module.
- **Open:** Open the selected form. Clicking the form name also opens the form.
- **Copy:** Copy the selected form. You are prompted to select the target business object and to name the form.



**Note:** The label of the form remains the same after it is copied. A common mistake is to forget to change the label after you copy a form.

- **Delete:** Delete the selected form.
- **Data Modeler:** Navigate to the Data Modeler tool from the Form Builder.

## Form wizard interface



© Copyright IBM Corporation 2013

7

### Form wizard interface

When you create or open a form, the window to configure its properties opens. This window is called the **form wizard**.

The form wizard is organized into three tabs:

- **Layout tab**

Use this tab to specify the presentation structure of a form.

- **Includes/Forms tab**

The Includes portion of this tab is used to describe the connections of forms that are part of a hierarchy. In other words, it is where the parent/child relationships of the form are registered.

The Forms section of this tab is used to specify which Form Reports can be used in the Reports tab of the form.

- **State Transition tab**

- Use this tab to manage the state transition family of the form.
- Use this tab to control the available state-based actions.

## Layout tab

The Layout tab is the default tab and the one where all of the properties of the form are defined. It is organized into the following four panels:

- **Navigation panel:** Displays the form in a hierarchical manner. Each part of the hierarchy, or node, represents an element of the form. Each node can be expanded to show the elements that it contains. When you select an item in the navigation panel, you bring **focus** to it.
- **Layout panel:** Shows a mock-up of the form. The layout panel gives a pictorial representation of the same information that is seen in the Navigation panel. Elements can be selected in the Layout panel to bring focus to them.
- **Properties Panel:** Shows the properties of whatever component of the form has focus.
- **Component Panel:** Displays a list of components that can be added to the element in focus. When you select a Form section, this panel displays a list of fields that are defined in the business object and not used on the current tab. If a tab or multi-tab section has focus, this panel contains a list of smart sections.

The size, shape, and position of these panels is manipulated in the same way as in the Data Modeler: You click a heading and drag it to move the panel. You hold the Shift key and click a heading to resize the panel, and you click the side of a panel and drag it to resize the panel.

These actions are available on the Layout tab:

- **View:** Displays any of the four panels that are closed.
- **Reset:** Returns panels to the positions they were in the last time the form was saved.
- **Delete:** Deletes the component that has focus.
- **Apply:** Applies changes, saves the form.
- **Preview:** Displays the form. Does not show runtime features such as pickers.
- **Publish:** As with business objects, forms follow the Create-Publish-Revise life cycle. Use Publish to make the form available to other people and processes.
- **Revise:** Puts the form into an editable state.
- **Cancel:** The Form Wizard window does not close when you publish a form. Use this action to close the window, or click the X in the corner of the window.

## Creating a form

1. Open the Form Builder
2. Select a module
3. Click **New**
4. Set form properties
5. Click **Apply**

Properties	
Attributes	Values
Business Object	cstCar
Name	cstCar
Label	Car
Description	
Default Form	<input checked="" type="checkbox"/>
Single Tab	<input type="checkbox"/>
Pre-Load Workflow	
Allow bookmark to create record	<input checked="" type="checkbox"/>
Allow bookmark to specific record	<input checked="" type="checkbox"/>
Calculate Excel	<input type="checkbox"/>
Show Association	<input checked="" type="checkbox"/>
Show Workflow Instance	<input checked="" type="checkbox"/>
Show Reports	<input checked="" type="checkbox"/>

© Copyright IBM Corporation 2013

8

### Creating a form

You create forms in the Form Builder by using these steps:

1. Navigate to the Form Builder by clicking **Tools > Builder Tools > Form Builder**.
2. Select the module that contains the business object that the form is based on.
3. Click **New**.  
This action opens the Form Wizard with a blank form.
4. Set the properties of the form, and click **Apply**.

The following are the key properties to set when you create a form:

- **Business object:** The name of the business object that the form is associated with. After the form is saved, this value cannot be changed. This property is required.
- **Name:** The name that uniquely identifies the form. By convention, the default form for a business object has the same name as the business object. This property is required.
- **Label:** The text that is displayed to the people who are using the application.

- **Default form:** Select this check box to identify this form as the default form for the business object.
- **Allow bookmark to create record:** If this check box is selected, the form can be added to the user's bookmarks, providing a shortcut for making records with this form.
- **Allow bookmark to specific record:** If this check box is selected, records created from this form can be added to the user's bookmarks, providing a shortcut to them. This property must be selected for records from this form to be displayed in the Last Visited portal section.
- **Show Association:** If the check box is selected, the form includes an Association tab. The Association tab graphically shows a record's association with other records.
- **Show Workflow Instance:** If this check box is selected, the form includes a Work Flow Instance tab. The Work Flow Instance tab shows all workflows that ran against the current record.
- **Show Reports:** If the Show Reports check box is selected, the form includes a Reports tab. On a Reports tab, you can view the content of a record as a form report.

## Adding a tab to a form

1. Click Add Tab
2. Set tab properties
3. Click Apply

Properties	
Attributes	Values
Name	cstSystem
Label	System
Tab Information	System Information
Instruction	Review the system information
Visible	<input checked="" type="checkbox"/>
Access Key	S
Style Class	<input type="button" value="▼"/>
Custom	<input type="checkbox"/>

© Copyright IBM Corporation 2013

9

### Adding a tab to a form

The tab is the next presentation layer after the form is created. You must add it to the form before you add sections and fields. There are two types of tabs: top-level tabs and section tabs. Top-level tabs occupy the entire display area and are created when a *form* has focus. Section tabs are defined in a multi-tab section and are created when the *section* has focus.

When the focus is at the form level, the **Add Tab** action is visible. Click it to add a top-level tab to a form.

When you create a top-level tab, you set the following key properties:

- **Name:** The name of the tab.
- **Label:** The label that people see for the tab.
- **Tab Information:** Guidance for the user. It is shown at the top of the tab in parentheses.
- **Instruction:** Extended guidance for the user. It is shown after the Tab Information value. It is used primarily on tabs that are rarely visited, to give instructions or guidance to the user.
- **Visible:** Selected by default. When cleared, the tab is not visible to the user unless this property is modified by a workflow.

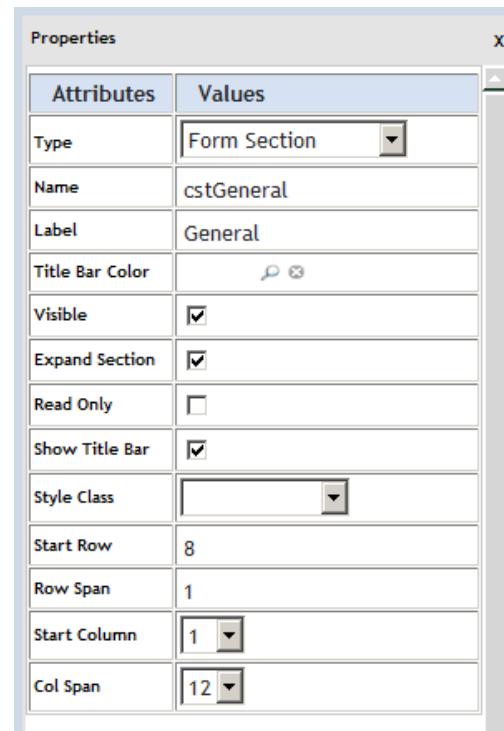
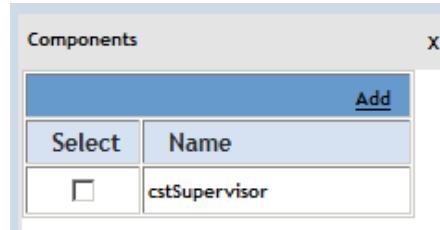
- **Access key:** A single letter that becomes a keyboard shortcut to the tab. When you have a record open in a form, press and hold the Alt key and click the letter of the shortcut to go to the tab.

## Adding a section to a tab

Use one of the following tasks:

1. Click **Add Section**
2. Set section properties
3. Click **Apply**

1. Select smart sections from the Components panel
2. Click **Add**



© Copyright IBM Corporation 2013

10

*Adding a section to a tab*

The **Add Section** action is visible when a tab has focus. Click it to create a section.



**Note:** When you add a section to a form, you cannot see it in the Layout panel until it has fields.

When you create a section, you set the following key properties:

- **Type:** There are various section types that present different kinds of information. The most commonly created section types include the following ones:
  - **Form section:** This type of section can contain fields and buttons.
  - **Multi-tab section:** A multi-tab section cannot directly contain any fields, only other sections. A tab is created for each section in a multi-tab section.
  - **Query section:** Displays the results of a query.
- **Name:** The name of the section. Cannot contain spaces or special characters.

- **Label:** The text that is displayed to the user. Can contain spaces and special characters.
- **Title Bar Color:** The default title bar color is a light blue. It is common to change the color for the Attention section to make that section more noticeable.
- **Visible:** If this check box is selected, the section is visible. If the check box is not selected, the section is hidden. Workflows can change the value of this property, making the section hidden or visible as appropriate. This behavior is common with the Attention section.

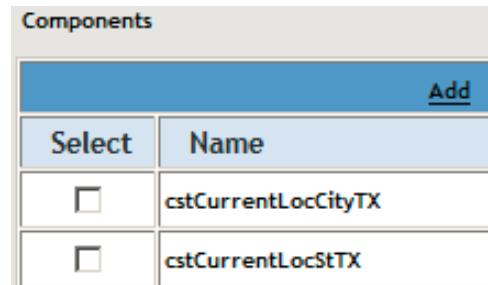
As an alternative to creating a section, you can add smart sections to a tab by using the Components panel. Select one or more sections from the list in the Components panel and click **Add**. The selected section or sections are added to the form.

All fields that are defined in the business object for the smart section are included in the section on the form. If you do not want a field to be displayed in a smart section, select the field and click the Delete action. Deleting the field removes it from the form but not from the smart section in the business object.

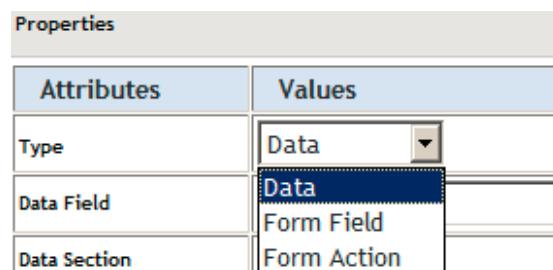
## Adding fields to a Form section

Use one of the following tasks:

1. Select business object fields from component list
2. Click Add



1. Click **Create Field**
2. Select the type
3. Set field properties
4. Click **Apply**



### Adding fields to a Form section

When a Form section or Smart section has focus, you see fields in the Components panel that are defined in the business object and can be added. Select one or more of these fields and click the **Add** link. A Data field is created in the form for each selected field, in the order in which you select them from the Components panel.

You can also add fields manually by using the **Add Field** action. It is visible when a section has focus. You can add the following three types of fields:

- **Data:** A field that is explicitly connected to a field in the associated business object.
- **Form Field:** A field that is *not* explicitly connected to any field in a business object.

Form fields are commonly used as *fillers* when you arrange fields in a section. They can be created with a blank label to occupy a position in a section without being seen.

- **Form Action:** This field is actually a button that you can put in a section and use to trigger a workflow.

For convenience, you might want to include the same field in more than one tab. For example, you can put the user message field on every tab, in case there is a message to display. You can use the

Form Builder to include the same field in multiple tabs, but only once per tab. You cannot publish a form with multiple copies of the same field on a single tab.

## Data field properties

Properties	
triUserMessageTX	
Attributes	Values
Data Field	triUserMessageTX
Data Section	General
Name	triUserMessageTX
Label	Attention!
Type	Text
Minimum Size	0
Maximum Size	0
Use Custom Display Mask	<input type="checkbox"/>
Input Width	Default <input type="button" value="▼"/>
Label Style Class	labelStyleClass1 <input type="button" value="▼"/>
Data Style Class	dataStyleClass2 <input type="button" value="▼"/>
Data Alignment	Left <input type="button" value="▼"/>

Start Row	1
Row Span	3
Start Column	1 <input type="button" value="▼"/>
Col Span	1 <input type="button" value="▼"/>
Group Seq	0
Tab Order	0
Required	<input type="checkbox"/>
Hide Label	<input checked="" type="checkbox"/>
Position	Both <input type="button" value="▼"/>
Visible	<input checked="" type="checkbox"/>
Read Only	<input type="checkbox"/>
Type Field	-Select- <input type="button" value="▼"/>
Content Field	-Select- <input type="button" value="▼"/>
Required Field	-Select- <input type="button" value="▼"/>
Source Details...	

© Copyright IBM Corporation 2013

12

### Data field properties

This slide shows all of the properties of a Data field. The properties of a Form field are similar but omit the properties that connect it to a field in the business object.

You typically add data fields to a form from the Components panel. The most common changes to the field properties involve the positioning of the field.

The following key properties are the ones that you use to set or modify a Data field:

- **Data Field, Data Section, Name, Type:** These properties come from the business object field that this data field is connected to. You cannot modify them in the form.
- **Label:** The label that is displayed for the field on the form.
- **Required:** If this check box is selected, a user cannot save the record until they enter a value into this field.
- **Visible:** If this check box is selected, this field is visible. If the check box is not selected, this field is hidden. Workflows can change the value of this property, making the field hidden or visible as needed.

- **Read Only:** If the check box for this property is selected, a user cannot directly edit the value of the field.



**Note:** If a field is read-only, it does not have a border. An editable field has a border.

- **Source Details:** This property is a link to the properties of this field in the Data Modeler.

Some field properties, such as Required and Read Only, are found in both the form and the business object. In these cases, the property in the form can be more restrictive than the property in the business object, but not less restrictive. Here are two scenarios that explain this concept:

Scenario 1: In the business object, the Required property for a field *is not* selected. The same property in the form can be selected or not selected. If the property is selected, the form is more restrictive than the business object.

Scenario 2: In the business object, the Required property for a field *is* selected. The property in the form cannot be less restrictive. In the form, it is automatically selected and also locked. In fact, the property is double-selected in the form, which means that clearing the check mark in the business object does not also clear the check mark in the form. It unlocks the property in the form, but the check mark remains and you must clear it manually.

## Positioning fields in a section

By default, the fields are in two columns with no more than 12 columns

The screenshot shows a 'General' section with the following fields:

- \* Make
- \* Year
- Car Category
- Name
- \* Model
- \* ID
- VIN 0
- Status

Position fields by using **Start Row**, **Row Span**, **Start Column**, **Col Span**

Start Row	2
Row Span	1
Start Column	2 ▾
Col Span	1 ▾

© Copyright IBM Corporation 2013

13

### Positioning fields in a section

A Form section is laid out in a grid that is 12 columns wide and has as many rows as needed to hold the data. When you add fields to a section from the Components list, they are placed in the first two columns in the order that you selected them.

If needed, you can manually position the fields in a section by using these properties:

- **Start Row:** Specifies in which of the section's rows the field is shown. The rows in sections are numbered from the top to the bottom: row 1 is above row 2.
- **Row Span:** Specifies the height of the field in rows.
- **Start Column:** Specifies in which of the section's columns this field is displayed.
- **Col Span:** Specifies the width of the field in columns.

If a section has nothing in a column, the column has no width and it looks like the column is not there. The same applies to a row. In other words, blank rows and columns are *squeezed out* of a section. For example, if you position a field in column 2 of a row and have nothing in column 1, the field is seen in column 1. Similarly, if you have fields in row 2 and nothing in row 1, the fields are seen in row 1.

You can use form fields to help with field placement by filling a position. They do not contain a value, and if you use a blank for the name they are not visible. If multiple fields are targeted for the same position, the platform randomly puts one of the fields there and the other field next to it. Ensure that you have distinct positions for fields.

If fields are positioned in a section and you must position another field on a line between them, use the **Insert Row** action. It increments the row number of all fields on the line where it is used, and all fields below that line. This action has no visible effect on these fields, but it makes a line available for a field, which must be manually positioned on the line.

## Other actions

- Sort tab
- Sort section
- Sort field

Order	Sequence	Value
▼	1	cstAttention
▼ ▲	2	cstGeneral
▼ ▲	3	cstDetails
▼ ▲	4	cstSupervisor
▲	5	cstMultiTab

- Copy tab

Apply | Cancel

Name»	cstGeneral
Target Business Object»	cstCar
Target Form»	cstCar

- Copy section

Apply | Cancel

Name»	cstGeneral
Target Tab»	cstGeneral
Target Multi Tab	

© Copyright IBM Corporation 2013

14

### Other actions

You can perform other actions by using the following options in the Form Builder:

- **Sort tab:** Use this action to sort all of the tabs in a form. This action is available when the form has focus. Clicking it opens a window that lists all of the tabs of the form. Use the arrows in the Order column to change the order of the tabs. Click Apply when done. This action is the only mechanism for changing the order of tabs.
- **Sort section:** Use this action to sort all of the sections in a tab. This action is available when a tab has focus. Clicking it opens a window that lists all of the sections on the tab. Use the arrows in the Order column to change the order of the sections. Click Apply when done.
- **Sort field:** Use this action to sort all of the fields in a section. This action is available when a section has focus. Clicking it opens a window that lists all of the fields of the section. Use the arrows in the Order column to change the order of the fields. Click Apply when done.



**Note:** This action puts all of the fields in a single column in the section.

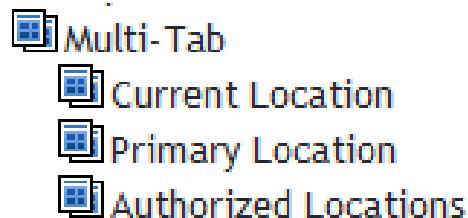
- **Copy tab:** Use this action to copy a tab to any form in the same module. This action is available when a tab has focus. Clicking it opens a dialog box to select the target business object and form.
- **Copy section:** Use this action to copy a section to any tab in the same form. This action is available when a tab has focus. Clicking it opens a dialog to select the target tab and specify the section name.

## Multi-tab sections

- Used to group multiple sections in the same area of the form
  - Reduces the need to scroll vertically to access the data
  - Often used to organize the display of nonessential data



- In Form Builder, it looks like a section; you use it like a tab



© Copyright IBM Corporation 2013

15

### Multi-tab sections

When you create a section, you can select **Multi-Tab**. Multi-tab sections are used to group multiple sections in the same area of the form. Using a multi-tab section reduces the need to scroll vertically to access the data and makes for a more compact display of information.

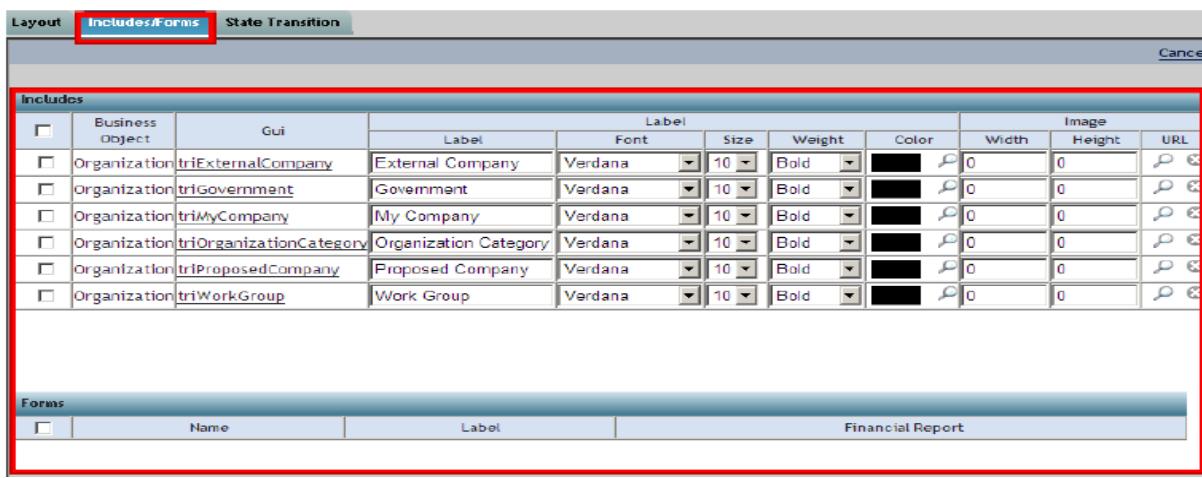
Top-level tabs have their own properties. Tabs in a multi-tab section do not have their own properties; they are controlled by the properties of the multi-tab section that contains them.

Sections are added to a multi-tab section in the same manner as for a top-level tab. You select the multi-tab section from the navigation panel and click **Create Section**, or add sections from the Components panel. A tab is created in the multi-tab section for each section added to it. You can add only one section per tab.

Because users do not readily see the data in multi-tab sections, the multi-tab format is often used to organize the display of nonessential data. It is good practice to not contain critical data in a multi-tab section.

## Includes/Forms tab

- **Includes:** Used by forms in hierarchy modules to control the hierarchy menu structure
- **Forms:** List of reports that you can run from the **Print Preview** tab of the record



© Copyright IBM Corporation 2013

16

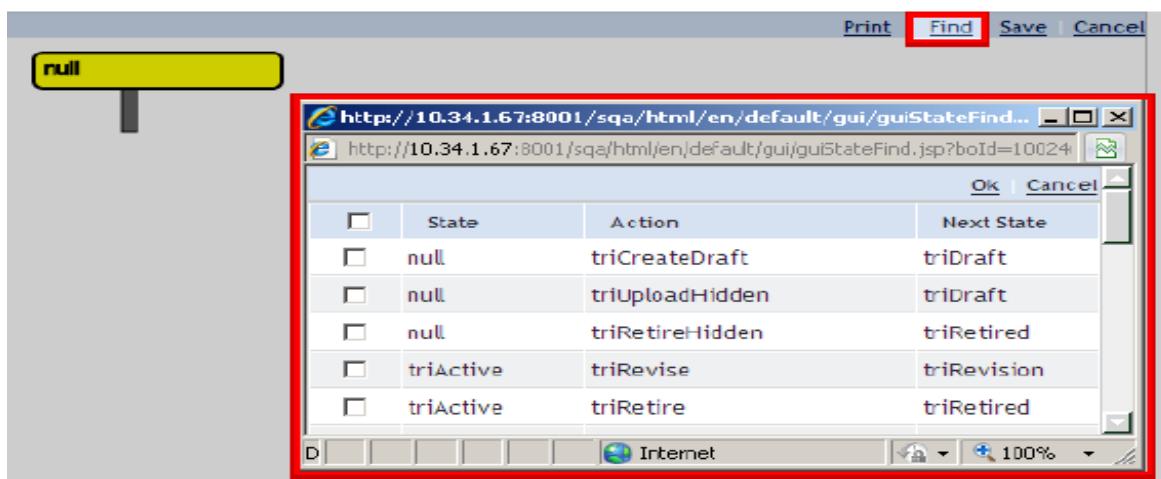
### Includes/Forms tab

The Includes/Forms tab of a Form Wizard window is used for two separate purposes:

- The top section is labeled **Includes**. You use it to control how records that are part of a hierarchy are displayed in a manager, and control which form to use for them. This determination is based on the type of parent they have.
- The bottom section is labeled **Forms**. You use it to specify what form report templates are available for use in the Reports tab.

## State Transition tab

- A form has a state family based on the BO state family
- Form state family can be subset of BO state family, never more
- Use the **Find** link to add any states or transitions that exist in the BO but not in the form



© Copyright IBM Corporation 2013

17

### State Transition tab

A form begins with the state transition family of the business object that it is based on. You can use the State Transition tab of the Form Wizard to change the state family in many ways. You *cannot* add states and transitions that are not in the business object. The state family of a form can contain fewer states and transitions than the state family of the business object, but never more.

You can modify the state family of a form in the following ways:

- Make only part of the state transition family available through the form

To limit the state transition family, you remove selected states and transitions from the state family of the form. Select a state or transition that you want to remove and click the **Delete** action. If you delete a state, all transitions that are connected to that state are also deleted.

- Change the properties of a transition

Click any transition, and a properties window opens. You can change the settings of the **Default Display**, **Close Window**, **Read Only**, and **Secondary Action** properties. You can also set a keyboard shortcut for the transition in the **Access Key** property. Set this property to a single lowercase letter. When you view a record with this form, press and hold the Alt key and click the key for that letter to trigger the transition.

- Cause a state transition to create a record

Each transition has a Module and Form property. If these properties have values, then when the transition is triggered, a new record is created with the specified module and form.

- Find missing states and transitions

Some transitions or states are defined in the state family of the business object but are not included in the state family of the form. This circumstance can happen for several reasons:

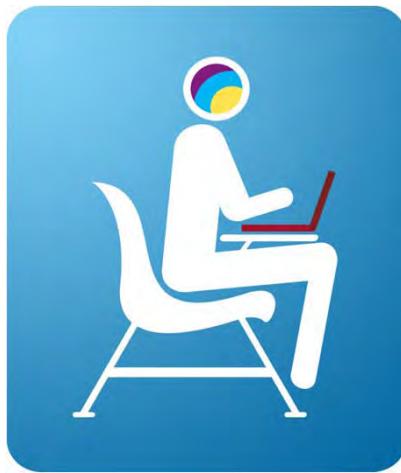
- They were previously deleted from the state family of the form.
- They were added to the state family of the business object after the form was created.
- The form was created with one business object, and then copied onto a different business object with a different state transition family. Only those states that are in the state family of both business objects are kept in the state family of the form.

You can add the missing transitions to the state family of the form by clicking the **Find** action on the **State Transition** tab. When you click the **Find** action, a window opens. It lists the transitions in the state family of the underlying business object that are not present in the state family of the form.

You select the check box for each transition that you want to add to the state family of the form, and then click the OK action. If you add transitions that connect to a state that is not in the form's state family, that state is added to the form's state family.

If you change the state transition family of a form, click the **Save** action to save the changes. When you are done with the State Transition tab, return to the Layout tab.

## Student exercises



© Copyright IBM Corporation 2013

18

### *Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

---

1. True or False: A form can be based on multiple BOs
2. Name the layers of a form
3. How are fields laid out on a form?
4. What four properties are used for positioning fields?
5. How do you add new states and transitions to a form?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. True or False: A Form can be based on multiple BOs  
*False. A BO can have multiple forms A form has one BO*
2. Name the layers of a form  
*Module, BO, form, tab, section, field*
3. How are fields laid out on a form?  
*By default, in two columns for an unlimited number of rows  
The columns can range from 1 to 12*
4. What four properties are used for positioning fields?  
*Start Row, Row Span, Start Column, Col Span*
5. How do you add new States and Transitions to a form?  
*Use the Find action to bring in states and transitions that are in the BO but not included in the form. It is not possible to define entirely new states and transitions in the form*



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Use the Form Builder tool
- Create a form that contains tabs, sections, and fields
- View and modify the state family of a form



# 6 Navigation



# 6 Navigation



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

<b>What this unit is about</b>	This unit is an introduction to navigation in IBM TRIRIGA and to the Navigation Builder tool.
<b>How you check your progress</b>	You can check your progress in the following ways: <ul style="list-style-type: none"><li>• Review questions</li><li>• Lab exercises</li></ul>
<b>References</b>	<i>Application Building for the IBM TRIRIGA Application Platform</i> <i>User Experience User Guide</i>

---

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Describe navigation in IBM TRIRIGA
- Build a navigation collection
- Add navigation items to a navigation collection

# Lesson 1. Navigating the user interface



## Lesson 1: Configuring the navigation of the user interface



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is an introduction to navigation in IBM TRIRIGA and to the Navigation Builder tool.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

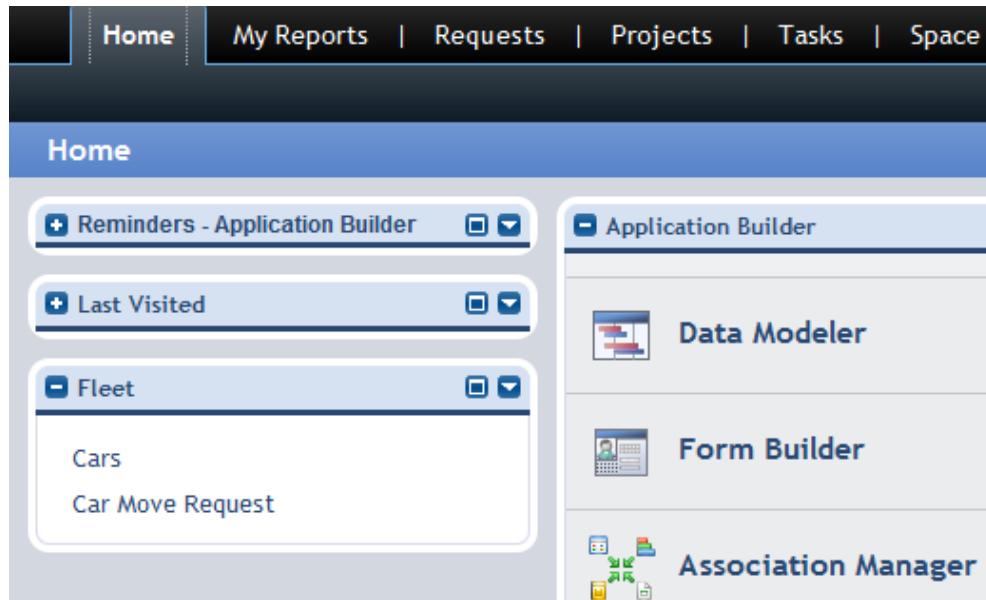
- Describe navigation in IBM TRIRIGA
- Build a navigation collection
- Add navigation items to a navigation collection

### References

*Application Building for the IBM TRIRIGA Application Platform*  
*User Experience User Guide*

## Navigation items and navigation collections

- Navigation items control movement in IBM TRIRIGA
- Navigation items are contained in navigation collections



© Copyright IBM Corporation 2013

4

### Navigation items and navigation collections

You use navigation items to control your movement in IBM TRIRIGA. Navigation items are flexible and generic enough to represent menus and portal quick link sections, display hierarchical data, run reports, and link to other builder tools. You can also configure them to display a default **master / detail** query, which is known as a **manager** query. It provides a standard way for the records of a form to be displayed.

Sometimes, users that work in different roles need different views of records or different ways to manipulate the same records. Their needs can be served by creating navigation items that use a customized query to manage the same kinds of records but in different ways.

A **navigation collection** is a hierarchical structure that contains navigation items. You can use a navigation collection as a menu, quick link portal section, or related links. The navigation collection is one of the primary mechanisms that are used to access and manipulate records. You can associate a navigation collection with multiple users, and it can be accessed from many places.

The slide shows various ways that navigation items are used in IBM TRIRIGA, including in the menu, as quick links, and as links to builder tools.

## Navigation Builder tool

- Use this tool to create, maintain, and organize navigation items and navigation collections
- Click **Tools > Builder Tools > Navigation Builder**

The screenshot shows the Navigation Builder tool interface. At the top, there is a breadcrumb navigation: Home > Tools > Builder Tools > Navigation Builder. Below the breadcrumb are four buttons: Add, Edit, Copy, and Delete. A 'Name' input field is followed by a list of navigation items, each with a checkbox and a label. To the right, a 'Label' column lists the corresponding labels for each item. At the bottom, there is a copyright notice: © Copyright IBM Corporation 2013.

Name	Label
<input type="checkbox"/> triApplicationAdministrator - Quick Links - Home Application Ad	Application Administration
<input type="checkbox"/> triApplicationAdministrator - Quick Links - Home Application Bui	Application Builder
<input type="checkbox"/> triApplicationAdministrator - Quick Links - Home License and Se	License and Security
<input type="checkbox"/> triApplicationAdministrator - Quick Links - Home Reminders	Reminders
<input type="checkbox"/> triApplicationAdministrator - Quick Links - Home Utilities	Utilities
<input type="checkbox"/> triApplicationBuilder - Quick Links - Home Application Administrat	Application Administration
<input type="checkbox"/> triApplicationBuilder - Quick Links - Home Application Builder	Application Builder
<input type="checkbox"/> triApplicationBuilder - Quick Links - Home License And Security	License And Security
<input type="checkbox"/> triApplicationBuilder - Quick Links - Home Reminders	Reminders
<input type="checkbox"/> triApplicationBuilder - Quick Links - Home UTILITIES	

5

### Navigation Builder tool

The Navigation Builder is a tool that is provided in the IBM TRIRIGA Application Platform to create, maintain, and organize navigation items in menus or portals. You can use it to set up different ways to view your data.

Using the Navigation Builder tool, you work with three types of objects:

- **Navigation item:** An element of a navigation collection.
- **Navigation collection:** A hierarchical structure that contains navigation items. The navigation collection is one of the primary mechanisms that is used to access and manipulate records.
- **Navigation target:** A destination to which you can navigate, such as a form, query, scorecard, the Data Modeler tool, or an external web address. Navigation targets are identified in navigation items.

You access the tool by clicking **Tools > Builder Tools > Navigation Builder**. When Navigation Builder opens, it displays a list of navigation collections. Filters are available for searching through the collections.

Action buttons are also available; for example, Click **Add** to create a navigation collection. Select a collection and click **Edit**, **Copy**, or **Delete** to perform the named action on that collection. You must have a collection open (through Add or Edit) to access the Navigation Items Library.

## Navigation collections

- Hierarchical container of navigation items
- The type of collection determines its use

The screenshot shows the 'Navigation Collection Properties' configuration page. On the left, there's a form with fields for Name (triContracts - Quick Links - Leases Reminders), Label (Reminders), Description, Help, and Type (Quick Links). The 'Type' dropdown has 'Quick Links' selected. Below the form is a 'Navigation Items Library' section. On the right, there are two tabs: 'Navigation Items' (selected) and 'Group Overrides'. The 'Navigation Items' tab displays a list of navigation items grouped by category. The categories and their items are:

Name	Label
Menu Group - Catalogs	Catalogs
Hierarchy - Catalog	Catalog Hierarchy
Master Detail - Request Catalogs - Request Catalogs	
Master Detail - Product Catalogs	Product Catalogs
Menu Group - Cost Estimates	Cost Estimates
Menu Group - Maintenance	Maintenance
Master Detail - Service AssignmentService Assignments	
Record Add - Service Assignment Service Assignment Wizard	
Menu Group - Moves	Moves

**Navigation Collection Properties**

\* Name: triContracts - Quick Links - Leases Reminders  
 \* Label: Reminders  
 Description:  
 Help:  
 Type: Quick Links  
 Menu  
 Quick Links  
 Related Links

**Navigation Items**

Clear Selection Remove Edit

Name	Label
Menu Group - Catalogs	Catalogs
Hierarchy - Catalog	Catalog Hierarchy
Master Detail - Request Catalogs - Request Catalogs	
Master Detail - Product Catalogs	Product Catalogs
Menu Group - Cost Estimates	Cost Estimates
Menu Group - Maintenance	Maintenance
Master Detail - Service AssignmentService Assignments	
Record Add - Service Assignment Service Assignment Wizard	
Menu Group - Moves	Moves

**Navigation Items Library**

© Copyright IBM Corporation 2013

6

### Navigation collections

A navigation collection is a hierarchical structure that contains navigation items; it displays navigation items in a hierarchical manner. When you create or edit a navigation collection, you see the navigation collection configuration page. The navigation collection configuration page is divided into the following two halves:

- On the left is the Navigation Collection Properties window. There is also a heading for the Navigation Items Library.
- On the right is a **Navigation Items** tab. It displays the navigation items that are contained in the collection.

There is also a tab named Group Overrides. Group Overrides are related to security.

### Navigation Collection Properties

A navigation collection has the following properties:

- **Name:** The system name for the navigation collection. The name must be unique. The name can contain spaces and special characters. The convention is to use a three-part name for a navigation collection, separating them with dashes as follows:
  - The module that the collection is associated with
  - The type of collection (Menu, Quick Links, or Related Links)
  - A description of the collectionAn example of a name is triContracts - Quick Links - Leases Reminders.
- **Label:** The display label for the navigation collection.
- **Description:** An explanation of the navigation collection.
- **Help:** Instructional text.
- **Type:** The purpose of the collection, including the following values and meanings:
  - **Menu:** The navigation collection is intended for use as a primary menu. A collection of this type is limited to four hierarchy levels
  - **Quick Links:** The navigation collection contains links that are intended for use in a Quick Links portal section. This type of collection is limited to two hierarchy levels.
  - **Related Links:** The navigation collection contains links that are intended for use in any portal section type. The Related Links property in the portal section form associates a related links navigation collection to a portal section. This collection can have only one hierarchy level.

## **Navigation Items tab**

This tab shows the navigation items that are contained in the navigation collection and their hierarchical relationship to each other. Navigation items that have other items below them have a plus sign (+) to their left. Items that do not have other items below them have a minus sign (-).

Navigation items that are shown on the tab can be removed from the navigation collection by clicking the item and then clicking the **Remove** action. Items can be edited by selecting them and clicking the **Edit** action. Items can be dragged onto other items to add them to the hierarchy of that item.

## **Navigation Items Library**

To display existing navigation items, scroll to the bottom of the Navigation Collection Properties section and click **Navigation Items Library**. Clicking the section heading collapses the Navigation Collection Properties section and expands the Navigation Items Library section. You add navigation items to a navigation collection through the Navigation Items Library.

## Navigation Items Library

**Navigation Items Library**

Add to Collection   Add   Edit   Copy   Delete

Name	Label	Target Type	Target Name
<input type="text"/>	Builder		
<input checked="" type="checkbox"/> Builder - Form Builder	Form Builder	Builder	
<input checked="" type="checkbox"/> Builder - Navigation Builder	Navigation Builder	Builder	
<input checked="" type="checkbox"/> Builder - Portal Builder	Portal Builder	Builder	
<input checked="" type="checkbox"/> Builder - Workflow Builder	Workflow Builder	Builder	

© Copyright IBM Corporation 2013

7

### Navigation Items Library

Navigation items are the building blocks from which navigation collections are constructed. A navigation item can be referenced from many navigation collections. The Navigation Items Library displays a list of existing navigation items. The items are sorted by target type. You can use filters to search the Navigation Items Library list. Type a value in one or more of the filters and press Enter. Clear the filter value and press Enter to reset the list.

Create a navigation item by clicking the **Add** button. The new item is added to the library but not automatically added to the navigation collection. Select a navigation item by selecting the check box next to it, and click **Add to Collection** to add the navigation item to a navigation collection. Click **Edit** to edit the properties of a navigation item.

Be careful about using the **Delete** action. Using this action removes the selected navigation item from the Navigation Items Library itself, not just from the navigation collection. To remove an item from the navigation collection only, select the item on the right of the window and click **Remove**.



## Creating a navigation item

Save Save & Close Cancel

**Item Details**

\* Name :

\* Label :

Help Text:

Visible in Sitemap only

Dynamic Label

**Icon Details**

Upload Icon Clear Icon

Icon:

Icon Selector:  Browse...

**Target Details**

Target Type:

Where Used

### Target types

- Builder
- Call Center
- Document Management
- External Url
- Form - Record Add
- Form - Record View
- Master/Detail Default Query
- Master/Detail Hierarchy
- Master/Detail Query
- My Calendar
- My Profile
- My Timesheet
- Portal
- Portal Full
- Report
- Scorecard

### Creating a navigation item

Clicking the **Add** action from the Navigation Items Library opens the Navigation Items Editor window. You create a navigation item by setting the following key properties:

- **Name:** The system name for the navigation item. It must be unique. The name can include spaces and special characters.
- **Label:** The text that is displayed to the user for this item.
- **Target Type:** This property defines the type of target that the navigation item points to. There are many choices, including the following ones:
  - **Master/Detail Default Query:** The platform generates a query to display records that are created by using a specific business object.
  - **Master/Detail Query:** The results of a query are displayed in a Master/Detail format.
  - **Portal:** Access the portal.

When a value is selected for target type, the system expands the Target Details section to include properties appropriate for that target type.

## Sample Master/Detail Query navigation item

Target Details

Target Type: Master/Detail Query

Target Title: Fleet Management

Target Name

Module: cstFleetManagement

Business Object: cstCar

\* Report: cstCar - Master Detail Query

Form

Module: cstFleetManagement

Form: cstCar

Form Properties

Show Related Reports

Actions

Action	Action Label
<input checked="" type="radio"/> Add	Add
<input type="radio"/> Delete	Delete

© Copyright IBM Corporation 2013

9

### Sample Master/Detail Query navigation item

When you select Master/Detail Query for the target type, you can specify these additional properties for the navigation item:

- **Target Title:** Text that is displayed to the user.
- **Target Name section:** Identify the query to be displayed by selecting values for the Module, Business Object, and Report properties.
- **Form section:** Specify an optional form to be used for the query display by selecting values for the module and form.

If the query does not have actions, you can create them in the **Actions** section. Typically, an Add action (System Add) and a Delete action (System Delete) are created. With these actions, you can create and remove records that are displayed by the query. After you set the properties of the navigation item, click the **Save and Close** action to save the item to the Navigation Items Library.



## Adding a quick link to a portal

---

1. Create a navigation collection, if necessary
2. Create a navigation Item
3. Add it to the collection
4. Create a portal section, if necessary
5. Have the portal section point to the navigation collection
6. Add the portal section to a portal

### *Adding a quick link to a portal*

You can add a navigation item of type **Quick Link** to a portal by following these steps:

1. Create a navigation collection, if necessary. It must have a Type value of **Quick Links**.
2. Create a navigation item. Choose any of the **Target Type** values.
3. Add the navigation item to the navigation collection. You might have to search the library to find the item before you can select it and use the **Add To Collection** button.
4. Create a portal section, if necessary. It must have a type of **Quick Links**.
5. Use the picker for the Quick Links Collection property and select the navigation item that you created in step 2. Save the portal section.
6. Add the portal section to a portal.

## Student exercises



© Copyright IBM Corporation 2013

11

### *Student exercises*

Perform the exercises for this unit.



## Checkpoint questions

---

1. How many levels of quick links can you have in a navigation collection?
2. True or false: The name of a navigation item cannot contain blanks or special characters.
3. What is the primary difference between these target types?
  - a) Master/Detail Query
  - b) Master/Detail Default Query
4. Name the different type values for a navigation collection.
5. You have a portal that displays a quick link to courses. You want to add another link for course sections. How do you accomplish this task?

### *Checkpoint questions*

Enter your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. How many levels of quick links can you have in a navigation collection?  
*2*
2. True or false: The name of a navigation item cannot contain blanks or special characters.  
*False*
3. What is the primary difference between these target types?
  - a) Master/Detail Query
  - b) Master/Detail Default Query*The platform creates a query for b. You must create the query for a.*
4. Name the different type values for a navigation collection.  
*Menu, Quick Links, Related Links*
5. You have a portal that displays a quick link to courses. You want to add a link for course sections. How do you accomplish this task?  
*Create a navigation item that points to course sections. Add it to the same navigation collection as courses.*



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

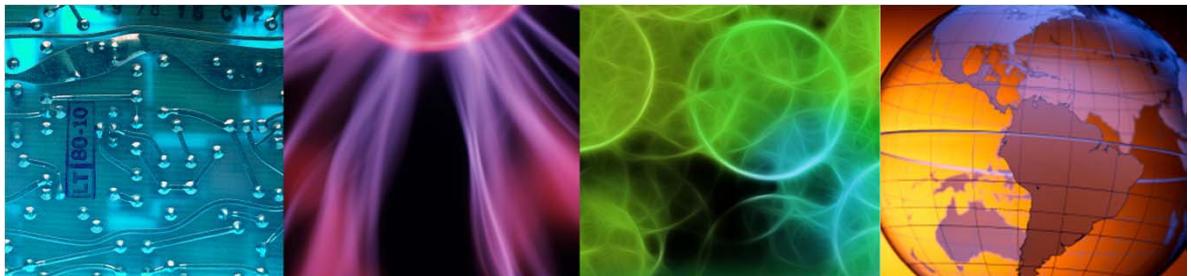
- Describe navigation in IBM TRIRIGA
- Build a navigation collection
- Add navigation items to a navigation collection



## 7 Portals



## 7 Portals



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

This unit is an introduction to portals and the Portal Builder in IBM TRIRIGA.

**How you check your progress**

You can check your progress in the following ways:

- Review questions
- Lab exercises

**References**

*Application Building for the IBM TRIRIGA Application Platform*

*User Experience User Guide*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Describe portals in TRIRIGA
- Add a navigation collection to a portal section
- Add a portal section to a portal

# Lesson 1. Portals and Portal Builder



## Lesson 1: Portals and Portal Builder



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is an introduction to portals and the Portal Builder in IBM TRIRIGA.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

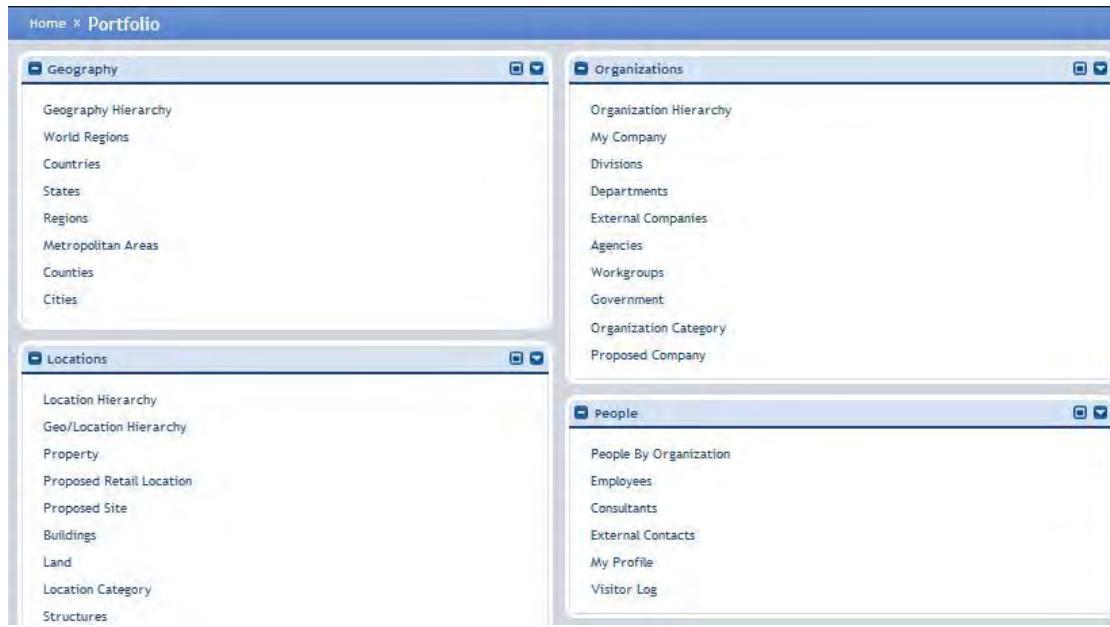
- Describe ports in IBM TRIRIGA
- Add a navigation collection to a portal section
- Add a portal section to a portal

### References

*Application Building for the IBM TRIRIGA Application Platform*  
*User Experience User Guide*

## Portals and portal sections

- The portal is the central organizing element of an application
- Portal sections are the building blocks of portals



© Copyright IBM Corporation 2013

4

### Portals and portal sections

The IBM TRIRIGA **portal** is the central organizing element of the application user interface. You can access all functions in the product through this interface. A portal is the home page for an application. A portal that is the target of a menu item (other than Home) is also called a **landing page**.

When someone signs in to the IBM TRIRIGA Application Platform, they see their home portal. A user can return to their home portal at any time by clicking the **Home** menu item. The home portal and the menu for each user is set in the My Profile record that defines their user ID. For many people, this is set on the Profile tab of their Employee record.

**Portal sections** are the building blocks from which portals are constructed. Portal sections are displayed as encapsulated areas within a portal. IBM TRIRIGA ships with hundreds of portal sections. These standard portal sections include system components that provide data and navigation help, and also action items, notifications, queries, adding a record, and viewing a record. The slide shows a portal that contains four quick links portal sections. You create portals and portal sections by using the Portal Builder.

## Portal Builder tool

- Use this tool to create and manipulate portals and portal sections
- Click **Tools > Builder Tools > Portal Builder**

The screenshot shows a web-based application titled "Portal Builder". At the top, there is a blue header bar with the title "Home > Tools > Builder Tools > Portal Builder". Below the header are four action buttons: "Add", "Edit", "Copy", and "Delete". A search bar labeled "Name" is positioned below these buttons. The main content area displays a list of portal entries, each with a checkbox and a name. The names listed are: Contracts - Agreements Landing Page, Contracts - Contracts Landing Page - Contract Manager, Contracts - Contracts Landing Page - Default, Contracts - Contracts Landing Page - Real Estate Contract Manager, Contracts - Leases Landing Page, Contracts - Requests Landing Page, Global - Action Items And Notifications Landing Page, and Global - Application Setup Landing Page. The "Contracts - Contracts Landing Page - Contract Manager" entry is highlighted with a light gray background.

Name
<input type="checkbox"/> Contracts - Agreements Landing Page
<input type="checkbox"/> Contracts - Contracts Landing Page - Contract Manager
<input type="checkbox"/> Contracts - Contracts Landing Page - Default
<input type="checkbox"/> Contracts - Contracts Landing Page - Real Estate Contract Manager
<input type="checkbox"/> Contracts - Leases Landing Page
<input type="checkbox"/> Contracts - Requests Landing Page
<input type="checkbox"/> Global - Action Items And Notifications Landing Page
<input type="checkbox"/> Global - Application Setup Landing Page

© Copyright IBM Corporation 2013

5

### Portal Builder tool

You use the Portal Builder to create and manage portals and to control the creation and arrangement of portal sections within each portal. You access the Portal Builder by clicking **Tools > Builder Tools > Portal Builder**. When the tool opens, you see a list of available portals and a set of action buttons. There is also a filter for the Name field that you can use to search for a specific portal.

Click the **Add** action to create a portal, or select an existing portal and click the **Edit**, **Copy**, or **Delete** actions to perform that function on the portal. You must open a portal by using the Edit or Add actions to reach the Portal Sections Library.



## Portal Properties panel

**Portal Properties**

\* Name: Contracts - Agreements Landing Page

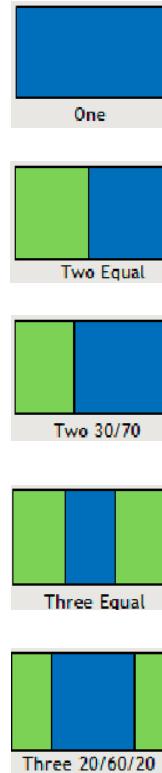
\* Label: Agreements

Description:

Layout: Two 30/70

One  
Two Equal  
**Two 30/70**  
Three Equal  
Three 20/60/20

**Portal Sections Library**



© Copyright IBM Corporation 2013

6

### Portal Properties panel

In the Portal Properties panel, you set the properties of a portal and access the Portal Sections Library. The Portal Sections Library is where you manage the portal sections. The name of a portal is used to identify the portal in the Portal Builder. The name can contain blanks and special characters, and it must be unique. The label of a portal is shown to the user. The label can contain blanks and special characters. Make it something meaningful to the users.

Portals are laid out in columns. You use the **Layout** property to specify how many columns the portal contains and their relative size. The choices for the Layout property are as follows:

- **One:** The portal has one column.
- **Two Equal:** The portal has two vertical columns of equal size.
- **Two 30/70:** The portal has two vertical columns. The left column occupies 30% of the portal. The right column occupies 70%. Large columns are useful for displaying reports and graphical queries without having to squeeze them into a small space.
- **Three Equal:** The portal has three vertical columns of equal size.

- **Three 20/60/20:** The portal has three vertical columns. The left and right columns each occupy 20% of the portal. The middle column occupies 60% of the portal.

Portal sections belong to one column on a portal and cannot span multiple columns. At the bottom of the Portal Properties panel is the Portal Sections Library section. When you click the Portal Sections Library section to expand it, the Portal Properties panel section collapses.

## Portal Layout panel

- View layout of portal sections in the portal
- Arrange sections by using drag and drop

The screenshot shows the 'Portal Layout' tab selected in the top navigation bar. Below it, there are two main sections:

- Reminders - Contract Agreements**:
  - Name:** Reminders - Contracts Agreements Landing Page
  - Type:** Links
  - Description:** (empty)
  - Section Configuration:**
    - Collapsed By Default
- My Owned Property (Fee) Agreements**:
  - Name:** Query - triOwnedProperty - Owned Agreements
  - Type:** Tabular
  - Description:** (empty)
  - Section Configuration:**
    - Collapsed By Default

© Copyright IBM Corporation 2013

7

### Portal Layout panel

The Portal Layout panel is to the right of the Portal Properties panel. It shows the layout of the portal sections within the portal. You can use this panel to remove sections from the portal or rearrange portal sections within the portal. You move portal sections by dragging them. To remove a portal section from a portal, you click the X in the upper right of the portal section. You add portal sections to the portal from the Portal Sections Library.

Each portal section has a property that is called **Collapsed by Default**. Using this property means that a portal section is collapsed when a portal opens and you see only the heading. You must expand the portal section to see the contents. The default value of this property is cleared, meaning that the sections are expanded by default. On the **Related Portal Sections** tab, a user can add to their portal through the personalization process.

## Portal Sections Library

- View a list of portal sections, with filters for searching
- Use buttons to create, edit, delete, and add to portal

The screenshot shows a web-based application titled "Portal Sections Library". At the top, there are three buttons: "New", "Edit", and "Delete". To the right of these is a blue button labeled "Add To Portal". Below this is a table with three columns: "Name", "Type", and "Header". The table contains four rows of data, each with a checkbox and a link to a specific portal section. The footer of the page includes navigation arrows and the text "1-50 of 559".

Name	Type	Header
<input type="checkbox"/> Application Administration - Application Administrator Portal	Quick Links	Application Administration
<input type="checkbox"/> Application Administration - Application Builder Portal	Quick Links	Application Administration
<input type="checkbox"/> Application Administration - Tools Landing Page	Quick Links	Tools - Application Administration
<input type="checkbox"/> Application Builder - Application Administrator Portal	Quick Links	Application Builder

© Copyright IBM Corporation 2013

8

### Portal Sections Library

The Portal Sections Library is the part of the IBM TRIRIGA Application Platform that you use to create, edit, and manage portal sections. It lists the available portal sections and includes filters for searching. You access the Portal Sections Library from the Portal Builder by clicking the Portal Sections Library section heading.

You click **New** to create a portal section. To add a portal section to a portal, select a portal section and click the **Add to Portal** button. The portal section is added at the bottom of the portal that is being edited. You can click the **Edit** or **Delete** button to perform that function; however, when you click the Delete button you remove a portal section from the entire portal sections library, not only from a portal. To remove a portal section from a portal, you use the Portal Layout panel.

**Creating a portal section**

**Portal Section**

Name: Fleet - Fleet Management Portal Section

Header: Fleet

Tooltip: This is a tooltip

Description: This is used for the cstFleetManagement module's records, Course and Course Section.

Related Links: cstFleetManagement - Related Links - Fleet Related Links

Type: Quick Links

Quick Links Collection: triTools - Quick Links - Application Builder

**Types**

Report (selected)

Report

External

Scorecard

Quick Links

Form - Record Add

Form - Record View

### Creating a portal section

Clicking **New** in the Portal Sections Library opens the Portal Section properties window. Specify the properties of the portal section and click **OK** to save the properties and close the window. Click **Apply** to save changes without closing the window.

The key properties of a portal section are as follows:

- **Name:** Identifies the portal section in the Portal Section Library.
- **Header:** Displays in the section bar of the portal section.
- **Tooltip:** Displays if the user positions the cursor over the header.
- **Related Links:** Names a navigation collection of type related links. Related links are links to other items that the user of this portal section might be interested in.
- **Type:** Determines the contents of the portal section, which includes the following options:
  - **Report:** Contains the results of any IBM TRIRIGA report
  - **External:** Shows the contents of a specific URL
  - **Scorecard:** Displays a scorecard

- **Quick Links:** Shows a navigation collection of type Quick Links
- **Form - Record Add:** Displays a chosen form for creating a record
- **Form - Record View:** Displays an existing record through its form

When a Type value is selected, other properties that are related to that selection become visible. For example, if Report is selected, you must select a query for the Query property. If Quick Links is selected, the Quick Links Collection (a navigation collection) must be selected. Also, you must choose whether icons are displayed next to the links.



## Adding a quick link to a portal

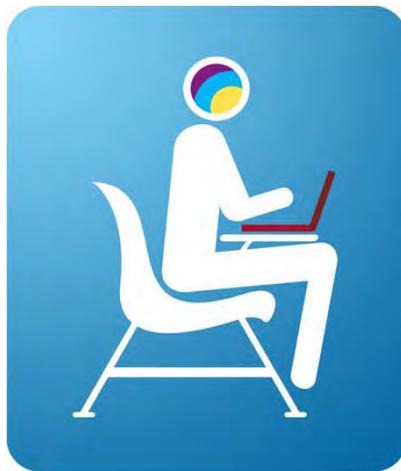
1. Create a navigation collection, if necessary
2. Create a navigation item
3. Add it to the collection
4. Create a portal section, if necessary
5. Have the portal section point to the navigation collection
6. Add the portal section to a portal

### *Adding a quick link to a portal*

You can add a navigation item of type Quick Link to a portal by following these steps:

1. Create a navigation collection, if necessary. It must have a **Type** value of **Quick Links**.
2. Create a navigation item. Choose any of the target type values.
3. Add the navigation item to the collection. It might be necessary to search the library to find the item before you can select it and use the **Add To Collection** button.
4. Create a portal section, if necessary. It must have a **Type** of **Quick Links**.
5. Have the portal section point to the navigation collection.
6. Add the portal section to a portal.

## Student exercises



© Copyright IBM Corporation 2013

11

### *Student exercises*

Perform the exercises for this unit.



## **Checkpoint questions**

1. What is a portal section?
2. What is the maximum number of columns that you can use for a portal layout?
3. How do you rearrange portal sections on a portal?
4. How many different portals can a portal section be in?
5. How do you remove a portal section from a portal?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. What is a portal section?  
*The building block of a portal.*
2. What is the maximum number of columns that you can use for a portal layout? **3**
3. How do you rearrange portal sections on a portal?  
*Drag and drop.*
4. How many different portals can a portal section be in?  
*There is no limit.*
5. How do you remove a portal section from a portal?  
*Open the portal, and in the Portal Layout panel, click the X at the upper right of the portal section.*



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Describe portals in TRIRIGA
- Add a navigation collection to a portal section
- Add a portal section to a portal



## 8 Basic reporting



## 8 Basic reporting



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

This unit is an introduction to basic queries in IBM TRIRIGA.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

User Experience User Guide



## **Objectives**

After completing this unit, you should be able to perform the following tasks:

- Open the Report Builder tool
- Define a report or query
- Filter the results of a report or query

# Lesson 1. Creating reports



## Lesson 1: Creating reports



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is an introduction to basic queries in IBM TRIRIGA.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Open the Report Builder tool
- Define a report or query
- Filter the results of a report or query

### References

User Experience User Guide

## Reports in IBM TRIRIGA

- The IBM TRIRIGA platform has two ways to generate reports:
  - By using a built-in reporting tool called the **Report Manager**
  - By using the advanced reporting feature: Business Intelligence and Reporting Tools (**BIRT**)
- You can access Report Manager in two ways:
  - Click **Tools > Builder Tools > Report Manager**
  - Click **My Reports** tab



© Copyright IBM Corporation 2013

4

### Reports in IBM TRIRIGA

The IBM TRIRIGA reporting solution is called the **Report Manager**. You use the Report Manager to create simple tabular reports, queries, and graphs that combine data from multiple records into a single presentation. These presentations of data can be displayed as reports, as portal components, or in navigation items. They are also used in multiple places throughout the TRIRIGA applications, including query sections, workflows, filters for other queries, and extended formulas. IBM TRIRIGA reports include filter, search, and sort capabilities and also the ability to link to other reports.

You can access the Report Manager in the following two ways:

- Clicking **Tools > Builder Tools > Report Manager**
- Clicking the **My Reports** tab

If you need a report presentation that is beyond the capabilities of the IBM TRIRIGA Report Manager, you can use the IBM TRIRIGA advanced reporting feature. IBM TRIRIGA uses Eclipse Business Intelligence Reporting Tools (BIRT) as the enabling technology for the advanced reporting feature. The advanced reporting feature is an advanced topic that is not covered in this course.

## Report Manager

The screenshot shows the Report Manager interface. At the top, there's a blue header bar with 'Home > My Reports'. Below it is a navigation bar with four tabs: 'My Reports' (selected), 'Community', 'System Reports', and 'Administration'. Underneath the navigation bar, there's a search/filter section with a magnifying glass icon, '10 total found', 'Apply Filters', and 'Clear Filters'. The main area is a table listing reports. The columns are 'Title' (with a checkbox and a 'Contains' dropdown), 'Name' (with a checkbox), and two more columns that are partially visible. The reports listed are:

Title	Name
Car Move Requests Requested by current user	cstCarMoveRequest - Report - Car Move Requests Requested by current user
Car for current move request	cstCar - for current move request
Cars	cstCar - Master Detail Query
Cars - Draft	cstCar - triDraft - Editable
Cars - Retired	cstCar - triRetired - Editable
Cars I supervise	cstCar - Report - Cars I supervise

At the bottom of the table are buttons for 'New', 'Copy', 'Delete', 'Copy as My Report' (highlighted in blue), and 'Share as Community'.

### Report Manager

When you open the Report Manager, you can see all of the reports in the platform divided across the following four tabs:

- **My Reports:** The My Reports tab has local reports. These reports are typically used only by the person who created the report. The user is not required to be an administrator because these reports do not affect the core TRIRIGA applications. Community or system reports can be copied to My Reports by using the **Copy as My Report** action. Reports that are created on the My Reports tab cannot be used in applications.
- **Community:** On the Community tab are listed the reports that were selected by the developer to be shared with the community. Users with access to this tab can run these reports but cannot modify them. Community reports can be copied into the My Report page, where the copy can be modified as needed. Community reports are indicated by a C in the column named **C**.
- **System Reports:** The System Reports tab is where the administrators and developers can create reports that can be used by applications. These reports can be shared as community reports and can be created and edited only by administrators and developers.

- **Administration:** The Administration tab contains all the reports from every My Reports tab in the system. From here the administrators can change the owner of a report, copy reports, share reports with groups, and delete reports. Access to this tab is limited.



**Note:** The Report Manager uses a feature that is called Prompt before Query. This delays the execution of the query until after the user enters filter values. So when any of the tabs are viewed for the first time, nothing is listed. The user must click either Apply Filters or Clear Filters to see the list of reports.

Other features of the Report Manager:

- **Filters:** The Report Manager provides filters to allow searching for reports. Enter a value in one or more of the filters and click Apply Filters to limit the display to those reports that match your criteria. Click Clear Filters to remove those limits.
- **Actions:** The My Reports tab has actions to create, copy, and delete reports. An authorized person also sees actions to copy a report to the community tab, and to share a report with others in a group. The Community tab has an action to copy a report to the My Reports tab. The user can make a local copy of a report and modify it to suit their needs. The System Reports tab has actions to create, copy, and delete reports. It also has an action to copy a report to the My Reports tab, and another action to copy a report to the Community tab.
- **Running reports:** To the left of each report is an icon that looks like a piece of paper with a play symbol. Click this icon to run the report. Or you can open the report and use the Run Report action button to run the report.
- **Editing reports:** Every value that is displayed for a report, such as Name and Title, is a link to the report. Click the value to open the report in the Report Builder tool.

## Report Builder, General tab

Step 1 of 6 (Required):

**General**

Name: cstCar - Master Detail Query | ID: CUSTOM

Header (Title): Cars | Tag:

Description:

Type: Query | Data Scope: All Projects

Created By: Iverson, Don - 1000000

Show As Community Report

Business Objects Options Related Reports Security

Business Object: Add Business Object | Edit | Remove

	Module	Business Object	Form	Association Type
<input checked="" type="radio"/>	-cstFleetManagement	Car	Car	-

© Copyright IBM Corporation 2013

### Types

- Report
- Query (selected)
- Graphic
- Summary
- Chart
- External
- Reserve Legacy
- Reserve
- Hierarchy
- Metric

### Scopes

- Active Project
- Company
- All Projects (selected)

Report Builder, General tab

The tool for building reports is the **Report Builder**. Open it by clicking the New action in the Report Manager, or by clicking a link to a report.

The Report Builder is organized into six tabs:

- **General:** Define the properties of the report, including the Name, Type, and Business Object. This tab is required.
- **Columns:** Select the fields to be used in the report. This tab is required.
- **Order & Group:** Define the sorting and grouping of the results. This tab is optional.
- **Filters:** Define filters to limit results in the report, and filters to help users sift through the results. This tab is optional.
- **Advanced:** Create association filters and action buttons for the report. This tab is optional.
- **Where Used:** View a list of places where this report is used. This tab is optional.

You can set the following properties on the **General** tab for a basic report:

**Name:** Specify the name of the report. Names must be unique, and can contain spaces and special characters.

**ID:** Use CUSTOM for a custom report.

**Header:** The text that shows at the top of the report.

**Type:** Select the type of report from these choices:

- **Query:** A basic tabular report.
- **Report:** A basic tabular report.
- **Graphics:** Shows any location information added to a graphic image.
- **Chart:** Pie, bar, line, or stacked bar chart.
- **Summary:** Like a report, but does not display details.
- **External:** Provides a link to external reporting tools. These reports show up when the Reports icon is accessed in a manager.
- **Hierarchy:** Used to display data in hierarchical format.
- **Metric:** Used for WPM metric queries.
- **Reserve, Reserve Legacy:** These types are used with Reserve applications.

Things to consider when you are selecting the type:

- In the past, Reports and Queries were different from each other but are now essentially the same. The type to use is now a matter of convention:
  - A Report is ready for the user to run directly, and is accessed from Report Manager.
  - A Query is hooked into the system somehow, such as the target of navigation, or used in a form, or a formula. It is not accessed through Report Manager.
- When you develop reports, a common mistake is to be working from the My Reports tab of the Report Manager instead of the System Reports tab. The Query report type is not available when you work from the My Reports tab. If you do not see Query in the list of report types, then you are on the wrong tab.

**Data Scope:** This limits the records that are available for the report. Select the scope of the data from these choices:

- **Active Project:** The data available to the report is restricted to the active project.
- **Company:** Only data that exists outside a project is accessible to the report.
- **All projects:** The report includes all data, without regard to project or company.

**Business Objects:** The Business Object subtab identifies the data available to this report. Use the Add Business Object link to add business objects to the report. Most reports use only a single business object. Reports that use multiple business objects are a more advanced topic and are covered separately.

## Report Builder, Columns tab

Select a Business Object to show associated columns

**Business Object:**

Module	Business Object	Form	Association Type
<input checked="" type="radio"/> cstFleetManagement	Car	Car	<input type="radio"/>

Select column(s) to display on report

**Columns:**

General (General)	
Field Label	Field Name
<input checked="" type="checkbox"/> !	triUserMessageFlagTX
<input checked="" type="checkbox"/> % KBB	cstPercentKbbNU
<input type="checkbox"/> Attention!	triUserMessageTX
<input type="checkbox"/> Business Object Label	triBusinessObjectLabelSY
<input type="checkbox"/> Business Object Name	triBusinessObjectNameSY
<input checked="" type="checkbox"/> Car Category	cstCarCategoryCL

**Display Columns:**

Field	Report Label	Width
<input checked="" type="radio"/> ! (triUserMessageFlagTX)	!	<input type="text"/>
<input type="radio"/> GeographyName (GeographyName)	GeographyName	<input type="text"/>
<input type="radio"/> OrgName (OrgName)	OrgName	<input type="text"/>
<input type="radio"/> Image (trImageIM)	Image	<input type="text"/>
<input type="radio"/> Name (triNameTX)	Vehicle	<input type="text"/>
<input type="radio"/> VIN (cstVinTX)	VIN	<input type="text"/>
<input type="radio"/> Car Category (cstCarCategoryCL)	Car Category	<input type="text"/>
<input type="radio"/> % KBB (cstPercentKbbNU)	% KBB	<input type="text"/>
<input type="radio"/> KBB Value	IKRA Value	<input type="text"/>

© Copyright IBM Corporation 2013

7

### Report Builder, Columns tab

The Columns tab defines which fields to use in the report. The information in the Columns tab is organized into three sections: Business Object; Columns; and Display Columns.

- **Business Object:** The Business Object section, in the upper left of the screen, indicates the source business objects for the query. The fields of the selected business object are listed in the Columns section. Select a different business object to change the list of fields.
- **Columns:** The Columns section, in the lower left of the screen, identifies which fields in the selected business object are included in the report. The name and label of all fields in the business object are grouped by section and listed, with a check box next to each for selection. As fields are selected in the Columns section, they are added to the Display Columns list, below the field that is selected in that section. By default, the list is sorted by the Field Label. Click a column heading to sort by that column.
- **Display Columns:** The Display Columns section, on the right side of the screen, specifies the order in which the selected fields are presented. It also defines the labels that are used to identify them.

The fields in a report are displayed from left to right in an order that matches their position in the Display Columns list, from top to bottom. Use these actions in the Display Columns section to change the order of the fields: Move up; Move down; Move to top; Move to bottom. Fields can be removed from the report by selecting them and using the Remove action. The label that is shown on the report for a field can be changed by modifying the Report Label property for the field. Use the Width column to specify a percentage of the display space to use for a field.

Properties on the General tab and the Columns tab are required. After you define these properties, you can use the **Run Report** action to run the report and display the results.



**Note:** This action runs the last saved copy of the report. Use the **Save** action before you use Run Report.

## Report Builder, Order & Group tab

The screenshot shows the 'Order & Group' tab in Report Builder. It has two main sections: 'Group By' and 'Order By'.  
**Group By:** On the left, there is a list of fields: !, GeographyName, OrgName, Image, Vehicle, VIN, Car Category, % KBB, KBB Value, and Current Mileage. On the right, there is an empty rectangular area with up and down arrows for reordering.  
**Order By:** On the left, there is a list of fields: GeographyName, OrgName, Image, VIN, Car Category, % KBB, KBB Value, and Current Mileage. On the right, there is a list with one item: ! (ASC) Vehicle (ASC). An up arrow is positioned above the list on the right.

© Copyright IBM Corporation 2013

8

Report Builder, Order & Group tab

The **Order & Group** tab is where you define the grouping of values on the report, and also the way that the results are sorted. There are two sections on this tab: Group By and Order By. Each section has a list on the left and a list on the right. The lists on the left contain the fields that are selected on the Columns tab. The lists on the right are the fields that are selected for this property and the order that they are in.

Select a field from a list on the left and click a right-facing arrow to move the field to the list on the right. Use the up and down arrows to change the order of the fields, if needed. Select a field on the right and click the arrow that faces left to remove the field from the list on the right.

- **Group By:** The Group By section defines the grouping of the results. When the report is run, the results are grouped by the fields on the right, in order from the top to the bottom.
- **Order By:** Order By defines the sorting that is done on the results. When the report is run, the results are sorted by the fields on the right, in order from the top to the bottom. When you move fields to this list, you can choose to sort the field in ascending or descending order.

## Report Builder, Filters tab

The screenshot shows the Report Builder interface with the 'Filters' tab selected. It displays three sections: 'Columns', 'User Filter Columns', and 'System Filter Columns'.  
**Columns:** A table titled 'General (General)' showing fields like triUserMessageFlagTX, cstPercentKbbNU, triUserMessageTX, and triBusinessObjectLabelSY.  
**User Filter Columns:** A table with columns: Join Operator, Field, Report Label, Filter Operator, and Value. It contains four rows:

- Join Operator: Radio button (selected), Field: % KBB (cstPercentKbbNU), Report Label: % KBB, Filter Operator: Less Than or Equals, Value: User Input.
- Join Operator: Radio button (unchecked), Field: % KBB (cstPercentKbbNU), Report Label: % KBB, Filter Operator: More Than or Equals, Value: User Input.
- Join Operator: Radio button (unchecked), Field: Make (cstMakeLI), Report Label: Make, Filter Operator: Contains, Value: User Input.
- Join Operator: Radio button (unchecked), Field: Model (cstModelLI), Report Label: Model, Filter Operator: Contains, Value: User Input.

  
**System Filter Columns:** A table with columns: Join Operator, Field, Report Label, Filter Operator, and Value. It contains one row:

- Join Operator: Radio button (selected), Field: Status (triStatusCL), Report Label: Status, Filter Operator: Not In, Value: 'Retired', 'Upload Error'.

At the bottom right of the interface is a copyright notice: © Copyright IBM Corporation 2013.

9

### Report Builder, Filters tab

The Filters tab uses fields, operators, and values to focus the results to display. There are two types of filters:

- **System filters:** These filters limit the results that are returned by the query. Only records with fields that contain values that meet the specified criteria are included. Records that are excluded by system filters are not available to the user. System filters are not visible to the user.
- **User filters:** Also called runtime filters, these filters do not affect the query results. These filters help the user sift through the records that are returned by the query. Records that are excluded by user filters can be restored to the result list by clearing the filters.

User filters are seen above the fields that they apply to, provided that those fields are displayed in the report. User filters that apply to fields that are not displayed in the report require the Additional Filters action to be used to see them.



**Note:** User filters are not displayed when less than two records are shown in the query results.

The information in the Filters tab is organized into four sections:

- **Business Object:** section on the upper left indicates the source business objects. All fields from the selected business object are listed in the Columns section.
- **Columns:** section on the lower left identifies which fields in the selected business object to use as filters. There are two columns of check boxes, one for User and one for System. Checking the box in the User column for a field adds it to the User Filter Columns section. Checking the box in the System column adds the field to the System Filter Columns section. It is possible to have both a user and a system filter on the same field.
- **User Filter Columns:** section on the upper right lists the user filters and defines their label and their filter operator. Each field type has its own set of filter operators. For text fields, the most useful operator is Contains.

All user filters are joined by the AND operation, meaning that all of the conditions must be met or the record is excluded from the results.

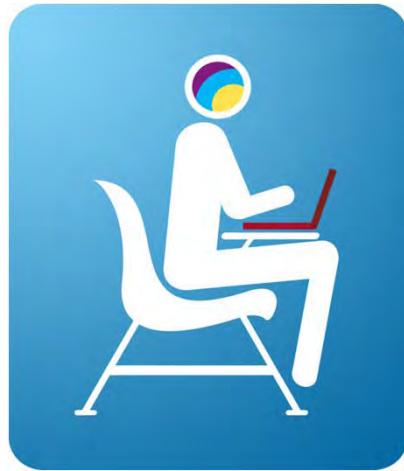
- **System Filter Columns:** section on the lower right lists the system filters and their properties. System filters allow more conditions than user filters. They can also be grouped and nested using parentheses, and have a larger range of filter operators than user filters.

Strings that are specified in the Value field must be enclosed in **single** quotation marks. System filters are used sequentially, from top to bottom. Plan the sequence ahead of time for better performance. Use the Move Up, Move Down, Move to Top, and Move to Bottom to change the order of the filters. Put the filter that excludes the most records first. Put the filter that excludes the next most records second. And so on. Reports run faster with fewer records. To apply multiple filters of the same type to a field, create a filter on that field and then make a copy of the filter.

---

## Student exercises

---



---

© Copyright IBM Corporation 2013

10

### *Student exercises*

Perform the exercises for this unit.

## Checkpoint questions

1. What reports are on the My Reports tab of the Report Manager?
2. What is the purpose of a system filter?
3. What is the purpose of a user filter?
4. How can you put more than one filter of the same type on a field?
5. Which report type is not available on the My Reports tab?

## Checkpoint answers

1. What reports appear on the My Reports tab of the Report Manager? *Local copies of reports*
2. What is the purpose of a system filter?  
*To reduce the records returned by the query*
3. What is the purpose of a user filter?  
*To help the user search through the records*
4. How would you put more than one filter of the same type on a field? *Create a filter, then copy it*
5. Which report type is not available on the My Reports tab?  
*Query*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Open the Report Builder tool
- Define a report or query
- Filter the results of a report or query





## 9 Lists and classifications

---



## 9 Lists and classifications



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

### What this unit is about

Lists and classifications are a means of ensuring data integrity by providing a set of choices to the user. This unit covers the different types of lists in IBM TRIRIGA, how to create them, and how to use them. It also covers classifications and how to use them.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

Application Building for the IBM TRIRIGA Application Platform

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Create, view, and edit lists
- View and edit classifications

# Lesson 1. Lists



## Lesson 1: Lists



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Lists are a means of ensuring data integrity by providing a set of choices to the user. This lesson covers the different types of lists in IBM TRIRIGA, how to create them, and how to use them.

### What you should be able to do

After completing this lesson, you should be able to create, view, and edit lists.

### References

Application Building for the IBM TRIRIGA Application Platform



## Introduction to lists

---

- Lists are a defined set of choices
- Types of lists:
  - **Manual:** Basic manual entry list
  - **Dependent:** Contain values that depend on master list value
  - **Dynamic:** Contain values that come from records
- Manage lists in the List Manager
  - Click **Tools > Administration > Lists**

© Copyright IBM Corporation 2013

4

---

### *Introduction to lists*

The purpose of a list is data integrity. With a list, you get a defined set of choices to minimize the possibility of errors. When valid choices are the only ones that are available, the field always contains valid data.

There are three kinds of lists in IBM TRIRIGA:

- **Manual:** Lists are manually maintained by using the List Manager tool.
- **Dependent:** Lists offer a set of values that is dependent on a value that is selected in a master list. For example, if the master list is a list of automobile manufacturers, the dependent list shows only those models for the selected manufacturer.
- **Dynamic:** Lists have values from a specified field in all of the records that are created from a specified business object. For example, a list of user names can be based on the Full Name field of My Profile. The list automatically adjusts as new records are created.

You use the **List Manager** to create new lists and to change the contents of a list. Open the List Manager by clicking **Tools > Administration > Lists**.

# List Manager

[Sort List](#) | [Save Sequence](#) | [New List](#) | [View/Edit List](#) | [Delete List](#) | [Label List](#)

The screenshot shows the List Manager interface. On the left, there's a 'Manage By' section with a dropdown menu for 'Name' containing various system lists like 'Affinity', 'Allocation Type', etc. Below it is a 'Filter By' section with a dropdown menu for 'Select' containing modules like 'triInspection', 'triIssue', etc. On the right, there's a table titled 'Save Entries' with columns for 'Value', 'Select', 'Order', 'Sequence', and 'Value'. The table lists 14 entries with values such as 'About', 'Authorized Locations For', 'Category For', etc.

Value	Select	Order	Sequence	Value
1	About			About
2	Authorized Locations For			Authorized Locations For
3	Category For			Category For
4	Destination For			Destination For
5	Enrollment For			Enrollment For
6	Enrollment For dmo			Enrollment For dmo
7	Grade For			Grade For
8	Grade For dmo			Grade For dmo
9	Has Authorized Location			Has Authorized Location
10	Has Destination			Has Destination
11	Has Grade			Has Grade
12	Has Grade dmo			Has Grade dmo
13	Has Origin			Has Origin
14	Has Student			Has Student

© Copyright IBM Corporation 2013

5

## List Manager

The List Manager has the following three sections:

- The list display section is on the right side. It is where the entries in a list are displayed.
- The Manage By section is in the upper left. It is used to find lists, either by name or by type. Select a list to have the entries in it displayed.
- The Filter By section is in the lower left. It contains a list of all modules in the platform. Selecting one or more modules limits the display in the Manage By section to those lists associated with the selected module or modules.

The following actions are in the action bar at the top of List Manager:

- **Sort List:** Sort the values in a list alphabetically. You must use **Save Sequence** after you sort the list to keep the entries in that order.
- **Save Sequence:** Save the order of the entries in the list. Use after the Sort List action, or after manually changing the order of the entries by using the arrows.
- **New List:** Create a list.

- **View/Edit List:** Open the properties of a list.
- **Delete List:** Remove a list from the platform.

## Creating a manual list

### Set Source Type to Manual

**Important:** You cannot change most properties after you create them  
You must delete and re-create the list to change the properties

List Type Description		OK   Cancel
>> Name	cstSchedule	
Label	Schedule	
Description		
Type		
Language	US English	
System List	<input type="checkbox"/>	
Module	cstCourseManagement	
Dependent List	<input checked="" type="checkbox"/>	
Source Type	Manual	

© Copyright IBM Corporation 2013

6

#### *Creating a manual list*

To create a list, click the **New List** action. A blank properties panel opens.

Enter one or more of the following properties of the list, and click OK.

- **Name:** The unique name for the list.
- **Label:** The text that people see in the user interface to identify the list.
- **Type:** A text value that is used to group related lists, or any value that helps people find the list. In the Manage By section, you can sort by this property.
- **Language:** Is used for applications that must function in more than one language. The list is not automatically translated into this language.
- **System List:** If this property is selected, the list cannot be deleted.
- **Module:** This property specifies the module that the new list is associated with.
- **Dependent List:** Designates this list as a dependent list.
- **Source Type:** Manual or Dynamic. In the example in the slide, Manual is selected.

After you create a list, the List Manager does not automatically refresh the displayed list names. To force the list names to be refreshed, select **Type** in the Manage By window, and then select **Name** from the Manage By window.

## Editing a manual list

To add entries:

1. Add entries in Value field and press Enter after each entry
2. Click **Save Entries**

The screenshot shows a user interface for managing a list. At the top, there are two buttons: "Save Entries" and "Delete Entries". Below these are four input fields labeled "Value" containing the entries "F", "T, TH", and "M, W". There are also four checkboxes labeled "Select" next to each value. At the bottom of the list area are four buttons: "Select", "Delete", "Sequence", and "Value".

To delete entries:

1. Select entries
2. Click **Delete Entries**

The screenshot shows a user interface for managing a list. At the top, there are two buttons: "Save Entries" and "Delete Entries". Below these is a single input field labeled "Value" containing the entry "F". Below the input field is a table with four columns: "Select", "Value", "Sequence", and "Value". The first row contains a checked checkbox, a value "F", a sequence number "1", and a label "F". The second row contains an unchecked checkbox, a value "M, W", a sequence number "2", and a label "M, W". The third row contains an unchecked checkbox, a value "T, TH", a sequence number "3", and a label "T, TH".

To change the order of entries:

1. Use arrows to rearrange, or click **Sort List**
2. Click **Save Sequence**

[Sort List](#) | [Save Sequence](#)

© Copyright IBM Corporation 2013

7

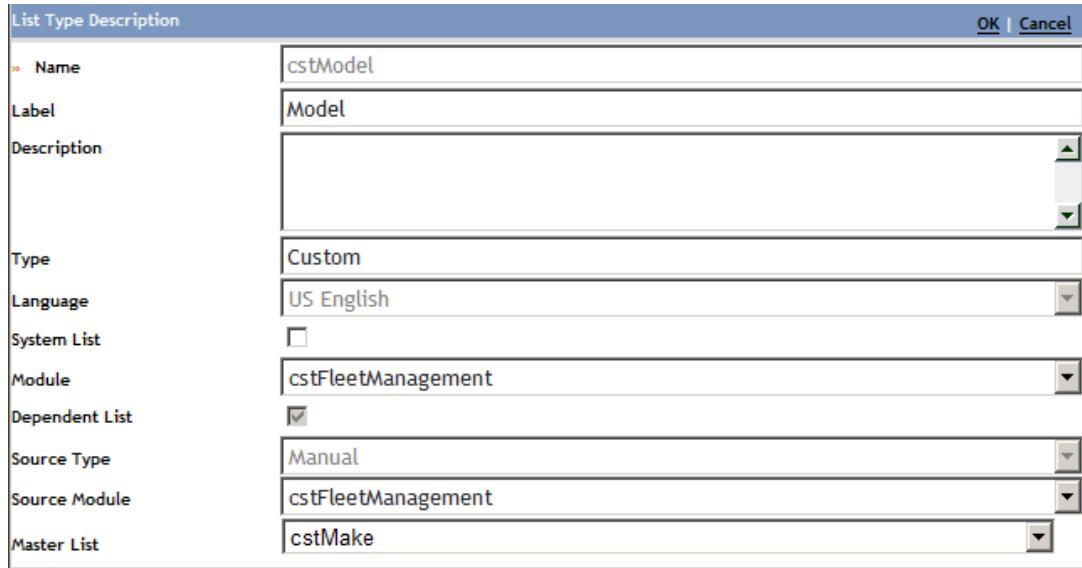
### Editing a manual list

To modify a manual list, select the list in the Manage By section. The values in the selected list are shown in the display area. To add an entry to a list, enter the text of the entry in the Value input area and press Enter. A new value input area is shown. You can add entries one at a time or in a batch. After you enter all the new values, click the **Save Entries** action. To delete an entry from the list, select the check box next to the entry and click the **Delete Entries** action. After an entry is saved in a list, it cannot be modified. If you want to change a list value, you must delete it and enter the correct value.

Entries are typically added to a list in alphabetical order. To change the sequence of the entries in the list, click an arrow in the order column. Clicking an upward pointing arrow moves the entry up a row; clicking a downward pointing arrow moves the entry down a row. To sort the list in alphabetical order, click the **Sort List** action. Click the **Save Sequence** action after you change the order of the values in a list.

## Creating a dependent list

- Parent list must exist before you can create dependent list
- Select the **Dependent List check** box, and select a master list
- The source type for dependent lists must be **Manual**



© Copyright IBM Corporation 2013

8

### Creating a dependent list

A dependent list is a manual-entry list that contains multiple sets of values. Each set of values depends on a selection from a master, or parent, list. A common example of a dependent list is that the model of a car depends on the maker. You must first select the maker of a car before you can see the list of models.

The following limitations apply to dependent lists:

- The parent of a dependent list must exist before you create the dependent list.
- Dependent lists cannot be dynamic.

You create a dependent list like a manual list, with these differences:

- The **Dependent List** check box must be selected. Setting this property causes the Source Type to be set to Manual and locked, because dependent lists cannot be dynamic.

When the Dependent List property is selected, two more properties are displayed:

- **Source Module** is the module that is associated with the Master List.
- **Master List** is the master, or parent, list. A set of values can be defined in the dependent list for each entry in the master list.

## Dependent list example

Parent List	Value	Sequence	Value
BMW			
Chevrolet			
Chrysler			
Ford			
Toyota		1	318i
		2	325i
		3	750

Parent List	Value	Order	Sequence	Value
Toyota				
			1	Avalon
			2	Camry
			3	Prius

© Copyright IBM Corporation 2013

9

### Dependent list example

Each entry in the parent list can have a different set of entries in the dependent list. Select an entry from the parent list to see the entries in the dependent list that are connected to that value from the parent list.

After you select an entry from the parent list, a dependent list works like any other manual list. You enter values in a dependent list like you do a manual list. You also sort the entries in the same way, by using the arrows or by using the Sort Sequence action.



## Dynamic lists

- Contents of lists are dynamically generated by using records from a specific module and BO
- The default order of list entries is alphabetical
- Dynamic lists cannot be dependent
- Fields must have a **Result Column** property selected to be available for a dynamic list

© Copyright IBM Corporation 2013

10

### *Dynamic lists*

A dynamic list is built from all records that are created from a specific business object. One or more fields from the selected BO can be used to make the list values. All fields that are used to create a dynamic list must have the Result Column property enabled in the BO.

If the value of a field that is used in a dynamic list changes, the dynamic list is not updated until the metadata cache is cleared. When a record that is the source of a dynamic list entry is deleted, that entry is removed from the list.

## Creating a dynamic list

1. Set **Source Type** to **Dynamic**
2. Select a **Source Module** and an **Object Type**
3. Select one or more fields from the list

The screenshot shows the 'List Type Description' dialog box. The 'Name' field is set to 'cstBuilding'. The 'Label' field is set to 'Building'. The 'Type' field is empty. The 'Language' field is set to 'US English'. The 'System List' checkbox is unchecked. The 'Module' dropdown is set to 'triCommon'. The 'Dependent List' checkbox is unchecked. The 'Source Type' dropdown is set to 'Dynamic'. The 'Source Module' dropdown is set to 'Location'. The 'Object Type' dropdown is set to 'triBuilding'. In the 'Fields' section, there is a table with three rows. The first row has 'Select' checked, 'Sequence' set to 1, 'Section Name' set to 'RecordInformation', and 'Field Name' set to 'Name'. The second row has 'Select' unchecked, 'Sequence' set to 0, 'Section Name' set to 'RecordInformation', and 'Field Name' set to 'Tenure'. The third row has 'Select' unchecked, 'Sequence' set to 0, 'Section Name' set to 'RecordInformation', and 'Field Name' set to 'Image'. At the bottom right of the dialog box, there are 'OK' and 'Cancel' buttons.

© Copyright IBM Corporation 2013

11

### Creating a dynamic list

A list cannot be both dynamic and dependent. When you create a list and set the source type to **Dynamic**, the Dependent field is cleared and locked.

Also, the Source Module and Object Type properties become visible. Together, these 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.

After the business object is selected, the Fields section is populated with a list of fields from the BO that have the **Result Column** property enabled. Select one or more of these fields. The values of the list come from the selected fields. In the example that is shown in the slide, only the Name field from each record is used to populate the list.

If more than one field is selected, the values of the fields are combined based on the order that is specified in the Sequence column. For example, suppose that two fields, Name and Short Name, are selected. If the sequence number for Name is 1 and for Short Name is 2, a value from the list might be Canada CA.

## Editing a dynamic list

- You cannot add or delete entries from a dynamic list
- You can change the order of the entries
  - Manually by using arrows in the **Order** column
  - Alphabetically by clicking **Sort List**
- You click **Save Sequence** after you change the order of the entries

Save Entries   Delete Entries		Sort List   Save Sequence   New List   Delete List   Label List		
Select	Order	Sequence	Value	
	▼	1	ATL01 - 1421 Peachtree Street	
	▼ ▲	2	BAL01 - 9 W. Mt. Vernon Place	
	▼ ▲	3	BLO02 - Tokyo Site	
	▼ ▲	4	BLO03 - Dallas Site	
	▼ ▲	5	BLO04 - San Francisco Site	
	▼ ▲	6	BLO05 - Austin Site	
	▼ ▲	7	CAL01-455 5th Street SW	
	▼ ▲	8	CHI01-6133 North River Road	
	▼ ▲	9	DAL01-3000 Executive Parkway	

© Copyright IBM Corporation 2013

12

### Editing a dynamic list

Because the entries in a dynamic list are created from records, they cannot be manually added or deleted from the list. You must create or delete records to change the entries in a list. It is possible to change the order of entries in a dynamic list. The entries can be moved manually by using the arrows in the Order column. The entries can also be sorted alphabetically by using the Sort List action. Remember to use the Save Sequence action to save the order of the entries.

## Attaching a list to a field

13

### Attaching a list to a field

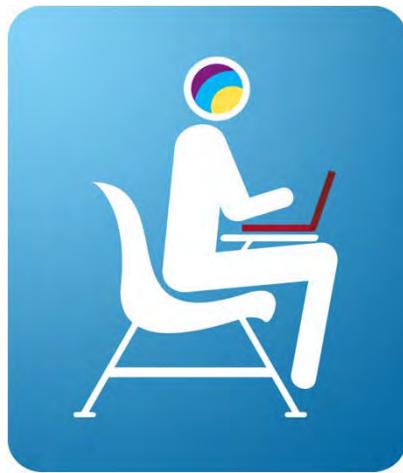
To attach a list to a field in the Data Modeler, you use the following steps:

1. Click **Tools > Builder Tools > Data Modeler**.
2. Open the business object that contains the field where you want to attach a list. If the field is not defined in the business object, use **Add** or **Find** to add it to the business object.  
The Field Type value must be **List**.
3. In the Field List of the business object, click the name of the field to open its properties.
4. At the bottom of the properties window is a property named **Module**. Select the module that the list is defined under.  
The List property shows the lists that are defined under that module.
5. Select the list that you want attached to this field. If you do not see the list that you want, return to the List Manager and examine the properties of the list. Correct any errors and try again.
6. Select a value for the Default Value property, if you want this field to have a default value.
7. Click **Save Field** to save the changes to the properties of the field.

---

## Student exercises

---



© Copyright IBM Corporation 2013

14

### *Student exercises*

Perform the exercises for this unit.

# Lesson 2. Classifications



## Lesson 2: Classifications



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Classifications are another means of ensuring data integrity by providing a set of choices to the user. This lesson covers classifications in IBM TRIRIGA and how to use them.

### What you should be able to do

After completing this lesson, you should be able to view and edit classifications.

## Definition of classifications

A set of hierarchical choices

The screenshot shows a software interface with a header bar containing the word 'Hierarchy'. Below the header is a toolbar with three buttons: 'Open', 'New', and 'Delete'. The main area displays a hierarchical tree structure under the heading 'Classifications'. The tree includes the following nodes:

- 0-Current Preventive Maintenance Level (Current Preventive Maintenance Level)
- Accounting Service Code (Accounting Service Codes)
- Accounting Service Code Group (Accounting Service Code Group)
  - Leased Location Cost Code Structure (Accounting Service Code Group)
  - Owned Location Cost Code Structure (Accounting Service Code Group)
- Amendment Type (Amendment Type)
- Amenities (Amenities)
- Appropriateness (Appropriate For Use)
- Area Type (Area Type)
- Asset Ownership (Asset Ownership)
- Assignment Type (Assignment Type)

© Copyright IBM Corporation 2013

16

### Definition of classifications

Classifications are like lists in that both allow someone to select a single item from multiple possibilities. The difference is that classifications consist of records that are organized in a hierarchical fashion. Because they are records, a classification has fields that provide more information about the classification. In other words, these fields can provide *context* for the classification.

The fields in a classification can also contain data that is used to determine the outcome of computations. For example, a classification of Equipment Type might have a field whose value is the phone number to call if the equipment needs repair.

Each classification directly under the root Classifications is a different type of classification. There are many such classifications, including Status, Request Class, and Service Class. Each of these classifications can be a parent for their own branch of the Classifications hierarchy. Classifications not directly under the Classifications root are the same type of classifications as their parent.

## Comparison of lists and classifications

**Similarity:** Defined set of choices for the user

### Differences:

- Classification choices are determined in advance  
Lists can be a dynamic set of choices
- Classifications are all hierarchical  
Lists are typically flat
- Lists only identify names  
Classifications are records that can contain information and provide context for the choice

© Copyright IBM Corporation 2013

17

#### *Comparison of lists and classifications*

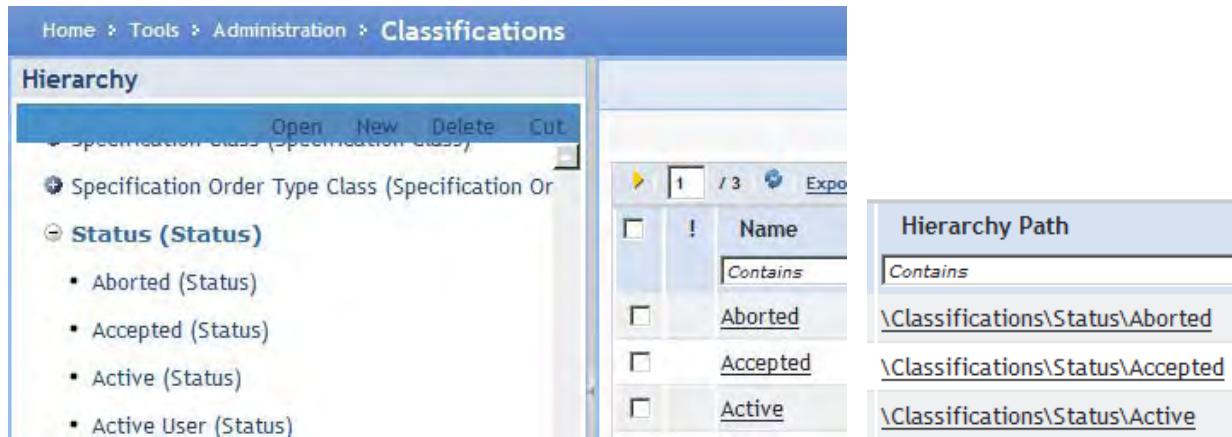
You cannot make a definite statement that Classifications are better than Lists or that Lists are better than Classifications. Both serve the same purpose: to provide a defined set of choices to the user, which helps ensure data integrity.

Lists and classifications are different in the following ways:

- Lists are typically a static set of choices that are determined in advance and are not changed by an application. The exception is a dynamic list that is generated from the contents of records. Classifications can be a dynamic set of choices that reflect data that the application can control.
- Lists are easier to set up and are a good choice to represent single options that rarely change. Classifications are more difficult to set up, but they provide more flexibility if future changes are needed.
- Classifications are useful for creating a hierarchy. Some hierarchical organization is possible with dependent lists, but managing dependent lists with more than three levels is not practical.
- You can use classifications to identify live records. Lists are only names.

## Classification hierarchy

Click **Tools > Administration > Classifications**



© Copyright IBM Corporation 2013

18

### Classification hierarchy

You create and control classification records in the classifications *hierarchy*. You can access the hierarchy by clicking **Tools > Administration > Classifications**. In the classification hierarchy, you see the hierarchy panel on the left side and the classification record panel on the right side.

The classifications Hierarchy panel starts with the Classifications root entry and shows the entire tree of classification entries. In the Hierarchy panel, each entry has an icon to its left, with these meanings:

- If the entry can be expanded, it has a blue circle with a plus sign in it. Click the circle to expand this level of the hierarchy. The blue circle changes to a white circle with a minus sign when you click it.
- An entry that is expanded has a white circle with a minus sign in it. Click this icon to collapse this level of the hierarchy.
- Entries that have no child records are shown with a small blue dot next to them.
- Clicking any classification entry in the Hierarchy brings focus to it and changes the text to bold. It also causes any records immediately below the selected record to be displayed in the right

half of the screen. When an entry in the classification hierarchy has focus, you can use these actions:

- **Open:** Open the record.
- **New:** Create an entry in the hierarchy with the selected record as its parent. After you click this action, you see a list of classification types that can be created below this level of the hierarchy. Select one of them to create that type of record below the current record.
- **Delete:** Delete the selected entry from the hierarchy.
- **Cut:** Temporarily remove the entry and its child records from the hierarchy. Selecting another entry in the hierarchy shows the **Paste** action. Use this action to place the cut entries in the hierarchy.

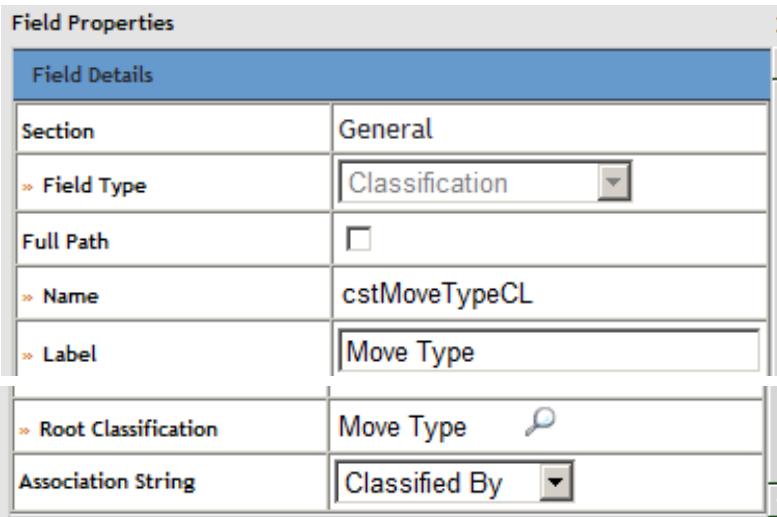


**Note:** Entries must be defined as child records of the classification where you want to paste them, or the paste fails.

The classification records panel displays all classification records immediately below the classification record that has focus. The panel shows the name and hierarchy path of each record, with filters for sifting through the list. Click the name of any record to open it.

## Classification fields and associations

- Classification fields in a business object identify a classification record
- For each classification field in a record, an association exists from the record to the selected classification record



© Copyright IBM Corporation 2013

19

### Classification fields and associations

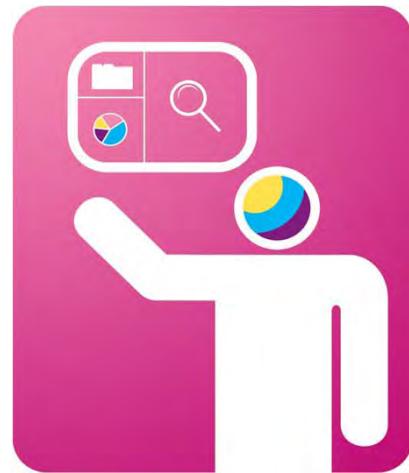
A field with a Field Type value of Classification can store a selection from the classification hierarchy. Before you can create a classification field, an association definition must exist from the business object that contains the field to and the Classification business object.

In the Data Modeler, open the properties of a classification field, and set these properties for the field:

- Root Classification:** This property defines the scope of the classification hierarchy that is available to this field. Selections for this field are limited to the selected classification and all entries below it. Entries outside of the specified root are not available for selection. If Classifications is selected, the entire classification hierarchy is available to this field.
- Association String:** This property identifies the association that is defined from this business object to the Classification business object.
- Full Path:** This property controls the text that is displayed for this field. If this check box is selected, the full path of the hierarchy is shown for the selected entry. An example is **\Classifications\Car Category\SUV**. If this property is cleared, which is the default, only the name of the classification is shown. An example is **SUV**.

## Instructor demonstration

- Demonstration of Classification \Classifications\Request Class\Building Services\Climate Control\Room Too Warm
- Demonstration of Room Too Cold



© Copyright IBM Corporation 2013

20

*Instructor demonstration*



## **Checkpoint questions**

- 1.** What is the purpose of lists and classifications?
- 2.** What types of lists exist in the platform?
- 3.** How do you change a value in a Manual list?
- 4.** What is the key difference between a list and a classification?
- 5.** What can a classification provide that a list does not?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. What is the purpose of lists and classifications?  
*To ensure data integrity by limiting the user's choices.*
2. What types of lists exist in the platform?  
*Manual, Dynamic, and Dependent.*
3. How do you change a value in a Manual list?  
*Delete it and add the correct entry.*
4. What is the key difference between a list and a classification?  
*A list entry is a simple value. A classification entry is a record.*
5. What can a classification provide that a list does not?  
*The fields of a classification record can provide context.  
A list has no way to provide context.*



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Create, view, and edit lists
- View and edit classifications



# 10 Data Integrator



# 10 Data Integrator



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

## What this unit is about

This unit is about learning how to load data into IBM TRIRIGA from tab-delimited text files.

## How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

## References

*Application Building for the IBM TRIRIGA Application Platform*



## **Objectives**

After completing this unit, you should be able to perform the following tasks:

- Describe the purpose of the Data Integrator and how it works
- Use the Data Integrator to create a header file
- Use Data Integrator to import data

# Lesson 1. Using the Data Integrator



## Lesson 1: Using the Data Integrator



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is about learning how to load data into IBM TRIRIGA from tab-delimited text files.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe the purpose of the Data Integrator and how it works.
- Use Data Integrator to create a header for a file.
- Use Data Integrator to import data.

## References

*Application Building for the IBM TRIRIGA Application Platform.*

*Application Building for the IBM TRIRIGA Application Platform 3: Data Management*

## Defining the Data Integrator

- Definition of the Data Integrator
  - A tool for creating or updating records in IBM TRIRIGA
  - Works only with tab-delimited text files
- Two specific steps in Data Integrator
  - Create tab-delimited text files
  - Process the tab-delimited text files
- Access to the Data Integrator
  - Click **Tools > Administration > Data Integrator**

© Copyright IBM Corporation 2013

4

### Defining the Data Integrator

IBM TRIRIGA Data Integrator is a tool for importing data into IBM TRIRIGA. You can use the Data Integrator to capture groups of data records and upload them into the system without entering each data record manually. Importing data saves time and avoids the data integrity and quality issues inherent in manually entering the records.

Record data can come from any source, such as a database, Excel file, or any other external system. Regardless of your data source, Data Integrator accepts only files in a **tab-delimited text** format (.txt).

Another function of Data Integrator is that it can be used for bulk updates of existing records. Bulk updates are much more efficient than working through the form and performing the numerous actions required to update each record individually.

There are two distinct steps in using Data Integrator:

- Create tab-delimited text files with the header line and data lines.
- Process the tab-delimited text files to create or update records.

You can access the Data Integrator by clicking **Tools > Administration > Data Integrator**.

## Requirements for Data Integrator files

- Header row for the fields in the file
  - Fields for the publish name
  - Required fields, if you create a record
- One row of tab-delimited data for each record

The screenshot shows a Windows Notepad window titled "Instructors.txt - Notepad". The menu bar includes File, Edit, Format, View, and Help. The content of the file is tab-delimited data:

triIdTX	triLastNameTX	triFirstNameTX	triworkPhoneTX	triEmailTX
INST10055	Cockcroft	Julie	(218)494-1896	jcockcr@company.com
INST10056	Meredith	Jennifer	(380)855-2948	
INST10057	Waterhouse	Daniel	(958)274-3084	dwaterh@company.com
INST10058	Tulip	Nicole	(409)119-2105	ntulip@company.com
INST10059	Bond	Seth	(570)733-2916	sbond@company.com

© Copyright IBM Corporation 2013

5

### Requirements for Data Integrator files

Two types of requirements for Data Integrator are functional and usability. If the functional requirements are not met, Data Integrator fails on upload. If the usability requirements are not met, the data upload is successful but can impose problems on the user.

Functional requirements have the following format:

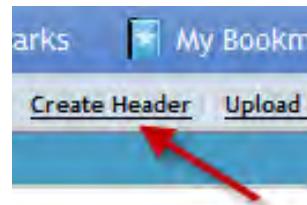
- **File format:** The file must be formatted as tab-delimited text, with a file extension of **.txt**.
- **Header row:** The first row of the data file must contain the names of the fields in the business object for which you are uploading data. The fields can be in any order that you choose. The field names must be spelled and capitalized the same as they are in the business object.
- **Publish Name:** You must include all fields that define the publish name of a business object in the data upload. Otherwise, Data Integrator might fail to create a record, create a record rather than update a record, or update the wrong record.

Usability requirements have the following format:

- **Required fields:** Data Integrator does not require that the data files contain the fields that are required by the business object. However, any records that are created without the required fields can affect the user. The user must populate all required fields before they can complete any action on the record. It is preferred practice that new records contain all required field values to ensure a smooth user experience.
- **Field value positioning:** After the header row, each row of the file describes a record that is to be created or updated. The order of the field values in these rows must match the order in the header row.
- **Control Number:** Data integrator always creates new control number values when it processes a row from a data file. If the publish name contains a control number field, existing records cannot be updated. It is good practice not to include the control number in the publish name of any business object that you plan to update with Data Integrator. For publish names that contain control number fields, you must redefine the publish name to eliminate the control number before the file is uploaded. After the upload is finished, you can change the publish name back to what it was before.
- **Calculations:** Do not populate read-only fields directly. Instead, run a custom workflow to trigger the calculations after the upload.

## Creating a header file

1. Specify the module, BO, and form
2. Click the **Create Header** link



A screenshot of the 'Step 1 Select File to Import' screen. The 'Upload File' panel contains the following configuration:

Module	triPeople	Business Object	triPeople
Form	triEmployee		
Import Type	Add	Action	triCreateDraft
File Type	Tab Delimited (*.txt)		
File Char Set	UTF-8		
Batch Upload	<input type="checkbox"/>		

© Copyright IBM Corporation 2013

6

### Creating a header file

A **header file** is an import template in which the first header row declares field names into which data is imported. You can create this header in many ways. No matter how the file is created, the template field headers must match the corresponding business object field names in case and spelling.

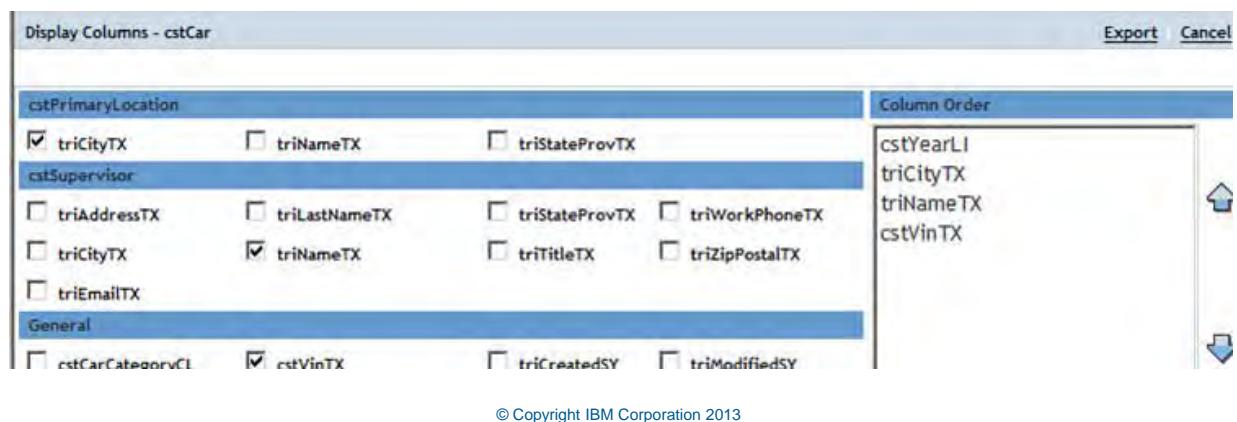
To facilitate creating the header file, Data Integrator has a built-in tool for creating an Excel-compatible file that contains the field names in the first row. This header file becomes the upload template into which all records are entered.

Navigate to the Data Integrator by clicking **Tools > Administration > Data Integrator**, and then follow these steps to create a header file:

1. Select values for the **Module**, **Business Object**, and **Form** fields to determine the fields that are available for use in this file.
2. Click the **Create Header** link to open the Display Columns field selection panel.

## Creating a header file (continued)

3. Select the fields that you want, in the order you want them  
Include publish name and required fields
4. Change the order by using the arrows, if needed
5. Click **Export**
6. Open the file by using Excel
7. Save as tab-delimited text



*Creating a header file (continued)*

3. Review the Display Columns window. It contains a check box for every field in the selected business object and form, and is organized by section.  
Select the fields that you want for the file. As you select a check box, the system adds the name of the corresponding field to the Column Order box in the upper right.  
Be sure to include all elements of the Publish Name and all required fields.  
The fields in the Column Order section are in the order in which they were selected. The sequence in the Column Order section is the order that the fields are displayed in the Header File. You can reorder fields by clicking the name of a field to select it and clicking the up arrow or down arrow until the field is positioned.
4. When the list of fields in the Column Order section is complete and the fields are in the correct sequence, click **Export**.
5. When prompted, click **Save**.
6. Select the destination on your local computer.

Data Integrator gives the file an .xls extension, although the file is technically generated in HTML format.

7. Change the extension to **htm**. Click **Save**.
8. Find the saved file and open it in Excel. Notice that the field names you chose are in the first row of the file.
9. Enter your data into the appropriate columns. (Skip this step if you are not entering data directly into the Header File.)



**Hint:** Change all cells in your spreadsheet to text format to eliminate the Excel conversion formatting of your data. To change the format of your cells, select the entire spreadsheet, and click **Format > Cells > Text > OK**.

10. Save the file as a tab-delimited text file.

## Uploading a data file

1. Select the module, business object, and form
2. Select an action
3. Click **Browse** to find the file
4. Click **Upload File**

[Upload File](#)

The screenshot shows the 'Upload File' dialog box. The 'Module' dropdown is set to 'triPeople'. The 'Business Object' dropdown is also set to 'triPeople'. The 'Form' dropdown is set to 'triEmployee'. The 'Import Type' dropdown is set to 'Add'. The 'Action' dropdown is set to 'triUploadHidden'. The 'File Type' dropdown is set to 'Tab Delimited (\*.txt)'. The 'File Char Set' dropdown is set to 'UTF-8'. There is a 'Batch Upload' checkbox which is unchecked. The 'Name' field contains the path 'C:\Users\Administrator\Desktop\Employees.txt'. A 'Browse...' button is located to the right of the 'Name' field. At the bottom left of the dialog box, there is a copyright notice: '© Copyright IBM Corporation 2013'.

8

*Uploading a data file*



**Hint:** Test all Data Integrator files against a copy of your production system data before you upload the data into production.

You can upload data into the IBM TRIRIGA system with Data Integrator one file at a time or in batch mode. The instructions in this lesson are for a single file upload.

After the Header File is created, populated with record information, and saved as tab-delimited text, you are ready to upload the file into Data Integrator. Follow the same steps regardless of the type of data you are uploading (associative, hierarchical, non hierarchical).

1. Navigate to the Data Integrator by clicking **Tools > Administration > Data Integrator**.
2. Set the following properties in the Upload File section:

- Select the values in the **Module**, **Business Object**, and **Form** fields to match the header file. Failure to select the matching settings results in a failed upload or a successful record upload into the incorrect object.
- Accept **Add** as the value for **Import Type**. **Add** means that if the values for the publish name fields in a row match an existing record, the other values in the row are used to update that record. If the publish name values do not match an existing record, the data in the row is used to create a record.
- Select the **Action** to be performed on a record after the Data Integrator creates it. The triUploadHidden action is preferred for all uploads, when available.

When a record is first created, it is in a special state called *null*. Records in the null state disappear after the operation that is using the record is done. It is important for the Data Integrator to perform the selected action on new records to move them out of the null state. The actions in the Action list are actions that are defined in the state family to transition records out of the null state.

To run a workflow after the Data Integrator creates a record, the workflow must be triggered by a sub action that is attached to the selected action.

- The only choice for the **File Type** property is **Tab delimited**.
  - Ensure that the **Batch Upload** check box is cleared. This property tells Data Integrator to read the data file you specify only once.
  - Accept the default **File Char Set**. It specifies the character set that is used for the file. It is best not to change this value unless you are certain that your file uses a different character set than the default.
3. Click **Browse**, and find and open the data file that you want to upload.
  4. Click **Upload File**.
  5. Click **OK**. Processing of the file begins. After the process is complete, you receive notification in the Notifications section of your home portal. Ensure that all records are processed correctly.

To upload a file, Data Integrator reads a row from the input file. It creates a record with that data, as specified by the header row. Data Integrator checks for the existence of a record from the selected business object with the same publish name as the record it created. If an existing record is not found, the created record is added to the database. If an existing record is found, then Data Integrator does a field-by-field update of the existing record with values from the input file. Any fields that are contained in the existing record but are not matched by a field in the input file remain unchanged. This process is repeated for each line of the input file.



**Note:** New values are created for any control numbers that are used in the record, which prevents records with control numbers in the publish name from being updated by Data Integrator.

## Adding to a hierarchy

Data Integrator can add new records into an existing hierarchy

- Input file must have a column named **Parent**
- Parent column contains hierarchical name of parent record
- Parent record must exist before you upload data

triIdTX	triNameTX	triLevelNU	triGrossAreaNumNU	triGrossMAreaNU	Parent
1015	Main Floor	1	50000	40000	\Locations\Main Campus\Smith Building
1016	Second Floor	2	52000	42000	\Locations\Main Campus\Smith Building
1025	Main Floor	1	50000	40000	\Locations\Main Campus\Jones Building
1026	Second Floor	2	52000	42000	\Locations\Main Campus\Jones Building
1035	Main Floor	1	50000	40000	\Locations\Main Campus\Kimball Building
1036	Second Floor	2	52000	42000	\Locations\Main Campus\Kimball Building

© Copyright IBM Corporation 2013

9

### Adding to a hierarchy

Data Integrator can also import or update records that are child records to another business object. In other words, Data Integrator can upload hierarchical data. Examples of hierarchical data include floors, which are child records of buildings, and spaces, which are child records of floors.

Uploading data for a hierarchical module requires adding a column with the key word of **Parent** to the header row of the data file. This column links the business object child record to the appropriate parent in the hierarchy tree.

Follow these steps to create a header file for a hierarchical module:

1. Perform the steps to create a non hierarchical header file.
2. Open the file in Excel.
3. In the header row, type the keyword **Parent** in a new column immediately to the right of the last column of data.
4. In each data row, in the Parent column, enter the full path of the parent of the existing record, including the first backslash (\). Do not put a backslash (\) at the end of the Parent path.

Data Integrator uses the full path in the Parent column to find the parent record. If no parent record is found, then orphan records are created that do not display in the Hierarchy window.

## Creating an association

Data Integrator can associate existing records to records that you upload

- Input file must have a column named **Associate**
- Associate column defines properties of association
- Association is created from existing record to uploaded record

	A	B	C	D
1	cstYrLI	cstMakeLI	triIdTX	Associate
2				%%triPeople:triPeople%%
3				%%Supervises%%
4	1999	Lexus	DI-1	Wilk, Valerie-EMP100007
5	2000	Toyota	DI-3	

© Copyright IBM Corporation 2013

10

### Creating an association

You can have Data Integrator create associations between existing records in the IBM TRIRIGA Application Platform environment and the records it uploads. To arrange for these associations, add information to identify each association in the data file. The information specifies the following information:

- The module and business object that are used to create the existing records that are to be associated to the records that are uploaded.
- The name of the association from the existing records to the new records. This association must already be defined at the business object level.



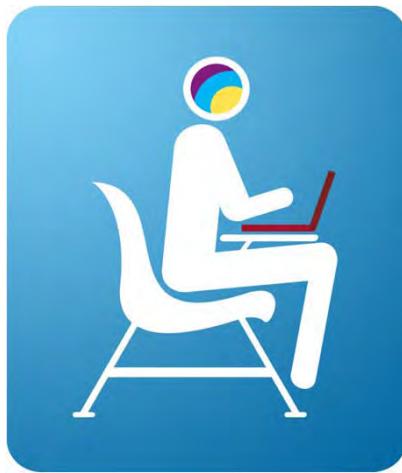
**Note:** The associations are created *from* the existing records *to* the records that are being uploaded. Associations in the reverse direction are also created by the platform, provided that such an association definition exists at the business object level.

For each uploaded file, you can use only the records from one business object for the associations. You must create multiple upload files if you have multiple sets of records that come from different business objects for which you want to create associations.

Follow these steps to add associations to the header file that contains the records to be uploaded:

1. In the header row, type the keyword **Associate** in a new column immediately to the right of the last data field.
2. Insert two rows between the header row and the first row of data. The data now begins in row 4.
3. In row 2 of the Associate column, enter the module and business object name. These values must have a colon between them and have two percent symbols on each end, as **%%Module:Businessobject%%**. An example is **%%tripeople:tripeople%%**. This row reflects the module and business object of the existing records.
4. In row 3 of the Associate column, enter an association string that has two percent symbols on each end. For example, **%%Manages%%**. This string identifies the specific association that is defined from the module and business object that is named in row 2 to the business object selected for the file.
5. In each row of the spreadsheet, in the Associate column, enter the publish name of the existing record that you want to associate. It must match the publish name exactly in spelling and case. You do not have to create an association for every data row in the file. Leave the Associated column blank for any rows where an association is not to be created.

## Student exercises



© Copyright IBM Corporation 2013

11

### *Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

---

- 1.** How must you format a Data Integrator file?
- 2.** What happens if your input file does not contain all the fields that are used in the publish name?
- 3.** What happens if your input file does not contain all the required fields?
- 4.** When importing a file, what is the purpose of the action? What actions are available?
- 5.** What column must be in the input file for hierarchical data?

*Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. How must you format a Data Integrator file?

*Tab-delimited text, with a header row naming the fields.*

2. What happens if your input file does not contain all the fields that are used in the publish name?

*Records cannot be identified correctly without all the Publish Name fields. Data Integrator might fail to create a record, it might create a record when it is supposed to update a record, or it might update the wrong record.*

## **Checkpoint answers (continued)**

3. What happens if your input file does not contain all of the required fields?

*You can create records from the data in the file but do not have the missing fields. When edited in a form, these records cannot be saved without entering the missing required values.*

4. When importing a file, what is the purpose of the action? What actions are available?

*To move the record away from the null state. The transitions that are defined in the state family at the null state.*

5. What column must be in the input file for hierarchical data?

*Parent.*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Describe the purpose of the Data Integrator and how it works
- Use the Data Integrator to create a header file
- Use Data Integrator to import data





## 11 Object migration: Import



## 11 Object migration: Import



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

Object migration is the process of transferring development work from one environment to another. Import is the part of the process where objects are loaded into an environment.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform: Object Migration User Guide*

## Objectives

After completing this unit, you should be able import an object migration package

# Lesson 1. Importing an object migration package



## Lesson 1: Importing an object migration package



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Object migration is the process of transferring development work from one environment to another. Import is the part of the process where objects are loaded into an environment.

### What you should be able to do

After completing this lesson, you should be able to import an object migration package.

## References

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform: Object Migration User Guide*

## Object migration

### Definition of object migration

- Applications are composed of objects, such as modules, business objects, forms, workflows, reports
- Object migration is the process that you use to transfer these objects from one environment to another
  - From development to test to production environments
- Object migration is used in IBM TRIRIGA for Application and Platform upgrades

© Copyright IBM Corporation 2013

4

#### Object migration

When you finish developing or customizing an application, you want to migrate your work, such as business objects, forms, or workflows, to other environments. This migration might be done for testing, or to put the items into production. You might want to create a copy of the objects that exist in the system for an archival copy or an offline backup.

IBM TRIRIGA Application Platform has an element to select objects and write them to a file. There is also a corresponding element to read objects from a file into an IBM TRIRIGA Application Platform environment. Both of these elements are part of a tool in the platform named **Object Migration**. Object Migration is the same process that is used to distribute upgrades to the IBM TRIRIGA applications and platform.



**Note:** Do not allow users to access the IBM TRIRIGA environment while you are processing object migration packages. User activity can result in data corruption or locked tables.

## Object migration, continued

Object migration involves exports and imports:

- Export specifies a set of objects to send to a compressed file
- Import loads objects from a compressed file
- Click **Tools > Administration > Object Migration**

Export Packages		
Name	Status	User
<a href="#">Cert Package 2</a>	Exported	Iverson, Don - 1000000
<a href="#">PULSE Lab Workflow</a>	Exported	Iverson, Don - 1000000
<a href="#">cert class stuff</a>	Exported	Iverson, Don - 1000000
<a href="#">one workflow</a>	Exported	Iverson, Don - 1000000

Import Packages		
Name	Status	User
<a href="#">DMO Platform Certification Ver 10.0 4-28-2011</a>	Imported	Iverson, Don - 1000000
<a href="#">Grades 10.0</a>	New	Iverson, Don - 1000000
<a href="#">cstSmartSectionDemo_With_Link_Disabled Ver 10.0</a>	Imported	Iverson, Don - 1000000

© Copyright IBM Corporation 2013

5

### Object migration, continued

The Object Migration process is about imports and exports. An export specifies a set of objects to be written out to a file so that it can be read by other IBM TRIRIGA Application Platform environments. The objects are put into a special compressed (.zip) file. If you disturb the control information, then the file cannot be used by the Object Migration process. For this reason, it is not a good idea to manipulate the contents of the export file.

An import contains objects from a file to be migrated into this platform environment. The means of transferring the object migration file from one environment to another is your choice. It might be something that you accomplish by yourself, or another person might be involved.

The Object Migration page is accessed through **Tools > Administration > Object Migration**. The Object Migration page shows all object migration packages, with the status of each package, the user who created the package, and the description of the package. The packages are separated into an Export Packages section and an Import Packages section. Packages in each section are sorted alphabetically by name.

The status of the object migration package can be New, Validated, Export Pending, Exported, Imported, Validation Pending, or Validation Failed. To see the specifications of an Object Migration

package, click the hyperlinked name. To delete an Object Migration package, select the check box to the left of the package name and click Delete. If you want to create a package for export that contains the same items as an existing package, click the **copy package** icon for that package. Only existing objects are put into the new package.



## Importing

---

Steps for importing an Object Migration package:

1. Upload the Object Migration package into the IBM TRIRIGA Application Platform environment
2. Validate the objects that you migrate
3. Import the migrated objects into the platform environment

### *Importing*

There are three basic steps to importing the contents of an object migration file:

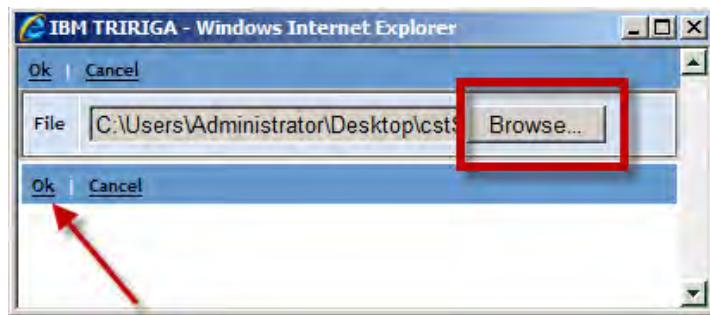
1. Upload the Object Migration package into the IBM TRIRIGA Application Platform environment. The system performs a preliminary validation as it converts the contents to a form that can be processed. During this validation, if the module or business object of a module does not exist in the package or the target environment, the object is marked as Invalid.
2. Validate the contents of the package. The purpose of this step is to detect problems with the uploaded objects before actually importing them into the target environment. For example, an upload package might contain a form that has a dependency on a workflow. If the workflow does not exist in the package or the target environment, the import process cannot correctly re-create that form definition, which is a problem.
3. Import the migrated objects into the platform environment. This step performs the task of adding to or updating the target environment with the uploaded objects or data.



**Hint:** Import an Object Migration package when the environment has minimal system activity.

## Uploading the Object Migration package

1. From the Object Migration page, click **New Import Package**
2. Click **Browse** to find the file
3. Click **Ok** to upload the Object Migration (OM) package



© Copyright IBM Corporation 2013

7

### Upload the Object Migration (OM) package

The object migration import process begins by uploading the contents of the Object Migration .zip file to the IBM TRIRIGA Application Platform environment. During this step, the system performs these actions:

- Uploads the file
- Extracts all objects from the file
- Performs a preliminary validation on the objects
- Shows the resulting objects and validation information in the Object Migration window

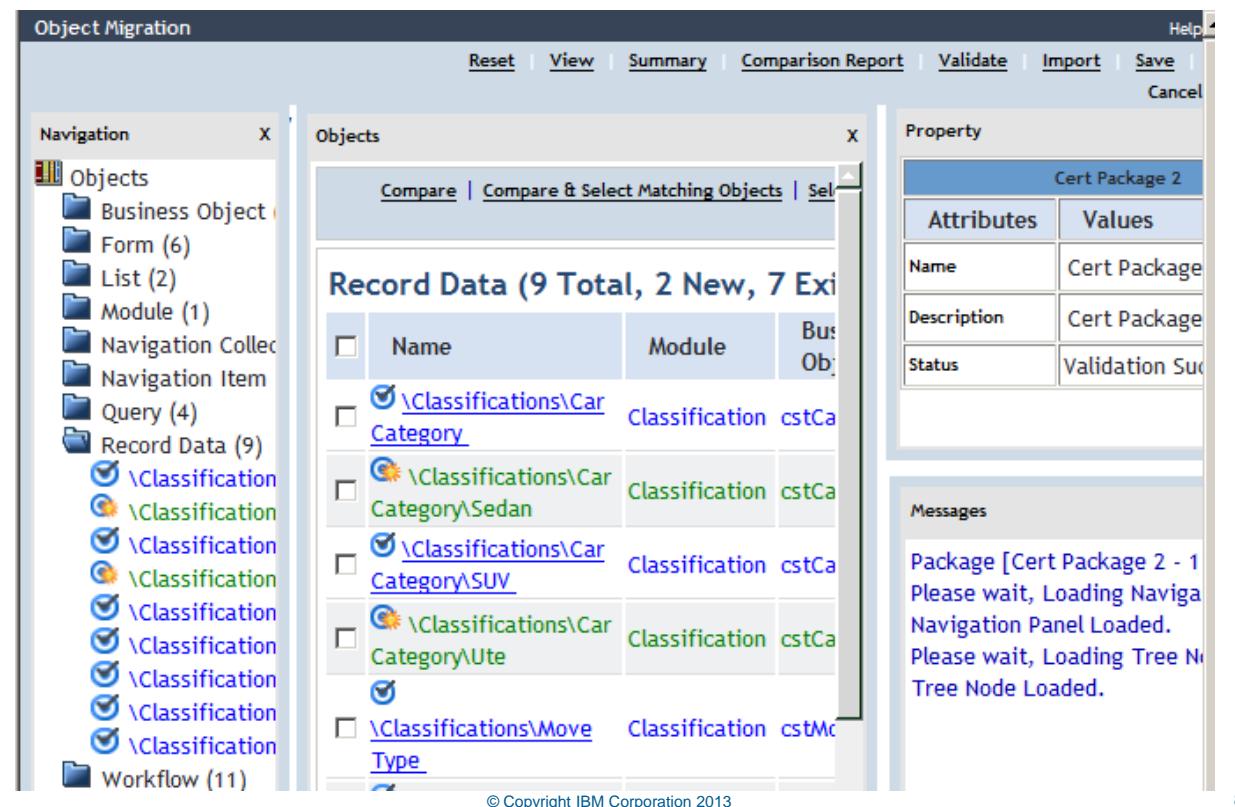
To begin the upload process, use the following steps:

1. Click **Tools > Administration > Object Migration**.
2. From the Object Migration page, click **New Import Package**.  
The Create Package window opens.
3. Click the **Browse** button to open a search window for the Object Migration file.

4. Find the file to be imported and either double-click it or click it and click **Open**. Click **Cancel** at any time if you want to stop the Import process.
5. When the file is selected, click **Ok**.

This action uploads the package to the IBM TRIRIGA Application Platform for preliminary validation and for viewing in the Object Migration window.

## Object Migration window for an import



Object Migration window for an import

The Object Migration window shows the objects in the package, the status of each of them, the overall status of the migration, and all relevant messages. The Object Migration window is organized into these panels:

- **Navigation:** Shows what objects are included in the import package.
- **Objects:** Presents what is included in the selected item in the Navigation panel.
- **Property:** Contains the definition of the import package.
- **Messages:** Displays what the system is doing with the package. Click **Clear** to remove all messages.

The names of objects that are shown in the Navigation and Objects panel are shown in Green, Blue, and Red. These colors indicate the status of the object, as follows:

- Green means that the object is in the migration package and that there is no matching object with the same name in the target environment.
- Blue means that the object is in the migration package and that there is a matching object with the same name in the target environment. The import process causes the platform object to be

updated from the object in the migration package according to the import rules. These rules are described in the *Object Migration User Guide*.

- Red means that dependent items are not in the package and not in the target environment. Red can also indicate that the object has an error. If you have red objects, you cannot import the migration package. You must resolve all errors before you can import the package into the platform environment.

The actions at the top of the Object Migration window are as follows:

- **Reset:** Returns the Object Migration window to the state it was in when you last saved it.
- **View:** Is used to restore one of the panels in the Object Migration window.
- **Summary:** Produces the Summary report, which has details about the warnings and errors that are found during validation.
- **Comparison Report:** Produces the Comparison report, which shows a detailed view of differences between the uploaded items and the existing items in the platform.
- **Validate:** Performs the validation process.
- **Import:** Imports the objects from the package to the destination environment.
- **Save:** Saves the information in the Object Migration window.
- **Cancel:** Closes the Object Migration window. Unsaved changes are lost.

## Validation process

During the validation process, the system analyzes the uploaded XML (from the .zip file), and checks for invalid or missing references to other objects. For example, if a business object has a pre-create workflow, the validation process checks whether the workflow is in the package or in the system. If not, the system logs a warning against that business object.

Sometimes, this analysis discovers more serious situations, causing the system to log an error against the object and rendering it invalid. One example of an error is attempting to change the business object type from Stand Alone to Embedded. IBM TRIRIGA Application Platform system does not allow this, and therefore Object Migration considers it an error. You must resolve all invalid objects before you can import the package.



**Note:** The system always validates *all* objects before it reports a successful or failed validation. The person who is responsible for importing the package can then see all warnings and errors at one time.

## Summary report

Includes warnings and errors that are found during validation

Package Summary								
Name:	Cert Package 2							
Description:	Cert Package 2							
Status:	Comparison Report Complete							
Object Summary								
Type	# of Objects	New	Existing	Valid	Invalid	Warnings	Errors	
1 Business Object	8	-	8	8	-	-	-	
2 Form	6	-	6	6	-	-	-	
3 List	2	-	2	2	-	-	-	
4 Module	1	-	1	1	-	-	-	
5 Navigation Collection	2	1	1	2	-	-	-	
6 Navigation Item	3	-	3	3	-	-	-	
7 Query	4	-	4	4	-	-	-	
8 Record Data	9	2	7	9	-	-	-	
9 Workflow	11	1	10	11	-	-	-	
	46	4	42	46	0	0	0	

© Copyright IBM Corporation 2013

9

### Summary report

To see all warnings and errors that are found during validation, click the **Summary** action at the top of the Object Migration window. The Package Summary report is created and displayed.

The Object Summary at the top of the Package Summary report lists statistics for each object type. Warnings and Errors are hyperlinked to the corresponding summary section further down in the report.

The body of the Package summary report contains a separate section for each object type with warnings and errors. Within each section of the report are the object names, with their module and business object (if appropriate), and the corresponding warnings or error messages.

## Comparison report

Includes a detailed comparison of objects in the package to existing objects

OBJECTID	OBJECTNAME	OBJECTTYPE	CATEGORY	ELEMENT	SOURCE	TARGET	PATH
118044	cstCar - tri	Query	No Differences				
10027243	cstCar	Business Object	Item Conflict	Audit Access Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	Item Conflict	Audit Action Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	Item Conflict	Do Audit Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	Item Conflict	Do Audit Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	New Source Item	Map	General.cstCurrentLocStTX	Business Object	
10027243	cstCar	Business Object	New Source Item	Map	General.cstCurrentLocCityTX	Business Object	
10027243	cstCar	Business Object	Item Conflict	Do Audit Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	Item Conflict	Do Audit Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	Item Conflict	Do Audit Flag	FALSE	TRUE	Business Object
10027243	cstCar	Business Object	New Source Item	Field	triCityTX		Business Object

© Copyright IBM Corporation 2013

10

### Comparison report

A **Comparison Report** compares every object in the uploaded package with objects in the target system. The result is a tab-delimited text file.

Each row in the comparison Report represents one difference or conflict between the uploaded package and the target system. For example, if a business object has 10 differences, there are 10 lines in the Comparison Report, one for each difference.



**Note:** Object Migration performs only a partial comparison for workflows. Object Migration compares only the header information of workflows, not the tasks. Do not be falsely assured by a **No Differences** comparison result.

Because the file is tab-delimited text, it is difficult to read in its raw form. It is much easier to read when opened in a spreadsheet tool. An example of a file opened in a spreadsheet is shown in the slide.

To create a comparison report, use the following steps:

1. Click **Comparison Report** at the top of the Object Migration page.
2. When prompted whether you want to wait for the report or have the report run in the background, click either **Wait** or **Background**.
  - If you select Wait, the system generates the file while you wait, and then prompts you to download it. Click **Save**. Select the location for the report, and click **Save**.
  - If you select Background, the file is generated in the background and saved in the **userfiles\ObjectMigration** directory of the IBM TRIRIGA installation.

# Import

## Click Import

The import process begins in the background

Please wait - the Package is being Imported. A notification will be sent once the process is complete.

When the import is complete, a notification is sent

Importing package [Grades 10.0] is complete.  
[Click here to Open Package.](#)

© Copyright IBM Corporation 2013

11

### Import

After you validate a package and there are no invalid (red) objects, you are ready to import. The import might take some time. Rather than forcing you to wait for the import to finish, the platform puts a message in the Objects panel that the import is started. When the import finishes, you receive a notification in your portal home page.

To import the migration package, click **Import**. The following changes occur:

- The display in the Objects panel changes to a notice about the import.
- The system sends you a notification that the import is complete.
- The status also shows on the Object Migration window in the Import Packages section.

### Import process

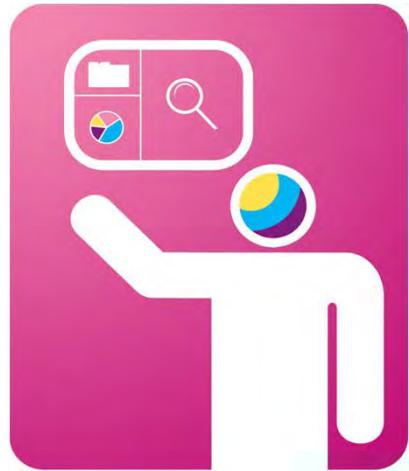
Before the import starts, the system revalidates all of the objects in the package. The system can then ensure that it is always using the most current snapshot of what is contained within the system. Getting a current snapshot is important, especially if a package is uploaded several days before it is imported. During this lag time, anything can happen. For example, objects that are

identified as new within a package might be manually created or imported as part of a different package. Without a new snapshot, the import process might try to re-create the same object.

The revalidation is one of the most important stages of the entire process. This import validation is the same validation process that can be triggered by the user with the Validate action. After this revalidation, if any errors are encountered, the import stops and the package status changes to Import Failed. To help resolve any circular dependencies or references, the system automatically imports the package two times. After each pass, the system refreshes all system caches on the environment in which the import is being performed.

## Instructor demonstration

- Import package creation
- Validate import package
- Import package comparison report
- Import objects from file



© Copyright IBM Corporation 2013

12

### *Instructor demonstration*

Guide the students through the following features of the Object Migration tool: Navigation panel; Colors used for the object names and what they mean; Comparison report; Summary report

## Student exercises



© Copyright IBM Corporation 2013

13

### *Student exercises*

Perform the exercises for this unit.

## Checkpoint questions

1. What does it mean if an object is shown in blue on the object migration window? What do you need to do about it?
2. What does it mean if an object is shown in red on the object migration window? What do you need to do about it?
3. When the validation process compares workflows, how does it do that? What are the implications?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.

## Checkpoint answers

1. What does it mean if an object is shown in blue on the object migration screen? What do you need to do about it?

*The object has a match in the target environment. You need to determine whether the target object has changes that will be lost when the object in the migration package is imported.*

2. What does it mean if an object is shown in red on the object migration screen? What are the implications?

*The object has dependents that are not in the package or in the target environment. It must be resolved or the package cannot be imported.*

3. When the validation process compares workflows, how does it do that? What are the implications?

*It means that the properties of both workflows match. Tasks are not compared, so there might still be differences.*

## Summary

---

Now that you have completed this unit, you should be able to import an object migration package





# 12 Associations

---



# 12 Associations



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

**What this unit is about**

Associations in the IBM TRIRIGA Application Platform connect records to make them more useful. This unit shows you what they are and how to create them.

**How you check your progress**

You can check your progress in the following ways:

- Review questions
- Lab exercises

**References**

*Application Building for the IBM TRIRIGA Application Platform*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Describe business object and record-level associations
- Create associations in the Data Modeler
- Create associations in the Association Manager
- Use the Associations tab of a form

# Lesson 1. Creating and using associations



## Lesson 1: Creating and using associations



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Associations in the IBM TRIRIGA Application Platform connect records to make them more useful. This lesson shows you what they are and how to create them.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe business object and record-level associations
- Create associations in the Data Modeler
- Create associations in the Association Manager
- Use the Associations tab of a form

## References

*Application Building for the IBM TRIRIGA Application Platform*

## Associations

- Records are most useful when connected to other records
- In the platform, an association is a connection made between records by using a specific string that describes their relationship



- Associations are typically used in these places:
  - Locator fields and smart sections
  - Workflows
  - Reports and queries

© Copyright IBM Corporation 2013

4

### Associations

Most records are of limited use 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 course materials that are required 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. In the IBM TRIRIGA Application Platform, an association is both a link between two records and a name that describes the relationship between the records.

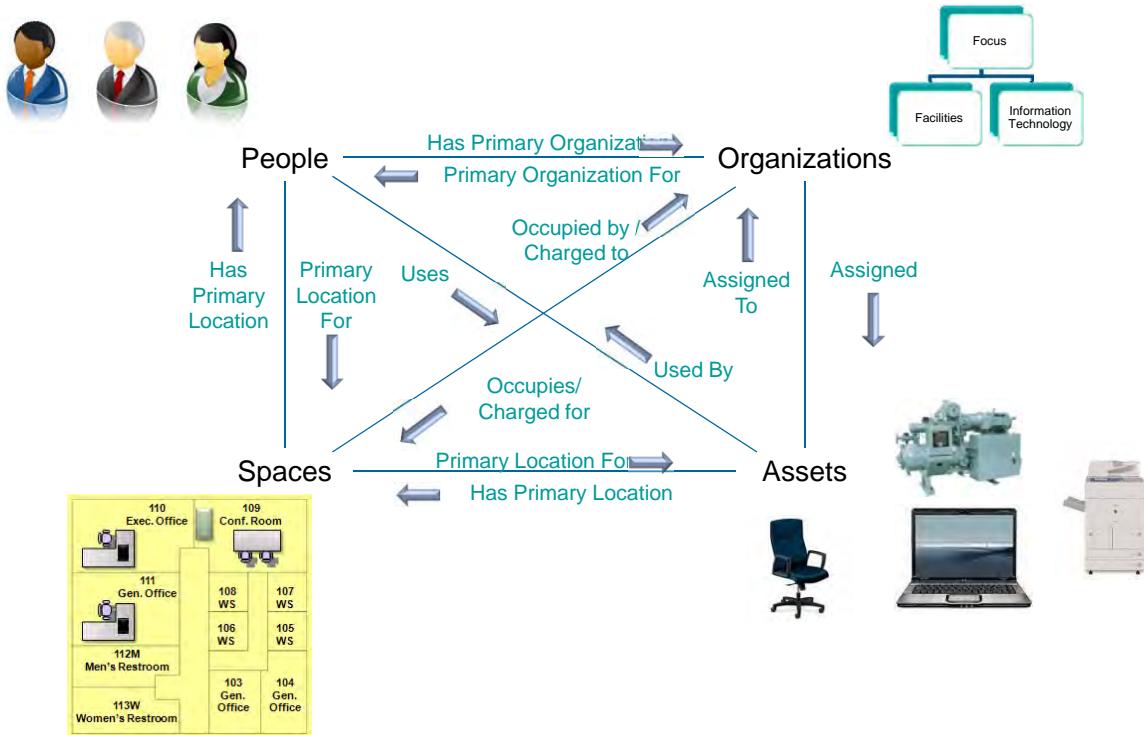
Each end of the association has a name to identify the association from that end. Looking at the diagram in the slide from one direction, you see **Employee Belongs To Organization**. Looking at the diagram from the other direction, you see **Organization Has Employee**. The platform can use an association between two records to navigate from one of the records to the other.

You use associations primarily under the following conditions:

- Population of smart sections to display information from associated records

- Population of locator fields (similar to smart sections)
- By workflows
- In queries
- In reports

## Examples of associations



© Copyright IBM Corporation 2013

5

### Examples of associations

This slide shows an example of the associations that are used in the IBM TRIRIGA Portfolio Manager application. Organization information is required to track and manage internal and external business groups. These groups can occupy space, and might be charged for the space that they occupy and a portion of the operating costs of the building. People belong to internal and external business groups (organizations) and can occupy space. Assets are assigned to people and organizations and are in an assigned location or space.

## Types of associations in IBM TRIRIGA

### Business object-level associations

- Association definitions, not actual connections
- Require strings in the Association Types list
- You can define them in Association Manager or Data Modeler

### Record-level associations

- Actual connections
- Require an explicit action to create

© Copyright IBM Corporation 2013

6

#### Types of associations in IBM TRIRIGA

The two types of associations in IBM TRIRIGA are associations at the business object level and associations at the record level. Associations at the business object level are *metadata* associations. They are the rules that define the types of records that can become associated. Business object-level associations are also called **association definitions**.

Record-level associations are *data* associations. They are the actual connections between records. Generally, you can create an association between records only if there is a business object association that allows it.

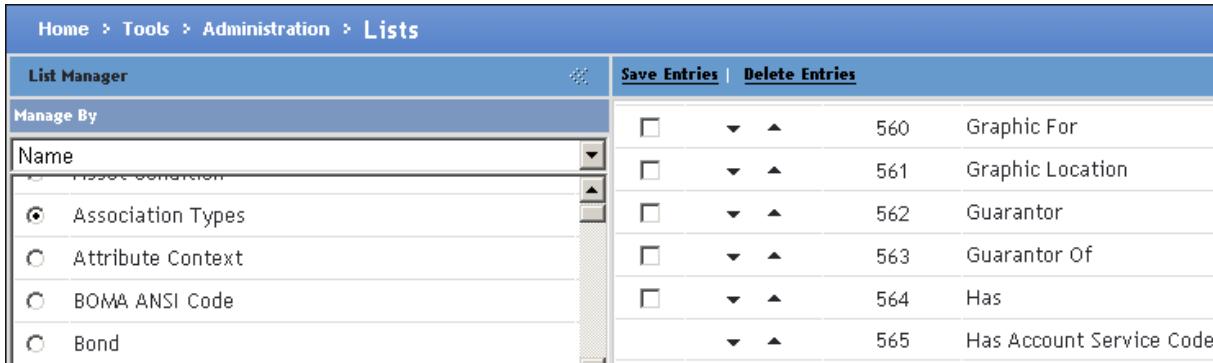
Association definitions include an association name that is connected with each end of the association definition. Association names are often referred to as the *forward association* and *reverse association* strings. For example, say that you create an association definition named *Belongs To* from the Course Section business object to the Course business object. You then create an association definition named *Has* in the other direction. You are now able to create corresponding associations between course section and course records.

Before you create a business object-level association, ensure that the list of association types includes the names that you need for the association. Then, use either the Association Manager or the Data Modeler to create the association definition.

## Association Types list

Strings used for associations must be defined in advance

- Stored in the Association Types list
- Accessed by clicking **Tools > Administration > Lists**



The screenshot shows the List Manager interface with the following details:

- Header:** Home > Tools > Administration > Lists
- Left Panel (List Manager):**
  - Manage By:** Set to "Name".
  - Options:** Includes "Insert Condition" and several radio buttons for "Association Types", "Attribute Context", "BOMA ANSI Code", and "Bond".
- Right Panel (List View):**
  - Buttons:** Save Entries | Delete Entries
  - Table Headers:** Checkmark, Down arrow, Up arrow, ID, Name
  - Table Data:**

<input type="checkbox"/>	▼	▲	560	Graphic For
<input type="checkbox"/>	▼	▲	561	Graphic Location
<input type="checkbox"/>	▼	▲	562	Guarantor
<input type="checkbox"/>	▼	▲	563	Guarantor Of
<input type="checkbox"/>	▼	▲	564	Has
	▼	▲	565	Has Account Service Code

© Copyright IBM Corporation 2013

7

### Association Types list

The names that are used for associations are stored in the **Association Types** list. Ensure that the names you want to use are present in the list before you create association definitions.

To ensure that the names that are required for associations are in the list of association types, click **Tools > Administration > Lists** to open the List Manager. Select the Association Types list. The list is large so you might notice a delay while it loads.

When the list is displayed, look for names in the list suitable for describing the new 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 already in the list to describe associations, you can either use a name with a similar meaning that is already in the list, or add a name to the list.

After all the names you want to use for describing associations are in the list, you can define the associations. Business object-level associations can be defined in either the Association Manager or the Data Modeler.

## Viewing associations in the Association Manager

- A platform tool for managing association definitions
- Associations originating from selected module are listed
- Accessed by clicking **Tools > Builder Tools > Association Manager**



The screenshot shows the 'Associate Business Object List' interface. On the left is a sidebar titled 'Module' with a tree view containing various business objects like Classification, ctTesting, Data Utilities, Document, Geography, Group, Location, Locator Directions, Mail, Organization, and Polymorphic Standard. The main area displays a table with columns: Source BO, Secondary BO, Condition ID, Module module, and Reverse module. The table lists 10 associations, all of which are of type 'Classification'. The associations are:

Source BO	Secondary BO	Condition ID	Module module	Reverse module
Classification	Auto Recorded By	triPeople	triPeople	Auto Recorded
Classification	Classifies	triClimateZone	Classification	Classified By
Classification	Classifies	triREPaymentAdjustment	triLog	Classified By Payment Type
Classification	Created	triAsset	triAsset	Created From
Classification	Found	triAssetFound	triLog	Has Found By
Classification	Geography Contains	triProposedSite	Location	Belongs To Geography
Classification	Has Notification	triNotificationDetails	triRouting	Is Notification For
Classification	Is Contained By	triCapitalProject	triProject	Contains
Classification	Is Open Year For	triProperty	Location	Has Open Year
Classification	Is Parent Of	triCarbonSinkType	Classification	Is Child Of

At the bottom of the interface, there are buttons for 'Add' and 'Edit', and a copyright notice: © Copyright IBM Corporation 2013.

### *Viewing associations in the Association Manager*

You use the Association Manager tool for managing business object-level association definitions. Each entry in the Association Manager tool defines a possible association that can be created between records. The reverse association from the perspective of the secondary business object is typically also defined during design.

When you create a business object-level association definition, the records are not necessarily automatically associated. However, if a record-level association is created that has a corresponding association definition, then the association definition dictates which record-level reverse association is created.

Do not use the Association Manager to create *Include* associations (Is Parent Of/Is Child Of associations that define the hierarchy rules between hierarchical business objects). You must create these special associations within the Data Modeler.

To see the reverse associations, you must select the associated module and business object and see which associations exist from that starting point. There is no place in the platform where you can see all associations that point to a specific module and business object.

Clicking a column header sorts the display by that column.

You can also see associations that are listed in the Association Manager when you use the Data Modeler.

## Creating associations in the Association Manager

Associate Business Object	
Module	Classification
Business Object	Classification
Association	About
Associate Module	Classification
Associate Business Object	Classification
Reverse Association	About
Dependent Flag	<input type="checkbox"/>

© Copyright IBM Corporation 2013

9

### *Creating associations in the Association Manager*

Clicking **Add** in the Association Manager opens the Associate Business Object window. The Associate Business Object window is used to define an association between two business objects, the primary business object and the associated, or secondary, business object. Enter the properties of the association definition and click **OK**.

The properties of a business object association definition are as follows:

- **Module:** The module that contains the primary business object. It is set to the selected module and cannot be changed. Because of this limitation, ensure that you select the module of the primary business object in the Association Manager before you click Add.
- **Business Object:** The primary business object. It is at the *primary* end of the association, and the association is considered to originate with this business object.
- **Association:** The string that defines the association from the primary business object to the associated business object.
- **Associate Module:** The module that contains the secondary business object.

- **Associated Business Object:** The secondary business object. It is in the module that is named by the Associate Module property, and is considered to be the target of this association.
- **Reverse Association:** The string that defines the association from the secondary business object to the primary business object. Even though this string is named in this association, the platform does not automatically create the association in the reverse direction. The reverse association must be created manually.
- **Dependent Flag:** This flag indicates that the existence of records from the secondary business object depends on the existence of the record from the primary business object. Deleting the record from the primary business object cascades and results in the deletion of all of the dependent records. But the reverse is not true: deleting a record from the secondary business object has no effect on records from the primary business object.

For example, you might have line items that are dependent on a purchase order. You create an association definition named *Has* from the Purchase Order business object to the PO Line Item business object. In this association, the Dependent flag check box is selected. You define the reverse association named *Belongs To* from the PO Line Item business object to the Purchase Order business object. In this association, the Dependent flag check box is not selected. With this setup, the line items are dependent on the purchase order, but the purchase order is not dependent on the line items.

The association definition is a one-way definition, from the primary business object to the secondary business object. The platform does not automatically create the reverse association definition, even though the definition properties include both the forward and reverse association strings. You must manually create the reverse association definition if you want it. You do not have to define associations in both directions but it is good practice to do so.

You might wonder why an association definition includes the reverse association string. When a record-level association is created, the platform automatically creates the reverse record-level association, if there is a reverse association definition. However, multiple associations can be defined between two business objects with different strings. The platform uses the reverse association string from the forward association to find the correct reverse association definition. It uses the properties from the reverse association definition to create the reverse record-level association.

As an example, assume that three associations are defined from business object A to business object B. The strings that are used in the associations are shown in the following table.

Forward association string	Reverse association string
Has current location	Current location of
Has primary location	Primary location of
Has authorized location	Authorized location of

You define only two associations from business object B to business object A by using the strings that are shown in the following list.

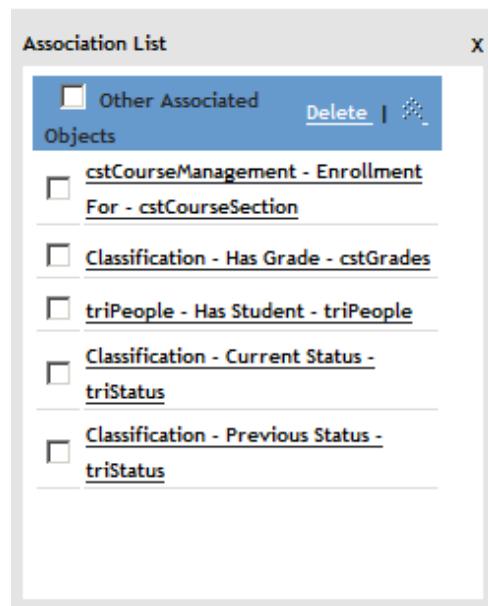
Forward association string	Reverse association string
Current location of	Has current location
Primary location of	Has primary location

Now, assume that a record-level association is created with the **Has current location** association definition. The platform uses the reverse association string **Current location of** to search for a matching association definition. There is a match, and the platform uses that association definition to create the reverse record-level association.

What happens when a record-level association is created with the **Has authorized location** association definition? The platform uses the reverse association string **Authorized location for** to search for a matching association definition. There is no match, and the reverse record-level association is not created.

## Viewing associations in the Data Modeler

- Click **View > Association List** to open the Association List window
- Associations which originate from selected business object are listed
- Associations listed here are also listed in the Association Manager



© Copyright IBM Corporation 2013

10

### *Viewing associations in the Data Modeler*

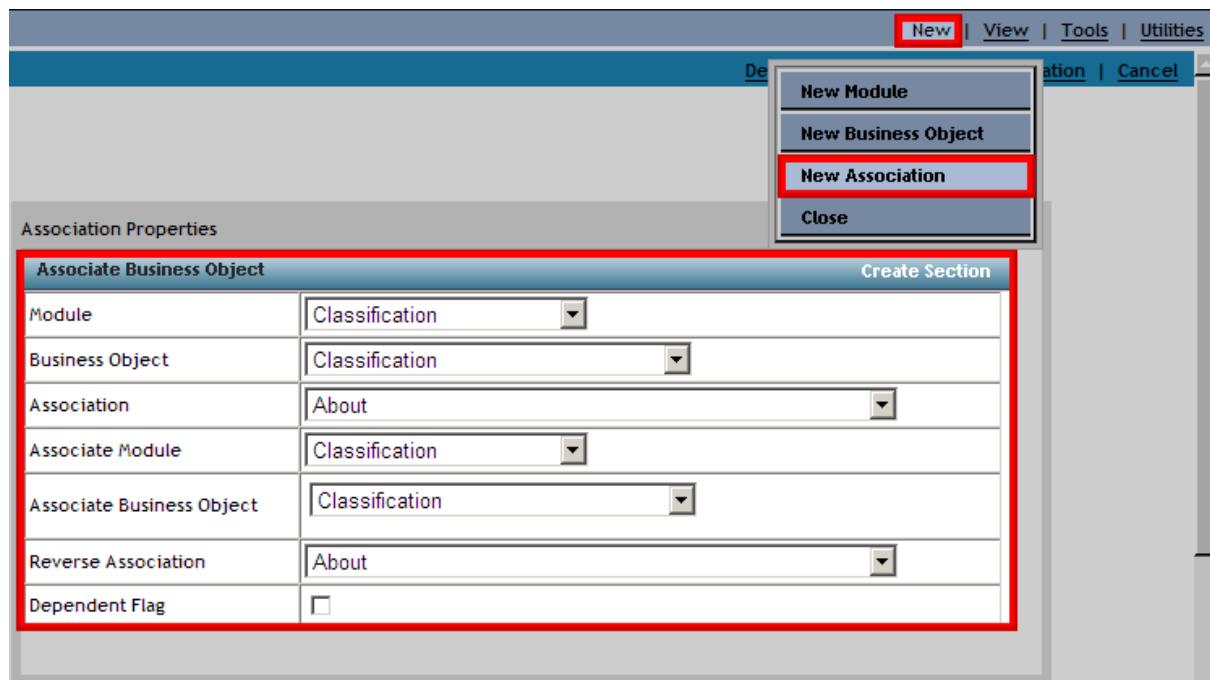
You can view and create associations in the Data Modeler. Open a business object and click **View > Association List** to see them.

All association definitions that are shown in the Association List panel originate from the selected business object. The following information is displayed for each association definition:

- The name of the module that contains the target business object
- The name of the association on the side of the originating business object
- The name of the target business object

You can click an association to see the full details of it. You can also see these associations in the Association Manager.

## Creating associations in the Data Modeler



© Copyright IBM Corporation 2013

11

### *Creating associations in the Data Modeler*

To add an association in the Data Modeler, click **New > New Association**. The Association Properties panel opens, and the Save Association action is displayed. The same properties are contained in this panel as in the Association Properties panel in the Association Manager. Notice the ability to specify the module when you define associations in the Data Modeler, as compared to the Association Manager. This ability can be helpful when you create association definitions that originate from a different module and business object. After you specify the properties of a new association definition, you create the association by clicking the **Save Association** action.



**Important:** When you create an association in the Data Modeler that originates from a different business object than the one that has focus, the Association List panel displays the associations that originate from the other business object. This change of focus can be confusing if you are not aware of this behavior. Select the original business object from the Object Browser to return to the correct Association List.

To edit the properties of an existing association definition, click the association definition in the Association List panel. When you are finished editing the properties of the association definition, click the **Save Association** action to save the edits.

## Record-level associations

- In most cases, you must define the BO-level association first
- Many mechanisms create associations between records:
  - Smart sections
  - Locator fields
  - Classification fields
  - Workflows
  - Associations tab
  - Data Integrator

© Copyright IBM Corporation 2013

12

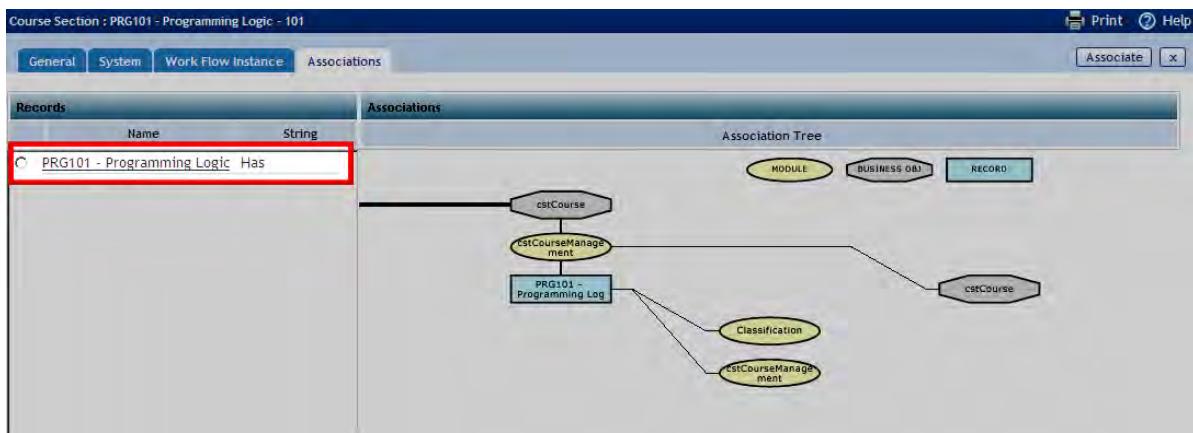
### Record-level associations

A person cannot explicitly create an association between records unless there is an association that is defined at the business object level that allows it. There are many mechanisms that create associations between records, including the following ones:

- **Smart sections:** An association is created when a record is selected by using Find.
- **Locator fields:** An association is created when a record is selected by using the picker.
- **Classification fields:** An association is created when a classification value is put into the field.
- **Workflows:** The Associate, Create Record, and Modify Record tasks can all create associations.
- **Associations tab:** The Associate action can be used to manually create associations.
- **Data Integrator:** Associations can be formed by using the Associate feature.

## Associations tab

- Select **Show Association** in Form properties to see the association
- Use it to graphically navigate the associations of a record
- Use it to manipulate associations manually



© Copyright IBM Corporation 2013

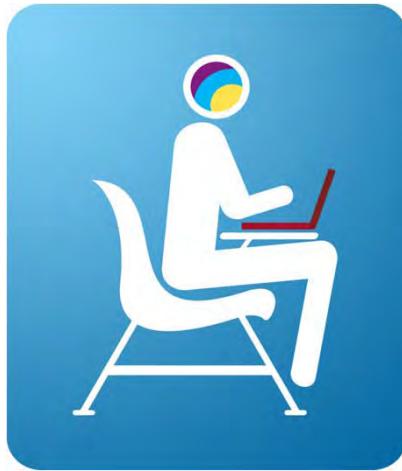
13

### Associations tab

The Association tab is a runtime feature of a form. It graphically shows the associations of a record with other records. You can manually create and remove associations between records, but this task is discouraged.

The Association tab is automatically defined by the platform, if Association is selected in the properties of the form. When opened, the tab shows the module for each association to the current record. Clicking a module expands the diagram to show the business object. Clicking the business object shows a link to each record that is a target of the association. Clicking the record link opens the record. Access to this tab can be controlled through Security Group permissions.

## Student exercises



© Copyright IBM Corporation 2013

14

### *Student exercises*

Perform the exercises for this unit.



## Checkpoint questions

---

1. What do associations represent?
2. What is the difference between BO-level associations and record-level associations?
3. Where are the strings used for associations defined?
4. Name two places where BO-level associations can be defined. What is the difference between them?
5. Where can you view record-level associations?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. What do associations represent?

*Relationships between data.*

2. What is the difference between BO-level associations and Record-level associations?

*BO-level associations are potential associations.*

*Record-level associations are actual relationships between records.*

3. Where are the strings used for associations defined?

*In the Association Types list.*

## **Checkpoint answers (continued)**

4. Name two places where BO-level associations can be defined. What is the difference between them?

*In the Association Manager and the Data Modeler.*

*The Data Modeler allows the Module to be changed in the Association properties window.*

5. Where can you view record-level associations?

*On the Association tab of a record.*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Describe business object and record-level associations
- Create associations in the Data Modeler
- Create associations in the Association Manager
- Use the Associations tab of a form





## 13 Smart sections and locator fields



## 13 Smart sections and locator fields



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

Smart sections and locator fields are mechanisms in the IBM TRIRIGA Application Platform that allow direct connections between records. This unit describes them and shows you how to create them.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building for the IBM TRIRIGA Application Platform*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Create a smart section
- Create a locator field
- Add a smart section to a business object
- Add a smart section to a form

# Lesson 1. Using smart sections and locator fields



## Lesson 1: Using smart sections and locator fields



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Smart sections and locator fields are mechanisms in the IBM TRIRIGA Application Platform that allow direct connections between records. In this lesson, you learn how to create them.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a smart section.
- Create a locator field.

## References

*Application Building for the IBM TRIRIGA Application Platform*

## Locating records

- Associations are an indirect way to find records
  - Platform must search association table to find them
- For a direct connection, you can use locator fields or smart sections
  - They contain a reference to other records
  - They are based on BO-level associations

© Copyright IBM Corporation 2013

4

### *Locating records*

An association is an indirect connection between records. The platform must search through the associations to navigate from one record to the other. The platform also offers direct connections to records through smart sections and locator fields. Each of these stores a reference to another record so that the platform can access the referenced record directly without searching for it. Smart sections and locator fields are based on business object association definitions.

## Smart sections

- Associate another record to the current record
  - Create a BO section, with fields, to store data from the associated record
  - Store a reference to the associated record in the section
  - Offer the ability to modify the stored data
  - Offer Live Link to retrieve updates from the associated record
- Can be single-record sections or multiple-record sections

© Copyright IBM Corporation 2013

5

### Smart sections

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

In addition to containing a *General* section, records can also contain smart sections. Each smart section is based on an association between the record that contains the smart section and the record that is referenced by the smart section. Each section has a name and can reference fields in one or more other records, depending on the smart section properties.

A smart section contains fields for each record that it references. The fields in a smart section contain either a reference or a copy of the value in the corresponding field of the referenced record.

There are two fundamental varieties of smart sections:

- A single-record smart section can reference one record or none. The presentation of fields in a single-record smart section is form-like. There is a label next to each field and fields that are 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 multiple-record smart section in a record can reference any number of other records. The fields in a multiple-record smart section are presented in a table-like way. The fields are arranged in columns, and each record is in a different row. A smart section can also 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 references one more record than it did before.



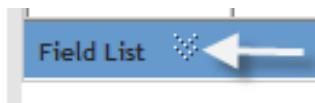
**Note:** Most multiple-record smart sections are now query sections, which are more flexible.

## Creating a smart section

1. In the Data Modeler, open an association
2. At the top of the panel, click **Create Section**

Classification - Has Category -  
cstCarCategory

3. Specify the properties of the smart section
4. Click **Field List**



5. Select fields for the smart section
6. Click **OK** to close the field list
7. Click **Save Section**

| Save Section | Cancel

© Copyright IBM Corporation 2013

6

### Creating a smart section

To create a smart section, follow these steps:

1. In the Data Modeler, choose an association in the Association List. Click the name of the association to open it.

The Association Properties are displayed. In the Association Properties panel is a heading named Associate Business Object. It has a link named **Create Section**.

2. Click the **Create Section** link to open the section properties.



**Note:** If the Create Section link is not visible, there are two possible solutions:

- If the link is not seen, the business object is in a published state. Revise the BO and look again.
- If the panel is too narrow, the link is not visible. Widen the panel to see the link.

3. Specify the properties of the Smart Section, as shown in [Figure , "Smart section properties,"](#) on page 272.
4. Click the chevron (double downward arrow) to open the Smart Section Field List.



**Note:** Look closely for the chevron. It is difficult to see.

5. Select fields for the smart section.
6. Click **OK** to close the field list.
7. Click **Save Section**.

## Smart section properties

Section Properties	
Section Details	
» Section Name	<input type="text"/>
» Section Label	<input type="text"/>
Associated Business Object	Classification - Has Category - cstCarCategory <input type="button" value="▼"/>
Temporary Association	--Select a Temporary Association-- <input type="button" value="▼"/>
Associate One Record	<input checked="" type="radio"/>
Associate Multiple Record	<input type="radio"/>
Vertical Section	<input type="checkbox"/>
Dependent	<input type="checkbox"/>
Reference Only	<input checked="" type="radio"/>
Live Link	<input checked="" type="checkbox"/>
Reference With Modify	<input type="radio"/>
Workflow to Initialize Record	<input type="button" value="🔍"/> <input type="button" value="✖"/>
Used by DataConnect	<input type="checkbox"/>
Values Required	<input type="checkbox"/>

Field List

© Copyright IBM Corporation 2013

7

### Smart section properties

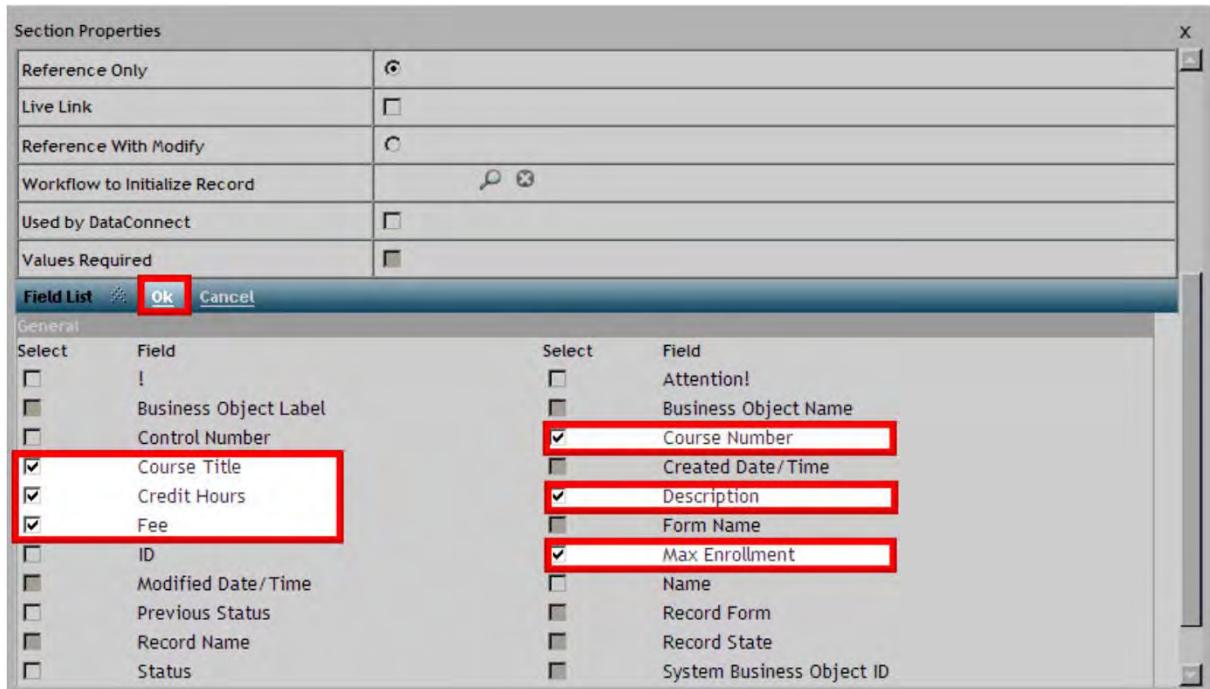
When you create a smart section, the **Associated Business Object** property is automatically set based on the association that you used to create the smart section.

You can also set other smart section properties, like the following ones:

- **Section Name:** Is the unique name of the section. It cannot contain spaces or special characters.
- **Section Label:** Is the text that is displayed at the top of the section. It can contain spaces or special characters.
- **Associate One Record:** Is used to create a single-record smart section that can reference either no records or one record. Select the **Associate Multiple Record** property to create a multiple-record smart section that can reference any number of records.
- **Live Link:** The values in the section are always pulled from the referenced record. This property ensures that the most recent data is displayed in the smart section. If you do not select this property, the values of the fields are obtained when the record is selected and they do not change.

- **Reference Only:** Is used to make the section read-only and not allow user modification of values.
- **Reference With Modify:** Is used to allow modifications. The values of fields in the underlying record are used to initialize the smart section. Live Link is turned off, and there is no further connection to the underlying record. Changes to the fields in a smart section do not affect the fields in the underlying record. Changes to the fields in the underlying record do not affect the smart section.
- **Dependent:** Deletion of the primary record also deletes the referenced record. This property is automatically enabled if the association that is used to define the section is a dependent association or if the secondary BO of the association is an embedded BO.

## Selecting smart section fields



© Copyright IBM Corporation 2013

8

### Selecting smart section fields

The Section Properties Field List shows the fields in the underlying record. Select those fields that are to be contained in the smart section and click OK. Click Save Section.

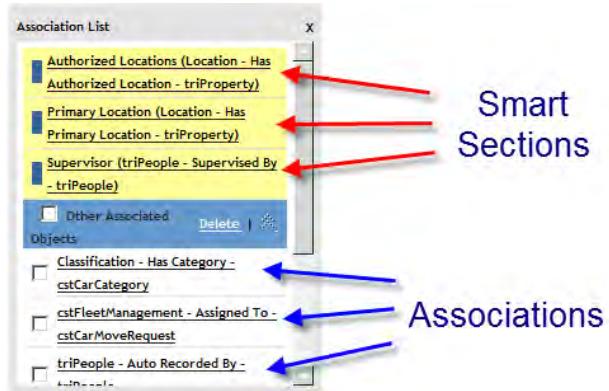
Fields are automatically selected for a smart section if they have the Result Column property enabled. Clear the check mark of any fields that you do not want in the smart section.



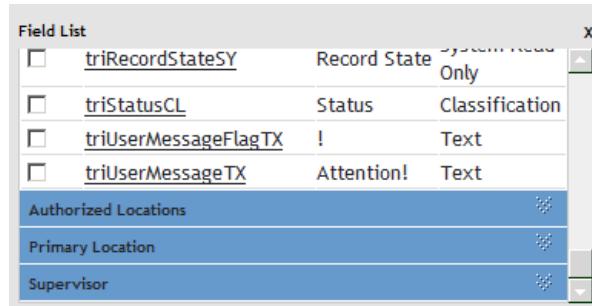
**Note:** After you save the smart section, you can add fields through this field list, but you cannot remove them.

## Results of saving a smart section

- In the Association List, smart sections are above the line and have a yellow background



- In the business object Field List, smart sections are at the bottom as separate sections



© Copyright IBM Corporation 2013

9

### Results of saving a smart section

Saving a smart section moves the entry in the Association List above the line and changes the background color to yellow. This appearance clearly distinguishes them from plain associations and from Include associations (which have a light blue background).

A smart section is based on only one association, and only one smart section can be based on each association. After a smart section is built on an association, you cannot view the association properties, only the section properties.

If a smart section is deleted, the association returns to a position 'below the line'. It can then be used again to create a smart section.

## **Viewing the smart section field list**

You can view smart section field lists in three ways  
Each method has different capabilities and limitations

1. Use the chevron at the bottom of the Smart Section properties panel
2. Open the section in the business object field list
3. In the Association List panel, click the blue box

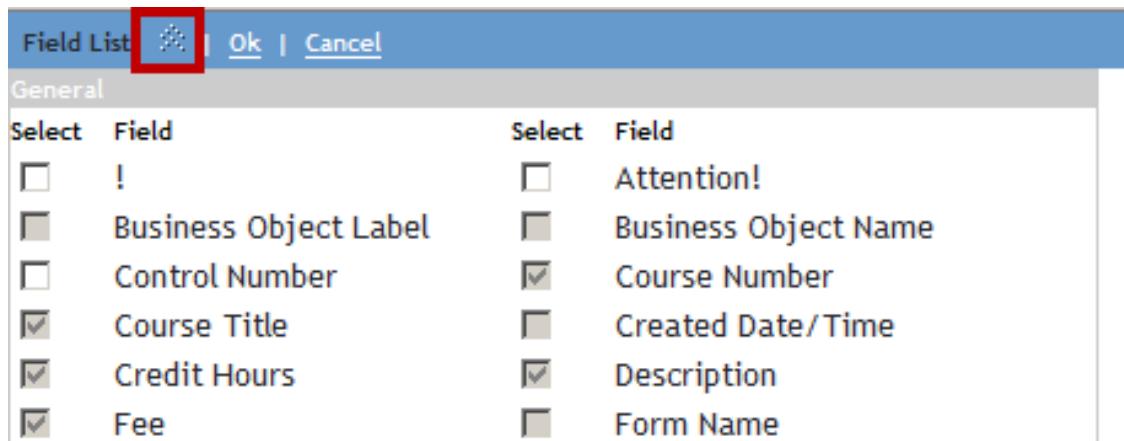
### *Viewing the smart section field list*

There are three different methods for viewing the field list of a smart section, as shown on the slide.  
Each has different capabilities and limitations. These methods are covered in the following slides.

## Using the chevron

Use the chevron at the bottom of Smart Section properties panel

- You can add fields to the smart section
- You cannot edit properties or remove fields



© Copyright IBM Corporation 2013

11

### Using the chevron

The first method for viewing the field list of a smart section is the same method that was used when the smart section was created. Click the name of the smart section in the Association List window to open its properties. Then, click the chevron at the bottom of the properties panel to open the Field List.

Fields that are already included in the smart section cannot be removed by using this Field List. Other fields can be added by using this list. You cannot access the properties of the fields from this list. If you change the fields in the smart section, click **OK**, and then click **Save Section**.

## Opening the section in the business object field list

Click chevron to open the section in the BO Field List panel

- You can edit the field properties
- You cannot add or remove fields

The screenshot shows the 'Field List' panel for a business object. At the top, there's a toolbar with 'Add | Find | Delete |' and a search icon. Below it, a section header 'Classroom' is expanded, indicated by a blue chevron icon with a red box around it. The main area is a table with three columns: 'Field Name', 'Field Label', and 'Field Type'. The rows show four fields: 'triCapacityNU' (Capacity, Number), 'triNameTX' (Name, Text), 'triPathTX' (Hierarchy Path, Text), and 'triStatusCL' (Status, Classification). At the bottom of the table, there's another blue chevron icon with a red box around it, indicating that the 'Course' section is collapsed.

Field Name	Field Label	Field Type
triCapacityNU	Capacity	Number
triNameTX	Name	Text
triPathTX	Hierarchy Path	Text
triStatusCL	Status	Classification
Course		

© Copyright IBM Corporation 2013

12

### *Opening the section in the business object field list*

The second method for viewing the field list of a smart section is through the Field List panel of the business object. Go to the bottom of the field list for the business object where there is a heading for each smart section. Click the chevron of any section heading to open its field list.

Fields cannot be added to or removed from smart sections by using this field list. There are links for such actions at the top of the panel, but these actions are for the General section only.

Click the name of any field to open it in the Properties panel. If you change the properties of the field, click **Save Field**.

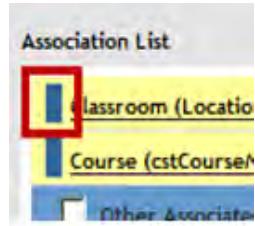


**Note:** Only one section in the Field List panel can be open at any time.

## Using the Association List panel

In the Association List panel, click the blue box

- You can edit the field properties
- You can delete unwanted fields
- You cannot add fields to the list



Fields	Delete
triCapacityNU (Number)	
triNameTX (Text)	
triPathTX (Text)	
triStatusCL (Classification)	

© Copyright IBM Corporation 2013

13

### Using the Association List panel

The third method for viewing the Field List of a smart section is through the Association list. The small blue rectangle next to a smart section in the Association List panel is a link. Using it opens the Field List of the smart section.

Fields can be deleted from the smart section by using the Delete link in this Field List, but *only* if the field is not in use. For example, if the smart section is used on a form, you must delete the field from the form before you can delete it from the smart section. Fields cannot be added to the smart section by using this field list.

Click the name of any field to open it in the Properties panel. If you change the properties of the field, click **Save Field**.

## Single-record smart section on a form

- You can add a smart section to a tab from components list
- You can lay out the fields in the section as needed
- Field labels are hyperlinks to the associated record; you can disable this option
- You use a **Find** action to locate an associated record

Requested By		<a href="#">Use Car Supervisor</a>   <a href="#">Find</a>   <a href="#">Clear</a>	
<u>Name</u>	Don Iverson	<u>Work Phone</u>	
<u>Title</u>		<u>State/Province</u>	Nevada
<u>City</u>	Las Vegas		

© Copyright IBM Corporation 2013

14

### Single-record smart section on a form

In the Form Builder, you can add a smart section to a tab from the Components list. Adding it this way includes all of the fields in the section. You can also create a smart section on a tab by using the **Add Section** action and specifying Smart for the section type. Adding a section in this manner requires that you select the fields for the section individually.

Fields can be laid out in a single-record smart section as needed by using the field positioning properties: Start Row, Row Span, Start Column, Col Span. All field labels in a single-record smart section on a form are links to the associated record. Clicking any of these links opens that record. This ability can be disabled so that the user can see data from the associated record but cannot open it by using the smart section. The associated record is chosen by using a section **Find** action.

## Locator fields

- Special type of text field
- Stores a reference to, and a value from, an associated record
- Can capture extra values from associated record
- Does not get updates from associated record (no live link)

Locator Field	<input checked="" type="checkbox"/>
Locator Module	Location <input type="button" value="▼"/>
Locate Using	<a href="#">triProperty - Name</a> <a href="#">Edit Mapping</a>
Locator String	Has Authorized Locations

© Copyright IBM Corporation 2013

15

### Locator fields

A locator is another mechanism that is used to reference a record in the IBM TRIRIGA platform. A locator field is like a single-record smart section, except that it is a single field. Only a text field can be made into a locator field. A locator field inherits a single value from the referenced record. It also creates an association from the record that contains the locator field to the referenced record. Using Locator Mapping, you can configure locator fields to copy field values from the referenced record to fields in the record that contains the locator field.

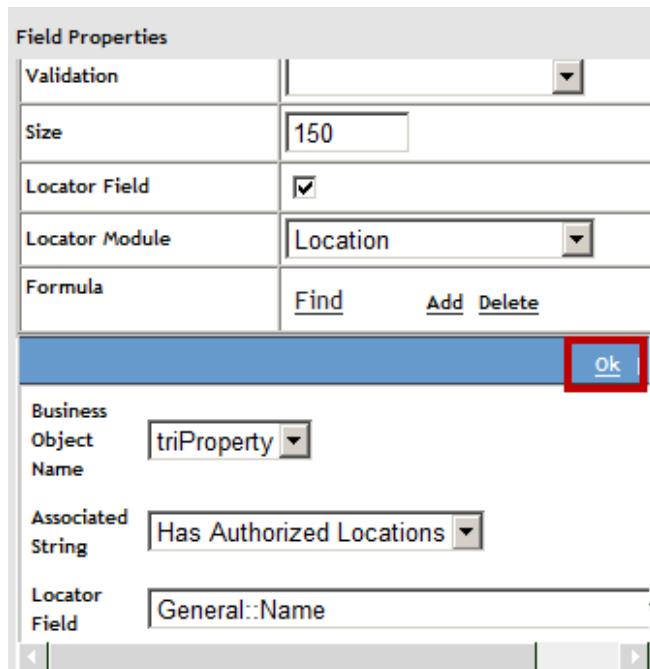
### Locator field versus smart section

For a single record smart section, you can provide a live link to the referenced record. You do not have this option for a locator field. A locator field receives a copy of the value from the referenced record when the connection is made. If the field value later changes in the referenced record, the updated value is not reflected in the locator field.

Smart sections are limited to a connection to a single business object. A locator field has an ability called *overloading* that enables it to work with different business objects. Overloading is used when working with intermediate objects, such as those found in the triIntermediateObject module.

## Creating a locator field

1. Open a text field
2. Select the **Locator Field** check box
3. In Locator Module field, select the module of the target business object
4. Select values for the **Business Object Name, Association String, and Locator Field**
5. Click **Ok**
6. Click **Save Field**



© Copyright IBM Corporation 2013

16

### *Creating a locator field*

A text field is designated to be a locator field when you select its Locator Field property. The locator field must store a field value from the referenced record. Typically, you want to choose a field whose value is unique, or at least sufficiently descriptive to identify the referenced record. Common field choices that provide a unique value are the Record Name field for standard business objects and the Hierarchy Path field for hierarchical business objects. Before you can create a locator field, an association must exist between the referenced BO and the BO that contains the field.

Follow these steps to create a locator field:

1. In the Data Modeler, open the properties of a text field.
2. Select the **Locator Field** check box.  
The Locator Module property is displayed.
3. In the **Locator Module** property, select the module of the associated business object as specified in the association.  
A panel is displayed below the field properties.
4. Specify the referenced business object.

5. Select the association string that is used to define the association.
6. In the **Locator Field** property, select the field from the referenced business object whose value you want to copy into the locator field.
7. Click **OK**.

## Locator field mapping

- After you save the field, the Edit Mapping link is displayed
- Click the link to open the Locator Mapping panel
- Use the picker to select fields from associated business object to be mapped into fields in current business object

Locator Field	<input checked="" type="checkbox"/>
Locator Module	Location
Locate Using	triProperty - Name
Locator String	<a href="#">Edit Mapping</a>

Locator Mapping

General::Car	Category	General::City	General::City
<a href="#">General::Car</a>	<a href="#">Category</a>	<a href="#">General::City</a>	<a href="#">General::City</a>
<a href="#">General::Control</a>	<a href="#">Number</a>	<a href="#">General::Control</a>	<a href="#">Number</a>

### Locator field mapping

You use Locator Mapping to configure the locator field to copy other field values from the referenced record when the locator field is populated. Click the Edit Mapping link to set the mapping of other field values. The Edit Mapping link is seen only after the field is saved.

Follow these steps to define the mapping:

1. Open the properties of a text field.
2. Select the **Locator Field** property, and define the properties of the locator field.
3. Click **Save Field**. The Edit Mapping link is now visible.
4. Click it and define the mapping of other fields.

There are three columns in the Locator Mapping panel. On the left is the list of fields from the BO containing the locator field. Next to each of these fields is a field picker. On the right are any fields from the associated BO that are mapped into fields in the original BO.

To map a field, click the field picker icon next to the field. Choose the field from the associated business object whose value you want to copy into the mapped field. When you finish mapping fields, click OK and click **Save Field** to save the field mappings.

## Locator field on a form

- Field is added to form like any General section field
- Field label is a hyperlink to an associated record
  - Only for the Locator field itself, not extra fields
  - You cannot disable this field
- Use the picker (looks like a magnifying glass) to select an associated record

Current Location [McCarran Airport Rental Center](#)  

City	Las Vegas	State / Province	Nevada
------	-----------	------------------	--------

© Copyright IBM Corporation 2013

18

### Locator field on a form

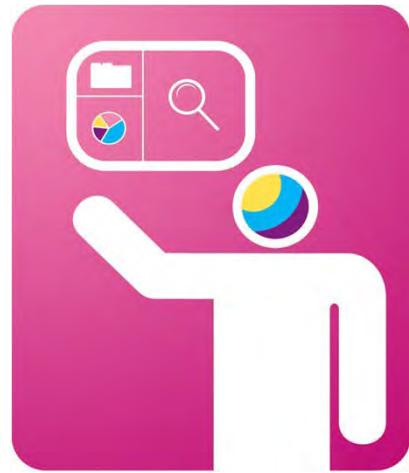
A locator field is like any other general section field when it is added to a form. It can be put in any section where other general section fields can be placed. The same applies to any other fields that are mapped by the locator field.

The label for the locator field is a link to the associated record. Click the label to open that record. This behavior is true only for the locator field itself. Any other fields that are mapped by the locator field are not links to the associated record because the reference to the associated record is contained in the locator field itself, unlike fields in a smart section. This link cannot be disabled.

Records are selected by using a picker, which looks like a magnifying glass. The picker is a runtime feature. It is not seen in the layout panel of a form, nor in the preview of a form.

## Instructor demonstration

- Smart sections
- Locator fields
- Live link

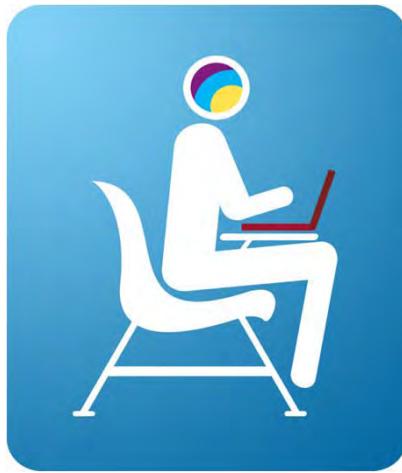


© Copyright IBM Corporation 2013

19

*Instructor demonstration*

## Student exercises



© Copyright IBM Corporation 2013

20

### *Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

- 1.** Why would you prefer to use smart sections and locator fields to find records instead of associations?
  
- 2.** What does Live Link do?
  
- 3.** How many smart sections can be built on an association?
  
- 4.** A developer plans to set up a Locator field that is based on a Number field. What advice would you give?
  
- 5.** You plan to show information from another record in a form and do not want the fields to be used as links to the record. What has that ability, a Locator field or a smart section?

© Copyright IBM Corporation 2013

21

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. Why would you prefer to use smart sections and locator fields to find records instead of associations?

*Speed, and the ability to display information from the associated record.*

2. What does Live Link do?

*Retrieves updates from the associated record when the originating record is opened or saved.*

3. How many smart sections can be built on an association?

*One.*

## **Checkpoint answers (continued)**

4. A developer plans to set up a Locator field that is based on a Number field. What advice would you give?

*Choose a different field. Only text fields can be used as locators.*

5. You plan to show information from another record in a form and do not want the fields to be used as links to the record. What has that ability, a Locator field or a smart section?

*Smart section. Locator fields do not offer this ability.*

## Summary

---

Now that you have completed this chapter, you should be able to perform the following tasks:

- Define a smart section
- Define a Locator field
- Add a smart section to a business object
- Add a smart section to a form





## 14 Formulas



## 14 Formulas



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

**What this unit is about**

In this unit, you learn about the formulas that are used in the IBM TRIRIGA Application Platform to set field values and are used as conditions in workflows.

**How you check your progress**

You can check your progress in the following ways:

- Review questions
- Lab exercises

**References**

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform 3: Calculations*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Create a regular formula
- Create an extended formula in a business object
- Create an extended formula in a workflow

# Lesson 1. Using formulas



## Lesson 1: Using formulas



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013. US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Formulas are used in the IBM TRIRIGA Application Platform to set field values and use as conditions in workflows.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Create a Regular formula
- Create an Extended formula in a business object
- Create an Extended formula in a workflow

## References

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform 3: Calculations*

## Two types of formulas

### Regular formula

- Basic, limited function
- Used only in Data Modeler to define field values

### Extended formula

- Powerful and flexible
- Used to define field values in Data Modeler, workflows
- Used to set conditions in workflow tasks

© Copyright IBM Corporation 2013

4

### *Two types of formulas*

Some kinds of fields can have their value determined based on the values of other fields. In the IBM TRIRIGA product, this is done by the use of formulas.

The following two kinds of formulas are supported:

- **Regular:** These formulas are simple formulas that use the values of fields in a record to set the value of a field. You can define Regular formulas only in the properties of a field in the Data Modeler.
- **Extended:** These formulas are more complicated formulas that are supported for most field types and are more flexible than Regular formulas. You can use Extended formulas to define field values in the Data Modeler or in some workflow tasks. You can also use them to control conditional logic in workflows.

## Regular formulas

- You can use fields only from same record
- They are for simple operations
  - Concatenation for text fields
  - Add, subtract, multiply, divide for number fields
- You cannot use other formulas or queries
- You cannot use constants directly
  - Put constant in another field in record, such as triOneNU
- They are unbreakable

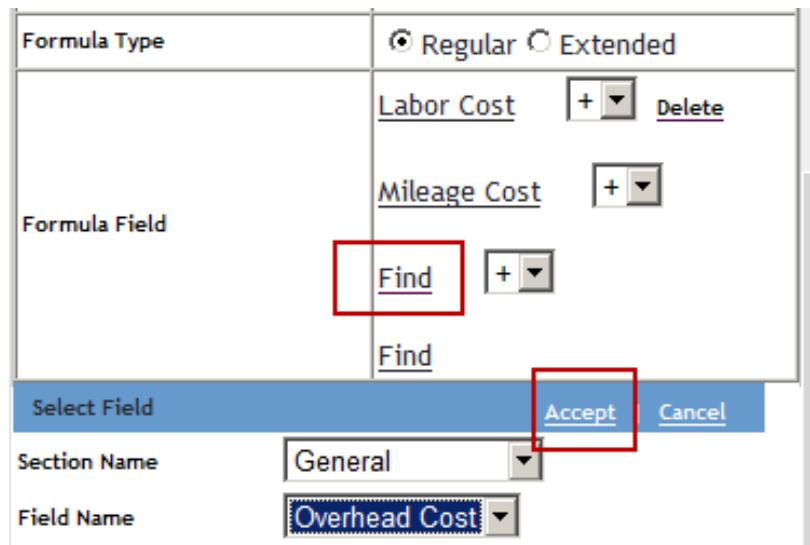
### Regular formulas

In a Regular formula, the value of the field that contains the formula is derived from the values of other fields in the same record. For example, you might use a formula to determine when a training session ends based on the time that it starts and its duration. Regular formulas are supported for text, date, date and time, duration, and number. For number fields, you can use Regular formulas for addition, subtraction, multiplication, and division. For text fields, the Regular formula is limited to concatenation.

Some formulas might require the use of a constant. For example, you might want to add **1** to the value of a field. Because Regular formulas work only with fields, not constants, you cannot put **1** directly into a Regular formula. However, if you create a read-only field in the business object, you can use it like a constant in a Regular formula. An example of such a field is triOneNU, which has a default value of 1 and is created for this purpose.

Generally, creating fields that are constants is not a good idea because the system must store the value for each record that is created. The mechanism that is used for defining a Regular formula ensures that it cannot be created incorrectly. For this reason, Regular formulas are considered unbreakable.

## Defining a regular formula in Data Modeler



© Copyright IBM Corporation 2013

6

### Defining a regular formula in Data Modeler

To specify a formula, you typically select a field, and then an operator, and then a field until the formula is complete. As an example, you want to add the value of the Labor Cost, Mileage Cost, and Overhead Cost fields together. These fields are all number fields, and the results must be put into a number field.

To create a Regular formula and obtain a sum of the fields in this example, perform the following steps:

1. In the Data Modeler, click the name of a number field in the Field List panel to open its properties. Scroll to the bottom of the properties.
2. Select the **Formula** check box and select **Regular** for the Formula Type property.
3. In the Formula Field property area, click the first **Find** hyperlink.  
A section is displayed below the Formula Field section.
4. Select the section name that contains the field, and then select the field. In this example, the section name is **General**, and the field name is **Overhead Cost**.
5. Click the **Accept** action.

The name of the field is displayed in the Formula window in place of the Find hyperlink.

You select operators from the list to the right of the field. The operators that are available are based on the field type. Number fields have addition, subtraction, multiplication, and division operators. Text fields have only concatenation. Other field types have other operators. The default operator for a number field is addition.

6. Accept the default operator as addition.
7. Repeat the process to select the Mileage Cost field.
8. Add operators and fields until the formula is complete. In this case, click **Find** to select the **Overhead Cost** field.

The Formula Field window shows another operator after the last selected field, and another Find link. Ignore them. They have no effect on your formula.

9. After the formula is complete, use the **Save Field** action to save the field properties.

## Extended formula

- You use fields or queries as inputs
- You use data from same record and from other records
- You use constants directly
- You use results of other formulas
- They can break if not created correctly

© Copyright IBM Corporation 2013

7

### *Extended formula*

Extended formulas are a more flexible mechanism for computing values than regular formulas. Extended formulas can use data that is external to the record and can support more advanced operations than add, subtract, multiply, and divide.

You enter extended formulas as text, and can use constants. Besides referring to fields in other business objects, you can also use the results of a formula or query as an input.

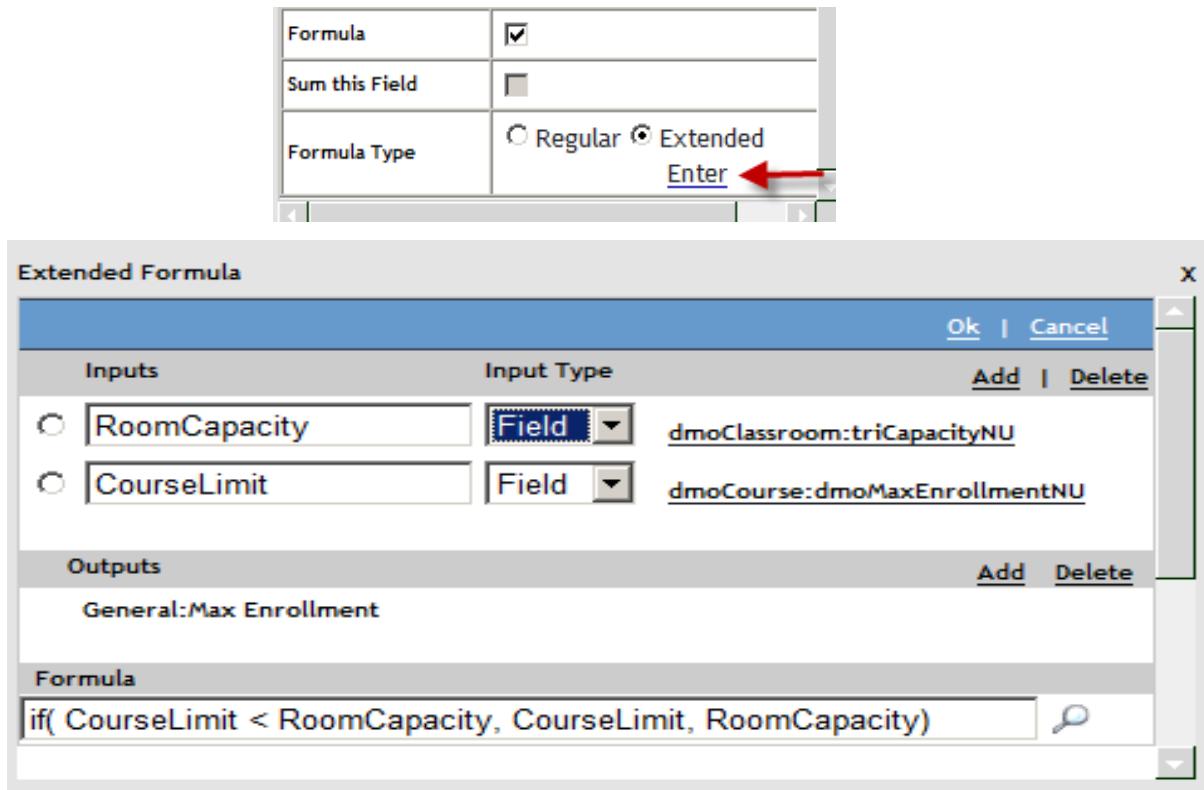
In the Data Modeler, extended formulas are available for text, number, date, or any other field type with a Formula check box in its properties.

In the Workflow Builder, in tasks like Create Record or Modify Records, extended formulas can access all fields in the source record of the object map. Extended formulas are available for fields of any data type. Extended formulas can be used in the Workflow Condition Builder as criteria for conditions in workflow tasks like Start, Switch, and Break.



**Important:** At run time, if all parameters in an extended formula are not defined, the formula is broken. The system continues processing as though the extended formula did not exist and puts an entry in the log.

## Defining an extended formula in Data Modeler



© Copyright IBM Corporation 2013

8

### Defining an extended formula in Data Modeler

You cannot supply an extended formula for a field until after the field is saved for the first time. In the Data Modeler, after a field is saved for the first time, you can edit the properties of a field to add an extended formula. When you define an extended formula, you select the **Formula** check box and select **Extended** for the Formula Type property. When you select the Extended option, you see a link named Enter. To create or edit an extended formula, click the **Enter** link in the Formula Type property. Clicking the Enter link causes the Extended Formula properties panel to open.

The Extended Formula panel is divided into three sections:

- The Inputs section specifies the fields or queries from which the formula gets values.
- The Outputs section identifies where the result of the formula goes.
- The Formula section contains the text of the actual formula.

Define an extended formula by setting values in the three sections. When the formula is complete, click the **Ok** action at the top of the form. If you do not click Ok, the formula is lost. You must also click **Save Field** on the section bar at the top of the Data Modeler to save the formula.

## Inputs section

To add an input to an extended formula, click the **Add** action on the Inputs section bar. This action adds a blank input in the Inputs section, with these properties:

- A selection field, in case you want to delete the input.
- An empty text field. Type a name for the input here.
- A selection list that has the default value Field. The other choice is Query. This selection defines the input type.
- A link named Select. This link is used to select the field that is the source of the input value.

To select a field for the input, click the **Select** hyperlink. The Pick Element window opens to show a tree for choosing a field. The elements in the tree are ordered alphabetically. Sections on the business object of this field are listed first, followed by sections in business objects that are associated to this business object of this field. Click the plus icon to expand the view. Click the minus icon to close the view. A dot icon indicates a field. Click the name of a field to select it.

## Outputs section

An extended formula is created with the field it belongs to as one of its outputs. Typically, a formula has only one output, but the results can be put into other fields. You can add the other fields that are to receive the result of the formula to the Outputs section of the extended formula.

## Formula section

You enter the formula into the formula window. The formula can consist of the inputs that you defined, constants, and predefined formulas, all connected by operators. Note the following details about these items:

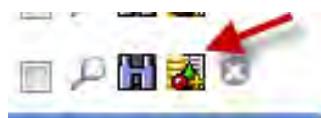
- For the inputs, the name must match exactly in case and spelling.
- For string constants, you can use either single or double quotation marks.
- To see a list of available predefined formulas, click the picker (magnifying glass icon) to the right of the Formula input area.
- The operators for number fields are: add, subtract, multiply, and divide (+, -, \*, /). For text fields, the plus sign (+) is used to concatenate values.
- Parentheses can be used to group portions of the formula to change the order of operations.

More information about creating extended formulas is available in *Application Building in the IBM TRIRIGA Application Platform: Calculations*.

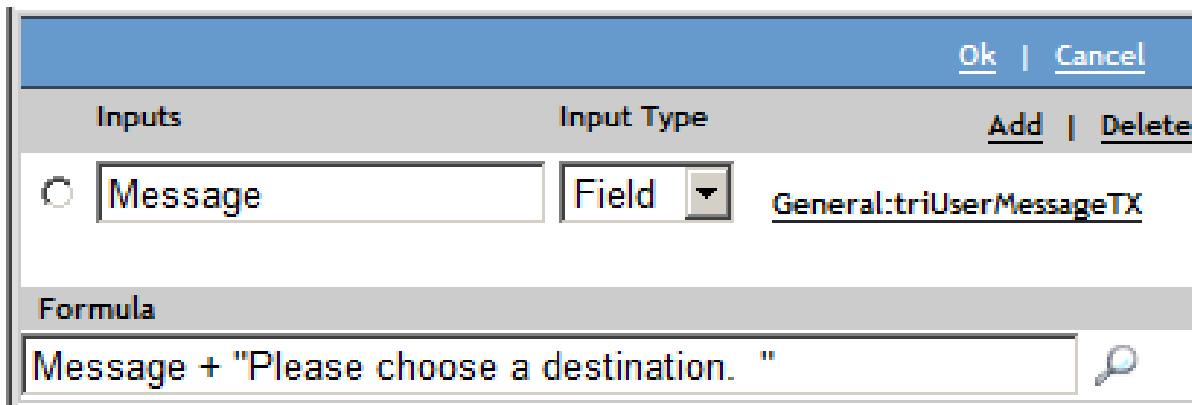
## Defining extended formula in workflow object map

In the Edit Map of Modify Records or Create Record task:

- Click formula picker



- Set formula



Inputs	Input Type	Add   Delete
<input checked="" type="radio"/> Message	Field	<a href="#">General:triUserMessageTX</a>

**Formula**

Message + "Please choose a destination. "

© Copyright IBM Corporation 2013

9

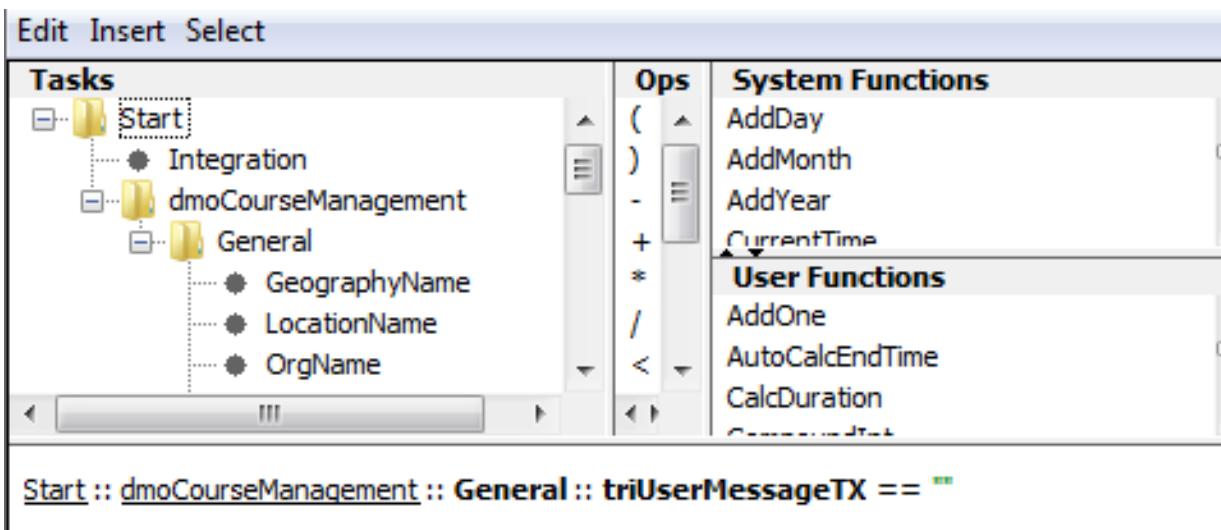
### Defining extended formula in workflow object map

The Create Record and Modify Records workflow tasks have an object map that is used to set field values. One of the mechanisms available to set the value of a field is an extended formula. You can access the extended formula in the Edit Map window by clicking the multicolored icon at the right of a field. Clicking this icon opens the Extended Formula properties window.

Unlike the Extended Formula properties window in the Data Modeler, the window in the workflow builder does not have an Outputs section. The results of the extended formula can be stored only in the field that the formula is associated with. Other than missing the Outputs section, the process of defining an extended formula in the Workflow Builder is the same as in the Data Modeler.

## Using an extended formula as a workflow condition

Set formula in a Start, Switch, or Break task



© Copyright IBM Corporation 2013

10

### Using an extended formula as a workflow condition

Some types of workflow tasks, such as the Start, Switch, and Break tasks, use a condition to decide what they do. A condition is a comparison between values, and it is either true or false. The properties of these tasks contain the Workflow Condition Builder, which is used to create and edit conditions.

The purpose of the Workflow Condition Builder is to build a formula in its bottom panel that is either true or false. You build the formula by using the Edit, Insert, and Select menus, along with the Tasks, Basic Ops, and System Functions panels. You use each of these menus and panels for the following purposes:

- **Edit:** Has options for cutting, pasting, and deleting portions of a condition.
- **Insert:** Is used to add numbers or text values to the condition. When you add a text value, the Condition Builder automatically puts quotation marks around it.
- **Select:** Includes methods for selecting parts of the condition.
- **Tasks:** Is used to insert task attributes and fields of records into the condition. Each record that is produced by a task before the current task is shown as a folder in the tasks panel. Click the folder to expand the contents of the record. Sections that are contained in the record are also

shown as folders. Click these folders to see the list of fields in the sections. Insert a field into the condition by clicking the field name.

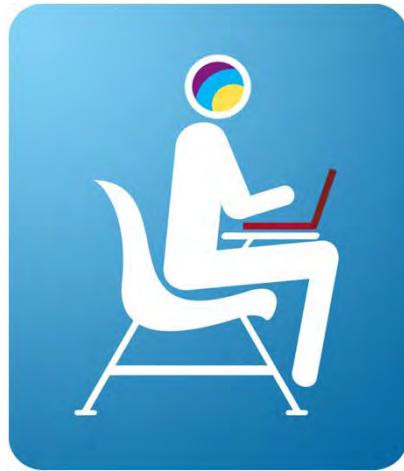
- **Basic Ops:** Is used to insert operators into the condition. There are arithmetic operators, such as + and -, and comparative operators, such as <, >, and ==. There are also parentheses that can be used to group clauses in the condition and change the order of evaluation. Use the && (and) and || (or) operators to join conditions together.
- **System Function:** Is used to insert system-defined and user-defined functions into a condition. The functions that you can select from the System Function panel are the same functions that are used for extended formulas.

The *Application Building in the IBM TRIRIGA Application Platform* guide has a comprehensive description of the Condition Builder. It also has a detailed example of how to build a condition.



## Student exercises

---



---

© Copyright IBM Corporation 2013

11

### *Student exercises*

Perform the exercises for this unit.

## Checkpoint questions

1. True or false: You can use a Regular formula anywhere that you use an Extended formula.
2. True or false: You can use constants and queries in Regular formulas.
3. How do you know if an Extended formula breaks?
4. Name two common errors that are made when setting a field value with an Extended formula.
5. What must be done before you can apply an Extended formula to a field?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.



## Checkpoint answers

---

1. True or false: You can use a regular formula anywhere that you use an extended formula.

*False. Regular formulas can only be used in Data Modeler.*

2. True or false: You can use constants and queries in regular formulas.

*False. Regular formulas can use only fields. Extended formulas can use constants and queries.*

3. How do you know if an extended formula breaks?

*It does not produce results, and a message is put in the log.*

## Checkpoint answers (continued)

4. Name two common errors that are made when setting a field value with an extended formula.
  - *An input is created but a field is not selected.*
  - *In the formula, the name of an input is spelled wrong, or the wrong case is used.*
5. What must be done before you can apply an extended formula to a new field?

*You must save the field.*



## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

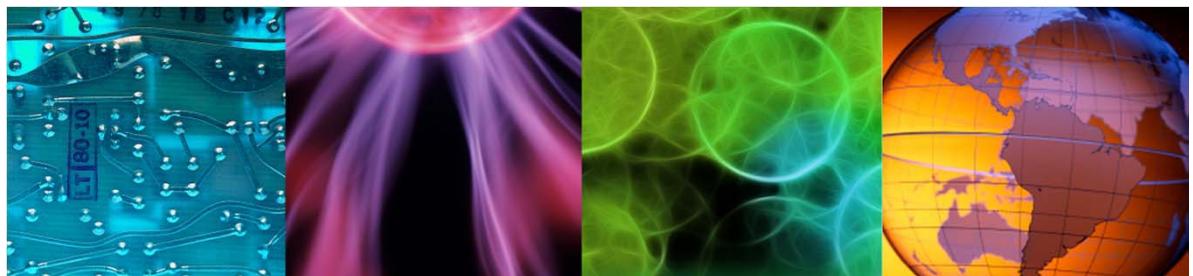
- Create a regular formula
- Create an extended formula in a business object
- Create an extended formula in a workflow



## 15 Workflows



## 15 Workflows



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

Workflows in IBM TRIRIGA are how you add business logic to an application. The Workflow Builder tool is used to create and modify workflows and workflow tasks.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building for the IBM TRIRIGA Application Platform*

## Objectives

---

After completing this unit, you should be able to perform the following tasks:

- Define workflows
- Use the Workflow Builder tool
- Create and publish a workflow

# Lesson 1. Building workflows



## Lesson 1: Building workflows



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Workflows in IBM TRIRIGA are how you add business logic to an application. The Workflow Builder tool is used to create and modify workflows and workflow tasks.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe what workflows are
- Use the Workflow Builder tool
- Create and publish a workflow

### References

*Application Building in the IBM TRIRIGA Application Platform*

## Definition of a workflow

- A workflow is a sequence of tasks for the platform to perform automatically
- Use workflows to add business logic to applications
- Use workflows for the following tasks:
  - Validate user input
  - Set the value of a field
  - Vary the appearance of a record
  - Create and manipulate records

© Copyright IBM Corporation 2013

4

### *Definition of a workflow*

A **workflow** is a specified sequence of tasks that is performed automatically. You use a workflow to specify the tasks that you need an application to perform automatically, one that the IBM TRIRIGA Application Platform does not provide.

You can add business logic to your applications by creating a workflow. You can also create workflows to define any business process that is associated with the system or the business objects in the system. Many predefined workflows are delivered with the IBM TRIRIGA applications.

You must understand many concepts to successfully create a workflow. It helps to have a background in programming, but it is not necessary.

The first thing to 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:

- Set the values in the fields of a new record. Formulas are a simpler way to initialize the values of fields, but there are computations that a workflow can do that a formula cannot. There is no

certainty to the order in which formulas are computed. If you need initial values for fields that are computed in a certain order, you must use a workflow.

- Change the appearance of a record in a form based on the contents of underlying records or a user's actions. You can hide sections, make sections visible, change font size and color, and make many other changes the appearance of a record.
- Validate the contents of a record before it is saved.
- Perform computations to set the values of fields in a record.
- Create or manipulate records without requiring any interaction with a person.
- You can use workflows to route work to people. Routing work can be done entirely within the IBM TRIRIGA Application Platform by having workflows put action items in a person's portal.

## Starting a workflow

You can start a workflow in response to the following items:

- Creating a record
- Opening a form
- Changing a field value
- Clicking an action
- Establishing an association between records
- Addressing events that happen inside the platform

### *Starting a workflow*

You can start a workflow in several ways, which are determined by whether it is synchronous, subflow, or asynchronous.

Asynchronous workflows are started in response to an event that occurs on a business object. A workflow is registered for a particular event or business object combination by using the Event property in the workflow Start task. There are two types of events: events that are generated by state transition actions, and system events, such as scheduled events or the creation of associations between records.

Asynchronous workflows are not directly connected to a state transition action. Instead, they can 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 starts the workflow.

An asynchronous workflow can also be started by a system event. System events happen to records that might not be the direct result of an action by a user. For example, when a record becomes disassociated from another record, there is a system event.

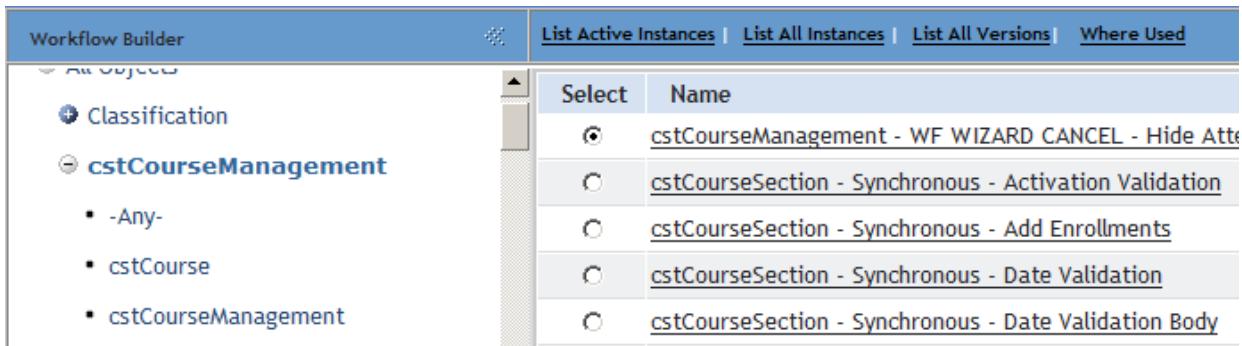
You can start a synchronous workflow in the following ways:

- From a sub action that is attached to a state transition action.
- From an action that a user performed on part of a form. Form actions include actions that are at the top of a section and actions that are triggered when a button is clicked. Workflows can also be triggered when the user changes the value of a field.
- When a record is created. The way to specify when to start a workflow in this way is to name it in the Pre-Create Workflow property of the business object.
- A synchronous workflow can be started when a record is loaded in a form. The way to specify that a workflow is to be started in this way is to name it in the Pre-Load Workflow property of the form.
- A synchronous workflow can be started from another workflow by using a Call Workflow task.
- A synchronous workflow can be started from certain navigation items.
- A synchronous workflow can be started from a report or query.

You can set a condition in a Start task of a workflow to prevent the workflow from running unless the conditions are met. Conditions in a start task can include contents of its record, the state of the record, or the current time. These limitations that you can impose on the start of a workflow are called **start conditions**.

## Workflow Builder tool

- Workflows are created and managed in the Workflow Builder
- Workflows are organized by module and then by BO
- Click **Tools > Builder Tools > Workflow Builder**



© Copyright IBM Corporation 2013

6

### Workflow Builder tool

You use the Workflow Builder to create and modify workflows. You access the tool by clicking **Tools > Builder Tools > Workflow Builder**.

The Workflow Builder organizes workflows by the module that they are associated with. Select a module by clicking its name on the left side of the window. Workflows that are associated with the module or one of the business objects in the module are listed on the right side of the window.

Most workflows are associated with a particular business object in a module. Click the name of a business object to see only those workflows that are associated with that business object.

If you do not see the names of business objects in a module, click the plus sign next to the name of the module to expand it. The business objects in the module now show under the name of the module. Also, the plus sign next to the module changes to a minus sign. To collapse the list of business objects, click the minus sign.

## Right side of the Workflow Builder

List Active Instances   List All Instances   List All Versions   Where Used		
Select	Name	Revision
<input type="radio"/>	<a href="#">cstFleetManagement - Synchronous - Set ID field</a>	0
<input type="radio"/>	<a href="#">cstFleetManagement - WF WIZARD CANCEL</a>	2
<input checked="" type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	7
<input type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation body</a>	0

New   Copy   Publish   Retire   Delete		
Object	Action	Status
-Any-		Published
-Any-	WF WIZARD CANCEL	Revision In Progress
cstCar		Revision In Progress
cstCar		Published

© Copyright IBM Corporation 2013

7

### Right side of the Workflow Builder

Workflows that are associated to the selected module and business object are listed in the right side of the Workflow Builder. By default, the workflows are sorted by name. Click any other column header to sort by that column in ascending manner. Click that column again to sort in a descending manner.

Notice the following columns in the display:

- The Revision column shows the version number of the workflow.
- The Action column shows the event that triggers the workflow, for asynchronous workflows. This column is blank for synchronous workflows.
- The Object column shows the business object that the workflow is associated with. If the workflow is not associated with a specific business object, this column shows **-Any-**.

The Workflow Builder has a menu of actions that you can perform on workflows. First, select a workflow. Then, select one of the following actions:

- New:** Click this action to create a workflow.
- Copy:** This action creates a workflow that is a copy of the selected workflow.

- **Publish:** Click this action to make the selected workflow available for use.

Workflows follow the same Create-Publish-Revise life cycle that business objects and other components follow. 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 version of the workflow does not replace the version of the workflow currently in use until the modified version of the workflow is published.

- **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 it.

To change a published workflow, select it in the Workflow Builder and click the Revise action. The Revise action creates a version of the workflow that can be edited. The revision number of the workflow is listed in the Workflow Builder to the right of the workflow name. Actions continue to run the published version of the workflow until the revised version is published.

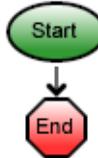
- **Retire:** Clicking the Retire action takes the currently selected workflow out of use. It can be put back into use by publishing the workflow again.
- **Delete:** Clicking the Delete action deletes the selected workflow.

The Workflow Builder also has a menu of actions with information about a workflow. Select a workflow and then click one of the following actions:

- **List Active Instances:** Shows instances of this workflow that are still running.
- **List All Instances:** Shows current and past instances of this workflow.
- **List All versions:** Shows all versions of this workflow.
- **Where Used:** A window opens and shows what references or uses the selected workflow. You can export the information in the Where Used window by clicking the Export Usage action.

## Workflow Editor

- Click **New** to create a workflow in the Workflow Editor
- Workflows always start with two tasks: Start and End
- The Workflow Editor has no save option
  - All changes that you make to the workflow or task properties are saved automatically
- Click a task to open its properties window



© Copyright IBM Corporation 2013

8

### Workflow Editor

You use the Workflow Editor to view and modify workflows. Clicking the **New** action causes the workflow editor to open so that you can edit the new workflow. To modify or view an existing workflow in the Workflow Editor, click the name of the workflow in the Workflow Builder.

The Workflow Editor is organized into three sections: Diagram; Properties; and Task palette. The Diagram section is the only section of the Workflow Editor that is always visible. It shows the tasks of a workflow as shapes. Each kind of task has a different color and shape. A newly created workflow has two tasks: a Start task and an End task.

The Workflow Editor also shows arrows that connect tasks. The purpose of the arrows is to show the order in which the tasks are performed. As tasks are added to a workflow, the arrows automatically adjust to accommodate the new tasks.

The Properties section of the workflow becomes visible when you click a workflow task. It is always displayed at the bottom of the window. The selected task is highlighted in the diagram section when its properties are displayed.

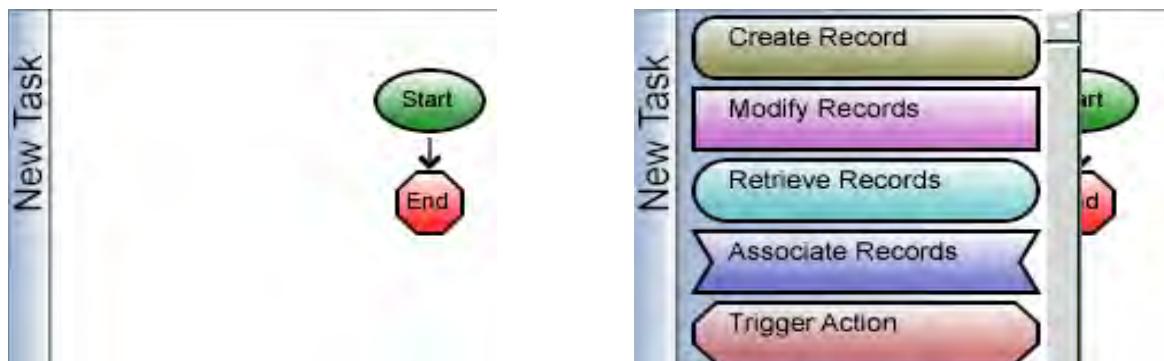
Each kind of task has a different set of properties. The properties of each kind of workflow task are described as part of the description of each kind of task.

If the properties section is already visible and you click a different task, the properties for that task are displayed in the properties section. To make the workflow editor stop displaying any properties section, click the background of the diagram section.

The **Ok**, **Apply**, **Save**, and **Cancel** actions are not in the Workflow Editor. The Workflow Editor works differently in this respect from the rest of the IBM TRIRIGA Application Platform. All changes that are made to a workflow or a task take effect as you click another task or click in the background. They are not saved if you close your browser without clicking elsewhere.

## New Task palette

- You hover your cursor over the palette to see the available tasks
- Temporary data tasks are visible only if Temporary Data is selected in the Start task



© Copyright IBM Corporation 2013

9

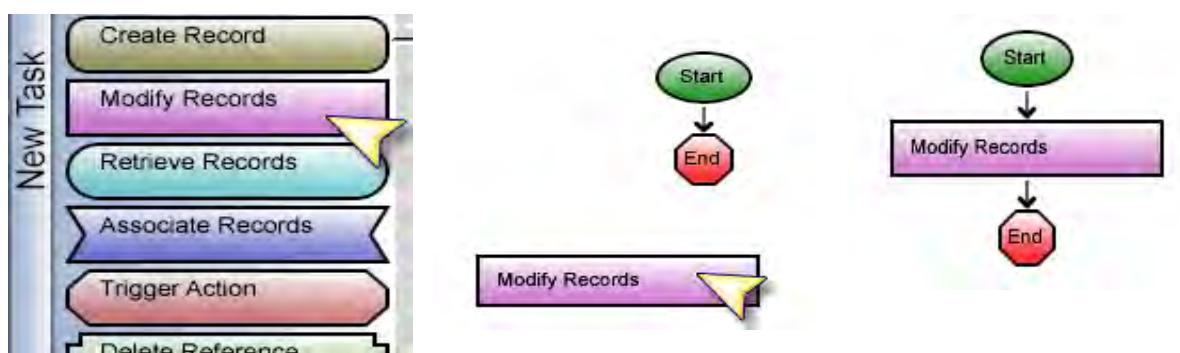
### New Task palette

The Task palette section is displayed 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 added to a workflow.

Depending on the context and properties that are set in the Start task of the workflow, some tasks might not be visible in the task palette. For example, if a workflow does not use temporary data, the temporary data tasks (Get Temp Record, Save Permanent Record) are not visible.

## Adding a task from the New Task palette

- Left-click the task with your pointer so that it is attached to the pointer
- Drag it to where you want it
  - The workflow changes to include the new task
- Left-click the task to position it



© Copyright IBM Corporation 2013

10

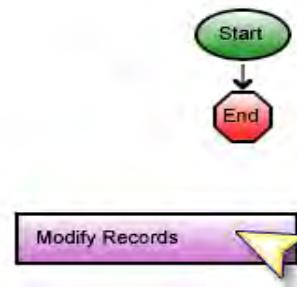
### *Adding a task from the New Task palette*

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 is displayed, click the shape of the task in the task palette. The task palette is no longer visible 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 into position in the diagram. When the task is in an acceptable position, the arrows in the diagram change to include the task. Click the task shape to leave it where it is or move it elsewhere.

## Moving and removing a task

- Moving a task

- Click a task while moving your pointer to get the task
- Drag it to where you want, and drop it



- Removing a task

- Open its properties, and click **Delete**
- If the task is attached to your pointer, add it to your workflow first



© Copyright IBM Corporation 2013

11

### Moving and removing a task

You can pick up and move workflow tasks to another place in the workflow. Click a task while you move your mouse, and the task attaches itself to your mouse pointer. The workflow is also adjusted to preview what it looks like with the task removed. Move the task to the place where you want it to be in the workflow, and click to place it there. If the task that you are moving contains other tasks, those tasks are also moved.

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 is a Delete action in the menu of the properties section. Clicking the Delete action deletes the task. If the task that is being deleted contains other tasks, those other tasks are also deleted.

## Instructor demonstration

- Workflow Builder
- Creating a workflow
- Task palette
- Adding tasks
- Moving tasks
- Deleting tasks



© Copyright IBM Corporation 2013

12

*Instructor demonstration*

## Two types of workflows

### Synchronous workflows

- You are typically the one who triggers the workflow  
It runs immediately
- The workflow must complete before you can perform other tasks

### Asynchronous workflows

- They are triggered by an event  
It runs soon after that
- You can perform other tasks while the workflow is running

© Copyright IBM Corporation 2013

13

#### *Two types of workflows*

One of the key properties of a workflow is whether it is synchronous or asynchronous. If a workflow is synchronous, it generally means that the workflow is started in response to something that a user did in the user interface. The workflow begins running immediately, and control is not returned to the user until the workflow completes.

If a workflow is asynchronous, it means that the workflow is started in response to an event that occurred in the system. The workflow is added to a queue and waits to be run based on its position in the queue.

Asynchronous workflows and the events that trigger them are independent of each other, meaning neither must wait for the other. For example, assume that a workflow is started in response to a state transition event that a user triggered. The workflow might not start running immediately, but the user can do something else immediately.

Whether a workflow is synchronous or asynchronous is determined by the value of its concurrence property. 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.

A workflow that is marked synchronous can also be called, by using the Call Workflow task. If the workflow is called from an asynchronous workflow, it is run as part of that asynchronous process. Workflows that are marked asynchronous are clear, but synchronous really means running in-line with the process that triggered it.

An asynchronous workflow cannot be called but must be started by an event. Use the Trigger Action workflow task to trigger actions that in turn cause events that can cause asynchronous workflows to be started.

## Two types of data

**Permanent** data is a record that is stored in the database

- It persists in the database until it is explicitly changed

**Temporary** data is a record that is opened in a form

- Someone might have modified it after it was loaded into the form
- It is the most current copy of the data available
- Asynchronous workflows cannot use temporary data

### *Two types of data*

The data that workflows typically use is considered to be permanent data. Permanent data is the values in records that are kept in the database indefinitely. If a workflow changes permanent data, the changes are made in the database immediately.

Temporary data is data that the user created or modified but is not permanently saved. For example, when a record is opened in a form, the form contains temporary data. The permanent data is not affected by any changes that are made by the user until the record is saved.

Workflows can access the temporary data by using the Get Temp Record task. Workflows that validate user input generally use temporary data, as it is the most current data available. If a workflow changes the temporary data but does not save it (by using Save Permanent Record), then the changes are lost. Asynchronous workflows, which are connected to events and not to user actions, have no access to temporary data. They can use only permanent data.

## Start task of a synchronous workflow

The screenshot shows the 'Workflow Properties' dialog for a workflow named 'cstCar - Synchronous - Activate Validation'. The 'Description' field contains a note about validating data in a record before activation. The 'Concurrency' setting is 'Synchronous'. The 'Temporary Data' setting is 'Temporary'. The 'Module' is 'cstFleetManagement' and the 'Object Type' is 'cstCar'. Several checkboxes are present: 'Save Workflow Instances' (unchecked), 'Lock Record For Other Users' (unchecked), and 'Propagate Integration Status' (checked).

### Start Conditions

© Copyright IBM Corporation 2013

15

#### *Start task of a synchronous workflow*

A Start task is always the first task in a workflow. It cannot be deleted or moved. Because you cannot delete a Start task, it does not have a Delete action in the menu for its properties.

The following actions can be used on a Start task:

- **Publish:** This action is displayed only if the workflow is new or under revision. Publishing makes a new workflow available for use, or puts revisions to an existing workflow into use.
- **Revise:** This action is displayed only if the workflow is published. Clicking this action creates a copy of the workflow that you can revise without affecting the production version of the workflow.
- **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.
- **Callers:** You click this action to see a list of the workflows that call this workflow.

The Start task always refers to the permanent version of the record in the database. Any workflow tasks that refer 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 are made immediately to the record in the database.

The properties of a Start task are actually properties for the entire workflow. The workflow properties section of the start task has these fields:

- **Name:** This property is the name that is seen in the Workflow Builder and the name that is used to identify the workflow. This property is *not* a label for the Start task. Start tasks do not have a label that you can change.
- **Description:** This property holds a description of the entire workflow. Take full advantage of it. The more complex a workflow is, the more important it is to document the workflow.
- **Concurrency:** This property is used to determine whether the workflow is synchronous, subflow, or asynchronous. This example is for a synchronous workflow.
- **Temporary Data:** This property is visible only if the value of Concurrency is Synchronous. The value of this property determines what type of data the workflow can access. If Permanent is selected, the workflow can access only permanent data. If Temporary is selected, the workflow can access temporary data, or permanent data, or both.



**Note:** If **Temporary** is selected and the workflow does not use temporary data, the workflow fails.

- **Module:** The value of this property is the module with which the workflow is associated. It defaults to the module selected in the Workflow Builder when the workflow was created. A different module can be selected by using the picker next to the field.
- **Object Type:** This property identifies the workflow as being a module-level workflow, or being associated with a specific business object. The value of this property is either the name of a business object in the specified module or **-Any-**.

If this field contains the name of a business object, the workflow is associated with that business object. The workflow is started only as a result of something that happens to a record created from that business object.

If the value of this property is **-Any-**, the workflow is a module-level workflow. It can be started by something that happens to a record that is created from any business object in the specified module. The workflow is restricted to accessing only the fields in a record that have the same name as fields in the base business object of the module.

- **Save Workflow Instances:** When a workflow is triggered, the platform makes a copy of it called an **Instance**. It is this copy of the workflow that is run. If this check box is selected, a record is kept of each workflow instance. The record includes the execution path that the workflow followed. There is a similar setting in the administrative console that can override this property.

When you design or develop workflows, they are much easier to understand if you follow a naming convention for your tasks and workflows. Naming conventions help someone who is reviewing the workflow to know your intentions for the workflow as a whole and for each task. Use the standard naming conventions for each task type. Also, have a description in each task that further clarifies its intent. Synchronous workflows have a three-part name: business object name - Synchronous - description.

For Business Object names, use the business object upon which the task is performing work. If the task can perform work on any business object in the module, use the module name. If a workflow references an as-delivered business object, use **cst-** plus the business object name. If it references a new business object, you do not need this prefix. An example of a synchronous workflow that uses an as-delivered business object is cst-triLocation - Synchronous - Populate parent fields on create. Another example of a synchronous workflow that uses a custom business object is cstCarMoveRequest - Synchronous - Activation validation.

## Start task of an asynchronous workflow

**Properties**

Name: CstFleetManagement - WF WIZARD CANCEL

Description:

Concurrency:  Synchronous  Subflow  Asynchronous

Integration:  On Process Completion:

Module: cstFleetManagement Object Type: -Any- Event: WF WIZARD CANCEL

Save Workflow Instances:

Lock Record For Other Users:

Propagate Integration Status:

**Actions**

© Copyright IBM Corporation 2013

16

### Start task of an asynchronous workflow

An asynchronous workflow has the following properties:

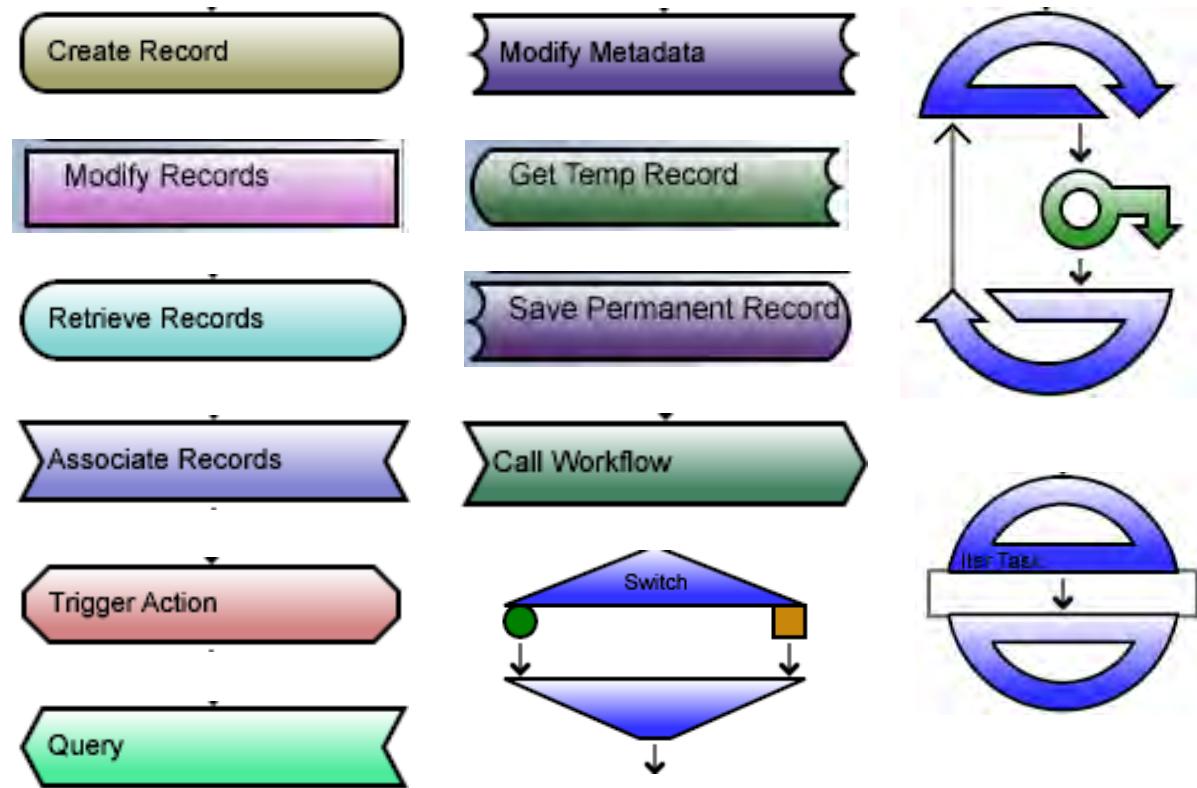
- **Temporary Data:** Temporary data is not available to asynchronous workflows. This property is not visible for an asynchronous workflow.
- **Integration:** When the property is selected, the workflow is used to migrate data from staging tables in IBM TRIRIGA records. This type of workflow is used extensively in IBM TRIRIGA DataConnect.
- **On process completion:** When this property is selected, the workflow runs only when all other asynchronous workflows for the primary event complete.
- **Event:** The value of this property is the action or system event that can start this workflow.



**Note:** The Event list is connected to the object type. If you select an event and then change the object type, the event is reset and you must select it again.

The naming convention for an asynchronous workflow is like a synchronous workflow, except that the event is used in the middle: business object name - event - description. Using the event makes it easier to visually distinguish a synchronous workflow from an asynchronous one. Examples of asynchronous workflow names are cst-triPeople - triActivate - Trigger Approval Process and cstFleetManagement - WF WIZARD CANCEL - Hide attention section.

## Common workflow tasks



© Copyright IBM Corporation 2013

17

### Basic workflow tasks

Workflows are organized into simple pieces called tasks. There are different kinds of workflow tasks, and each performs a different activity. For example, there are workflow tasks to create a record, change the contents of a record, and create associations between records. By using the different types of tasks in the right way, you can create a workflow that can perform whatever business logic is needed.

The following basic tasks are commonly used in workflows:

- **Create Record**: Create a new record by using an existing record as input.
- **Modify Records**: Modify the values in existing records.
- **Retrieve Records**: Get a list of records based on their association with another record.
- **Associate Records**: Create or remove an association between records.
- **Trigger Action**: Trigger a state transition action.
- **Query**: Run a query and get a list of the resulting records.
- **Modify Metadata**: Change the presentation of a form for a particular record.

- **Get Temp Record:** Access the temporary data that is associated with a record; only available for synchronous workflows that select temporary data.
- **Save Permanent Record:** Update the permanent data of a record with temporary data; only available for synchronous workflows that select temporary data.
- **Call Workflow:** Run a synchronous workflow. Processing in the current workflow is suspended until the called workflow completes.
- **Switch:** Conditionally direct the flow of a workflow to one path or another.
- **Loop:** Repeat a set of tasks until a condition is met, or while a condition is met.
- **Iterator:** Repeat a set of tasks for all records in a list.

Each workflow task has properties that you use to specify the details of what the task does. These properties are covered individually on the following pages, for some of the tasks. Most workflow tasks have a name that is visible in the workflow diagram. The following conventions for naming workflow tasks can make workflows easier to read and understand:

- Always give each task in a workflow a descriptive and distinct name to clearly distinguish it from every other task in the workflow. Although workflows can have more than one task with the same name, giving each task a different name can help prevent confusion. Different names also help identify the correct task when continuing development or troubleshooting a workflow.
- Any tasks in a workflow that are modified are to have **cst** prefixed on the name. This prefix clearly identifies tasks that are customized versus tasks that are unchanged. Workflows that are entirely custom have **cst** prefixes on all task names.
- Use the business object name in place of **Record** in the name of the Get Temp Record and Save Permanent Record tasks. For example, use **cst Get Temp cstCar** instead of **cst Get Temp Record**.
- Use a short description of a workflow task instead of a generic name. For example, use **cst Hide Attention Sections** instead of **Modify Metadata**.

## Modify records task

**Modify Task Properties**

**Label:** cst Clear Attention messages

**Description:** Clear the attention messages, to initialize the workflow

**Formulas:** Disable Auto Recalculation

**Map To Records**

Take the **Business Object** of Task **Get Temp cstCar (cstCar)**

Use it

Only if association is -Any-

**Object Type:** cstCar

**Map From Records**

Workflow Activity  Existing Record

Take the **Business Object** of Task **Get Temp cstCar (cstCar)**

Use it

**Object Type:** cstCar

© Copyright IBM Corporation 2013

18

### Modify records task

The Modify Records task modifies values in existing records. The properties of a Modify Records task are organized into four sections:

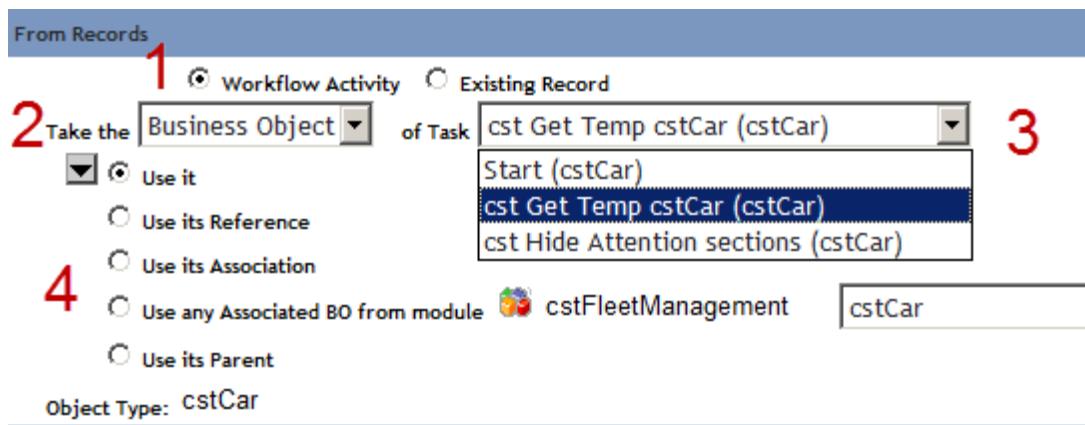
- **Modify Task Properties:** The properties of the task, in addition to the Label and Description properties is the Formulas property. This property defines the formula recalculation behavior on this task when the record is saved. The default setting is Disable Auto Recalculation.
- **Map To Records:** Identifies the record to be modified by the task. It is typically called the target record.
- **Map From Records:** Identifies the record that is the source of values for the task.
- **Transaction:** Specifies the scope of the transaction that is being processed.

There is a **Delete** action on the Modify Task Properties section heading. Use it to remove this task from the workflow.

There is an **Edit Map** action on the Map To Records section heading. This action opens the Object Mapping window. Use the Object Mapping window to define how to change values in the target record (**To record**).

## Specifying the record to work with

1. Specify **Workflow Activity** or **Existing Record**
2. Select **Take the Assignee** or **Take the Business Object**
3. Select the previous task
4. Choose from the *Use it* list of choices



© Copyright IBM Corporation 2013

19

### Specifying the record to work with

Many workflow tasks manipulate a record in some way, either as input to the task, or as output, or both. You must use the properties of these tasks to specify the record or records that they use.

To specify a record, you use the following process:

1. Identify whether the record comes from an activity (task) in the workflow, or is an existing record. Choosing an existing record is uncommon. For this example, the results of a prior workflow activity are chosen. This property is only present in the properties of the From (or source) record.
2. Determine whether you want the business object (record) from that prior activity or the Assignee. Choosing the assignee is uncommon. In this example, the business object is selected.
3. Select the specific workflow task that produced the record you want to work with. Try to avoid *daisy chaining* your tasks, that is, pointing each task to the previous one for its input. This approach works but can lead to breakage if tasks are rearranged or new tasks are inserted. If temporary data is used, use the Get Temp Data task as the source and target of your data. If temporary data is not being used, the Start task can serve as the source and target of your data.

Using either approach means that your pointers to records can remain unchanged if you add or rearrange tasks in your workflow.

4. Specify what to do with the record that you selected, based on the following choices:
  - **Use it:** This choice is the default setting. Choose it when the selected record contains the data that you want.
  - **Use its reference:** If the selected record does not have the data that you want, perhaps it connects to a record that has the data that you want. Choose this option if there is a locator field or smart section in the selected record that references the record with the data that you want. Choosing this option opens another dialog window where you identify the reference to use.
  - **Use its association:** Choose this option if the selected record does not have the data that you want but it has an association to a record with the data. Choosing this option opens another dialog window where you identify the association to use to access the other record.
  - **Use any associated BO:** Use this option with temporary associations to other records.
  - **Use its parent:** Selected when the selected record is hierarchical and you want its parent.

## Other choices for the record

Map From Records

Workflow Activity     Existing Record

Module:  triPeople      Object:       Record: Hammond, Brent - EMP1 (Hammond, Brent - EMP1)

Object Type: My Profile

- Selecting an **Existing Record**

- Taking the **Assignee**

Map From Records

Workflow Activity     Existing Record

Take the  of Task

Use its Association     Associated To

Object Type: triPeople

<input type="radio"/> triPeople	triPeople	<input type="button" value="Has Delegate"/>
<input checked="" type="radio"/> triPeople	triPeople	<input type="button" value="Associated To"/>
<input type="radio"/> triPeople	triPeopleLink	<input type="button" value="Has Reservation Deleg"/>

20

Other choices for the record

Two other methods that are used for selecting the record to work with are as follows:

- **Selecting an Existing Record:** Use this property to identify a single, specific record that the workflow uses every time that it is run. This record is typically some type of configuration record that can provide initial values or settings that are required by the workflow. This selection is only available for identifying the input to a task.
- **Taking the Assignee of a task:** If Assignee is selected, then the My Profile record of the user that is assigned to the selected task is the target record. The user that is assigned to the task is typically the person who triggered the workflow.

The My Profile record of a user has less information than their triPeople record. Common practice when Assignee is selected is to also choose **Use its association** to navigate from the My Profile record to the triPeople record. Choosing Use its association opens a dialog window to identify the specific association to be used. After you select the **Associated To** association, you click **Accept**.

## Object Mapping window

The screenshot shows the 'Object Mapping' window with the following data:

Source Business Object	Target Business Object	Action Buttons
cstCar	cstCar	-Select-
cstAuthorizedLocations	cstAuthorizedLocations	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
cstPrimaryLocation	cstPrimaryLocation	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triCityTX	cstPrimaryLocation::triCityTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triNameTX	cstPrimaryLocation::triNameTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triStateProvTX	cstPrimaryLocation::triStateProvTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
cstSupervisor	cstSupervisor	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triAddressTX	cstSupervisor::triAddressTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triCityTX	cstSupervisor::triCityTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triEmailTX	cstSupervisor::triEmailTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>
triLastNameTX	cstSupervisor::triLastNameTX	<input type="checkbox"/> <input type="button" value="Search"/> <input type="button" value="History"/> <input type="button" value="Audit"/> <input type="button" value="Delete"/>

© Copyright IBM Corporation 2013

21

### Object Mapping window

The Object Mapping window is used to specify how values are copied from one record to another. Access the Object Mapping window by clicking the **Edit Map** action on the Map To Records section bar of a Create Record or Modify Records task.

The Object Mapping window is laid out in three columns:

- The left column is based on the target business object. The name of the business object is at the top, and the fields of the business object are listed below it, grouped by section.
- The middle column shows what is being mapped into each target field. By default, all fields from the source business object are mapped into fields with the same name in the target business object. Fields in this column are identified as **section-name::field-name**, like **cstSupervisor::triCityTX**. The name of the source business object is at the top of this column.
- The right column has a set of selectors on each line that can be used to identify values to be mapped into the target field. The choices for selectors are as follows:
  - **Check box:** Maps a value into the target field. When you select the check box, an input window opens in the middle column for you to enter a value for the field.

- **Field picker:** Maps a field from the source record into the target field. Clicking this picker opens a selection window that lists fields from the source record. Select one to map it into the target field. The section and name of the source field are shown in the middle column. You can use the field picker for smart sections and locator fields, but certain rules apply.
- **Binoculars:** 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 binoculars icon. It causes a window to open where you can select a computation. For each of the tasks that precedes the current task, you can pick a count, sum, or list of records that the task computed.
- **Formula picker:** You can use an extended formula as a source of data for the field. Clicking this icon opens an Extended Formula window.

The Object Mapping window has these actions at both the top and bottom of the window:

- **Clear All:** This action clears all mapping. The default behavior of mapping all fields with the same name is inefficient, especially when you use the same record for the source and target. There is no reason to map fields that are not changing. Begin by clearing the default mapping and then map only those fields that are being changed.
- **OK:** Use this action to save the mapping and close the Object Mapping window.
- **Cancel:** Use this action to discard mapping changes and close the Object Mapping window.

You can also create an association from the source record to the target record. Associations are necessary for some processes such as notifications. Choose the association to be created from the list in the **Association Type** field. The field lists all associations that are defined at the business object level between the source record and the target record.

Locator fields and smart sections require a reference to be mapped into them in addition to a value. The field picker can provide the reference with source mapping.

When you are mapping a locator field, click the field picker to open the selection window. You select **Source** from the selection window to map a reference to the record into the locator field. Source mapping automatically populates a value into the field that is based on the locator field mapping that is defined in the business object of the target field. However, it does not put values into any other fields that have mapping defined in the locator field. You must map the value of those fields individually. Do not use source mapping for the non-locator fields.

Because a locator field contains both a reference and a value, you can map Locator fields into locator fields with the field picker.

To map values into a smart section, select the field picker in the section heading. Then, select **Source** for the mapping. All fields in the section are automatically populated based on the smart section definition in the target business object. Do not map values directly into the fields of a smart section.

## Modify metadata task properties

Modify Metadata Task Properties

**Label:** cst Hide Attention sections

**Description:** Hide attention sections, to initialize for workflow

**Form:** -Current-

**Reset Before Modification:**

**Records**

Take the **Business Object** of Task **Start (cstCar)**

Use it

**Object Type:** cstCar

**Edit Map** **Delete**

© Copyright IBM Corporation 2013

22

### Modify metadata task properties

You can use the Modify Metadata task to modify the metadata that controls the presentation of a form for a particular record. You can use this task to change many form properties, including making form sections hidden or visible.

Changes to metadata are specific to each record. They do not affect other records, and they do not change the form as defined in the Form Builder. They change how an individual record is displayed.

Within a workflow, you can work with temporary and permanent data. Temporary metadata does not exist. All modifications to metadata are immediate and permanent, and any changes remain until the next time the metadata is modified.

The Modify Metadata task has a label, description, and these other properties:

- Form:** Determines which forms are affected by this task. The selection list shows the names of all the forms that are associated with the business object selected in the Record section.

The default value is Current. This value causes the task to modify the metadata for whichever form is being used with the record identified in the Record section.

A selection of **All** modifies the metadata of all forms of the business object for the record that is identified in the Record section.

- **Reset Before Modification:** If this check box is selected, before this task modifies the metadata, the metadata of the record is reset to the values specified in the Form Builder.

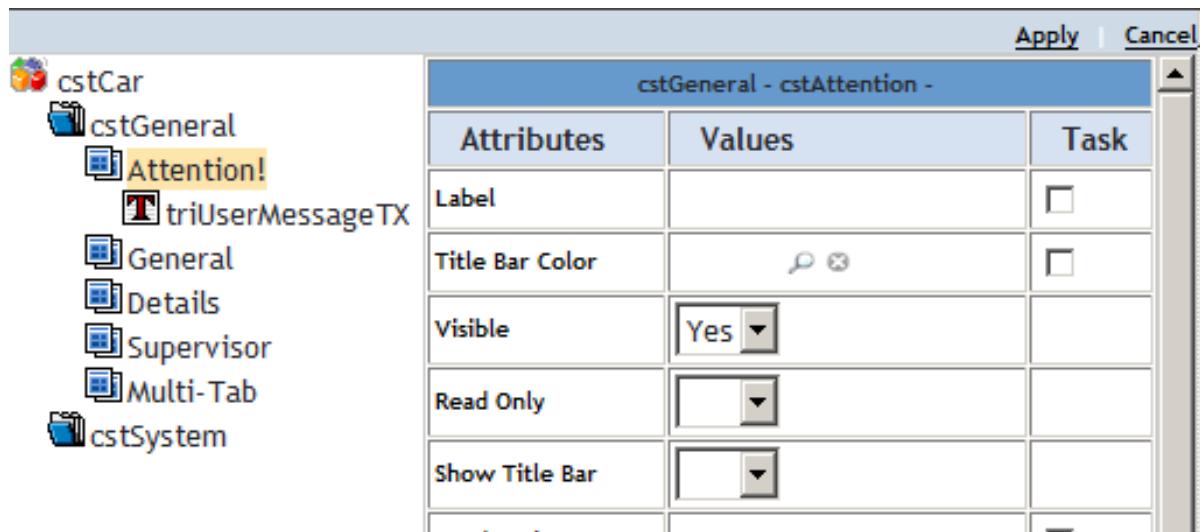
Select a record to work with by setting the properties in the Records section. Selecting a record from a temporary data task versus a permanent data task makes no difference in the behavior of this task. Metadata changes are always immediate and permanent.

The Modify Metadata task uses form mapping to specify the modifications to the metadata of the record. To open the Form Mapping window, you click the **Edit Map** link.



**Note:** You cannot determine the current value of a metadata property. You must assume that it is not what you want and change it to be what you need it to be.

## Form Mapping window



© Copyright IBM Corporation 2013

23

### Form Mapping window

You use the Form Mapping window to modify the properties of the metadata of a record. It has a similar appearance to the Form wizard, with the navigation tree on the left and the properties on the right.

The navigation of the Form Mapping tree is the same as in the navigation panel on the Layout tab of the Form wizard. When you click one of the nodes in this tree, you see attributes that correspond to your selection. They are different if you select a tab, section, field, or action.

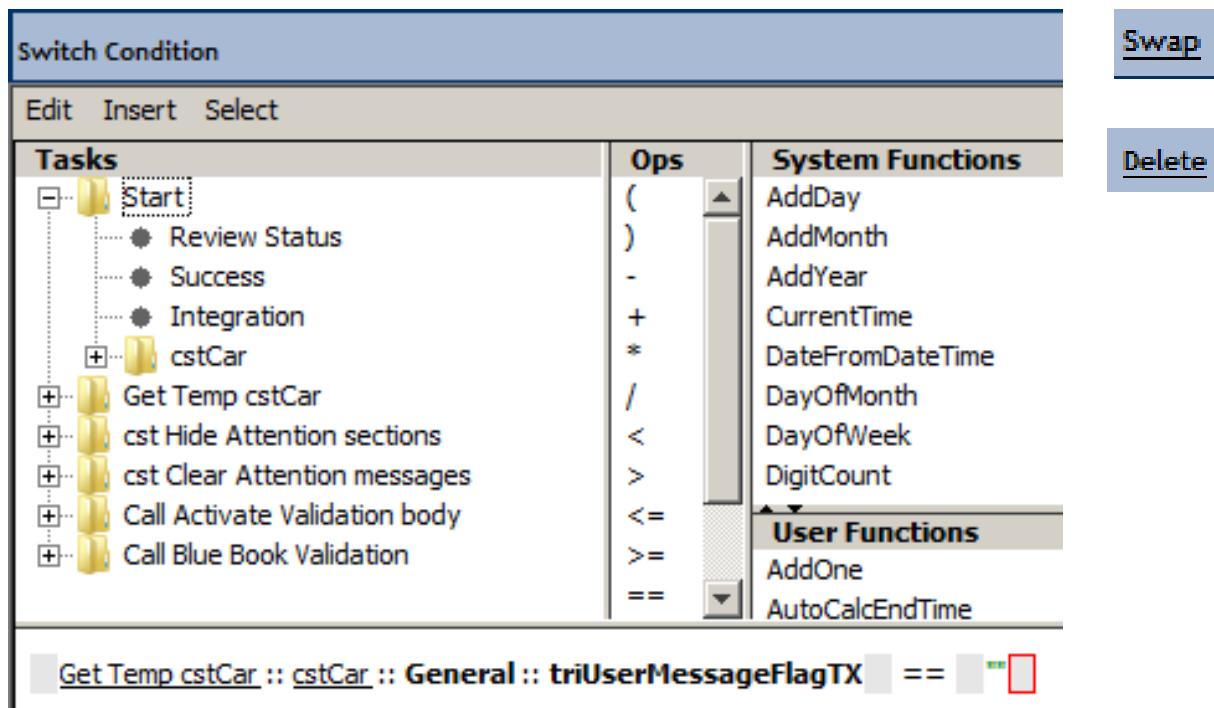
A special attribute that can be set at the root level of the Form Map is **Close Window**. This attribute overrides the Close option of the state transition that is defined for this instance of the workflow. Typically, this attribute is set to No in the *fail* path of a validation workflow to ensure that the form is not closed if the workflow created messages.

There are three ways to populate the map. You can type directly into the Value column, select from a list in the Value column, or select the check box in the Task column. When you select the check 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.

**Apply:** As with the Form Builder, you must click Apply after you change a property to save the change. If you change a setting and shift focus before you click Apply, the change is lost.

**Cancel:** Use this action to close the window when you are done mapping the metadata.

## Switch task properties



© Copyright IBM Corporation 2013

24

### Switch task properties

A Switch task contains two sequences of tasks. Based on conditions that are specified as properties of a Switch task, one or the other of the task sequences is performed. After the task sequence completes, the task after the switch is performed. The Switch task is much like the traditional *if* statement.

Unlike most tasks, the switch task does not have a Name or Description property. The label of the task is always Switch. There is no ability to select the record to work with because it does not work with a specific record. The only property that a switch task has is the condition, which determines the path that is taken.

The switch task is shown in the workflow diagram as a two-part figure. The top part of the switch is a triangle that points upward. It has a green circle below it on one side and a yellow box below it on the other side.

- The green circle identifies the **true** path for the switch, and the tasks below it are performed if the switch condition evaluates to true.
- The yellow square identifies the **false** path, and the tasks below it are performed if the switch condition evaluates to false.

The bottom part of the task is a triangle that points downward. Here both paths converge. You can click either the top or bottom part of the task to open its properties.



**Note:** Be careful when you delete a Switch task from a workflow, because deleting a switch task also deletes the task sequences that are contained in it.

The switch task has a Swap action. By default, the green circle is on the left of the task and the yellow box is on the right. Using the Swap action swaps the position of the circle and the square, but it leaves the task sequences where they were. After a Swap action, the task sequence that was attached to the green circle is attached to the yellow square. The task sequence that was attached to the yellow square is now attached to the green circle. In essence, the true and false paths are reversed.

Some types of tasks, such as the Switch task, use a condition to decide what they do. A condition is a comparison between values, and is either true or false. The properties of these tasks contain the Condition Builder, which is used to create and edit conditions.

The purpose of the Condition Builder is to build a formula in its bottom panel that is either true or false. You build the formula by using the Edit, Insert, and Select menus, along with the Tasks, Basic Ops, and System Functions panels. A description of these menus and panels is as follows:

- **Edit:** For cutting, pasting, and deleting portions of a condition.
- **Insert:** To add numbers or text values to the condition. When you add a text value, the Condition Builder automatically puts quotation marks around it.
- **Select:** Includes methods for selecting parts of the condition.
- **Tasks:** Is used to insert task attributes and fields of records into the condition. Each record that is produced by a task before the current task is shown as a folder in the tasks panel. You click the folder to expand the contents of the record. Sections that are contained in the record are also shown as folders. You click these folders to see the list of fields in the sections. You insert a field into the condition by clicking the field name.
- **Basic Ops:** Is used to insert operators into the condition. There are arithmetic operators, such as + and -, and comparative operators, such as <, >, and ==. There are also parentheses that can be used to group clauses in the condition and change the order of evaluation. Use the && (and) and || (or) operators to join conditions.
- **System Function:** Is used to insert system-defined and user-defined functions into a condition. The functions that you can select from the System Function panel are the same functions that are used for extended formulas.

The *Application Building for the IBM TRIRIGA Application Platform* guide has a comprehensive description of the Condition Builder. It also has a detailed example of how to build a condition.

## Get Temp Record task properties

The screenshot shows the 'Get Temp Task Properties' dialog box. At the top right is a 'Delete' button. The 'Label' field contains 'cst Get Temp cstCar'. The 'Description' field contains 'Get a copy of the record that the user has open in a form'. Below these fields is a section titled 'Records' with the following settings: 'Take the' dropdown set to 'Business Object', 'of Task' dropdown set to 'Start (cstCar)', and 'Object Type' dropdown set to 'cstCar'.

© Copyright IBM Corporation 2013

25

### Get Temp Record task properties

The purpose of the Get Temp Record task is to access the temporary data that is associated with a record. Temporary data is data that the user created or modified but did not save. This kind of task is in the task palette of the workflow editor only if the workflow is synchronous and allowed to access temporary data. After a Get Temp Record task successfully completes, it has the temporary data that is associated with the record that was used to start the workflow. In the **of Task** property selection list, this task lists only the Start task (unless it is in a DataConnect workflow).



**Note:** There is only instance of the temporary record. Each use of Get Temp Record for the same business object returns a connection to the same temporary record. To clarify, assume that there is a workflow that calls another workflow. Both use temporary data, and both use a Get Temp Record task to get the temporary record for the same business object. All changes that are made to the temporary data in the called workflow are seen by the calling workflow.

## Save Permanent Record task properties

The screenshot shows the 'Save Permanent Task Properties' dialog box. At the top right is a 'Delete' button. The 'Label' field contains 'CST Save Permanent cstCar'. The 'Description' field contains 'Save the temp record to the database'. Under 'Formulas', there is a note 'Disable Auto Recalculation'. In the 'Records' section, it says 'Take the Business Object of Task cst Get Temp cstCar (cstCar)' and 'Object Type: cstCar'.

© Copyright IBM Corporation 2013

26

### Save Permanent Record task properties

The Save Permanent Record task is used to update the permanent data of a record with temporary data that was 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. This kind of task is in the task palette of the workflow editor only if the workflow is synchronous and allowed to access temporary data. The **of Task** selection is always forced to be a Get Temp Record task.

## **State transitions and temporary data**

- If a workflow is triggered on a state transition and uses temporary data, it controls whether the record changes states
  - If the workflow performs a Save Permanent Record task, the record is saved, and the transition is completed  
The record is in the new state
  - If the workflow does not perform a Save Permanent Record task, the transition is halted  
The record is in the same state, and the record is not saved

© Copyright IBM Corporation 2013

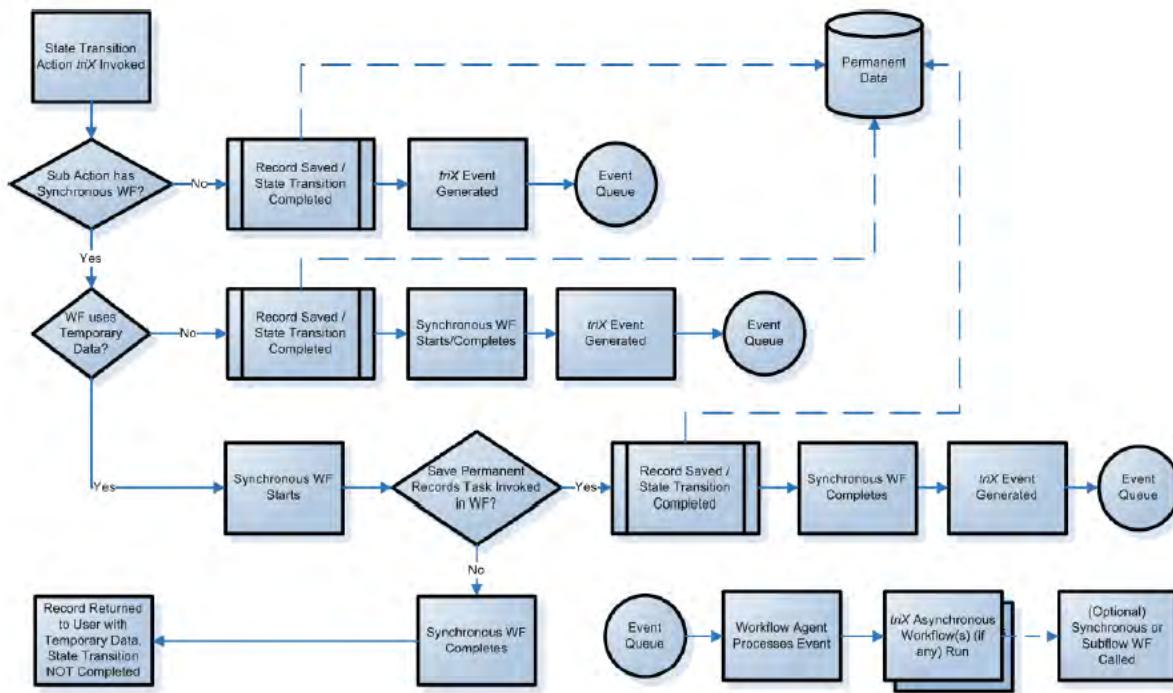
27

### *State transitions and temporary data*

A validation workflow can be triggered when the user clicks the Activate action. The workflow examines the contents of the record to determine whether it meets the business criteria that are defined for an active record. If it meets the criteria, the record can transition states. If the record does not meet the criteria, it must be prevented from changing states until its deficiencies are corrected.

If a workflow is triggered on a state transition and uses temporary data, it controls whether the record changes states. If the workflow performs a Save Permanent Record task, the record is saved and the transition completes. The record is returned to the user in the new state. If the workflow does not perform a Save Permanent Record task, the transition is halted and the record is not saved. The record is returned to the user in the same state.

## State transitions and temporary data flowchart



© Copyright IBM Corporation 2013

28

*State transitions and temporary data flowchart*

In general terms, if a workflow is triggered on a state transition and uses temporary data, it controls whether the record changes states. If the workflow performs a Save Permanent Record task, the record is saved and the transition completes. The record is returned to the user in the new state. If the workflow does not perform a Save Permanent Record task, the transition is halted and the record is not saved. The record is returned to the user in the same state.

More specifically, when a state transition occurs, you have the following decision points:

1. Decision point: Does this state transition have a sub action with a synchronous workflow?

If there is no synchronous workflow, then the record is saved and the state transition occurs. An event for the transition is put into the event queue.

2. Decision point: There is a synchronous workflow involved. Does it use temporary data?

If the workflow does not use temporary data, then the record is saved and the state transition occurs, and then the workflow begins. After the workflow completes, an event for the transition is put into the event queue.

A synchronous workflow is on this state transition, and it uses temporary data. Therefore, the workflow is started.

3. Decision point: Does the workflow use a Save Permanent Record task to save the temporary data? The workflow must contain a Save Permanent Record task, and the task must be performed for the answer to be Yes.

If the workflow performs a Save Permanent Record task, then at the time that the record is saved, the state transition occurs. The workflow then continues. After the workflow completes, an event for the transition is put into the event queue.

If the workflow does not perform a Save Permanent Record task, then the record is not saved and the transition does not occur. The workflow completes, and control is returned to the user. No event is put into the event queue because the transition did not occur.

## Message handling in workflows

- Workflow messages are stored in triUserMessageTX and displayed in Attention sections
- At the start of a workflow, the following changes occur:
  - Clears existing messages
  - Hides the attention section on all tabs
- When displaying error messages, the following changes occur:
  - Changes the attention section to visible on all tabs
  - Changes the Close Window setting to No for the form

© Copyright IBM Corporation 2013

29

### *Message handling in workflows*

By convention, synchronous workflows in the IBM TRIRIGA Application Platform that generate messages store them in the triUserMessageTX field for display to the user. This field is contained in an Attention section that is placed on each tab of a form. Such placement is done so that the user can see the message no matter what tab they are on.

At the start of a workflow that can generate a message, it is wise to perform some cleanup so that the workflow starts fresh. The message cleanup tasks of a workflow are as follows:

- Clear the message field (triUserMessageTX) so that messages from past runs are not seen. This clearing of messages is done with a Modify Records task that stores a null in the field. It is best to also clear the triUserMessageFlagTX field at the same time.
- Hide the Attention sections. Leaving message sections visible but without messages can cause the user to learn to ignore them. Hiding message sections by default and displaying them only when they have a message makes them more noticeable.

Hiding the Attention sections is done in a Modify Metadata task. Make sure to hide the Attention section on each tab where you find it.

When it is time for the workflow to display messages, use a Modify Metadata task to change all Attention sections to be visible. It is also important to change the properties at the form level so that the Close Window property is No. If the form closes, the messages are likely to be lost.

## Publishing a workflow

- A workflow is not available for use until you publish it
- You can publish a workflow in the following two ways:
  - Open the Start task of the workflow and click **Publish**
  - In Workflow Builder, select the workflow and click **Publish**
- A workflow in **Published** status is read-only
  - Click the **Revise** action to make the workflow editable

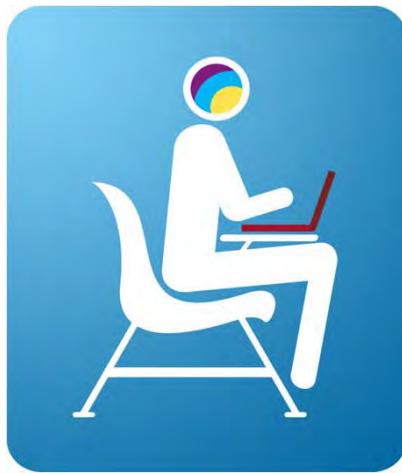
© Copyright IBM Corporation 2013

30

### *Publishing a workflow*

Workflows follow the same Create-Publish-Revise life cycle as business objects and other metadata. A new workflow must be published before it can be used. Changes to a workflow must also be published before they can be used. A workflow can be published in two different ways: Open the Start task of the workflow and use the Publish link, or from Workflow Builder, select the workflow and click Publish. 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, you click the Revise action.

## Student exercises



© Copyright IBM Corporation 2013

31

### *Student exercises*

Perform the exercises for this lesson.

# Lesson 2. Triggering workflows



## Lesson 2: Triggering workflows



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I) are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013. US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp. Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

There are many ways to trigger a workflow in the IBM TRIRIGA Application Platform, and this lesson describes the primary ones.

### What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Describe what workflows are
- Use the Workflow Builder tool
- Create and publish a workflow

### References

*Application Building in the IBM TRIRIGA Application Platform*

## Triggering asynchronous workflows

### Transition events

triCopy  
triCreate  
triCreateWizardHidden  
triDelete  
triInvalidUploadHidden  
triRemove  
triRetireHidden  
triSave  
triSaveAndClose  
triUnretire  
triUpdateClause  
triUpdateOption  
triUploadHidden  
triValidateUpload

### System events

Associate  
De-Associate  
WF Q Accept  
SCHEVENTCREATE  
SCHEVENTSTART  
SCHEVENTEND  
WF DELETE FROM MGR  
**WF WIZARD CANCEL**  
SYSTEM DC PROCESS JOB  
MEETING REQUEST  
ICALENDAR RECEIVED  
OUTLOOK EVENT

© Copyright IBM Corporation 2013

33

#### *Triggering asynchronous workflows*

Asynchronous workflows are triggered in response to an event that occurs on a business object. The particular event and business object combination that triggers an asynchronous workflow is defined in its Start task.

Two types of events can be generated to trigger an asynchronous workflow: state family transition events and system events. Asynchronous workflows are not directly connected to a state transition action. When a state transition action is triggered on a record, an event of the same name is put into the system event queue for that record.

System events happen to records that might or might not be the direct result of an action by a user. For example, when a record becomes disassociated from another record a system event is put into the event queue.

When an event is put into the event queue, the system checks the asynchronous workflows. If there is an asynchronous workflow that corresponds to the business object of the record and the event in the queue, the system triggers the workflow.

When triggered, asynchronous workflows are not run immediately. They are added to the Workflow Events queue and started when a thread is available. Asynchronous workflows that are triggered and are waiting to run can be seen in the administrative console, in Workflow Events.

Separate state transition actions of the same name generate a single event of the same name as the action. For example, clicking the triSave action from either the triDraft state or the triRevision state generates a single event called triSave. System events include the following events:

- **Associate**: Two records are associated with specified association.
- **De-Associate**: Two records are disassociated with specified association.
- **SCHEVENTSTART**: A time-based event is associated to a specified record starts.
- **WF USER LOGIN**: The user login event is triggered from the My Profile record of a user upon logging in.
- **WF USER LOGOUT**: The user logout event is triggered from the My Profile record of a user upon logging out.
- **WF WIZARD CANCEL**: The cancel action is clicked in a form of the specified type.

## Triggering synchronous workflows

- **Sub Action:** Workflow runs when the sub action is triggered
- **Pre-Create:** Workflow runs as a record is created
- **Pre-Load:** Workflow runs each time that a record is loaded into form
- **Field change (onChange):** Workflow runs when the field value is changed by the user
- **Form action:** Workflow runs when button is clicked

© Copyright IBM Corporation 2013

34

### Triggering synchronous workflows

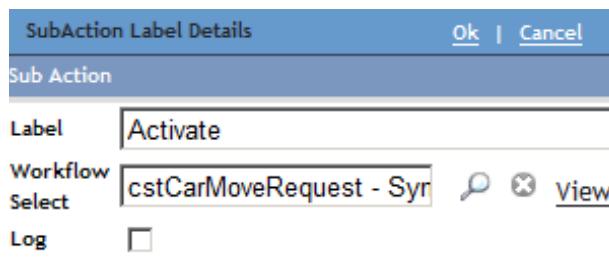
A synchronous workflow pertains to user actions. A synchronous workflow can be started in many ways, including the following ways:

- **From a sub action that is attached to a state transition action.** Unlike an asynchronous workflow, this ties the synchronous workflow directly to the state transition action.
- **From a form action.** Form actions include actions that are shown at the top of a section and buttons that are seen in the body of a form.
- **From a field action.** Field actions are triggered when the value of a field changes.
- **When a record is created.** This type of workflow is called a **Pre-Create** workflow. A Pre Create workflow runs only once in the life cycle of a record. The workflow is triggered after you click Add in a query, and it runs before the form shows the new record.
- **Before opening a form to edit a record.** This type of workflow is called a **Pre-Load** workflow. Such workflows can perform housekeeping or perform any last-minute changes of records.

You can make the start 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 start of a workflow are called the **start conditions**. You can use start conditions in both synchronous and asynchronous workflows.

## Attaching a workflow to a sub action

1. Open the Data Modeler, and select the business object
2. Click **Tools > BO State Transition** to open BO state family
3. Click the sub action 
4. Click **Open**
5. Use the picker to select the workflow to be run



6. Click **Ok**, and click **Save**

© Copyright IBM Corporation 2013

35

### Attaching a workflow to a sub action

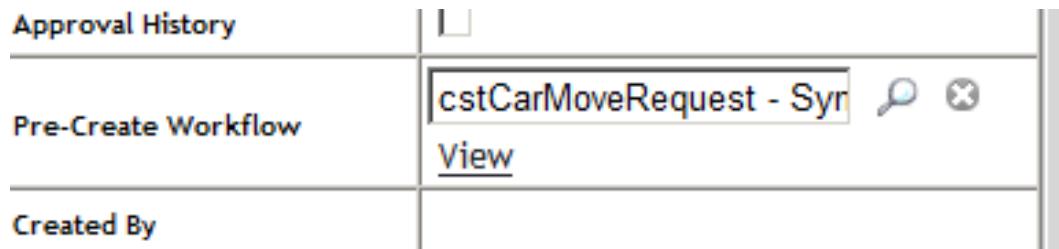
You can add a synchronous workflow to a sub action on a state transition by using the Data Modeler, in the state family of a business object.

The sequence of adding a synchronous workflow to a sub action is as follows:

1. Open the Data Modeler, and select the business object.
2. Click **Tools > BO State Transition** to open the state family of the business object. Find the transition where you want to attach a workflow. If it does not have a sub action, add one.
3. Click the sub action.
4. Click **Open**.
5. Use the picker next to the **Workflow Select** field to select the workflow to be run.
6. Click **OK**, and click **Save**.

## Setting a pre-create workflow in a business object

1. Open Data Modeler
2. Open selected business object
3. In BO properties, find **Pre-Create Workflow** property
4. Use picker to choose workflow
5. Click **Save BO**



© Copyright IBM Corporation 2013

36

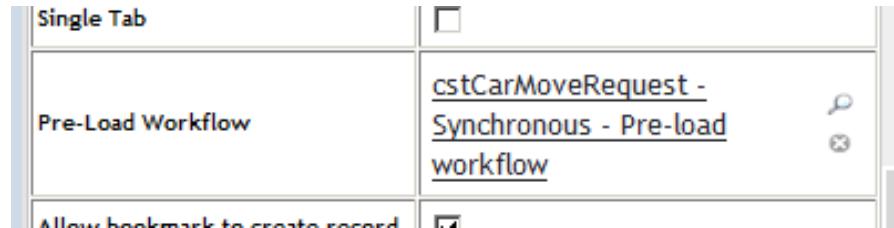
### *Setting a pre-create workflow in a business object*

A Pre-Create workflow is defined in the properties of a business object. Follow these steps to add a Pre-Create workflow to a business object:

1. Open Data Modeler.
2. Open the selected business object.
3. In the properties panel, find the **Pre-Create Workflow** property.
4. Use the picker to choose the synchronous workflow that you want to run.
5. Click **Save BO**.

## Setting a pre-load workflow in a form

1. Open Form Builder
2. Open selected form
3. In Form properties, find **Pre-Load Workflow** property
4. Use picker to choose workflow
5. Click **Apply**



© Copyright IBM Corporation 2013

37

### Setting a pre-load workflow in a form

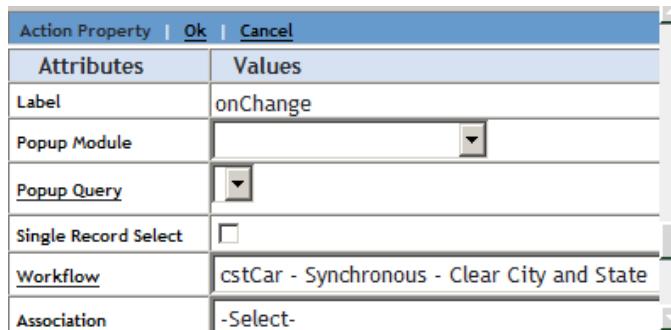
A Pre-Load workflow is defined in the properties of a form. Follow these steps to add a Pre-Load workflow to a form:

1. Open Form Builder.
2. Select the module of the form. Open the selected form. Revise the form if necessary.
3. In the form properties, find the **Pre-Load Workflow** property.
4. Use the picker next to the field to choose a workflow.
5. Click **Apply** to save the change.

The changed form must be published before the change takes effect.

## Attaching a workflow to a field

1. In the Form Builder, locate the field
2. Go to the bottom of the properties for that field
3. Add an onChange action
4. In the Workflow property, select the workflow to run
5. Click **Ok**
6. Click **Apply**
7. Publish the form



© Copyright IBM Corporation 2013

38

### Attaching a workflow to a field

If you want a workflow to run each time the value of a field changes, it must be specified in an onChange action for that field. You specific the action in the Form Builder.

Follow these steps to trigger a workflow in an onChange action for a field:

1. Open a form in the Form Builder. Revise the form if necessary. Bring focus to a field by clicking it, either in the Layout panel or the Navigation panel.
2. Go to the bottom of the properties of the field, to the Actions section.
3. Click **Add** to add an onChange action.
4. Find the Workflow property. Click the picker for this property and select the workflow to run.
5. Click **Ok** to save the action properties.
6. Click **Apply** to save the field properties.
7. Click **Publish** to publish the form.

## Workflow instances

- When a workflow is run, a copy of the workflow is made and the copy is run
- This copy is called an *instance*

Work Flow Name

1 [cstCarMoveRequest - Synchronous - Show Cost Section](#)  
2 [cstCarMoveRequest - Synchronous - Pre-load workflow](#)

List Active Instances | List All Instances

List Templates	
Record Name	Status
1 <a href="#">2004-Honda-Odyssey - 1000001</a>	Completed
2 <a href="#">2004-Honda-Odyssey - 1000001</a>	Completed

© Copyright IBM Corporation 2013

39

### Workflow instances

When a workflow is run, a copy of the workflow is made and the copy is run. This approach prevents a workflow that is running from being affected by changes in the definition of the workflow. The copy of the workflow and the details about its execution is called an *Instance*.

Workflow instances are not saved automatically. You can save workflow instances in the following two ways:

- Select the **Save Workflow Instances** check box in the properties of the workflow.
- Open the Workflow Agent Manager of the administrative console. Configure the Workflow Instance Recording property to have a value of **Always** or **As configured in the workflow**. Setting the value to As configured in the workflow saves only the instances of those workflows that have the Save Workflow Instances property set.

You can see the workflow instances in the following two places:

- In a form, click the **Work Flow Instance** tab to see the instances of workflows that ran against that record. This tab is visible only if the Show Workflow Instance property is selected in the form definition.

- In the Workflow Builder, select a workflow and click **List All Instances** to see all instances of this workflow that were run against any record.

When you click **List Active Instances** to show all instances of a workflow that are currently running, you see the **Work Flow Instance** tab with the name of each workflow that ran, the status of the workflow, and when the workflow started. There is no indication of what started the workflow.

The statuses that are shown in the Work Flow Instance tab are as follows:

- For synchronous workflows:
  - **Active**: The workflow is running.
  - **Completed**: The workflow completed.
- For asynchronous workflows:
  - **Aborted, Aborted-Warn**: The workflow was aborted by using the Stop capability in the Administrator console.
  - **Active**: The workflow is running.
  - **Completed, Completed-Warn**: The workflow completed.
  - **Failed**: The workflow encountered an error and could not continue. Information about the problem was written to the system log.
  - **Skipped**: The workflow was triggered, but the start conditions were not met.
  - **Stopped, Stopped-Warn**: The workflow used a Stop task.
  - **Waiting, Waiting-Warn**: The workflow is waiting for a user (for a user action or user approval task).

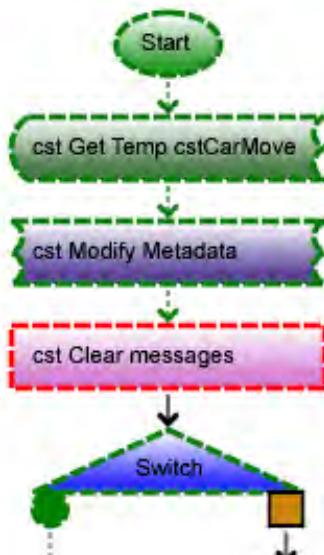
If **Warn** is on the status, a problem was encountered, information was written to the system log, and the workflow continued processing.

The workflow names in the Work Flow Instance tab are hyperlinks. If you click a workflow name, the workflow editor opens to show the workflow instance.

The IBM TRIRIGA Application Platform does not keep workflow instances indefinitely. The Cleanup agent discards workflow instances after a specified number of days elapse. The number of days that the platform keeps workflow instances is determined by the configuration properties of the platform.

## Execution path

- Green dashed line shows the path through the instance
- If a task has an error, its border is a red, dashed line



© Copyright IBM Corporation 2013

40

### Execution path

In a workflow instance, green dashed lines show the path that is taken through the workflow tasks. If a workflow task has an error, its border is a red, dashed line.

You can open the properties of any task in a workflow instance by clicking the task. If a task uses an edit map, you can click the Edit Map link to open a read-only copy of the edit map. Both of these abilities are useful when you diagnose problems in a workflow.

## Revising a workflow

- After you publish a workflow, it is read-only
- To change a published workflow, you must revise it
- You can revise a workflow in one of the following two ways:
  - In Workflow Builder, select the workflow and click **Revise**
  - Open the Start task of the workflow, and click **Revise**
- If the workflow uses temporary data, you must open the **Start** task for the Workflow Builder to show temporary data tasks in the New Task palette

© Copyright IBM Corporation 2013

41

### Revising a workflow

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 modify a published workflow, click the Revise action. This action creates a copy of the workflow and sets its status to Revision In Progress. While the status of a workflow is Revision In Progress you are can modify the workflow.

Revising a workflow also increments the version number of the workflow. The version number is displayed in the Revision column on the Workflow Builder page.

The Revise action is available in two places:

- From the Workflow Builder window, select a workflow and click the Revise action. This step opens the workflow in the Workflow Editor window.
- Open a workflow in the Workflow Editor window. Open the properties of the Start task of the workflow, and click the Revise action. This closes the properties window.

When the Workflow Editor opens, by default it does not include temporary data tasks in the New Task palette. If the workflow uses temporary data, you must open the Start task for the Workflow Editor to detect that setting. The Workflow Editor then shows the temporary data tasks in the New Task palette.



## Workflow version control

List Active Instances   List All Instances   List All Versions   Where Used		
Select	Name	Revision
<input checked="" type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	7
<input type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation body</a>	0
<input type="radio"/>	<a href="#">cstCar - Synchronous - Clear City and State</a>	1
<input type="radio"/>	<a href="#">cstCar - Synchronous - Editable update</a>	10

List Active Instances   List All Instances   List Templates   Where Used		
Select	Name	Revision
<input checked="" type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	7
<input type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	6
<input type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	5
<input type="radio"/>	<a href="#">cstCar - Synchronous - Activate Validation</a>	4

**Publish**

© Copyright IBM Corporation 2013

42

### Workflow version control

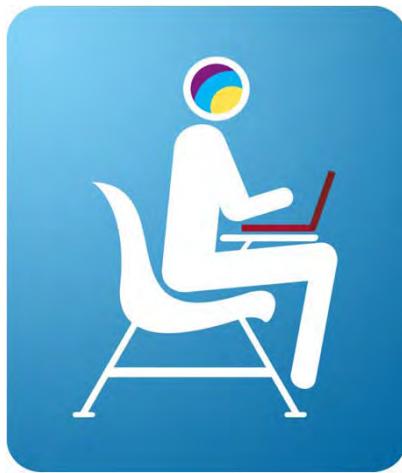
The platform retains old versions of a workflow when new versions are published. To select the workflow, you click List All Versions to view the history of revisions. The historical versions of a workflow are listed in descending order, with the most recent version first. Older versions of a workflow can be viewed and restored.

To restore a previous version of a workflow, select the version and click Publish. This action causes the following behaviors to take place:

- The current published version of the workflow is retired.
- The older version of the workflow is copied and its status is set to Published.
- The version number of the workflow is incremented.

For example, assume that the published version of a workflow is at Revision 7. If you select Revision 5 of that workflow and click Publish, Revision 7 is retired and a copy of Revision 5 is published as Revision 8.

## Student exercises



© Copyright IBM Corporation 2013

43

### *Student exercises*

Perform the exercises for this lesson.

## Checkpoint questions

1. Name and describe the two types of workflows.
2. Name and describe the two types of data.
3. You have a workflow task that must connect from one record to another record. If both options are possible, would you prefer to use **Use its reference** or **Use its association**, and why?
4. What activities do you perform at the start of a workflow that can issue messages?
5. If a user tries to activate a record, and the action triggers a workflow, and the data in the record fails validation, how can the workflow prevent the record from changing states?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. Name and describe the two types of workflows.

*Synchronous, which is triggered by user action.*

*Asynchronous, which is triggered by system event.*

2. Name and describe the two types of data.

*Permanent data: records that are stored in the database.*

*Temporary data: records that are opened in a form for the user.*

3. You have a workflow task that must connect from one record to another record. If both options are possible, would you prefer to use **Use its reference** or **Use its association**, and why?

*Use its reference is faster and therefore preferred.*

## Checkpoint answers, continued

4. What activities should be performed at the start of a workflow that can issue messages?

*The message fields should be cleared by using Modify Records, and the Attention sections that are hidden by using Modify Metadata.*

5. If a user tries to activate a record, and the action triggers a workflow, and the data in the record fails validation, how can the workflow prevent the record from changing states?

*If the workflow does not perform a **Save Permanent Record** task, the record does not move to another state.*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Define workflows
- Use the Workflow Builder tool
- Create and publish a workflow





## 16 Advanced queries

---



## 16 Advanced queries



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

---

### What this unit is about

This unit is about the advanced features that are available for the reporting tool that is built into the IBM TRIRIGA Application Platform.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform Reporting Guide*

## Objectives

After completing this unit, you should be able to perform the following tasks:

- Put a custom Find query in place
- Create an Editable query
- Set up related reports
- Use association filters

# Lesson 1. Using advanced queries



## Lesson 1: Using advanced queries



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

This lesson is about the advanced features that are available for the reporting tool that is built into the IBM TRIRIGA Application Platform.

## What you should be able to do

After completing this lesson, you should be able to perform the following tasks:

- Put a custom Find query in place
- Create an editable query
- Set up related reports
- Use association filters

## References

*Application Building in the IBM TRIRIGA Application Platform*

*Application Building in the IBM TRIRIGA Application Platform Reporting Guide*

## Using Find queries

- When you click Find in a smart section of a form, a default query is used to create a list of records to select from
- The default query might not be appropriate for your needs:
  - You might want to restrict records that go into the list
  - You might want to provide runtime filtering capabilities
- You can solve this problem by creating a customized query and having the platform use that query instead of the default query for the Find action

© Copyright IBM Corporation 2013

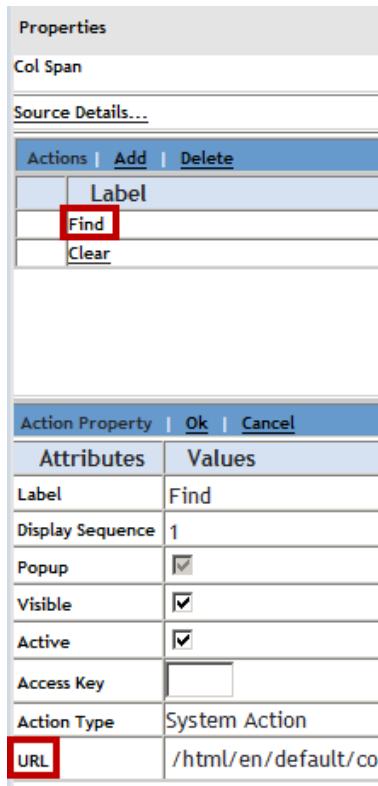
4

### Using Find queries

When you click Find in a smart section of a form, a default query is used to create a list of records to select from. The default query might not be appropriate for your needs. For example, you might want to restrict records that go into the list, or you might want to provide runtime filters. You can solve this problem by creating a customized query and having the platform use that query instead of the default query for the Find action.

## Replacing a default Find query

1. Open a form
2. Select the smart section that you want to change
3. Scroll down to the Actions section of the properties panel
4. Click the **Find** action to open its properties
5. Click the picker for the URL field
6. Select the query that you want to use
7. Click **Ok**
8. Click **Apply**



© Copyright IBM Corporation 2013

5

### Replacing a default Find query

Follow these steps to replace a default Find query with a different Find query:

1. Open a form.
2. Select the smart section that you want to change.
3. Scroll down to the Actions section of the properties panel.
4. Click the **Find** action to open its properties.
5. Click the picker for the URL field.
6. Select the query that you want to use.
7. Click **Ok**.
8. Click **Apply**.

## Editable queries

- User can edit values directly from the report
- User can perform actions on selected records

Limitation: All records must be in the same selected state

Courses - Draft

Add To Bookmarks | **Activate** | **Save** | Export | Cancel

Date : 07/08/2013 08:40:42

### Courses - Draft

1 total found				
	Course Number	Course Title	Credit Hours	Fee
<input type="checkbox"/>	 301	Really advanced topics	0.3 hours	

© Copyright IBM Corporation 2013

6

#### Editable queries

You can use editable queries to edit records directly in the query without having to open the record in the form. In editable queries, you can use actions on multiple selected records. The actions that are available must be appropriate for all records that are displayed. Because of this requirement, records in Editable queries are limited to a single state and a single form. This limitation ensures that the same state family is used by all of the records, and that the available actions are appropriate for all records in the query.

The name of an editable query includes the form name, state, and description. An example of an editable query name is *cstCourse - triDraft - Courses in Draft status*.

## Creating an editable query

- Defined on the **Options** subtab of the **General** tab
- Set columns to editable on the **Columns** tab
- Use **Import State Actions** on **Advanced** tab to add actions

The screenshot shows the Business Objects interface with three main panels:

- Business Objects** panel:
  - Random Result Count: 0
  - Fixed Column Count: 0
  - Result Size: ▾
  - Excel Template:
  - Prompt Before Query:
  - Editable**: A group of radio buttons. The first one, "Yes", is selected and highlighted with a red box.
  - State**: A dropdown menu showing "triDraft".
- Editable** column: A vertical list of items, each preceded by a checkmark in a column header.
- Import State Actions** dialog:

Action	Next State
cstUpdate	triDraft
triActivate	triReview
triCopy	triDraft
triDelete	null
triInvalidUploadHidden	triUploadError
triSave	triDraft
triSaveAndClose	triDraft

© Copyright IBM Corporation 2013

7

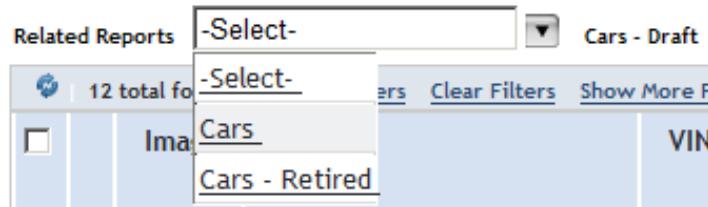
### Creating an editable query

Follow these steps to set the properties that make a query an editable query:

- On the **Options** subtab of the **General** tab, select the following choices:
  - Select **Yes** for the **Editable** property.
  - Select the state for the query in the **State** property. The states available for selection are based on a combination of the business object and form. An editable query is limited to a single business object and a single form.
- On the **Columns** tab, you see a column of check boxes under the **Editable** heading in the **Display Columns** section. Selecting a check box makes that field editable.
- On the **Advanced** tab, click the **Import State Actions** button to see the actions that are available for the selected state in the state family of the form.
- Select the check box next to the ones that you want to make available to the user for this query, and click **Ok**. Arrange the order of the buttons as needed.
- Click **Save** to save the query.

## Related reports

- A Related Report defines a connection to another report  
To display more information the user might find relevant
- A report can have more than one related report
- A report that is a related report typically contains a reference back to the main report



© Copyright IBM Corporation 2013

8

### Related reports

A related report is a self-contained report that has its own definition and format. Its usual purpose is to display extra information that the user might find relevant. A report can have more than one related report. Generally, a related report contains a reference back to the main report.

## Defining related reports

Defined on the **Related Reports** subtab of the **General** tab

The screenshot shows the 'Related Reports' subtab in the Report Builder. The 'Add' button is highlighted with a red box. Below it, two reports are listed:

Select	Order	Sequence	Name	Header	Type
<input type="checkbox"/>	▼	1	cstCar - Master Detail Query	Cars	Query
<input type="checkbox"/>	▲	2	cstCar - triRetired - Editable	Cars - Retired	Query

Below this is a 'Reports List' window titled 'Reports List - IBM TRIRIGA - Windows Internet Explorer'. It has four numbered arrows indicating steps:

- Arrow 1 points to the 'Module' dropdown set to 'cstFleetManagement'.
- Arrow 2 points to the 'All' filter icon.
- Arrow 3 points to a report entry: 'cstCar - Cars I supervise which have move requests'.
- Arrow 4 points to the 'OK' button.

	C	Name	ID	Business Object	Created By
<input type="checkbox"/>	C	cstCar - Cars I supervise which have move requests		cstCar	Iverson, Don - 1000000
<input type="checkbox"/>	C	cstCar - Cars and people		cstCar	Iverson, Don - 1000000

Report types sidebar:

- All
- Graphic
- Query
- Report

### Defining related reports

Related reports are specified in the Related Reports subtab of the General tab in the Report Builder.

To add a Related Report, you use the following steps:

1. Click **Add** to open the Reports List window.
2. Select the module. The list contains only those modules for which you have security access.
3. Select the type of report you want to see. To see all types of reports, select **All**.



**Note:** By default, only reports of type Report are listed. To see other report types, click the Filter icon, which looks like a funnel.

The reports that match the selected report type and module are displayed.

4. Select the check boxes of reports to use as related reports.

5. Click **OK**.

Related reports are presented for selection in the same order as shown in the Related Reports subtab, from top to bottom. To change the sequence of the reports, click the arrows in the Order column to move a related report up or down in the list. Click **Save Order** if you change the order of the reports. To delete a related report, select the check box and click **Remove**.

## Association filters

- Filters records based on associations with other records  
Not based on the contents of the records
- Can use single record or results of a query as input
- Defined on the **Advanced** tab

Association Filters								<a href="#">Add</a>   <a href="#">Delete</a>
Select	Association Type	Module	Business Object	Filter Type	Record	Conditional		
<input type="checkbox"/>	<a href="#">Supervises</a>	triPeople	triPeople	Record	\$\$USERID\$\$	false		AND
<input type="checkbox"/>	<a href="#">Has</a>	cstFleetManagement	cstCarMoveRequest Query	- Master Detail Car Move Request				

© Copyright IBM Corporation 2013

10

### Association filters

Another type of filter that you can use in reports is an Association filter. These filters exclude records from a report that is based not on the contents of the records, but on the relationships between records. You define association filters on the Advanced tab.

An association filter is defined from the perspective of one or more filtering records. You specify the filtering record and the association to use, and only those records with that association from the filtering record are retained for the report.

You can define multiple association filters for a report. They are joined by the AND condition.

You can also make the following choices in the Association Filters section:

- Select a filter and click Delete to remove it.
- Click the link in the Association Type column to edit the properties of an association filter.
- Click Add to add an association filter.

## Defining an association filter

Add Associations      [OK](#) | [Cancel](#)

Module	triPeople
Business Object	triPeople
Association Type	Supervises
Reverse Association	No
Filter Type	Record
Record / Query	<input type="text" value="\$\$USERID\$\$"/> 

© Copyright IBM Corporation 2013

11

### Defining an association filter

To create an association filter, you perform the following steps:

1. In the Association Filters section, click **Add**.

The Add Association window opens, where you define the filtering record and the association for the filter.

2. Select the module and business object of the filtering record. To include all business objects in the selected module, select **ALL**.

To avoid potential report performance issues, avoid using **-Any-** for the module selection and **All** for the business object selection whenever possible.

3. Select the **Association Type**.

This value is the association type that is defined from the perspective of the filtering module and business object. To include all association types, select **ALL**.

4. Choose one of the following options:

- To restrict the records that are selected by the filter to only the ones with the specified association type from the filtering record, select **No** for **Reverse Association**.

- To allow records with the association type on either end of the association, select **Yes** for **Reverse Association**.

You must select Yes for Reverse Association if an association exists only from the records in the report to the filtering record, and not in the reverse direction. Reverse Association can also be used as a diagnostic tool. If you are not seeing the results that you expect from the association filter, select Yes for Reverse Association. If you then get the results you expect, then you used the wrong association string. Use a reverse association only when functionally necessary. Selecting Yes for Reverse Association might return more records and take more time to do so.

5. Select the Filter Type and Record / Query properties to specify the filtering records:

- Specify a single filtering record by selecting **Record** for the Filter Type value and entering a keyword in the Record / Query property.
- Specify multiple filtering records by selecting **Query** for the Filter Type value and then selecting a query in the Record / Query property.

6. Click **OK**.

7. Click **Save** to save the report.

## Association filter keywords for Record filters

- **\$\$USERID\$\$** filters by the user who is logged in
- **\$\$RECORDID\$\$** filters by the record that is open in the form
- **\$\$PARENT\$\$** uses the reference to the find filtering record:
  - **\$\$PARENT::SectionName\$\$** - for smart sections
  - **\$\$PARENT::SectionName::FieldName\$\$** - for Locator fields

© Copyright IBM Corporation 2013

12

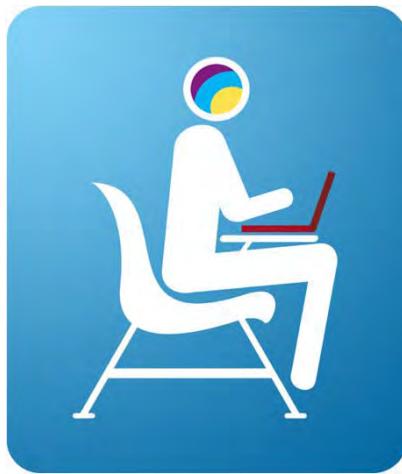
### Association filter keywords for Record filters

Some of the keywords to use with a Record association filter are as follows:

- **\$\$USERID\$\$**: Filters by the user who is logged in. Using this association filter is a way to make a report that customizes itself for each user. If the Record field is set to **\$\$USERID\$\$**, set the Module field to triPeople and the Business object field to All.
- **\$\$RECORDID\$\$**: Filters by the record that is open in the form. The filtering record is the record that is open in a form. This keyword is useful for queries that are used in query sections and for Find action queries in data sections. Queries that are used in query sections are often required to return only those records that are associated with the record that contains the section. The same applies to custom Find queries. Setting the Record field to **\$\$RECORDID\$\$** means that the records pass this filter if they are associated directly with the context record.
- **\$\$PARENT\$\$**: These keywords use a reference in the current record to find the filtering record as follows:
  - **\$\$PARENT::SectionName\$\$**: For smart sections. SectionName is the name of the section, like **cstCar**. Ensure that you use the name of the section and not the label.

- **\$\$PARENT::SectionName::FieldName\$\$:** For Locator fields. SectionName is the name of the section, like **cstCar**. FieldName is the name of a locator field in that section.

## Student exercises



© Copyright IBM Corporation 2013

13

### *Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

1. Why are records in an Editable query limited to one state?
2. You have a query that displays cstCar records. There is an association from cstCar to triPeople with the string Supervised By, and a reverse association with the string Supervises. Which string is used in an association filter that uses the \$\$USERID\$\$ keyword? Why?
3. In an association filter, what is a keyword in the form of \$\$PARENT::SectionName::FieldName\$\$ used for?

### *Checkpoint questions*

Enter your answers here:

- 1.
- 2.
- 3.

## Checkpoint answers

1. Why are records in an Editable query limited to one state?  
*To ensure that the action buttons are appropriate to all records displayed in the query.*
2. You have a query that displays cstCar records. There is an association from cstCar to triPeople with the string Supervised By, and a reverse association with the string Supervises. Which string is used in an association filter that uses the \$\$USERID\$\$ keyword? Why?  
*Supervises. The string is from the associated business object to the business object displayed in the report.*
3. In an association filter, what is a keyword in the form of \$\$PARENT::SectionName::FieldName\$\$ used for?  
*It uses a locator field reference to find the filtering record.*

## Summary

---

Now that you have completed this unit, you should be able to perform the following tasks:

- Put a custom Find query in place
- Create an Editable query
- Set up related reports
- Use association filters



## 17 Object migration: Export



## 17 Object migration: Export



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this unit is about

Object Migration is the process of exporting objects from the IBM TRIRIGA Application Platform environment into a flat file, and then importing them into another environment. This unit covers the export process.

### How you check your progress

You can check your progress in the following ways:

- Review questions
- Lab exercises

### References

*Object Migration User Guide*

## Objectives

---

After completing this unit, you should be able to export selected objects into an Object Migration package

# Lesson 1. Exporting objects



## Lesson 1: Exporting objects



All files and material for this course (TOD18, IBM TRIRIGA 10.3 Application Platform I)  
are IBM copyright property covered by the following copyright notice. © Copyright IBM Corporation 2013  
US Government Users Restricted Rights: Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.  
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

### What this lesson is about

Object Migration is the process of exporting objects from the IBM TRIRIGA Application Platform environment into a flat file, and then importing them into another environment. This lesson covers the export process.

### What you should be able to do

After completing this lesson, you should be able to export selected objects into an Object Migration package.

### References

*Object Migration User Guide*

## Object migration

### Definition of an object migration

- Applications are composed of Objects, such as modules, business objects, forms, workflows, reports
- Object Migration is the process that is used to transfer these objects from one environment to another
  - From development to test to production environments
- Object migration is used by IBM TRIRIGA for Application and Platform upgrades

© Copyright IBM Corporation 2013

4

#### *Object migration*

After you develop or customize an application, you want to migrate your work (business objects, forms, workflows, and so on) to other environments. This migration might be done for testing, or to put the items into production. IBM TRIRIGA Application Platform has an element to select objects and write them to a file. There is also a corresponding element to read objects from a file into an IBM TRIRIGA Application Platform environment. Both of these elements are part of a tool in the platform named **Object Migration**. Object Migration is the same process that is used to distribute upgrades to the IBM TRIRIGA applications and platform.



**Important:** Do not allow users to access the IBM TRIRIGA environment while you are processing object migration packages. User activity can result in data corruption or locked tables.

## Object migration (continued)

- Object migration involves exports and imports:
  - Export specifies a set of objects to be written out to a .zip file
  - Import loads objects from a .zip file
- Click **Tools > Administration > Object Migration**

Export Packages		
Name	Status	User
<a href="#">Cert Package 2</a>	Exported	Iverson, Don - 1000000
<a href="#">PULSE Lab Workflow</a>	Exported	Iverson, Don - 1000000
<a href="#">cert class stuff</a>	Exported	Iverson, Don - 1000000
<a href="#">one workflow</a>	Exported	Iverson, Don - 1000000
Import Packages		
Name	Status	User
<a href="#">DMO Platform Certification Ver 10.0 4-28-2011</a>	Imported	Iverson, Don - 1000000
<a href="#">Grades 10.0</a>	New	Iverson, Don - 1000000
<a href="#">cstSmartSectionDemo_With_Link_Disabled Ver 10.0</a>	Imported	Iverson, Don - 1000000

© Copyright IBM Corporation 2013

5

### Object migration (continued)

The Object Migration process includes imports and exports. An export specifies a set of objects to be written out to a file so that it can be read by other IBM TRIRIGA Application Platform environments. The objects are put into a special compressed (.zip) file. If you disturb the control information, then the file cannot be used by the Object Migration process. For this reason, it is not a good idea to manipulate the contents of the export file. An import contains objects from a file to be migrated into this platform environment. The means of transferring the object migration file from one environment to another is up to you. It might be something that you accomplish by yourself, or another person might be involved.

The Object Migration page is accessed through **Tools > Administration > Object Migration**. The Object Migration page shows all object migration packages, with the status of each package, the user who created the package, and the description of the package. The packages are separated into an Export Packages section and an Import Packages section. Packages in each section are sorted alphabetically by name.

The status of the object migration package can be New, Validated, Export Pending, Exported, Imported, Validation Pending, or Validation Failed. To see the specifications of an Object Migration package, click the hyperlinked name. To delete an Object Migration package, select the check box

to the left of the package name and click Delete. If you want to create an export package that contains the same items as an existing package, click the **copy package** icon next to the package. Only existing objects are put into the new package.



**Hint:** Use the copy package feature to re-export a set of objects that you previously imported. For example, use the copy package feature to import a package into an environment, change those objects, and then migrate them somewhere else.

When you click the copy package icon, the system opens the Create Package window. Change the name as appropriate and enter a description. The name of a package cannot contain special characters. The Create mode for a copied package is preset to *From Package*.

## Exporting

Use the following steps in exporting objects:

1. Create an export package
2. Select objects to include in the export package
3. Export the package to write the selected objects to a file

© Copyright IBM Corporation 2013

6

### *Exporting*

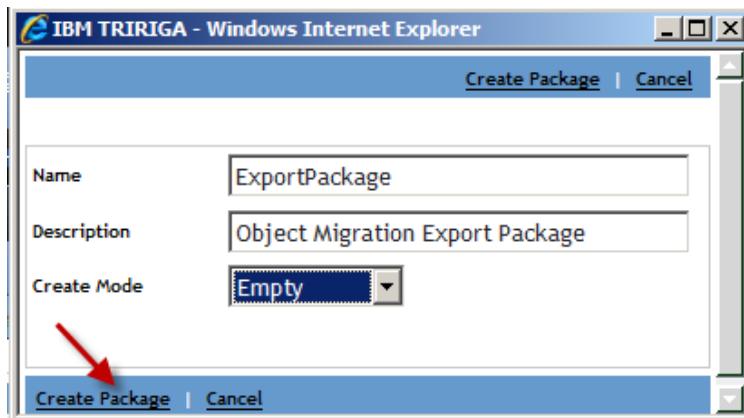
Exporting is the process of taking objects that you want to migrate to other IBM TRIRIGA Application Platform environments and writing them to a file.

There are three basic steps in exporting objects:

1. Create an export package.
2. Specify what objects to include in the export package. This step is the most complicated part of the process. It involves including the correct objects in the package. It also might involve excluding objects from the package.
3. Use the export package to write the specified objects to a file. If you want to migrate newer versions of the same objects that you previously migrated, you can reuse a previous package and repeat this step.

## Creating an export package

1. From the Object Migration page, click **New Export Package**
2. Enter a name and description for the package
3. Select one of the following values for **Create Mode**:
  - **Empty**
  - **Full Package**
  - **By Date**
4. Click **Create Package**



© Copyright IBM Corporation 2013

7

### Creating an export package

Follow these steps to create an export package:

1. From the Object Migration page, click **New Export Package**.  
The Create Package window opens.
2. Enter a **Name** and a **Description** for the new export package.
3. Select a value from the Create Mode list.
  - If you select **Empty**, the package does not contain any objects.
  - If you select **Full Package**, the package contains **ALL** objects from the entire system, except for record data. The system uses a recursive find to make sure that the Full Package is complete and includes all levels of dependents.

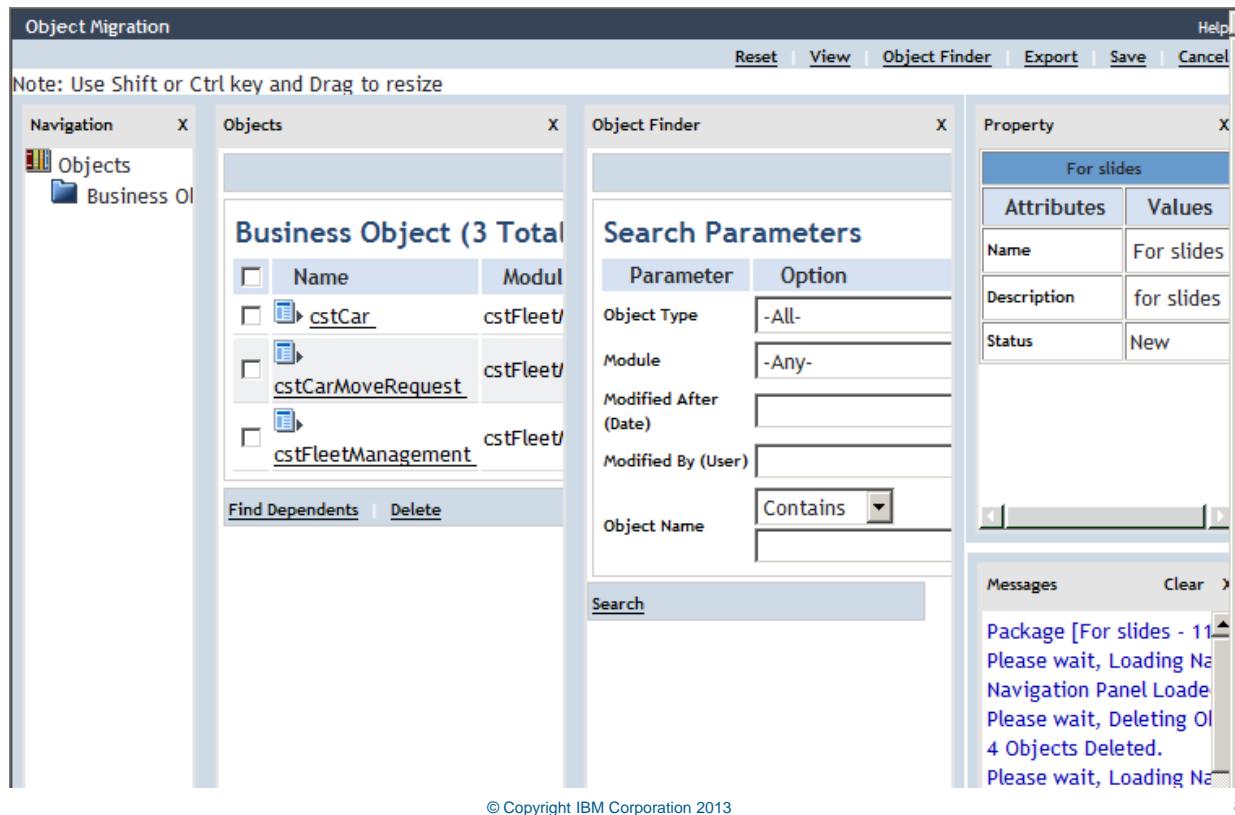


**Note:** Expect the Full Package Create Mode to take a long time to complete.

- If you select **By Date**, the package contains objects from the entire system that were published on or after a date that you specify. The system uses a recursive find to capture a single level of dependents. A *By Date* package can contain a larger number of objects than you might expect.
4. Click **Create Package**.

This action creates the package and opens the Object Migration window.

## Object Migration export window



Object Migration export window

The Object Migration window for export packages is like the window for import packages. The window is divided into the following panels:

- **Navigation:** Shows what objects are included in the export package. Unlike the import window, none of these objects is color-coded.
- **Objects:** Presents what is included in the selected item in the navigation panel and display objects. Unlike the import window, none of these objects is color-coded.

The following two actions are at the top of this panel:

- **Find Dependents:** Select an object and use this action to search for dependents of the object.
- **Delete:** Remove objects from the export package by selecting them and using this action.
- **Object Finder:** Is used to look for objects to include in the package. It has extensive search capabilities.

The **Search** action is at the top of this panel. Use this action to search for objects.

- **Property:** Contains the definition of the export package.

- The **Messages**: Displays what the system is doing with the export package.

The **Clear** action is at the top of this panel. Clicking Clear removes all messages.

The actions on the top of the Object Migration window are as follows:

- **Reset**: Returns the Object Migration screen to the state it was in when last saved.
- **View**: Used to restore one of the panels in the Object Migration screen.
- **Object Finder**: Clears the information in the Object Finder panel.
- **Export**: Exports the selected objects into a file.
- **Save**: Saves the information in the Object Migration screen.
- **Cancel**: Closes the Object Migration screen. Unsaved changes are lost.

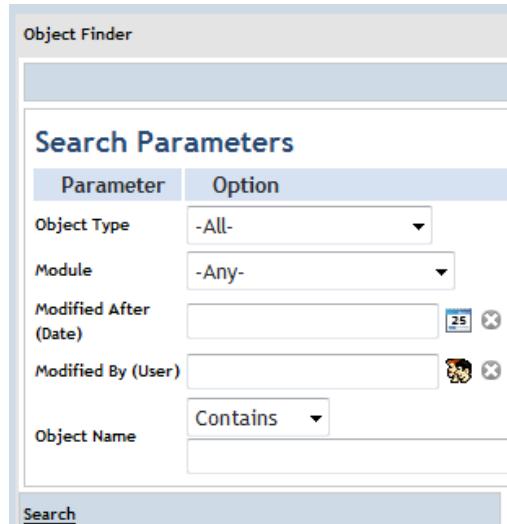
The export package now exists. If you selected either Full Package or By Date as the value for the Create Mode, the package contains the objects that are described earlier. If you selected Empty for the Create Mode, the package is empty.

Review the contents of the export package, by clicking an object type in the Navigation panel and seeing the list of objects in the Objects panel. If the export package contains any objects that you do not want in your object migration file, select them and click the **Delete** action.

## Selecting objects to be exported

Use Object Finder to search for objects

- Related to a module
  - Module, BO, form, query, list, record data, workflow
- Not related to a module
  - Portal, portal section, navigation collection, navigation item
- By user
- By name



© Copyright IBM Corporation 2013

9

### Selecting objects to be exported

The next step in the creation of an export package is to search for objects and add them to the package. You search for objects by setting search parameters in the Object Finder panel and then clicking the Search action. The search parameters can be used individually or in combinations.

You can use the following search parameters:

- **Object Type:** Select a type of object to search for. You can select only one object type at a time. Use the default of **-All-** to search for all object types.
- **Module:** Select a specific module to search for objects that are associated with that module. You do not see the Module field for object types that are not dependent on modules. Those object types include Navigation Collection, Navigation Item, Portal, and Portal Section.
- **Business Object:** If you select a module that has business objects, the system displays a Business Object parameter and selection list. The list contains the business objects for the selected module. Select a business object from the list to limit the search to those objects that are associated with the selected business object. Use the default of **-Any-** to search for objects that are associated with any business object in the module.

- **Modified After:** To restrict the search to objects modified on a date or later, click the Calendar icon next to Modified After (Date) and select a date. If you leave the Modified After (Date) blank, the system returns objects regardless of when they were last modified.

- **Modified By:** To restrict the search to objects modified by a specific user, click the People icon next to Modified By (User) field and select from the choices. If you leave Modified By (User) blank, the system returns objects regardless of which user last modified them.

Some object types do not support a *modified by* search. For those object types, the system does not return any objects if you identify a user in the Modified By (User) field. The supported object types for the Modified By (User) field are Query, Workflow, Document, Group, Navigation Collection, Navigation Item, and Record Data.

- **Object Name:** To further refine the search, enter some or all of the object name and select the filter (Equals, Not Equals, or Contains). For example, Contains Building finds objects that contain the word Building in their name. This search parameter is not case-sensitive.

## Adding objects to export package

- Select objects
- Click **Add Selected Objects**
  - Adds selected objects to package
  - Triggers search for dependents

<input checked="" type="checkbox"/> List (2)	Module	Description
<input checked="" type="checkbox"/> cstMake	cstFleetManagement	
<input checked="" type="checkbox"/> cstModel	cstFleetManagement	
<input type="checkbox"/> Module (2)	Description	
<input checked="" type="checkbox"/> cstFleetManagement		
<input type="checkbox"/> cstSmartSectionDemo		

© Copyright IBM Corporation 2013

10

### Adding objects to export package

After you click Search, the Search Results section of the Object Finder panel shows the objects that match the values in the Search Parameters section. The objects are grouped by object type and sorted alphabetically within each type.

To add objects that are listed in the Search Results section to the export package, select the check boxes of the items and click Add Selected Objects. Selecting the check box at the top of a section selects all objects of that type. You can then clear the check box for any of the objects that you do not want in the package. When the system finishes adding objects, the objects are shown in the Navigation panel. The system also displays any dependent objects in the Object Finder panel.



**Note:** You do not have to add anything to the Object Migration package to use the Object Finder. Set the initial search parameters and click Search. Click Find Dependents. In the resulting list of dependents, find and select the one of interest. Click Find Dependents. At any time, you can click the hyperlinked name to see the definition of the object in the system. Continue to iterate through each layer of dependents until you find exactly what you want.

## Adding dependent objects to export package

- Review dependents
- Select the ones that you want in the package
- Click **Add Selected Objects**

### Search Results

Business Object (16)	Module	Description
<input type="checkbox"/> <a href="#">Classification</a>	Classification	Dependent [Base BO] for [cstCarCategory Business Object].
<input type="checkbox"/> <a href="#">cstCar</a>	cstFleetManagement	Dependent [cstCar section's reference BO] for [cstCarMoveRequest Business Object].
<input type="checkbox"/> <a href="#">cstCarCategory</a>	Classification	Dependent [BO's Include BO] for [cstCarCategory Business Object].

© Copyright IBM Corporation 2013

11

#### Adding dependent objects to export package

The Search Results panel shows the dependents of the objects from your search, grouped by object type. There is a Description column that gives the dependency of each object. Select the dependents that you want included in the package and click Add Selected Objects.

You can use either of the following two approaches for adding dependents:

- Add all of the objects and then find dependents.
- Add an object and find its dependents, add another object and find its dependents, and so on.

Use the method that best matches your objectives and familiarity with your IBM TRIRIGA implementation.



**Note:** If you attempt to add the same object to the package more than once, the system ignores the subsequent attempts. The object is added only once.

## Exporting the objects to a file

1. Click **Export**
2. When prompted, click **Wait** or **Background**

Would you like to [[Wait](#)] for the Export or let the system handle it in the [[Background](#)]?

- **Wait**
  - You must wait for file to be generated
  - You specify where to save the file, when prompted
- **Background**
  - You can perform other tasks while file is generated
  - File stored in **userfiles\ObjectMigration** directory on server

### Exporting the objects to a file

When an export package contains the objects that you want, you are ready to use it to create the object migration file. Follow these steps to export the objects into a file:

1. Click **Export**.
2. In the Objects panel, click **Wait** if you want to wait for the file to be created, or **Background** if you want to run it in the background. IBM TRIRIGA Application Platform generates a .zip file containing .xml for each of the objects in the package. This file is used to transport your changes from one environment to another.
  - If you clicked **Wait**, the system generates the file and prompts you to download it. Click **Save**. Select the location where you want to store the file, and click **Save**.
  - If you clicked **Background**, the file is saved in the **userfiles\ObjectMigration** directory of the IBM TRIRIGA installation. For example, C:\Tririga\userfiles\objectMigration\My Export Package.zip.

While the Object Migration agent is working on the Export package, you cannot change the definition of the package. The actions that are available in the Object Migration window change

because it is now read-only. You can look at what is in the package, but you cannot search, add, or delete objects.

After the .zip file is downloaded, the status of the Object Migration package changes to Exported, and the system sends a notification that the Export is complete. You are now ready to migrate these objects into other IBM TRIRIGA Application Platform environments.

## Instructor demonstration

- Exporting package creation
- Adding objects to export package
- Exporting objects to file

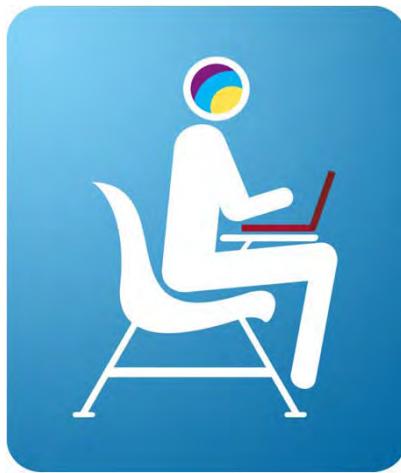


© Copyright IBM Corporation 2013

13

*Instructor demonstration*

## Student exercises



© Copyright IBM Corporation 2013

14

### *Student exercises*

Perform the exercises for this unit.

## **Checkpoint questions**

---

1. What is the purpose of exporting objects?
2. Where is the export file stored if you select Wait?
3. What is a key to the object migration export process?
4. What happens if you select an object to be in the package more than once?
5. What can happen if you open the export file and change it?

### *Checkpoint questions*

Put your answers here:

- 1.
- 2.
- 3.
- 4.
- 5.

## Checkpoint answers

1. What is the purpose of exporting objects?  
*To transfer them to other environments.*
2. Where is the export file stored if you select Wait?  
*You choose the location in the Save window.*
3. What is a key to the object migration export process?  
*The search for dependents.*
4. What happens if you select an object to be in the package more than once?  
*Nothing. It is put into the package only once.*
5. What can happen if you open the export file and change it?  
*The header can be corrupted and the file becomes useless.*

## Summary

---

Now that you have completed this unit, you should be able to export selected objects into an Object Migration package



## APPENDIX A Design

### Design elements

The use of standard naming conventions when you design or develop applications on the IBM TRIRIGA Application Platform helps keep your applications consistent. The IBM TRIRIGA Application Platform distinguishes between the labels that users interact with and the names with which developers interact. Users can rename states, sections, and fields that are seen in the user interface without damaging the business logic behind the object. This appendix outlines the different naming standards that are used in the IBM TRIRIGA Application Platform.

Most design elements in the IBM TRIRIGA Application Platform have a prefix for their name. The following guidelines for the prefixes are used for names:

- Design elements that are delivered as part of the IBM TRIRIGA Application Platform have the prefix **tri**, as in **triPeople**.
- New design elements are prefixed with **cst**, such as **cstCourseSection**.
- If a system design element is customized, the combination prefix of **cstTri** is used in some cases. An example of this prefix in use is **cstTriPeople**.

In this table, the Name column defines the standard for the system name of new design elements. The Label column is the standard for the name that the user sees.

Design element	Name/example	Label/example
Module	cst + mixed case, no spaces. Example: <b>cstContract</b>	None (modules do not have labels).
Business Object	prefix + mixed case, no spaces. Example: <b>cstPurchaseOrder</b>	User-readable name for the Business Object. Example: <b>Purchase Order</b>
Field	cst + mixed case, no spaces + field type suffix (see next section for suffixes). Example: <b>cstAssignedFullNameTX</b>	User-readable name for field. Example: <b>Assigned Full Name</b>

Design element	Name/example	Label/example
Smart / Query section	prefix + Business Object name + mixed case, no spaces. Example: <b>cstOrganizationBillTo</b>	User-readable name for section. Example: <b>Bill To</b>
	A section name can be only 30 characters; there might not always be enough space for the full business object name. Try to make the name reflect what the intent of the section is.	
State family	cst + context. Examples: <b>cstActionItem, cstData, cstDocument</b>	None
State family state	cst + (action) mixed case, no spaces. Example: <b>cstDeleted</b>	User-readable name for the action. Example: <b>Deleted</b>
	The null state is the exception to this rule. Its name cannot be changed.	
State family action	cst + (action) mixed case, no spaces. Example: <b>cstFinalDelete</b>	User-readable name for action (Transition). Example: <b>Final Delete</b>
Form	cst + Business Object name + mixed case, no spaces. Example: <b>cstPurchaseOrderShortForm</b>	User-readable name for the form. Example: <b>Short Form Purchase Order</b>
	Each business object can have many forms. This rule applies to only the first form, as you must use different names for each form.	
Form sections	cst+ mixed case, no spaces. Example: <b>cstGeneral</b>	User-readable name for section. Example: <b>General</b>
Portal sections	prefix + Type - Business Object name – Description. Examples: (1) cstGraph - triBuilding - Portfolio by Tenure (Pie Graph) (2) cstReports - Project Team Member (3) cstURL - My Calendar	User-readable name for Display on Portal. Examples: (1) Portfolio by Tenure (2) Project Team Member Reports (3) My Calendar
	Types: Graph: Query section pointed to a graph Query: Query section pointed to a query Reports: Query List URL: External	

Design element	Name/example	Label/example
Portals	<p>Customer name + role – <i>Type</i>  Sequence. Examples:  <b>(1) TRIRIGA Contact Center Agent – Standard</b>  <b>(2) TRIRIGA Contact Center Agent - Graphic 2</b></p>	None
	<p>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</p>	

Design element	Name/example	Label/example
Report, Query, Graph	<p>[Lowercase cst prefix +] Form Name (or Business Object Name or Module Name) - Keyword – Context.</p> <p>Example: <b>cstEmployee – Display - Active Status Associated to Current Record</b></p> <p>Use CUSTOM or a customer subscribed numbering system for customer reports and SYSTEM for TRIRIGA delivered reports.</p> <p>Context should include filters to states, data filters, association filters, if editable, for example. For queries tied to a \$\$RECORDID\$\$ or other Platform keys, ensure that you include the context. Also notice that the Query might be a custom query for an IBM TRIRIGA module, BO, or form. In this case, ensure that you use the cstTri prefix for the object.</p> <p>Keywords:</p> <ul style="list-style-type: none"> <li>Display: Query to display data in Form query sections</li> <li>Filter: Query to filter data for other queries</li> <li>Find: Query for Find action of a locator or section</li> <li>Formula: Query for extended formulas</li> <li>Graph: Graph reports that are intended for users</li> <li>Portal: Query for portal section</li> <li>Report: Report that is intended for users</li> <li>Summary: Summary Report that is intended for users</li> <li>Graphics: Graphics Editor Reports</li> <li>Reserve: Calendar/Reserve-based query of data that is associated with record</li> <li>Workflow: Query for workflow query task</li> <li>HGRID: Hierarchical Queries</li> </ul> <p>There are special naming conventions for Master Detail Queries. They are as follows:</p> <p>Primary Query: Form Name – Module Name – Master Detail Query</p> <p>Sub Queries (one for each State available for the Form):</p> <p>Form Name – State – Editable</p> <p>Hierarchy Views: Form Name – Module Name – Hierarchy View</p>	<p>Form label (or Business Object label or module label) – Context. What the report displays to user. Example: <b>Employees - All Associated Active Employees</b></p>
Security groups	Customer name + role. Example: <b>TRIRIGA Contact Center Agent</b>	None

# Field suffixes

When you design or develop your data dictionary, use a common naming convention for your fields. This convention eliminates the mismatching of names-to-types across multiple implementations. Each field type is detailed in this table:

Field type	Suffix	Example
Action Button	AB	cstLookupGeographyAB
Boolean	BL	cstProcessFlagBL
Binary	BI	cstOrganizationBI
Business Object	BO	cstPersonBO
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	cstCommittedCostsFR
Image	IM	cstPortraitIM
List	LI	cstYesNoLI
Note	NO	cstCommentsNO
Number	NU	cstRateNU
Password	PA	cstPasswordPA
System Read Only	SY	Do not create. Use Find to add to Business Object.
Text	TX	cstNameTX
Time	TI	cstStartTimeTI
UOM	UO	cstAreaUomUO
URL	UR	cstExternalInfoUR

## Publish name

Select the combination of fields that make the record unique (composite primary key). Try not to use the control number as part of the published name. Records that have the control number as one of its published name fields cannot be updated through Data Integrator or Bulk Loader. Use the control number in the publish name only for log or nonpermanent data; for example, action forms, or helpers.

You can use the following standards and tips:

- **Control Number:** Start With = 1000000.
- **Image:** Select the image field that represents the record (if applicable).
- **Cost:** Select the number fields that represent the cost of the record (if applicable).
- **Quantity:** Select the number fields that represent the quantity of the record (if applicable).
- **Conversion Group:** If the record is a document that has currency fields, include a cstConversionGroupLI field that defines which conversion group to use for converting to base fields. Use **Default** for the default value.
- **Exchange Date:** If the record is a document that has currency fields, include a cstExchangeDT field that defines which date to use for converting to base fields. Use **Current Date** for the default value.

## Associations

Associations define the relationships between records. The available associations for any record are defined in the Association Manager for the record's Business Object. Give each association a meaningful association string that describes the relationship.

Some examples of associations are as follows:

- Employee Reports To an employee with the reverse association of Managed By.
- Purchase Order Has Line Item has Purchase Order Line Items with the reverse association of Line Item For.

For each association defined, also define the equivalent reverse association.

- Employee Managed By an employee with the reverse association of Reports To.
- Purchase Order Line Items Line Item For has a Purchase Order with the reverse association of Has Line Item.

Use a different association, with a different association string, for each locator and smart section.

# Workflows

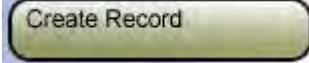
When you design or develop your workflows, use a naming convention for your tasks and workflows. This convention allows someone who is reviewing the workflow to know your intentions for each task without interpretation. The standard naming convention that is used by IBM TRIRIGA for each task type is listed in the following table. Give each task a description that further describes its intent when applicable.

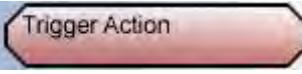
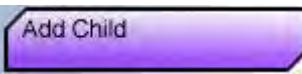
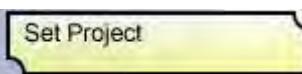
For a new workflow, prefix each task name with **cst**. For a copy of a system workflow, prefix each task with **cst** as you modify it. This prefix makes it obvious which tasks are changed, and which are the same as when they were shipped.

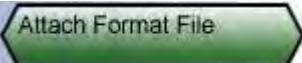
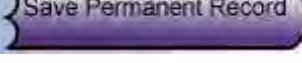
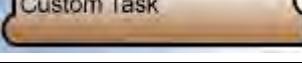
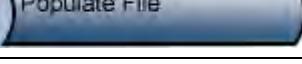
For the form or business object names, substitute the form or business object on which the task is performing work. If the task can work on any business object in a module, use the module name. You might need to further qualify the business object name to give it a more descriptive context.

For instance, if you have multiple employee records that are used in the workflow, you can 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 (when applicable).

All tasks in a workflow must have unique names, remembering to add context where required. Static wording is shown in the following table in bold.

Task	Naming format	Example
 (Asynchronous workflow)	Business Object name - action – description	cstEmployee – Associate – Primary Location
 (Synchronous workflow)	Business Object name - action - description	cstLocation – Synchronous - Populate parent fields on create
	<b>cst Create BusinessObject name from BO name (source)</b>	cst Create cstAddress from Building Primary cstAddress

Task	Naming format	Example
	<b>cst Update</b> Business Object name (target) <b>from</b> Business Object name (source) OR <b>cst Append</b> Business Object name (target): Section name <b>with</b> Business Object name (source)	cst Update cstProperty from cstLocation OR cst Append cstProperty: cstAddressOther with cstAddress
	<b>Note:</b> Use the <b>Append</b> option when mapping records from the other task as the source to append to a section of a record.	
	<b>cst Get</b> Business Object name (target) <b>from</b> Business Object name (source)	cst Get cstProfile from cstGroup associated to cstBuilding
	<b>cst Associate</b> Business Object name (target) <b>to</b> Business Object name (source), <b>OR</b> <b>cst De-Associate</b> associate Business Object name (target) <b>from</b> Business Object name (source)	cst Associate cstNotification to cstLocation OR cst De-Associate cstPeople from cstLocation
	<b>cst Action [space]</b> Business Object name	cst Remove Temp cstAddress; cst Issue cstPurchaseOrder
	<b>cst Delete Reference to</b> Business Object name (source) <b>from</b> Business Object name (target)	cst Delete Reference to cstProfile from cstGroup Associated to cstBuilding
	<b>cst Add Child</b> Business Object name (source) <b>to</b> Business Object name (target)	cst Add Child cstSpace to cstFloor
	<b>cst Set Project of</b> Business Object name (target) <b>from</b> Business Object name (source)	cst Set Project of cstInvoiceItem from cstContract

Task	Naming format	Example
	<b>cst Attach</b> file name <b>to</b> Business Object name	cst Attach Deleted Cost Code.rpt to cstNotification
	<b>cst Get</b> query	cst Get cstPeople Associated with cstBuildings
	<b>cst Assign</b> user or group name <b>to</b> action Business Object name	cst Assign Admin Group to Remove cstContactRoles Associated to Retired cstPeople
	<b>cst Assign</b> user or group name <b>to Approve</b> Business Object name	cst Assign At Bat cstPeople to Approve cstPurchaseOrder
	<b>cst Schedule</b> action <b>for</b> Business Object name	cst Schedule cstFinalIssue for cstBidResponse
	<b>cst Get Temp</b> Business Object name	cst Get Temp cstPeople
	<b>Note:</b> When performing other tasks to the results of a temp task, use the modifier Temp for the BO name. For example, if you are using a Modify task <i>Append Temp cstProperty: cstAddressOther with cstAddress</i> .	
	<b>cst Set Context</b> <b>for</b> Business Object name, form name (if to a particular form)	cst Set ID to Read Only for cstPeople
	<b>cst Commit Temp</b> Business Object	cst Commit Temp cstPeople
	<b>cst Iterate</b> for each Business Object	cst Iterate for each cstPeople
	<b>cst Call</b> context	cst Call Update cstTask triActualEndDT
	<b>cst Populate</b> context <b>with</b> Business Object	cst Populate bid.xls with cstBid
	<b>cst Distill</b> context <b>into</b> Business Object	cst Distill bid.xls into cstBid

Task	Naming format	Example
	<b>cst Call</b> Business Object - context	cst Call cstTask - Permanent Save Validation <b>Note:</b> This is the synchronous workflow name with the - <b>Synchronous</b> removed.

# More about Cloud & Smarter Infrastructure

You can find the latest information about IBM Cloud & Smarter Infrastructure education offerings online at the following location:

[www.ibm.com/software/tivoli/education/](http://www.ibm.com/software/tivoli/education/)

Also, if you have any questions about education offerings, send an email to the appropriate alias for your region:

- Americas: [tivamedu@us.ibm.com](mailto:tivamedu@us.ibm.com)
- Asia Pacific: [tivtrainingap@au1.ibm.com](mailto:tivtrainingap@au1.ibm.com)
- EMEA: [tived@uk.ibm.com](mailto:tived@uk.ibm.com)

## Cloud & Smarter Infrastructure user groups

You can get even more out of Cloud & Smarter Infrastructure software by participating in one of the 91 independently run Cloud & Smarter Infrastructure user groups around the world. Learn about online and in-person user group opportunities near you at [www.tivoli-ug.org](http://www.tivoli-ug.org).

## Certification

All IBM certifications are based on job roles. They focus on a job a person must do with a product, not just the product's features and functions. Online certification paths are available to guide you through the process for achieving certification in many IBM Cloud & Smarter Infrastructure areas. See [ibm.com/tivoli/education](http://ibm.com/tivoli/education) for more information about certification.

**IBM**<sup>®</sup>



[ibm.com/training](http://ibm.com/training)

Authorized  
**IBM** | Training