

IBM TRIRIGA Application Platform
Version 3 Release 5

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

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 61.

This edition applies to version 3, release 5, medication 0 of IBM TRIRIGA Application Platform and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2011, 2015.**

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

Contents

Chapter 1. Integrating data with the TRIRIGA DataConnect tool. 1

Overview of the TRIRIGA DataConnect process ..	1
Object migration packages.	1
Staging table settings	1
Business object publication	3
Business object deletion.	3
Job business objects	3
Mechanisms to move data into tables	4
Staging table fields	5
DC_JOB table fields	7
ETL maps	8
Workflows to move data into records	9
Workflow task settings	10
Workflow calls	13
Runtime data validations.	13
Iteration statuses	13
Temporary data	14
DataConnect agent	18
Workflow processing	18
Cleanup agent	19
Common issues with DataConnect	19
Tips for operating DataConnect.	22

Chapter 2. Integrating data with the TRIRIGA Data Integrator 25

Overview of the TRIRIGA Data Integrator process	25
Field types	25
Requirements.	31
Functional requirements	31
Usability requirements.	32
Tab-delimited files	33
Header files	34
Header files for non-hierarchical modules ..	34
Header files for hierarchical modules.	35
Creating header files	35

Preparing associative uploads	36
Association strings	36
Identifying associations	37
Verifying association rules	37
Adding associations to header files	37
Preparing workflow tasks for associative uploads	38
Uploading data	40
Uploading data manually.	40
Uploading data with batch uploads	41
Batch upload settings	41
Scheduling batch uploads	41
Batch upload termination.	43
Tips for operating Data Integrator	43

Chapter 3. Integrating data with the TRIRIGA Data Load Manager 47

Data load sets	47
Creating data load sets and data load items ..	48
Process data load sets in the Data Load Manager	49
Generating spreadsheets from the Data Load Manager	50
Creating a data load spreadsheet	51
Data load spreadsheet validation macros ..	51
Processing and loading sample data by using the Data Load Manager	56
Processing sample data	56
Configuring the sample data	56
Modifying the data source	57
Implementing data load changes	57
Loading sample data	57
Portfolio sample data	58

Notices 61

Trademarks	63
Terms and conditions for product documentation..	63
IBM Online Privacy Statement	64

Chapter 1. Integrating data with the TRIRIGA DataConnect tool

You can import data from an external system into your IBM® TRIRIGA® database with staging tables and workflow tasks with the DataConnect tool.

Overview of the TRIRIGA DataConnect process

IBM TRIRIGA DataConnect is a tool for importing external data into the IBM TRIRIGA database. It is useful for both initial loads of data and for batch insert and batch update of data on a recurring basis.

With DataConnect, you can write data from an external source directly into the IBM TRIRIGA staging tables. Then, you can use workflows to process the data for insertion into IBM TRIRIGA business objects. In addition to pushing the data, you also can store the data into the language tables.

Implementing the DataConnect process includes the following basic steps:

- Prepare the staging tables for the business objects that are populated or updated from the external source.
- Prepare the DataConnect job control business object. This business object controls the integration instance.
- Prepare the mechanisms to move data from the external source into the appropriate business object staging tables and the DataConnect job control table. You can use ETL maps or SQL.
- Prepare the workflows to move data from the business object staging tables into new or updated records. Initiated by the DataConnect agent, these workflows can also validate the input data and implement other business logic.

Object migration packages

Depending on your IBM TRIRIGA application version, certain DataConnect object migration packages must be applied to your upgrades. The packages are named `triDataConnectJob.zip` and `triDataConnectJobLog.zip` and are in the IBM TRIRIGA installation directory in the `\userfiles\upgrades` folder.

In the `triDataConnectJob` module, if the `triDataConnectJob` business object is not available on your installation, you must apply the `triDataConnectJob` object migration package. In the `triLog` module, if the `triDataConnectJobLog` business object is not available on your installation, you must apply the `triDataConnectJobLog` object migration package.

DataConnect works with a TRIRIGA 8.x application database. After you import the `triDataConnectJob` object migration package, the `triCreatedBy` field is empty. This condition is not a problem and must not be modified. The DataConnect Agent populates the field with the IBM TRIRIGA user from the DataConnect Job entry.

Staging table settings

For a business object to have a staging table that is associated with it, the **Has Staging Table** property of the business object must be on. In addition, fields within the business object must be identified as staging table fields with their **Staging**

Table Field property on. Publishing a business object establishes or updates the associated staging table. A business object can have only one staging table.

Business objects with staging tables

In the IBM TRIRIGA Administrator Console, select the **DataConnect** managed object then **Business Objects with Staging Tables** to see a list of the business objects with staging tables.

To see the staging table fields for a particular entry, select the line item and click **Expand Data**. The Fields column then shows the IBM TRIRIGA name for the field, a hyphen, and the database name for the field. If the field is a key field that is used for Upsert, the field name has an asterisk. DataConnect smart section keys display the section name followed by the field name. Field definitions are useful for identifying the corresponding staging table fields for inbound data. For more information, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

Has staging table

The **Has Staging Table** property identifies those business objects for which IBM TRIRIGA maintains staging tables. If the property is on and the business object is published, a staging table is either created or updated. If the property is off and the business object is published, the publish process deletes the staging table if it exists. By default the **Has Staging Table** property is off.

When the business object is brought in through object migration and object migration publishes the business object, a staging table is created for a business object that has the **Has Staging Table** selected.

Staging table field

The **Staging Table Field** property identifies those fields to be included in the business object staging table. If the property is on, the field is included in the staging table. Changes to this property after a staging table is created are not reflected in the staging table until the business object is republished. By default the **Staging Table Field** property is off unless the field is required.

The **Staging Table Field** property is supported for the following field types: Boolean, Business Object, Classification, Color, Date, Date and Time, Duration, List, Locator, Number, Password, Text, Time, UOM, and Url. Only fields that are in the General section are supported to be staging table fields. Fields in a smart section can be added to the staging table. All required fields have the **Staging Table Field** property on.

If you set the **Staging Table Field** property for a number field that has a related UOM field in the business object, be sure to set the **Staging Table Field** property for the related UOM field. You must also set the **Staging Table Field** property for the related UOM, if the UOM that is being imported is different from the base UOM. For example, if the base UOM for area is square feet and the UOM being imported is acres, set the **Staging Table Field** for the UOM that is imported.

You can also include formatted values for Date, Date and Time, Duration, and Time fields in staging tables. If the property for these DataConnect format fields is turned on, the field is included in the staging table. The name of the field is appended with **_F** in the staging table. For example, if the **Staging Table Field**

property for the date field that is named TRIDATEDA is turned on, the field TRIDATEDA_F is included in the staging table. The format of these fields is determined by the user preferences of the user. Usage of these format fields is optional. If you want to include these format fields in the staging table, you must publish the business object to apply the change.

Staging table key

The **Staging Table Key** property identifies fields to be used as keys to find a record through the Upsert or Update action. The Insert action does not use this property. By default the **Staging Table Key** property is off. It can be selected only when the **Staging Table Field** property is selected.

The **Staging Table Key** property is supported for the following field types: Boolean, Business Object, Classification, Color, Date, Date and Time, Duration, List, Locator, Number, Password, Text, Time, UOM, and Url.

Business object publication

When you select the **Publish BO** action for a business object with the **Has Staging Table** property on, certain checks are triggered. If there are any jobs in Processing or Ready state that use the staging table and business object, the publication fails and a message is posted to the user immediately.

The business object publication is handled by an agent and is not an immediate action. So, the Publish agent checks the jobs again in case a new job was started or the state of an existing job was changed in the interim. If there are any jobs in Processing or Ready state that use the staging table and business object, the publication of the business object fails. Then, IBM TRIRIGA posts a notification to the user, and writes details about why the publishing failed to the server.log.

Business object deletion

When you select the **Delete BO** action for a business object with the **Has Staging Table** property on, certain checks are triggered. If there is a job in the Processing state that uses the staging table and business object, the deletion fails and a message is posted to the user immediately. If there is a job in the Waiting state, the business object and staging table are deleted and the state of the job is changed to Obsolete.

Job business objects

The DataConnect Job Control (DC_JOB) table manages the jobs that move data from staging tables into the IBM TRIRIGA database. A business object must exist in the triDataConnectJob module for each integration process. Creating the business object in the triDataConnectJob module is a one-time task for the integration process.

You need a DataConnect Job Control business object for each integration process. For example, if you are populating purchase orders from an external source, you might have a cstPurchaseOrderJob business object as your DataConnect Job Control business object. Then, the process might pull in data for the cstPurchaseOrderBody and cstPurchaseOrderLine business objects (both with the **Has Staging Table** property on).

In general, you create the business object with the following steps:

- Open the triDataConnectJob module through the Data Modeler.
- Create the business object in the triDataConnectJob module.

After you create the new business object, you can create a form and menu navigation for the business object. DataConnect is delivered with a base form and a base menu for the base triDataConnectJob business object.

The state transitions for your new business object are inherited from the base business object and must not be modified. The triDataConnectJob base business object provides the following state transitions:

- Null -> triDCStartProcessing -> In-Process
- In-Process -> triDCCompleted -> Completed
- In-Process -> triDCFailed -> Failed

All fields in the first section are read-only. The second section contains log records for that DataConnect job. The log records are triDataConnectLog records in triLog. Click a line item to see the details for that event. By default the second section shows issues that are found during the DataConnect run. However, when you select **DataConnect Runtime** in the **Platform Logging** managed object in the IBM TRIRIGA Administrator Console, the second section also shows positive information about records that were inserted or updated.

Logging levels for the various agents is determined by the platform log settings and the log4j settings on the server where the agent is running. DataConnect uses two agents. One is the DataConnect Agent, which creates the job object and provides it to the Workflow Agent asynchronously through an event. Its job is over when the job object is created and the event is posted. The second agent, the Workflow Agent, runs the business logic (the workflows) and then the DataConnect tasks that create and update objects in the IBM TRIRIGA database. If you want to see reporting on your job objects even when there are no issues, you must set the DataConnect Runtime logging on the server of the Workflow Agent. Setting it on the server of the DataConnect Agent has no effect since the Workflow Agent is running the workflow and creating or updating the objects from the staging tables.

Mechanisms to move data into tables

You must provide the mechanism, either through an ETL tool or direct SQL statements, for moving data from the external source to the appropriate tables in the IBM TRIRIGA database. These tables include the business object staging tables and a DataConnect Job Control table.

The mechanism must create a unique job number to reference within both the DataConnect Job Control table and the business object staging tables. Furthermore, the mechanism can include one or more correlation and sequence numbers.

Although no meaning is associated with the job number, it must be a unique identifier. One approach to creating the job number is to use a generator of globally unique identifiers (GUID) or universally unique identifiers (UUID). Then, the approach can attempt to insert the DataConnect Job Control table entry with a state of New. If the attempt succeeds, use that as the job number for the entries that are written to the staging tables. If it fails with a duplicate key error, generate another GUID/UUID/FUID and try again.

You can use the Spoon tool from Adobe Kettle. This method is similar to how the IBM TRIRIGA Workplace Performance Management products and IBM TRIRIGA Real Estate Environmental Sustainability products use Spoon to create ETL mappings for Fact Tables that pull data from IBM TRIRIGA tables.

Staging table fields

In addition to the fields that you identify with the **Staging Table Field** property, each staging table includes specific fields.

Table 1. Staging table fields

Column name	Key	Type	Default	Null-able	Description
DC_JOB_NUMBER	Y	NUMBER (20)		N	A job identification number.
DC_CID	Y	NUMBER (20)		N	A correlation identification number that is used for parent-child relationships from staging tables.
DC_SEQUENCE_ID	Y	NUMBER (20)		N	A sequence number for a group of entries.
DC_STATE	N	NUMBER (4)	0	N	A state identifier to indicate the current state of this entry during the processing of a DataConnect Job. Values are: 0 - New, 1 - Ready, 2 - Processing, 3 - Completed, 4 - Failed, 5 - Obsolete.
DC_ACTION	N	NUMBER (4)	1	N	The action that is taken. Values are: 1 - Insert, 2 - Update, 4 - Upsert.
DC_PATH	N	VARCHAR ()		Y	The path (including Record Name) for the object that is being acted upon. For an Insert, this field can indicate to the DataConnect task where in the hierarchy to place the record. Children must have the parent path in DC_PATH. If an object is at the top of the hierarchy or if the object is not within a hierarchy, DC_PATH must be empty. For an Update, if no keys exist, this field must be set to the record you want to update, the IBS_SPEC.OBJECT_PATH of the record to update.
DC_GUI_NAME	N	VARCHAR (100)		Y	The form name that you want the new or updated row to use. On insert, if this form is not set, the system uses the default form for the business object if there is one. Otherwise, the object has no associated form. On update, if this form is not set, the system does not modify the current form setting.

Table 1. Staging table fields (continued)

Column name	Key	Type	Default	Null-able	Description
DC_PROJECT	N	VARCHAR (100)		Y	<p>On Insert or Update, sets the project for the record if it is not empty. The value is what appears in the portal menu bar when a project is selected. DC_PROJECT must use the format ProjectID-ProjectName.</p> <p>Only the record that is involved in the Insert or Update has its project changed. No changes occur to children of that record. If you are inserting or updating data with a parent-child relationship, be sure to set the appropriate project for all data you want changed.</p>

Table 1. Staging table fields (continued)

Column name	Key	Type	Default	Null-able	Description
Column names are determined by the IBM TRIRIGA Application Platform. Use the Administrator Console to find the staging table name and the database name.	N	Depends on the IBM TRIRIGA Application Platform column type.		Y	<p>Fields in the business object with Staging Table Field checked (on). An extra column is added in the staging table for Date, Date and Time, Duration, and Time fields. The name of these fields is appended with _F. Based on the Staging Table Key property, the fields might be keys. Used by Upsert and Update.</p> <p>For an Upsert, the DataConnect task generates SQL by applying the designated key values in the staging table to see whether the record exists. If the record exists (that is, one record is found), the record is updated with the data in the staging table. If more than one record exists, the staging table row is marked Failed. If the record does not exist (that is, no records are found), a new record is inserted by applying the staging table data.</p> <p>For an Update, if keys are defined on the staging table, the DataConnect task generates SQL by applying the key values in the staging table to see whether the record exists. If the record exists (that is, one record is found), the record is updated with the data in the staging table. If more than one record exists, the staging table row is marked Failed. If the record does not exist (that is, no records are found), the staging table row is marked Failed.</p>

DC_JOB table fields

The DC_JOB table is filled by the external process, updated by the DataConnect Agent and the asynchronous workflows that act on it, and deleted by the Cleanup Agent. The DC_JOB table includes specific fields.

Table 2. DC_JOB table fields

Column name	Key	Type	Default	Null-able	Description
JOB_NUMBER	Y	NUMBER (20)		N	A job identification number.
JOB_TYPE	Y	NUMBER (4)	1	N	An identifier of the type of job. Value is: 1 - Create/Update IBM TRIRIGA records, inbound data.

Table 2. DC_JOB table fields (continued)

Column name	Key	Type	Default	Null-able	Description
JOB_NAME	N	VARCHAR (100)		Y	The name of the job.
JOB_RUN_CTL	N	VARCHAR (100)		N	Controls concurrency. The DataConnect Agent uses the JOB_RUN_CTL with the JOB_NAME to throttle itself. If a Job is single threaded, no other job with that JOB_NAME runs until the running job is Completed or Failed. Values are: 1 - single threaded, 2 - multithreaded.
BO_NAME	N	VARCHAR (100)		Y	The business object name for the record that is created to represent the job. This business object must be created in the triDataConnectJob module.
USER_ID	N	NUMBER (20)		Y	User ID to use for processing. Must be a valid IBM TRIRIGA user ID (T_MYPROFILE ID).
STATE	N	NUMBER (4)	0	N	A state identifier to indicate the current state of the job. Values are: 0 - New, 1 - Ready, 2 - Processing, 3 - Completed, 4 - Failed, 5 - Obsolete.
SOURCE_SYS_ID	N	VARCHAR (100)		Y	Identifier for the system that created the job (used for informational reporting).
PROCESS_SYS_ID	N	VARCHAR (100)		Y	Identifier for the system that is processing/processed the job (used for informational reporting). You do not need to provide this information.
CREATED_DATE	N	DATETIME	Date	N	The datetime the entry was added (defaulted).
UPDATED_DATE	N	DATETIME		Y	The datetime the entry was last changed. You do not need to provide this information.

Use the Administrator Console to see the contents of the DC_JOB table. Select DataConnect Jobs in the DataConnect managed object in the Administrator Console. For more information about the Administrator Console and the DataConnect managed object, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*.

ETL maps

Your implementation of a DataConnect integration determines how you move your data into the staging tables, either through an ETL tool or direct SQL statements. Regardless of how you move your data from the external source into the staging tables, the data must be populated in the external source before you run the mapping.

If you use an ETL tool, the integration drives which ETL maps you must develop to move data from the external source to the staging tables. Use a tool to define an ETL mapping between the external source and the staging tables. One such tool is the Spoon tool from Pentaho Kettle. Otherwise, you can use another tool of your choice.

When the mapping runs, it must accomplish the following tasks:

- Create a unique job number.
- Create an entry in the DataConnect Job Control table (DC_JOB), setting the state to New.
- Populate the staging tables for the business objects, including any child staging tables. The entries include the job number, one or more correlation numbers, and the action. The path is required on Update actions where no keys are defined and optional for Insert actions.
- Optional: Update the entry in the DataConnect Job Control table (DC_JOB), setting the state to Ready. This update can be set manually through the Administrator Console by selecting the DataConnect managed object, the DataConnect job, and then the **Ready Job** action.

Workflows to move data into records

You use the workflow capabilities from IBM TRIRIGA to move data from the staging tables to the actual business object records. The exact workflow logic depends on your company business rules and the business object that is being filled. Workflows for controlling DataConnect must be created in the triDataConnectJob module.

Use the DataConnect workflow task to retrieve staging records and iterate through the staged entries, creating or updating a smart record for each one. DataConnect workflow tasks can be nested to support parent-child relationships and business logic.

The DataConnect task acts in two parts:

- Retrieves the set of records to work on from the appropriate staging table as defined by the DataConnect task property settings.
- Acts as an iterator for the following tasks:
 - Creates or updates a record for each row of data from the staging table.
 - Runs the body of the DataConnect task for each row. Other task steps can be positioned within the body of the DataConnect task.

While inside a DataConnect workflow task, the Review Status of the task is INSERT, UPDATE, or FAILED, based on the current action of the task iteration. Upon exiting a DataConnect workflow task, the Status is Completed or Failed (if any of the staging table rows failed).

Example of a workflow with a DataConnect task

The following example uses a purchase order (PO) and its PO line items for purposes of illustration. First, is the complete workflow.

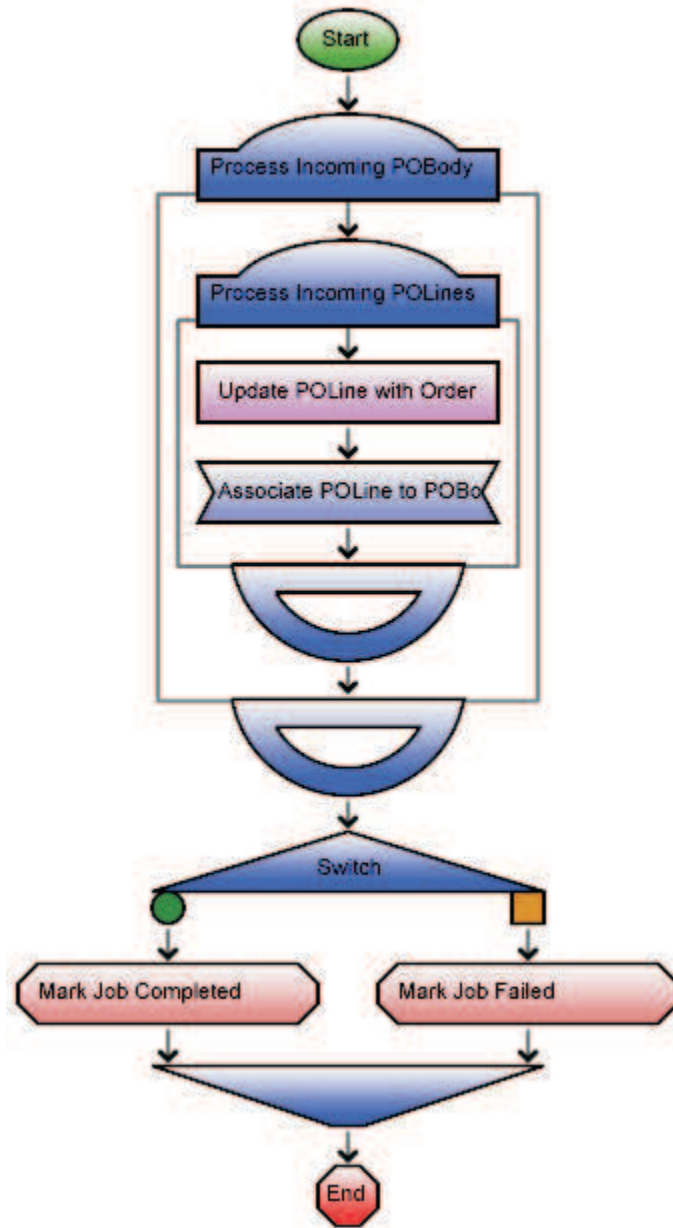


Figure 1. Workflow that includes DataConnect task

Workflow task settings

In the Workflow Builder, create a workflow in the triDataConnectJob module. The sample workflow deals with several key workflow tasks and task settings.

Start task

The Object Type that is related to triDataConnectJob, and the Event of SYSTEM DC PROCESS JOB identify the correct workflow for the DataConnect Agent. The DataConnect Agent pulls the BO_NAME field, which must be the name of the Object Type from the DC_JOB entry for the job.

Table 3. Start task settings

Field	Description
Name	In the workflow Start task, Name the workflow by applying the IBM TRIRIGA naming standards.
Concurrency, Integration	Set Concurrency to Asynchronous. Select the Integration flag, which allows DataConnect tasks to be a part of the workflow.
Module	Set Module to triDataConnectJob.
Object Type	Set Object Type to the business object you created in the triDataConnectJob module.
Event	Set Event to SYSTEM DC PROCESS JOB. This event is the DataConnect event.
Propagate Integration Status, Integration	Select the Propagate Integration Status flag, which specifies to propagate the Integration flag when this workflow calls another workflow.

DataConnect task

The DataConnect task retrieves records from the staging table and iterates through the staged entries, creating or updating a smart record for each one.

Table 4. DataConnect task settings

Field	Description
Label, Description	Create the workflow body with the DataConnect task where appropriate. Enter a Label and Description .
Module, Business Object	Specify the Module and Business Object on which the DataConnect task operates.
Initial State	Make Initial State the state in which a new record must be created. The DataConnect task ignores the value of Initial State if the record is being updated. The choices in the list box are the states that are defined for the business object. The record is created in that state; it is not transitioned to that state. If you want the record to be transitioned, have the DataConnect task create the object in the null state. Then, use a Trigger Action task within the DataConnect task body to transition the record to the intended state.
Use Temporary Data	Select Use Temporary Data when the DataConnect task is to use temporary data. When a DataConnect task with Use Temporary Data checked applies incoming data, the platform inserts or applies updates to temporary data instead of to the permanent data. See Use Temporary Data for examples and more details.

Table 4. DataConnect task settings (continued)

Field	Description
In-Sequence, Task Step	<p>In the Correlation section:</p> <ul style="list-style-type: none"> • In-Sequence indicates that the correlation on the staging entry and the sequence number are used to determine the order by which the records are processed. This scenario is the default and the most common scenario unless you have DataConnect tasks one within another. • Task Step indicates that the correlation must come from an enclosing DataConnect task and must be used in the Where clause to determine what entries to process. For example, Job Number + Correlation Number, ordered by Sequence Number. Using Task Step allows the DataConnect task to get the correlation number from one of the enclosing DataConnect tasks.
None, Per X Iterations, All Iterations	<p>Select the scope in the Transaction section.</p> <ul style="list-style-type: none"> • None: The record is committed right after it is created or updated. • Per X Iterations: A new context is created for every X iterations and committed when that number of iterations is complete. • All Iterations: A new context is created before the task starts any processing and is committed when all iterations are complete.

Trigger action task

The last step in the workflow must be a Trigger Action task to mark the job Completed or Failed.

Table 5. Trigger Action task settings

Field	Description
Label, Action	In Trigger Action Task Properties, enter Mark Job Completed in the Label field, and select triDCCCompleted for the Action field.
Take the, of Task, Use it, Object Type	In the Records section, select Business Object for the Take the field and Start (loDCPurchaseOrder) for the of Task field. Mark the Use it option and select loDCPurchaseOrder for the Object Type field.

Break task

For transactions to roll back or commit, put a Break (or Continue) in the DataConnect body. With the Break (or Continue) task, you can check the DataConnect task status, and determine what block to go to. You can also decide whether to keep iterating or not (Continue versus Break) and whether to commit or roll back (Success or Failure). The transaction setting on the DataConnect task works with the Break (or Continue) task settings.

The Break also has settings to control temporary data. Find more details and examples in Use Temporary Data.

If the DataConnect task is not using transactions, rows are committed with each task step. If you do not reach a Break (or Continue) that changes the processing flow and are using transactions, a commit takes place that is based on the DataConnect task settings.

If the DataConnect task is using transactions and the workflow Breaks (or Continues), the transactions roll back or commit based on the Success or Failure setting on the Break. Processing continues at the scope indicated in the Break (or Continue). Current Scope means the DataConnect task that you are in. Break Flow Type means continue processing at the task after the DataConnect task. Continue Flow Type means continue onto the next row in the staging table batch for this job. Selected Block means Break (or Continue) at that DataConnect task.

Workflow calls

If the workflow with the DataConnect task calls other workflows, those workflows might need to be updated to check the **Integration** setting and to change the **Propagate Integration Status** flag as appropriate.

You can analyze existing workflows that might be called by the workflow for data movement. If there are steps within the workflow that must not be triggered if called from an Integration workflow, check the **Integration** setting with a Switch task or in the Start task. If the workflow must not propagate the **Integration** status, clear the **Propagate Integration Status** flag in the Start task.

The value of 1 for the **Integration** flag means that the flag is on.

Runtime data validations

The DataConnect run time conducts some data validation. For example, if there is a unit of measure (UOM) field in the staging table and its value is invalid. When the DataConnect task encounters a failed row, it stops processing and fails the DataConnect iteration (not the entire DataConnect task). The row is marked in the staging table as Failed.

Here, you have a choice, whether to continue with the next iteration or to break out of the task. Place a Break into the workflow to tell the workflow how to proceed. If no Break is added, the DataConnect task continues processing.

DataConnect includes the following built-in runtime data validations:

- UOM and locator values are valid.
- Data type is valid for the field.
- Form Name is valid for the object.
- Project name, if provided, is valid for the system.

You can add more data validation and cleansing with the **Use Temporary Data** property.

Iteration statuses

As with most task iterators, the DataConnect task remembers information from its last iteration, including status. For example, you might build a workflow that has two DataConnect tasks, one within the other. If a row fails within an inner iteration and you want to set the proper status at the end, save that row in a variable. This practice ensures that the status is not lost.

Temporary data

When you select the **Use Temporary Data** property in a DataConnect task, you allow the workflow tasks within the DataConnect task to process the incoming records from the staging table in session. In other words, the data can be manipulated before it is saved to the database.

Use a Get Temp Record task to retrieve a temporary record and use a Save Permanent Record task to save the current session changes. The Break task can remove a specific iteration from the session, or clear the session completely. The Break task can also fail a specific staging table row, or fail all staging table rows since the last Save Permanent Record.

When **Use Temporary Data** is checked, then as the workflow is processed, the record for the DataConnect task is the permanent record without changes from the staging table. Also, the record for the Get Temp Record task is the updated or incoming values from the staging table. This process enables data validation and cleansing of incoming values within the workflow and even data mapping on update if fields are blank in the staging tables.

Example of a single DataConnect task with temporary data

The following excerpt from a workflow shows a single DataConnect task that processes temporary data.

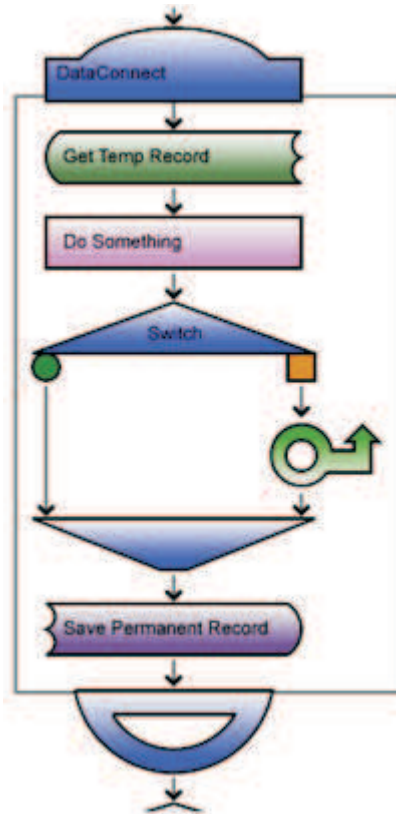


Figure 2. Single DataConnect task that processes temporary data

Table 6. DataConnect task settings

Field	Description
Use Temporary Data	Selected

Table 7. Break task settings

Field	Description
Flow Type	Continue
Task Status	Success
Break Scope	Current Scope
Discard Temporary Data	All
Fail Staging Rows	Selected, which sets All

The DataConnect task iterates through the staging table records, placing the inserted or updated records into the session. For each iteration, the workflow gets a temporary record, does something with it, and decides whether to keep it.

To keep the record, save the record permanently with a Save Permanent Record task. The Switch task fail condition uses the settings in the Break/Continue task to discard all session data. The fail condition also fails the staging rows that are used to generate the record in session, and continues to the next record in the iteration.

In this example of a single DataConnect task, setting **Discard Temporary Data** in the Break task to **Current** instead of **All** would have the same outcome.

Example of multiple DataConnect tasks with temporary data

The following excerpt from a workflow shows a DataConnect task that contains a second, child DataConnect task. This excerpt is an extension of the purchase order example that was shown earlier.

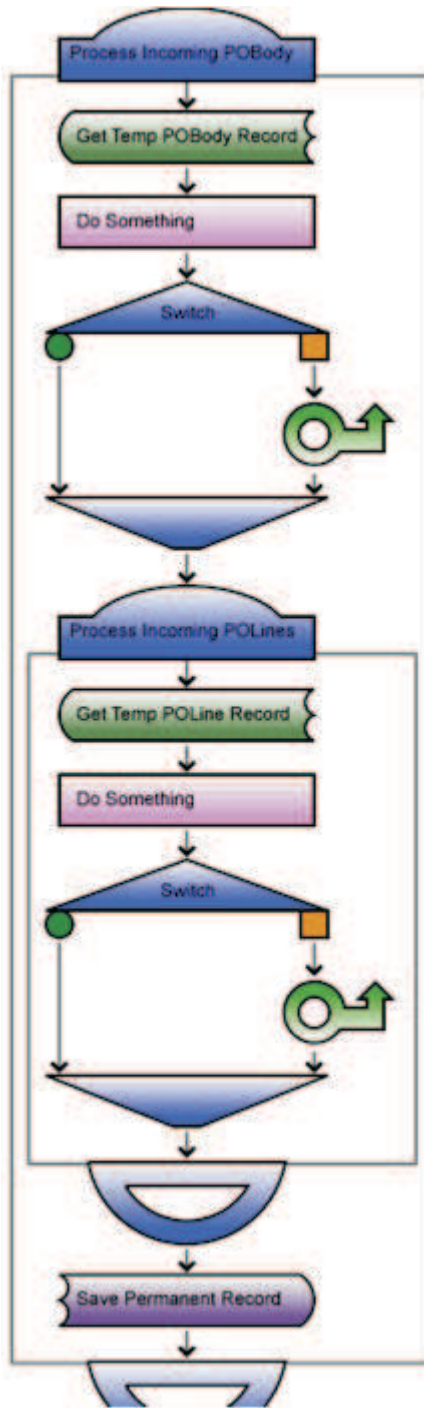


Figure 3. Multiple DataConnect tasks that process temporary data

Table 8. Outer DataConnect task settings

Field	Description
Use Temporary Data	Selected

Table 9. Outer Break task settings

Field	Description
Flow Type	Continue

Table 9. Outer Break task settings (continued)

Field	Description
Task Status	Success
Break Scope	Current Scope
Discard Temporary Data	All
Fail Staging Rows	Selected, which sets All

The outer DataConnect task iterates through the POBody staging table. For each iteration, the workflow gets a temporary record, does something with it, and decides whether to keep it. If there is an issue with the POBody record, the workflow runs the Break/Continue and discards all session data, fails the row, and moves on to the next POBody record.

If the POBody passes the Switch task, processing continues with the inner DataConnect task to handle the POLines for this POBody. The workflow might do one of several things based on what it finds and on the setting of the Break. You might have several Switch/Breaks and depending on your processing, the workflow might fail everything or just the one row.

The following scenarios show different uses of the Break and what would happen with the processing and the session:

- Fail the current POLine, remove the record that it created or modified from the session, and fail that staging table row only. Processing continues on the next POLine for the POBody. The inner DataConnect task's Break task uses the following properties:

Table 10. Inner Break task settings

Field	Description
Flow Type	Continue
Task Status	Success
Break Scope	Current Scope
Discard Temporary Data	Current
Fail Staging Rows	Selected, which sets Current

- Clear the current POBody iteration, remove the record that it created or modified from the session, fail the POBody staging table row, clear the POLines, and fail the POLines staging table rows. Processing continues on the next POBody. The inner DataConnect task's Break task uses the following properties:

Table 11. Inner Break task settings

Field	Description
Flow Type	Continue
Task Status	Success
Break Scope	Selected Block, Process Incoming POBody
Discard Temporary Data	All
Fail Staging Rows	Selected, which sets All

- Nothing (no Break task in inner DataConnect).

Having the Save Permanent Record task after the end of the inner (POLine) DataConnect task ensures that the platform saves the POBody and the POLines that go with the POBody.

DataConnect agent

The DataConnect Agent checks entries in the DataConnect Job Control table and runs those entries in Ready state. In the TRIRIGAWEB.properties file, the **DATACONNECT_SLEEP_TIME** property for the application server that is running the DataConnect Agent controls how often the agent checks the DataConnect Job Control table.

When it finds a job in the DataConnect Job Control table (DC_JOB) in the Ready state, the DataConnect Agent takes the following steps:

- Creates a record for the record indicated in the DataConnect job entry. This record is an instance of the Job Control business object that is created in the triDataConnectJob module.
- Populates the record with information from the job entry, including job number, job type, source system ID, process system ID, create date, updated date, user, and state.
- Sets the state of the DataConnect Job Control table entry to Processing.
- Posts the workflow event SYSTEM DC PROCESS JOB along with the record, which triggers the workflow.
- If there are issues with the DataConnect Job Control table entry or with creating the Job Control business object, the state for the DC_JOB table entry is set to Failed. The reason for failure is logged in the server.log. When the issue is fixed, you can try to run the job again from within the administrator console.
-

For more information about the agent manager, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*. For more information about the TRIRIGAWEB.properties file, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Workflow processing

When a workflow with a SYSTEM DC PROCESS JOB event and Job Control business object is triggered, the DataConnect task in the workflow runs certain steps. This process assumes that the workflow was defined with the Integration property on (true).

The DataConnect task runs the following steps:

- Use the configured information to determine what type of record to process, the state to leave the new record, the correlation type, and the transaction setting. Use the record on the Start task as the source for the job number to use.
- In the DataConnect tasks, get the staging table name for the record it needs to process and query for rows with the corresponding job number. Depending on the configuration setting for the correlation number, use the correlation number to order by, or in a Where clause to find entries with the correlation number. Get the staging table entries based on job number, correlation number, sequence number, action, and path.
- For each row fetched, create or update a smart record. Put the record in the state that is dictated in the DataConnect task definition (it is not transitioned to that state). Change the state of the row in the staging table. Set the project and form. Depending on the Use Temporary Data flag, the changes might be permanent or

- made in session. Run steps that are defined within the iterator, creating parent-child relationships and following business logic processing.
- Run the pre-create workflow on the new records. If you do not want the pre-create workflow to run, update the pre-create workflow to check the Integration flag, so when called by an Integration workflow it is skipped.
- If there is a call to another workflow, check the setting of the Integration status to change processing based on whether the workflow was called from an Integration process.
- If there is an event, check the FROM_INTEGRATION column to determine whether the Integration status is set in the workflow context.
- Process the transaction based on the DataConnect task configuration setting.

Cleanup agent

The Cleanup Agent processes the automatic cleanup of DataConnect jobs. Use the Agent Manager in the Administrator Console to define the name of the processor that is running the Cleanup Agent. In the TRIRIGAWEB.properties file, set the **DC_HISTORY_RETENTION_DAYS** property.

When a DataConnect entry in the DataConnect Job Control table is in the Obsolete or Completed state for the number of days in the **DC_HISTORY_RETENTION_DAYS** property, the Cleanup Agent is triggered. Then, the Cleanup Agent takes the following steps:

- Finds the job control number of the entry and deletes entries in the staging tables for that job control number.
- Deletes the DataConnect Job Control table (DC_JOB) entry.

The Cleanup Agent does not delete the Job Control business object entries (the business object in triDataConnectJob). Follow your company's business rules for managing triDataConnectJob entries that are no longer needed. You can delete them from the system directly or with a menu that you create.

For more information about the Agent Manager, see the *IBM TRIRIGA Application Platform 3 Administrator Console User Guide*. For more information about the TRIRIGAWEB.properties file, see the *IBM TRIRIGA Application Platform 3 Installation and Implementation Guide*.

Common issues with DataConnect

Common IBM TRIRIGA DataConnect issues include missing staging tables or staging table fields, invalid table columns, and changed business objects that might need to be republished.

Table 12. Common issues with DataConnect

Error message	Where found	How to fix
A business object: XX, does not have staging table. [XX = business object name]	From the DataConnect task run time when the workflow is running, a WFRuntimeException in the server.log	The staging table is missing for the XX business object. Task fails in the workflow.

Table 12. Common issues with DataConnect (continued)

Error message	Where found	How to fix
An error occurs while processing the staging records: XX. [XX = staging table record fields]	From the DataConnect task run time when the workflow is running	Something wrong occurred during the DataConnect task execution. One possible issue is on update the DC_PATH did not accurately define a smart object to update.
Cannot delete the BO, there are Pending DataConnect jobs.	User Interface	The Processing DataConnect job needs to complete before the business object can be deleted. Delete when the processing job is completed.
Cannot publish the Business Object at this time. There are DataConnect jobs either WAITING or PROCESSING. These must complete first before the publish can complete.	User Interface	A Processing or Waiting DataConnect job needs to be cleared out before the Publish of the object is successful. Both must clear out without intervention. Republish when there are no pending jobs.
DataConnect Agent - The triDataConnectJob Business Object cannot be found. Make sure to import before using the DataConnect Agent.	From the DataConnect Agent, a PlatformRuntime Exception in the server.log	Apply the most recent triDataConnectJob object migration package before you run or use DataConnect.
DataConnect Agent - The triDataConnectJob Business Object is not the correct version. Make sure to import before using the DataConnect Agent.	From the DataConnect Agent, a PlatformRuntime Exception in the server.log	Apply the most recent triDataConnectJob object migration package before you run or use DataConnect.
DataConnectTaskHandler - The Staging table: XX does not contain the staging field: YY. [XX = staging table name] [YY = field name]	From the DataConnect task runtime when the workflow is executing, a warning entry in the server.log	The staging table is missing fields that are expected. This result might occur if the staging table is out of sync with the business object. Republish the business object when the DataConnect Job is completed.
DataConnectTaskHandler: Cannot find a parent smart object whose object path is: XX. [XX = DC_PATH from the staging table]	From the DataConnect task runtime when the workflow is executing, a warning entry in the server.log	The run time cannot find the parent that is identified in the staging table field DC_PATH. The smart object is created; just the parent is not set for the smart object.
DataConnectTaskHandler: performing transition action with an invalid state, XX on the SmartObject: YY. [XX = state] [YY = smart object ID]	From the DataConnect task runtime when the workflow is executing, a WFRuntimeException in the server.log	Check the job's business object state transitions to make sure that it defined the state transitions in the base triDataConnectJob.

Table 12. Common issues with DataConnect (continued)

Error message	Where found	How to fix
<p>The BO for the job YY, XX is not a valid BO.</p> <p>From the DC_JOB table</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The BO_NAME column in the DC_JOB table for job YY is not a valid business object. Check to make sure that the name is correct and is in the triDataConnectJob module. Fix the entry and reset the STATE to READY (or 1).</p>
<p>The JOB_RUN_CTL XX is not valid for job YY.</p> <p>From the DC_JOB table</p> <p>[XX = JOB_RUN_CTL]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The JOB_RUN_CTL column for job YY is invalid in the DC_JOB table. Only 0 (single threaded) or 1 (multi-threaded) are valid entries. Fix the entry and reset the STATE to READY (or 1).</p>
<p>The JOB_TYPE XX is not valid for job YY.</p> <p>From the DC_JOB table</p> <p>[XX = JOB_TYPE]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The JOB_TYPE column for job YY is invalid in the DC_JOB table. Only 1 (in-bound) is valid. Fix the entry and reset the STATE to READY (or 1).</p>
<p>Unable to associate user ID BB for the job YY using the BO XX.</p> <p>From the DC_JOB table</p> <p>[BB = USER_ID]</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The agent was unable to use the USER_ID column to associate the newly created DataConnect Job to the user. Make sure the USER_ID is a valid T_MYPROFILE spec_id. Fix the entry and reset the STATE to READY (or 1).</p>
<p>Unable to create the BO for the job YY using the BO XX.</p> <p>From the DC_JOB table</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The agent was unable to create the DataConnect Job smart object. Check to make sure that the name is correct and is in the triDataConnectJob module. Fix the entry and reset the STATE to READY (or 1).</p>
<p>Unable to map fields necessary for DataConnect to work properly for the job YY using the BO XX.</p> <p>From the DC_JOB table</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The agent was unable to map the necessary fields to the newly created DataConnect Job. Check to make sure that the XX business object defined all of the fields in the base triDataConnectJob BO. Fix the entry or business object, possibly republish if the business object changed, and reset the STATE to READY (or 1).</p>

Table 12. Common issues with DataConnect (continued)

Error message	Where found	How to fix
<p>Unable to post workflow event SYSTEM DC PROCESS JOB for the job YY using the BO XX.</p> <p>From the DC_JOB table</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>The agent was unable to post a workflow event for the newly created DataConnect Job. The reason might be because there is no published workflow for the SYSTEM DC PROCESS JOB and the XX business object. Check to make sure that the workflow is published for the event and the XX business object. Fix the entry or workflow and reset the STATE to READY (or 1).</p>
<p>Unable to Publish, there are existing DataConnect jobs processing or waiting for the Business Object.</p>	<p>From the Publish Agent, a PlatformRuntime Exception in the server.log</p>	<p>The Processing or Waiting DataConnect job needs to be cleared out before the Publish of the object is successful. Both must clear out without intervention. Republish when there are no pending jobs.</p>
<p>Unable to transition job to triDCstartProcessing for the job YY using the BO XX.</p> <p>From the DC_JOB table</p> <p>[XX = BO_NAME]</p> <p>[YY = JOB_NUMBER]</p>	<p>From the DataConnect Agent, an info entry in server.log</p>	<p>Check the state transitions for business object XX to make sure that XX defined the state transitions in the base triDataConnectJob in Module triDataConnectJob. Fix the entry or business object, possibly republish if the business object changed, and reset the STATE to READY (or 1).</p>

Tips for operating DataConnect

You can use the following tips to enhance how your company operates the IBM TRIRIGA DataConnect tool.

Table 13. Tips for operating DataConnect

Tip	Description
Approval task	An Approval task cannot be run from within a DataConnect task.
DataConnect job remains in Processing state; workflow probably does not contain a Trigger Action to move job to Failed or Completed.	If a DataConnect job remains in Processing status, check the workflow instance for the job. It is likely that the workflow completed but there is no workflow task to transition the job to Completed or Failed.

Table 13. Tips for operating DataConnect (continued)

Tip	Description
DC_PATH column in the staging table can be used to set a hierarchy on insert. On update, the DC_PATH must be the record to update.	<p>The DC_PATH column in the staging table needs to be used during an update. The DC_PATH must be set to the record you want to update. It must be the IBS_SPEC.OBJECT_PATH of the record you want to update.</p> <p>DC_PATH used with Insert can indicate to the DataConnect task where in the hierarchy to place the record. The children must have the parent's path in the DC_PATH. With this path, the DataConnect Task sets the hierarchy. The objects at the top of the hierarchy or objects not in a hierarchy must have nothing in the DC_PATH.</p>
Execution schedule	To avoid impacting database connections and threads available to the Application Server, schedule DataConnect jobs to run during non-peak hours.
Issues found by the DataConnect Agent are logged to the server.log	<p>Most of the common issues with entries in the DataConnect job table (DC_JOB) are handled by the DataConnect Agent. The issue is logged in the server.log and the DataConnect job entry is set to Fail. Additional information can be obtained by turning on debug logging in the DataConnect Agent.</p> <p>After the issue is resolved, the job can be tried again from the Administrator Console by using the Retry Job link.</p>
Job control business object	The business object for the workflow that contains the DataConnect task must be a subtype of triDataConnectJob, or in other words, created in the triDataConnectJob module. If not, the system generates an error.
Number field's UOM	If you set the Staging Table Field property for a Number field with a related UOM field that is defined in the business object, set the Staging Table Field property for the related UOM field.
Staging table fields come from General section	Only fields that are in the General section are supported to be staging table fields. If you want to pull data from other sections, use temporary fields in the business object and use the workflow engine to move the data into the appropriate non-General fields. Otherwise, use a helper business object to pull all of your data and use the workflow engine to move the data into the real business object.
Staging table name	Use the DataConnect managed object in the Administrator Console to find out the staging table name. Click Business Objects with Staging Tables . The database field name is listed along with other information. The Expand Data link provides more information for the selected rows, such as field names in the staging table and jobs that are using the staging table.
Suspended workflow	A workflow cannot be suspended while it processes a DataConnect task.
The DataConnect process can handle business object with smart section association.	If the business object has a smart section that must be populated through the integration, add an extra field to the business object as a staging table field. Populate that field in the staging table with the identifier that can be queried to find the record you want associated to the new smart object. In the workflow, Query the record and Associate it to the new smart object.

Table 13. Tips for operating DataConnect (continued)

Tip	Description
The Initial State is what the smart object is created in; it is not transitioned.	If the integration wants the smart object to be transitioned to the state, set the Initial State to null in the DataConnect task. Within the body of the DataConnect task, use a Trigger Action to transition the object to the intended state.
Tracking errors	<p>You might want to create a record to track errors that are encountered during workflow validations. One approach is to create a custom <code>cstDataMigrationError</code> business object in the <code>triIntermediate</code> module. This business object can include the following fields:</p> <ul style="list-style-type: none"> • <code>triCreatedSY</code>: System-created date and time. • <code>triNameTX [Error Type]</code>: Stores the migration that caused the error. • <code>triLinkedRecordTX</code>: Locator field that links to the actual record that failed validation. Remember to define the association between the error business object and any other business object that is used in the validation. • <code>triDescriptionTX [Error Message]</code>: Holds the details of the specific error. • <p>Then, create a form for the data migration error record.</p>
User Action task	A User Action task cannot be run from within a DataConnect task.
Workflow exceptions	The DataConnect task fails if the workflow encounters a hard exception during workflow processing.

Chapter 2. Integrating data with the TRIRIGA Data Integrator

You can import tab-delimited data from an external system into your IBM TRIRIGA business objects with the Data Integrator tool. You can also make mass updates of existing data with the Data Integrator tool.

Overview of the TRIRIGA Data Integrator process

The IBM TRIRIGA Data Integrator tool is used to import data into your business objects without writing technical scripts. Record data can come from any source, such as a database, a Microsoft Excel file, or an external system. Regardless of the data source, Data Integrator accepts files only in tab-delimited format, in a .txt file.

The Data Integrator tool gives you the ability to capture and upload groups of records into IBM TRIRIGA without manually entering each record one at a time. Importing a preexisting database saves time and avoids the data integrity and quality issues inherent in manually keying records.

Data Integrator always imports data into the IBM TRIRIGA base language tables. The import of translated data into the localized tables is not supported.

Data Integrator can create associations between two records. The parent record must exist in the system before you can create an association. The way that you design the modules in the IBM TRIRIGA system determines what data you can import and associate together.

Data Integrator can also be used to do mass updates of existing records. To update the value of a business object across multiple existing records, you can use Data Integrator to alter the field values in bulk. This method avoids interacting with the user interface to update each record individually.

With all Data Integrator uploads, a notification is sent to the user who initiated the upload. The notification informs them of the success of the upload or that errors occurred. To receive a notification, the user must have the Notifications portal section in their Home Portal.

There are a few system policies to ensure that your data is properly imported. The first row of your tab-delimited file (.txt) or the Header File must contain a column for each field name in the business object where you are importing or updating data. To facilitate creating the Header File, Data Integrator has a built-in tool for creating a file that is compatible with Microsoft Excel which contains the field names in the first row. This Header File becomes the upload template into which all records are entered.

Field types

Certain field types can be used with Data Integrator, and only with the accepted field formatting of the record data. Failure to follow these guidelines results in a partially successful or a failed upload of the records.

If a field type does not have an example in the table, it indicates that the field type is not supported by Data Integrator. Refer to the comments for the reason.

Table 14. Record field types

Field type	Suffix	Use	Import format	Example	Comments
Action Button	AB	An onscreen image that is used as a clickable button.			Not for use with Data Integrator. Defined in the Form Builder only. Modification required.
Boolean	BL	Used to contain a value that is either TRUE or FALSE.	FALSE = not selected TRUE = selected	triReservableBL = TRUE	
Business Object	BO	Used to associate a Location, Organization, or Geography with a record.		BOTypeName (Business Object Type) FullPath (Full Path) RecordName (Record Name) SysModified DateTime (Update Time)	Not for use with Data Integrator. System-generated by workflow.
Classification	CL	Used to contain a value that is selected from a defined hierarchy of values.	Exact spelling and case.	A new classification object requires a hierarchical upload.	Hierarchical Module. Requires Header File to contain: Parent = \Classifications and full path of child Classification. Do not end the path with a slash (\).
Classification Rollup	CR	Used for rolling up sums by classification.			Not for use with Data Integrator. Populated by the system either by workflow or by the Rollup action.
Color	CO	Used to contain a color.	Color value with exact spelling and case.		Enter color values with a number sign (#) followed by the hexadecimal value for the color. For example, black is #000000.

Table 14. Record field types (continued)

Field type	Suffix	Use	Import format	Example	Comments
Control Number	CN	Used to generate and contain unique ID numbers.			Control Numbers are populated by the system on Publish or Create, based on criteria that are specified on the business object's Publish tab. Not required if the Control Number field is auto-populated by workflow or set to populate by default.
Date	DA	Used to contain a date.	Default date format.	triStartDateDA = 01/01/2006	
Date and Time	DT	Used to contain a date and time combination.	Long integer in milli seconds.	triDateDT representing 08/21/2006 15:30:00 = 1156199400000	
Duration	DU	Used to contain the length of a time interval. Described as #Years #Months #Weeks #Days #Hours #Minutes #Seconds.	Expressed in the following format: 8.x - use #w#d#h#m#s 9.x - use #y#M#w#d#h#m#s 10.x - use #y#M#w#d#h#m#s	cstDurationDU representing 5 days, 6 hours, and 30 minutes = 5d6h30m0s	The formats for 8.x, 9.x and 10.x are different. Be sure to adopt the proper format for your system.
Financial Rollup	FR	Used to contain totals from other fields that are involved in financial transactions.			Not for use with Data Integrator. Populated by the system either by workflow or by the Rollup action.

Table 14. Record field types (continued)

Field type	Suffix	Use	Import format	Example	Comments
Image	IM	Used to contain an image.		triImageIM = //Company-1 /fileMyImage Name.jpg	All images must exist in [root install folder]/userfiles/images folder before the upload starts. Image paths down to the file names must match exactly. All image file names must begin with "file" to display properly.
Label Only	LA	Used to specify a label with no corresponding field.			Not for import. Defined in Form Builder only. Modification required.
List	LI	Used to select a value from a list of values.	Exact spelling and case.	triMonthLI = December	
Note	NO	Used to contain arbitrary length formatted text.			Note fields are not supported by Data Integrator.
Number	NU	Used to contain numbers.	Straight numeral with decimal place setting, as necessary.	triGross MeasuredNU = 12000.5	No commas, \$, or other formatting is required, unless decimals exist.
Password	PA	Used to contain text value that can be modified but not displayed in a user interface.	Alpha numeric text.		Encrypted.

Table 14. Record field types (continued)

Field type	Suffix	Use	Import format	Example	Comments
System Read Only	SY	Used to access information about a record and the business object that is used to create it.	Populated by IBM TRIRIGA system workflows. Not required to be imported.		
Text	TX	Used to contain text values.	Straight text.	The only limitation is the character limit of the business object's field. If 30 characters are uploaded to a 20 character field, 10 characters are truncated.	
Time	TI	Used to contain a time of day.	Upload as long integer in milli seconds.	triStartTimeTI representing 9:30:35 PM = 77435000	
UOM	UO	Used to associate a unit of measure to a field or business object.	Exact spelling and case.		
URL	UR	Used to contain a URL.	Full directory path, including http, ftp, or \\.		

Table 14. Record field types (continued)

Field type	Suffix	Use	Import format	Example	Comments
System Tab::Record Information Section					<p>Generally, these fields are populated by workflows. There is a risk of creating a workflow loop by uploading data into these fields, which is why they are read-only in many areas.</p> <p>Some examples include the following fields:</p> <p>LocationName (Graphics Location)</p> <p>OrgName (Organization Name)</p> <p>GeographyName (Geography Name)</p>
Locator Field		A field reference to another business object record.	Requires full path for hierarchical records or the value that is displayed in the field.		Might require an associative upload if the name displayed is not unique in the result set of the query that is used by the locator field.

Publish names

In the Data Modeler, in the BO Mapping tool, the Mapping Properties indicate which fields contain the Publish Name for a record that is created from the business object. The first 100 characters of the combination of field values in a Publish Name uniquely identify the record among all records that are created from the same business object. Tab-delimited files to be processed by Data Integrator must include fields to provide unique values for the Publish Name fields. If the field values for a Publish Name are provided for an existing record, Data Integrator updates the existing record with the additional field values provided in the uploaded file.

If an uploaded tab-delimited file does not provide unique Publish Names, Data Integrator might fail to create a record. Otherwise, it might create a record when it was supposed to update a record, or it might update the wrong record. It is also possible that the first 100 characters from a tab-delimited file might match a record name, but after the first 100 characters, the record name might not match. In this case, the record is rejected and no update or create occurs.

Number fields

Fields that are tied to the sums in multiple-record smart sections specify that the fields in the smart-section field list must be summed. Fields are created in the General section of the business object that is auto-populated with the selected sums from the field list. IBM TRIRIGA uses these fields in some business objects for summary sections and reports. They are Number fields.

The Unit of Measure value for Number fields is controlled at the business object level. You get the default value that is defined in the Data Modeler for that business object.

Control number fields

Data Integrator cannot handle updates to records for business objects with Publish Names that include a control number field with the **Generate on Create** check box selected. In this scenario, Data Integrator is unable to determine that a record from an upload file exists, causing new records to be created each time that the file is uploaded.

Requirements

Data Integrator requires that the Data Import Agent in the Agent Manager of the Administrator Console must be started. It also requires that the file that is being uploaded must contain the proper Publish Name fields for the imported business object.

However, if an upload file doesn't also contain the required fields for that business object, then a user who signs in and opens the record in IBM TRIRIGA sees a required-field error. New records that are imported must also contain required field values to ensure a smooth user experience.

As a result, there are two types of requirements for Data Integrator, functional and usability. Failure to meet either of these requirements can result in undesired consequences. If the functional requirements aren't met, Data Integrator fails on upload. If usability requirements aren't met, the data is successfully uploaded but imposes problems on the user experience.

It is always a good practice to test all Data Integrator files against a copy of a production system data before you conduct the actual upload or update.

Functional requirements

Data Integrator has several functional requirements that affect publish names, tab-delimited file formats, and parent paths. When these requirements are met, the Header File is in a compatible format.

Publish names

All business objects must be published before users can access them. The **Publish** menu item is on the **Tools** tab in the Data Modeler. It is important to include all fields that define a business object's Publish Name in the data upload. Typically, a name or ID field is used as the Publish Name, but other fields can be used.

For example, if the Publish Name for the triPeople business object consists of the surname, given name, and ID, then "Smith, John - 10025884" is the Publish Name.

You can view the Publish Name definition for a business object by opening the business object in the Data Modeler and viewing the BO Mapping properties in the **Tools** menu.

For Publish Names containing control number fields, the Publish Name must be redefined to a unique value that does not use a control number. After the upload is finished, you can change the Publish Name back to using the control number field.

Attention: The Publish Name must not contain a Control Number field when existing records are updated in IBM TRIRIGA. This situation causes a new record to be created instead.

Tab-delimited file format

If all Publish Name fields are in the Header File and spelled properly, the only other functional requirement is that the file must be output into a tab-delimited file. The file extension must be .txt, but the file name can be whatever is preferred.

Parent paths

To upload Hierarchical record data into Hierarchical modules and business objects, the Parent column must contain the full path of the uploaded record's parent. Failure to define a proper Parent path results in orphaned records that do not display in the hierarchy window.

Usability requirements

Although files might appear to be uploaded successfully, the upload is only the first requirement. When a user opens a record, a required field might not be populated, producing an error message. Required fields, calculations, and rollups are all instances where usability might be an issue.

Required fields

Although Data Integrator requires properly spelled and case-sensitive Publish Names to upload data, there are other requirements. For new records, the Data Integrator Header File must contain any required fields. In addition, all fields that are listed in the header row of the Data Integrator file are case-sensitive and require proper spelling.

If required fields are not contained in the Header File, the record might be uploaded successfully. However, a user who signs in and opens the record sees an error message that a required field is missing. Then, the user must populate all required fields before any state transition action can be successfully triggered on the record. Also, if the data upload process transitions the record to a read-only state, the user can't transition the record to an editable state to enter values in the required fields.

You can prevent this scenario by including all required fields in the Header File for upload. You can open the Form Builder tool to identify which fields are required and add them to the Header File.

Calculations

You must trigger a custom workflow to run calculations that populate read-only formula fields. Do not populate read-only fields directly. Instead, trigger workflow logic to run the calculations after upload.

Rollups

You must trigger a custom workflow to roll up information across multiple business objects. Be sure to write a workflow that moves up all the way from the lowest child to the highest parent across all objects within a hierarchy.

Custom Workflows

Custom workflows define all upload logic against a record. Create them ad hoc and trigger them on upload. For example, a workflow can be triggered by a state transition action that leaves the null state, or by an asynchronous workflow on an Associate event.

Attention: Certain workflows, such as those associated with rollups or calculations, must be retired before a Data Integrator upload is started.

Tab-delimited files

A tab-delimited file contains rows of data. Each row of data contains one or more pieces of data. Each piece of data is called a field. Tab-delimited files to be read by the Data Integrator must contain the same number of fields in every row, although not every field necessarily needs a value.

Before Data Integrator starts reading a tab-delimited file, the user assigns a module, business object, and form to the file. This information instructs the Data Integrator where the data in the file must go when uploaded.

The Data Integrator uses the data fields in each row (except the first) of a tab-delimited file to create or update a record in the IBM TRIRIGA Application Platform. If Data Integrator determines that the business object assigned to the file already contains the uploaded record, it updates that record with values from the file. If the Data Integrator determines that the uploaded record does not yet exist, it uses the business object and form that are associated with the file to create the record.

Data Integrator needs to determine which field contains the data from each field from a row of the tab-delimited file. Data Integrator finds out where to put data from each field by looking at the first row, called the Header, of the tab-delimited file. Data Integrator does not use the first row of a tab-delimited file as data to create or update a record. Instead, it expects the fields of the first row to contain the name of a field in the business object that is associated with the file. The field name identifies the field to which the data for that field goes.

The easiest way to generate this tab-delimited file is to use an application that can generate a tab-delimited file and place the field names in the first row. For example, some applications can export or print a report to a tab-delimited file. If you can specify the column heading for each column, that probably means that the application can place the heading value in the fields of the first row. If you're working with an application to generate tab-delimited files for Data Integrator, specify column headings that match the IBM TRIRIGA field names where you

want the field values to go. After you correctly set up the column headings to match the field names of the IBM TRIRIGA object, creating tab-delimited files for Data Integrator becomes an automated process.

If an application cannot put headings on a tab-delimited file, you can still use data from it with Data Integrator. However, the process of preparing the tab-delimited file for Data Integrator requires extra work. If the program you want to import data from cannot put field names in the first row of a tab-delimited file, you can add them by editing the file. Use Microsoft Excel to edit tab-delimited files.

When you edit a tab-delimited file with Excel, Excel might prompt you for some information about the file. Excel asks these questions just to be certain it understands the kind of file you are asking to edit. Excel might ask if the file is Delimited or Fixed Width. Respond to this question by selecting Delimited. Excel might then prompt you to select the delimiters that the file uses. Select tab. Do not select any other delimiter or options.

After you insert a new first row that contains the field names, save the file as a tab-delimited text file. If you previously saved the Excel file in its native format, you can ignore any warnings that the file might contain features not compatible with the tab-delimited format. The tab-delimited file is now ready to be processed by Data Integrator.

You can use Data Integrator to import data from an application that cannot create a tab-delimited file if the application can export data in a format that Excel can read. If the application can create a file that Excel can read, then read the file into Excel. If necessary, add a first row to the file that contains the appropriate field names. Then, save the file as a tab-delimited text file.

If you need to edit a data file to add a first row that contains field names, it is not necessary to manually type the names of the fields. You can use the Data Integrator to create the first row.

Header files

A Header File is an import template where the first header row declares field names into which data is imported. You can create this header in several ways. Most importantly, the template field headers must match the corresponding business object field names in case and spelling and contain the Publish Name fields for the business object.

There are two types of modules in IBM TRIRIGA: Non-hierarchical and hierarchical. The process of creating a Header File is the same for both. However, the exception is that hierarchical uploads require a Parent column that contains the full path of an uploaded record's parent record.

Header files for non-hierarchical modules

Non-hierarchical objects are flat objects that do not have a parent-child association and are not dependent on the existence of a parent object. Examples of IBM TRIRIGA non-hierarchical business objects are triPeople, triSpecification, and triAsset.

You can confirm that a module is non-hierarchical in the Data Modeler by selecting the module and reviewing the module properties.

After you create the Header File, you must have security access to Data Integrator to upload a file. When the steps are followed properly and the data values fit the

formatting of the field, the upload is successful. As with all Data Integrator uploads, the system sends a notification to you about the status of the upload.

Header files for hierarchical modules

Hierarchical objects have parent-child associations where the children depend on the existence of the parent object. Examples of IBM TRIRIGA hierarchical business objects are Geography, Location, and Organization.

You can confirm that a module is hierarchical in the Data Modeler by selecting the module and reviewing the module properties.

Hierarchical modules are unique in that they require an extra field in the header row of the Data Integrator Header File. The key word is Parent, which links the child record to the appropriate parent business object in the tree.

Every hierarchical tree has a root that is the same as the name of the module. For organization business objects, the root is Organization. For location business objects, the root is Location. For geography business objects, the root is Geography.

In addition to importing data, it might be necessary to import or append records that are child records to another business object. For example, building records are children of the location root. Floor records are children of buildings. Space records are children of floors.

On the Header File, manually add an extra column labeled Parent. The key word Parent is one of the reserved words in IBM TRIRIGA and is never found as an option when the header is created.

Creating header files

You can use Data Integrator to create a Header File for a non-hierarchical or hierarchical module.

Procedure

1. From the main navigation bar, select **Tools > Administration > Data Integrator**.
2. Select the module with the business object that the Data Integrator must associate with the file. The field shows all defined modules.
3. Select the business object that the Data Integrator must associate with the file. The field shows all published business objects in the selected module.
4. Select the form (GUI) to be associated with the records in the file. The field shows all published forms for the selected business object. During upload, it is possible that there is no existing record for the selected form for a specific publish name. But if there is another record with the same publish name with another form, the uploaded data is appended to the existing record even though it has a different form.
5. Select **Create Header**.
6. Review the Display Columns screen that appears. It contains a check box for every field in the selected business object or form.
7. Select the fields that you want. As you click 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. By default, the fields in the Column Order section are shown in the order in

which they are selected. The sequence in the Column Order section is the order that the fields are displayed in the Header File.

8. To change the sequence of fields, click to select the name of the field and click the up arrow or down arrow to position the field.
9. When the list of fields in the Column Order section is complete and the fields are in the correct sequence, select **Export**.
10. Select **Save**.
11. Select the destination on your local computer. Although Data Integrator gives the file a .csv extension, the file is technically generated in .htm format. Change the file name extension from .csv to .htm, change **Save as type** to **All Files**, and select **Save**.
12. Find and open the saved file in Microsoft Excel. Notice that the field names you chose are in the first row of the form.
13. If this Header File is for a hierarchical module, type the key word Parent in the first unoccupied cell on the first row. If this Header File is for a non-hierarchical module, skip this step.
 - a. The Parent column contains the full path of the uploaded record's parent. You must enter the full path of the Parent business object for Data Integrator to find the parent and append the new child record in the hierarchy tree.
 - b. Always include the full path of the Parent business object, including the first backslash (\). But do not put a slash (\) at the end of the Parent path.
14. Enter your data into the appropriate columns. If you are not entering data directly into a Header File, skip this step.
 - a. Change all cells in your Microsoft Excel spreadsheet to Text format. With this setting, you can enter any value and avoid the conversion formatting of your data. Select the cell in the upper left corner (above the 1 and to the left of A), which selects the entire spreadsheet. Then, select **Format > Cells > Text > OK**.
 - b. List and Classification values must match exactly in both spelling and case.
15. Save the file as a tab-delimited .txt file. Verify that the file name ends in .txt and that the **Save as type** is **Text (Tab delimited)(* .txt)**.
16. Select **Yes**.

Preparing associative uploads

Importing data records with associations requires more work than hierarchical uploading. With the Associative Upload feature of Data Integrator, you can associate records to each other, such as linking people with organizations, assets with spaces, and assets with people. You can define this link between any two data records by specifying the association string.

Association strings

Before you can create an upload with associations, confirm the existence of association strings that are to be used in an Associative Upload with Data Integrator. Two methods of confirming association strings are by reviewing Associations tabs on records and through the Association Manager.

Every business object has several associations that are defined with other business objects. It is important to understand the relationship verb phrase that is defined to associate two business objects. If necessary, define the association and the reverse

association in the Association Manager first.

Identifying associations

Most IBM TRIRIGA records have an **Associations** tab on their respective forms. You can use this **Associations** tab to identify the record-level associations and their corresponding association strings for a specific record.

About this task

This example looks at the associations for a departmental record.

Procedure

1. From the main navigation bar, select the **Portfolio** menu. The Portfolio page appears.
2. Select **Departments** in the Organizations portal section.
3. Select the name of any record in the Departments page.
4. In the record, select the **Associations** tab.
5. Select the module.
6. Select the business object.
7. Review the results. In the Records list, the application shows all records that have an association from the record open in the form. The String value, which is also known as the Association String, is a verb phrase that describes the association that exists between the two records.

Verifying association rules

You can use the Association Manager to confirm the business object-level association rules and their corresponding association strings.

Procedure

1. From the main navigation bar, select **Tools > Builder Tools > Association Manager**.
2. In the Module pane, select the module where you plan to upload records with the Data Integrator.
3. In the main pane, locate and confirm the association rules in the Association column.

Adding associations to header files

You can create associations between records that exist in the IBM TRIRIGA Application Platform environment and the records that are uploaded by Data Integrator. To arrange this process, you can add specific information to define each association to the tab-delimited Header File.

About this task

The specific information identifies:

- The module with the business object that was used to create the record to be associated to the uploaded records.
- The business object that was used to create the record to be associated to the uploaded records.
- The name of the association from the existing records to the new records.

After the upload that uses the association linking is complete, the two records reflect the relationship in the **Association** tab. The association is created "from" the existing record in the Associate column "to" the record that is being uploaded. If a business object-level association exists in the Association Manager for the association string that is specified in the Associate column, a corresponding association is created in the reverse direction.

Data Integrator cannot be used to populate a smart section and certain types of locator fields. You might need to create a custom workflow for each type of association to append the associated record to a smart section or a locator field. Just associating two records does not make them appear in smart sections or locator fields. An exception is a query section that uses the same association.

Procedure

1. Type the key word **Associate** in a new column immediately to the right of the last column of data.
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 fully qualified name of the business object and surround it with double percent symbols. For example, `%%triPeople:triPeople%`. This row reflects the module and business object of the records from which an association is made as `%%Module:BusinessObject%`.
4. In row 3 of the Associate column, enter an association string and surround it with double percent symbols. For example, `%%Manages%`. This string must be the association string that is defined from the `%%Module:BusinessObject%` in row 2 to the business object for the file that is uploaded.
5. In each row of the spreadsheet, in the Associate column, enter the name of the existing record from which the new record is associated.

Preparing workflow tasks for associative uploads

Workflow tasks are not required for associative uploads, since an associative upload can exist on its own to create an association. However, when a record in the Associate column also needs to be appended to a smart section, Data Integrator can also process an associative upload with a workflow.

About this task

In this scenario, a temporary association string in the Associate column triggers a workflow that appends the record to the correct section of the uploaded record or maps anything else. Then, the workflow removes the temporary association that is created by Data Integrator and creates the correct one by mapping the record into the smart section or locator field. After the record is mapped, the IBM TRIRIGA system automatically creates the correct association between the records in the section.

Attention: In your workflow, use an association string that neither one of the objects currently uses. This practice ensures that the workflow is triggered only when the records are associated from the upload and not by any user action. Make sure you use an association string that is in the Association Types list in the List Manager. If needed, you can add your own string to this list.

The directionality of an association is important. An association is created "from" the record in the Associate column "to" the record that is uploaded or updated.

Unless there is a business object-level association for the association that is specified in the Associate column, an association is not created in the reverse direction.

In this scenario, there is only an association (and corresponding Associate event) created "from" the organization record "to" the uploaded or updated triPeople record. Since there was no business object-level association for the association type that is specified in the Data Integrator upload, no association is defined in the reverse direction.

Procedure

1. Inspect the Data Integrator file. In this scenario, the Associate column dictates that the association of type Upload Associate is created "from" the specified organization record "to" the triPeople record.
2. If the Data Integrator file uses an association string that does not exist in the system, you must add it to the Association Types list in the List Manager.
3. To open the List Manager, select **Tools > Administration > Lists**.
4. In the Manage By section, select the **Association Types** list. It might take more than a minute for this list to load.
5. Enter the association string from the Data Integrator file into the **Value** field.
6. Select **Save Entries**.
7. Build a new workflow in the organization module as follows. The Start Task defines the trigger condition for the workflow. In this scenario, when an organization record is associated to a triPeople record with the string of Upload Associate, the workflow runs.
8. Set the Modify Task to map the primary record of the workflow to the secondary record. In other words, the primary record of the workflow is the business object of the Start task. Meanwhile, the secondary record is the secondary business object from the other end of the association in the Start task. In this scenario, you map from Organization to a triPeople record to map the organization record into a smart section in the people record.
9. Select **Edit Map** to open the Object Mapping window.
10. Select **Clear All** to remove all field mappings. It might take a few moments for the mappings to clear.
11. Scroll down to the triPrimaryOrganization smart section.
12. Select the **Search** icon.
13. Set the section to **Source**. This setting maps the source record (Organization) into the section of the target record (triPeople) and creates the reference and association to populate the smart section.
14. Scroll back up and click **OK**.
15. Choose an Associate Records task from the New Task pane. This task is changed to a De-Associate task next.
16. Change the option from **Associate using** to **Remove the association** and the association to **Upload Associate**. This task disassociates the temporary association that is used to trigger this workflow as specified in both the tab-delimited Data Integrator file and the Start task.
17. Set the **Where the associated record is** section to the secondary business object (that is, the triPeople record). If you are uploading embedded records, the De-Associate task must be run before the Modify (Map Record) task to avoid duplicate associations.

18. Publish your workflow. You might need to clear the server's workflow cache if this environment is a Production environment.

Uploading data

You can upload data into the IBM TRIRIGA application with Data Integrator one file at a time or in batch mode. The name of the file that is imported into Data Integrator must not exceed 50 characters.

Uploading data manually

After the Header File is created, populated with record information, and saved as tab-delimited, you are ready to upload the file into Data Integrator.

About this task

The procedure is the same regardless of the type of upload you are running: Associative, Hierarchical, or Non-Hierarchical. The only requirement is to ensure that the proper `Module::BusinessObject::Form` is selected. Failure to select the proper settings results in a failed upload or a successful record upload into the incorrect object.

When a record is first created, it is in a special state called null. Records in the null state disappear after the operation that currently uses the record is done. It is important for the Data Integrator to run an action on newly created records to take them out of the null state. If the action does not take the record out of the null state, the record will disappear immediately after it is created. All of the options in the **Action** list are actions that transition records out of the null state.

Procedure

1. From the main navigation bar, select **Tools > Administration > Data Integrator**.
2. Verify that the values in the **Module**, **Business Object**, and **Form** fields match the Header File. The value of **Import Type** must be **Add**. The **Add** value means that if a row contains values for a record's Publish Name fields that match an existing record, the other values in the row update the record. If a row contains values for a record's Publish Name fields that do not match an existing record, the data in the row is used to create a new record.
3. From the **Action** list, select the action to run on a record after the Data Integrator creates or updates it. This list contains the valid actions for a new record, based on the state transition family that is used by the selected business object.
4. If a workflow must start after the Data Integrator creates a record, you must start the workflow with a sub action that is attached to the action in the **Action** list.
5. The only value of **File Type** is tab delimited (*.txt). But from the **File Char Set** list, select the character set in which the imported file was written.
6. Verify that the **Batch Upload** check box is cleared. Clearing this option tells Data Integrator to read the specified data file only one time.
7. Select **Browse** to find and open the Header File that you created.
8. Select **Upload File**.
9. Select **OK**. The data is read from the specified file. After the upload process is complete, a notification is sent to the Notifications section of your Home Portal.

10. Verify that all records were imported correctly by viewing the related portal or page.

Uploading data with batch uploads

With batch uploads, you can automate the upload process and schedule events so that the IBM TRIRIGA Application Platform automatically uploads the defined file according to the scheduled settings. This function is handy for automating the integration of data from an external system without requiring hardcoded scripts or integration development.

Batch upload settings

The `TRIRIGAWEB.properties` file contains settings for Batch Upload that define the paths to input, processing, output, errors, and log files.

The folders that are specified in the **BatchInputLoc**, **BatchProcessLoc**, **BatchOutputLoc**, **BatchErrorLoc**, and **BatchLogLoc** parameters in `TRIRIGAWEB.properties` must exist and the names must match exactly.

Before you implement batch uploads, run `CreateBatchProcess.bat` (Windows) or `CreateBatchProcess.sh` (Linux) scripts in the `[TRIRIGA installation directory]\userfiles` directory to create the default directories that are required to do the processing.

A batch upload is usually the best choice when the Data Integrator periodically reads data files that are all organized the same way into the same type of records.

Scheduling batch uploads

After the Header File is created, populated with record information, and saved as tab-delimited, you are ready to automate the upload process and schedule batch uploads with Data Integrator.

About this task

When a record is first created, it is in a special state called null. Records in the null state disappear after the operation that currently uses the record is done. It is important for the Data Integrator to run an action on newly created records to take them out of the null state. If the action does not take the record out of the null state, the record disappears immediately after it is created. All of the options in the **Action** list are actions that transition records out of the null state.

Often it turns out that the interval you specify between times that the Data Integrator runs divides evenly into 24 hours. If so, data is read from the file at the same times every day.

If you choose an interval that does not divide evenly into 24 hours, such as 5 hours, data is read from the file at the same times every day. In other words, every day, at the time that is specified in the **Reset clock every day at** field, data is scheduled to be read from the specified file. It is then rescheduled at the specified interval. For example, assume that the upload process begins every day at 9:00 AM. If you set the interval for 5 hours, 0 minutes, the batch process would run starting at 9:00 AM and increment every 5 hours for the next 24 hours. Therefore, it would run every day at 9:00 AM, 2:00 PM, 7:00 PM, 12:00 AM, 5:00 AM. It would then start again at 9:00 AM, iterating through the same time increments again.

A running agent runs about every 10 seconds. The agent iterates through all the batch process objects that are created and based on the current time, runs the batch objects at the appropriate time. Since the exact time when the agent runs is unknown, the IBM TRIRIGA Application Platform puts a 1-minute buffer on the start time of each time increment. In the example, if the agent runs at 9:00:10, the process runs since the platform checks between 9:00 AM and 9:01 AM. In the example with a 5-hour increment, the batch files are run if the current time is between 9:00 AM - 9:01 AM, 2:00 PM - 2:01 PM, 7:00 PM - 7:01 PM, 12:00 AM - 12:01 AM, and 5:00 AM - 5:01 AM.

Procedure

1. From the main navigation bar, select **Tools > Administration > Data Integrator**.
2. Verify that the values in the **Module**, **Business Object**, and **Form** fields match the Header File. The value of **Import Type** must be **Add**. The **Add** value means that if a row contains values for a record's Publish Name fields that match an existing record, the other values in the row update the record. If a row contains values for a record's Publish Name fields that do not match an existing record, the data in the row is used to create a new record.
3. From the **Action** list, select the action to run on a record after the Data Integrator creates or updates it. This list contains the valid actions for a new record, based on the state transition family that is used by the selected business object.
4. If a workflow must start after the Data Integrator creates a record, you must start the workflow with a sub action that is attached to the action in the **Action** list.
5. The only value of **File Type** is tab delimited (*.txt). But from the **File Char Set** list, select the character set in which the imported file was written.
6. Select the **Batch Upload** check box. The fields that are labeled **Upload Batch Data Every ___ Hours ___ Minutes** control how often the Data Integrator is scheduled to read data.
7. In the **Upload Batch Data Every** field, set the time increment for the upload intervals that start from the time in the **Reset clock every day at** field. The default is 0 hours and 0 minutes.
8. Set **Reset clock every day at** to the time you want the batch upload process to start each day. If **Reset clock every day at** contains a value greater than 24 hours, the upload occurs only one time per day at the designated **Reset clock every day at** time.
9. Specify the Header File name in the **Filename (*.txt)** field. The file must be in a directory on which the IBM TRIRIGA Application Platform runs. If the designated file is found in the appropriate location, the platform processes the file and places the process time in the **Last Run Time** field of the Data Upload business object.
10. Enter the **Confirmation eMail** address. After each attempt that the Data Integrator makes to batch upload a file, it sends an email to the address specified. This email is sent whether or not the upload is successful.
11. Select **Upload File**.
12. Select **OK**. The platform schedules the upload of the specified file, and continues to schedule the file upload at the specified interval indefinitely. The data is read from the specified file. After each upload process is complete, an email notification is sent to the **Confirmation eMail** address.
13. Verify that all records were imported correctly by viewing the related portal or page.

Batch upload termination

If you want to change any parameters for a batch upload, you must first end the scheduling of the batch upload and then create a replacement batch upload. From the main navigation bar, select **Tools > System Setup > System > Data Upload** to delete the batch upload records.

Tips for operating Data Integrator

You can use the following tips to enhance how your company operates the IBM TRIRIGA Data Integrator tool.

Table 15. Tips for operating Data Integrator

Tip	Description
Associate existing records	After all records are created, reuse the Header File to create an Associative upload Header File. Establishing associations after all records are created eliminates issues that surround nonexistent records.
Checking for data upload errors	Every time the Data Integrator runs and has errors, it creates a Data Upload Error record. To find a Data Upload Error record, select Tools > System Setup > System > Data Upload Error . Scroll through the records in the result page. Click the file name hyperlink.
Confirm through BO result page	After the data produces a successful notification, review the data through the BO result page. For hierarchical objects, ensure that the child records appear under the proper parent in the hierarchy window.
Data Integrator cautions	Data Integrator does less checking of the data than a form into which you manually enter data: <ul style="list-style-type: none">• It creates or updates records with missing values for required fields.• It allows any value in a list field, even if the value is not in the list.• It does not run requested verifications on field values. If an upload does not explicitly set the organization and geography of the records it creates, the security access to those records might not be what is intended. You might need to include organization or geography fields in flat file records to ensure proper security access to created records.
Data Upload records	Every time that you use the Data Integrator to initiate the reading of data, it creates a Data Upload record. If you initiate a schedule of batch uploads, there is still just one Data Upload record. After an upload completes, you can learn how the upload went by looking at the Data Upload record that corresponds to the upload. The Data Upload record has a smart section that contains Data Upload Error records. If any errors occurred, you see them in the smart section. To find a Data Upload record, select Tools > System Setup > System > Data Upload . Scroll through the records in the result page. Click the file name hyperlink.
Folder cleanup	System Administration processes for Data Integrator batch upload must include periodic cleanup of the input and log folders.

Table 15. Tips for operating Data Integrator (continued)

Tip	Description
<p>Managing files for upload</p>	<p>When files are uploaded in batch mode, the upload happens without the direct participation of a person. The platform looks for the files on the computer it runs on rather than on a computer that a person is using.</p> <p>To identify the computer and the directory that is used for batch data uploads, you might need to consult with the person who administers the IBM TRIRIGA Application Platform environment for your organization.</p> <p>To ensure that files are properly processed, observe the following rules that regard when and how to put a file in the directory for processing:</p> <ul style="list-style-type: none"> • Before you put a file in the directory for batch upload, first check to see whether there is already a file with the same name in the directory. If there is, it means the last file with that name was not processed. After a file is processed, the file is moved to another directory. • To ensure that a file in the directory is correctly processed, wait for the file to disappear before you put a new file with the same name in the directory. • When it is time to put a file in the directory, do not put it in the directory by copying the file from somewhere else. The problem with copying is that the file gets put in the directory one piece at a time. In other words, if a file is copied into the directory there is a duration when only part of the file is in the directory. If the IBM TRIRIGA Application Platform tries to process the file when it is not all there, it might not process the file correctly. • Instead of copying the file into the directory, first put the file somewhere else on the same file system as the directory. Then rename or move the file into the directory. When you rename or move a file within the same file system, no file copying takes place. Instead an entry is added to the directory that points to the file where it already is.
<p>Processing an entire batch</p>	<p>It is possible to have a workflow that processes all the records in the batch after the individual records are created or updated. When an upload of a tab-delimited file is finished, the platform runs an Apply action on the Data Upload record that is associated with the upload.</p> <p>As records are processed, an association is created from the Data Upload record to the affected records. The association name is Data Load Status Of. You can use the Association Manager to define an association with this name from the Data Load Status Of business object to the other relevant business object. If you do so, a workflow that is started by an Apply action on a Data Upload record can process all of the uploaded records.</p> <p>You can use a Retrieve Workflow task to retrieve records that are sorted in the order of their value in a particular field. Therefore, you can use any field in the records to determine the order in which the workflow processes them.</p>

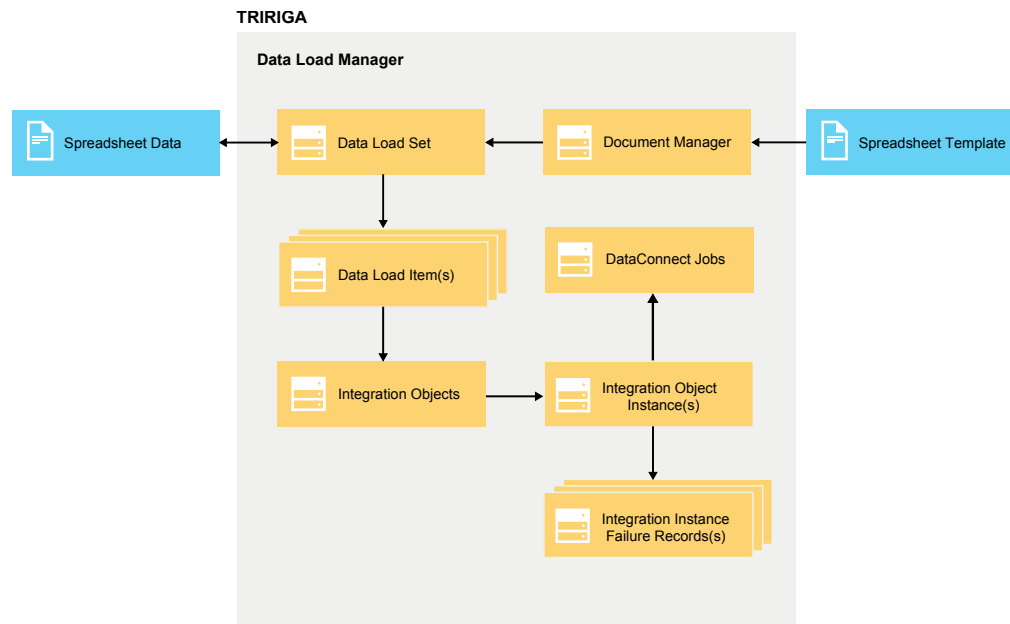
Table 15. Tips for operating Data Integrator (continued)

Tip	Description
Retire formula, rollup, and calculation workflows	In Production environments, all formula, rollup, and calculation workflows must be retired against the business object before the upload is run. This procedure ensures that the records are being created without forcing the platform to run unnecessary workflows against partial data. It also helps eliminate looping that might occur against hierarchical objects. After the upload is successful, republish and run the workflows against your new data set.
Run all files during off peak hours	Depending on the amount of information that is uploaded, server resources can become taxed and performance can degrade. Users might notice a performance hit, which might result in false positives and calls to your help desk.
Timing of record processing	<p>The IBM TRIRIGA Application Platform does not necessarily process uploaded records in the same order they appear in the tab-delimited file. It reads the data from each tab-delimited record in sequence and then uses a different thread to process the data to create or update a record.</p> <p>In most cases, it does not matter in what order records are processed. If you must process the batch records in the same order that they are read, you must force the Data Integrator to process all the records with the same thread.</p> <p>To accomplish this task, set the number of threads that are used by the Data Import Agent to 1.</p>
TRIRIGAWEB.properties for batch upload	The TRIRIGAWEB.properties file contains file settings for batch upload, specifically for input, processing, output, errors, and log. The files that are defined in TRIRIGAWEB.properties must exist and the file names must match.
Upload records first	Do not upload a Data Integrator file with all fields and all associations in one file. Create all records for all business objects first. After you import all records, begin the task of associating the records to one another.
Upload sets of records	Do not push massive amounts of data in a single Data Integrator file. Use sets of 5000 records per Data Integrator file. This practice reduces errors that are related to memory and system capacity.
Uploading images	<p>If you must assign images to newly uploaded records, you can use the following format. Assume that the image field is named triImageIM.</p> <pre>triImageIM //Company-1/file123.jpg</pre> <p>The file name must start with the word "file" such as filemyimagename.jpg or file12345.gif.</p> <p>Based on the example, the file must be placed in the [TRIRIGA install directory]\userfiles\Company-1 directory before you upload the Data Integrator file.</p>

Chapter 3. Integrating data with the TRIRIGA Data Load Manager

You can import data from an external system into your TRIRIGA database with the Data Load Manager. The data load process includes configuring and loading multiple business objects sequentially, providing a mechanism to monitor the data load progress, and identifying errors.

The following figure illustrates how you can use the Data Load Manager to define the interactions between the data load sets, data load items, integration objects, and DataConnect jobs. You can also generate a spreadsheet that is customized for data load sets. By using the spreadsheet, you can organize and validate your external data before you load it into TRIRIGA.



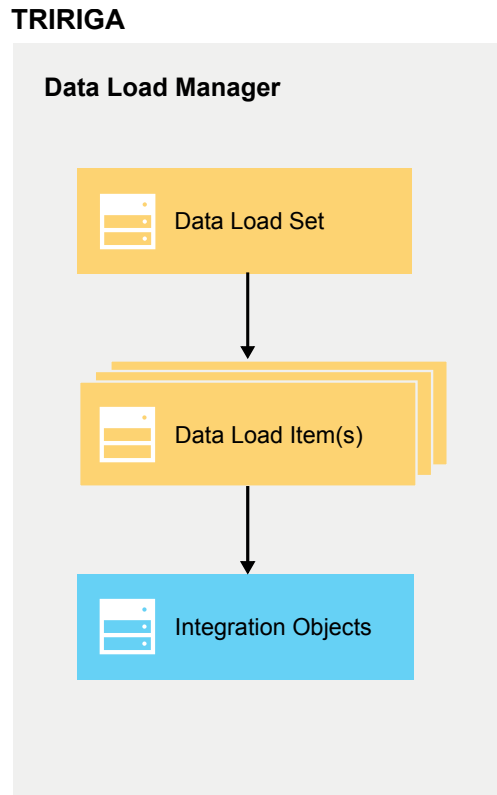
Note: The TRIRIGA Data Load Manager supports only integration objects that are defined with the **File to DC** (DataConnect) scheme. The functions of creating and uploading data load spreadsheets and adding new data load items are designed specifically for integration objects that are defined with the **File to DC** scheme. These functions do not work for integration objects that are defined with other schemes. Any other functionality is only tested by using integration objects with the **File to DC** scheme and might not work with integration objects that are defined with other schemes.

Data load sets

A data load set is the TRIRIGA business object that is used to define the groupings of integration objects that are processed sequentially.

For example, "Geography" might be the name of a data load set that can contain references to integration objects for "World Region", "Country", "State", and "City". You can add any integration objects to a data load set, and the integration objects do not necessarily need to be related in any way.

The following figure illustrates the relationships between data load sets, data load items, and integration objects.



Creating data load sets and data load items

When you create a data load set, or select an existing data load set, you must define the data load items and the sequence for processing these data load items. A data load set uses data load items to reference integration objects. The data load set is used to map the data from the external source to existing TRIRIGA business objects.

Before you begin

Before you create new data load items, you can first specify the default options for the integration object by modifying the **DC Load - Template** integration object.

Procedure

1. Go to **Tools > System Setup > Integration > Data Load Manager > Data Load Sets**.
2. Add a data load set.
3. Specify the data load set name and the process type.
4. Add existing data load items or create new data load items to include in the data load set.
 - a. Specify the load module, load business object, staging table form, and the DataConnect job.
 - b. Create and save the data load item.
5. Specify the processing sequence for the data load items.
6. Create and save the data load set.

7. Process the integration.
8. Inspect any errors.

Process data load sets in the Data Load Manager

You can integrate data into TRIRIGA by processing multiple data load items from a single data load set form, or by processing data directly from a data load item form. Data load items within data load sets are processed sequentially.

A data load set is the TRIRIGA business object that is used to define the groupings of integration objects that are processed sequentially. Processing an integration object generates one or more integration object instances, DataConnect jobs, and integration object instance failure records.

Processing data by using the Data Load Set form

You can process data from all data load items in a data load set. You can attach data files to each of the data load items to be processed. Based on the specified process type, processing occurs in one of the following ways:

Process entire set

Sequentially process each data load item in the "Ready" status in the data load set.

Process set until failure

Sequentially process each data load item in the "Ready" status in the data load set. Processing runs until either one of the data load items fails or the end of the data load set is reached.

Process next item

Process only the data load item in the "Ready" status with the lowest sequence number.

When processing is complete, evaluate the process history and process failure log to see whether the records from the data files were loaded.

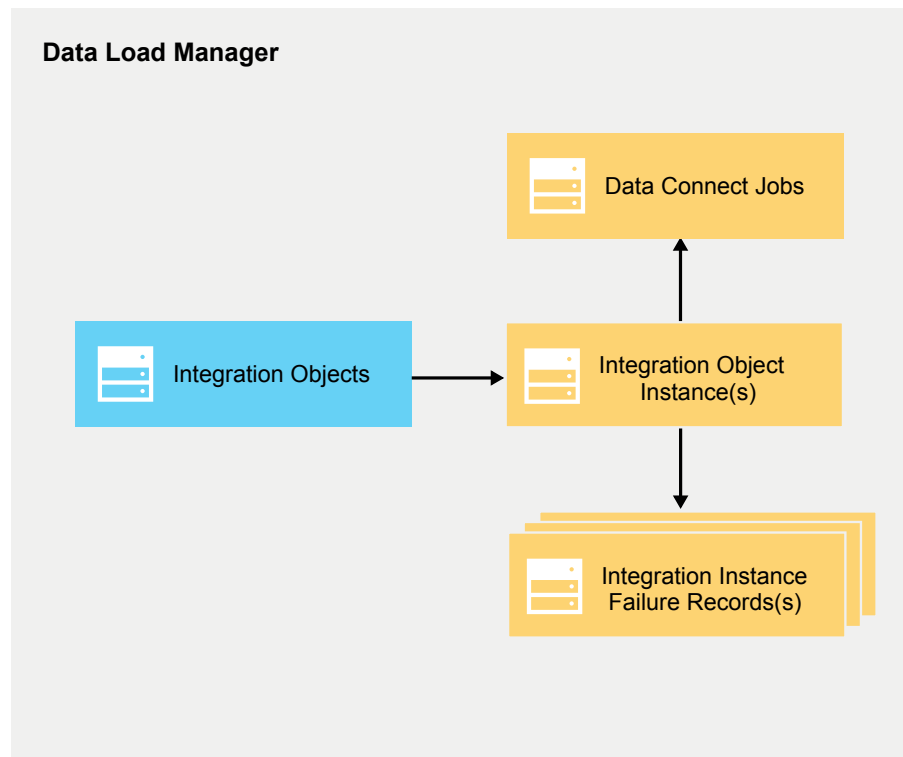
Processing data by using the Data Load Item form

You can process data directly from an individual data load item. You can attach a data file to the data load item to be processed.

When processing is complete, evaluate the process history and process failure log to see whether the record from the data file was loaded.

The following figure illustrates the process:

TRIRIGA



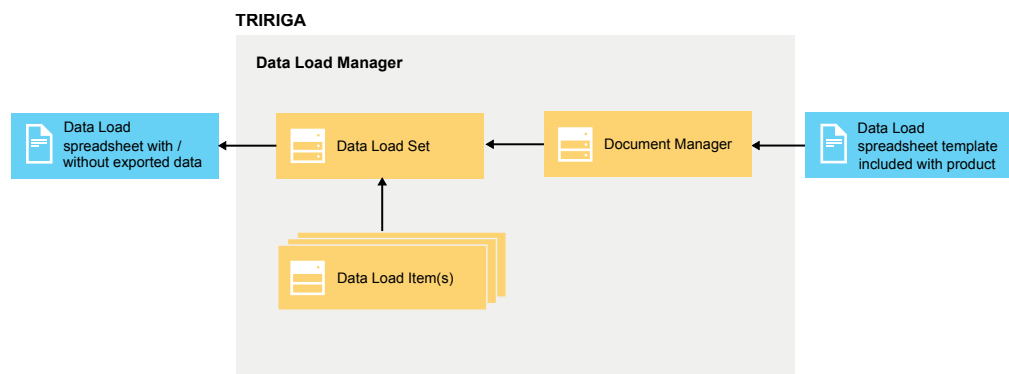
Generating spreadsheets from the Data Load Manager

You can generate data load spreadsheets to collect and validate your data load sets for use in TRIRIGA.

Data load spreadsheets serve the following functions:

- To provide a customized spreadsheet for a data load set to organize and validate your external data before you upload it into TRIRIGA.
- To export existing records from TRIRIGA by using the data load spreadsheet. This usage allows you to validate these existing records against the external data that is uploaded.

The following figure illustrates how you can use the Data Load Manager to generate data load spreadsheets.



Creating a data load spreadsheet

You can generate a spreadsheet that is customized for a data load set to organize and validate your external data before you load it into TRIRIGA. You can also generate a spreadsheet that contains existing records that are already in TRIRIGA.

Procedure

1. Go to **Tools > System Setup > Integration > Data Load Manager > Data Load Sets**.
2. Open the data load set record that you want to create a spreadsheet for.
3. To export existing records from TRIRIGA into the spreadsheet, select the **Export Data** check box in the Create Data Load Spreadsheet section of the form. Data is not exported to the staging table columns **DC_...** (excluding **DC_PATH**), **Smart section DataConnect** (generated when **Used by DataConnect** is selected in Data Modeler), **..._UOM** (unit of measure), and **..._UOMTYP** (unit of measure type). These staging table columns can be used to upload data into TRIRIGA.
4. Click **Save** and then click **Create**.
5. In the **Spreadsheet File** field, click **View Content** to view the spreadsheet.

Results

When you view the spreadsheet, each business object that is referenced by the data load set has the following sheets in the spreadsheet:

Overview

Includes macro descriptions and logging configuration data.

Classifications

Used to store classification data.

Lists Used to store list data.

Locators

Used to store locator data.

ObjectHierarchy

Used to store information about hierarchical data.

Fields Used to store field-level metadata.

Data The column headers in the data sheets correspond with the fields that are specified in the data map of the integration objects that are referenced by the data load set.

Data load spreadsheet validation macros

The data load spreadsheet template includes a set of macros that you can run to validate business object data before you load the data into IBM TRIRIGA. Running the validation macros verifies data conditions.

When the spreadsheet is created, you can download the file, input data and run any of the macros that are identified in the following table:

Table 16. Spreadsheet validation macros

Macro	Keyboard shortcut	Run from	Description
ValidateDataSheet	Ctrl-Shift-V	Fields or data sheet	Validates data on a single sheet in the workbook. Uses the fields sheet to validate data that is specified in the corresponding data sheet. All conditions that are specified are checked. Detailed error messages are added to the VALIDATION_ERRORS column in the data sheet and are also added to a log file. A tab delimited file is automatically generated for the data sheet.
ValidateAllDataSheets	Ctrl-Shift-A	Any sheet	Validates data on all sheets in the workbook.
ClearAllValidationErrors	Ctrl-Shift-C	Any sheet	Clears the error messages that are contained in the VALIDATION_ERRORS column of all data sheets.
GenerateTextFile	Ctrl-Shift-T	Data sheet	Generates a tab delimited text file for a data sheet. The drive and path parameters that are specified on the Overview sheet are used to define the location of the file. The default location for the files is in the C:\TRIRIGA folder. If the folder does not exist, it is created. The file that is generated can be used to load data into TRIRIGA. The generated file follows the naming format: B0Name.txt

Table 16. Spreadsheet validation macros (continued)

Macro	Keyboard shortcut	Run from	Description
GenerateDCPathAllSheets	Ctrl-Shift-G	Any sheet	Generates DC_PATH values for all data sheets in the workbook. It uses the DC_PATH_PARENT field to generate DC_PATH values. The DC_PATH_PARENT field identifies the name of the parent record by using the value of TRINAMETX field in parent sheet. All records in the hierarchy path must be included in the spreadsheet to generate the DC_PATH. The Parents row in the corresponding fields sheet is used to determine which sheets to search for a TRINAMETX value that matches the value in the DC_PATH_PARENT column. The ObjectHierarchy sheet identifies the root record in the hierarchy.
HideFieldSheets	Ctrl-Shift-H	Any sheet	Hides all field sheets in the workbook.
UnHideFieldSheets	Ctrl-Shift-U	Any sheet	Unhides all field sheets in the workbook.
ConvertDateToUnixFormat	Ctrl-Shift-E	Fields or data sheet	Converts all date field values to UNIX Epoch numeric values that represent the date in the number of milliseconds from 1 January 1970.

Table 16. Spreadsheet validation macros (continued)

Macro	Keyboard shortcut	Run from	Description
ConvertUnixFormatToDate	Ctrl-Shift-D	Fields or data sheet	Converts all date field values that contain UNIX Epoch numeric values, that is, the number of milliseconds from 1 January 1970, to the corresponding date values.

The following table provides details about the conditions that are verified when data validation macros are run:

Table 17. Conditions that are verified by the spreadsheet validation macros

Condition	Macro description
Required field	When the Required property specified on the fields sheet is true, the macro checks to see if all values for this field in the corresponding data sheet are not null.
Numeric field type	Checks to see if all values for this field in the corresponding data sheet contain numeric values.
Date data type	Checks to see if all values for this field in the corresponding data sheet contain either valid date values or valid epoch values, that is, numeric values greater than 0.
Boolean data type	Checks to see if all values for this field in the corresponding data sheet contain boolean values. The macro converts the column value to lowercase and validates that it is either true or false.
Classification lookup column exists	Checks to see if the Classifications sheet contains a column with a header value that matches the Classification value in the Fields sheet.
Classification value	Verifies that the field value in the data sheet matches a value in the corresponding column in the Classifications sheet.
List lookup column exists	Checks to see if the Lists sheet contains a column with a header value that matches the List value in the Fields sheet.
List value	Verifies that the field value in the data sheet matches a value in the corresponding column in the Lists sheet.
Text field length	Checks to see if any values for this field exceed the size that is specified for the field size. If the value for field size is blank, the validation is not performed.
Locator field lookup column exists	Checks to see if the Locators sheet contains a column with a header value that matches the Locator value in the Fields sheet.

Table 17. Conditions that are verified by the spreadsheet validation macros (continued)

Condition	Macro description
Locator field value	Verifies that the field value in the data sheet matches a value in the corresponding column in the Locators sheet.
Staging table key field	When the value for the staging table key is true for one or more fields in the Fields sheet, the macro uses those fields to generate the unique key for each row in the data sheet and identifies any rows that contain a duplicate key.
Parent hierarchy	Checks to see if the DC_PATH values in a data sheet are specified in one of the parent data sheets. The parent data sheets are identified by the Parents row in the corresponding field sheet. The ObjectHierarchy sheet identifies the top level in the hierarchy.
Duplicate paths	Checks all records in a data sheet and identifies the condition where a row exists with the same DC Path that is used to load this record.

The following table provides details about the properties in the Fields sheet of the data load spreadsheet:

Table 18. Properties in the Fields sheet of the data load spreadsheet

Property name	Property description
Data sheet column	Identifies the name of the corresponding header column on the data sheet.
Field type	Identifies the TRIRIGA data type.
Field size	Identifies the maximum size for text fields. The value is blank for all other fields.
Classification	Identifies the name of the classification and the unique identifier. The format is <code>classificationName: identifier</code> .
List	Identifies the unique identifier for the list.
Required	A value of true is specified to indicate that the field is required.
Staging table key	A value of true is specified to indicate that the field is a staging table key.
Locator	Identifies the name of the locator and the unique identifier. The format is <code>locatorName: identifier</code> .
Parents	Identifies all potential parent forms for this business object. Values are only available if the module contains hierarchical data.

Processing and loading sample data by using the Data Load Manager

Sample data load sets and integration objects are included with TRIRIGA. You can modify the samples and use them to load geography, location, organization, people, specification, and asset portfolio data.

Each sample data load set references multiple sample integration objects. Each integration object contains a sample data file with header data and a few rows of sample data that matches the data map. DataConnect jobs and workflows are also included for each data type.

Processing sample data

You can process the data load samples that are included with TRIRIGA. You can then load this sample data by using the Data Load Sets application in the Data Load Manager.

Before you begin

To load sample data, you must first process the data load samples that are included with the product.

Procedure

1. “Configuring the sample data” load sets and integration objects included with the product and using the Data Modeler to configure staging tables.
2. “Modifying the data source” on page 57 to include credentials for your local TRIRIGA database.
3. “Implementing data load changes” on page 57 to the data load items you would like to load into TRIRIGA.
4. “Loading sample data” on page 57 by using the Data Load Sets application.

Configuring the sample data

You can configure the sample data load sets and integration objects that are included with the product and use the Data Modeler to configure staging tables.

Before you begin

To ensure that the DataConnect Agent efficiently loads sample data from staging tables, the TRIRIGA application server administrator can edit the TRIRIGAWEB.properties file and set the property `DATACONNECT_SLEEP_TIME = -10`. Changing the property configures the DataConnect Agent to check for sample data every 10 seconds, the shortest interval supported.

Procedure

1. Go to **Tools > Builder Tools > Data Modeler**.
2. Configure each integration object that you want to process.
 - a. Select the associated business object.
 - b. In the Data Modeler window, select **Tools > Revise BO**. Revise the associated business object by setting the **Staging Table** property to **DataConnect** and save the business object.
 - c. From the **Field List**, identify and select business object fields that are to be added to or removed from the staging table. For each field, select or clear the **Staging Table Field** property and save the field.

- d. Select **Tools > Publish BO** within the Data Modeler window to publish the business object and to create or modify the staging table.

Modifying the data source

You can modify the data source to include credentials for your local TRIRIGA database.

Procedure

1. Go to **Tools > System Setup > Integration > Data Load Manager > Data Sources**.
2. Select the **DB-DataLoad** data source and modify the database credentials.
3. Select the **Test DB Connection** link to verify that you can successfully connect to your database with the modified credentials.
4. Save and close to apply the changes to this data source and to existing integration objects that specify this data source.

Implementing data load changes

You can modify the data load items that you want to load into TRIRIGA. Modification can include changes to data sources, data maps, and data files.

Procedure

1. Go to **Tools > System Setup > Integration > Data Load Manager > Data Load Sets**.
2. Select the data load set that you want to modify.
3. Add or remove data load items from the data load set.
4. Select the data load item that you want to modify.
5. Select the **Integration Object** field to modify the data load item.
6. In the **Data Source** section of the integration object form, select the **Test DB Connection** link to verify that the staging table exists.
7. On the **Data Map** tab, add or remove fields.
8. Save and close the integration object.
9. On the **Details** tab of the data load item form, download and modify the file that is stored in the **Data File** field.
10. Upload the modified file.

Loading sample data

You can load sample data, including geography, location, organization, people, specification, and asset portfolio data, by using the Data Load Sets application.

Procedure

1. Go to **Tools > System Setup > Integration > Data Load Manager > Data Load Sets**.
2. Select the data load set that contains the integration objects that you want to process.
3. Select the process type.
4. Click **Process** to execute the integration objects that are contained in the data load set.

What to do next

To obtain further information on loading sample data, you can take the following actions:

- Refresh the Data Load Items section to view the updated status for each data load item.
- Refresh the process history section to view the detailed processing status. When one or more records are successfully loaded into the staging table, a data connect job is created and the records are processed by the data connect agent.
- Select the **DataConnect Jobs** tab to view the list of jobs that were processed. Alternatively, use the Admin Console to view the job status.
- Select the **Process Failure Log** tab to view any errors that are identified during validation.

Portfolio sample data

TRIRIGA provides data load set and integration object portfolio sample data for use in the Data Load Manager.

The following table provides details about the available data load set and integration object portfolio sample data:

Table 19. Integration object portfolio sample data

Data load set name	Data type	Integration object name
Geography	World Regions	DC - Geography - triWorldRegion
	Countries	DC - Geography - triCountry
	States	DC - Geography - triState
	Regions	DC - Geography - triRegion
	Metropolitan Areas	DC - Geography - triMetropolitanArea
	Counties	DC - Geography - triCounty
	Cities	DC - Geography - triCity
	Location	Property
Buildings		DC - Location - triBuilding
Land		DC - Location - triLand
Floors		DC - Location - triFloor
Space		DC - Location - triSpace
Organization	My Company	DC - Organization - triMyCompany
	Divisions	DC - Organization - triDivision
	Departments	DC - Organization - triDepartment
	External Companies	DC - Organization - triExternalCompany
	Workgroups	DC - Organization - triWorkGroup
People	Employees	DC - triPeople - triEmployee
	Consultants	DC - triPeople - triConsultant
	External Contacts	DC - triPeople - triExternalContact

Table 19. Integration object portfolio sample data (continued)

Data load set name	Data type	Integration object name
Specification	Building Equipment	DC - triSpecification - triBuildingEquipmentSpec
	Consumables	DC - triSpecification - triConsumablesSpec
	Fixtures	DC - triSpecification - triFixtureSpec
	Food Service Equipment	DC - triSpecification - triFoodServiceEquipmentSpec
	Furniture	DC - triSpecification - triFurnitureSpec
	Office Equipment	DC - triSpecification - triOfficeEquipmentSpec
	Point of Sale Equipment	DC - triSpecification - triPointofSaleEquipmentSpec
	Security	DC - triSpecification - triSecurityEquipmentSpec
	Services	DC - triSpecification - triServicesSpec
	Space Standards	DC - triSpecification - triSpaceStandardSpec
	Technology	DC - triSpecification - triTechnologyAssetsSpec
	Telephones	DC - triSpecification - triTelephoneSpec
	Tools & Test Equipment	DC - triSpecification - triToolsandTestEquipmentSpec
	Vehicles	DC - triSpecification - triVehicleSpec
	Work Area Outlets	DC - triSpecification - triWorkAreaOutletsSpec
Asset	Building Equipment	DC - triAsset - triBuildingEquipment
	Fixtures	DC - triAsset - triFixture
	Food Service Equipment	DC - triAsset - triFoodServiceEquipment
	Furniture	DC - triAsset - triFurniture
	Office Equipment	DC - triAsset - triOfficeEquipment
	Point of Sale Equipment	DC - triAsset - triPointofSaleEquipment
	Security	DC - triAsset - triSecurityEquipment
	Technology	DC - triAsset - triTechnologyAssets
	Telephones	DC - triAsset - triTelephones
	Tools & Test Equipment	DC - triAsset - triToolsandTestEquipment

Table 19. Integration object portfolio sample data (continued)

Data load set name	Data type	Integration object name
	Vehicles	DC - triAsset - triVehicles
	Work Area Outlets	DC - triAsset - triWorkAreaOutlets
Space Management Associations	Association	DC - Association - triSpaceAllocation - triSpaceLevelAllocation - Allocates - Locators
	Association	DC - Association - triPeople - triEmployee - Has Primary Location \ Primary Organization - SmartSection
	Association	DC - Association - Data Utilities - triPeopleReportsToDTO - Reports To - DTO

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those

websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample

programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a © (your company name) (year).

Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

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

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

Other product and service names might be trademarks of IBM or other companies.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy/>.



Printed in USA