InterSystems®
Creative data technology

Data Engineering Guide

Version 2.4
2024-05-06

Data Engineering Guide
InterSystems   Version 2.4   2024-05-06
Copyright © 2024 InterSystems Corporation
All rights reserved.

For Support questions about any InterSystems products, contact:

InterSystems Worldwide Response Center (WRC)
Tel: +1-617-621-0700
Tel: +44 (0) 844 854 2917
Email: support@InterSystems.com

Table of Contents

1

Welcome, Data Engineers

As a data engineer for your InterSystems TotalView™ For Asset Management solution, your task is to set up the *data pipeline*, a term that refers to all the infrastructure needed to make data available within the solution. This page provides an overview of that infrastructure, from start to finish.

When data is available within the system, you and data analysts can create analysis cubes and reports based on those tables.

1.1 Data Sources

In InterSystems TotalView For Asset Management, a *data source* is a named configuration item that provides all the information needed to retrieve data from an external source. A classic data source is a database, but another possibility is a financial system that provides an API via which an authenticated user can retrieve data. Yet another option is delimited files, either retrieved from a cloud location or pushed to the local file system. In all cases, the data source definition, usually configured by an administrator, contains all the information needed to retrieve structural information as well as data.

1.2 Schemas

A data source can provide multiple kinds of data, each following a specific structure. The classic case is a database that contains many tables, where each table has a specific structure. Similarly, a specific API call returns data in a specific format. In InterSystems TotalView For Asset Management, each of these structures is a *schema*, and the goal is to create a Data Catalog that consists of the schemas that are relevant to the business (and that are ultimately needed for reports and analysis). As a data engineer, your task is to import schemas from the data sources into the Data Catalog and then refine them and categorize them in ways that are useful to your organization.

To get started with this task, see Importing Schemas.

1.3 Recipes

Recipes describe how to load data from external sources into InterSystems TotalView For Asset Management. Any recipe consists of some or all of the following steps, in order:

1. Staging activities, each of which loads data into a generated staging table.

2. Transformation activities, which can clean data in various ways.

3. Validation activities, which can compare data to desired ranges, as an example.

4. Reconciliation activities, in which you specify comparisons that define a valid reconciliation of the data.

   Transformation, validation, and reconciliation all update the applicable staging table, so that the processing is fully transparent. You can directly examine the staging table at any time.

5. Data promotion activities, which uses SQL to update a final table, based on the contents of one or more staging table or tables.

A recipe can also include custom steps at any stage in the processing.

To get started with this task, see Defining Recipes.

1.4 Analytics Cubes

InterSystems TotalView For Asset Management includes InterSystems IRIS® Adaptive Analytics, a Business Intelligence tool powered by co-development with AtScale. This means that data engineers and data analysts can define cubes based on those tables, and then use those cubes for analytics. See Defining Cubes.

1.5 Snapshots

InterSystems TotalView For Asset Management provides an additional mechanism to support your data needs: snapshots. With snapshots, you can easily save data for later inspection by regulators; the snapshot can pull data from multiple tables as needed, writing them to a snapshot table, and the system applies a tag to the records. The product automatically stores all snapshot runs; that is, a new snapshot run does not overwrite a previous snapshot run.

When you have run a snapshot multiple times, applying a different tag each time, you can examine how that data changes over time. In particular, you can build a cube on the snapshot data, using the tag values as a dimension.

See Defining Snapshots.

1.6 Scheduling

The Business Scheduler provides an easy way to schedule the running of tasks: running recipes, building cubes, and performing snapshots. See Scheduling and Running Tasks.

In the initial phases of implementation, you can configure tasks to be run manually (from the Business Scheduler).

Later, when you want the tasks to be executed on a schedule, it is necessary to define the applicable calendar information and to manage dependencies among tasks. This works as follows:

- To define calendar information, you define *entities*, each which has its own calendar. To simplify scheduling, the product supports a hierarchical system of entities, each of which can have its own business calendar but can inherit calender details from its parent. See Defining Entities.

- To specify dependencies among tasks, you apply a tag to a scheduled task and define dependency expressions in other tasks, referring to that tag. Then a task can be run (on a given day) only after the dependencies are fulfilled. See Managing Task Dependencies.

1.7 See Also

- [Welcome, Data Analysts](#)
- About Your Solution: What Is Not Documented

2

Importing Schemas

This page describes how to import the *schemas* that make up the Data Catalog. Each of these schemas (not to be confused with SQL schemas) is associated with a specific external data source and describes the structure of a single data element such as a table or a delimited file provided by that data source.

After importing a schema, it is generally necessary to edit the details such as the data extraction strategy and field types.

2.1 Adding Tables from a JDBC Data Source

To add one or more tables from a JDBC data source:

1. Click the Data Catalog 🖫 icon in the application menu.

2. Click **Data Schema Importer**.

3. Select a JDBC data source from the **Data Source** dropdown menu.

4. Select an SQL schema (group of tables) from the **Schema** dropdown list.

    The page then displays the tables in that SQL schema.

5. Add tables to be imported in either of two ways:

    • Click **Add All Members**. This option adds all tables in the currently displayed SQL schema (group of tables).

    • Click the Add to Imports ⥅ button in the row next to the table or tables.

    The table or tables are then listed on the right side of the page; the system has not yet imported their metadata.

6. Optionally select a different item from the **Schema** dropdown list and repeat these steps as needed. This process then adds to the list on the right side of the page; the system has not yet imported their metadata.

7. Optionally, to remove a table from the list on the right side of the page, click the Delete 🗑 icon in the applicable row.

8. To import the metadata for the selected tables, click **Import**.

    The system then imports metadata for all the selected tables and displays the **Results of Last Import** tab, which lists all the tables whose metadata it just imported.

9. From the **Results of Last Import** tab, optionally start refining any of the imported schemas. To do so, click the Inspect

    👁 icon in the applicable row. Then edit that item as described in Editing Schemas.

2.2 Adding an SQL Query from a JDBC Data Source

Instead of (or in addition to) adding tables from a JDBC data source, you can directly use a custom SQL SELECT query. To do so:

1. Click the Data Catalog icon in the application menu.

2. Click **Data Schema Importer**.

3. Select a JDBC data source from the **Data Source** dropdown menu.

4. Click the **Query** tab.

5. Type an SQL SELECT statement into the box.

6. Optionally modify **Row Count**, which controls the number of rows that are sampled and displayed.

7. Click **Run Query**.

   The page then displays the query results, which you can use to verify that the query is as expected.

8. To add a schema to the catalog based on this query, click **Save to Catalog**.

   The system then displays a dialog box.

9. For **Name**, type a short, unique name for the new schema. You cannot change this name later.

10. For **Description**, type an optional description of the new schema.

    The system then generates metadata for the query and displays the new schema, which you can now edit as described in Editing Schemas.

2.3 Adding a File-Based Schema

A file-based data source is a UTF-8 encoded file with one record per line, where each line follows a convention that uses a specific delimiter between fields (typically a comma or a tab). This convention implicitly defines a schema. To define this schema within InterSystems TotalView™ For Asset Management, you need to upload and then import a sample file and specify the delimiter and the field names. (Excel files are a special case of a file-based data source in which you do not need to specify the delimiter that separates fields.)

FileDir, ExcelSingleFileDir, and S3Delimited data sources are all file-based data sources.

To add a file-based schema:

1. Obtain a sample file for the schema.

   The sample file can consist of only one line. Also, the sample file must have a header row—an initial line that contains the names of the fields. (In other words, the sample file may or may not contain any actual data.)

2. Rename the sample file so it has an appropriate and useful short filename, because this short filename becomes the schema name within the system (with punctuation characters removed), and it cannot be edited.

   For example, if you load a sample file named sampledata.csv, that becomes a schema named sampledatacsv.

3. Click the Data Catalog icon in the application menu.

4. Click **Data Schema Importer**.

5. Select the appropriate file-based data source (the data source to which this schema belongs).

   The **Available Members** section lists any files that you have previously uploaded to the file system. (Specifically this section lists the files in the Samples subdirectory for this data source.)

   Here you can do either or both of the following:

   • Upload new schema files to the file system, possibly overwriting the files listed in **Available Members**

   • Import the files listed in **Available Members** (creating or updating schemas)

   No single file can be larger than 500 MB.

6. Drag and drop one or more files from your local computer to the box labeled **Upload file(s) no bigger than 500 MB**.

   Or click this box, browse to the directory that contains the files, and then select one or more files.

   As you do this, the list in **Available Members** is updated.

7. Click the Add to Imports ≡+ button in the row next to the file or files to import.

8. If the right side of the page contains a long list of files to import, you may want to double check the list. In doing so, if you want to filter this display, you can type into the **Members Selected for Import** filter. This option affects what this page *displays*, but does not affect the import of metadata.

9. Optionally, to remove a file from this list, click the Delete 🗑 icon in the applicable row.

10. To import the selected files, click **Import**.

    The system then imports all the selected files and generates metadata from them.

    The page then displays the **Results of Last Import** tab, which lists all the files it imported.

11. From the **Results of Last Import** tab, optionally start refining any of the imported schemas. To do so, click the Inspect

    👁 icon in the applicable row. Then edit that item as described in Editing a Schema.

2.4 Filter Options When Importing Schemas

Depending on the type of data source, there may be a large number of schemas to choose among. The import page provides options to help you filter the display, as follows:

• On the left side, below **Select Schemas to Import**, there are options to filter the schemas that are shown on this side of the page. First, optionally select a different display option from the following set:

   – **All**—all the tables in this SQL schema (default)

   – **In Catalog**—only the tables that are also currently in catalog

   – **Not in Catalog**—only the tables that are *not* currently in the catalog

   The list is updated automatically when you select an option.

   Second, optionally type a string into the **Available Members** filter. This filters the display so that only names containing the given string are shown.

• On the right side, next to **Members Selected for Import**, you can type a string to filter the list shown in this section of the page. This filters the display so that only names containing the given string are shown.

For a JDBC data source, the page also provides the **Filter Selected Schemas** option. For this option, you can select a SQL schema (group of tables) to display, as another way to filter what is shown.

Important: These filters do not affect what is imported when you click **Import**. The only change is to the display of the list on the right side of the page.

2.5 See Also

- Defining Data Sources
- Editing and Managing Schemas
- Defining Categories

3

Editing and Managing Schemas

In InterSystems TotalView™ For Asset Management, a schema describes the structure of a single data element such as a table or a file using a specific file format. This page describes how to edit and manage schemas; another page provides information on importing them.

3.1 Schema Evolution

One of the key objectives is to permit schemas to change while preserving your data. This works as follows:

1. When you import a schema and first edit it, the schema is a draft and cannot be used by recipes.

2. When you publish a schema, it can then be used by recipes—specifically in a staging activity (which loads data into the system).

3. When you define a recipe, each of its activities is initially a draft.

4. When you publish a staging activity of a recipe, the system then generates the table that will store the data to be loaded.

5. When you edit or reimport a schema, you are implicitly creating a draft that is not yet used by anything. The existing schema is unchanged; existing data is also unchanged.

   While a schema is in draft state, you have the option of deleting the draft (and reverting to the previous published version).

6. When you republish a schema, *if* the change affects the structure needed to contain the data, the system automatically increments the version number of the schema and automatically generates a new draft of any recipe staging activity that is affected by this change. The new staging activity draft is not automatically published, and the previous staging activity is unchanged.

7. When you republish a staging activity, the system generates a new table as needed, considering the structural change that has occurred. This leaves the old table (and its data) unchanged.

3.2 Viewing the Data Catalog

To view the Data Catalog and the schemas currently contained in it:

1. Click the Data Catalog ▣ icon in the application menu.

2. Click **Data Catalog Browser**.

   This page enables you to see all available schemas, either as a table or as a set of rectangular cards.

   For each schema, the page displays the name of the data source that contains this schema, the actual name of the schemas, and any categories to which it belongs. If you click **Card View**, the cards also display the descriptions.

3. Optionally filter the display by using the options at the top of the page:

   • **Schema Definition Item Name**—Filters the display to include only the schemas whose short names contain the given string. After typing into this field, click the Find $\mathbb{Q}$ icon.

   • **Data Source**—Filters the display to include only schemas from a specific data source.

   • **Category**—Filters the display to include only schemas labelled with a specific category. Note that schemas can be labelled with any number of categories, including no categories at all.

   • **Recipe**—Filters the display to include only schemas used in a specific recipe.

   • **Drafts** —If selected, displays only the schemas that are drafts.

   • **Reset** —Clears all the filter options.

Click any schema in order to view it or make changes.

3.3 Displaying a Schema

If you have just imported the schema and you are viewing the **Results of Last Import** tab, click the Inspect ⊙ icon in the row for a schema.

Otherwise, to display a schema:

1. Click the Data Catalog 🗒 icon in the application menu.

2. Click **Data Catalog Browser**.

3. Click the schema name.

   If there is a draft of the schema, you can choose to display the published version or display the draft (which is shown as a separate item as in the following example).

Data Source	Item Name
📄 filedir	sampledatacsv
📄 filedir	comparisondatacsv
DRAFT 📄 filedir	comparisondatacsv

3.4 Editing a Schema

To edit a schema, first display it. If the schema has been published, click **Edit Schema** in the upper right to make the schema editable. (If there is a draft of the schema and you are viewing the published version, this button opens the draft version of the schema.)

Then make any of the following changes:

- To add or edit a description, click the Edit $\mathsf{T}\!\vert$ icon next to **Description** and then make edits.

- To add a category, click **Add Category...** and then click a category.

- To remove a category, click the X icon for the category.

- To modify the details in the box on the right, click the Edit ✏ icon in that right box. This displays a dialog box where you can make changes. The details depend on the kind of schema.

  For a *file-based schema*, **Extraction Strategy** is always **Simple Load**, but you can specify the following items:

  – **FileNamePattern**—Specifies which files are read and used as input, for this schema. That is, input files whose names conform to this pattern are processed, and other files are ignored. You can specify any legal filename string or a filename wildcard expression, consistent with the operating system.

  – **Header Row**—Specifies whether the files have a header row. Select **Yes** or **No**.

  For an *SQL-based schema (either a table or a custom query)*, you can specify **Extraction Strategy**, which specifies the records to get from this table or query, when running a recipe that uses this schema. The options are as follows:

  – **General table**—Retrieves all records, whether or not the system has them already.

  – **Append only transactional table with a numeric sequential key** —Retrieves all new records since the last sequential primary key.

    If you specify this, also specify **Extraction Strategy Field**, which must be the name of the field that contains the sequential primary key.

  – **Transactional table with a changed timestamp field**—Retrieves all records changed since the last timestamp.

    If you specify this, also specify **Extraction Strategy Field**, which must be the name of the field that contains the timestamp.

  – **Simple load**

  For an *SQL-based schema (other than one using a custom query)*, you can specify the following *additional* items:

  – **SchemaName**—Specifies the SQL schema (group of tables) to which the source table belongs.

  – **TableName**—Specifies the short table name of the source table.

  – **UsePrimaryKeyPseudocolumn**—Specifies whether to use a pseudocolumn in this table as the primary key. If you select this option, also specify **PrimaryKeyPseudocolumnName**, which must be the name of the pseudocolumn.

  When you are done, click **Save** to save these changes or **Cancel** to discard them.

- To make changes to the schema fields, see the next section.

When you are done with the schema, click **Save Draft** to save these changes or **Delete Draft** to discard them (and return to the previous schema definition).

Note that if you leave this page without saving your changes, the system automatically saves your changes as a draft.

3.5 Editing Schema Fields

For any kind of data source, when you edit the schema, the bottom part of the page displays a table labeled **Schema Fields**. Here you can modify the schema fields as follows:

- To specify the primary key for this schema, click the **Primary Key** check box for all the fields that make up the primary key.

- To modify other details for a specific field, click the Edit Attributes ✎ icon for that field. This displays a dialog box where you can edit the following:

  – **Data Type**—Specifies the data type for this field.

  – **Required**—Specifies whether this field is permitted to be null. If a field is required, an error is thrown if the value is null.

  – **Default Value**—Specifies the value to use if this field is null.

  – **Data Format** (for date, time, and timestamp fields)—Select the format that data is expected to be in for this field.

  – **Min Value** (for double and integer fields)—Specifies the minimum allowed value of this field.

  – **Max Value** (for double and integer fields)—Specifies the maximum allowed value of this field.

  – **Scale Value** (for double fields)—Specifies the number of digits to display after the decimal point. This has no effect on how data is stored.

  – **Length Value** (for string fields)—Specifies the maximum number of characters allowed in this field.

  When you are done with this dialog box, click **Save** to save these changes or **Cancel** to discard them.

3.6 Publishing a Schema

A schema cannot be used in recipes unless it has been published. To publish a schema:

1. Display it.

2. Click **Publish Schema** in the upper right.

3.7 Reimporting a Schema

To reimport a schema:

1. Display it.

2. Click **Re-import Schema** in the upper right. This change automatically creates a new draft of the schema.

3. Edit the schema again as needed.

If the schema has not yet been published, you cannot reimport it. You can, however, delete it and then import it again.

3.8 Deleting a Schema

To delete a schema:

1. Display it.

2. Click **Delete** in the upper right.

3. Click **Delete** to confirm.

You cannot delete a schema that is used in a recipe; if you attempt to do so, you are notified about any recipe activities that depend on this schema. You can then go edit the recipe to remove the dependency.

3.9 See Also

- Importing Schemas
- Defining Categories
- Defining Recipes

4

Defining and Managing Categories

Categories enable you to define groupings of schemas in the Data Catalog; each schema can belong to any number of categories (including zero). Then when you search for a schema, you can use the categories as filters.

4.1 Defining a Category

To create a category:

1. Click the Data Catalog ▣ icon in the application menu.

2. Click **Categories**.

   The system displays any existing categories.

3. Click **New Category** in the upper right.

4. For **Name**, enter a short, unique name.

   Category names cannot differ only by case. For example, you cannot have a category named `sample` and another category named `SAMPLE`.

5. Click **Submit**.

4.2 Renaming a Category

To rename a category:

1. Click the Data Catalog ▣ icon in the application menu.

2. Click **Categories**.

   The system displays any existing categories.

3. Click the Edit ✎ icon in the row for the category.

4. For **Name**, enter a new name.

5. Click **Submit**.

The category name is automatically updated in all places where it is used.

4.3 Deleting a Category

To delete a category:

1. Make sure that no schema belongs to this category. To do this, display the Data Catalog, filter to this category, and edit the schemas as necessary.

2. Click the Data Catalog [icon] icon in the application menu.

3. Click **Categories**.

   The system displays any existing categories.

4. Click the Delete [icon] icon in the row for the category.

5. Click **Delete** to confirm this action.

4.4 See Also

- Importing Schemas
- Editing Schemas

5

Defining Recipes

Recipes describe how to load data from external sources into InterSystems TotalView™ For Asset Management, specifically data for schemas contained in the Data Catalog. Once you have defined and published recipes, you can run them, either on a schedule or manually, thus loading data into the system.

This page discusses creating recipes; another page has details on managing recipes and using the Recipe Dashboard.

5.1 Structure of a Recipe

Any recipe consists of some or all of the following steps, in order:

1. Staging activities. For each staging activity, you select a data source and then the schemas of interest from that data source.

   When the recipe is run, the data for each schema is loaded into a generated staging table. The system automatically uses the extraction strategy specified in the schema. In the recipe, however, you can choose which fields to copy.

2. Transformation activities. For each transformation activity, you select a schema (from the set used in the staging activities). Then, for any fields in that schema, you specify any transformations to apply to the staging table after the data is loaded. A transformation can replace an existing value in a field or can insert a value into a new field that you specify, also in the same table.

3. Validation activities. For each validation activity, you select a schema (from the set used in the staging activities). Then, for any fields in that schema, you specify a validation function to use. The result of the validation checks are saved in new fields that you specify, all in the staging table. Validation errors can halt the recipe or can simply issue warnings, as you choose.

4. Reconciliation activities. For each reconciliation activity, you select a schema (from the set used in the staging activities) and you also select a secondary source (another schema, also from the set used in the staging tables) that contains comparison data. Then you specify the comparisons that define a valid reconciliation of the data.

5. Data promotion activities. For each data promotion activity, you define a set of steps, each of which uses SQL to update a final table, based on the contents of the staging table or tables.

A recipe can also include custom steps at each stage in the processing.

5.2 Staging Tables

As an intermediate step, a recipe involves loading data into generated staging tables. The names of these tables are generated, and the table name format is as follows:

`Staging_`*recipeshort*`_`*stagingshort*`_v`*version*`.`*schemaname*

Where *recipeshort* is the **Recipe Short Name**, *stagingshort* is the short name of the staging activity, *version* is the version of the recipe, and *schemaname* is the name of the schema as seen in the Data Catalog. (Note that `Staging` is a default part of this name but is configurable.)

Any staging table includes the following fields:

- Fields maintained and used internally by the system: %IRISRowID, %BatchId, and %StagingAction. The %BatchId field contains the ID of the batch in which the given record was updated; you use this to identify the records to use in the data promotion activity.

- A set of fields whose names match the field names from the data that is loaded into that table; these fields are automatically added as needed.

- Any additional fields defined within the transformation, validation, and reconciliation activities. These field names start with the prefixes `%T_`, `%V_`, and `%R_`, respectively.

As you create a recipe, you may find it helpful to view the structure of the staging table, particularly when you define the promotion activities.

5.3 Creating a Recipe

To define a recipe:

1. Click the Recipes icon in the application menu.

   The system then displays the Recipes Dashboard, which lists all the recipes.

2. Click **New Recipe** in the upper right.

3. For **Recipe Name**, enter a descriptive name to display in the Recipes Dashboard.

4. For **Recipe Short Name**, enter a short name to be used within the names of the generated tables used by this recipe.

   The system then displays a page where you can add activities to the recipe.

5.4 Adding a Staging Activity

To add a staging activity to a recipe:

1. Display the recipe (from the Recipes Dashboard).

2. Click **Add Staging Activity**.

3. For **Name**, enter a descriptive name to display within the recipe.

4. For **Short Name**, enter a short name to be used within the names of the generated tables used by this recipe.

If you are going to use only one data source, you could use a generic short name like `Load`, but if you are going to use multiple data sources, it may be helpful for the short names to indicate the data source.

5. For **Data Source**, select a data source.

6. Click **Submit**.

   The system then displays a page where you can edit the details of how to use this data source. This page lists all the schemas that belong to this data source, and each schema represents a data asset such as a table, an specific API call, or a specific format of file. You can load data from any number of these data assets, within this staging activity.

7. For each schema of interest, click the schema name in the left. The right side of the page then displays information about the schema, including (if applicable) information about the last time data was loaded from the associated data source.

   Then specify the fields to load into the staging table. To do so, either click **Select All Fields** or click the check box for each field of interest.

8. When you are temporarily done with this activity, click **Save Draft**. Or, if you want to add activities that use the staging table, click **Publish Activity**, which generates the staging table.

9. Click the **Recipe** link in the upper left to return to the recipe.

5.5 Adding a Transformation Activity

To add a transformation activity to a recipe:

1. Display the recipe (from the Recipes Dashboard).

2. Click **Add Transformation Activity**.

3. For **Name**, enter a descriptive name to display within the recipe.

4. For **Target Source**, select a schema; the list includes all schemas used in *published* staging activities within the same recipe.

   The system then displays the fields in the associated staging table, with options to specify transformations for any of the fields.

5. For each field where you want to apply a transformation, select the field name.

   The system then displays options on the right side of the page. Make edits described below and then click **Apply**.

   - **Function Type**—Select one of the following; note that the available options depend on the type of field you are transforming:

     – **Code Table Conversion**—Transforms the data by using a code table, which provides a set of key/value pairs. If the value being converted matches a key in the given code table, that value is converted to the value associated with that key. *Only for string fields.*

     – **Custom SQL Expression**—Transforms the data by using an SQL expression that returns a single value.

     – **Lowercase String**—Converts to lowercase. *Only for string fields.*

     – **Replace Substring**—Performs a substring replacement. *Only for string fields.*

     – **Trim Whitespace**—Trims whitespace (leading, trailing, or both). *Only for string fields.*

     – **Trim Characters**—Trims a specified set of characters (leading, trailing, or both). *Only for string fields.*

     – **Uppercase String**—Converts to uppercase. *Only for string fields.*

- – **Round Number**—Rounds the number to the given number of decimal places. *Only for numeric fields.*

- • **Overwrite**—Select this if you want the new value to replace the old value. In this case, the system will not generate a new field in the staging table.

  Otherwise, leave **Overwrite** cleared and specify a new field name (see below).

- • **Transformation Name**—Specify a short name for this field transformation.

- • **Result Field Name**—Specify a name for the new field that will store this transformed value. In this case, the system will automatically add a field to the staging table; the actual name of the new field is %T_*newname* where *newname* is the name you specified.

- • **Result Field Type**—Select the type for the new result field.

For the **Code Table Conversion** function, also specify:

- • **Code Table Name**—Specify the name of a table that contains key/value data.

- • **Code Table Key Column**—Specify the name of the field in that table that contains the keys.

- • **Code Table Value Column**—Specify the name of the field in that table that contains the associated values.

For the **Custom SQL Expression** function, also specify **SQL Expression** as an SQL expression that returns a single value, such as an expression that combines field names and SQL operators. You must delimit any field name that is an SQL reserved word.

For the **Replace Substring** function, also specify:

- • **Old Substring**— Type the text to be replaced.

- • **New Substring**— Type the replacement text.

For the **Trim Whitespace** function, select **Trim Leading Whitespace**, **Trim Trailing Whitespace**, or both.

For the **Trim Characters** function, specify one or more characters for **Characters To Trim**. Then select **Trim Leading Characters**, **Trim Trailing Characters**, or both.

6. When you are done editing this activity, click **Save Draft**. Or click **Publish Activity** so that the recipe will use it.

7. Click the **Recipe** link in the upper left to return to the recipe.

5.6 Adding a Validation Activity

To add a validation activity to a recipe:

1. Display the recipe (from the Recipes Dashboard).

2. Click **Add Validation Activity**.

3. For **Name**, enter a descriptive name to display within the recipe.

4. For **Target Source**, select the applicable data asset by choosing the schema that describes the asset. The list includes all schemas used in *published* staging activities within the same recipe.

   The system then displays the fields in the staging table, with options to specify validation rules for any of the fields. Note that any field can have at most one validation rule, within a given validation activity.

5. For each field where you want to apply a validation rule, select the field name.

The system then displays options on the right side of the page. Make edits described below and then click **Apply**.

- **Function Type**—Select one of the following:

    – **Custom SQL Expression**—In the case, the field value fails validation if an SQL predicate expression returns false. (You can, for example, compare the value in this field to the value in another field in the same record.)

    – **Not Null**—In this case, the field value fails validation if it is null.

    – **Fixed Numeric Range**—In this case, the field value fails validation if it falls outside of a fixed range of numeric values

- **Validation Name**—Specify a short name for this field transformation.

- **Result Field Name**—Specify a name for the new field that will store either 1 (if the field value passes the validation rule) or 0 (if it does not). The system will automatically add this field to the staging table; the actual name of the new field is `%V_`*newname* where *newname* is the name you specified.

    It is useful to use a name that indicates the validation check being used, for example `CheckItemCount`, which becomes `%V_CheckItemCount` in the staging table.

- **Validation Name**—Specify how the system should respond when it encounters a field value that fails this specific rule. A **Fatal** failure halts the recipe. A **Warning** failure logs a warning. Both kinds of failures are visible in the Workflow Inbox.

For the **Custom SQL Expression** function type, also specify:

- **SQL Operator**—Select a comparison operator.

- **SQL Expression**—Enter a scalar SQL expression such as a field name or a constant.

For the **Fixed Numeric Range** function type, also specify:

- **Minimum**— Minimum permitted numeric value.

- **Maximum**— Maximum permitted numeric value.

- **Tolerance**—Specify the tolerance, or the amount by which the value is permitted to be outside of the specified range.

6. When you are done editing this activity, click **Save Draft**. Or click **Publish Activity** so that the recipe will use it.

7. Click the **Recipe** link in the upper left to return to the recipe.

5.7 Adding a Reconciliation Activity

A reconciliation activity uses comparison data that you have loaded (and if needed transformed and validated) within the same recipe.

To add a reconciliation activity to a recipe:

1. Display the recipe (from the Recipes Dashboard).

2. Click **Add Reconciliation Activity**.

3. For **Name**, enter a descriptive name to display within the recipe.

4. For **Primary Source**, select a schema—this indicates the data that you ultimately want to promote and use downstream.

The area below **Primary Source** then lists the fields in this schema, including the field or fields that make up the primary key (shown with a special symbol).

5. For **Secondary Source**, select another schema—this indicates the comparison data to use.

   The area below **Secondary Source** then lists the fields in *this* schema.

   The purpose of this page is to create the *key map*, which specifies how the data will be compared. The key map specifies how to identify records to compare, and it outlines which pairs of fields hold the same or similar values.

6. The first step is to match up the primary keys so that the data can be compared record by record. The easiest way to do this is to click **Auto Match Primary Keys**. When you do so, the **Key Map** area shows the primary key fields from the primary source on the left, with the primary key fields from the secondary (comparison) source on the right.

   This indicates how the data will be compared: a record that has a specific primary key in the one source will be compared to the record that has the same primary key in the second source.

7. Now add fields to the comparison, as follows:

   a. Click a field name under **Primary Source**—this adds the field to the left column in **Key Map** area.

   b. Look for the field under **Secondary Source** that should be used for comparison for this field. Click that field name. The system adds this field to the right column in **Key Map** area.

      Or click **Try to Match Selected Column**. In this case, the system looks for an appropriately named field in the secondary source.

   To remove an item in the **Key Map** area, click its X button.

8. When you are done adding fields, click **Create**. The system then creates the key map for this reconciliation activity. If needed, you can return later and make edits.

   Now the page displays a table listing the fields from the two sources. Here you specify rules used to compare the values.

9. Briefly review the table for familiarity. The **Primary Field** column lists the fields from the primary source, and the **Secondary Field** column list the fields from the secondary source. This information is the same as in the key map. In addition, note that the Data Type field indicates the data types of the fields from the primary source (which is helpful in choosing a comparison function).

10. If many or most of the fields have the same name in the two tables, click **Quick Match All** to quickly create a set of reconciliation rules that you can then review and edit. With this option, the system automatically creates a rule for each pair of fields that have the same name (including case). Each automatically generated rule includes a unique rule name, a comparison function, a generated field name (to hold the comparison result), and (for numeric comparisons), an initial setting for **Tolerance**.

    Some fields may not yet have rules (which is OK).

11. Make edits (as needed) for each rule:

    a. In **Function Type**, choose a comparison function.

    b. For **Rule Name**, click the Edit ✎ icon and then type a unique rule name.

    c. For **Result Field**, click the Edit ✎ icon and then type a unique field name. The reconciliation activity will write a 1 or a 0 to this field, to indicate whether reconciliation was successful.

    d. For **Tolerance** (required for numeric comparison), click the Edit ✎ icon and then type the numeric tolerance.

    e. If failing this reconciliation check should be a fatal error (halting the recipe), select the **Fatal** check box.

12. When you are done editing this activity, click **Save Draft**. Or click **Publish Activity** so that the recipe will use it.

13. Click the **Recipe** link in the upper left to return to the recipe.

5.8 Viewing the Staging Table Details

Before you can add a promotion activity, it is helpful to review the structure of the staging table or tables used by the recipe:

1. Display the recipe (from the Recipes Dashboard).

2. At the bottom of the page, see the list in **Indices and Staging Table Details**.

   This section shows one row for each data source used in the recipe.

3. Click the row for a given data source. The page then lists the generated staging tables associated with this data source. For example:

   Data Source: FS Data Lake Data Source

   Staging_DataLakeRecipe_dsjdbc_v1.Position
   Staging_DataLakeRecipe_dsjdbc_v1.TradeWithIndex
   Staging_DataLakeRecipe_dsjdbc_v1.MarketCurve
   Staging_DataLakeRecipe_dsjdbc_v1.SecurityMaster
   Staging_DataLakeRecipe_dsjdbc_v1.Limits
   Staging_DataLakeRecipe_dsjdbc_v1.EquityPriceDataN
   Staging_DataLakeRecipe_dsjdbc_v1.BondAnalytics

4. Click a table name in this list. The right side of the page then provides details about the fields in this table, along with information about the indices on the table. For example:

   Staging Table Indices

Index Name	Fields	System Index	Build Status	Error Det
%ExtractionStrategyFieldValueIndex	%BatchId,AsOfDate	true	Built	No Error
%PrimaryKeyFieldsIndex	%BatchId,ID	true	Built	No Error
%ExtentIndex		true	Built	No Error

   Staging Table Fields

Name	Type	Description
%BatchId	%String	All records staged by a recipe run (recipe record), no matter the staging table, will have the same batchId which i
%StagingAction	%String	StagingAction indicates the action to be taken by this staging record when updating the Analytical Data Model. A I,U,D which correspond to descriptions 'INSERT','UPDATE','DELETE'
Alias	%String	
AsOfDate	%DateTime	

5. Optionally add indices as needed. To add an index, click **New Index** and then specify an index name and select one or more properties to be indexed.

5.9 Adding a Promotion Activity

A promotion activity loads data from the staging table to a final table, generally by means of custom SQL. The final table must already exist within the system.

To add a promotion activity to a recipe:

1. Display the recipe (from the Recipes Dashboard).

2. Click **Add Promotion Activity**.

3. For **Name**, enter a descriptive name to display within the recipe.

4. For **Run Order**, select or type an integer, which controls the order in which the promotion activities are run.

5. Click **New Promotion Item**.

6. For **SQL Expression**, enter an SQL expression that updates the final table using data from the staging table.

   The right side of the screen provides samples to assist you in writing the query.

7. Click **Create Activity Item**.

8. If necessary, click **New Promotion Item** and add other promotion steps.

9. When you are done editing this activity, click **Save Draft**. Or click **Publish Activity** so that the recipe will use it.

10. Click the **Recipe** link in the upper left to return to the recipe.

5.10 Adding a Custom Activity

A recipe can include custom activities. For example, it may be necessary to have a custom activity that retrieves the data from a data source (for example, to place files within the file system). In such cases, two items are needed:

- Custom code in the form of a business host added within the interoperability engine that system uses internally. The business host must be a business operation or a business process that accepts as input a message of type SDS.DataLoader.CustomActivityRequest. Details are currently beyond the scope of the documentation.

- Configuration of the recipe to use that code at the applicable phase, as follows:

  1. Display the recipe.

  2. Click **Add Custom Activity**. This displays a dialog box where you specify the details.

  3. For **Name**, enter the name of this activity.

  4. For **Target Business Host**, enter the configuration name of the business host.

  5. For **When to Run**, select **Before Staging**, **Before Transformation**, **Before Validation**, **Before Reconciliation**, **Before Promotion**, or **After Promotion**.

  6. For **Enabled**, select this check box if the activity should be enabled.

  7. Click **Submit**.

5.11 Step Mode

You can run a recipe in step mode, a mode that lets you use a subset of data—perhaps a very small subset—so that you can test the recipe quickly from end to end. In this mode, you specify the number of rows of data to load into the staging tables (and then to use in the subsequent processing).

To configure a recipe for step mode:

1. Display the recipe.

2. Click the **Step Mode** toggle.

3. In the box next to **Step Mode**, select or type the number of records to use.

A recipe remains in step mode until you change the **Step Mode** toggle back to off.

5.12 See Also

*   Editing and Managing Recipes

*   Scheduling and Running Tasks

*   Viewing Run History

6

Editing and Managing Recipes

In InterSystems TotalView™ For Asset Management, a recipe describes how to load data into the system. This page describes how to edit and manage recipes; another page provides information on defining them.

6.1 Recipe Changes

One of the key objectives is to permit schemas to change, while preserving your data. Schema evolution is described in detail elsewhere, but note the following:

1. When you edit or reimport a schema, you are implicitly creating a draft that is not yet used by anything. The existing schema is unchanged; existing data is also unchanged.

2. When you republish a schema, *if* the change affects the structure needed to contain the data, the system automatically increments the version number of the schema and automatically generates a new draft of any recipe staging activity that is affected by this change. The new staging activity draft is not automatically published, and the previous staging activity is unchanged.

3. When you republish any draft staging activities, the system generates a new staging table, considering the structural change that has occurred. This leaves the old table (and its data) unchanged.

Independent from any schema changes, a recipe can also require fields to be added to the staging table. The system handles this automatically.

6.2 Viewing the Recipes Dashboard

To view the Recipes Dashboard, click the Recipes icon in the application menu.

This page enables you to see all the recipes, either as a table or as a set of rectangular cards. For each recipe, the page displays the following information.

- **Status**—Status of the recipe, which can be any of the following:

  - **Idle**

  - **Staging**

  - **StagingError**

- — **Transforming**

- — **TransformingError**

- — **Validating**

- — **ValidatingError**

- — **Reconciling**

- — **ReconcilingError**

- — **Promoting**

- — **PromotingError**

Other than **Idle**, these statuses indicate which phase of activity is currently underway (and whether an error has occurred in this phase).

- **Recipe Name**—Name of the recipe.

- **Recipe Short Name**—Short name of the recipe.

- **Active**—Indicates whether this recipe is active (can be scheduled).

- **Activity Drafts**—Indicates the number of draft activities that this recipe currently has.

Click any recipe in order to view it or make changes.

To filter the display, use the options at the top of the page:

- **Recipe Name**—Filters the display to include only the recipes whose full names contain the given string. After typing into this field, click the Find 🔍 icon.

- **Status**—Filters the display to include only the recipes with the given status.

- **Lifecycle**—Filters the display to include only the recipes that have the given lifecycle state. A recipe can be **Active** (the initial state), **Disabled**, or **Deleted**.

  Only **Active** recipes can be scheduled.

- **Reset** —Clears all the filter options.

6.3 Editing a Recipe

To edit a recipe:

1. Click the Recipes 🍲 icon in the application menu.

2. Click the recipe. The system then displays the current definition, including any draft activities.

   If there is a draft activity, you can choose to display the published version or the draft version (which is shown as a separate item as in the following example).

Staging Activities			
Name	**Short Name**	**Data Source**	**Delta Version**
LoadCSV	LoadCSV	filedir	2
DRAFT LoadCSV	LoadCSV	filedir	3

Now you can make any of the following edits:

- To add a new activity, click **Add Staging Activity**, **Add Transformation Activity**, **Add Validation Activity**, **Add Reconciliation Activity**, or **Add Data Promotion Activity**, and then continue as described in Defining Recipes.

  The new activity will not take effect until you publish it via the **Publish Activity** button.

- To edit an existing activity, click it. Notes:

  - If you click a draft that the system generated because of schema evolution, the system displays a message that explains why it generated this new draft.

  - If you are viewing a published activity, you must click **Edit Activity** to make it editable.

  - If you are viewing a published activity *and* there is a current draft for that activity, the **Edit Activity** button displays the draft in edit mode.

  Make changes as needed and save them via the **Save Draft** button.

  The changes will not take effect until you publish it via the **Publish Activity** button.

6.4 Publishing an Activity

To publish (or republish) an activity within a recipe:

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe. The system then displays the current definition, including any draft activities.

   If there is a draft activity, you can choose to display the published version or the draft version.

3. Click the activity that you want to republish.

4. Click **Publish Activity**.

6.5 Disabling an Activity

To disable an activity within a recipe (so that this activity is skipped if the recipe is used):

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe. The system then displays the current definition, including any draft activities.

   If there is a draft activity, you can choose to display the published version or the draft version.

3. Click the activity that you want to disable.

4. Click the slider in the **Actions** column for that activity. Then click **Disable Current Activity**. Here's an example (before):

Disabled	Actions
No	⬤

After:

Disabled	Actions
Yes	◯

6.6 Enabling an Activity

To enable an activity that has previously been disabled:

1. Click the Recipes icon in the application menu.

2. Click the recipe. The system then displays the current definition, including any draft activities.

   If there is a draft activity, you can choose to display the published version or the draft version.

3. Click the activity that you want to enable.

4. Click the slider in the **Actions** column for that activity.

6.7 Deleting an Activity

To delete an activity in a recipe.

1. Click the Recipes icon in the application menu.

2. Click the recipe. The system then displays the current definition, including any draft activities.

3. Click the activity that you want to delete.

4. Click **Delete**. (Or if you are deleting a draft activity, click **Delete Draft**.)

   Note that system-generated drafts cannot be deleted; system-generated drafts are created when, for example, an associated schema has changed.

5. Click **Delete** again.

6.8 Disabling a Recipe

To disable a recipe:

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe.

3. Click **Disable**.

4. Click **Disable** to confirm.

6.9 Enabling a Recipe

To enable a recipe that has previously been disabled:

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe.

3. Click **Enable**.

6.10 Deleting a Recipe

To delete a recipe:

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe.

3. Click **Delete**.

4. Click **Delete** to confirm.

6.11 See Also

- Defining Recipes

- Scheduling and Running Tasks

- Handling Task Errors

- Viewing Run History

7

Defining Snapshots

Snapshots are static copies of data in InterSystems TotalView™ For Asset Management, for future review. This page describes how to define them; another page describes how to edit and manage them.

Once you have defined snapshots, you can run them (loading data into the system), either on a schedule or manually.

7.1 Uses of Snapshots

With snapshots, you can easily save data for later inspection by regulators; the snapshot can pull data from multiple tables as needed, and the system applies a tag to the records.

When you have run a snapshot multiple times, applying a different tag each time, you can examine how that data changes over time. In particular, you can build a cube on the snapshot data, using the tag values as a dimension.

7.2 Snapshot Tables

For each snapshot, the system will generate a table. The table name is based on the snapshot definition, and the columns are based on the query used in the snapshot definition. The table name format is as follows:

`halp_snapshot_`*snapshot.tablenameinteger*

Where *snapshot* is the **Short Name** of the snapshot and *tablename* is the short table name, both of which you provide when defining the snapshot. At the end of the name, *integer* indicates the version number. (Note that `halp_snapshot` is a default part of this name but is configurable.)

If you change a snapshot definition so that it defines a different set of columns, the system automatically increments the version number and generates a new table that has the required columns. For example, when you define a snapshot, the generated table name may be `halp_snapshot_Sample.SampleTable1`, but if you redefine the query, the new generated table name becomes `halp_snapshot_Sample.SampleTable2`, and so on.

7.3 Defining a Snapshot

To define a snapshot:

1. Click the Snapshot icon in the application menu.

2. Click **New Snapshot Definition**.

   The system displays a dialog box where you specify initial information.

3. Specify the following information:

   - **Name**—*Required.* Specify a unique name for this snapshot.

   - **Short Name**—*Required.* Specify a unique short name for this snapshot, to be used within the fully qualified name of the generated table.

   - **Table Name**—*Required.* Specify a unique short table name (also to be used within the fully qualified name of the generated table).

   - **Tag**—*Required.* Specify a short string that becomes a tag applied to the snapshot when the snapshot is executed. As an example, a tag could indicate the state of the data, such as Preliminary or Final.

   - **Description**—Type a description of this snapshot and its purpose.

   Except for **Short Name** and **Table Name**, you can edit these values later as well.

4. Click **Submit**.

   Now you can define the rest of the snapshot.

5. To specify the query that returns the data for the snapshot:

   a. For **SQL Statement**, specify an SQL SELECT statement that retrieves the values you want in the snapshot. The query can refer to multiple tables. The query can use * where that is syntactically valid.

   b. Click **Parse SQL Statement**.

   The system then tries to parse the SQL statement and determine the structure of the table it will generate and populate. If the system can parse the SQL, the **Columns** section then displays the fields of the new table, and the **Indices** section displays the indices it will generate for the new table.

   If the system cannot parse the SQL, a warning is displayed, and you can edit the query.

6. Optionally add custom indices to this table. To add an index:

   a. Click **New Index**.

   b. For **Index Name**, type a unique index name.

   c. For **Property 1**, select the field to index.

   To create an index on multiple fields, repeat with **Property 2** and so on.

7. If there is a table that you want to lock while performing this snapshot (in addition to the table or tables used in **SQL Statement**):

   a. Click **Add Table**.

   b. For **Table Name**, type the fully qualified table name (not just the short name).

   c. Click **Submit**.

8. Click **Save**.

   Notice that after you have saved a snapshot for the first time, the **Tables To Lock For Snapshot** section lists the table or tables to which **SQL Statement** refers.

Tip: After you run a newly defined snapshot, use the SQL Explorer to verify that the snapshot definition has captured the data you need to save.

7.4 See Also

- Using the SQL Explorer
- Editing and Managing Snapshots
- Scheduling and Running Tasks
- Handling Task Errors
- Viewing Run History

8

Editing and Managing Snapshots

Snapshots are static copies of data in InterSystems TotalView™ For Asset Management, for future review. Once you have defined snapshots, you can run them, either on a schedule or manually. This page describes how to edit and manage them; another page describes the basics of defining them.

8.1 Viewing the Snapshot Dashboard

To view the Snapshot Dashboard, click the Snapshot [icon] icon in the application menu.

This page enables you to see all the snapshots, either as a table or as a set of rectangular cards. For each snapshot, the page displays the following information.

- **Status**—Status of the snapshot, which can be **Idle**, **Running**, or **Error**.

- **Snapshot Name**—Name of the snapshot.

- **Snapshot Location**—Name of the table in which the current version of this snapshot is stored.

Click any snapshot in order to view it or make changes.

To filter the display, use the options at the top of the page:

- **Snapshot Name**—Filters the display to include only the snapshots whose names contain the given string. After typing into this field, click the Find [icon] icon.

- **Status**—Filters the display to include only the snapshots with the given status.

- **Lifecycle**—Filters the display to include only the snapshots that have the given lifecycle state. A snapshot can be **Active** (the initial state), **Disabled**, or **Deleted**.

  Only **Active** snapshots can be scheduled.

- **Reset** —Clears all the filter options.

8.2 Editing a Snapshot

To edit a snapshot:

---

1. Click the Snapshot icon in the application menu.

2. Click the snapshot you want to edit.

   If there is a draft of the snapshot, you can choose to display the published version or display the draft (which is shown as a separate item as in the following example).

3. Click **Edit**.

4. Make changes in the same way as when you create a snapshot.

   If you change the SQL query used in the snapshot, the system again tries to parse the SQL statement and determine the structure of the table it will generate and populate. If the snapshot definition now defines a different set of columns, the system automatically generates a new table that has the required columns. The older table is left alone.

5. Click **Save**.

   Or click **Delete Draft** to cancel your edits and return to the previous snapshot definition.

8.3 Retagging a Snapshot Run

If you have run a snapshot, you can change that tag that is associated with any particular run of that snapshot. To do so:

1. Click the Snapshot icon in the application menu.

2. Click the snapshot.

3. Click **Snapshot History**.

4. In the row for the run that you want to retag, click the Retag icon.

5. For **New Tag Name**, enter the new tag.

6. Click **Submit**.

Also see Viewing the Run History of a Snapshot.

8.4 Disabling a Snapshot

To disable a snapshot so that it cannot be run:

1. Click the Snapshot icon in the application menu.

2. Click the snapshot you want to disable.

3. Click **Disable**.

4. Click **Disable** to confirm.

8.5 Deleting a Snapshot

To delete a snapshot:

1. Click the Snapshot icon in the application menu.

2. Click the snapshot you want to delete.

3. Click **Delete**.

4. Click **Delete** to confirm.

8.6 See Also

- Defining Snapshots
- Scheduling and Running Tasks
- Viewing Run History

9

Defining Entities

A key part of InterSystems TotalView™ For Asset Management is the ability to schedule automated processing, and these schedules generally need to align with relevant calendars. For example, a report may need to get run on every business day but not on holidays or weekends. However, not all regions and not all organizations necessarily use the same calendar.

To simplify the handling of calendars, the system provides the concept of entities. An *entity* defines a calendar; specifically, each entity has a time zone, a fiscal calendar, a business week, and a holiday schedule. An entity can be a child of another entity and can inherit some or all of that data from its parent.

In practice, an entity can represent a region, a subsidiary, or an organization. In any case, the entity defines a set of calendar information that you want to rely on, for scheduling purposes.

9.1 Defining an Entity

To create an entity:

1. Click the Entity Master ⬚ icon in the application menu.

2. Click **Create New Entity**.

   The system displays a dialog box where you can specify the details.

3. Enter the following information:

   - **Enable Entity**—Select this if you want the entity to be available for use in scheduling.

   - **Entity Description**—Enter the name of this entity, which must be unique at the top level.

   - **Time Zone**—Select the time zone to which this entity belongs.

   - **Fiscal Year Start Month**—Select the month that starts the fiscal year for this entity.

   - **Fiscal Year End Month**—Select the month that ends the fiscal year for this entity.

   - **Business Week First Day**—Select the first day of the business week for this entity.

   - **Business Week Last Day**—Select the last day of the business week for this entity.

4. Click **Submit**.

9.2 Defining a Child Entity

To create a child entity:

1. Click the Entity Master ⛓ icon in the application menu.

2. Select the entity to which you want to add a child.

3. Click **Create Child of Current Entity**.

   The system displays a dialog box where you can specify the details.

4. Enter the following information:

   • **Enable Entity**—Select this if you want the entity to be available for use in scheduling.

   • **Entity Description**—Enter the name of this entity, which must be unique within the parent entity.

   • **Use Parent Time Zone?**—Select this if you want the entity to use the time zone defined by the immediate parent entity.

   • **Time Zone**—Select the time zone to which this entity belongs. (Not available if **Use Parent Time Zone** is selected.)

   • **Use Parent Fiscal Calendar?**—Select this if you want the entity to use the fiscal year start and end month defined by the immediate parent entity.

   • **Fiscal Year Start Month**—Select the month that starts the fiscal year for this entity. (Not available if **Use Parent Fiscal Calendar** is selected.)

   • **Fiscal Year End Month**—Select the month that ends the fiscal year for this entity. (Not available if **Use Parent Fiscal Calendar** is selected.)

   • **Use Parent Business Week?**—Select this if you want the entity to use the business week start and end days defined by the immediate parent entity.

   • **Business Week First Day**—Select the first day of the business week for this entity. (Not available if **Use Parent Business Week?** is selected.)

   • **Business Week Last Day**—Select the last day of the business week for this entity. (Not available if **Use Parent Business Week?** is selected.)

5. Click **Submit**.

9.3 Editing an Entity

To edit an entity:

1. Click the Entity Master ⛓ icon in the application menu.

2. Select the entity you want to edit.

3. Make changes as needed. For details on the fields, see the previous sections.

4. Click **Submit**.

Note: Changes to an entity do not affect any tasks that are currently running, because the entity for a task is checked at the start of the task run.

9.4 Editing a Holiday Calendar

To edit the holiday calendar for an entity:

1.   Click the Entity Master ⛗ icon in the application menu.

2.   Select the entity (expanding the entity tree if necessary).

3.   Click **Edit Holiday Calendar**.

The system then shows the applicable holiday calendar. Here you can do the following:

- To add an entry:

    1.   Click **Add**.

    2.   For **Holiday Description**, enter a short holiday name.

    3.   For **Holiday Date**, select a date.

    4.   Click **Submit**.

- To remove an entry:

    1.   Click **Remove** in the row for that entry.

    2.   Click **Remove Holiday** to confirm.

- To deactivate an inherited entry:

    1.   Click **Deactivate** in the row for that entry.

    2.   Click **Deactivate Holiday** to confirm.

- To reactivate a previously deactivated inherited entry:

    1.   Click **Activate** in the row for that entry.

    2.   Click **Activate Holiday** to confirm.

Note: Calendar changes do not affect any tasks that are currently running, because the entity for a task is checked at the start of the task run.

9.5 Deleting an Entity

To delete an entity:

1.   Click the Entity Master ⛗ icon in the application menu.

2.   Select the entity (expanding the entity tree if necessary).

3. Click **Delete**.

4. Click **Delete Entity** to confirm.

9.6 See Also

- Scheduling and Running Tasks

10

Scheduling and Running Tasks

To schedule recipes and other tasks, you use the Business Scheduler, which you can also use to run tasks manually. There are three types of tasks:

- **Recipe**—runs a single recipe

- **AtScaleCube**—rebuilds a cube

- **Snapshot**—performs a snapshot run

10.1 Scheduling a Task

To schedule a task:

1. Click the Business Scheduler ⬚ icon in the application menu.

   The system displays a table that lists the currently defined schedule items; if needed, use the filters at the top of the table to narrow down what is shown.

2. Click **Schedule Task**.

   The system displays a table that lists the items that can be scheduled. If needed, use the filters at the top of the table to narrow down what is shown.

3. Click the row corresponding to the item to schedule.

4. Specify the following information:

   - **Entity**—*Required.* Select the entity whose calendar information you want to use.

   - **Enabled**—Optionally clear this check box if you want to disable this task (for example, if you are not ready to run or schedule it).

   - **Tag**, **Dependency Expression**, and **Dependency Inactivity**—See Managing Task Dependencies.

   - **Workflow Role for Handling Exceptions**—*Required.* Select the role that should receive any workflow items in case of exceptions related to this task.

   - **Email Distribution List for Error Notifications**—Select the email distribution list that should receive messages when this task encounters an error.

   - **Email Distribution List for Success Notifications**—Select the email distribution list that should receive messages when this task runs successfully.

Then for **Scheduling Details**, either select the **Manually Run** check box *or* specify the following information:

- **Type of frequency to run at** —*Required.* Select the option that best describes how often to run this task.

  Depending on your selection, the page may show additional items to control the scheduling, such as the specific days of the week, the specific months, and so on.

- **On Holidays** —*Required.* Select the option that describes how to handle holidays: **Don't Run**, **Run Anyway**, **Run Next Business Day**, **Run Previous Business Day**.

- **Schedule Task End Time**—Optionally select this if you want the task to stop running at a specific date and time. If you select this option, also specify **Task End Date** and **Task End Time**.

5. Click **Save Task**.

10.2 Managing Task Dependencies

There may be specific tasks that you want to run only after other tasks have been run, on the same day; the system provides a way to manage these dependencies. To specify that one task (task A) should not be run until another task (task B) has completed, you use the following system:

- Decide on a unique, short tag that describes this dependency (for example, `backup`). Note that tags are case-sensitive.

- When scheduling the task or tasks that need to occur first (in our example, task B), specify the **Tag** field, using the tag you have decided on.

- When scheduling the dependent task (task A), specify **Dependency Expression**, which uses a special kind of syntax. In its simplest form, the expression is as follows:

  `TODAY(tag)`

  Where *tag* is a tag. For example:

  `TODAY(backup)`

  This expression means that the task should not be started until the completion (today) of a task that is tagged with the `backup` tag.

  You can combine dependency expression via the keywords `AND` and `OR`, along with parentheses. For example:

  `TODAY(tagA) OR (TODAY(tagB) AND TODAY(tagC))`

- In addition to specifying **Dependency Expression**, you can specify **Dependency Inactivity** in seconds.

10.3 Modifying a Task Definition

To modify a task definition (for example, to change the schedule details):

1. Click the Business Scheduler 🗓 icon in the application menu.

   The system displays a table that lists the items that are currently scheduled. If needed, use the filters at the top of the table to narrow down what is shown.

2. Click the three dots ⋮ in the row for the task. This displays a menu.

3. Click **Open Task Details**.

4. Click **Edit Task**.

5. Modify details here in the same way as when you schedule a task.

6. Click **Save Task**.

   The system runs the task (or attempts to) within the next minute. When it does so, it updates the **Run History** section of this page.

10.4 Running a Task Manually

To run a task manually:

1. Click the Business Scheduler 🗓 icon in the application menu.

   The system displays a table that lists the items that are currently scheduled. If needed, use the filters at the top of the table to narrow down what is shown.

   Tasks that can be run manually are displayed here with **Next Run** as **Manual**; these are tasks defined with the **Manually Run** check box selected.

2. Click the three dots ⋮ in the row for the task. This displays a menu.

3. Click **Open Task Details**.

4. Click **Manually Run Now**.

5. Click **Run Task** to confirm.

   The system runs the task (or attempts to) within the next minute. When it does so, it updates the **Run History** section of this page.

10.5 Aborting a Task

To stop a task that is currently running:

1. Click the Business Scheduler 🗓 icon in the application menu.

2. Click the three dots ⋮ in the row for the task. This displays a menu.

3. Click **Open Task Details**.

4. Click **Abort Task**.

5. Click **Abort Task** to confirm.

   The system runs the task (or attempts to) within the next minute. When it does so, it updates the **Run History** section of this page.

Also see Handling Task Errors.

10.6 Deleting a Task

To delete a task definition:

1. Click the Business Scheduler ⏲ icon in the application menu.

2. Click the three dots ⋮ in the row for the task. This displays a menu.

3. Click **Open Task Details**.

4. Click **Delete**.

5. Click **Delete** to confirm.

10.7 See Also

* Defining Entities

* Defining Recipes

* Defining Snapshots

* Defining Cubes

* Filtering and Customizing the Business Scheduler

* Handling Task Errors

11

Handling Task Errors

This page describes what happens and how to proceed when an error occurs during a recipe, snapshot run, or other task.

11.1 When the System Encounters an Error

When the system encounters an error while executing a task (either scheduled or manually run), the system does several things:

- Sends email to a specified email distribution list (as specified in the task schedule).

- (In most cases) Generates a workflow task, and assigns the task to the applicable workflow role (as specified in the task schedule).

- Updates the display within Business Scheduler to indicate the error condition.

- Updates the display of the associated recipe or snapshot to indicate the error condition.

11.2 First Steps

The first step is to claim ownership of the problem and determine its cause, as follows:

1. Open the Workflow Inbox.

2. Click **Unassigned**.

3. Click the task to display its details. For example:
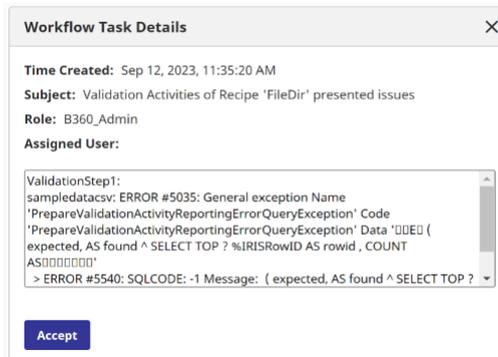
4.   Click **Accept**. This assigns the task to you.

5.   Then see Addressing a Task.

11.3 Viewing the Workflow Inbox

The Workflow Inbox shows all the workflow tasks. To display it, click the Workflow ⬇ icon in the application menu. The system then displays a table with the following fields:

*   **Create time** indicates when the workflow task was generated

*   **Role** indicates the workflow role that owns this task

*   **Assigned To** indicates the user to whom the task is currently assigned

*   **Subject** describes the error

*   **Message** contains more information about the error

Above the table, there are links for showing different sets of tasks:

*   **My Tasks** displays tasks assigned to any of the roles that you belong to.

*   **Unassigned** displays any unassigned tasks.

*   **Completed** displays all completed tasks.

*   **My Completed** displays all completed tasks assigned to any of the roles that you belong to.

There is also a search option you can use to search for a task. The text you enter is matched all the fields displayed on the page.

When you click an item in the Workflow Inbox, the system displays details for that workflow item as shown previously.

You have the following options for specifying the owner of the open tasks that you can see:

*   In **My Tasks**, click the task and then click **Relinquish**. This unassigns the task so that it has no owner.

*   In **Unassigned**, click the task and then click **Accept**. This assigns the task to you.

11.4 Addressing a Task

To address a workflow task, first assign the task to yourself. Read the message to determine what the error is and how you plan to correct it. For example, you may need to do any of the following:

*   Ensure that the file to be loaded has the correct filename pattern and exists in the correct location (in the case of recipes involving a file-based data source)

*   Correct the data being loaded (for example, if a value is out of range is or is the wrong type). This may involve making a correction in a source system.

*   Reimport a schema, if the new data does not match the schema in the system.

*   Adjust a schema (changing data types or required fields, for example).

*   Adjust a step in a recipe.

- Adjust the query used by a snapshot.

Then, depending on the kind of underlying activity, you may have some or all of the following options within the workflow task:

- **Abort**—Halts the task without retrying it.

  Once you have aborted a task, you cannot retry it.

- **Retry**—Retries the task. This is suitable if the data has been corrected or if a missing file has been made available. It is not useful if you have made changes to a schema or to a recipe.

- **Retry All with New Version**—Retries the task with the new version of the recipe, considering new versions of the failed activity and of any later activities. The system will retry the recipe from the point where it failed, using the updated version of that activity and updated versions of any subsequent activities.

- **Retry Failed Activity with New Version**—Retries the task with the new version of the recipe, considering only the new version of one activity in the recipe. The system assumes that the new activity is the one where the failure occurred, and it will retry the recipe from the point where it failed, using the updated version of that one activity. It uses the original version of any other activities in the recipe.

11.5 See Also

- Scheduling and Running Tasks

12

Viewing Run History

You can view the history for any recipes and snapshots that you have run. There are different ways to do this, and the results vary slightly.

From the Business Scheduler, you can easily see the history of any task that has been run, but the data shown is more generic. From a recipe or a snapshot, you can see the run history of that item, with detail that is more specific to recipes and snapshots, respectively.

12.1 Run History from the Business Scheduler

To view the run history of any task:

1. Click the Business Scheduler ⬚ icon in the application menu.

2. Click the three dots ⋮ in the row for the task. This displays a menu.

3. Click **Open Task Details**.

The bottom of the page includes the **Run History** section, which presents the following information:

- **Status**—Indicates the status of the task. This can be any of the following:

  - **Idle**—The task is not running.

  - **Starting**—The task is starting.

  - **RunningWait**—The task is running but is in a waiting state.

  - **Running**—The task is actively running.

  - **Error**—The task has encountered an error and has stopped.

  - **Complete**—The task has completed.

  - **Aborted**—A user aborted the task.

- **Event Time**—Indicates the date and time when this task was started.

- **Elapsed**—Indicates how long the task took (or has taken).

- **Run by User**—Indicates which user ran this task.

- **Error Details**—Displays error details from the last run, if any.

- **Visual Trace**—Provides a link to a trace session (for internal use).

12.2 Run History of a Recipe

You can view the run history of a recipe from the Business Scheduler as described above, or you can view more specific run history as follows:

1. Click the Recipes ![icon] icon in the application menu.

2. Click the recipe.

3. Click **View Run History**.

The **Recipe Run History** section is similar to the previously discussed run history. Here, you can click the Record Report ![eye icon] icon and then see details on the activities.

12.3 Run History of a Snapshot

You can view the run history of a snapshot from the Business Scheduler as described above, or you can view more specific run history as follows:

1. Click the Snapshot ![camera icon] icon in the application menu.

2. Click the snapshot.

3. Click **Snapshot History**.

The system then displays a table listing all the times when this snapshot was run, with date and time information. This information is similar to the previously discussed run history, but also includes details on the number of records included in the snapshot, along with the tag applied to the data, as in the following example:

Snapshot Records

Snapshot ID	Start Time	End Time	Duration	Total Records	Status	Error	Visual Trace	Tag Name	Retag
10	8/30/2023, 5:27:06 PM	8/30/2023, 5:27:06 PM	< 1 sec	9034	Complete		Trace	sampletag	🏷
10	8/30/2023, 5:26:26 PM	8/30/2023, 5:26:26 PM	< 1 sec	9011	Complete		Trace	sampletag	🏷
10	8/30/2023, 5:25:46 PM	8/30/2023, 5:25:46 PM	< 1 sec	8989	Complete		Trace	sampletag	🏷

Here you can retag a snapshot run.

12.4 See Also

- Scheduling and Running Tasks

13

Filtering and Customizing the Business Scheduler

Depending on the number of items defined in the Business Scheduler, you may find it helpful to filter the display, to customize the display, or both.

13.1 Default Display

By default, the Business Scheduler lists all scheduled tasks and displays the following columns of information about those tasks:

- **Status**—Indicates the status of the task. This can be any of the following:

  - **Idle**—The task is not running.

  - **Starting**—The task is starting.

  - **RunningWait**—The task is running but is in a waiting state.

  - **Running**—The task is actively running.

  - **Error**—The task has encountered an error and has stopped. See Handling Task Errors.

  - **Complete**—The task has completed.

  - **Aborted**—A user aborted the task.

- **Task Name**—The name of the task (generally the name of the recipe or snapshot that the task is based on).

- **Enabled**—Indicates whether this task is currently enabled.

- **Type**—Indicates the task type: **Recipe**, **Snapshot**, or **AtScaleCube**.

- **Resource Status**—If the task is running, this indicates the current activity of the task. If the task is not running, this indicates the status of the last run of this task.

- **Exceptions**—If the system encountered an error during its most recent task run, this field displays an error message and includes a link to the Workflow Inbox, where you can see the details.

  For example:

Type ↓↑	Resource Status ↓↑	Exceptions ↓↑
Recipe	⊗ ValidatingError	1 exception(s)

See Handling Task Errors.

- **Entity**—The entity whose schedule this task uses.

- **Last Run**—The date and time when this task was last run.

- **Elapsed**—The length of time taken for this task on its last run.

- **Next Run**—Indicates when or how the task will next be run. This is either a future date and time or is simply the word **Manual** (which means that this task is defined to be run manually).

- **Tag**—The tag applied to this task.

- **Dependencies**—The dependency expression for this task, if any.

13.2 Filtering the Display

To filter the display, use the dropdown menus and type-in boxes above the table.

To clear all filters, click the grid ⊞ icon and then click **Clear Filters**.

13.3 Customizing the Columns

You can customize the columns that the Business Scheduler displays. This customization does not affect other users, and this customization is preserved until you modify it or restore the default set of columns.

To customize the columns:

1. Click the grid ⊞ icon and then click **Toggle Columns**.

   This displays a list of the columns that can be hidden or shown.

2. Clear or select check boxes in this list, depending on which columns you want to hide or show.

3. Click anywhere outside of the list to dismiss the list.

To restore the default set of columns, click the grid ⊞ icon and then click **Reset Columns**.

13.4 See Also

- Scheduling and Running Tasks

- Handling Task Errors

14

Using the File Manager

The File Manager enables you to manage data files and schema files within the InterSystems TotalView™ For Asset Management file system. Specifically, each file-based data source defines and uses an associated set of directories within this file system. Via the File Manager, you can view the contents of those directories (other than the actual file contents), remove files, and upload new files.

Typically you use the File Manager to upload schemas and to upload data for test purposes. In production use, the recommendation is to create an automation that writes files to the applicable directories from the source system or systems. As a consequence, you might use the File Manager only during the early phases of your work with the system.

14.1 Standard Subdirectories for a File Source

Each file-based data source defines and uses its own directory within the file system, given the name that you provide when defining that data source. That directory includes the following standard subdirectories:

- Samples—The purpose of this subdirectory is to hold any sample files that define schemas. You can upload samples to this directory and then import them into the Data Catalog.

  These files remain in this directory until you delete them, if ever.

- Source—This is the input directory for the data source. When you run a recipe that loads data from this data source, the recipe looks for a file in this directory (and the file name must match the filename pattern specified by the associated schema; see Editing a Schema).

  You can upload files to this directory or you can create an automation that writes files here from the appropriate source. You can manually delete files from this directory.

- Work—When a recipe is started, the input file is moved from **Source** to **Work**. This prevents the file from being accidentally processed by a staging activity in a different recipe.

- Archive—When a file has been successfully processed, it is moved from **Work** to **Archive** and is renamed with the timestamp appended to the original name.

  (If a file is not successfully processed, it is moved back to the Source directory.)

14.2 Viewing the Directory Contents for a File Source

To view the directories associated with a file-based data source:

---

1. Click the File Manager ⬆ icon in the application menu.

2. For **Choose a File Directory Source**, select the data source of interest.

   The system then displays a list of folders like this:

List of Folders in the excel1 File Source
Folder Name
/Archive
/Samples
/Source
/Work

   The folder names are always Archive, Samples, Source, and Work.

3. Click any folder name to display the current contents of that folder. For example:

List of Files in the Samples Folder	🔍 Search by file name...	⊕ Upload Files

File Name	**File Size**	**Date Uploaded**	**Actions**
sampledata.xlsx	9.72 KB	8/23/2023, 12:17:57 PM	🗑
comparisondata.xlsx	9.72 KB	8/23/2023, 12:17:39 PM	🗑

4. Optionally edit **Max Number of Files to View**. This change immediately affects the display.

14.3 Uploading Files

To upload one or more files:

1. Click the File Manager ⬆ icon in the application menu.

2. For **Choose a File Directory Source**, select the data source of interest.

3. Select either the **Samples** or the **Source** directory, depending on where you want to load this file.

   The page then displays any existing files in that directory.

4. Click **Upload Files**.

   This displays a new area on the right, the *upload area*.

5. Perform the following steps within the upload area:

   a. Drag and drop one or more files from your local computer to the upload area.

      Or click the upload area and then browse to the directory that contains the files. Then select one or more files.

      As you do this, the upload area displays a list of files that can be uploaded.

      Note that you cannot upload a file larger than 500 MB.

   b. Optionally, if you have selected files that you do not want to upload, click the Remove 🗑 icon next to those files. Or click **Clear Files**.

c. Click **Upload File(s)** in the upload area.

Now the files are uploaded and are displayed in the list of contents for the given directory. The upload area is reset and shows no files.

Note that the new files overwrite any existing files that have the same name. The upload process does not rename files. Also when you upload a new copy of a schema file, that does not automatically reimport the schema. When you are ready to reimport the schema, do that as described in Reimporting a Schema.

14.4 Deleting a File

To delete a file:

1.
   Click the File Manager icon in the application menu.

2. For **Choose a File Directory Source**, select the data source of interest.

3. Select a directory name.

4.
   Click the Delete icon next to the file.

5. Click **Remove** to confirm the action.

14.5 See Also

- Importing Schemas
- Defining Recipes