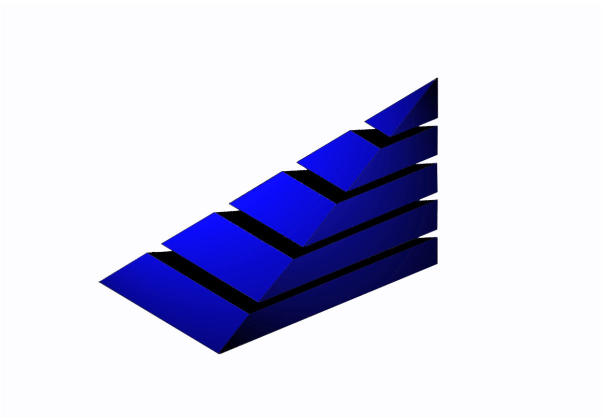
AXIOM SL
software laboratories, inc.

ControllerView®

Version 7

(Training Manual)

Regulatory and Financial Reporting Solution



AXIOM Software Laboratories, Inc., 67 Wall Street, New York, NY 10005, USA Phone: (212) 248-4188, sales@ axiomsl.com, www.axiomsl.com, Fax: (212) 248-4354

TABLE OF CONTENTS

1. INTRODUCTION

There are several agencies, such as the FFIEC and the Federal Reserve, that require banks and bank holding companies to submit regulatory reports on a regular basis. These reports are used to ensure compliance with different standards, to evaluate risk, and to see how the economy is doing overall. In order to complete just one report, a company might need to use data that exists across many different physical locations (servers, databases, etc).

AxiomSL's Controller View software enables implementation specialists to gather the appropriate data from different sources, enrich it to suit the needs of a specific report, and ultimately create a completed report to submit to the regulatory agency.

Controller View is a system designed for data management purposes. It allows the user to upload or manually insert data and organize that data in a way they see fit. Our clients, such as Goldman Sachs, Credit Suisse, RBC Capital, and Morgan Stanley, use this program as an easier and more efficient way to complete Aggregate reporting, Management reporting, Reconciliations, etc. Although it is advertised as a program created to assist Bank Holding Companies with Regulatory Reporting, it can be used for data integration, risk management, and financial control.

As an implementation specialist, you will be actively involved in all parts of this process. You will learn how to load data sources, register data models, create portfolios and aggregations, and use all of these components to produce a "paramount-to-the-financial-industry's-health" regulatory report.

Working with this reporting software, you will need to develop a concrete understanding of both the technical and the financial levels of report creation. This can be accomplished through practice exploration of Controller View and through reading about the forms themselves. If you visit http://www.federalreserve.gov/reportforms/default.cfm, you can gain access to all the forms. You will likely encounter several terms or concepts that seem unfamiliar, but feel free to Google any of these to further your understanding.

This document intends to cover the full ControllerView process covering all the major components of the system starting with the creation and loading of data sources proceeding through all the steps to generate a final report. It is designed to give new users an introduction to the system's capabilities and to illustrate the flexibility of the system while leading you through the procedures in data setup, data model setup, business rules setup (portfolio), aggregations and creation/report analysis.

For the sake of brevity many system features were merely be touched upon. Users should refer to the User's Manual for detailed instructions on main and "auxiliary" procedures, such as various system performance optimization methods, portfolio execution algorithms usage, user permissions setup etc.

Users should realize that it is beyond our scope to provide training in such disciplines as Database administration, SQL programming and regulatory reporting requirements. This manual is a merely a tool to help users to grasp the system's mechanics and should be used for this purpose only and in conjunction with a hands-on training session.

The sections of this manual are arranged following the general workflow of the system. The training begins with Data Source setup and loading, followed by Data Model setup / registration and other data management functions.

After the data warehousing related topics we cover the Portfolio Definition using Visual Business Rules (VBR), Shorthands and Portfolio Result Viewing. Aggregation Setup/Viewing section shows you how to summarize the data and how to drill down and perform analysis.

We will then proceed to the section on Report Generation/Production and Raw Data/Top side Adjustments. Finally we will conclude with the chapter on Process Automation and Workflow Setup/Execution.

Axiom SL is always on stand-by to assist its clients and answer their questions. The company is committed to working closely with the system's users maintaining a healthy and professional relationship with them. We therefore encourage users to come forward with comments and constructive suggestions on how to improve the system and enhance its performance to accommodate their unique needs.

2. KEY TERMS AND CONCEPTS

Below follow brief definitions of the system's key terms and concepts:

2.01 Dynamic Data Warehouse

The Dynamic Data Warehouse is the facility for advanced Data Management. Here users can perform variety of data warehousing tasks, e.g. introduce new data sources, edit data sources and load data, etc.

The Dynamic Data Warehouse provides users with the ability to decide and manage how you would like to physically store each data source in the database, maintain historical data, manipulate the data for data cleansing or enrichment purposes and set up calendar, the archival frequency and various defaults.

2.02 Instance

Instance is the snapshot of any data set for a certain date and time. Instance usually reflects the date and time of the Source download (Data Source Instance), Data Model update (Data Model Instance), or Portfolio or Aggregation execution (Portfolio Instance/Aggregation Instance).

2.03 Data Source

Data Source is a set of data that describes the physical data table's structure and location. Data Sources can be internally produced, imported via file, or referenced by the system.

Data Sources can represent various systems that could be found in financial institutions such as Loan, Deposit, Accounting, General Ledger, Benefits and Insurance systems etc. Each data source is initially introduced to the system via Data Source Setup/Edit screen.

As a result of creating executable objects within the system such as Portfolios, Aggregations and Free Form Reports (see below) the system will automatically generate *resulting* Data Sources, which in turn will be automatically connected to the automatically generated *resulting* Data Models.

The purpose of the resulting Data Sources/Models is so that they may be used in the subsequent steps in the data management process, making your design more efficient and transparent.

2.03.01 Reference (Static) Data Sources

Reference tables containing static data – counterparty information (id, credit rating, contact, etc), general ledger information (codes, descriptions), product type, collateral type, etc. This type of data is called "static" or "reference" data since it typically does not change frequently or regularly.

Reference data is used to enrich the Main Data Source by providing the Lookup facility to the Main Source Layout (see below). For example, if you have a Currency Code column in your transaction table, you may attach to it a Reference Data Source containing a list of all available currency codes and their descriptions to provide a lookup

2.03.02 Data Source Instance

After data is loaded into a data source for a particular time/date stamp, a Data Source Instance is created. A Source Instance is the successful creation of a table in the database. Data loaded on a regular and frequent basis will have a date/time stamp to identify source instances.

Since Static (Reference) data uses the "Permanent" storage method, it does not require a time or date stamp because only one table will exist in the database for this type of data. This table will be created during the initial data load.

Creating a Source instance may be achieved using several methods:

1. By loading data into a data source for a specified time and date using the systems data loading screen.

2. By manually creating a blank table in the database by pressing the "Create Instance" button in the Source Setup (Data Storage tab) and specifying the time and date when prompted by the system.

3. During an automated process, where your scheduler will be responsible for initiating the loading of the data and the time and date stamp will be one of the parameters, which will be automatically passed to the system. Selecting "Create Instance" as the Loader type in the Source setup will prompt the system to create a blank instance every time this data is loaded.

2.04 Data Model

Data Model is a group of related sources used to supply data for calculations. The Data Model can consist of one or several Data Sources.

Data Model is a virtual and logical representation of how related data sources should be joined together and treated by the system. Each data source in a Data Model is represented by a node. These nodes are organized into a "tree" structure, also known as Source Hierarchy.

Data Model always has one Main Source to which additional (auxiliary) sources are joined.

Additional data sources may have their own hierarchies.

2.04.01 Data Model Stream

A Stream is a subset of data based on the 'IN' condition. Streaming enables users to run executable tasks such as Portfolios, Aggregations, Reports and Workflows etc. for a selected set of values based on one or several columns which reside in a data source.

For example a user can run a workflow first against the country code column for the values USA and Canada and then for Holland and France. The combination of USA and Canada would then represent one stream, while Holland and France – another. The purpose of this functionality is to generate multiple sets of results based on some additional criteria or filtering without changing the existing structure and setup of your reports and workflows. This functionality can be used to generate the same report for multiple branches without having to maintain multiple sets of logic or criteria.

2.04.02 Data Model Instance (Instance Registration)

A Data Model Instance is the successful registration of a Data Model for a specified time/date. It is important to note that Data Models are not “loaded” but “Registered”. This process may be performed manually or may be automated by inserting the “Register Model Instance” object into the Workflow and selecting the Data Model you wish to register as a part of the process.

The Data Model may also be registered manually using the “Edit Instances” screen accessible by right clicking on the Data Model object in the Project Management screen’s Object pane.

All tasks in the system are based on Data Models. Therefore registering Data Model instance is a necessary step in order for users to be able to proceed with the rest of their Project.

The Data Model Instance Editor screen allows users to register their data models AS OF a specific date and time. However users may also mix-and-match the actual Source Data, which they want to use in the particular instance of their data model.

For example:
A user has a Data Model consisting of 2 data sources; the first data source contains transactions, which are loaded daily and the second source contains exchange rates, which are also loaded daily. The scenario may arise that the exchange rate source failed to load on a particular day. In this case the user could register the Data Model instance using today’s transactions and previous day’s exchange rates (provided nothing drastic took place on the market).

This will enable the user to generate the report and get results of acceptable degree of accuracy. Should the missing Forex data feed arrive eventually, the user will be able to re-execute the process for the same As Of date with the corrected Data Model instance.

2.05 Project

Project Management module was developed to meet the demand to streamline business management processes. The concept of the Project suggests breaking down the business issues into concise entities. In doing so the managers on every level can better deal with organizational problems when there is continuous accumulation of data tables, task execution results, data models, shorthands, reports and other vital and residual products of everyday business activities.

One of possible purposes of a Project can be generating a specific regulatory report. Such Project will include all executable tasks essential for producing such a report (data sources, portfolios, aggregations etc.). Also included would be supporting tasks designed to automate and facilitate the process (i.e. calendars, workflows, schedulers, etc.).

A project can also consist of a single task, such as Source Loader with sole purpose of providing loaded data to other projects. Another project could include a Portfolio and produce a data subset on which other projects will be based (see dependencies).

2.10 Branch

Branch allows breaking down the business issues further within the same Project. Branches can be created, for example, for different dates that the regulatory reporting forms have been filed. Think of a Branch as a Folder in the Briefcase of a Project.

2.11 Objects

Objects are the building blocks of the Project. They represent information-carrying entities which users put together in order to achieve the ultimate goal of the Project, be it generating a complex regulatory report or a simple data loading workflow.

Objects can be divided into two major categories:

1. **Executable objects**
2. **Auxiliary objects**

2.11.01 Executable Objects

Objects representing the assignment-related procedures (data-generating tasks) are broken down into the following categories:

- Data Source (Data Loading)
- Portfolio
- Aggregation
- Free-form Report
- Tabular Report
- Workflow

The list of tasks is client-specific and can be adjusted individually according to the client’s demands. For example, some clients could elect not to use Tabular reports. This task category therefore will not be included in their system setup.

2.11.02 Auxiliary Objects

Supporting user-defined objects included in the Project are:

- Calendar (mandatory)
- Data Model (mandatory)
- Detail Set (optional)
- Shorthand (recommended optional)

2.12 Dependency

Multiple projects can be dependent (based) on shared objects or tasks that could be data sources, data models, detail sets or entire projects. Instead of including such an object into each and every project, users can create a dependency in which a shared task/object is given a stand-alone status and is referenced by the tasks of all projects that are *dependent* on it.

The advantage of such approach is that should this shared task/object change, it will have to be done only once, followed by updating the tasks of the projects dependent on it.

Note: Tasks and objects inherited from the dependency are marked with an → on the display.

2.13 Freezing Project Branches

If you are satisfied with the existing setup of the Project Branch (e.g. active Call Report) you can put a Freeze on it. When frozen, the tasks can still be executed. No changes however can be made to the tasks of the frozen branch. You can however make a copy of the frozen branch and change it all you want under a different name.

2.14 Submission

To protect the results of the Project Branch execution for a particular date (Project Branch Instance) it may be marked as “Submitted”. This freezes the entire Branch instance including all setups and all calculation results of every task in the submitted branch instance.

Note: Submitted branches open in read-only mode and cannot be changed, executed or copied.

2.15 Favorites

This feature allows you to create Favorites to save tasks, data snapshots and other objects you use often allowing you to bypass lengthy searches through lookup lists. You can save Favorites in two modes:

Private – will be visible and available only to you

Shared – will be visible and available to other users of the current branch.

2.16 Validation

The system allows you to track changes in the Project components. As you build a project, more and more tasks and objects become involved creating ‘workflows’ where the successful execution of any task is based on the result of another. If changes are made to any of the components of the workflow chain, or any component of the chain is deleted, all subsequent elements of this chain become affected and need to be *validated*.

2.17 📁 Portfolio

Portfolio is a tool for selective data retrieval. Portfolios allow you to specify parameters and conditions to isolate any group of transactions and have it follow certain criteria. For instance, you can insert conditions into a portfolio so that it groups transactions by amounts, GAP codes, Maturity date, etc.

Typical Portfolio consists of a hierarchy of nodes. The *root* (main) node (highest in the hierarchy) bears the Portfolio name. A condition can be assigned to the *root* (main) node. This condition will be inherited by every other node in the portfolio. Conditions at the lower nodes should generally be more detailed then the higher nodes.

2.18 Σ Data Aggregation

Data Aggregation is a process of generating the summaries of the source data. In general, when data arrives, it is only viewable on the transaction level. In order to view to data one would have to open the entire source table consisting of hundreds of thousands of records. A more efficient way of seeing the data as they reflect business activities is to view the data in a summarized, or aggregated, form.

For example, you may want to see the totals broken down by one or several details (data fields) such as Transaction ID, Domicile and Profit Center and see the balance summaries that reflect to these requirements. In order to be able to do this you would define an Aggregation task based on the above-described details and execute it against the provided data set.

Drill down capabilities allow you to view the summaries from various angles and on different levels of granularity, including the transactional level.

2.19 Free Form Report

Free Form Report is essentially at the end product of the ControllerView working process. Free Form Report represents a virtual representation of the actual Federal Report while consisting of worksheets that make sure the data in the report is correct (See Chapter 8: Edit Checks for more information).

2.20 Tabular Report

Tabular report writer allows you to define and execute snapshots of the enterprise's transactional activity for a certain date. Reports generated by this subsystem allow combined views of the entire data set (or a part of it if you decide to use logical conditions) from all sources in the data model on all levels of aggregation simultaneously (as opposed to the viewer screens that allow you to see and print out one data source at a time).

2.21 Shorthands

Shorthand is a subset of conditions/values that have been defined and stored for the continuous use. The purpose is so that a user can simply refer to a frequently used set of conditions or values when setting up a data-generating task, such as Portfolio, Aggregation, Report, etc.

Another advantage provided by the shorthands is that should the requirements change, the adjustments would have to be made once to the shorthand and not to the setup of every task, where the set of conditions/values had been used.

2.22 System's Control Buttons

The following system control buttons are frequently encountered when working with the ControllerView.

	Create New		**Open Existing**
	Save		**Save As**
	Save as Favorite		**Exit Current Sheet**
✗	**Delete Current**		**Data Load**
▷	**Execute**		**Rename**
	Look Up		**Adjustments**
	Copy Row(s)		**Paste Row(s)**

→	Add Row	→	Insert Row
▲○	Move Up Selected Row	○▼	Move Down Selected Row
	Insert Node		Delete Selected Node
	View Results	A B	Head Manager
X≥Y	Insert Condition into Cell	DB	Insert Data from Database into Cell
	Insert Field into Cell	● ● ●	Insert Model

3. PROJECTS AND BRANCHES

The concept of the Project suggests breaking down the business issues into concise entities. Doing so allows managers to better deal with organizational problems when there is continuous accumulation of data tables, task execution results, data models, shorthands, reports and other vital and residual products of everyday business activities.

One of possible purposes of a Project can be generating a specific regulatory report. Such Project will include all executable tasks essential for producing such a report (data sources, portfolios, aggregations etc.). Also included would be supporting tasks designed to automate and facilitate the process (i.e. calendars, workflows, schedulers, etc.).

A project can also consist of a single task, such as Source Loader with sole purpose of providing loaded data to other projects. Another project could include a Portfolio and produce a data subset on which other projects will be based (see dependencies).

Branch is a version of the Project tailored to achieve a specific task.

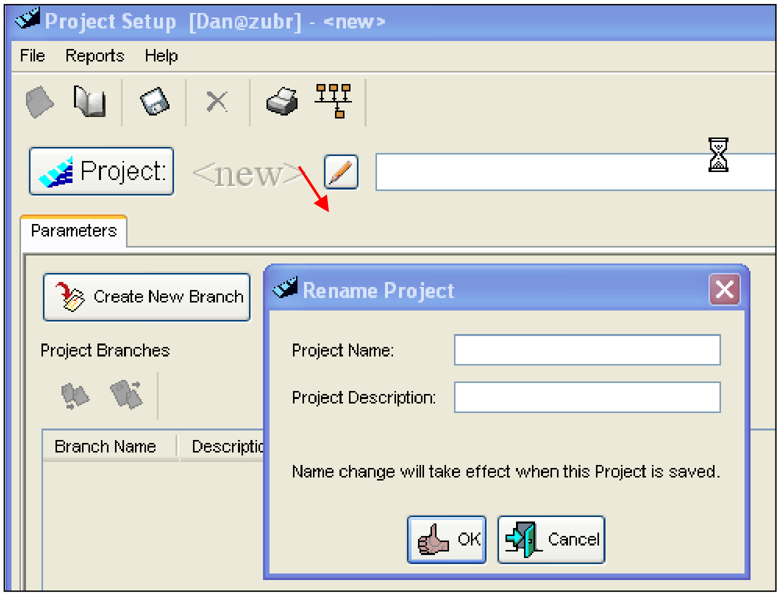
Projects and branches can be likened to briefcases and folders in the system. A Project represents a ‘briefcase’ containing many ‘folders’ (branches).

3.01 Creating a New Project

To create a new project:

1. Click the “File” Menu and select “Create New Project” option.

2. Enter the project name (use underscore as a space) and Project Description by clicking the “pencil” button (Fig. 1).



Name change will take effect when this Project is saved.

Fig. 1

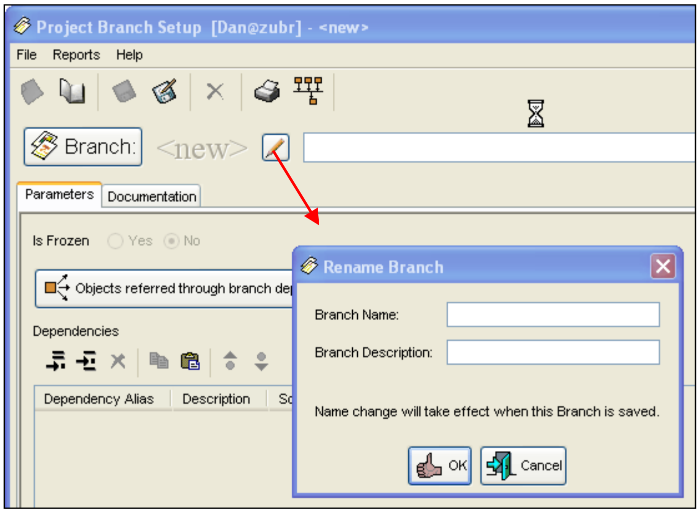
3. Save

3.02 Creating New Branch

To create a new Branch:

1. Click the “File” menu and select the “Create New Branch” option.

2. Enter the new Branch name (use underscore as a space) and branch description by clicking the “pencil” button (Fig. 2).



Name change will take effect when this Branch is saved.

Fig. 2

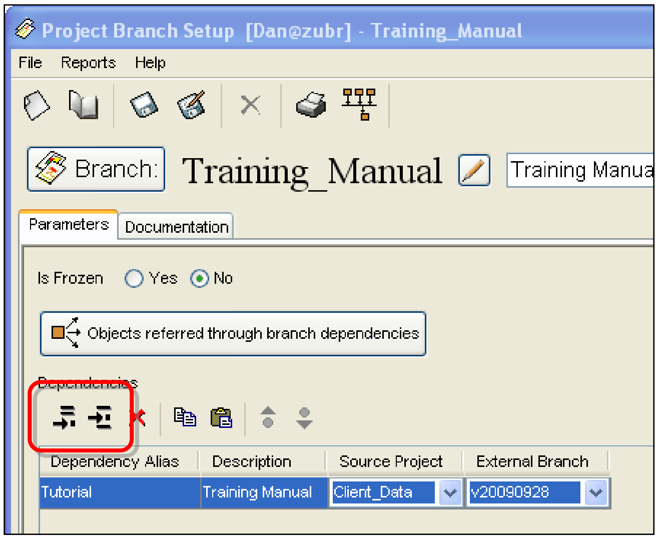
3. Save

3.03 Creating a Dependency

Creating dependencies (Fig. 3) allows you to access data from different branches of the same or other projects. For example, if you wanted to have access to formulas or conditions used in a different project and branch, you would create a dependency relationship with that particular project and branch. To create a dependency:

1. Open the Branch Setup window.

2. Use the Add Row or Insert Row buttons of the Data Grid tool bar.


Project Branch Setup [Dan@zubr] - Training_Manual
File
Reports
Help
Parameters
Documentation
Objects referred through branch dependencies
Dependencies

Branch: Training_Manual Training Manua

Is Frozen ○ Yes ● No

Dependency Alias	Description	Source Project	External Branch
Tutorial	Training Manual	Client_Data	v20090928

Fig. 3

3. Manually enter “Dependency Alias” and “Description” into the corresponding fields.

4. Click on the arrow button for the Source Project input field. The drop down menu with the list of all projects in the system will open.

5. Select the Project you need.

6. Click on the arrow button for the External Branch input field. The drop down menu with the list of all branches in the selected Project will open.

7. Select the Branch you need. In the Fig. 3, our dependency is to the project “Client_Data” and the branch “v20090928”.

4. SOURCE SETUP

Setting up data sources is a mandatory step in order to be able to load the data.
Any data set can serve as a source and the system is capable of loading any type of data whether it is a table in a database or a flat file such as a text or Excel file.

Loading data requires preliminary setup that is performed on the Source Editor screen. If you are working with the brand new project, the objects pane is populated with object categories. In order to access the Source Editor you have to start a brand new source. To do this:
On the Object Pane, right click on the Data Source object category.

Select the New option from the menu (Fig. 4)

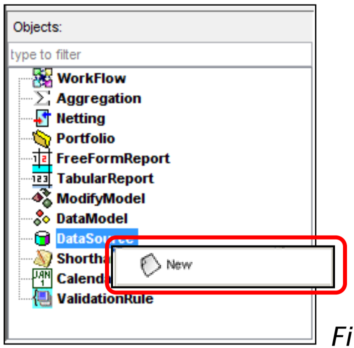


Fig. 4

If you are working with an active project, you can either start a new source or work with one of the existing sources.

4.01 Source Editor

Source Editor Screen (Fig. 5), the central facility for introducing and maintaining sources, allows control of all the source’s properties. The setup needs to be done once at the initial implementation or during the introduction of a new source. In most institutions the most frequent modifications done to the original source are relatively minor and involve changes to the attributes of columns such as column names.
The Source Editor consists of tabs that allow users to define various parameters of a data source.

4.01.01 General Tab

The following parameters are included in the General Tab:

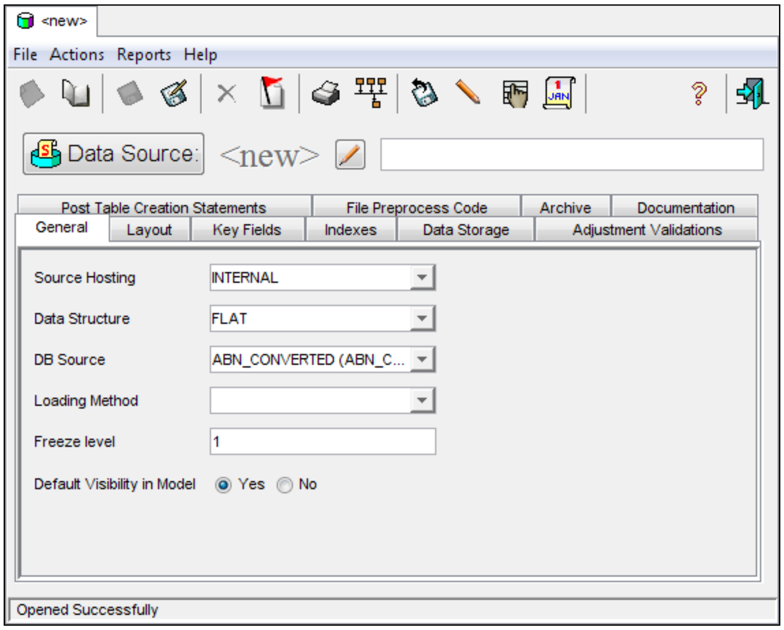
Source Hosting
The two options available from the drop-down menu are:

“Internal” means that users will be able to load, make manual adjustments, and that this source is maintained by the system in case you want to make changes to the setup of the data source.

“**External**” means that users can’t load and make manual adjustments and that this data source (table) is maintained outside of the system. External should be used for a table to which users have access to and would like to register an instance as of a certain date into the system.

Data Structure
Sources that users introduce to the system should be left to the default option which is **“Flat”.** This parameter specifies the composition of the incoming data and FLAT (default) indicates unstructured data.

New data sources should always be labeled as “Flat” since the AGGREGATION and AGGREGATION_WITH_SUBTOTALS are reserved for the system-generated sources.



Source Hosting: INTERNAL

Data Structure: FLAT

DB Source: ABN_CONVERTED (ABN_C...

Loading Method:

Freeze level: 1

Default Visibility in Model: ● Yes ○ No

Opened Successfully

Fig. 5

Note: Two other available selections: Aggregation and Aggregation_with_Subtotals are used and inserted by the system at the Save of the Aggregation Setup.

DB Source

DB source must be specified for each schema or database to which you would like to be able to store the tables in. This is the schema within the database into which you want to load the data into. You may also select a different DB Source for different parts of your workflow this way the results will be stored in various schemas you may have in your various databases.

Loading Method

This parameter is relevant only for the INTERNAL data sources. The following options are available:

1. **Create Instance** – will create a blank instance of the data source every time a workflow is executed which contains this data source as an object. You man also manually create a blank instance in the Data Storage tab.

2. **Load File** – should be used when loading any type of text or excel files.

3. **Load Table** – should be used to load data from a database table.

4. **Load with Journals –** enforces an additional set of rules to be applied to the data source. This loader will allow multiple users to load their own Excel files into the same data source instance. Once the data has been loaded into the system it will go into “Parked” status and it will be up to another designated user(s) to either accept or reject the records which are in the parked status.

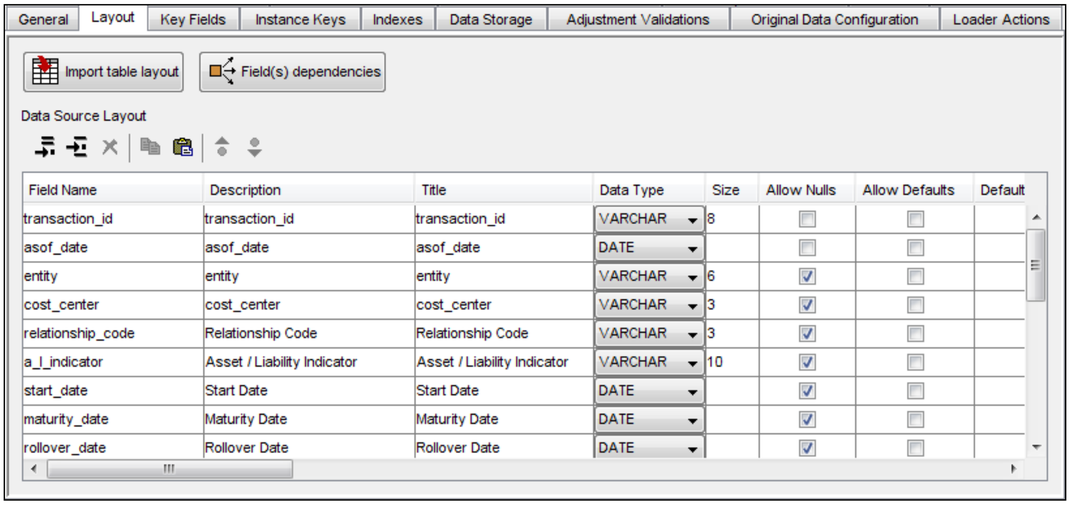
Default Visibility in Model

Check ‘Yes’ if you want to see this source on the Data viewer screen. As a rule the default visibility is required for the Main Source.

Control sources are usually defaulted to be invisible. This can be changed however on the "Header Manager" in data viewing screens.

4.01.02 Source Layout Tab

In order for the system to read and manage data from the incoming data source, the columns that constitute this source need to be identified. It is done by creating a Source Layout where for each field of the source, a corresponding column is entered and its attributes – a unique column name, column description, column title, data type, etc. specified.



Data Source Layout

Field Name	Description	Title	Data Type	Size	Allow Nulls	Allow Defaults	Default
transaction_id	transaction_id	transaction_id	VARCHAR	8	□	□	
asof_date	asof_date	asof_date	DATE		□	□	
entity	entity	entity	VARCHAR	6	☑	□	
cost_center	cost_center	cost_center	VARCHAR	3	☑	□	
relationship_code	Relationship Code	Relationship Code	VARCHAR	3	☑	□	
a_l_indicator	Asset / Liability Indicator	Asset / Liability Indicator	VARCHAR	10	☑	□	
start_date	Start Date	Start Date	DATE		☑	□	
maturity_date	Maturity Date	Maturity Date	DATE		☑	□	
rollover_date	Rollover Date	Rollover Date	DATE		☑	□	

Fig. 6

Before you start Source Layout setup, you must obtain data specs from the provider (i.e. how many columns there are in the source and their order, columns' names and sizes, method of delineation, etc). It is important to note that there are 3 possible names which the system allows for each column.

1. **Field Name** which should contain the most logical and clear name of the field.

2. **Description** – could contain a longer, more detailed name and explanation of the field's purpose.

3. **Title –** might contain a more technical or shortened name.

The columns must be named without spaces and special characters (underscores are allowed) and must be in lowercase.

In the Layout Tab (Fig. 6) at a minimum the Column Name, Data Type and Column Size must be set up for each field of the source.

Import table layout **Import Table Layout**

Import Table Layout button is used to populate the new source layout information with the existing setup.

Field(s) dependencies **Field Dependencies**

Two options are available:

"Field Dependencies" option shows objects which refer to the selected field.

"Dependencies of all fields" option shows objects, which refer to any of the fields in the current source.

Data Types

The following data types can be selected for the fields of the data source:

1. VARCHAR -
2. INTEGER-
3. FLOAT-
4. DATE-
5. TEXT-
6. UNICODE-
7. UNICODE_TEXT-

Allow Nulls

A space or empty string is considered a value except in Oracle where empty strings are not allowed. Checking the Allow Nulls box instructs the system to accept the data even if it is missing.

Put a check mark there if you are content with the fact that this field is empty (it does not contain a blank or a space or any other value).

Allow Defaults

If you anticipate having missing data you may allow for default values to be inserted.

Click the allow defaults button and you see two single quotes appear in the default value field. Between these single quotes you may enter any value that is compatible to the data type value that you want the system to use as default in case of the missing value.

If you have checked Allow Nulls, the Null becomes default.

If you enter a Default Value with Allow Nulls checked, it overrides the Allow Nulls setting.

Default Value

Into this input field you can enter any value you would like to appear in the column by default.

Note: DO NOT allow Nulls and Defaults if you do not want to load data with missing values.

Note: Default values can be used in conditions for data selection.

Note: Nulls can be only used in the Freehand Expression condition.

Synthetic Content

Synthetic Content field may be used in cases where you want to create a new field and populate this field with values resulting from calculation following a formula, which you input here.

Synthetic content fields do not physically exist in the database but only appear on the viewer screens.

The limitation of synthetic fields is that since they do not physically exist in the database they cannot be used in business rules mapping.

Key Fields

Each data source must have a unique identifier for each row. This is known as the Index of Key Field of a data source. The Key Field may be one or a combination of fields that can serve as a unique identifier for each record in the source. Any column containing data unique for every transaction (e.g. transaction ID) can serve as Key field.

If you are the owner of the data, you may create a new column as part of the Layout in the Layout tab and select that field as the “Loader Enumeration field”

Lookup Source

It is possible to attach lookup sources to the Source columns to provide an alternative for manual data entry. The lookup also acts as a validation tool making it impossible to input non-existing values.

Parent Columns

This is an advanced feature of the system and is not used very often. This functionality can give you a more optimized way of retrieving values from the lookup source. It does this by showing you only the reference values, which are associated with the parent column specified in this field.

Display Format

Display Format allows you to choose between various conventions of dates and numbers display. This feature is active only for the columns that contain numeric data and dates. The arrow button embedded in the field opens the menu listing formats supported by the system.

Setting up the Source Layout

There are two ways of creating a layout for your Data Source:

1. Import the layout from an existing source or external table and modify it if necessary. This method should be used in case of massive transactional sources that may have hundreds of columns.

2. Manually insert column names following the guideline from the data provider and filling in necessary parameters. This method is usually used when creating small in-house control data sources that often have only few columns.

Importing Layout from Existing Source

The easiest way to copy the layout from the existing source is as following:

1. Open the Layout Tab of the Original Source.

2. Copy all the rows and click ⎘ (copy rows) to transfer selected rows to the clipboard

3. Open the new source’s layout tab.

4. Click (Paste rows) to paste rows from the clipboard. The system transfers the entire source's layout, including all column settings, into the new source setup.

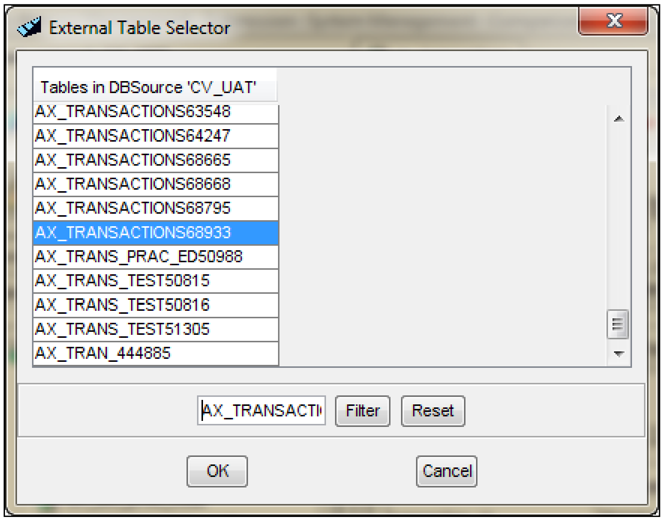
Importing Layout from the DB Source Table

If copy and paste is not possible, you can import layout table from the Original Source, where layouts for existing sources are stored as external tables.

Note: You can import several layouts into one. The system will remove redundant columns.

To import a Layout:

1. Import table layout button provides access to the lookup listing all external tables residing on the DB Source including existing sources' layouts. For our purposes we would like to import the layout for the source table called AX_TRANSACTIONS68933 (Fig. 7).



External Table Selector

Tables in DBSource 'CV_UAT'

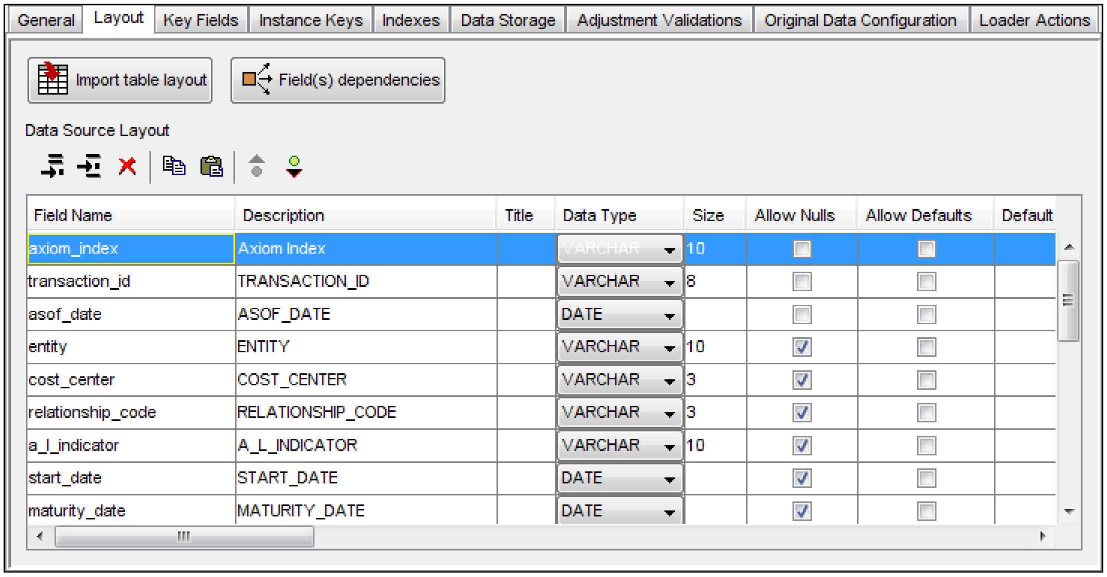
- AX_TRANSACTIONS63548
- AX_TRANSACTIONS64247
- AX_TRANSACTIONS68665
- AX_TRANSACTIONS68668
- AX_TRANSACTIONS68795
- AX_TRANSACTIONS68933
- AX_TRANS_PRAC_ED50988
- AX_TRANS_TEST50815
- AX_TRANS_TEST50816
- AX_TRANS_TEST51305
- AX_TRAN_444885

AX_TRANSACTI Filter Reset

OK Cancel

Fig. 7

2. Locate the table you need and press OK to close the lookup. The imported layout appears in the Layout tab's data grid (Fig. 8).



General | Layout | Key Fields | Instance Keys | Indexes | Data Storage | Adjustment Validations | Original Data Configuration | Loader Actions

Import table layout Field(s) dependencies

Data Source Layout

Field Name	Description	Title	Data Type	Size	Allow Nulls	Allow Defaults	Default
axiom_index	Axiom Index		VARCHAR	10	☐	☐	
transaction_id	TRANSACTION_ID		VARCHAR	8	☐	☐	
asof_date	ASOF_DATE		DATE		☐	☐	
entity	ENTITY		VARCHAR	10	☑	☐	
cost_center	COST_CENTER		VARCHAR	3	☑	☐	
relationship_code	RELATIONSHIP_CODE		VARCHAR	3	☑	☐	
a_l_indicator	A_L_INDICATOR		VARCHAR	10	☑	☐	
start_date	START_DATE		DATE		☑	☐	
maturity_date	MATURITY_DATE		DATE		☑	☐	

Fig. 8

Note: When importing an existing layout, only the basic setup will be copied. You will have to manually add Lookup sources and enter other parameters.

Note: Axiom Index field (Fig. 8) has been added to the source data to be used as a key.

Manual Layout Setup

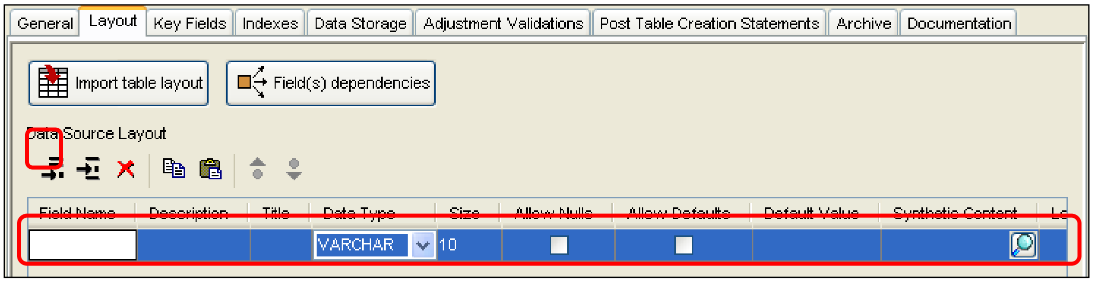
There are times when you are forced to create the source layout manually. Should this be the case, the following preliminary steps must be taken:

1. Obtain the list and order of columns in the actual source table.

2. Create columns for the source data to go to. It is important to notice, that it is **not** necessary to create the entire replica of the original source file. In fact you can only set up as many columns as you know you are going to need for your further work.

Note: In this case however, while the data upload will result in populating these columns, the remaining data will be lost.

To start setting up the Source layout click (Insert row) button. This will add a blank row in the data grid of the Layout (Fig. 9).



VARCHAR 10

Fig. 9

Note: The sequence of the columns defined in the Layout should match the order of the columns in the original source file.

Field Name – The columns must be named without spaces and special characters (underscores are allowed) and must be in lowercase. This will be displayed as the column heading if you leave the Description and Title fields empty.

Description – Uppercase and spaces can be used. This will be displayed as the heading if Title is empty.

Data Type – VARCHAR – Can be a combination of numbers, letters and symbols.
– INTEGER – Contains numbers without decimals.
– FLOAT – Calculated amount.
– DATE – Represents date.

Size – Number of characters the entry would not exceed.

Allow Nulls – Allows empty values to be used.

Note: Manual Layout setup can be used when adding fields to the source (e.g. axiom_index).

4.01.03 Data Storage Tab

On the Data Storage Tab (Fig 10) you specify the storage method for the incoming source data. Based on the data type you are working with you must specify how this data will be accumulated and disposed of.

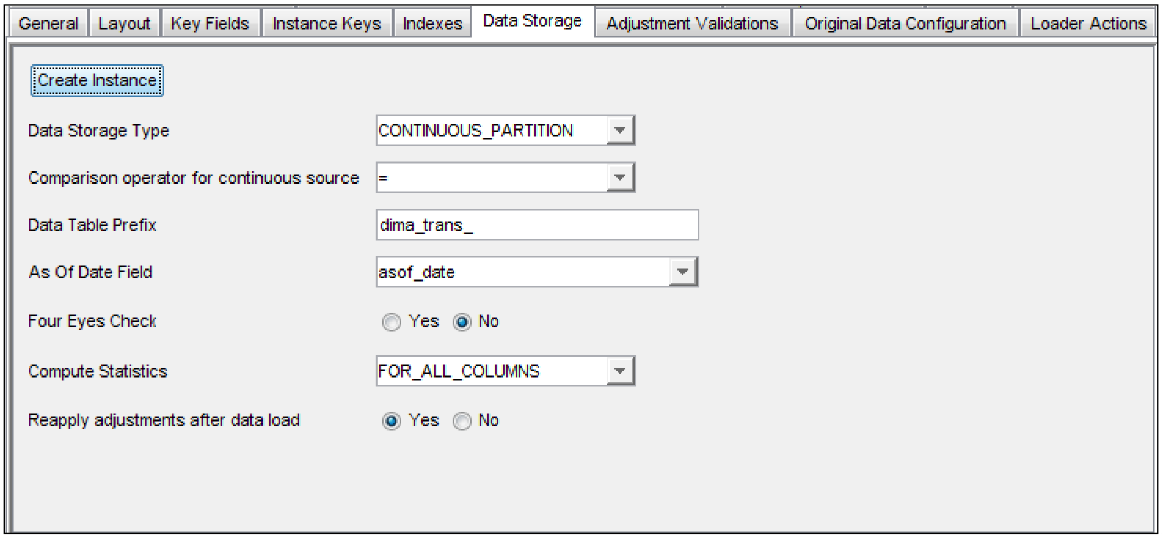
The method of data accumulation and disposal is determined by the Source’s data type that can be Segmented, Continuous or Permanent. Specifying one of these parameters will determine what routines the system will apply to the source data during the pre/post processing and the actual download.

Clicking Create Instance **button**

This feature is used primarily when creating the Permanent data (control sources) manually. Pressing this button generates an empty table, following the source layout structure.

Data Accumulation Mode

The type of data in the source determines the method for data accumulation. The options are listed in the Data Accumulation Mode menu that opens when you press the arrow button next to the input field.



Data Storage Type: CONTINUOUS_PARTITION

Comparison operator for continuous source: =

Data Table Prefix: dima_trans_

As Of Date Field: asof_date

Four Eyes Check: ○ Yes ● No

Compute Statistics: FOR_ALL_COLUMNS

Reapply adjustments after data load: ● Yes ○ No

Fig. 10

In the current version of the system it is not advisable to change Data Accumulation Mode once it has been established and tables created. It is difficult to retrieve data if the source is originally set up as *segmented* and at a later point switched to *continuous* or vice versa. The data will not be lost but it will require some database management work to retrieve it.

- **Permanent** – This type of data usually refers to control data (reference data, legacy data). Control data does not change very often and does not contain a timestamp.

- **Segmented** – Each data set is identified by the data source name and the date of the “snap shot” and stored in a separate table. Usually this type of data is transactional data.

- **Continuous** – Each data set is accumulated in a single table along with a timestamp. This type of data usually refers to market data.

- **Continuous Partitioned** – This is a variation of a Continuous Data storage type where each data load is partitioned separately within the same table. This method is the preferred method of storage over the segmented storage type due to the fact that is will be easier to work with the data through a time dimension such as to include multiple sets of data for the calculation of average's.

Data Accumulation Mode selection determines what parameters will need to be defined next.

Permanent

If the Permanent Accumulation Mode is selected the combination of the following already described parameters needs to be defined:

Data Table Prefix - The input here is done manually using low case and underscores. The system will use the prefix (e.g. *company_name_forex_*) when generating the name of the actual table created in the course of the data upload.

The prefix will be followed by the "as of date" of the source upload. This makes it easy to find all the tables in the database for a particular source because they will all have the same prefix only the time stamp changes.

Four Eyes Check Yes/No switch – Check YES if you want alerts to be sent to the designated individuals every time adjustments to the source data are made.

Compute Statistics – This parameter refers to the statistical map of the data table structure created by the system in order to expedite the execution of the database queries.

The following options are available:

a. NONE – no DB statistics will be calculated

b. FOR ALL CLOLUMNS – DB statistics will be calculated for every field in the Data Table.

c. FOR KEY COLUMNS – DB statistics will be calculated for the Key fields only

Reapply Adjustments after Data Load Yes/No switch – Check 'Yes' if the adjustments you have made to the data should recur with every download.

Segmented

If the Segmented Data Accumulation Mode is selected the following parameters need to be defined:

Data Table Prefix – The input here is done manually using low case and underscores. The system will use the prefix (e.g. *company_name_forex_*) when generating the name of the actual table created in the course of the data upload. The prefix will be followed by the "as of date" of the source upload. This makes it easy to find all the tables in the database for a particular source because they will all have the same prefix only the time stamp changes.

Four Eyes Check Yes/No switch – Check YES if you want alerts to be sent to designated individuals every time adjustments to the source data are made.

Compute Statistics – This parameter refers to the statistical map of the data table structure created by the system in order to expedite the execution of the database queries.

a. NONE – no DB statistics will be calculated

b. FOR ALL CLOLUMNS – DB statistics will be calculated for every field in the Data Table.

c. FOR KEY COLUMNS – DB statistics will be calculated for the Key fields only

Reapply Adjustments after Data Load Yes/No switch – Check 'Yes' if the adjustments you have made to the data should recur with every download.

Continuous/Continuous with Partitions

If Continuous/Continuous with Partitions Data Accumulation Mode is selected, the following parameters need to be defined:

Comparison Operator for continuous source – Should be set on '='

Data Table Prefix – The input here is done manually using low case and underscores. The system will use the prefix (e.g. *company_name_forex_*) when generating the name of the actual table created in the course of the data upload. The prefix will be followed by the "as of date" of the source upload.

As Of Date Field – Here you indicate which field you want the system to refer to when the As of Date parameter is required. The system will put into this field a time and date stamp for each record at the time of loading the data.

Four Eyes Check Yes/No switch – Check YES if you want alerts to be sent to designated individuals every time adjustments to the source data are made.

Compute Statistics – This parameter refers to the statistical map of the data table structure created by the system in order to expedite the execution of the database queries.
The following options are available:

a. NONE – no DB statistics will be calculated
b. FOR ALL CLOLUMNS – DB statistics will be calculated for every field in the Data Table.
c. FOR KEY COLUMNS – DB statistics will be calculated for the Key fields only

Reapply Adjustments after Data Load Yes/No switch – Check 'Yes' if the adjustments you have made to the data should recur with every download.

4.01.04 Key Fields Tab

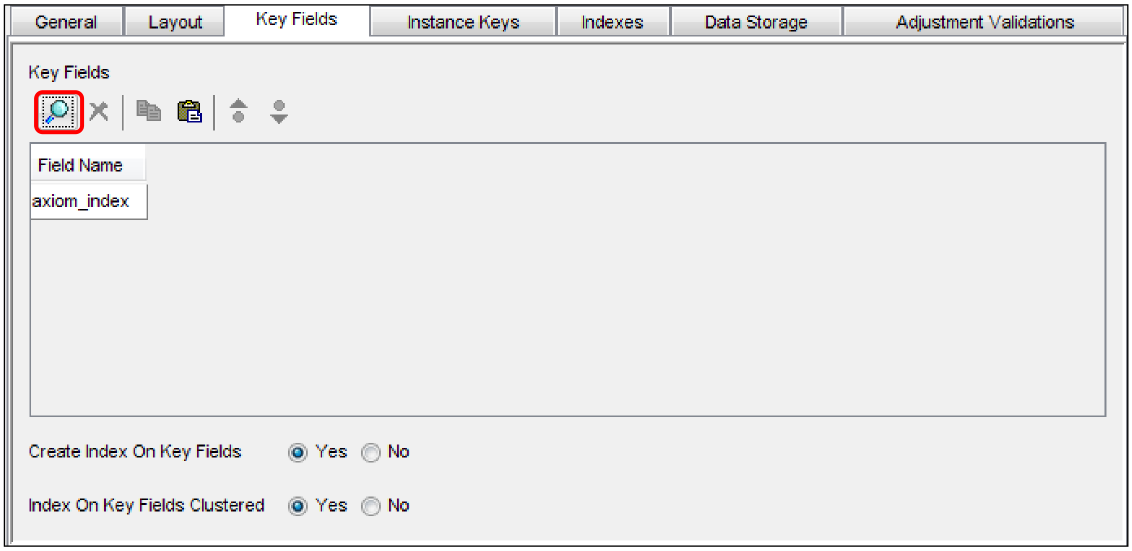
A key field is a column that can be used to find a particular entry. This column must contain the information that is unique for every record in the data table. If no such column exists in the source data, the combination of several fields can be used. Examples of key fields might be Customer Number, Social Security Codes, Transaction ID, etc.

If the client does not provide a key field, we manually insert "axiom_ index" column into the Source layout and set it as a key field.

For this example, we will use “axiom_index” as our key field. To set a Key Field, open “Key Fields” Tab:

1. Click 🔍 (Look up) and select “axiom_index” from the lookup

2. Check Yes on the “Create Index On Key Fields” and on “Index On Key Fields Clustered”.



Key Fields

Field Name
axiom_index

Create Index On Key Fields ● Yes ○ No

Index On Key Fields Clustered ● Yes ○ No

Fig. 11

Figure 11 shows what the Key Fields tab should look like when the Key field setup is complete.

4.01.05 Adjustment Validation Tab

In this tab you can establish safeguards against improper changes to data on the Adjustments screen. For example, you may not want to allow some columns to contain “0” value or you may not want values in some columns surpass certain limit. Use the Lookup button to select the column you want to protect. Then from another Lookup select a macro containing an expression to place in the right hand column of the tab’s grid.

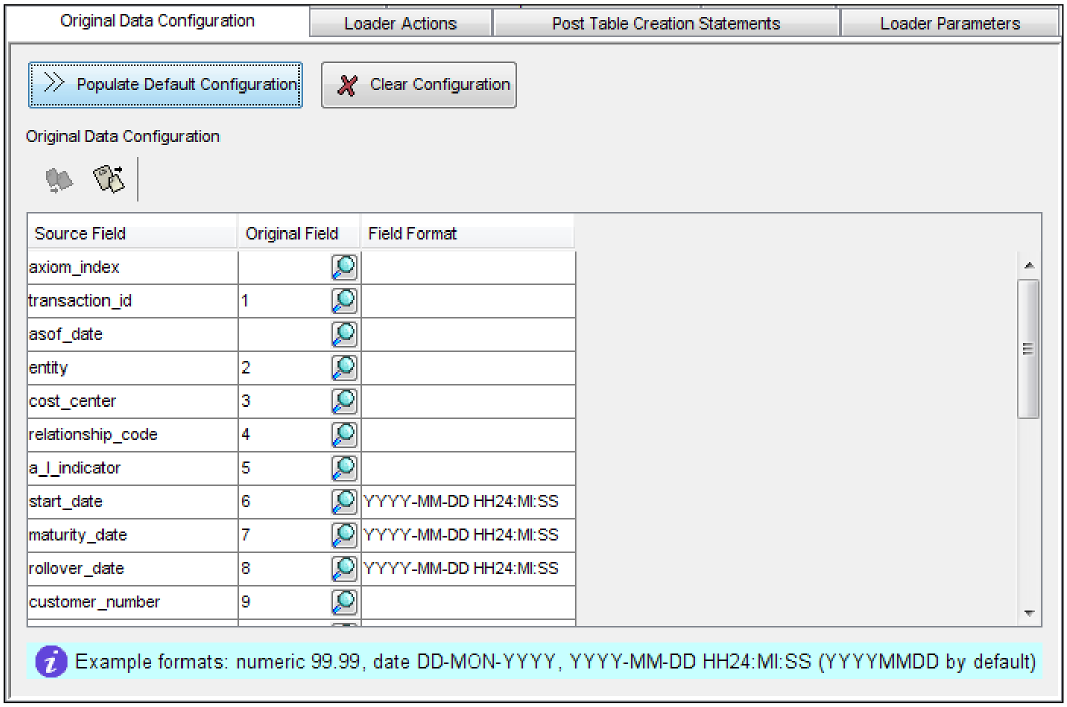
4.01.06 Original Data Configuration Tab

In order for the system to identify columns in the source they have to be mapped to the corresponding columns of the Source Layout. This is done on the Original Data Configuration tab (Fig. 12).

Left column (Source Field) of the tab’s grid is filled with the column names as they were defined during the Source Layout setup.

The right column of the grid is populated with the names of the corresponding columns of the actual source table or, if the original source is a file with columns delimited by tabs or other characters, by a number in order of which the column appear. In case of a brand new source, the right column comes up empty and needs to be filled manually.

If the names of columns are the same in the layout and the original table, the » Populate Default Configuration button above the grid copies a highlighted name on the left into the right hand column.


Original Data Configuration
Loader Actions
Post Table Creation Statements
Loader Parameters
Populate Default Configuration
Clear Configuration
Original Data Configuration

Source Field	Original Field	Field Format
axiom_index		
transaction_id	1	
asof_date		
entity	2	
cost_center	3	
relationship_code	4	
a_l_indicator	5	
start_date	6	YYYY-MM-DD HH24:MI:SS
maturity_date	7	YYYY-MM-DD HH24:MI:SS
rollover_date	8	YYYY-MM-DD HH24:MI:SS
customer_number	9	

Example formats: numeric 99.99, date DD-MON-YYYY, YYYY-MM-DD HH24:MI:SS (YYYYMMDD by default)

Fig. 12

If the names of the columns in the layout and the original table differ, use 🔍 buttons to select appropriate column names.

Additional Formatting

If you want certain formatting to be applied to the column you can insert a database-specific “Field format condition” which will be added to the loader program. The expression below, for example, allows the system operating in Oracle environment to accept numbers with decimal commas:

```
"TO_NUMBER(:gross_amount,'9999D9999','NLS_NUMERIC_CHARACTERS = '',.''')"
```

Note: If you are not clear on how to do this, consult Axiom SL.

4.01.07 Loader Parameters Tab

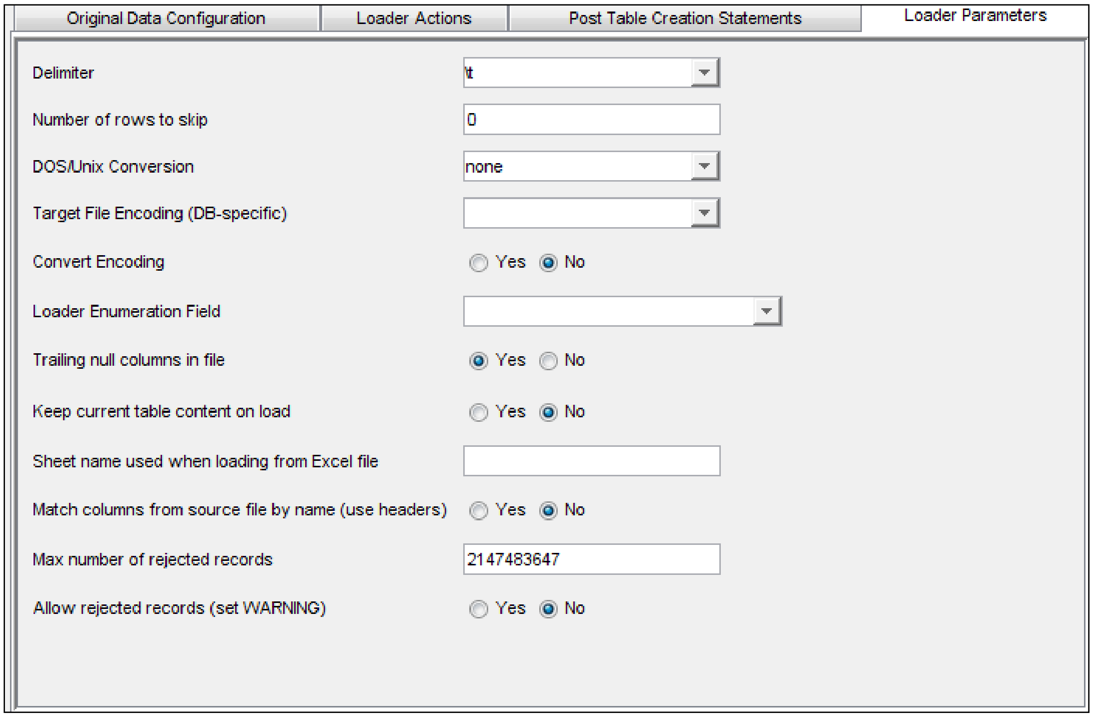
The composition of Loader Parameters tab depends on the Loading Method selection. On the illustration below (Fig. 13) the Loader Parameters tab reflects “Load File” loading method.

The following parameters need to be set up for the Load File loading method:

Delimiter – select from the menu a character to separate columns in the original source file. The following options are available:

1. TAB – Currently shown
2. COMMA
3. SEMICOLON
4. PIPE

5. COLON



Delimiter	\t
Number of rows to skip	0
DOS/Unix Conversion	none
Target File Encoding (DB-specific)	
Convert Encoding	○ Yes ● No
Loader Enumeration Field	
Trailing null columns in file	● Yes ○ No
Keep current table content on load	○ Yes ● No
Sheet name used when loading from Excel file	
Match columns from source file by name (use headers)	○ Yes ● No
Max number of rejected records	2147483647
Allow rejected records (set WARNING)	○ Yes ● No

Fig. 13

Number of rows to skip – enter number of lines taken by column headers in the original source file that should be omitted during the download to preserve the table's integrity.

DOS/UNIX Conversion – Convert from DOS to UNIX or UNIX to DOS

Target File Encoding (DB-specific) – Consult AXIOM SL.

Convert Encoding – Parameter used in the case where a file comes from a mainframe and needs to be converted to ASCII.

Loader Enumeration Field – Represents the Key Field when it is not located in the original source. The "axiom_index" for example, is the key field that we created needs to be selected in our case.

Trailing null columns in File – When the control file specifies more fields for a record than are present in the record, SQL*Loader must determine whether the remaining (specified) columns should be considered null, or whether an ERROR should be thrown.

TRAILING NULLCOLS clause tells the SQL*Loader to treat any relatively positioned columns that are not present in the record as null columns. Check 'No' if all the columns are populated.

Keep Current Table Content on Load – The system will create a temporary table and will compare all of the previously loaded records to the new records. This comparison is done using the unique key in the data source table.

If the system finds matching unique keys in both previously loaded data and the new data, it will keep the newly loaded versions of those records and discard the old. The records, which do not exist in the previously loaded table, will be inserted as well.

Sheet name used when loading from Excel file – Excel files can have multiple worksheets. When uploading an excel file, be sure to specify which worksheet you want to use.

Match columns from source file by name (use headers) – 'No'

Max Number of rejected records – enter the number of records, the loss of which you are willing to tolerate when loading the data.

Allow rejected records (set WARNING) – Checking 'Yes' will force the system to continue loading data despite rejected records. The WARNING message will be issued however.

Data Loading Method – If the "Load Table" loading method is selected on the General tab then this option becomes available in the Loader Parameters tab.

1. "Direct" is when you are loading from one database into the same kind of database i.e. from one DB2 database to another db2 database.

2. "In Memory" is when you are loading from one database such as Oracle into a completely different database such as Sybase.

3. "File" takes the data from a table then puts it into a file and then into another table.

Note: These parameters should be agreed upon with the data source provider in advance.

4.01.08 Loader Control/Format File Tab

This tab allows to control how and what data to load by using SQL Loader scripts. The control file's data type specifications tell SQL*Loader how to interpret the fields in the data files. The SQL*Loader uses this information when working with the fields, to describe the data that is being passed to ORACLE. ORACLE then converts the data into the data type specified by the table definition.

4.01.09 Loader Actions Tab

Loader Action is an executable program for data post processing. Loader Actions tab allows You to set up the parameters specific for each selected Loader Action.

ActionValidateStart/ActionValidateEnd

These two actions are used in tandem and do not require any additional parameters. The purpose of these actions is to allow running all other actions assigned to the current loader to execute and produce the error report prior to aborting.

ActionDateAdjust

Purpose: Updates ColumnName with the date equal to the value of the column plus offset in business days, passed via AdditionalValue parameter.

ActionIntegrity

Purpose: Validation of the column values against control table, associated with this column (see source layout). Depending on the AdditionalValue parameter the action can have several outcomes.

ActionStatement

Purpose: Executes SQL statement passed via AdditionalValue parameter.

ActionValidate

This action is used to validate records in the data source against a condition.

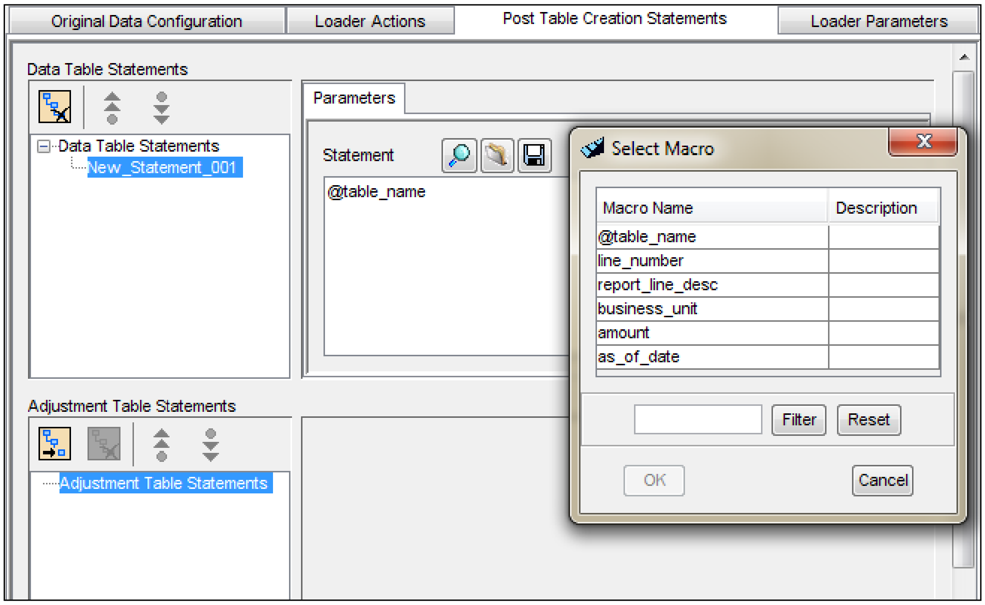
ActionValidateStatement

This action is used to validate records being loaded against an SQL statement. Similar in its function to The above-described ActionValidate, ActionValidateStatement allow using complex SQL expressions.

4.01.10 Post Table Creation Statements Tab

The system allows you to control the format and content of the tables generated in the system. The table control is performed via SQL statements that had been saved on the DB as 'macros'.

These 'macros' are available via lookup and can be inserted into the tab's Parameters pane (Fig. 14).



Statement: @table_name

Macro Name	Description
@table_name	
line_number	
report_line_desc	
business_unit	
amount	
as_of_date	

Fig.14

Post Table Creation Statements are SQL statements needed to be executed after the data table is created and before the actual loading of the data. These statements are used to create special indexes, triggers etc.

SEGMENTED and PERMANENT data storage types – A data table is generated each time the Source Instance is created.

CONTINUOUS/CONTINUOUS_ PARTITION data storage types – a table is generated only when the first source instance is created.

In the latter case, the data table will be recreated only after the source layout change or after last instance is removed and the first instance is created again.

Adjustment Table Statements are SQL statements executed each time the adjustment table is created.

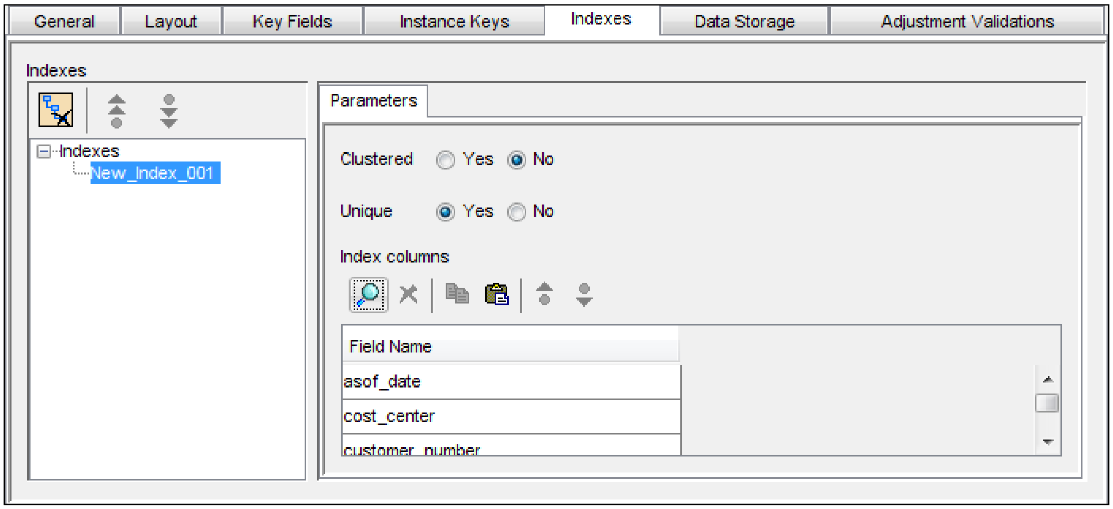
The system generates Adjustment table when the "Adjustments" option is selected from the menu that opens when you right click on the source object in the Project's Object Pane.

Note: Adjustment Tables are recreated with the source layout change.

4.01.11 Indices Tab

On the Indexes Tab (Fig. 15) additional indexing can be established in order to optimize and enhance the system's performance.

For each additional index you can enter any combination of columns that are often used in referencing and drill down.



Clustered ○ Yes ● No

Unique ● Yes ○ No

Field Name
asof_date
cost_center
customer_number

Fig. 15

Each additional index is represented by a node in the Indexes Pane. Index nodes will have unique names that define its purpose.

Clustered ○ Yes ● No

This parameter allows you to determine where the Index will be stored. It is Database-specific and works differently with various DB types. Please consult your DBA prior to using this feature.

Unique ● Yes ○ No

If Yes is checked, the Index will be regarded as Unique.

4.01.12 Archive Tab

On the Archive tab you can set up:

1. Time periods for which the source data will be preserved on the database and remain accessible prior to being archived
2. Time period the data will remain in the archived state prior to being deleted entirely.

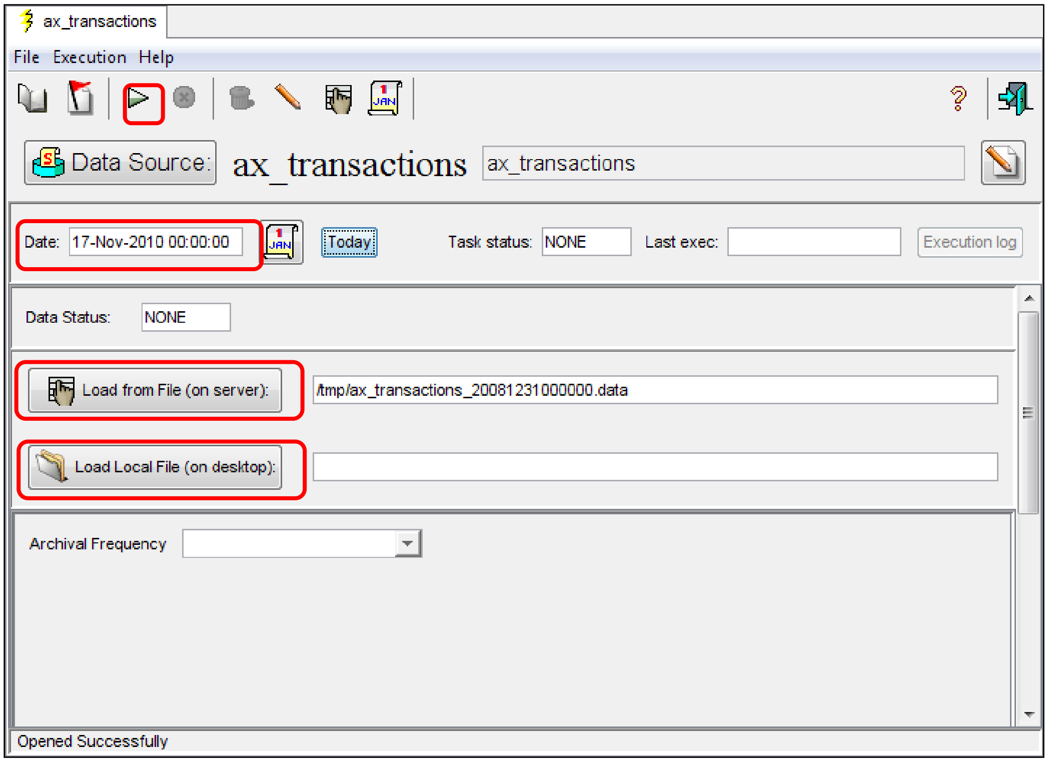
The available Archiving options are based on the archiving defaults that had been determined at the installation of the system.

4.01.13 File Preprocess Code Tab

Preprocess code is a script in AxiomSL language that is stored in data source setup. It is used to convert a source file into something different prior to load.

4.02 Data Source Loader

Data Source Loader interface (Fig. 16) is used to manually load data into a data source table thus making it immediately available in the manual adjustments screen for either adjustments or simply viewing the data. The manual adjustments facility is based on individual data sources as opposed to a data model, which is a collection of sources joined together.



Data Source: ax_transactions

Date: 17-Nov-2010 00:00:00

Task status: NONE

Last exec:

Data Status: NONE

Load from File (on server): /tmp/ax_transactions_20081231000000.data

Load Local File (on desktop):

Archival Frequency

Opened Successfully

Fig. 16

Data Loader can be accessed:

1. **From the Source Editor.** Click the button (Data Load).

2. **From the Project Management screen.** On the Object pane right click on the source object and select Data Load option from the menu.

Date – Should be set to the date of when the report applies. In this case we are loading data to be used for the Quarter 3 report of 2010, the Date should be 30-Sep-2010.

Task Status – Displays the progress the system is making when loading data.

Last exec – This is the date of when the data source was last loaded.

Data Status – Displays the status of the data after it has finished loading. “Active” means the data has been correctly loaded.

4.02.01 Introducing Control Data Sources

If you are introducing a Control (Reference data) source for the first time to the system it is necessary, while working on the source setup, to create an empty table for the new source by pressing the Create Instance button on the Data Storage tab of the Source Editor screen.

Then you may enter the Manual Adjustments screen, open the newly created table and start populating the columns of the new reference data source.

If you have source(s) of data which already exist in a database table(s) you still have to set it up so that the system knows about it, but instead of “pressing the start data loading” button to load all you have to do is press the Register Source Instance button instead.

4.03 Data Loading

If you are loading data for a Segmented or Continuous/Continuous with Partition data source the following steps should be followed:

1. Select the source, which you would like to load data into.

2. Select the As Of Date. This date should represent the time period, for which the data is actually delivered. The current date is shown by default.

Note: Check your company’s dating convention to make the appropriate choice

3. Determine the location of the data file and use the appropriate button to select it.

4. Press the ▷ START button.

To Load the original source into the ControllerView Software, click on the “Load Local File (on Desktop)” button. Select the file you wish to load and then click the button.

Note: After loading, the client can make changes to the data on the Data Adjustments screen.

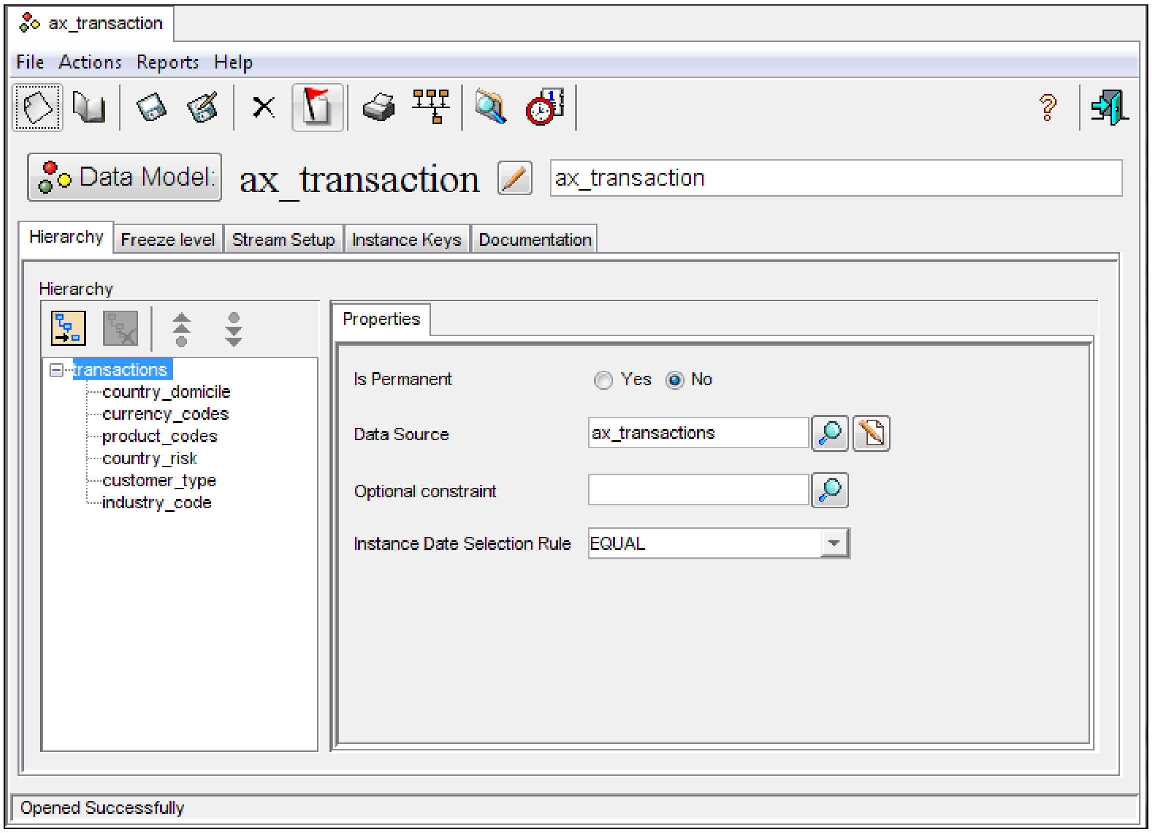
5. DATA MODEL

A data model typically consists of many data sources, which have a relationship between them. This relationship allows linking the sources into comprehensive entities making for a richer data set.
Note: One-source data models can also be used on specific occasions.

Data Models are based on a Segmented or Continuous data source and its extensions (control sources). There is always one Main source of Transactional (deal) data in the Data Model.

There might be several related transactional data sources included in a Data Model, each having its own set of extensions. However, depending on your preference, one will be dominant over others. For example, you may have Futures and Settlements sources in the same Data Model. For every Future transaction there is settlement information. If you mainly work with the Futures, you would make it a Main Source and Settlements – secondary enabling it for viewing only when the need arises.

For each segmented/continuous source in the Data Model you will have a set of control sources tables. Control sources (tables) provide descriptions for certain fields in the main source table. Control sources also provide lookups for the values in these fields. Each transaction source can have any number of control sources.



Data Model: ax_transaction ax_transaction

transactions
- country_domicile
- currency_codes
- product_codes
- country_risk
- customer_type
- industry_code

Is Permanent ○ Yes ● No

Data Source ax_transactions

Optional constraint

Instance Date Selection Rule EQUAL

Opened Successfully

Fig. 17

There are two types of Data Models:

1. Data Models created by users
2. Data Models generated by the system.
   a. Resulting from the Portfolio Execution. A table, created in the course of the portfolio execution, is added to the original Data Model on which the Portfolio was based.

b. Resulting from the Aggregation Execution. A table of summaries is created by the system as the new Data Model's Main Source with control tables inherited from the parent Data Model. This table of summaries is related to the transactional source enabling the user to view transactions for every summary.

5.01 Data Model Editor

Data Models setup takes place on the Data Model Editor (Fig. 17).

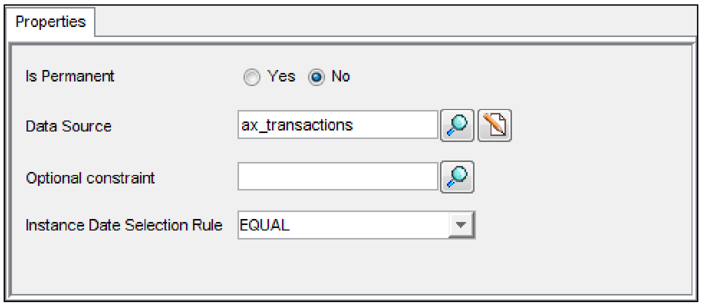
Each source in a Data Model is represented by a "node". There is always a Main node representing the Main source of segmented or continuous data and a group of extensions – auxiliary control sources and additional sources of the transactional data.

These nodes are organized into a "tree" structure. You can create as many levels as you want as soon as all sources are related. User defines how sources will interact by establishing hierarchies and defining types of relationships.

5.01.01 Node Properties

Properties for each node need to be defined for each node of the Model. Properties for the Root node differ from the properties of the auxiliary nodes.

The following information needs to be entered into the Properties input fields for the Root Node (Fig.18):



Is Permanent ○ Yes ● No

Data Source ax_transactions

Optional constraint

Instance Date Selection Rule EQUAL

Fig.18

1. **Data Source**. Contains the name of the actual data source represented by the Source Alias. The arrow button opens a menu listing all available data sources.

2. **Optional Constraint**. Into this field you enter a statement for data selection (i.e. currency='USD') and this condition will be applied every time this data model is used.

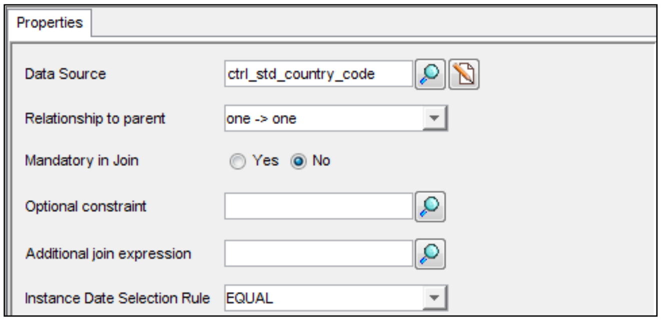
3. **Instance Date Selection Rule**. This parameter indicates to the system what method of data selection should be used when creating a Model Instance.

    a. **EQUAL** (default) – The system will search for data for **the same date as the Model Instance's that is being created.**

    b. **LATEST** – The system will pick the data from the **latest load prior to the Model Instance date**

    c. **Formula** – Allows you further refine Instance Data Selection by entering an expression.

The following parameters (Fig. 19) are entered for each Auxiliary Node:



Data Source	ctrl_std_country_code
Relationship to parent	one -> one
Mandatory in Join	○ Yes ● No
Optional constraint	
Additional join expression	
Instance Date Selection Rule	EQUAL

Fig. 19

1. **Data Source** – Name of the source represented by the current Node

2. **Relationship to Parent**. Input here indicates how many records in the source correspond to a single record in its parent and vice versa. The menu provides following selections (Fig. 20).

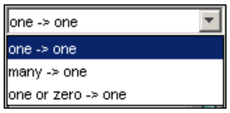


Fig. 20

3. **Mandatory in Join switch**. Determines whether the data in this source will be available for viewing on the Filter screen.

4. **Optional Constraint**. In case of children nodes, into this field, besides data selection criteria, you can enter additional join information.

5. **Additional Joint Expression**. Into this input field you can enter an SQL expression to stipulate additional joint requirements.

6. **Instance Date Selection Rule**. This parameter indicates to the system what method of data selection should be used when creating a Model Instance.

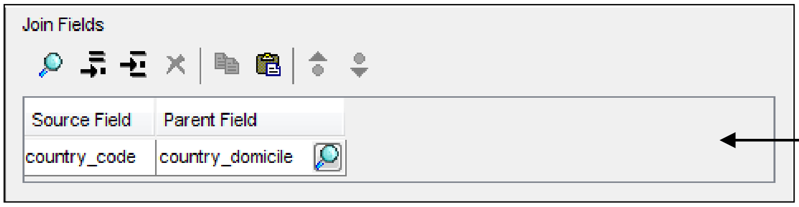
   a. **EQUAL** (default) – The system will search for data for **the same date as the Model Instance's that is being created.**

   b. **LATEST** – The system will pick the data from the **latest load prior to the Model Instance date**.

   c. **Formula** – Allows you further refine Instance Data Selection by entering an expression.

Join Fields Pane

On the Join Fields pane (Fig. 21) key fields that link the current source with its parent, are mapped. The pane's grid has two columns – Source Field and Parent Field.



Source Field	Parent Field
country_code	country_domicile

Join column selection

Fig. 21

The Source Field refers to the name of the column in the current source. The Parent Field is the name of the corresponding column in the Parent data source.

5.02 Data Model Setup

To create a new Data model, on the Project Management screen’s Object pane right click on “Data Model” category and then click “New” (Fig. 22).

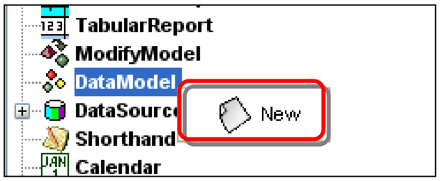
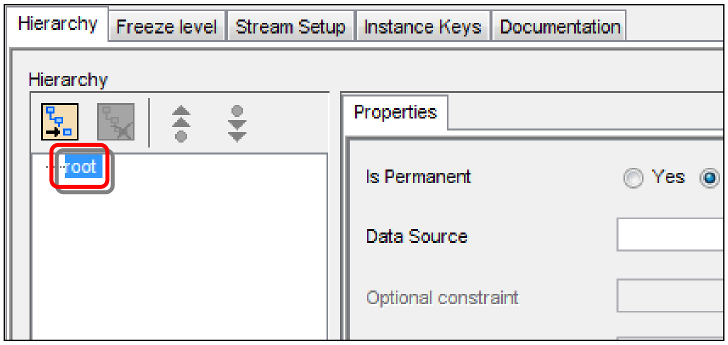


Fig. 22

A Data Model Editor will open with the Hierarchy tab shown by default (Fig. 23). This is where you will construct the Data Model by adding data sources.

The “*root*” node (Fig. 23) is the Main node. It is inserted by the system. To the *root* node you add nodes to represent other sources.



Is Permanent ○ Yes ●

Data Source

Optional constraint

Fig. 23

In most cases you will have one Main data source, and several reference data sources that are linked to it. This is useful when you need to combine two or more Data sources using the same column as the key.

For example, if you had two data sources, one with Transaction ID and Amount, and another with Transaction ID and Description, you would join them using the Transaction ID as the key. As the result, when accessing this data model on the data viewer, you would be able to see all three columns together.

5.02.01 Root Node Properties Setup

The following properties need to be set up for the Root node:

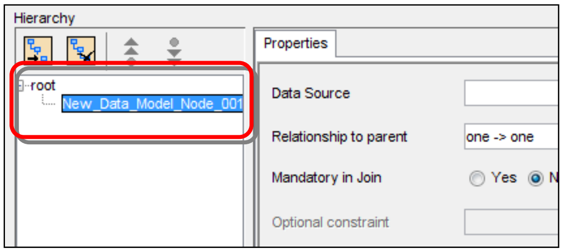
1. **Is Permanent** – should be set on **‘No’**
2. **Data Source –** use 🔍 button to specify which Data Source will serve as the Main Source.
3. **Optional Constraint –** specifies a condition to select specific data (i.e. currency=’USD’)
4. **Instance Date Selection Rule** – should be set on EQUAL

5.02.02 Building Data Model Hierarchy

Data Model is constructed by adding nodes to the root. To add a node:

1. Click on the Root node to select it

2. Click (Insert node) button. This will insert a node placing it under the selected node (Fig. 24).



Data Source

Relationship to parent: one -> one

Mandatory in Join: ○ Yes ● N

Optional constraint

Fig. 24

3. Rename the inserted nodes

    1) Double click on the Node’s name tag to switch it into the editable form

    2) Type in the name of the source the node is to represent

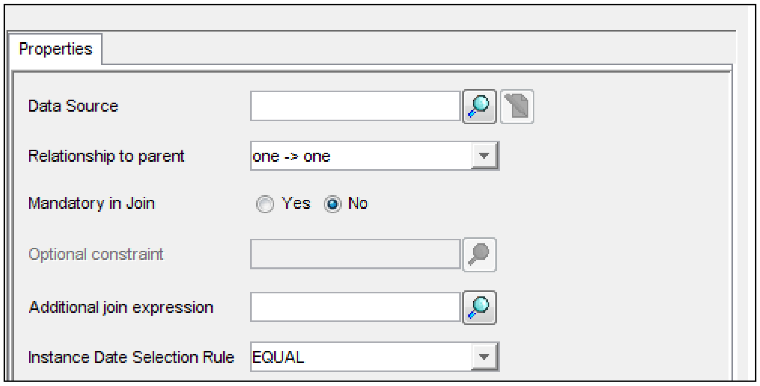
4. Repeat above steps to insert and name as many nodes as needed for your purpose.

5.02.03 New Node Properties Setup

The following properties need to be setup for each inserted node (Fig 25):

Data Source – use the button to select the source to be represented by the inserted node (Country Code in our example).

Relationship to parent – Input here indicates how many records in the source correspond to a single record in its parent and vice versa (one->one in our example).



Data Source

Relationship to parent: one -> one

Mandatory in Join: ○ Yes ● No

Optional constraint

Additional join expression

Instance Date Selection Rule: EQUAL

Fig. 25

Mandatory In Join – This parameter determines whether the data in this source will be available for viewing on the Filter screen (No in our example).

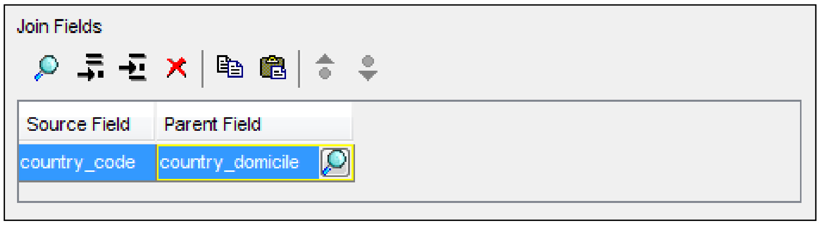
Additional join expression – Allows you to enter an SQL expression to create additional joint requirements (None in our example).

Instance Date Selection rule – The following options are available:

1. **EQUAL** (default) – The system will search for data for the **same date** as the Model Instance's that is being created.

2. **LATEST** – The system will pick the data for the **latest load** prior to the Model Instance date.

Join Fields
On the Join Fields pane select key fields that link the current source with its parent, are mapped. The pane's grid has two columns – Source Field and Parent Field. The Source Field refers to the name of the column in the current source. The Parent Column is the name of the corresponding column in the Parent source (Fig. 26).


Join Fields

Source Field	Parent Field
country_code	country_domicile

Fig. 26

5.02.04 Stream Setup

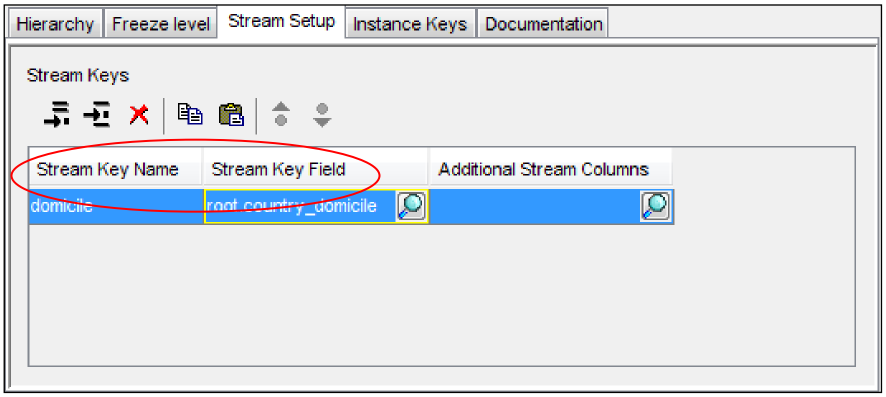
Stream is a subset of data based on the 'IN' condition. Streaming enables users to run executable tasks for selected set of values for one or several columns of the source.

For example a user can run a workflow first against the country code column for USA and Canada only and then for Holland and France. The combination of USA and Canada would then represent one stream, while Holland and France – another.

The following parameters should be set up for each stream (Fig. 27):

Stream Key Name is an alias of the stream column which you enter manually.

Stream Key Column term refers to the name of the actual source column to which the alias is mapped.



Stream Keys

Stream Key Name	Stream Key Field	Additional Stream Columns
domicile	root.country_domicile	

Fig. 27

The 🔍 Button opens the lookup listing source columns for your selection. Illustration above shows a sample stream column setup.

By entering a column name you set up the ‘IN’ condition for the selected values from this column.

This stream will appear on the Execution screen for the task where this data model is employed. There you will enter actual values prior to the execution of the stream.

The following points must be taken on account when working with Streams:

- If the data model is to participate in streaming, the user has to define stream columns – names, like country or currency, and map them to the actual source columns.

- Different data models can have different set of stream columns

- Portfolio resulting models inherit steams columns from the original source data models

- Aggregations inherit stream column names from input data models. If the stream column is not included in the Detail Set for an aggregation, you will not be able to drill down using this column as a parameter. In order to make the results of the Aggregation streaming visible you have to create a detail and map it to the stream columns of data models.

- During execution tasks with stream columns generate multiple tables with values for each stream column (country IN (‘USA’, ‘CANADA’), *counterparty_type* IN (“BANKS’, ‘CENTRAL BANKS’)

- If no value is defined for the stream column, no condition will be applied

- Stream execution creates a new data instance only if the instance did not exist before. Streams will write into the same data instance except for Free Form reports (see below).

- During the execution a stream will delete in the data instance all prior data for its columns (i.e. country IN (‘USA’, ‘CANADA’).

- All streams write into the same data instance. That means that adjustments can be done to all streams in one session.

- Every stream creates a new task instance. It means that on the Global Status screen they will appear as separate entities. When deleted, all data generated by this stream will be erased (users have to be careful since it may potentially remove data added by other streams since the streams’ column selection may overlap)

- Archival works on each stream independently.

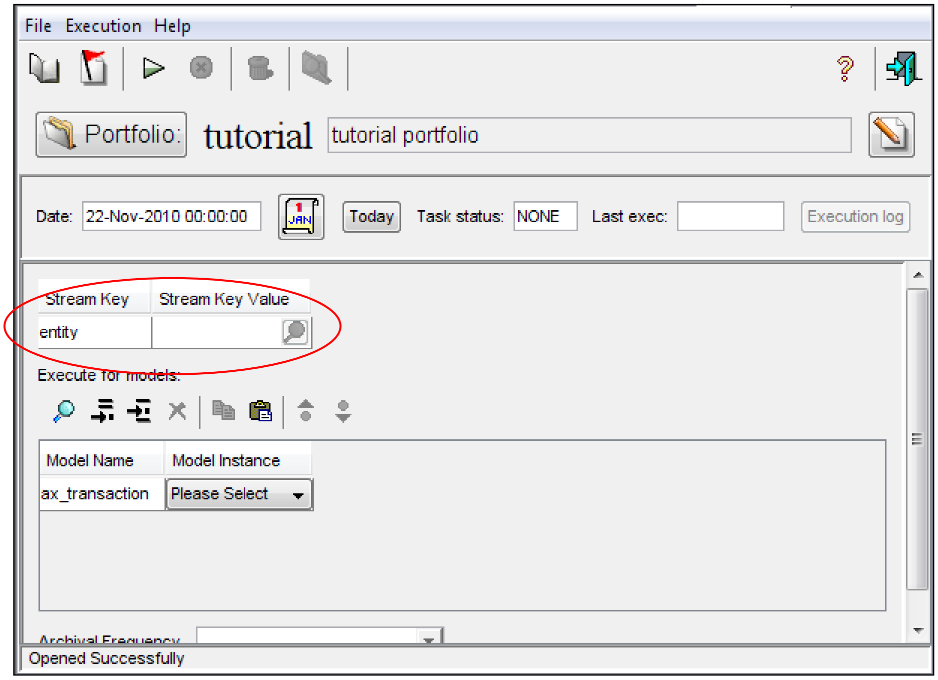
- In a Free-form report each stream produces both a new data and a new task instances.

- Workflow can be set up with parameters for streams: column names and lookup sources (optional). These parameters will be collated during run-time to all tasks.

- When executed without streams, task removes all previously executed streams first.

Stream Values Setup
To setup the stream value:

1. Open the Execution screen of a task you want to execute against the data model containing the stream – Portfolio Execution for example (Fig. 28).



Portfolio: tutorial tutorial portfolio

Date: 22-Nov-2010 00:00:00 Task status: NONE Last exec:

Stream Key	Stream Key Value
entity	

Execute for models:

Model Name	Model Instance
ax_transaction	Please Select

Archival Frequency

Opened Successfully

Fig. 28

2. On the Portfolio Execution Screen (Fig. 28) locate the Streams pane containing the name of the stream you have set up on the Model Editor

3. Use the 🔎 button of the Stream Key Value field or use manual input to enter as many values as needed, **separated by commas**.

6. DATA ADJUSTMENTS

Data Adjustment subsystem allows you to manually adjust existing records in the transaction table that results from the Task execution. The following submenu opens when the Data Adjustment option is selected from the Data Management menu:

1. Manual Adjustments
2. Adjustment History
3. Permanent Adjustments

6.01 Manual Adjustments

Selecting this option accesses Data Editor where you can modify records in the table generated as a result of the task execution. This feature is useful for example when it is required to force transactions into a different slate overriding portfolio execution results. Here you can also create new records and insert them in the transaction table.

Note: Adjustments made here takes effect for the current model instance only.

6.02 4-Eye Check

If the source has 4-eye Check enabled, two additional tabs – Parked Records and Rejected Records will be added to the Data Editor to display adjusted (parked) and rejected transactions. Modified records will move to the Parked tab's grid pending approval or rejection by the third party. Rejected records will move to the Rejected tab's grid where they will stay there until removed or modified again.

6.03 Adjustment History

Adjustment History module allows you to keep track of the changes to the records in a data source. Here you can also *revert* modified transactions to their original values, print out adjustments report or export it to various formats.

6.04 Permanent Adjustments

Functionally similar to Manual Adjustments, this module allows making changes that will affect all model instances for the modified source.

7. SHORTHANDS

Shorthand is a subset of conditions/values of one or more data attributes that have been defined and stored as macros for continuous use in portfolio definitions, aggregations, reports and analytical drill downs across the system.

Users create shorthands to combine frequently used sets of conditions for into one entity saved under a unique name.

Shorthands can be used to refer to when setting up conditions for Portfolio definitions and in setting up conditions to narrow down data for analytical Viewers, Filters and Report writers.

A simple shorthand could be a list of values taken from a particular column in the transaction table (a data attribute). For example, Product ID column may have a thousand values, but you may frequently want to isolate just few of them to be excluded from various reports. Instead of entering these Product ID codes one by one each time you set up the exclusion condition, you could group them in a shorthand and have the system to refer to it as a single value. This will minimize input errors and save time entering the values over and over again.

More complex shorthand could represent a *set of conditions for transactions selection* (business rules) of *any* degree of complexity. This way, if one or several conditions should change, you would have to modify it once in the original shorthand and not in every report or portfolio where this set of conditions is used.

Shorthands are source-specific because they use data attributes that may vary from one source to another. If you want to be able to use shorthands across several data source, you have to base these shorthands on a Detail set – an independent array of data attributes mapped to various sources.

7.01 Creating Shorthands

To create a simple shorthand with a list of customer types for the exclusion/inclusion, right-click on the “Shorthand” object category and select the “New” option. The new Shorthands Editor will open (Fig. 29).

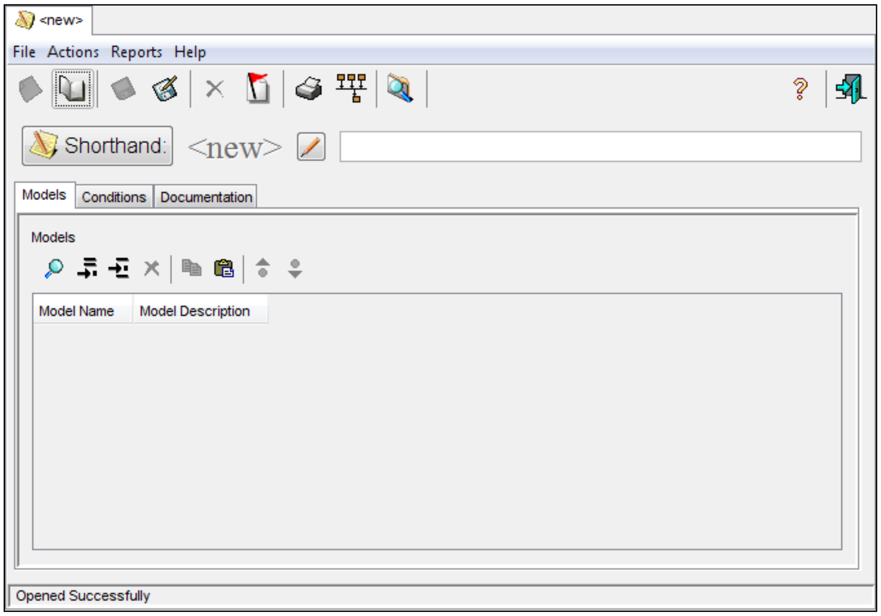


Fig. 29

You need to select one or several model(s) for which this shorthand will apply. To do this:

1. In the “Models” tab, click the 🔍 button to access the lookup

2. Choose from the lookup the model(s) with which you want to associate your shorthand.

The next step would be setting up the conditions. This is done on the Conditions tab:

1. Open the “Conditions” tab (Fig. 30).

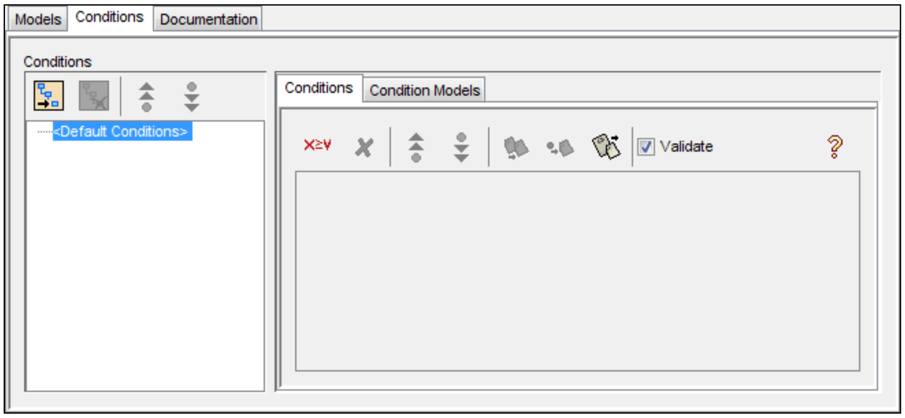


Fig. 30

2. Click on the X≥Y “Add Conditions” button to open the logical conditions menu (Fig. 31).

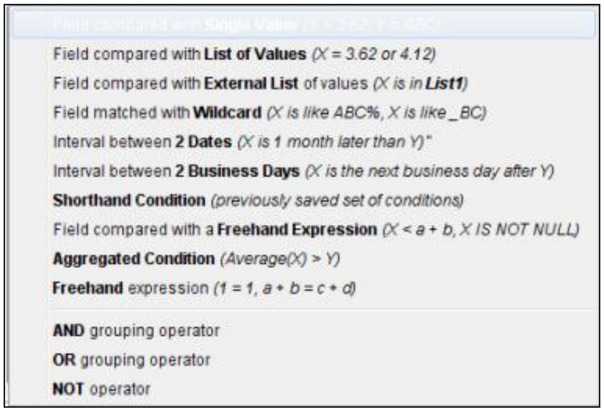

Field compared with Single Value (X < 3.62, Y = ABC)
Field compared with List of Values (X = 3.62 or 4.12)
Field compared with External List of values (X is in List1)
Field matched with Wildcard (X is like ABC%, X is like _BC)
Interval between 2 Dates (X is 1 month later than Y)"
Interval between 2 Business Days (X is the next business day after Y)
Shorthand Condition (previously saved set of conditions)
Field compared with a Freehand Expression (X < a + b, X IS NOT NULL)
Aggregated Condition (Average(X) > Y)
Freehand expression (1 = 1, a + b = c + d)
AND grouping operator
OR grouping operator
NOT operator

Fig. 31

3. Select a logical condition you want to use, (List of Values). The Condition pane appears (Fig. 32).

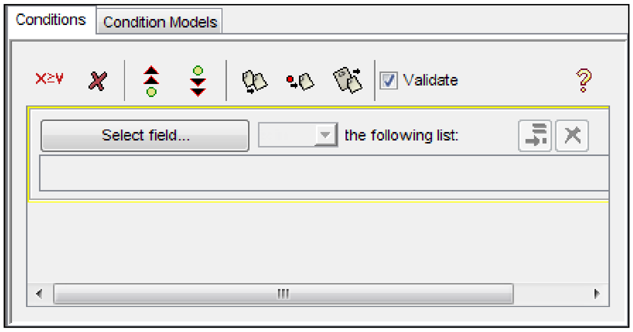
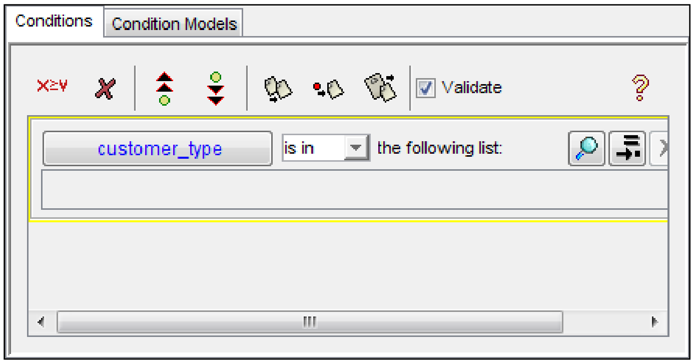


Fig. 32

4. Click on the Select Field button to open the column lookup and select the column that contains values you need to use. Select Customer Type and close the lookup. The Select Field button now displays the name of the selected column (Fig. 33).

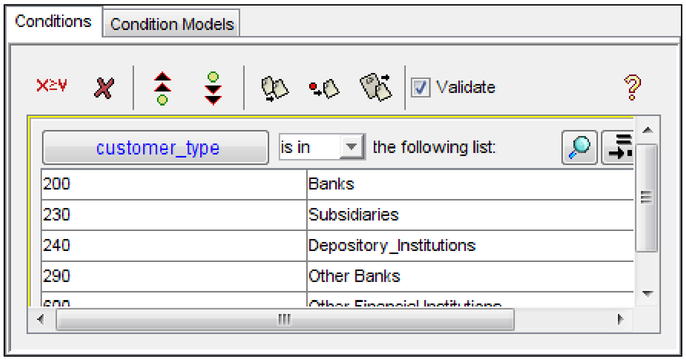


customer_type is in the following list:

Fig. 33

5. Leave the logical operator selection on ‘is in’.

6. Use the 🔍 button to populate the condition pane with values slated for inclusion/exclusion in the report line (Fig. 34).



customer_type is in the following list:

200	Banks
230	Subsidiaries
240	Depository_Institutions
290	Other Banks
600	Other Financial Institutions

Fig. 34

7. Save. When saving, assign a comprehensive name to the shorthand for identification.

Now, every time the specs of the report call for exclusion/inclusion of the listed customer types, you can refer to this shorthand instead of entering them manually every time. Should this list change, the correction would have to be done to this shorthand only.

8. PORTFOLIO

Portfolio is a tool for selective data retrieval. Portfolios allow you to:

1. Create an outline of the report and identify business rules for each report line.

2. Isolate groups of transactions from the data source following any combination of criteria you might choose, regardless of complexity of the requirements.

3. Sort data to reflect the reporting structure extracting and grouping transactions according to the specific requirements.

4. Create and populate additional columns in the transaction table.

5. Any combination of the above

Data sets generated in the process of Portfolio execution may in turn serve as data models (sources) for other portfolios, aggregations, reports etc.

Isolating Data
To isolate specific transactions you can setup a simple one-node portfolio with logical conditions where the selection criteria are specified.

For example, if you want to separate transactions dealing in US Dollars, you can create a Portfolio where in the Single Value Condition you select Domicile column as the transaction attribute and indicate that the value must equal USD in order for a transaction to qualify. The system then will go through the entire transaction table and pull all USD transactions into the resulting table.

Categorizing Data
In order to distribute data in the categories, you can create a Portfolio with a structure of various degree of complexity. You would want to create the levels in order to give names to the portfolio tiers and assign logical conditions to the nodes according to your design.

Adding Columns
You can create Portfolios in order to insert columns that are initially absent from the Source Table. In order to do this you would create a simple one-tier portfolio and use Additional Column feature to set up the column’s format and Actions feature to indicate how the column should be populated.

Multi-purpose Portfolio
You can use any combination of the above-mentioned uses in the complex multipurpose portfolios. For example, you can first isolate US Currency transactions, forcing system to discard all other records and then distribute the selected records among branches, types of instruments, etc.

You can also create additional columns and set up the system to perform some extra calculations (Portfolio actions) e.g. finding averages or calculating Credit Risk Weight to populate these fields.

8.01 Portfolio Definition

Before you start designing the Portfolio you have to define its scope, and do all necessary preliminary work that should include:

- Making sure that the transactional data source for the Portfolio does exist and is properly set up.
- Creating Control sources that will be included in Data Models.
- Building Data models, on which the Portfolio will be based.
- Setting up Shorthands for grouping often-used values (i.e. product codes, SAP accounts, date intervals, etc.)
- Setting up the Calendar to distinguish calendar and business days.

To create a new portfolio, on the Project screen's Object pane right click on "Portfolio" object category and select "New" from the menu (Fig. 35).

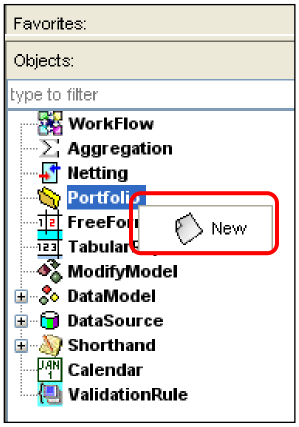


Fig. 35

Depending on the purpose, the portfolio's complexity may range from a simple single-action data extraction tool to a complex multiple level tree-like structure for advanced data management.

The elements of the Portfolio structure are presented by "Nodes" (Fig. 36). The Sample Portfolio node, shown on the illustration, is the **Root node** and represents **the first level of the Portfolio**.

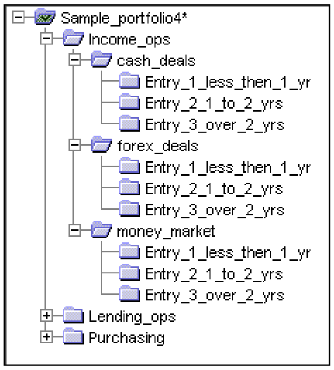


Fig. 36

The Root node is branching out with nodes that represent the second level. Each second level node of the Sample Portfolio shown here contains nodes of the third level, which branch out into the fourth and the last level of our Sample Portfolio.

Each node will be represented by a column in the table generated in the Portfolio execution (Destination Table) and can be used as data attribute in the Aggregation setup (see below).

To each node user can assign one or more conditions for transaction selection.

The nodes are processed by the system in the order they are positioned in the tree structure. Therefore, when designing a portfolio, one has to keep in mind that once processed the records retrieved by the condition assigned to a node are “extracted” from the source and will not be considered again.

The *Root Node* at the base of the structure represents the total Portfolio and bears Portfolio’s name and description. User can assign conditions to the Root node. This condition will be *inherited* by each and every node within the Portfolio.

8.01.01 Root Node Properties Setup

A crucial step in the Portfolio setup is defining Root Node’s properties. The Root Node properties include the following mandatory steps:

- Name the Root node
- Root Node (Portfolio) description
- Data Models Selection
- Portfolio Levels names and descriptions setup:
    - Portfolio level is generated by the system and bears the name of the entire Portfolio.
    - Additional levels must be created and named reflecting their purpose.
- Execution Algorithm selection

Other parameters may include:

- Additional columns setup
- Parameters setup – various parameters to increase the efficiency of the Portfolio execution.
- Logical Conditions setup – inserting logical conditions on the Portfolio level will create a subset of data that allows narrowing down the amount of data to be processed.
- Actions Setup – Action is a statement that defines rules for populating Additional columns. It is performed at the Node Properties panel’s Action tab (see below).

To name a node:

1. Select the node by clicking on it.
2. Use the Rename Node option from the Edit Menu to switch the selected node’s nametag into editing mode. Alternatively, you can use the Rename Node option from the *right click* menu.
3. Type in a new name for the selected node.

8.01.02 Selecting Models

Data models provide data for your Portfolio to run against. To select models for the Portfolio:

1. On the Portfolio Editor Find Models tab (Fig. 37)

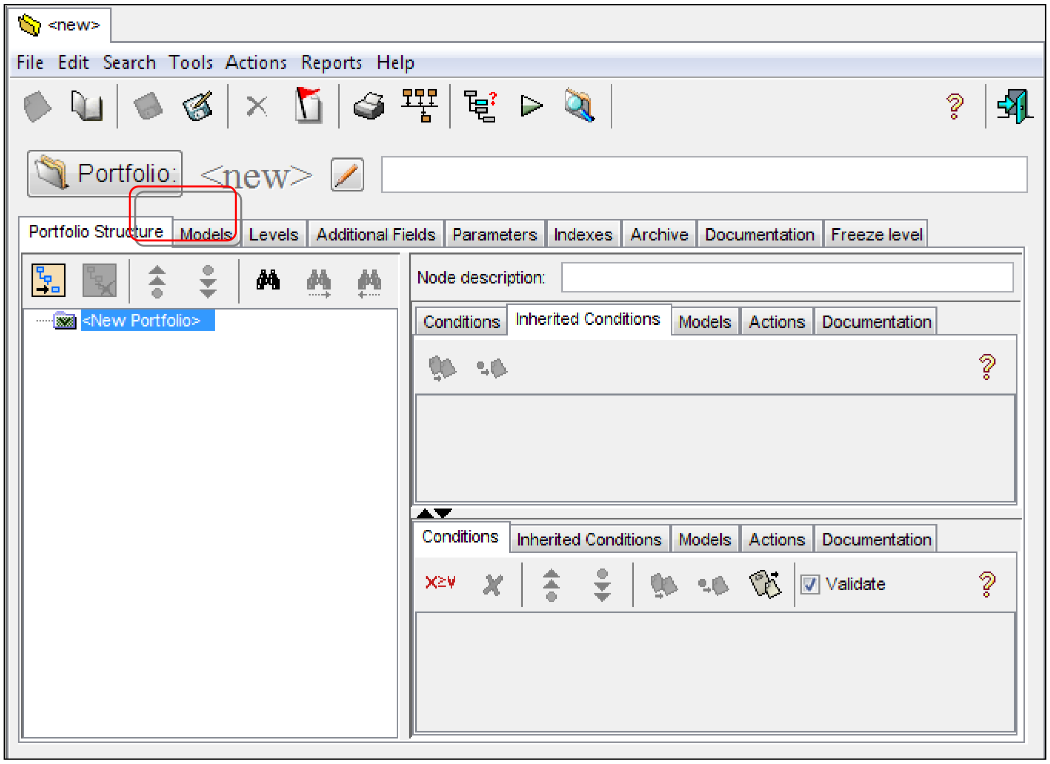
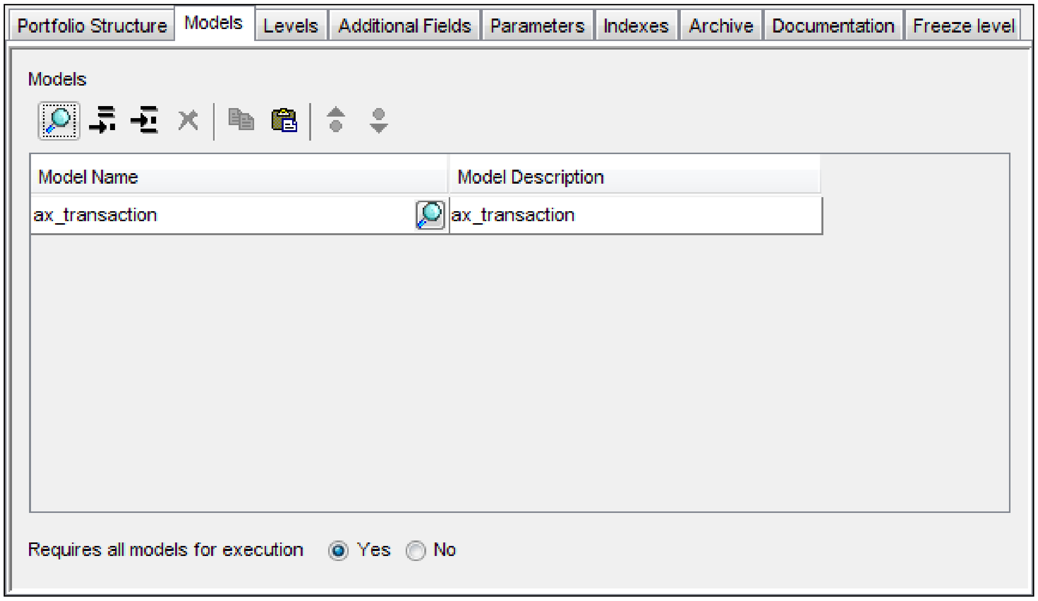


Fig. 37

2. Open the Models tab.

3. Click on Lookup button. The lookup will open listing all data models available for selection.

4. Select a model(s) and click OK to close the lookup. Use Ctrl button on your keyboard to do multiple select. The Models Tab displays model(s) selected for the Sample Portfolio (Fig. 38).



Model Name	Model Description
ax_transaction	ax_transaction

Requires all models for execution ● Yes ○ No

Fig. 38

8.01.03 Levels Setup

Level is a structural element of the Portfolio. Simple portfolios may consist of a single (Portfolio) level. Other Highly complex portfolios for reporting compliance may involve multiple levels.

Level names are used to identify the hierarchical components of the Portfolio. Naming levels prompts the system, when saving the portfolio, to create as many columns in the resulting table.

The system also generates the lookup sources for these columns and inserts them into the resulting database. These values can be later used as criteria for compartmentalizing data in aggregations and reports. The names you assign to Levels will become headers of these columns.

Note: If you do not name the levels, no hierarchy-identifying columns will be created.

Our Sample Portfolio structure includes two levels: Segments (1) and Entries (2).

To create levels:

1. Open the Levels tab.

2. Click on the Add button of the Levels tab's tool bar. A row appears in the field (Fig. 39).

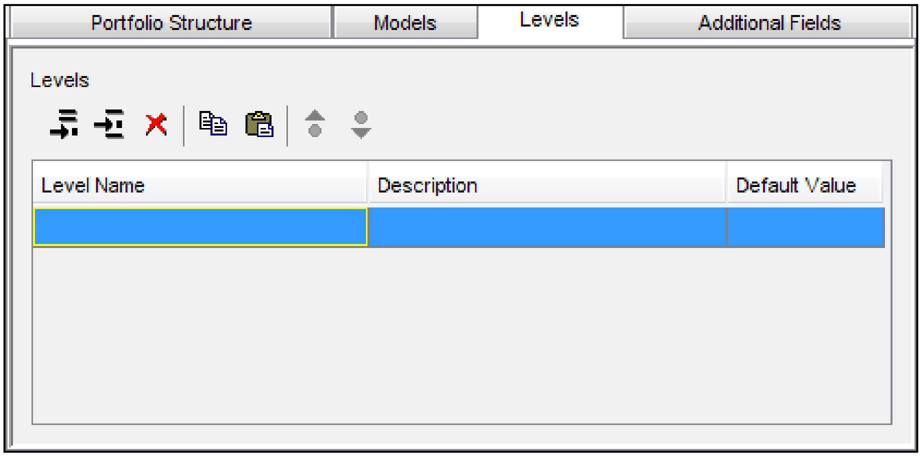


Fig. 39

3. In the Level Name field type in 'segment'.

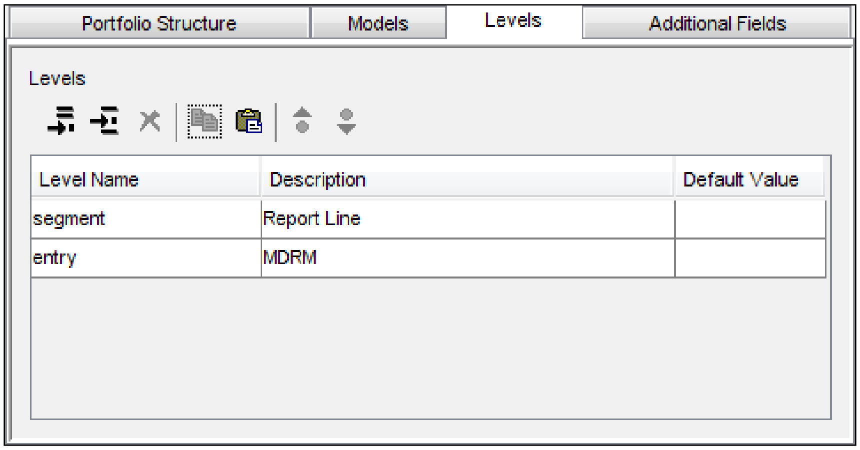
   Note: When naming levels avoid using UPPER CASE. These names will be used by the case-sensitive databases, which may not accept upper case characters in column names.

4. Click on the **Description field** and type in the description for the Segments Level name (*Report Line* in our example).

   Note: The Description you enter here will appear in the Control Table generated during the Portfolio save. This Control table can be enabled (made visible) on the Header Manager.

5. **Default Value** field contains the default value you want to appear in the level column in the absence of generated data. If you do not enter anything, the field will contain no entries.

6. Repeat the above steps for each additional level. At the end of the procedure the Levels tab's grid should look as shown on the illustration below (Fig. 40).



Level Name	Description	Default Value
segment	Report Line	
entry	MDRM	

Fig. 40

8.01.04 Setting up Portfolio Structure

For each portfolio, the user defines a root node at the top of the portfolio hierarchy. This node is generated by the system when you start a New Portfolio.

To the Root node other nodes are added in levels to build the hierarchy as required by the specs.

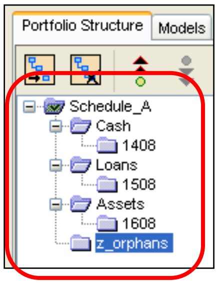


Fig. 41

In our sample (Fig. 41) “Schedule_A” is the root node.

“Cash”, “Loans”, and “Assets” are the 1st level nodes each representing a line in the report.

Note: The “z_orphans” node is also considered a 1st level node and we will touch upon that later.

The “1408”, “1508”, and 1608” are the 2nd level nodes each representing a specific MDRM. To create the first level of the Portfolio hierarchy:

1. Click on the Portfolio node to select it.

2. Press (Insert Node) button. Alternatively, you can *right click* on the root node and choose Insert Node option from the menu. The system inserts adds a node and names it *NewNode_001*.

3. Repeat steps 1 and 2 to insert as many nodes as you need on the First (Schedule) Level (three nodes in our example) following the outline of the sample Portfolio structure.

   The system will assign all the nodes the same default name – *NewNode* – followed by consecutive numbers.

4. Rename *Level 1* nodes using already described naming procedure.

To add the Second Level to the structure of the Sample Portfolio:

1. Highlight a Level 1 (*Cash*) node.

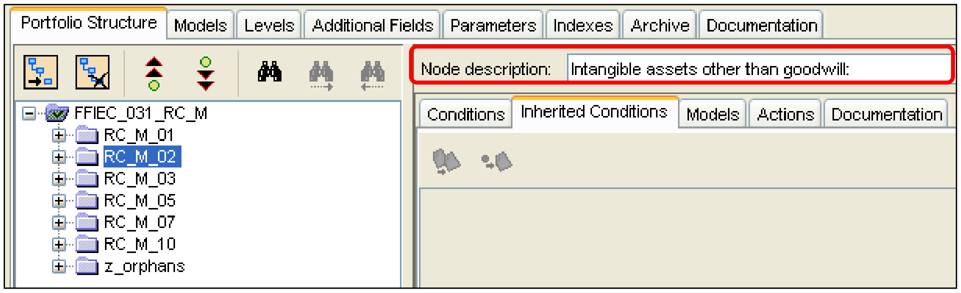
2. Press the Add Node button to insert a node of the Entries level.

3. Repeat steps 1 and 2 to insert as many nodes as required by the Sample Portfolio guidelines. The system will assign all the nodes the same default name – *NewNode* – followed by consecutive numbers

4. Rename *Level 2* nodes using the already described node naming procedure.

Node description

The description for each node should reflect what type of data will be displayed for that particular report line. On Fig. 42, you can see how the 1st level node, "RC_M_02" node has a decription stating what data should be expected to be extracted by the node's condition.



Node description: Intangible assets other than goodwill:

Fig. 42

8.01.05 Logical Conditions Setup

To get the nodes to actually pull the data, we need to set up logical conditions. In order to do this, click X≥V (insert condition). This will bring up a menu (Fig. 43). With this, we can set up the "1408" to pull only certain types of data, in this case, Cash.

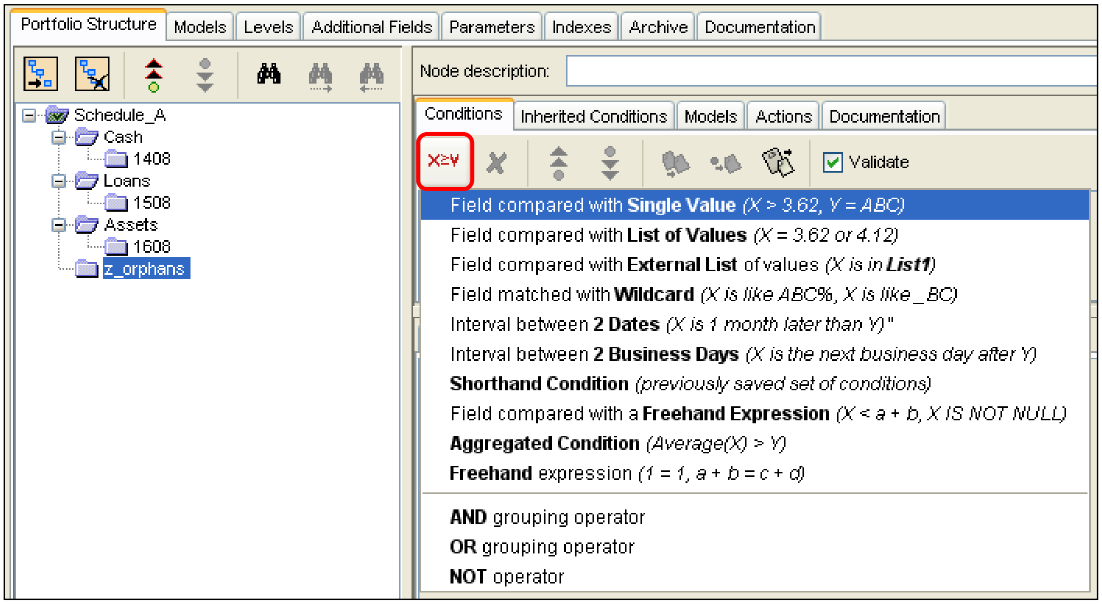

Portfolio Structure
Models
Levels
Additional Fields
Parameters
Indexes
Archive
Documentation
Node description:
Schedule_A
Cash
1408
Loans
1508
Assets
1608
z_orphans
Conditions
Inherited Conditions
Models
Actions
Documentation
X≥V
☑ Validate
Field compared with **Single Value** *(X > 3.62, Y = ABC)*
Field compared with **List of Values** *(X = 3.62 or 4.12)*
Field compared with **External List** of values *(X is in **List1**)*
Field matched with **Wildcard** *(X is like ABC%, X is like _BC)*
Interval between **2 Dates** *(X is 1 month later than Y)"*
Interval between **2 Business Days** *(X is the next business day after Y)*
Shorthand Condition *(previously saved set of conditions)*
Field compared with a **Freehand Expression** *(X < a + b, X IS NOT NULL)*
Aggregated Condition *(Average(X) > Y)*
Freehand expression *(1 = 1, a + b = c + d)*
AND grouping operator
OR grouping operator
NOT operator

Fig. 43

Conditions should usually be placed in the 2nd level nodes. However, in some cases however every node will have a condition. If, for example, we design a portfolio that represents one column that takes the ‘weighted average’ where each MDRM is a different asset, our root node would have a condition pulling only weighted average data and each MDRM would have a condition stating to pull that particular asset.

Note: Lower level nodes inherit the condition from the nodes above them, this way you don’t have to repeat the same condition for each MDRM.

Note: Once data is pulled, it will not be pulled again by another node. If you accidently set 1408 and 1508 to pull Cash, 1508 would not have any data because it would all be in 1408.

Note: Refer to the Appendix A for complete instructions on how to set up logical conditions.

Orphans Node

The orphans node “z_orphans” is created to pull all the data that due to various reasons was not collected by previous nodes. If the model we used for this portfolio had “Cash”, “Loans”, “Assets”, and “Securities”, “z_orphans” would pull all the “Securities”. We can either have a **Freehand Expression** stating “1=1” or we can have a **Shorthand Condition** set to a shorthand with the condition already created.

8.01.06 Additional Fields

The “Additional Fields” tab allows the user to add fields to the source data table. This is effective when the user is being rushed to complete a report and doesn’t want to hassle with going back into the Data Source, creating a new file, running the Data Source, and all the other steps that would be taken. Adding a field is exactly the same as adding field when working in the “Layout” tab of the Data Source.

The Additional Fields tab located in the Portfolio Node setup panel (Fig. 44) allows you to set up columns that will be added to the Destination Table (DT) during the Portfolio execution.

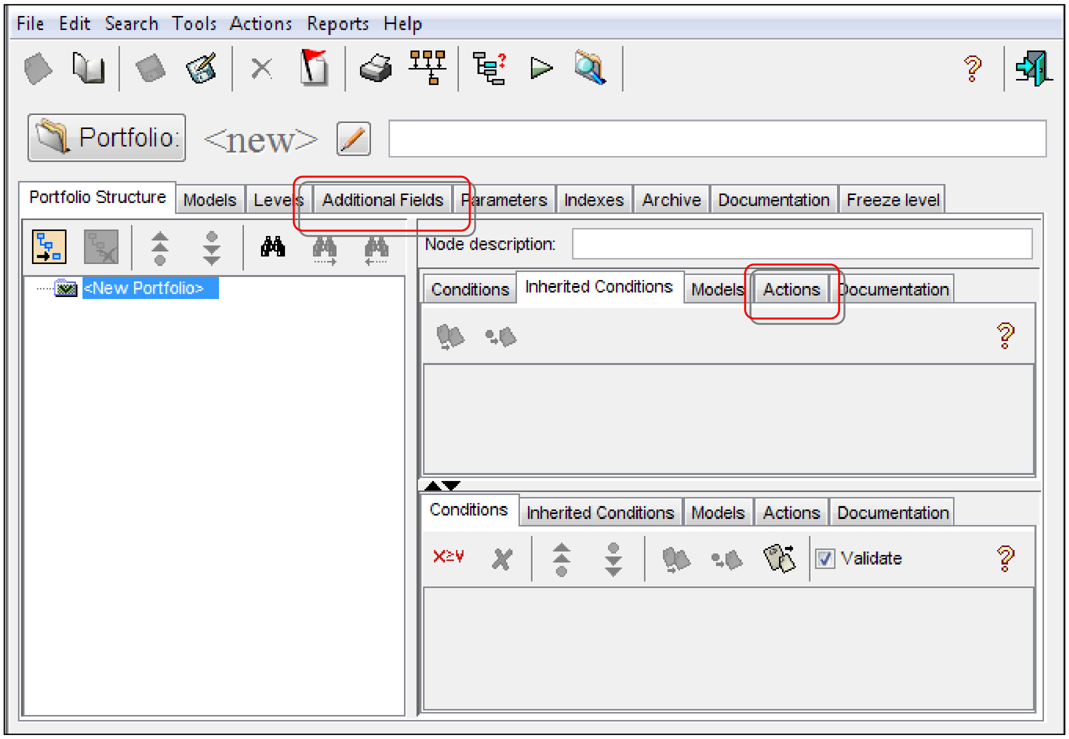


Fig. 44

These columns, in conjunction with subsequently defined Actions (see below), may be used to perform calculations based either on the original source data or on the results of other additional columns/actions.

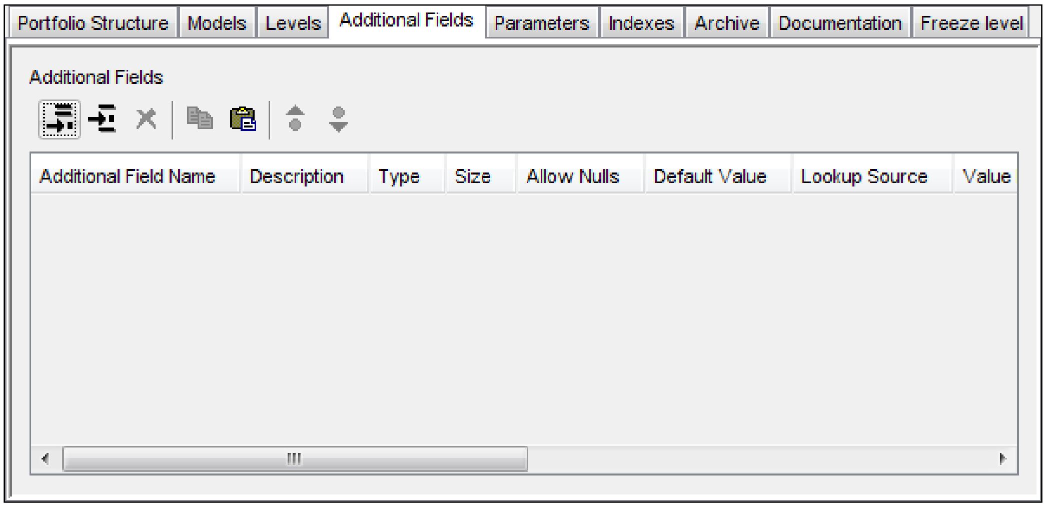
Additional fields setup has to be performed in conjunction with "**Portfolio Actions**", which will determine how to populate the created columns. The actions can be set for each node of the Portfolio.

If no action is set, the system will enter the default value assigned for the additional field. For example, on the Root level the value may be 0 by default (no action set), whereas on other levels, depending on the bracket, the value may vary according to the action defined for this particular node.

Note: Portfolio Actions may be constrained by the Execution Algorithm selected for the Portfolio.

On the Additional Fields tab (Fig. 45) you define the following mandatory column parameters:

1. Column name
2. Column description
3. Data type
4. Column size
5. Default value.



Portfolio Structure | Models | Levels | Additional Fields | Parameters | Indexes | Archive | Documentation | Freeze level

Additional Fields

Additional Field Name	Description	Type	Size	Allow Nulls	Default Value	Lookup Source	Value

Fig. 45

Note: Column size is expressed in the number of characters you anticipate your entry would NOT exceed. This parameter is relevant only when the data type is VARCHAR (text).

To complete the Additional Column setup you will have to:

1. Enter the Additional Column name into the Index input field of the Parameters tab (see Portfolio Definition Parameters tab section below).

2. Define an "**Action**" to be performed by the system in order to populate the Additional column with data (see Node Properties/Actions tab below).

8.01.07 Portfolio Actions

Action (Portfolio Action) is a condition expression that determines how the Additional Column will be populated. Actions can be assigned to an additional column for any node of the Portfolio at any level and may vary from one node to another depending on what you want the additional column to display in one or other transaction group.

Actions can refer to data in the fields in the original source table or to data in other additional columns generated as results of their own actions in the process of the Portfolio execution.

Portfolio Action constitutes the SET or UPDATE performed on execution of an **SQL statement** containing references to columns either in the Source or in the Destination table.

8.01.09 Saving Portfolio

When you press Save, the following events occur:

1. The system generates the resulting Data Source Portfolio-result[portfolio_name].data_model_name with the layout containing:

    a. The Index column – the column you selected as an index when setting up the main data source (Axiom Index in our case).

    b. The system-generated column called “path_portfolio_name” where, when viewing data, the location of every record in the Portfolio hierarchy could be traced.

    c. The system-generated column for each portfolio level

    d. The system-generated Column for each additional column specified.

2. The system generates *path_portfolio_name* column where you will be able to trace the exact location of every record in the Portfolio hierarchy.

3. The system generates Portfolio Control (lookup) sources – one for each system-generated column (level columns and *path_portfolio_name* column) and inserts them in the resulting source layout.

    The resulting data sources (including control) will be located in the DB sources specified by the 'DBSource' parameter.

    The data storage type of the resulting data source can be defined by 'Storage Type' parameter and defaults to 'Continuous_Partition'.

    Note: Control data sources always have 'Permanent' data storage.

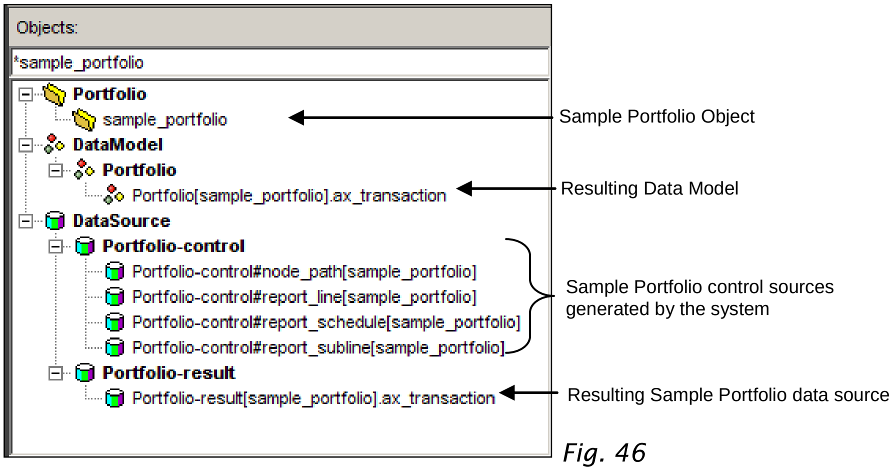
4. The system generates a resulting Data Model with the root node named Portfolio[portfolio_name].

    The original Data Model used in the saved Portfolio will form a *subordinate* node together with the Control sources either system-generated columns.

Note: In some cases users may want to override alias for the resulting data source in the resulting data model. That can be done by changing 'Custom Portfolio Result Alias' parameter

5. If you assigned a Control (lookup) Source for your Additional Column, it will also be included in the Resulting Data Model structure.

The Object Pane of the Project Management screen displays objects generated as a result of saving the new Portfolio (Fig. 46).



Objects:

*sample_portfolio

- Portfolio
  - sample_portfolio
- DataModel
  - Portfolio
    - Portfolio[sample_portfolio].ax_transaction
- DataSource
  - Portfolio-control
    - Portfolio-control#node_path[sample_portfolio]
    - Portfolio-control#report_line[sample_portfolio]
    - Portfolio-control#report_schedule[sample_portfolio]
    - Portfolio-control#report_subline[sample_portfolio]
  - Portfolio-result
    - Portfolio-result[sample_portfolio].ax_transaction

Sample Portfolio Object

Resulting Data Model

Sample Portfolio control sources generated by the system

Resulting Sample Portfolio data source

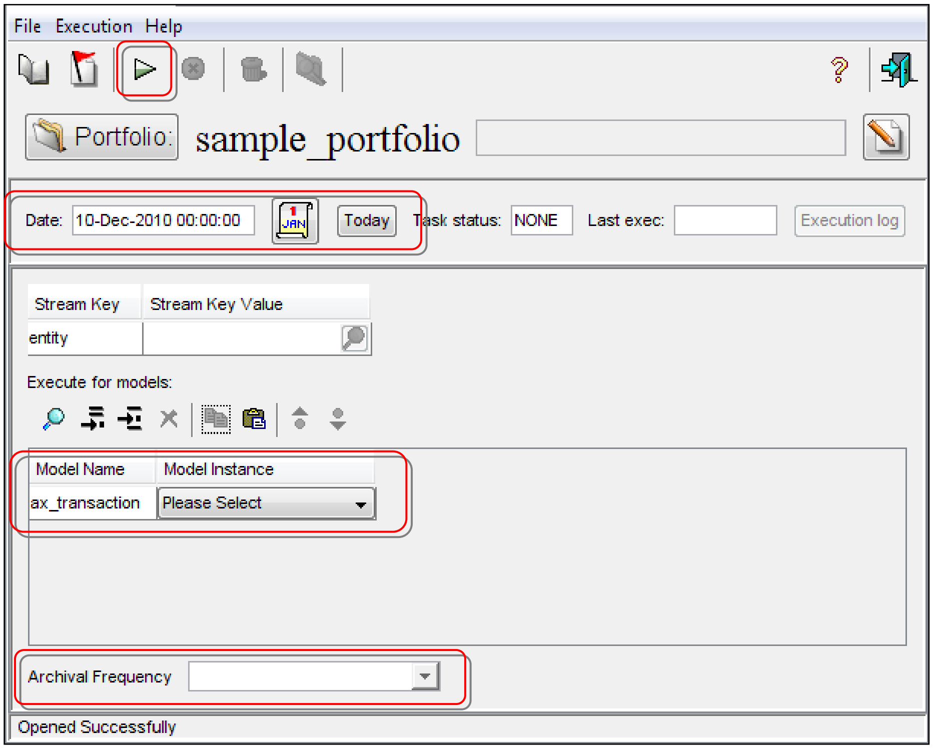
Fig. 46

8.02 Portfolio Execution

After you completed defining the Portfolio you can initiate Portfolio Execution. Portfolio Execution screen (Fig. 47) can be accessed directly from the Portfolio Editor or through the Main Menu.

The following must be done on the Portfolio Execution screen before you start the execution:

1. Set the Execution Date (Current date by default)
2. Set the Model Date – Select a model instance you want the system to use
3. Set the Archival schedule



sample_portfolio

Date: 10-Dec-2010 00:00:00

Task status: NONE

Last exec:

Stream Key	Stream Key Value
entity	

Execute for models:

Model Name	Model Instance
ax_transaction	Please Select

Archival Frequency

Opened Successfully

Fig. 47

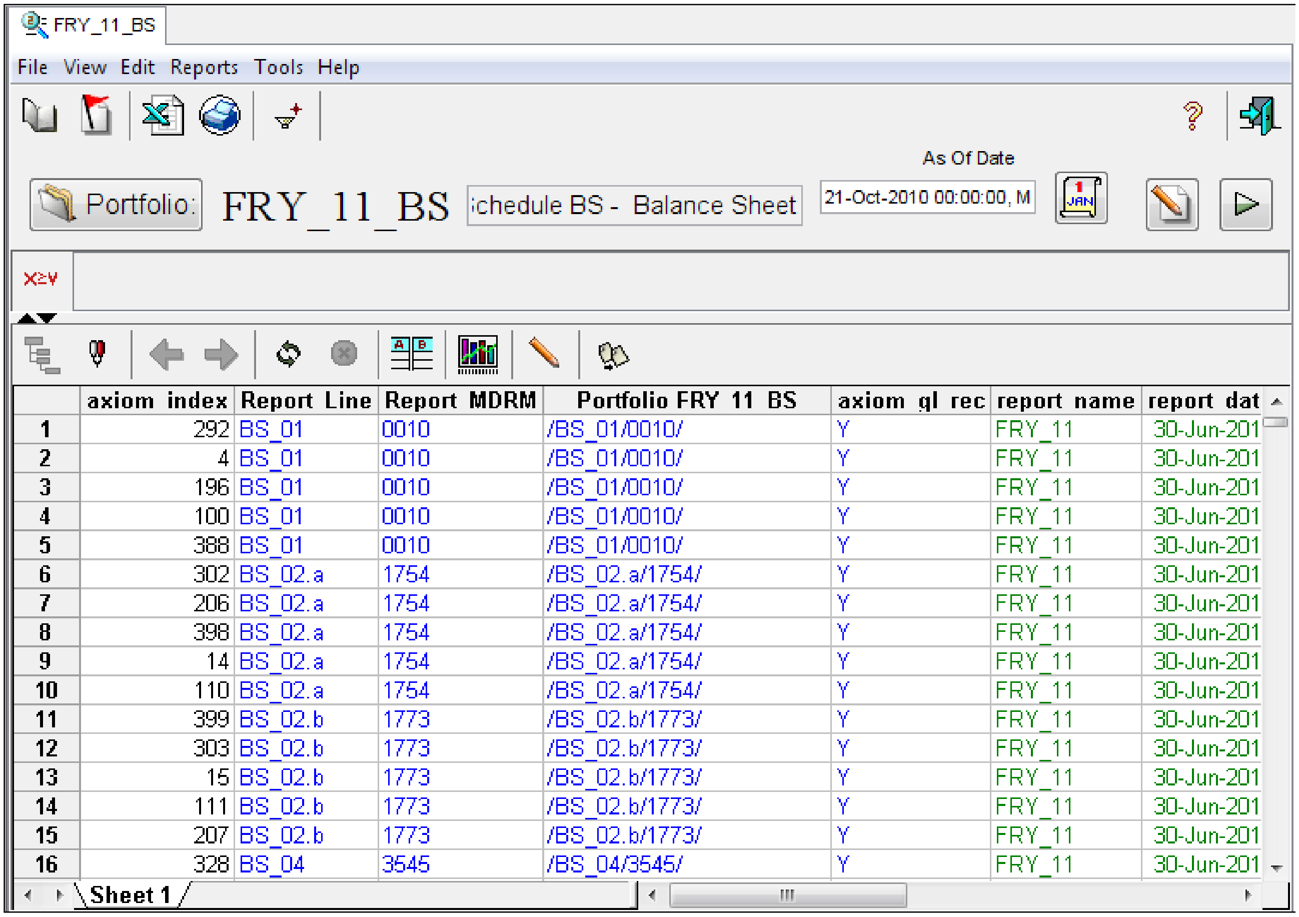
Note: If you intend to run the Portfolio against streams, you will need to define Stream Values at this point.

Pressing ▷ button initiates the execution process. The Task status will change to START and remain that until the finish. When completed successfully, the Task status would change to ACTIVE.

8.03 Portfolio Viewer

Portfolio Viewer screen (Fig.48) allows to see Portfolio execution results data **on the transactional level**.

You can get to the Portfolio Viewer screen directly from the Main Menu's Portfolio Management subsystem or from various other screens dealing with portfolio data management, e.g. Portfolio Execution, Global status etc.



FRY_11_BS ːchedule BS - Balance Sheet

As Of Date: 21-Oct-2010 00:00:00, M

	axiom index	Report Line	Report MDRM	Portfolio FRY 11 BS	axiom gl rec	report name	report dat
1	292	BS_01	0010	/BS_01/0010/	Y	FRY_11	30-Jun-201
2	4	BS_01	0010	/BS_01/0010/	Y	FRY_11	30-Jun-201
3	196	BS_01	0010	/BS_01/0010/	Y	FRY_11	30-Jun-201
4	100	BS_01	0010	/BS_01/0010/	Y	FRY_11	30-Jun-201
5	388	BS_01	0010	/BS_01/0010/	Y	FRY_11	30-Jun-201
6	302	BS_02.a	1754	/BS_02.a/1754/	Y	FRY_11	30-Jun-201
7	206	BS_02.a	1754	/BS_02.a/1754/	Y	FRY_11	30-Jun-201
8	398	BS_02.a	1754	/BS_02.a/1754/	Y	FRY_11	30-Jun-201
9	14	BS_02.a	1754	/BS_02.a/1754/	Y	FRY_11	30-Jun-201
10	110	BS_02.a	1754	/BS_02.a/1754/	Y	FRY_11	30-Jun-201
11	399	BS_02.b	1773	/BS_02.b/1773/	Y	FRY_11	30-Jun-201
12	303	BS_02.b	1773	/BS_02.b/1773/	Y	FRY_11	30-Jun-201
13	15	BS_02.b	1773	/BS_02.b/1773/	Y	FRY_11	30-Jun-201
14	111	BS_02.b	1773	/BS_02.b/1773/	Y	FRY_11	30-Jun-201
15	207	BS_02.b	1773	/BS_02.b/1773/	Y	FRY_11	30-Jun-201
16	328	BS_04	3545	/BS_04/3545/	Y	FRY_11	30-Jun-201

Fig. 48

The Portfolio Viewer screen will display transactions pulled from the original source in the course of the Portfolio execution.

Portfolio Viewer allows you to:

1. Isolate specific groups of transactions using Logical conditions for data selection
2. Configure data display on the screen and its printouts (Header Manager)
3. Save snapshot views of data

4. Adjust values of some transaction attributes (via Manual Adjustments)

5. Force transactions overriding the portfolio execution results (Portfolio Override)

6. Printout the contents of the data grid

8.03.01 Opening Portfolio

To open a portfolio, use one of several Open Portfolio functions available on the Portfolio Viewer controls. The lookup will offer you a list of portfolios available to you.

If the Portfolio you select is based on a single Data Model, the Data Grid of the screen will automatically load transactional data from the Main Source of the former. If you select a portfolio based on *several* Data Models, the screen will come up empty.

Use the button on the Data Grid's tool bar and select a Data Model from the Source Selector's menu.

8.03.02 Source Selector

Source selector button is available when several models are associated with the Portfolio or/and when there are one-to-many relationships embedded in the data model structure. Pressing this button opens a menu (Fig. 49) listing main sources of every data model associated with the selected Portfolio.

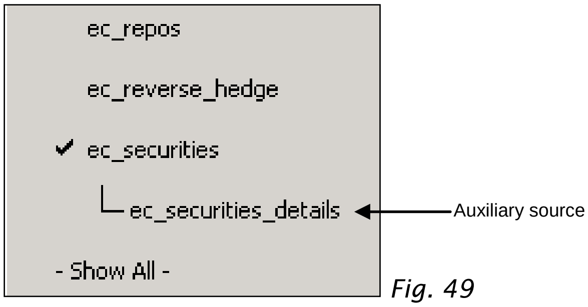


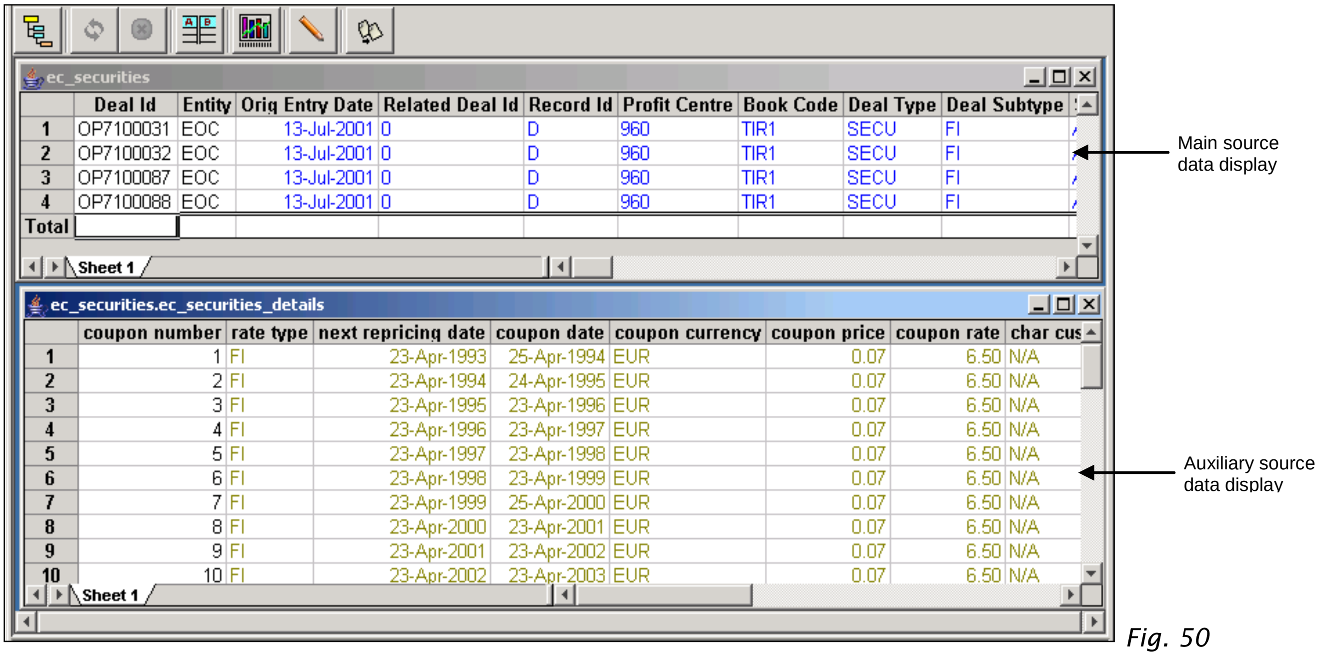
Fig. 49

- Show All - option allows you to open all models/sources at once. Each source will open in a separate grid, which you may close by pressing ☒ button in the top right corner of a grid.

8.03.03 Auxiliary Source Access

The Data Model list branches out if there are *one-to-many* auxiliary sources attached to any model's Main source. Users have an option to view data from these underlying sources (usually in conjunction with the main source data) by selecting them from the Data Model hierarchy display that opens when you press Model/ Source selector button (Fig. 49).

In our example, such source, *ec_securities_details*, appears on the Source Selector button's menu as a branch when the *ec_securities* model is selected for viewing (Fig. 49). Selecting *ec_securities_details* will create an additional grid in the pane displaying data from this source for each line in the *ec_securities* table (Fig. 50).


ec_securities
Sheet 1
ec_securities.ec_securities_details
Sheet 1
Main source data display
Auxiliary source data display

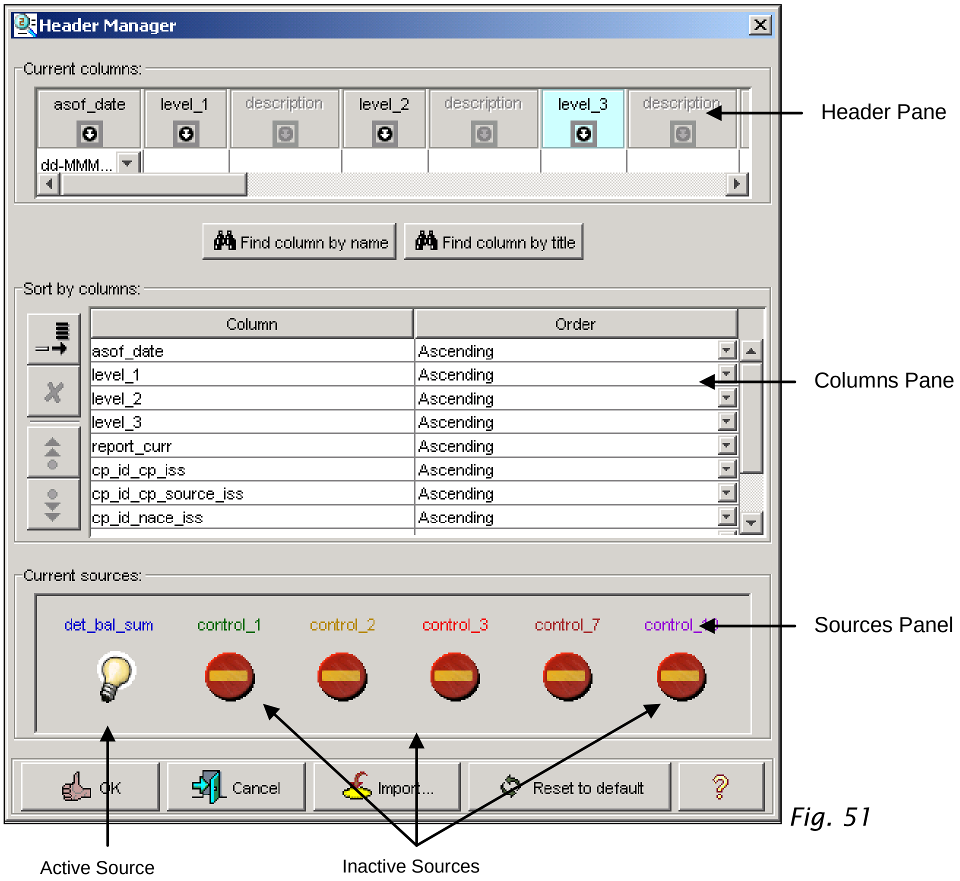
	Deal Id	Entity	Orig Entry Date	Related Deal Id	Record Id	Profit Centre	Book Code	Deal Type	Deal Subtype
1	OP7100031	EOC	13-Jul-2001	0	D	960	TIR1	SECU	FI
2	OP7100032	EOC	13-Jul-2001	0	D	960	TIR1	SECU	FI
3	OP7100087	EOC	13-Jul-2001	0	D	960	TIR1	SECU	FI
4	OP7100088	EOC	13-Jul-2001	0	D	960	TIR1	SECU	FI
Total									

	coupon number	rate type	next repricing date	coupon date	coupon currency	coupon price	coupon rate	char cus
1	1	FI	23-Apr-1993	25-Apr-1994	EUR	0.07	6.50	N/A
2	2	FI	23-Apr-1994	24-Apr-1995	EUR	0.07	6.50	N/A
3	3	FI	23-Apr-1995	23-Apr-1996	EUR	0.07	6.50	N/A
4	4	FI	23-Apr-1996	23-Apr-1997	EUR	0.07	6.50	N/A
5	5	FI	23-Apr-1997	23-Apr-1998	EUR	0.07	6.50	N/A
6	6	FI	23-Apr-1998	23-Apr-1999	EUR	0.07	6.50	N/A
7	7	FI	23-Apr-1999	25-Apr-2000	EUR	0.07	6.50	N/A
8	8	FI	23-Apr-2000	23-Apr-2001	EUR	0.07	6.50	N/A
9	9	FI	23-Apr-2001	23-Apr-2002	EUR	0.07	6.50	N/A
10	10	FI	23-Apr-2002	23-Apr-2003	EUR	0.07	6.50	N/A

Fig. 50

8.03.02 Header Manager

Header Manager (Fig.51) allows you to customize display in the data grid of the Viewer screens in following ways:



Column	Order
asof_date	Ascending
level_1	Ascending
level_2	Ascending
level_3	Ascending
report_curr	Ascending
cp_id_cp_iss	Ascending
cp_id_cp_source_iss	Ascending
cp_id_nace_iss	Ascending

Fig. 51

1. Activating/deactivating data sources display

2. Editing column names
3. Editing column sizes
4. Selecting date format
5. Selecting numerical data format
6. Controlling column visibility
7. Repositioning columns
8. Changing sorting options

9. DATA AGGREGATION

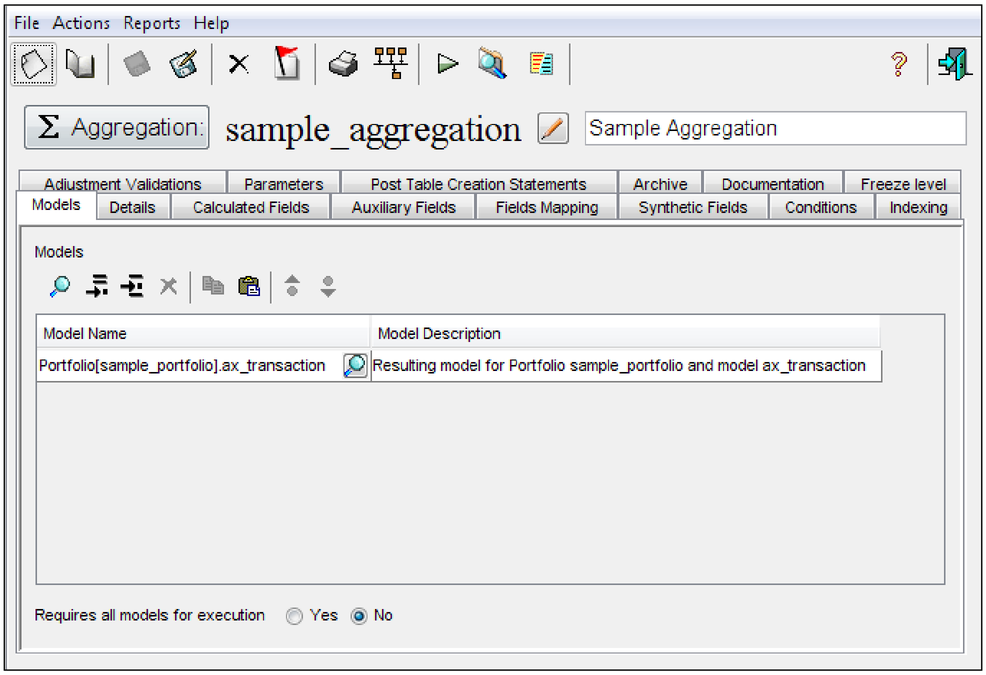
Data Aggregation is a process of creating *summaries* of the data. The data can be arriving from the original source table or data generated as a result of a Portfolio Execution. Summaries are created by using data attributes as the aggregation criteria. Data attributes (detail elements) are names of the columns in the source table that can be used to categorize the data into meaningful entities for analytical drill down.

As the data initially arrives from the source, it only becomes available for the viewing on the transactional level. The only way to view the data is by opening the entire transaction table with hundreds of thousands of records. Although you will be able to narrow down the data somewhat by applying logical conditions, this is hardly an efficient way to study the figures and prepare reports.

You will need to view the data in the *summarized* or aggregated form in order to study the numbers as they reflect the business activities. For example you may want to see the numbers broken down by such data attributes like Counterparty ID, Industry Code, Product ID and Location and see balance summaries for any combination of these details. To accomplish this you would have to *tell* the system what data attributes to use for the breakdown of the figures and what *calculated fields* to use to display the summaries. Thus, you have to define what we term an **Aggregation Task**, in which you would specify these parameters and execute it against the source data.

9.01 Aggregation Task Setup

The definition of the Aggregation Task is performed on the Aggregation Editor screen (Fig. 52) and consists of several mandatory and optional steps:



sample_aggregation Sample Aggregation

Model Name	Model Description
Portfolio[sample_portfolio].ax_transaction	Resulting model for Portfolio sample_portfolio and model ax_transaction

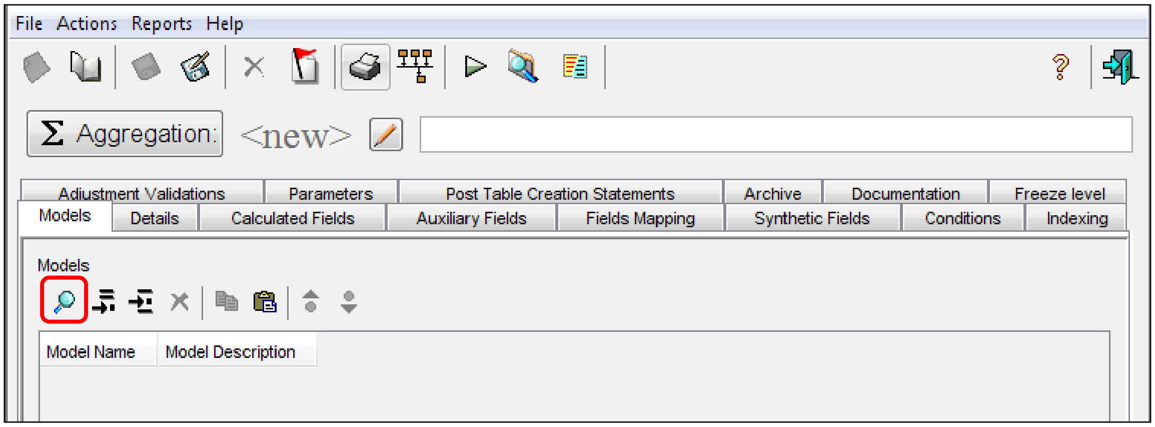
Requires all models for execution ○ Yes ● No

Fig. 52

9.01.01 Data Models Selection

First step in the Aggregation task setup is indicating to the system which data sources to apply the aggregation rules. You can select any number of data models available from the lookup.

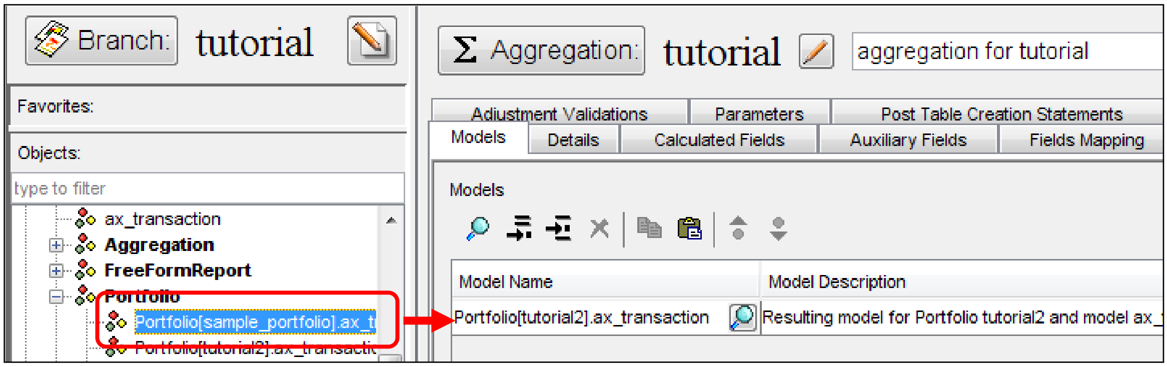
When you first create an aggregation, it opens into the “Models” tab, as shown on Fig. 53.



Model Name	Model Description

Fig. 53

Click on the (Lookup) and select the model that contains data source you need against which you will execute your aggregation. Most often it will be a model that results from the Portfolio execution (Fig. 54).

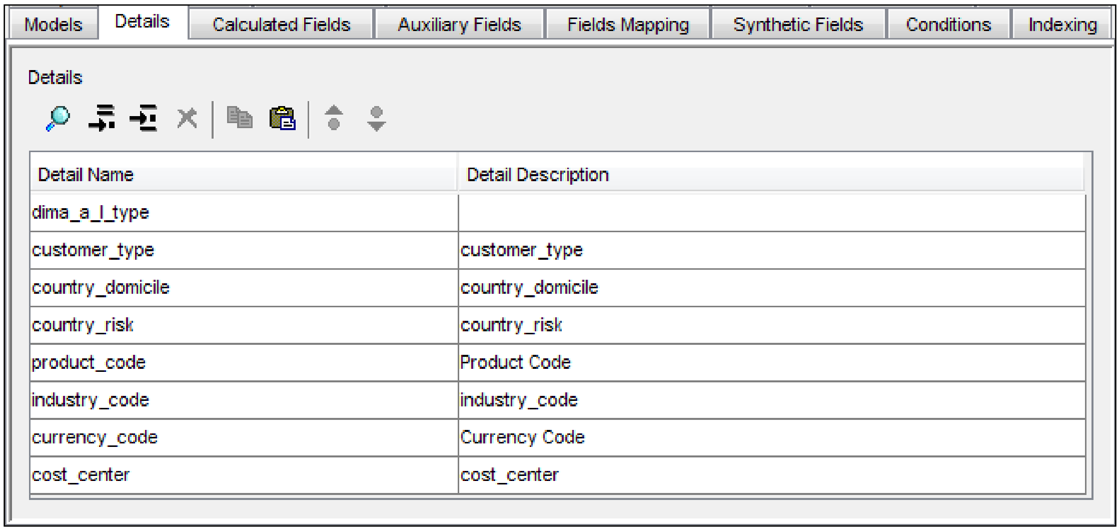


Model Name	Model Description
Portfolio[tutorial2].ax_transaction	Resulting model for Portfolio tutorial2 and model ax_

Fig. 54

9.01.02 Details Selection

Details selection is done on the Details Tab (Fig. 55) and provides the grouping criteria for the system to use when aggregating data. Selected data attributes will be also used for breaking down the numbers during the analysis.

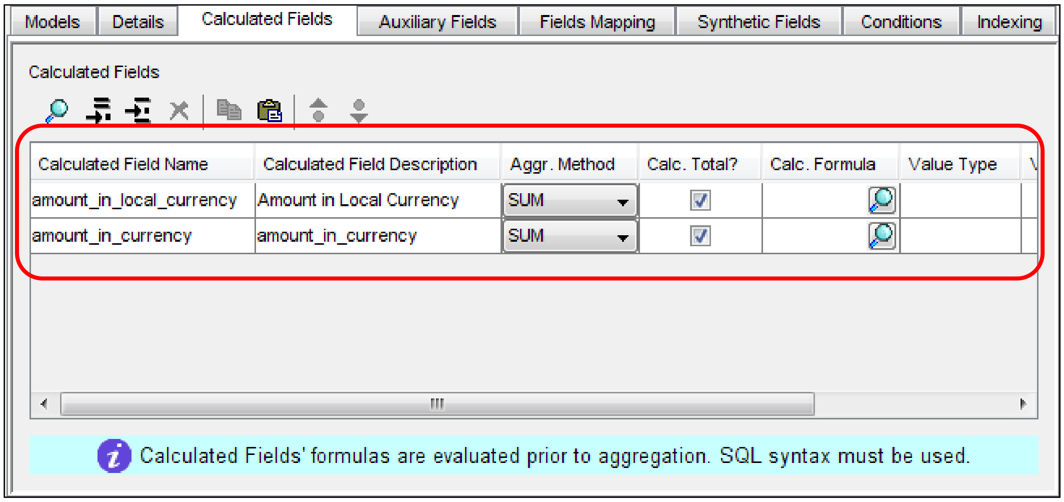


Detail Name	Detail Description
dima_a_l_type	
customer_type	customer_type
country_domicile	country_domicile
country_risk	country_risk
product_code	Product Code
industry_code	industry_code
currency_code	Currency Code
cost_center	cost_center

Fig. 55

9.01.03 Calculated Fields Selection

Calculated Fields contain numerical data resulting from aggregation. Calculated Fields selection (Fig. 56) depends on requirements and determines what numbers will be shown on the analytical screens display.



Calculated Field Name	Calculated Field Description	Aggr. Method	Calc. Total?	Calc. Formula	Value Type
amount_in_local_currency	Amount in Local Currency	SUM	☑		
amount_in_currency	amount_in_currency	SUM	☑		

Calculated Fields' formulas are evaluated prior to aggregation. SQL syntax must be used.

Fig. 56

The Calculated Fields can be:

1. Selected from columns of the original source. If several sources are involved columns may need to be mapped manually to every source where automatic mapping failed.

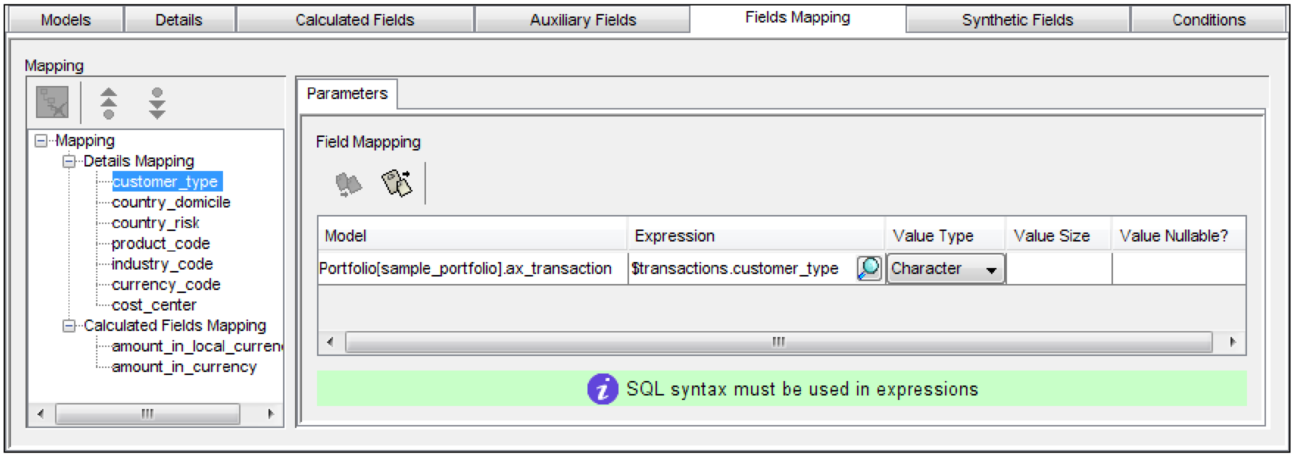
2. Created and named anew and manually mapped to the original source (sources).

3. Supplied with formulas for the additional calculations.

9.01.04 Field Mapping

In order for the Aggregation to function, the system needs to identify selected calculated and detail fields and map them to the corresponding fields in each Data Model associated with the task.

By default the system performs mapping automatically. Sometimes, however, the system fails to identify the appropriate column in the source and manual mapping needs to be performed. The sample mapping is shown on the illustration below (Fig. 57).



Model	Expression	Value Type	Value Size	Value Nullable?
Portfolio[sample_portfolio].ax_transaction	$transactions.customer_type	Character		

SQL syntax must be used in expressions

Fig. 57

Field Mapping Parameters Pane

Displays mapping detail for each field. Each row in the pane's grid represents a Data Model involved in the Aggregation.

Model

Displays names of models assigned to the current Aggregation.

Expression

Expression input line contain names of the source columns to which selected detail and calculated fields are to be mapped. As a rule the Mapping is done automatically. The system searches through the data models involved in the Aggregation task and identifies the columns that correspond to the selected details and calculation fields.

If the system fails to map a data attribute to a column in one or several data models, the corresponding cells will be left blank. In this case the mapping should be done either manually:

1. Click on 🔍 button to open the lookup window.

2. Select an appropriate column name from the lookup and click OK

Note: Users can modify the default mapping by selecting other column names from the lookup.

Mapping Expression

Normally Mapping Expressions are limited to names of the source columns to which the mapping is done. The system however allows you to fine-tune field mappings by writing SQL expressions. For example you may want to insert an expression referencing the appropriate columns for the currency conversion or Buy/Sell indicator selection. The lookup in this case will provide you with referenced column names preceded with the '$' sign.

Note: In order to appear in the Lookup, the referenced columns must be set up as the auxiliary fields (see below).

Value Type

This field becomes active when a Calculation formula is used. You are enabled to select the appropriate Data format from the arrow button menu (Fig. 58).

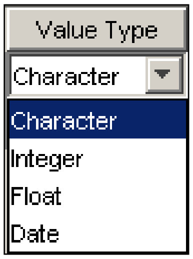


Fig. 58

Value Size

This field becomes active when a Calculation formula is used and Value Type is set on 'Character'. Input here indicates the number of characters the value in the field would NOT exceed.

Value Nullable?

This field becomes active when a Calculation formula is used. Check Mark here will allow nulls in the Calculated field.

9.01.05 Optional Steps

There are several optional steps in the Aggregation Setup that generate additional functionality and enhance the system’s performance.

Auxiliary Fields Setup

Auxiliary fields are fields from the model containing data that is used as part of an aggregation calculation but not appearing in the aggregation results.

For example, your model may have a field called "foreign gross amount" and another called "exchange rate" while the report requires the results in "Domestic gross amount" (that is expressed in domestic currency). In order to express the Foreign amount in the domestic currency you would multiply "foreign amount" by the "exchange rate".

If, at the same time, you don't want to show the "foreign gross amount" and "exchange rate" fields in your aggregation results you would:

1. Set them up as auxiliary fields into the Auxiliary Fields tab
2. On the Calculated Fields tab create a new field called "gross_domestic_amount” and as the formula enter "foreign gross amount" * "exchange rate".

Synthetic Fields Setup

The purpose of Synthetic Fields is to enable users to perform ‘on-the-fly’ arithmetical and logical operations on the **aggregation execution results** generating additional figures.

These fields are termed ‘Synthetic” because they exist only on the **viewer screens** and not on the database like the Additional Fields of the Portfolio.

To set up a Synthetic field, using the Exchange rate conversion example, you would:

1. Create a field in the Synthetic Fields tab and call it "domestic gross amount"
2. Use "foreign gross amount" * "exchange rate" expression as the formula for the calculation
3. Include "foreign gross amount" and "exchange rate" columns on the Calculated Fields tab

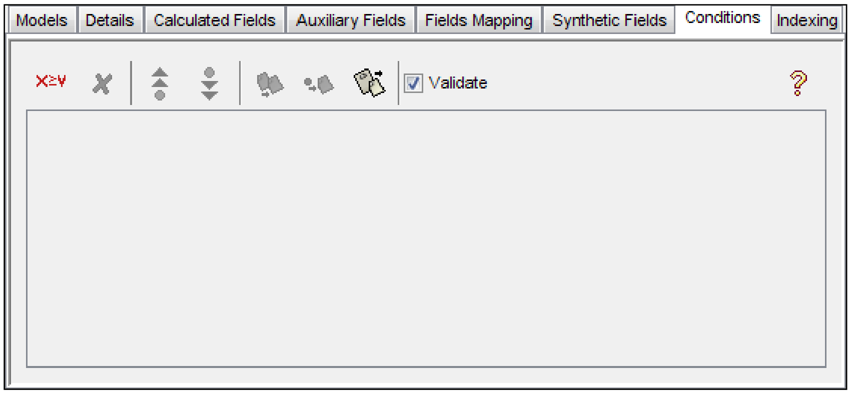
The results of the above setup will be that all three columns will be shown on the Aggregation Viewer.

Conditions

On the Conditions Tab (Fig. 59) you can insert logical conditions to be applied to the aggregated data. Conditions setup allows you to narrow down data for processing.

Insert Condition

Pressing this button and select a condition type from the menu (refer to the Logical Conditions section of this document).



☑ Validate

Fig. 59

Delete Conditions
Pressing this button removes the selected condition. The warning window will appear.

Move Up and Down Buttons
Although the “order of appearance” is irrelevant to the conditions processing, the Up and Down buttons are provided to arrange conditions in the most comprehensive way.

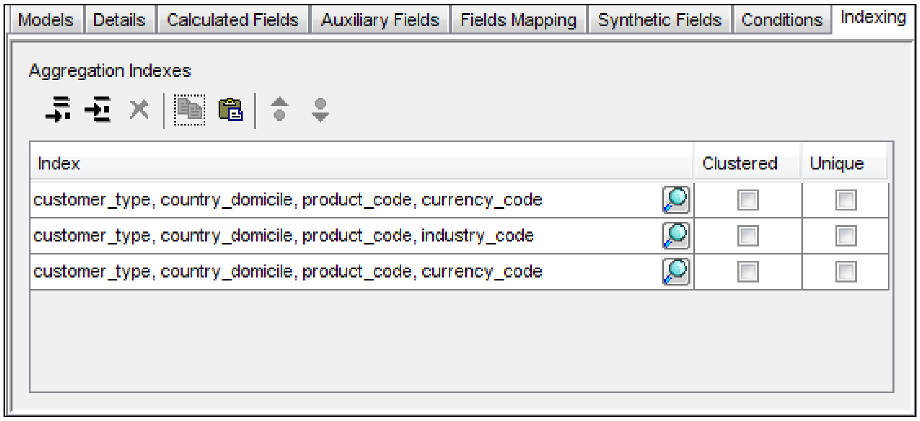
Copy All, Copy One and Paste Buttons
Copy and Paste function allow you to import conditions to other tasks.

☑ Validate **Validate Check Box**
Mark in this box will force the system to check if values you entered in the conditions actually exist. By default the system will reject non-existing values.

Note: If you need to enter a non-existent value i.e. infinite margin of a date interval, you must disable this feature.

9.01.06 Indexing

Indexing Tab (Fig. 60) is where you can create indices to optimize the Drill Down. Drill Down is the process in which you study summary numbers by breaking them down into the lower degrees of granularity with the transactional level being the lowest. Of course you can see the entire set of transactions at any moment, but you would be better off first narrowing down the data pinpointing your area of interest, which may be limited to a certain location, industry type or asset category.



Aggregation Indexes

Index	Clustered	Unique
customer_type, country_domicile, product_code, currency_code	□	□
customer_type, country_domicile, product_code, industry_code	□	□
customer_type, country_domicile, product_code, currency_code	□	□

Fig. 60

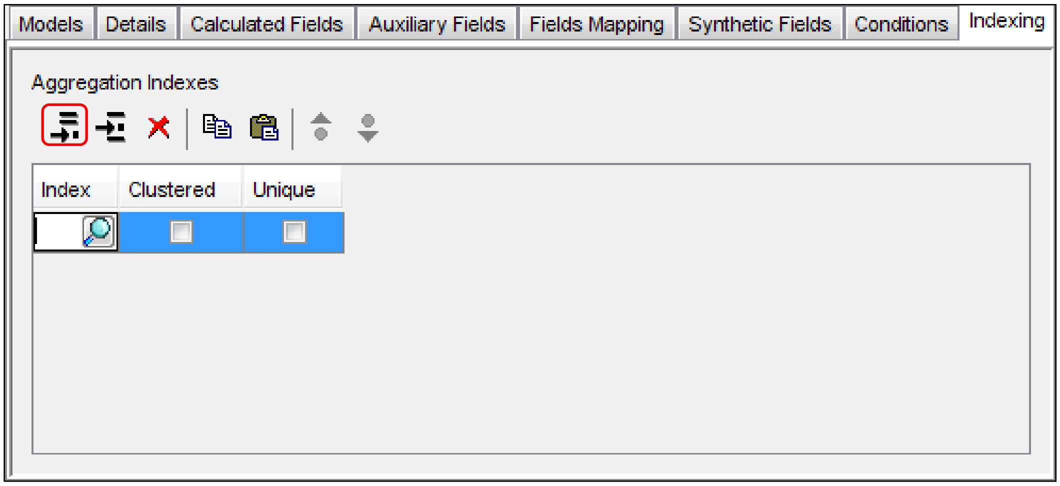
Indexing allows controlling the speed of viewing or extracting of the data. Users are encouraged to establish anticipated drill-down sequences that are likely to be employed most often and the system will generate indices accordingly. Properly created indices may speed up data viewing tens or even hundreds of times.

There is however the efficiency trade off. If you create too many indexing sequence combinations, time to generate the indices may exceed time of the aggregation run itself! Please consult your SA if you are not clear on how your database operates before you set out optimizing the drill down.

Indexing Setup

Assuming that you anticipate several possible paths the Drill Down would take (that is the sequence of the detail selection) you would create as many sequential indices arranging details in the anticipated order separating them with commas:

1. On the Index Tab grill press ⇥ to insert a row (Fig. 61).

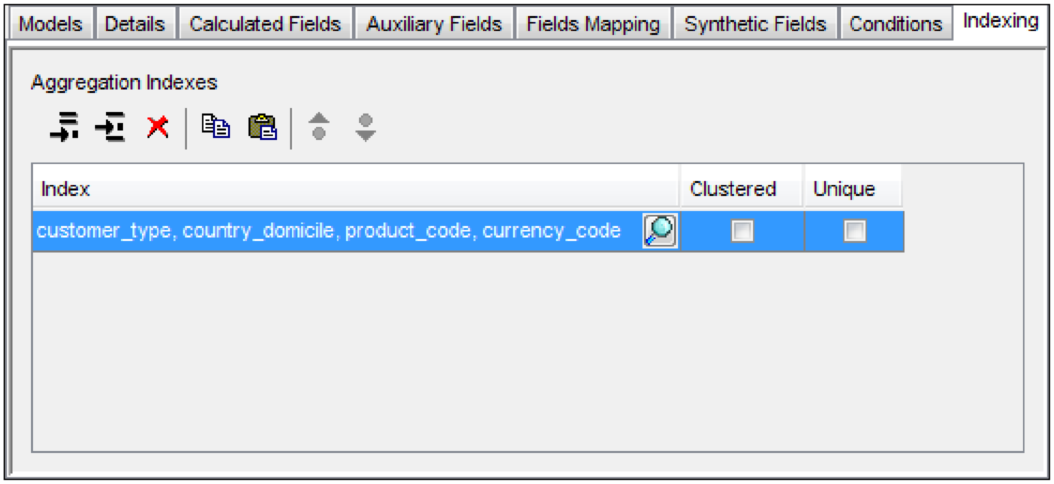


Index	Clustered	Unique
🔍	□	□

Fig. 61

2. Press 🔍 to open the Detail lookup.

3. Select details in an anticipated sequence (use Ctrl for multiple select) and press OK to close the Lookup. Selected details appear in the row separated by commas (Fig. 62).



Index	Clustered	Unique
customer_type, country_domicile, product_code, currency_code 🔍	□	□

Fig. 62

4. Set up the remaining parameters:

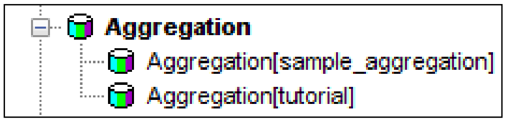
   - **Clustered** – This parameter allows you to determine where the Index will be stored. It is Database-specific and works differently with various DB types. Please consult your DBA prior to using this feature.

   - **Unique** – If checked, the Index will be regarded as ‘Unique’.

9.02 Saving and Executing

Before you can run an aggregation (*sample_aggregation* in our example), you must save it. The following occurs when you press the Save:

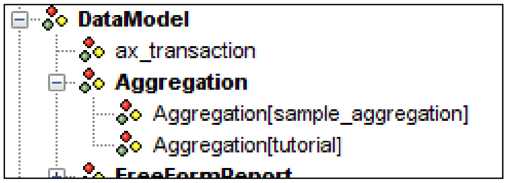
The system generates a new source that can be used for other executable tasks. This source can be found in the Objects Pane under the Aggregation category (Fig. 63).



Aggregation
- Aggregation[sample_aggregation]
- Aggregation[tutorial]

Fig. 63

The system generates a new Data Model. This Data Model can be found in the Objects Pane under the Aggregation category (Fig. 64).



DataModel
- ax_transaction
- **Aggregation**
  - Aggregation[sample_aggregation]
  - Aggregation[tutorial]
- **FreeFormReport**

Fig. 64

Once you have saved your work, click ▶ (Execute). Follow the same steps here as you did when you executed the Portfolio.

If the model is not there, click 🔍 (lookup) and select the model, change the date of the model instance if necessary.

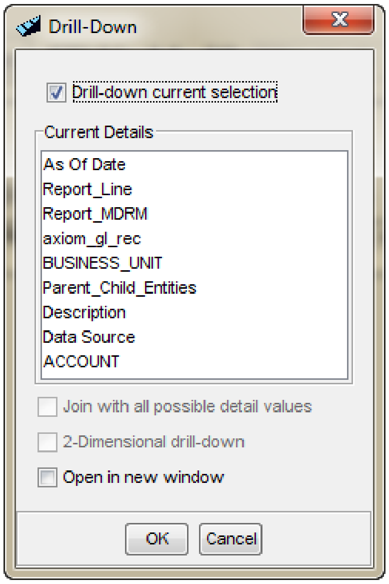
9.03 Viewing Aggregation Results

Results of Aggregation execution can be studied on the Aggregation Viewer screen. Aggregation Viewer has all standard features of AxiomSL’s data viewers.

Drill Down capability allows you to break down the numbers by one or several details al way to the transactional level.

To view Aggregation results:

1. Click 🔍 (View Aggregation). The Drill Down window opens automatically showing the list of details to select from for the initial Data breakdown (Fig. 65).



☑ Drill-down current selection

Current Details

- As Of Date
- Report_Line
- Report_MDRM
- axiom_gl_rec
- BUSINESS_UNIT
- Parent_Child_Entities
- Description
- Data Source
- ACCOUNT

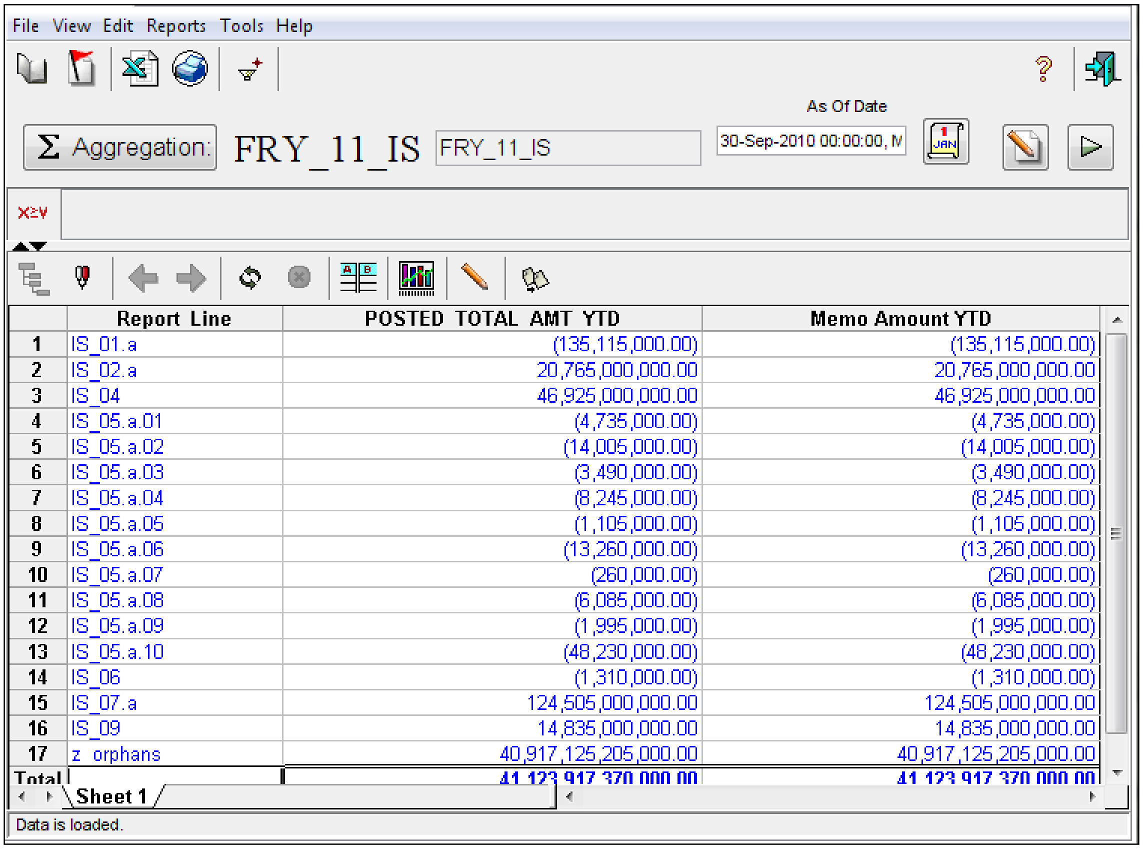
☐ Join with all possible detail values

☐ 2-Dimensional drill-down

☐ Open in new window

Fig. 65

2. Select one or several details (use Ctrl to do multiple select) and press OK. The screen redraws showing the data (Fig. 66).



FRY_11_IS FRY_11_IS

As Of Date: 30-Sep-2010 00:00:00, M

	Report Line	POSTED TOTAL AMT YTD	Memo Amount YTD
1	IS_01.a	(135,115,000.00)	(135,115,000.00)
2	IS_02.a	20,765,000,000.00	20,765,000,000.00
3	IS_04	46,925,000,000.00	46,925,000,000.00
4	IS_05.a.01	(4,735,000.00)	(4,735,000.00)
5	IS_05.a.02	(14,005,000.00)	(14,005,000.00)
6	IS_05.a.03	(3,490,000.00)	(3,490,000.00)
7	IS_05.a.04	(8,245,000.00)	(8,245,000.00)
8	IS_05.a.05	(1,105,000.00)	(1,105,000.00)
9	IS_05.a.06	(13,260,000.00)	(13,260,000.00)
10	IS_05.a.07	(260,000.00)	(260,000.00)
11	IS_05.a.08	(6,085,000.00)	(6,085,000.00)
12	IS_05.a.09	(1,995,000.00)	(1,995,000.00)
13	IS_05.a.10	(48,230,000.00)	(48,230,000.00)
14	IS_06	(1,310,000.00)	(1,310,000.00)
15	IS_07.a	124,505,000,000.00	124,505,000,000.00
16	IS_09	14,835,000,000.00	14,835,000,000.00
17	z_orphans	40,917,125,205,000.00	40,917,125,205,000.00
Total		41,123,917,370,000.00	41,123,917,370,000.00

Data is loaded.

Fig. 66

Selecting several details will give you a more rounded view of data at the initial breakdown allowing various options for the further drilldown.

Changing the Initial Detail Selection

To change the initial Detail selection with nothing selected in the Data Grid:

1. Press button of the grid's tool bar;

2. Select another combination of details from the Drill Down dialog;

If you have a row highlighted in the Viewer grid:

1. Press button;

2. Remove the check mark from the ☑ Drill-down current selection box of the Drill Down dialog;

3. Select another combination of details from the Drill Down dialog;

Alternatively you can accomplish the same result by drilling down the Totals:

1. Scroll down to Totals and highlight any cell in the Total row;

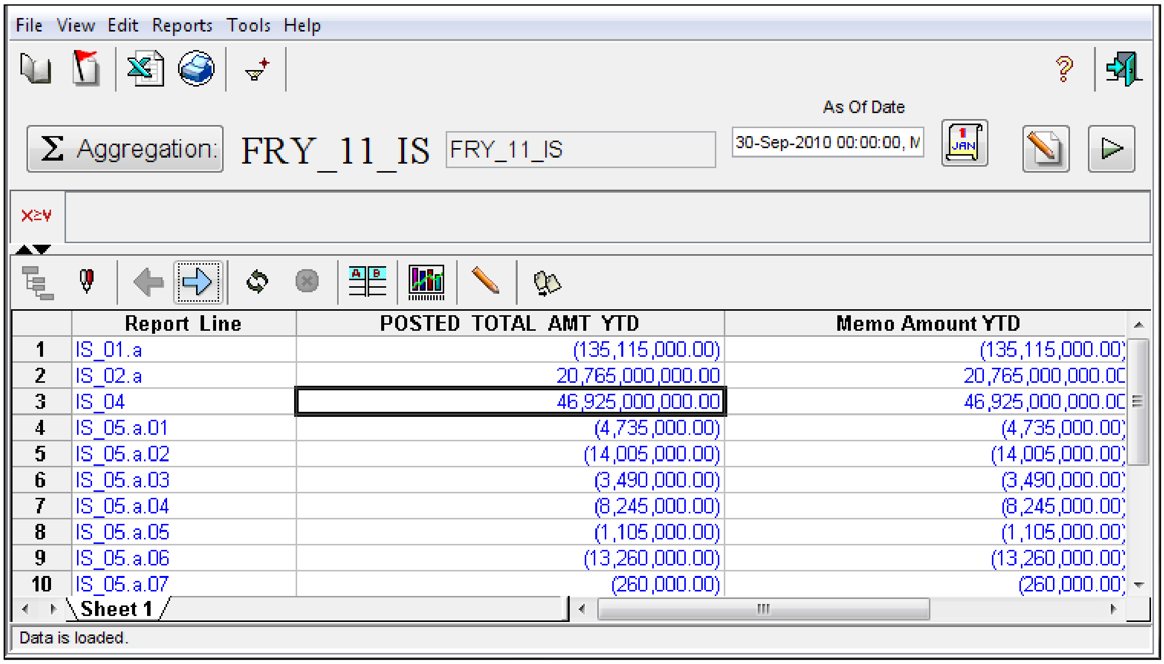
2. Click on the button. This will open the Detail Selector window.

3. Select another combination of details from the Drill Down dialog;

9.03.01 Starting Drilldown

Looking at the numbers you may want to explore a row in the grid and see how the Report Line IS_04 total breaks down by Business Unit. To do this:

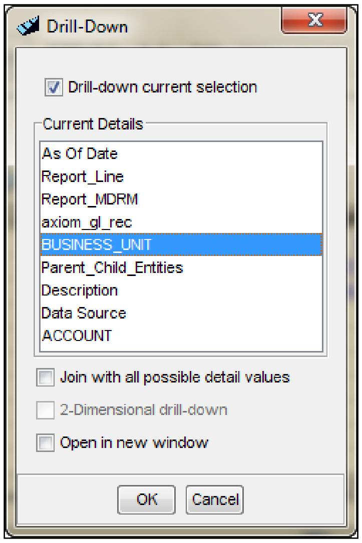
1. Click on any cell to drill down within the row of interest to select it (Fig. 67).



	Report Line	POSTED TOTAL AMT YTD	Memo Amount YTD
1	IS_01.a	(135,115,000.00)	(135,115,000.00)
2	IS_02.a	20,765,000,000.00	20,765,000,000.0
3	IS_04	46,925,000,000.00	46,925,000,000.0
4	IS_05.a.01	(4,735,000.00)	(4,735,000.00)
5	IS_05.a.02	(14,005,000.00)	(14,005,000.00)
6	IS_05.a.03	(3,490,000.00)	(3,490,000.00)
7	IS_05.a.04	(8,245,000.00)	(8,245,000.00)
8	IS_05.a.05	(1,105,000.00)	(1,105,000.00)
9	IS_05.a.06	(13,260,000.00)	(13,260,000.00)
10	IS_05.a.07	(260,000.00)	(260,000.00)

Fig. 67

2. Drill-down window will open (Fig. 68), from which you select criteria for displaying the result.



☑ Drill-down current selection

Current Details

- As Of Date
- Report_Line
- Report_MDRM
- axiom_gl_rec
- BUSINESS_UNIT
- Parent_Child_Entities
- Description
- Data Source
- ACCOUNT

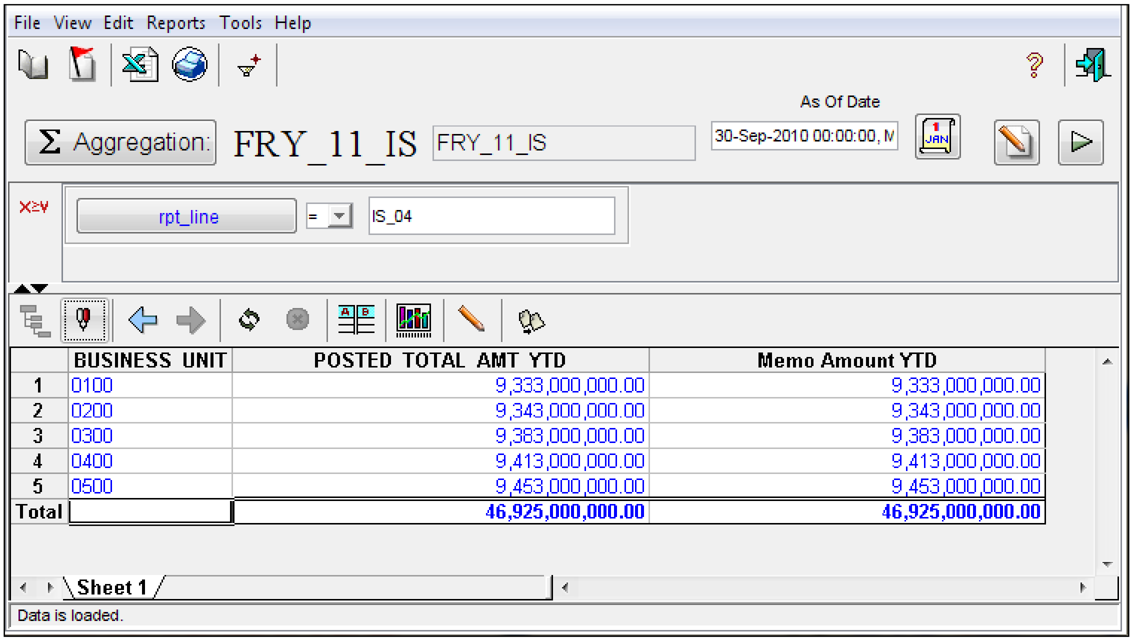
☐ Join with all possible detail values

☐ 2-Dimensional drill-down

☐ Open in new window

Fig. 68

3. In our example, we shall determine which Business Units make up the totals for the selected row. In the drill down window, select GL Account, GL Account Name and click OK. The Aggregation Viewer screen redraws showing the selected raw total broken down by the business_unit parameter (Fig. 69).



FRY_11_IS FRY_11_IS

As Of Date: 30-Sep-2010 00:00:00, M

rpt_line = IS_04

	BUSINESS UNIT	POSTED TOTAL AMT YTD	Memo Amount YTD
1	0100	9,333,000,000.00	9,333,000,000.00
2	0200	9,343,000,000.00	9,343,000,000.00
3	0300	9,383,000,000.00	9,383,000,000.00
4	0400	9,413,000,000.00	9,413,000,000.00
5	0500	9,453,000,000.00	9,453,000,000.00
Total		46,925,000,000.00	46,925,000,000.00

Data is loaded.

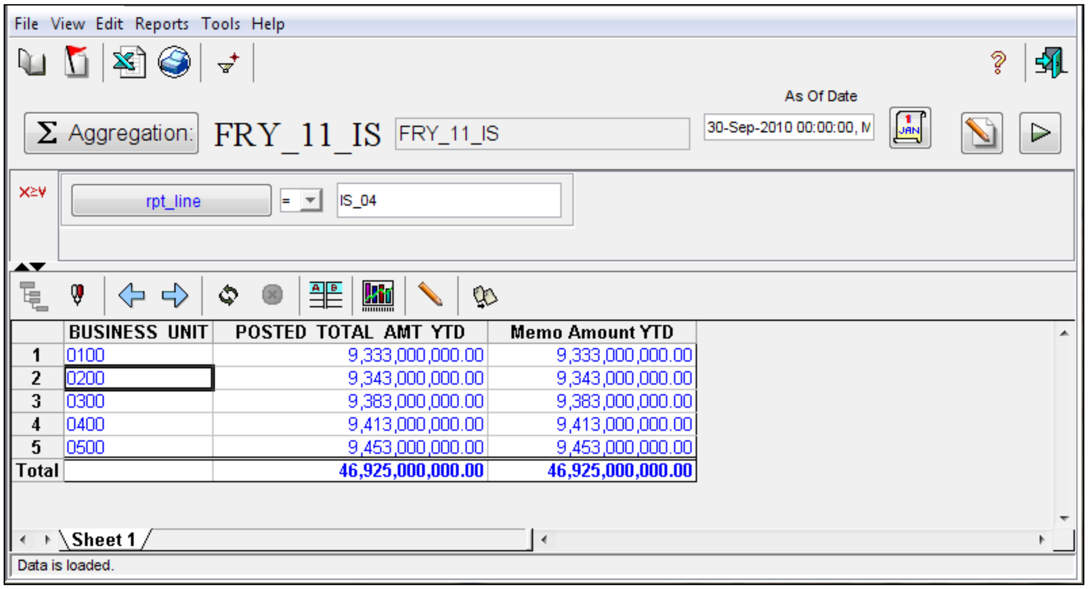
Fig. 69

Note: Into the Conditions Pane the system inserts a logical condition where the selected report line is specified.

Technically it is possible to use Logical Conditions pane to conduct your drill-down provided you know exactly what you want to see. In this case you would insert the condition into the Conditions Pane and select the required column and the value.

4. Next step in our drill-down would be to explore a particular Business Unit (0200 in our example) and break down the total of this Business Unit by Account.

5. Select a cell in the grid (Fig. 70).



FRY_11_IS FRY_11_IS

As Of Date: 30-Sep-2010 00:00:00, M

rpt_line = IS_04

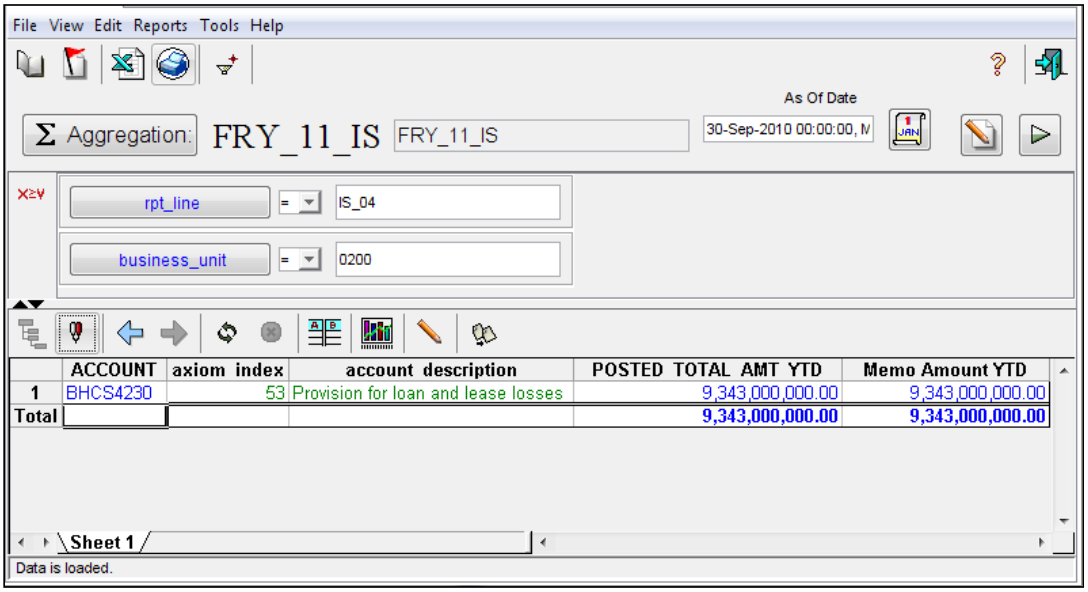
	BUSINESS UNIT	POSTED TOTAL AMT YTD	Memo Amount YTD
1	0100	9,333,000,000.00	9,333,000,000.00
2	0200	9,343,000,000.00	9,343,000,000.00
3	0300	9,383,000,000.00	9,383,000,000.00
4	0400	9,413,000,000.00	9,413,000,000.00
5	0500	9,453,000,000.00	9,453,000,000.00
Total		46,925,000,000.00	46,925,000,000.00

Data is loaded.

Fig. 70

6. The Drill-down window will open.

7. Select ACCOUNT and press OK. The Viewer redraws showing the selected Business Unit Total broken down by ACCOUNT (Fig. 71).



FRY_11_IS FRY_11_IS

As Of Date: 30-Sep-2010 00:00:00, M

rpt_line = IS_04

business_unit = 0200

	ACCOUNT	axiom index	account description	POSTED TOTAL AMT YTD	Memo Amount YTD
1	BHCS4230	53	Provision for loan and lease losses	9,343,000,000.00	9,343,000,000.00
Total				9,343,000,000.00	9,343,000,000.00

Data is loaded.

Fig. 71

At any point of the drill-down you can:

1. Use ⇦ ⇨ buttons to move back and forth along the drill-down path.

2. Use button to save the desired drill-down level as Favorite. In doing so you create a short cut to the particular level of drill-down to which you will be able to return later on to monitor the dynamics of a point if interest.

3. Use Filter button to view the current drill-down position on the transactional level (see below).

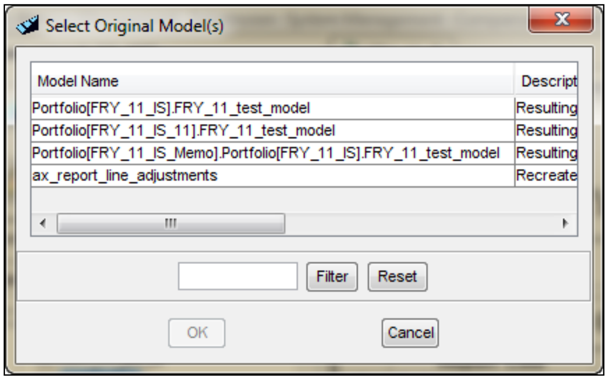
9.03.02 Filter

Filter function allows you to access the transactional data relevant to the current drill-down level. In our example, let us assume that you want to see what transactions are associated with ACCOUNT as shown on Fig. 71.

To do this:

1. Select an ACCOUNT row in the grid (in this particular case, since there is only one entry, selecting the Totals row will produce the same effect).

2. Press button. The Data Model lookup opens showing data models that are associated with the current Aggregation (Fig. 72).

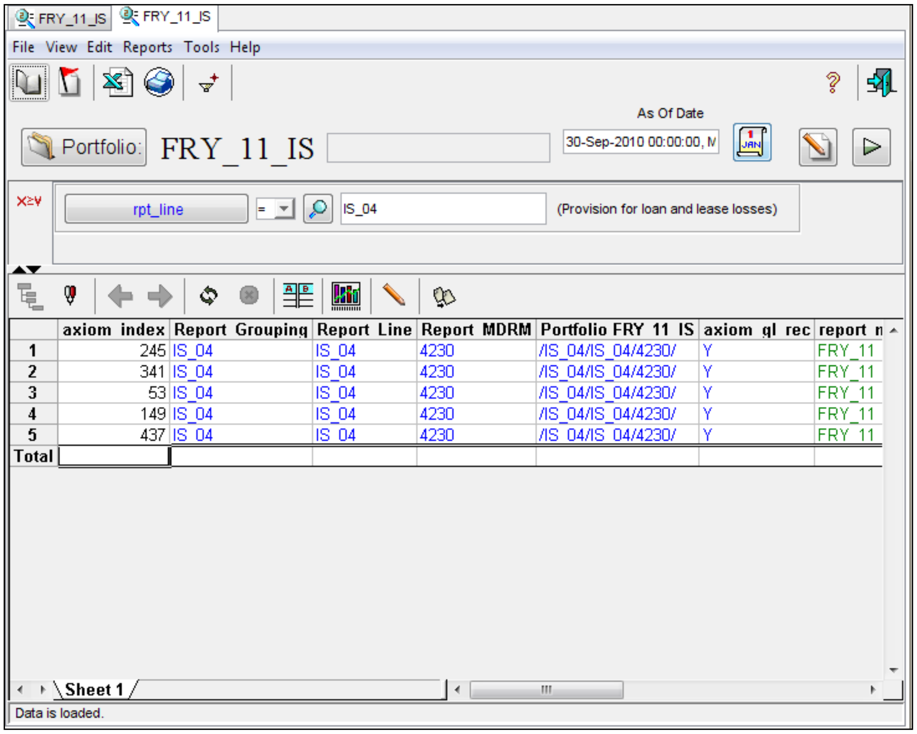


Select Original Model(s)

Model Name	Descript
Portfolio[FRY_11_IS].FRY_11_test_model	Resulting
Portfolio[FRY_11_IS_11].FRY_11_test_model	Resulting
Portfolio[FRY_11_IS_Memo].Portfolio[FRY_11_IS].FRY_11_test_model	Resulting
ax_report_line_adjustments	Recreate

Fig. 72

3. Select a model that contains the source with required transactions (The Data Model resulting from the FRY_11 Portfolio execution in our example) and press OK. The new Viewer window opens with the Main Source of the selected Data Model transactions pertaining to the current drill down level (Fig. 73).



Portfolio: FRY_11_IS

As Of Date 30-Sep-2010 00:00:00, M

rpt_line = IS_04 (Provision for loan and lease losses)

	axiom index	Report Grouping	Report Line	Report MDRM	Portfolio FRY 11 IS	axiom gl rec	report n
1	245	IS_04	IS_04	4230	/IS_04/IS_04/4230/	Y	FRY_11
2	341	IS_04	IS_04	4230	/IS_04/IS_04/4230/	Y	FRY_11
3	53	IS_04	IS_04	4230	/IS_04/IS_04/4230/	Y	FRY_11
4	149	IS_04	IS_04	4230	/IS_04/IS_04/4230/	Y	FRY_11
5	437	IS 04	IS 04	4230	/IS 04/IS 04/4230/	Y	FRY 11
Total							

Data is loaded.

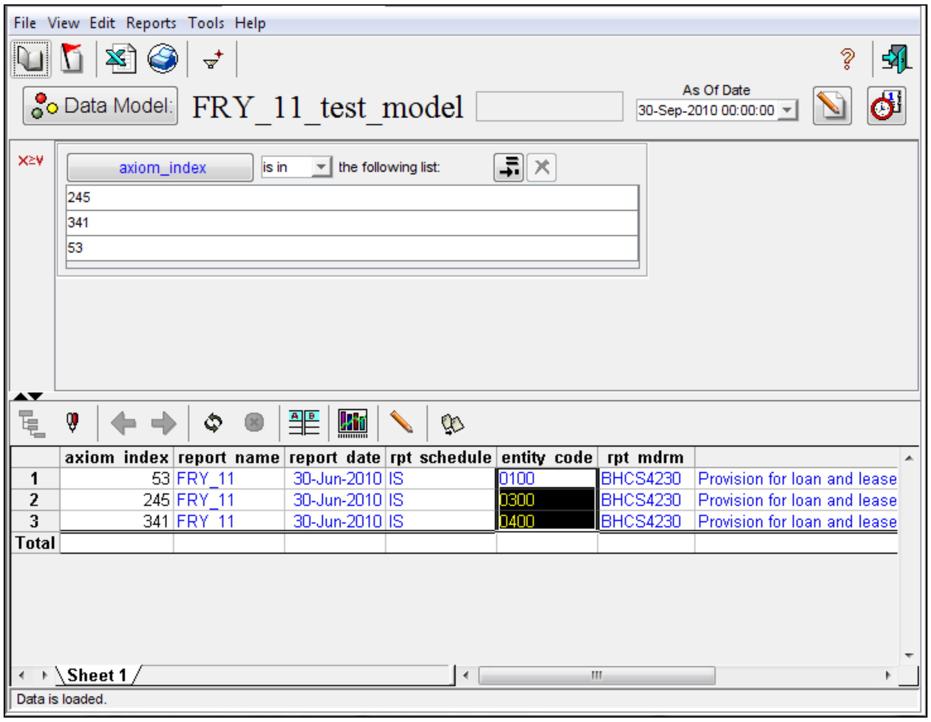
Fig. 73

Let us assume further that upon examining the data on display here you found errors and would like to make adjustments to several records. Technically, you could do it from here. But the appropriate way to do this, would be correcting the records at the original source. To do this:

1. Select records you would like to correct (Axiom Index 245, 341 and 53 in our example).

2. Press the Filter Button. The Model lookup will open displaying data models on which the Portfolio was based.

3. Select the model that contains the original source and press OK to close the lookup. A new Viewer screen opens containing the selected records (Fig. 74).

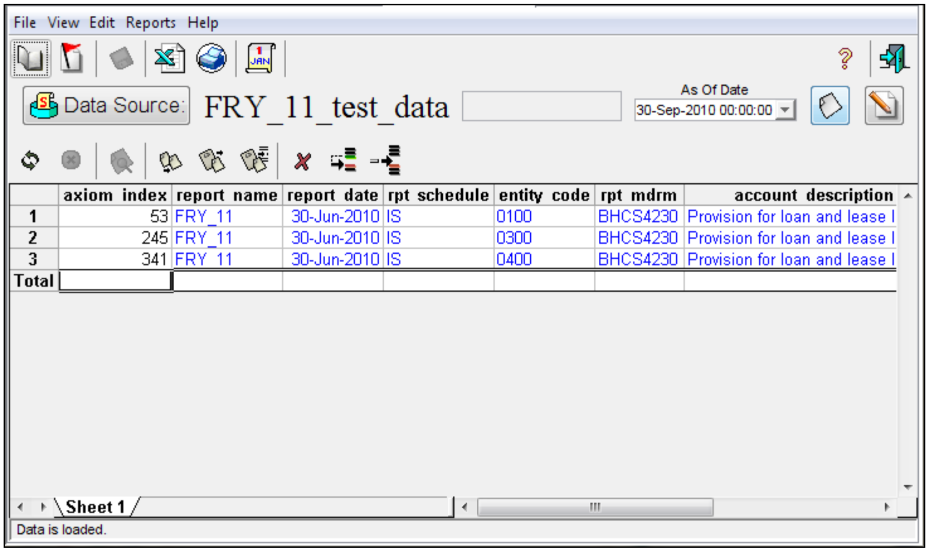


245
341
53

	axiom index	report name	report date	rpt schedule	entity code	rpt mdrm	
1	53	FRY_11	30-Jun-2010	IS	0100	BHCS4230	Provision for loan and lease
2	245	FRY_11	30-Jun-2010	IS	0300	BHCS4230	Provision for loan and lease
3	341	FRY_11	30-Jun-2010	IS	0400	BHCS4230	Provision for loan and lease
Total							

Fig. 74

4. Select all records in the grid and press Adjustments button of the Data Grid's Tool bar. The Adjustments screen opens containing the selected records (Fig. 75).

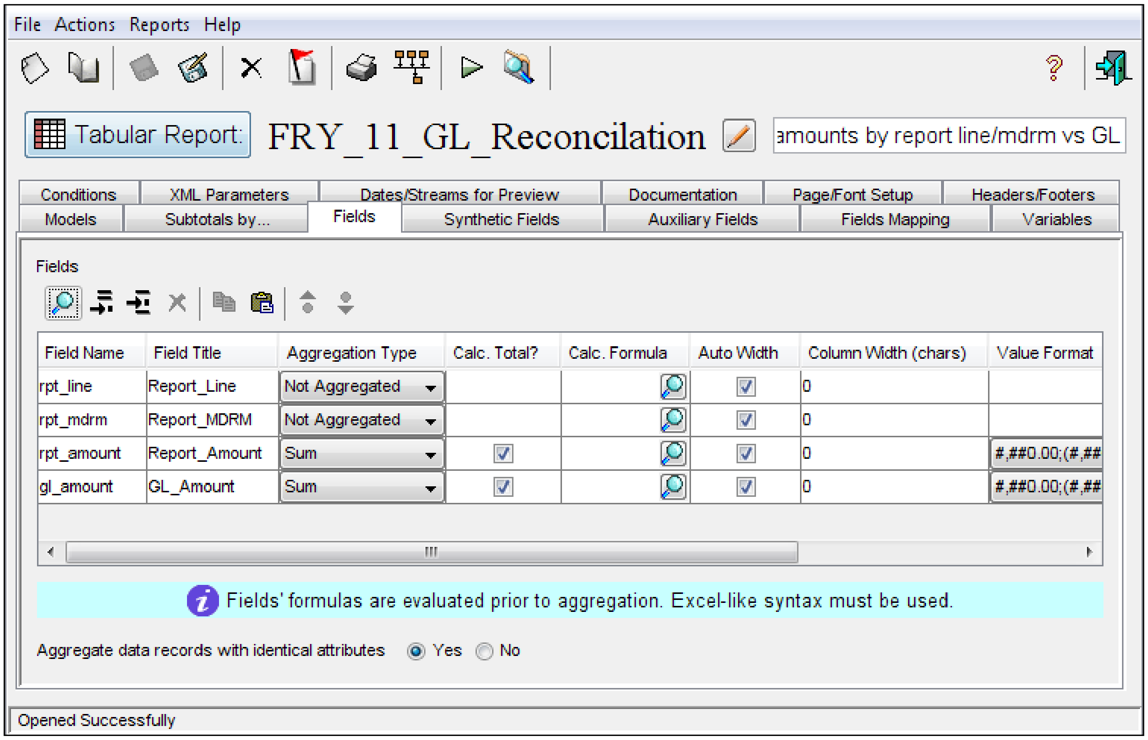


	axiom index	report name	report date	rpt schedule	entity code	rpt mdrm	account description
1	53	FRY_11	30-Jun-2010	IS	0100	BHCS4230	Provision for loan and lease l
2	245	FRY_11	30-Jun-2010	IS	0300	BHCS4230	Provision for loan and lease l
3	341	FRY_11	30-Jun-2010	IS	0400	BHCS4230	Provision for loan and lease l
Total							

Fig. 75

10. TABULAR REPORT WRITER

Tabular report writer (Fig. 76) allows you to construct customized tabular style reports based on any data model(s). Its other main function is that it can produce 4 files simultaneously during the report execution. Formats supported are Text, Excel, PDF and XML.



Field Name	Field Title	Aggregation Type	Calc. Total?	Calc. Formula	Auto Width	Column Width (chars)	Value Format
rpt_line	Report_Line	Not Aggregated			☑	0	
rpt_mdrm	Report_MDRM	Not Aggregated			☑	0	
rpt_amount	Report_Amount	Sum	☑		☑	0	#,##0.00;(#,##
gl_amount	GL_Amount	Sum	☑		☑	0	#,##0.00;(#,##

Fields' formulas are evaluated prior to aggregation. Excel-like syntax must be used.

Aggregate data records with identical attributes ● Yes ○ No

Opened Successfully

Fig. 76

10.01 Tabular Report Writer Screen

Tabular Report Writer screen consists of the series of tabs where you can setup various parameters that are essential for the Tabular Report to function.

10.01.01 Models Tab

On the Models tab you select data models containing sources of data for your report. Selected Data models may include original segmented or continuous sources, data models generated in the course of execution of the Portfolios and/or Aggregations or a combination of the above.

10.01.02 Subtotals Tab

On this tab you select fields for which you want the system to summarize data when compiling the report. The summaries will be calculated and displayed for every combination of the selected details.

10.01.03 Fields Tab

On the Fields tab you determine what data will be shown on the printout of the report. Obviously, there will be columns containing numerical information that reflects the purpose of the report. The system allows you however to include any other field available from the main source table or detail set on which the report is based. The data in the selected fields will be displayed throughout the report for every combination of the Detail field selected on the Subtotals tab.

Duplicating Details and Display Fields
Normally the system will not allow you to use the same detail fields(s) for display in your report.
To overcome this you have to do following:

1. On either Subtotals or Fields tab, use Add New button to insert the new column
2. Use any name to it, but different from the name of the field you are intending to duplicate.
3. Map this field to the same source column as the field you are duplicating.

10.01.04 Mapping Tab

Although the initial field mapping is done automatically, sometimes, especially in multiple source environment or/and when Add New button was used when entering Details or Display fields, the system fails to identify the appropriate field in the source leaving the field in the Fields Mapping blank. In this case you would have to do mapping manually by using Column Lookup.

The system also allows you to modify the initial mapping by replacing system-selected column names with other column names from the lookup.

10.01.05 Auxiliary Fields Tab

These are fields from the model that that contain data that is used in calculation to generate or modify data in the report’s columns.

For example, your model may have a field called "foreign gross amount" and another called "exchange rate" while the report requires the results in "Domestic gross amount" (that is expressed in domestic currency). In order to express the Foreign amount in the domestic currency you would multiply "foreign amount" by "exchange rate".

To accomplish this you would:

1. Set up *foreign_gross_amount* and *exchange_rate* as auxiliary fields.
2. On the Fields tab create a new field called "gross_domestic_amount” and as the formula enter "foreign gross amount" * "exchange rate"

10.01.06 Conditions Tab

The system allows you to narrow down the data in your report to the desired degree by using logical conditions. In the Conditions tab you can stipulate logical conditions for data extraction.

10.01.07 Synthetic Fields Tab

The purpose of Synthetic Fields is to enable users **to perform ‘on-the-fly’ arithmetical and logical operations** in a newly created column. These fields are named ‘Synthetic” because they exist **only when the task is opened** and not on the database like Additional Fields of the Portfolio.

For a Synthetic Field to function it requires a name, title and the calculation formula (Value Expression). For instance, if there are *exposures* expressed in the local currency and the exchange rates are provided, you can convert exposures into the home currency.

If, depending on the currency type, user needs to perform different calculations (i.e. division vs. multiplication) he may use logical conditions (if).

Fields in the formula get referenced by name, prefaced with the “$” (for example, formula for converting the exposure into the home currency will look like *$exposure_in_local_currency * $exchange_rate*.

Note: Both referenced fields should be included in the Execute for Fields Tab setup.

Note: The system supports MS Excel syntax for calculations and conditions.

To set up Synthetic fields, using an example, you would:

4. Create a field in the Synthetic Fields tab and call it "domestic gross amount"

5. Use "foreign gross amount" * "exchange rate" expression as the formula for the calculation

6. Include "foreign gross amount" and "exchange rate" columns on the fields tab

Note: Synthetic fields are displayed on the report only and data there is not stored in the Database. The values are inserted into the synthetic fields during the report execution.

10.01.08 Headers/Footers Tab

On this tab you specify contents and location of the report features. The working area of the tab is broken into the input fields for every feature of the report:

1. Header (1st page)
2. Header (subsequent pages)
3. Footer
4. Report Title

Into each input field you enter information specific to the purpose of the report.

Each input area, i.e. Header, Footer, etc. is provided with button that accesses the menu listing pre-set macros for the command to the system to insert necessary information. You can combine manual data entry with macros to create comprehensive “cover” for the report.

You can also adjust spacing and alignment of the information, as it should appear on the report.

10.02 Generating Tabular Report

After the report setup is complete you may go to the Report Execution screen (Fig. 76) and type the resulting file names followed by the “.” and the extension, depending on the type of file(s) you want to generate.

The extensions supported by the system are:

1. “.txt” for text file

2. “.xls” for excel files

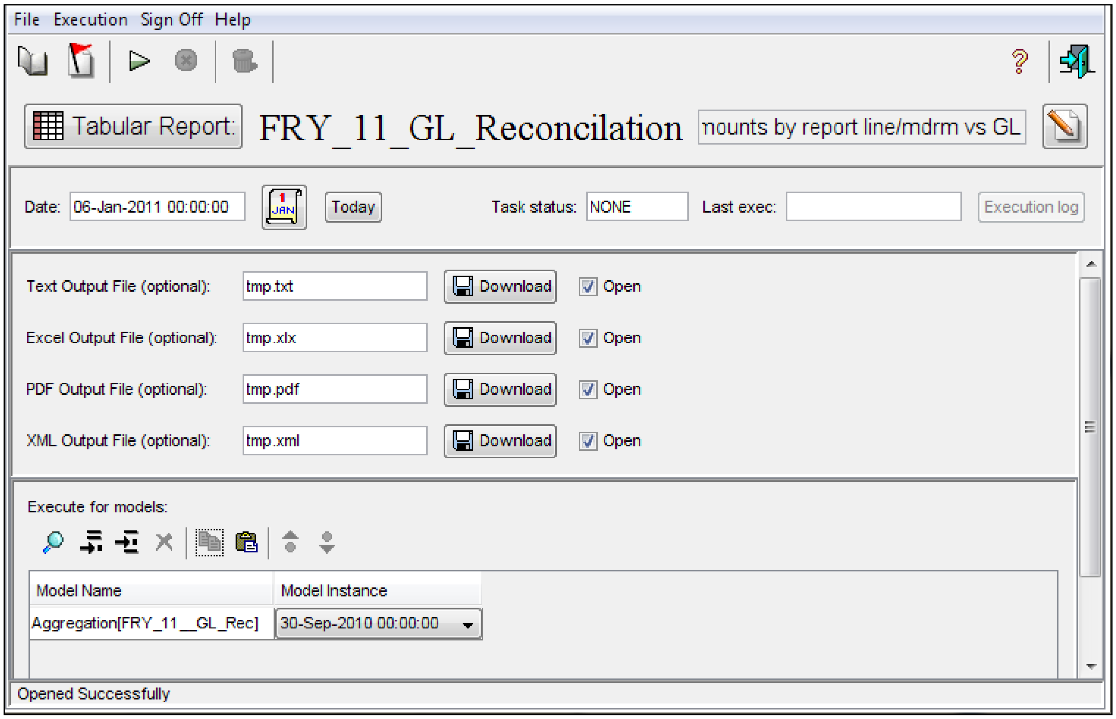
3. “.pdf” for Adobe portable document format files

4. “.xml” extensible markup language.

After generating the files you may press download to remove them from the server and save them locally on your hard disk. Also you may automatically open them after download by clicking on the open check box prior to pressing download.

10.02.01 Execute for Models Pane

Execute for Models pane is located in the bottom of the Execution screen (Fig. 76). Here you can select which data models and for what date you want to provide the report data.



FRY_11_GL_Reconcilation | nounts by report line/mdrm vs GL

Date: 06-Jan-2011 00:00:00 | Today | Task status: NONE | Last exec:

Text Output File (optional):	tmp.txt	Download	☑ Open
Excel Output File (optional):	tmp.xlx	Download	☑ Open
PDF Output File (optional):	tmp.pdf	Download	☑ Open
XML Output File (optional):	tmp.xml	Download	☑ Open

Model Name	Model Instance
Aggregation[FRY_11__GL_Rec]	30-Sep-2010 00:00:00

Fig. 76

As you press ▷ button, the Task Status will change to START and remain so until the execution is finished. When the task is successfully completed the status will change to ACTIVE.

11. FREE FORM REPORT

Freeform Report Writer allows you to define and execute reports of the enterprise’s transactional activity following any format required by regulators. The subsystem, designed following MS Excel standards, is able to tap into the Data Warehouse and extract information from multiple data models residing there.

It also allows users to export reports into the Excel file and back (some syntax adjustment may be required).

While preserving the Excel‘s capabilities, Report Writer provides several additional functions enabling users to:

1. Extract data from multiple sources including results of Portfolio/Aggregation calculations
2. Perform drill down of the report’s numbers
3. Refer to other reports

These functions are:

1. RESULT FROM DATABASE
2. FROM REPORT

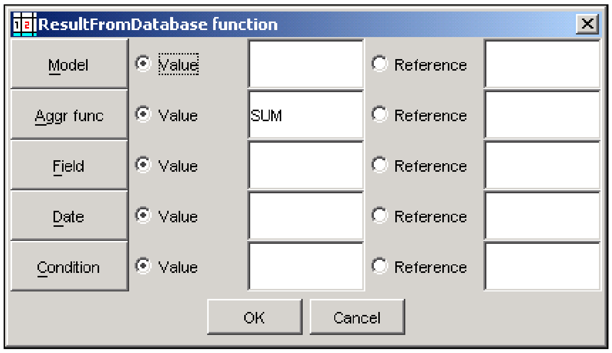
ResultFromDatabase

This function is a multi-argument function with the following syntax:

=ResultFromDatabase(“data_model_name”,”aggregation_function”,aggregated_field”,”data_model_date”, ”condition_1”,”condition_2”,….”condition_n”).

This string can be entered manually into the formula bar or you can use Result From Database Editor screen equipped with lookups to make it easier to enter specific values.

ResultFromDatabase Editor (Fig. 77) is accessible through button of the spreadsheet template tab’s tool bar (or by selecting Insert DB Data option from the Insert pull down menu).



Model	● Value		○ Reference	
Aggr func	● Value	SUM	○ Reference	
Field	● Value		○ Reference	
Date	● Value		○ Reference	
Condition	● Value		○ Reference	

Fig. 77

ResultFromDatabase Editor

ResultFromDatabase Editor allows you to specify precisely what data from the Data Warehouse should appear in the given cell. Parameters may include all or any combination of Data model, type of aggregation, aggregated field, date and a condition.

When entering database parameters you have option to click Value and look up the names, using the corresponding Lookup Button, or refer to another cell of the spreadsheet by clicking on Reference and entering the target cell's name or reference.

FromReport Function

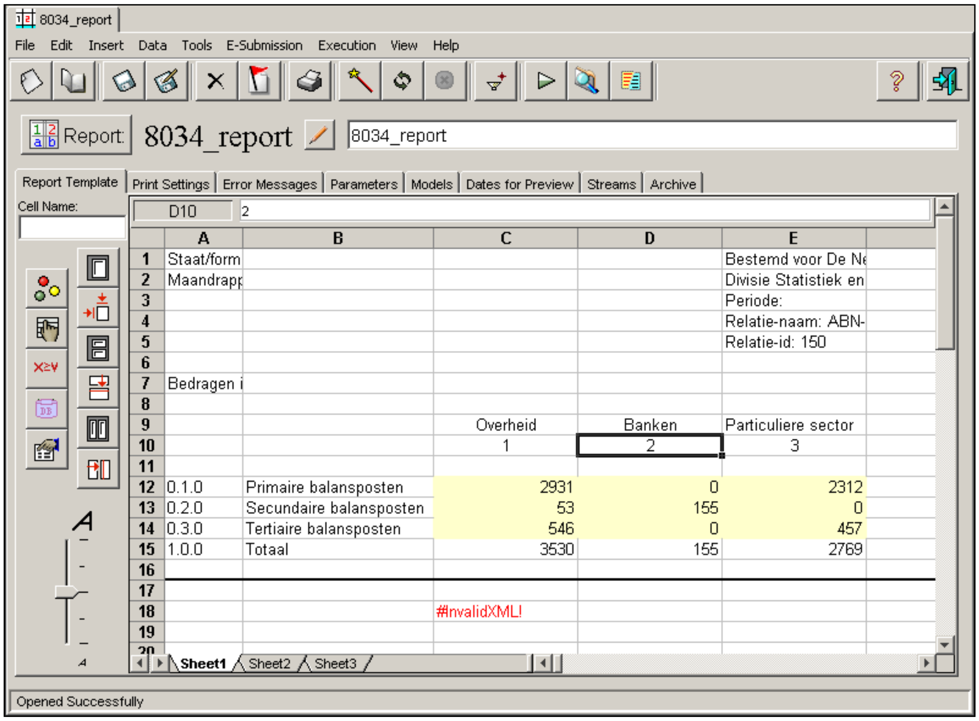
FromReport function allows you to reference other reports stored in the Data Warehouse. The function is entered directly into the formula bar. The syntax is as following:

=FromReport("report_name","cell_name")

Note: The referenced report must be open (in another window) in order for this function to work.

11.01 Creating Freeform Report

Below are the basic steps involved in preparing reports with the Freeform Report Writer (Fig. 78):



D10 2

	A	B	C	D	E
1	Staat/form				Bestemd voor De Ne
2	Maandrapp				Divisie Statistiek en
3					Periode:
4					Relatie-naam: ABN-
5					Relatie-id: 150
6					
7	Bedragen i				
8					
9			Overheid	Banken	Particuliere sector
10			1	2	3
11					
12	0.1.0	Primaire balansposten	2931	0	2312
13	0.2.0	Secundaire balansposten	53	155	0
14	0.3.0	Tertiaire balansposten	546	0	457
15	1.0.0	Totaal	3530	155	2769
16					
17					
18			#InvalidXML!		
19					

Fig. 78

1. The base of all reports is an Excel form (a template) with text, color, graphics, numeric fields (boxes or cells to be later filled in with results from database), formulas for subtotals calculations, etc.

The template can be built either in Report Writer screen, which emulates the Excel's functionality, or in Excel directly. It can be obtained from the regulatory authority in Excel format (if one is available).

Note: If the report template is built as an Excel file, it must be imported into Report Writer by using "Import Excel File" functionality.

2. As most reports are date-dependent, a single cell must be designated in the report spreadsheet to hold the value of the reporting date. If date information appears in multiple places throughout

the report (e.g. on front page, or on each page), all date fields must refer to the single designated date cell.

The date values *entered* into the date cell must be in mm/dd/yy or mm/dd/yyyy format. However, the display format of the date cell can be chosen in any manner to suit your needs (e.g. 01-January-2008, or dd/mm/yy, or anything else).

Note: Use "Format active cell" functionality to set this display format.

Once the date cell is chosen, you must enter the name or a reference of the date cell into the parameter "As of date cell in spreadsheet" on the "Parameters" tab of the Report Writer screen. For example, you would enter 'Page 1'!A1.

3. Populating the numeric fields in report with results from database.

Most of the regulatory reports require the *amounts* to be rounded to a nearest integer (whole number) and/or reported in thousands, or millions. Because of this, the data retrieved from the database needs to be transformed before it is inserted into the numeric fields on the report.

The database holds amounts as floating-point numbers – the amounts in a given currency, e.g. 1000.50 for 1 thousand dollars and 50 cents. Therefore, if the report requires the amounts to be displayed in thousands, the following rounding is necessary (using the standard excel function Round):

= Round(...amount.../1000, 0)

The recommended way to build such a report is the following:

a) Choose the cells outside of the printable area on the worksheet, which you will use to retrieve values from the database. For example, if the cells C10, D10, E10 are the numeric boxes in the report form, and columns starting from F are all unused (don't contain any text or other data which should be printed with the report form), you can choose cells G10, H10, I10 to retrieve corresponding values from database.

b) Insert the rounding formulas into the numeric boxes on the form. For example:

In cell C10 you would put

= Round(G10/1000, 0)

In cell D10 –

= Round(H10/1000, 0), and so on.

c) Insert =ResultFromDatabase(...) functions into the designated data cells G10, H10, I10, so that these cells are populated with values from database. (See step 4 below for more details).

This way the designated data cells will contain the original non-truncated amounts, and the printable cells in the report (C10, D10, E10) will contain the amounts converted to thousands.

If your report does not require rounding, the values retrieved from database can be put directly into numeric fields on the report spreadsheet. E.g., you would put =ResultFromDatabase(...) functions directly into cells C10, D10, E10 and then set the appropriate display format on these cells, using "Format active cell" functionality.

4. Bringing data from Data Warehouse's database into the Report Writer.

These are the most common sources of data for the reports:

- ***Transactional data*** from the original source
- ***Portfolio execution*** results
- ***Aggregation execution*** results

The Aggregation execution results are the preferable source of data for the Freeform Report Writer, because of its following properties:

- Fast rate of data retrieval - Aggregation results are the fastest data source for querying, because the transactional data is already pre-aggregated.

- Ability for drill-down - the amounts displayed on the spreadsheet in Report Writer can be drilled-down by any of the user-defined criteria. All you have to do is add the ***details*** that you want, in the ***Aggregation Setup*** screen.

If transactional data or Portfolio results are used as a source of data in the Report Writer, the drill-down functionality only allows you to view the full list of transactions that comprise the amount displayed on the spreadsheet.

It is always easy to create ***Aggregations*** just for the sake of using them in the Freeform Report Writer:

Take your transactional data, or Portfolio execution results, which you want to report on and create a new Aggregation in the ***Aggregation Setup*** screen.

Add the ***data model*** representing your transactional or Portfolio data to the Models tab.

Add the numeric column (the column in the data model that contains the amount to be shown in Report Writer) to the ***Calculated Fields*** tab.

Add the ***details*** which you want to drill-down by. You must not forget to add as *details* the columns in the data model which will be involved in ***conditions*** restricting transactional or portfolio data when calculating the aggregated amount for the report.

We will focus here on using Aggregation results in Report Writer. If you want to use any other data, then the ***ResultFromDatabase Editor*** screen will let you customize your data retrieval procedure.

There is an easy way to bring in the value from the Aggregation result into a cell in Report Writer spreadsheet, using the ***Aggregation Viewer***:

Open the Aggregation in the Aggregation Viewer screen, perform a desired drilldown, highlight the ***amount*** cell (the calculated field) for the item you want to bring into the report, and click "Copy to clipboard for Report Writer" button, which is the right-most button on the Data Grid tool bar. This will copy the necessary ***ResultFromDatabase()*** formula to the clipboard. You can then go back to the Report Writer screen, and paste the formula into the desired cell on the spreadsheet. Following this, the value will be retrieved from database and will appear in the cell.

If you need to restrict your data using ***logical conditions*** (e.g. your item on the report requires the sum total of all non-USD amounts), insert these conditions in the Aggregation Viewer screen.

In this particular example, the condition will be

Single-value condition
[currency_column] != USD

After you have inserted the conditions and refreshed the Aggregation view, repeat the copy/paste operation described above.
This copy/paste approach is very convenient, but has one important limitation. To be able to use it, the Aggregation data must contain all the values required on the report, and that may not always be the case.

There is a second approach, which can be used if a report is based on data from ***Portfolio Results*** or an Aggregation built on Portfolio Results.

You will still need to use ***ResultFromDatabase Editor*** in the Report Writer screen, or manually enter ResultFromDatabase() formula, if you are familiar with its syntax.

However, the hardest part - entering the ***logical condition*** can be done semi-automatically.

The steps are as follows:

1. Highlight the cell in report spreadsheet (Report Writer screen) where you want to insert the data from database

2. Click the purple "DB" button to open up ***ResultFromDatabase Editor*** for the selected cell

3. Click the "Model" button on the ResultFromDatabase Editor screen, and choose the ***data model*** from which you want to retrieve the data. If an Aggregation Result is used (as is recommended), the data model name will be **dbs**__[Aggregation Name]

4. The "Aggregated Function" field value can be left at "SUM", if you are reporting the summaries.

5. Click the "Field" button on the ResultFromDatabase Editor screen, and choose the numeric column containing the ***amount***, which you want to show on the report.

6. Click the "Date" button to choose the date (instance date of the data model you have just selected) of the data you want to use in the report. Or, alternatively, click the "Reference" radio button and enter the reference to the ***date cell*** created in step 2. It is always better to use a *date cell* by reference, rather than explicit (hard-coded) constant date values.

7. Open up ***Portfolio Definition*** screen (from the main menu), and open the Portfolio you will be using in the report (e.g. Portfolio on which the Aggregation is built). On Portfolio Definition screen, go to "Tools" menu, and click "**Generate Conditions for Aggregation**".

8. On ***Portfolio Definition*** screen, highlight the portfolio node, which you want to include on report in Report Writer. For example, if your portfolio has nodes

   --- External Liabilities

   a. +--- Banks

   b. +--- Non-Banks

   and the item on the report, which you are filling in, is "external liabilities to banks", highlight the "Banks" node. All transactions which fall under "Banks" category in your portfolio will be summarized and the sum total will appear in the cell on the report.

9. On ***Portfolio Definition*** screen, click the button "Copy to clipboard conditions identifying the node" (the last button of the Main Tool Bar before the help button).

10. Go back to Report Writer's ***ResultFromDatabase Editor*** screen. Click in the "Condition" "Value" input field, and paste the condition, using Shift-INS or Control-V keys.

11. Close ResultFromDatabase Editor screen by pressing OK. The selected cell in the Report Writer spreadsheet will contain the aggregated data from selected node in Portfolio Results.

When entering ResultFromDatabase() formula manually, the condition generated in Portfolio Definition screen can also be used. The easiest way is then to paste the condition into a separate cell on the spreadsheet, and then use a reference to that cell as the argument of ResultFromDatabase() function.
It is not recommended that you paste the condition directly as an argument of the ResultFromDatabase() function, because the pasted text will contain quotation marks, which need to be properly *escaped*.

5. Replacing constant date values with references to a *date cell*.

The ***ResultFromDatabase()*** formulas created in step 4 may contain constant date values embedded in them. It will especially be the case if formulas are obtained by copying/pasting from Aggregation Viewer. The date values are the instance dates of the underlying ***data models*** from which the data is queried into the Report Writer.

The constant date values will be an obstacle to running the report for different Reporting Dates. Therefore, all constant date values must be replaced with references to designated *date cell* created in step 2. Then, the *date cell* will be automatically or manually modified each time a report is run for a new Reporting Date.
Please use "Parameterize As Of Date Values" function from the "Data" menu in Report Writer screen to automatically perform this replacement.

You will be asked to specify the cell to hold the date value. For that, enter the *absolute* cell reference to the *date cell* created in step 2.

11.02 Report Writer Performance Optimization

When you click on the Magic Wand tool, the optimization screen opens (Fig. 79) listing queries.
The goal of optimization is to minimize the number of queries, while making sure that you don't end up with one huge query (which will also take a long time to execute).

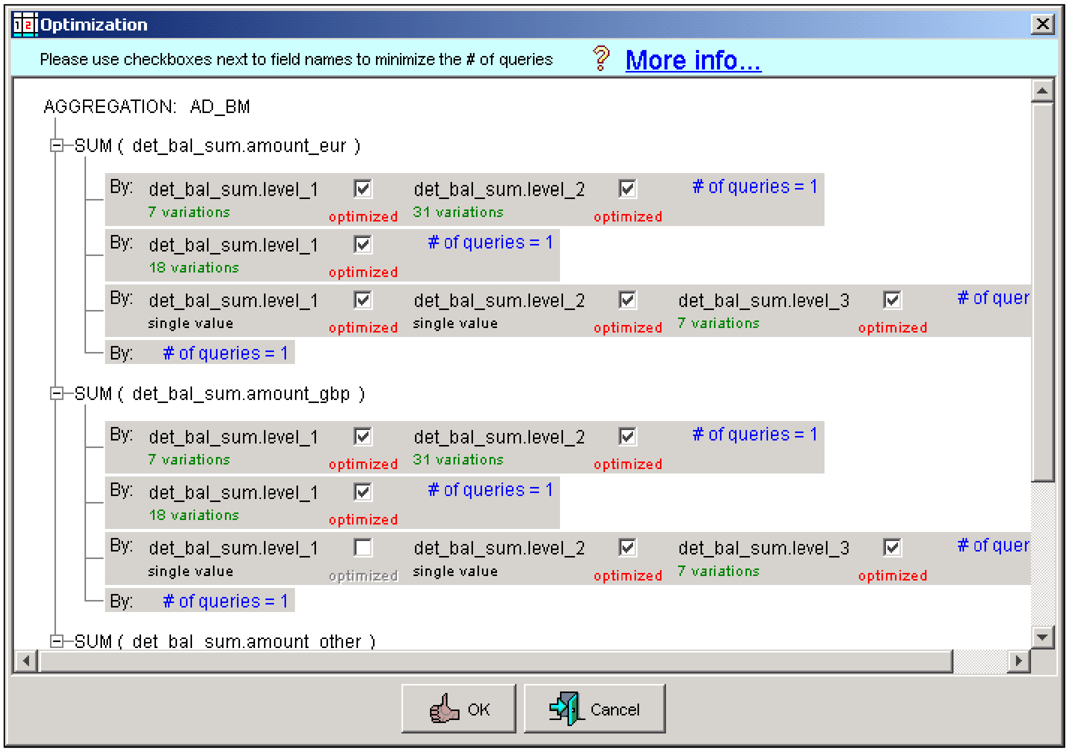
The optimization screen displays all data queries used in the current report. This information is organized hierarchically:

1. By *data model* (from which the data is retrieved)

2. By *calculated field* (a field from the data model whose aggregated value is displayed in the report)

3. By combination of *drill-down fields* (fields from the data model on which conditions are imposed, in order to aggregate the *calculated field*)

Next to each combination *of drill-down fields* the number of *queries* is displayed. This is the number of times the data has to be retrieved from the database in order to display values in the report. This number needs to be *optimized*, because each such query is time-consuming.



Optimization

Please use checkboxes next to field names to minimize the # of queries

AGGREGATION: AD_BM

SUM (det_bal_sum.amount_eur)

- By: det_bal_sum.level_1 ☑ optimized (7 variations) det_bal_sum.level_2 ☑ optimized (31 variations) # of queries = 1
- By: det_bal_sum.level_1 ☑ optimized (18 variations) # of queries = 1
- By: det_bal_sum.level_1 ☑ optimized (single value) det_bal_sum.level_2 ☑ optimized (single value) det_bal_sum.level_3 ☑ optimized (7 variations) # of quer
- By: # of queries = 1

SUM (det_bal_sum.amount_gbp)

- By: det_bal_sum.level_1 ☑ optimized (7 variations) det_bal_sum.level_2 ☑ optimized (31 variations) # of queries = 1
- By: det_bal_sum.level_1 ☑ optimized (18 variations) # of queries = 1
- By: det_bal_sum.level_1 □ optimized (single value) det_bal_sum.level_2 ☑ optimized (single value) det_bal_sum.level_3 ☑ optimized (7 variations) # of quer
- By: # of queries = 1

SUM (det_bal_sum.amount_other)

Fig. 79

When optimizing, you should use as a guide the *number of variations* of each field among *drill-down fields.* For example, 2 variations of field `country` means that only 2 different countries are used in the report. If the *number of variations* is large (e.g. 300 different countries are used), it makes sense to *optimize* the `country` field, by *checking the box next to it*. This would reduce the corresponding number of queries by 300 (at least).

You should avoid other extremes as well. For example, the `country` field has 2 *variations* in the report, but the underlying data covers 1000 different countries. If you choose to optimize this field, you will end up running a large query most of which will be useless (because you will use only 2 countries out of 1000). So it may be a good idea **not to optimize fields with small number of variations** (< 5).

Generally it makes no sense to optimize the filed, which is not varied (e.g. only one value is used). Such fields will be displayed for your information.

11.02.01 How the Optimization Works

When you choose the fields to be optimized (e.g. country), the Report Writer loads aggregated values of *calculated field* for all available values of the *optimized fields* (e.g. all countries) into memory, and later chooses specific values of these fields from memory, rather than from database.

When some of the *drill-down fields* are optimized, while others are not, the Report Writer runs a query for each relevant combination of *non-optimized* fields. Each query encompasses all values of the *optimized fields.*

Testing the Optimization

You can test the optimization after it is updated (the optimization window is closed) by pressing the *refresh* button on the Report Writer screen and taking notice of the time it takes to load all the data.

12. WORKFLOW AUTOMATION SUBSYSTEM

Workflow automation allows users to combine various system tasks into a continuous processes. These processes may be set up to perform automatically following the pre-set schedule or on-demand.

Setting up automated workflows allow users to avoid tedious executions of the tasks one by one risking committing errors i.e. omitting or misplacing tasks in the sequence.

The visual aspect of the workflow also allows users to grasp the logic of the process at a glance.

NOTE: Clients can use the Scheduler module provided by Axiom SL with the system or they can use their own scheduling system if they have one in place.

The subsystem consists of two modules:

1. Workflow Setup
2. Workflow Execution

12.01 Workflow Setup

Each task in the Workflow is represented by an object (icon) (Fig. 80) used as a building block for the workflow hierarchies.

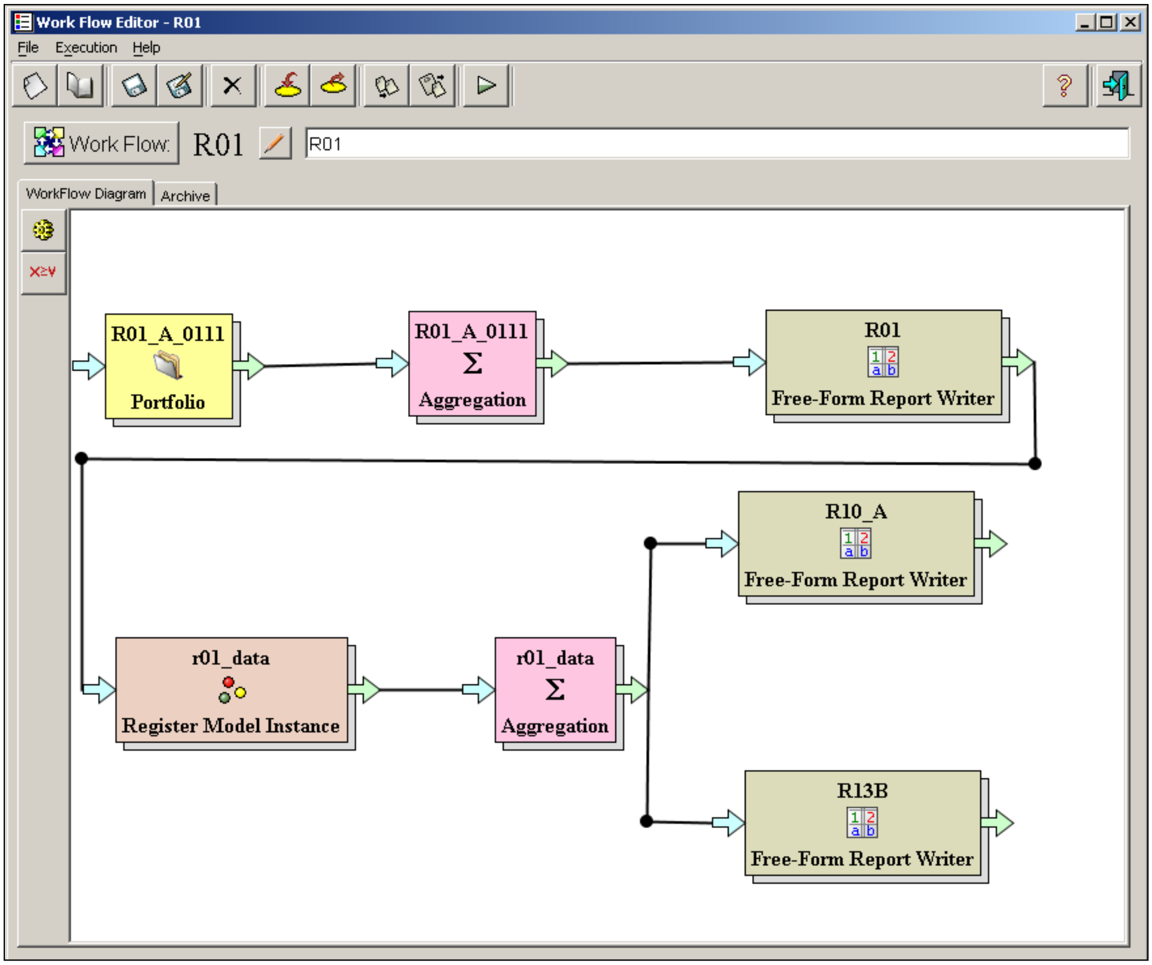


Fig. 80

The ⚙ button opens the menu of the available tasks. Selecting a task will turn the cursor into a "Insert" icon. Placing the cursor on the working area and clicking will *place* the selected object into which you will be able to enter additional task-specifying parameters.

The x≥y button inserts the logical condition object (Fig. 81) allowing you to specify a conditional expression that allows you to redirect the flow depending on the outcome of the previous task.

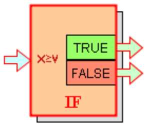


Fig. 81

The objects are connected by arrows forming strings following the logic of the process. Arrows can merge or branch out forming complex structures that may begin with data pre-processing and end with distribution of the final report.

Individual workflows can also be saved as units and incorporated into other workflows.

Workflow samples:
Below follow examples of workflow types (Figs 82, 83, 84):

Simple Workflow: A single string workflow designed to perform a straight forward sequence.

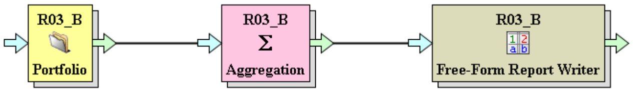


Fig. 82

Compound Workflow: Multiple string workflow designed to consolidate several independent tasks

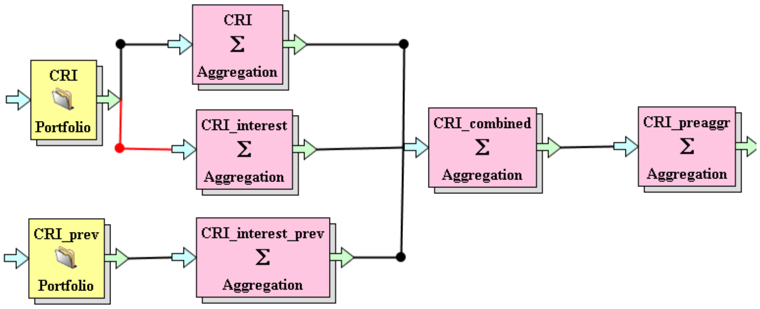


Fig. 83

Compound workflow built with other workflows as building blocks.

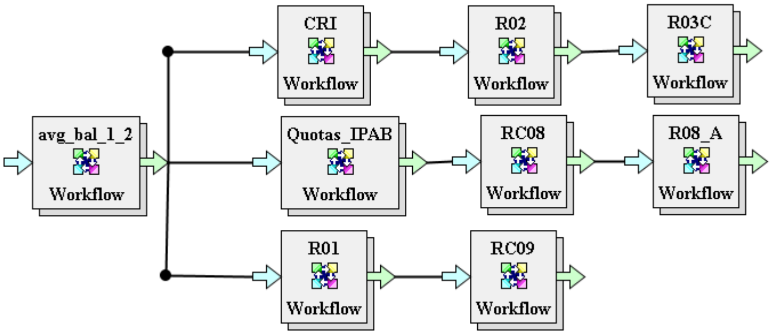


Fig. 84

12.02 Executing Workflow

To start Workflow execution you can open the Workflow Execution screen from the main menu and select the workflow from the lookup or press Execute button of the Main Tool Bar of the Work Flow Editor.
The system indicates that the process is in progress by following:

1. Task Status change from NONE (if the Workflow has never been executed for the current date) to START.
2. The ▶ appears next to the first executable
3. The title bar of the object changes color from gray (for never executed) to yellow.

When the system completes execution of the first task, the object's title bar color changes to green as the next task starts running.

When the Workflow is finished, all objects should have green title bars and ACTIVE should appear in the Task Status input field.

If the task fails, the object's title bar changes color to red and ❗ icon appears next to the object representing failed task. The whole process stops and the Task Status field displays ERROR message. You have to fix the problem and rerun the workflow skipping, if you want, successfully executed sections

- ⚠ icon indicates WARNING and DOES NOT prevent the Workflow run from the completion.

13. MIGRATING OBJECTS AND DATA

Migrating is a term used when we want to take data from one place and move it into another. We can move a whole branch or elements of that branch.

13.01 Migrating Branch (General migration)

To migrate an entire branch from one project to another:

1. Click on the “File” menu and select the Export option. (Fig. 85)

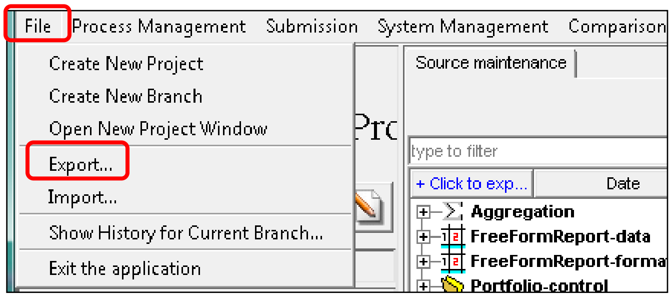
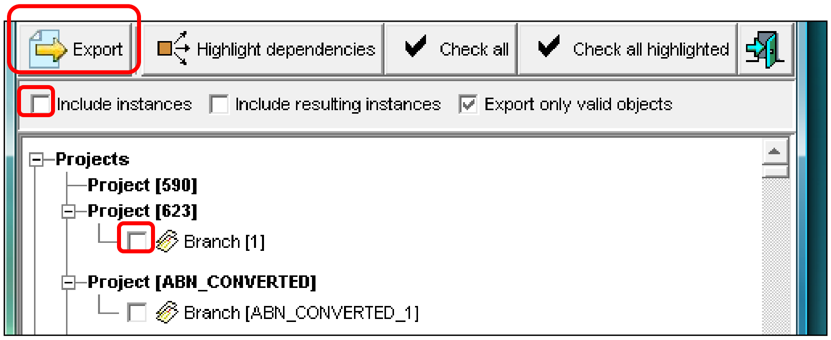


Fig. 85

2. A screen with all the projects and branches will come up (Fig. 86).



□ Include instances □ Include resulting instances ☑ Export only valid objects

Fig. 86

3. Check the box next to the branch you wish to migrate and click on the “Export” button.

4. Save the file in a specified folder. You can also export the instances of the branch, however, the data in that instance will not be exported. To do this, check the box next to “Include instances”.

Note: Remember to specify file format (zip vs. xml). Branches are zip.

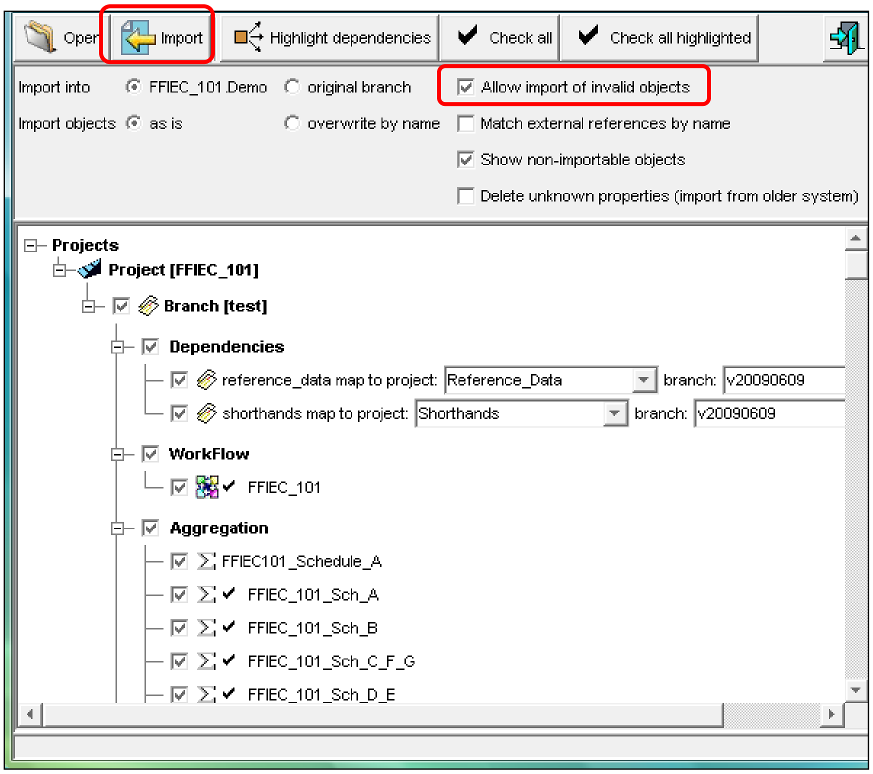
To import the exported data into a new branch:

1. Open the branch you want the data to migrate into.

2. Then click on the “File” menu and select the “Import” option.

3. Open the exported file. The Import screen will appear (Fig. 87).

4. Check the box next to branch (you can also select one or several components of the branch, if you do not wish to import everything)



Import into ● FFIEC_101 Demo ○ original branch ☑ Allow import of invalid objects

Import objects ● as is ○ overwrite by name □ Match external references by name

☑ Show non-importable objects

□ Delete unknown properties (import from older system)

- Projects
  - Project [FFIEC_101]
    - ☑ Branch [test]
      - ☑ Dependencies
        - ☑ reference_data map to project: Reference_Data branch: v20090609
        - ☑ shorthands map to project: Shorthands branch: v20090609
      - ☑ WorkFlow
        - ☑ ✓ FFIEC_101
      - ☑ Aggregation
        - ☑ FFIEC101_Schedule_A
        - ☑ ✓ FFIEC_101_Sch_A
        - ☑ ✓ FFIEC_101_Sch_B
        - ☑ ✓ FFIEC_101_Sch_C_F_G
        - ☑ ✓ FFIEC_101_Sch_D_E

Fig. 87

5. Check the box next to “Allow import of invalid objects”

6. Click the “Import” button.

13.02 Migrating a Single Object

To migrate a single object, right click on the icon of the object you wish to export and select “Export” from the menu (Fig. 88).

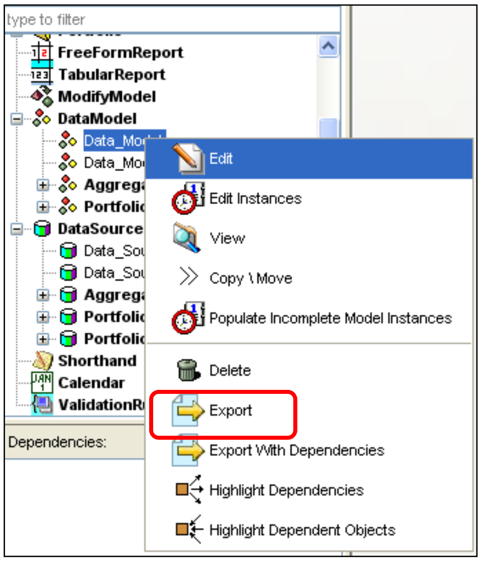


Fig. 88

Save the file in a specified folder. This imports only that object, not the data, dependencies, or instances that the object may contain.

Note: Remember to specify file format (zip vs. xml). Single objects are xml.

To import the exported object into a new branch

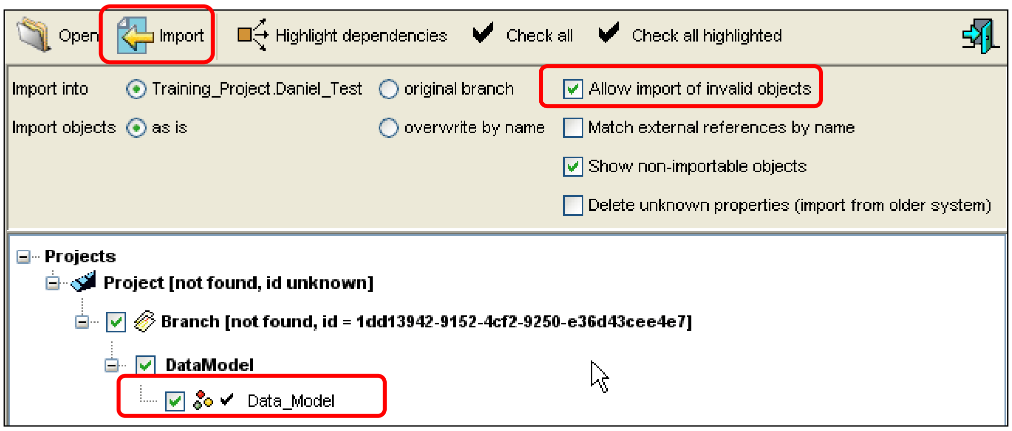
1. Open the branch you want the data to be imported into.

2. Click on the “File” menu and select the “Import” option. The Import screen will appear (Fig. 89).

3. Check the box next to the object in the data grid

4. Check the box next to “Allow import of invalid objects

5. Click “Import”.



Import into ● Training_Project.Daniel_Test ○ original branch ☑ Allow import of invalid objects

Import objects ● as is ○ overwrite by name □ Match external references by name

☑ Show non-importable objects

□ Delete unknown properties (import from older system)

Projects
- Project [not found, id unknown]
  - ☑ Branch [not found, id = 1dd13942-9152-4cf2-9250-e36d43cee4e7]
    - ☑ DataModel
      - ☑ ✓ Data_Model

Fig. 89

Note: When importing one object, checking the object will check all the previous boxes as well.

13.03 Migrating Instances

Let us assume we need to move an instance of the **ax_transactions** source to another project’s branch. To migrate a source instance:

Click the “Process Management” menu and select the “Source Maintenance” option (Fig. 90). The

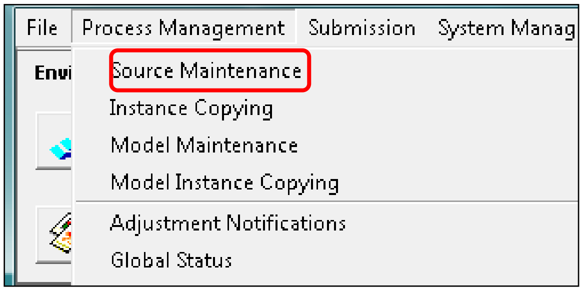


Fig. 90

Click on the “+” icon next to the source object you want to export the instance from. This will open all the available instances for that object.

Note: It is important to note that one object can have multiple instances, so be sure you are selecting the correct one.

Right click the instance and then click “Export (Internal with data, External without data)” (Fig. 91). In our example we are exporting the instance of “ax_gl_account” for 31-Dec-2010.

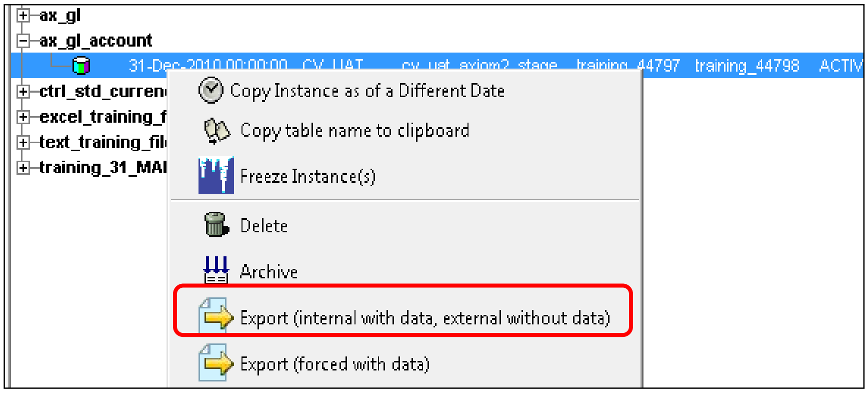

ax_gl
ax_gl_account
31-Dec-2010 00:00:00 CV_UAT cv_uat_axiom2_stage training_44797 training_44798 ACTIV
ctrl_std_curren
excel_training_f
text_training_fil
training_31_MAI
Copy Instance as of a Different Date
Copy table name to clipboard
Freeze Instance(s)
Delete
Archive
Export (internal with data, external without data)
Export (forced with data)

Fig. 91

To import an instance, follow the same steps as for importing an object. Click on the “File” tab and then click “Import”. Check the box next to the object, check the box next to “Allow import of invalid objects, and then click “Import”.

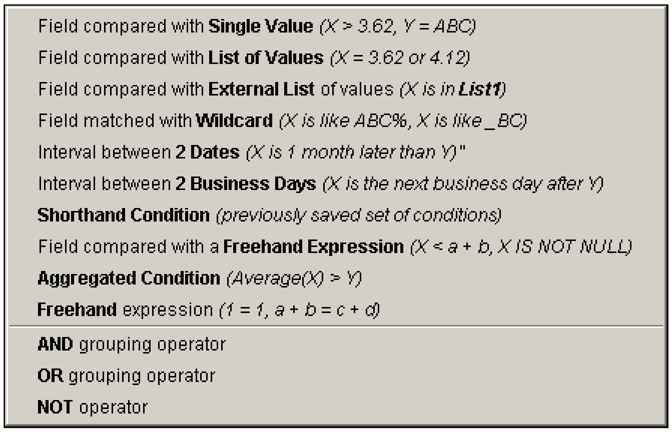
Appendix 1

Logical Conditions

The purpose of Logical conditions is to stipulate the criteria for records selection in the drill down process by applying Logical Operators to the specified data.

Available types of Logical conditions are listed on the menu (Fig. A1) of the Insert Condition button located on the Conditions pane of the following screens:

1. Portfolio Definition
2. Shorthand Manager
3. Viewers
4. Report Writer Filter
5. Filter



Field compared with **Single Value** *(X > 3.62, Y = ABC)*

Field compared with **List of Values** *(X = 3.62 or 4.12)*

Field compared with **External List** of values *(X is in **List1**)*

Field matched with **Wildcard** *(X is like ABC%, X is like _BC)*

Interval between **2 Dates** *(X is 1 month later than Y)"*

Interval between **2 Business Days** *(X is the next business day after Y)*

Shorthand Condition *(previously saved set of conditions)*

Field compared with a **Freehand Expression** *(X < a + b, X IS NOT NULL)*

Aggregated Condition *(Average(X) > Y)*

Freehand expression *(1 = 1, a + b = c + d)*

AND grouping operator

OR grouping operator

NOT operator

Fig. A1

Below follows the description of each condition type, its purpose and logical operators associated with it.

Logical Conditions Table

Condition Type	Logical Operators	Purpose
List of Values	IS IN, IS NOT IN	Allows building a list of frequently used values taken from a column from the Data Model. Depending on the Logical Operator selection, the selected values will be respectively included or excluded from the portfolio in the process of execution.
Single Value	=, !=, >, >=, <, <=	Allows to select a single numerical value from a column and setup the margin of acceptance
Wildcard	IS LIKE, IS NOT LIKE	Allows to include/exclude values similar to one defined in the window.
Date Interval	=, !=, >, >=, <, <=	Allows defining date interval within which you want the system to perform the data selection.
Date Interval (Business)	=, !=, >, >=, <, <=	Allows defining business date interval within which you want the system to perform the selection process.

Expression (SQL)	=, !=, >, >=, <, <=, IS, IS NOT	Allows inserting an SQL statement of any degree of complexity to perform tasks that may be too cumbersome to define using other conditions.
Expression (Shorthand)	=, !=	Allows to attach a shorthand to the definition
Freehand Expression	N/A	Allows applying an SQL expression to the entire data source. Especially useful when setting up the condition for the Orphan selection.
Aggregated	=, !=, >, >=, <, <=	Allows summarization of the figures by combining several transaction attributes.
'AND' Grouping Operator	N/A	Allows you to build complex transaction selection rules using several conditions at a time.
'OR' Grouping Operator	N/A	Allows you to build complex transaction selection rules using several conditions at a time.
'NOT' Grouping Operator	N/A	Allows you to build complex transaction selection rules using several conditions at a time.

Each Condition Type is represented by a pane with Field Selector button, fields for data input and Logical Operator Selector for setting the executable component of the condition.

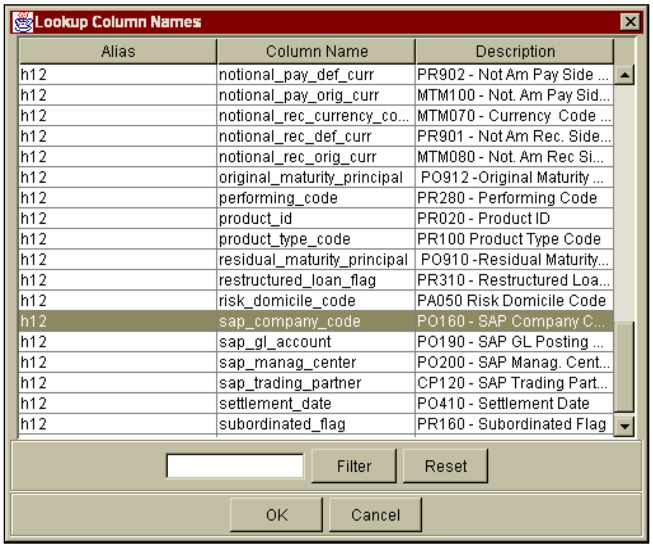
Once the Condition Type is selected, the Condition setup involves following steps:

1. Column selection
2. Data Input
3. Logical Operator Selection

Column Selection

Since the Conditions are applied to the specific data, you have to indicate where this data is located within the source table. Therefore, as the first step in the condition stipulation, you have to indicate to what column you want the system to refer. To select a column:

1. Use the Select Field button Select field... of the Condition pane to open the lookup listing all columns that comprise the source table (Fig. A2).



Lookup Column Names

Alias	Column Name	Description
h12	notional_pay_def_curr	PR902 - Not Am Pay Side ...
h12	notional_pay_orig_curr	MTM100 - Not. Am Pay Sid...
h12	notional_rec_currency_co...	MTM070 - Currency Code ...
h12	notional_rec_def_curr	PR901 - Not Am Rec. Side...
h12	notional_rec_orig_curr	MTM080 - Not. Am Rec Si...
h12	original_maturity_principal	PO912 -Original Maturity ...
h12	performing_code	PR280 - Performing Code
h12	product_id	PR020 - Product ID
h12	product_type_code	PR100 Product Type Code
h12	residual_maturity_principal	PO910 -Residual Maturity...
h12	restructured_loan_flag	PR310 - Restructured Loa...
h12	risk_domicile_code	PA050 Risk Domicile Code
h12	sap_company_code	PO160 - SAP Company C...
h12	sap_gl_account	PO190 - SAP GL Posting ...
h12	sap_manag_center	PO200 - SAP Manag. Cent...
h12	sap_trading_partner	CP120 - SAP Trading Part...
h12	settlement_date	PO410 - Settlement Date
h12	subordinated_flag	PR160 - Subordinated Flag

Fig. A2

2. This is a single selection lookup. Highlight the name of the column you want to use (*currency* for example) and click OK to close the Lookup.

   NOTE: You can use the lookup's Filter to narrow down the entries in the lookup to sift through by typing the first one or several characters of the code or description of the entry you need.

   The selected Column is displayed on the Selection button of the Condition pane (Fig. A3).

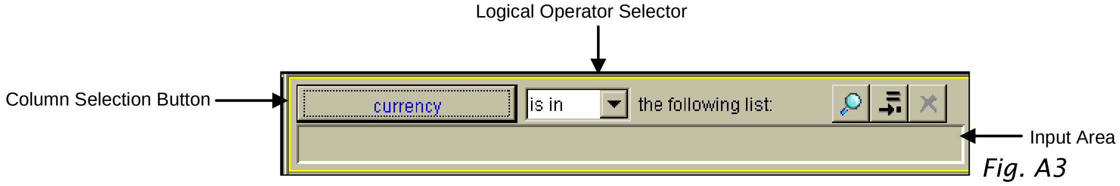


Fig. A3

Data Input

Once you specified the column, the Lookup button 🔍 becomes *active* (if there is a lookup available). Pressing this button will take you to the Lookup window with the contents of the column you selected for this Condition Window.

Depending on the Condition Type it may be single or multiple selection lookup. Scroll down to the entry(s) you want (you can use the Filter to narrow things down), select it and hit OK to close the Lookup.
In the case of some Conditions, you will also be able to use list menus and manual input.

Note: If no lookup has been set up for the selected field, the lookup button will not appear and you will have to enter the information manually.

Logical Operator Selection

Logical Operator is the *executable* component of a Condition. Logical Operators are listed in the list menu of the Logical Operator Selector arrow button (Fig. A3).

1. IS IN – will instruct the system to select the transactions bearing the attributes identical to those specified in the LIST OF VALUES condition pane input field.

2. IS NOT IN – will instruct the system to reject the transactions bearing the attributes identical to those specified in the LIST OF VALUES condition pane input field.

3. IS LIKE – will instruct the system to select the transactions bearing the attributes containing elements specified in the Wildcard condition pane input field.

4. IS NOT LIKE – will instruct the system to reject the transactions bearing the attributes containing elements specified in the Wildcard condition window input field.

5. = (equal) – will instruct the system to select the transactions bearing the attribute identical to the one specified in the condition window input field or fitting within the time period specified in the Time Period condition pane.

6. != (unequal) – will instruct the system to reject transactions bearing the attribute identical to the one specified in the condition window input field or fitting within the time period specified in the Time Period condition pane.

]

7. > (larger) – will instruct the system to select transactions bearing the attribute numerically larger than the one specified in the condition window input field or exceeding the time period specified in the Time Period condition pane.

8. >= (larger or equal) – will instruct the system to select transactions bearing the attribute numerically larger or identical to the one specified in the condition window input field or fitting within or exceeding the time period specified in the Time Period condition pane.
9. < (smaller) – will instruct the system to select transactions bearing the attribute numerically smaller than the one specified in the condition window input field or to the time period specified in the Time Period condition pane.

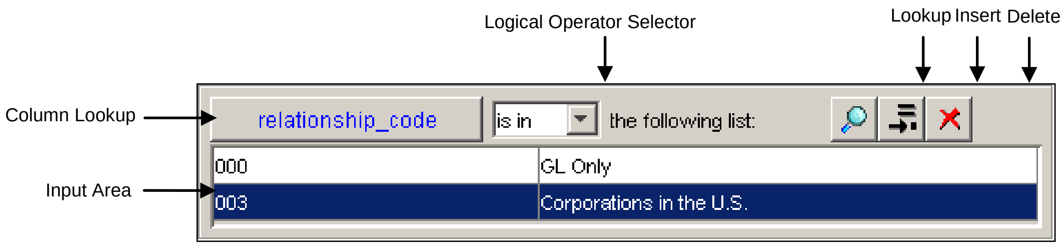
10. <= (smaller or equal) – will instruct the system to select transactions bearing the attribute numerically smaller or identical to the one specified in the condition window input field or to the time period specified in the Time Period condition pane.

11. IS – will instruct the system to select transactions bearing the attribute satisfying the SQL statement entered into the Expression Condition.

12. IS NOT – will instruct the system to disregard transactions bearing the attribute satisfying the SQL statement entered into the Expression Condition.

List Condition

Into the List condition pane (Fig. A4) you enter a set of values from a source table column (transaction attribute) with the default Logical Operator set on **IS IN**, which indicates that the items you enter into the List Condition will be *included* in the data selection process.



000	GL Only
003	Corporations in the U.S.

Fig. A4

The following components constitute the LIST Condition pane:

Column Lookup Button

Allows access to the lookup listing available columns (data attributes). When selection is made, the button displays the name of the selected field.

Logical Operator Selector

Logical Operator selector is set on '**IS IN'** by default. You can change it to '**IS NOT IN'** by clicking on Logical Operator Selector arrow button and choosing this option from the list menu (Fig. A5).

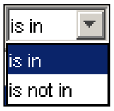


Fig. A5

Lookup Button

The lookup button accesses the multiple selection lookup window with the contents of a specific column. Not shown initially it becomes active when you complete the **Field selection** (see above).

Insert Button

Allows you to insert blank rows into the input area when entering values manually (see below). This feature is useful if you know the values and do not want to go through extra steps with the lookup.

Delete Button

Remove entries (rows) from the Input Area.

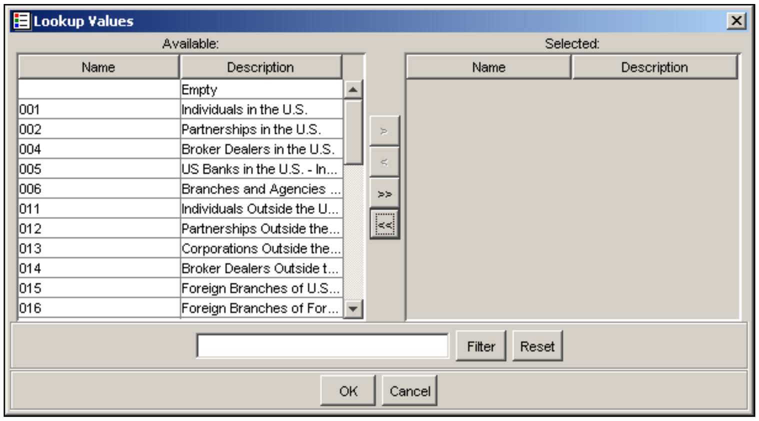
Input Area

In this area you specify the values to which the LIST Condition will be applied.

Inputting Data into the List Condition

The column selection for this Condition window is done following the steps outlined above in the Column Selection section of this chapter. After you made a column selection:

1. Click on the Lookup button to gain access to the contents of the selected column (Fig. A6).



Available:

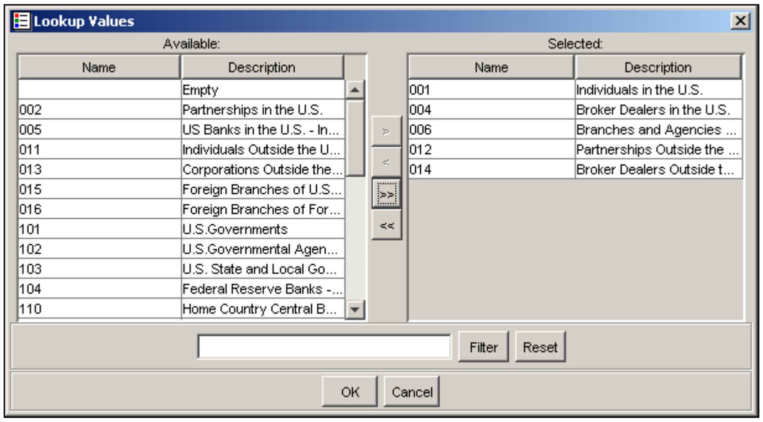
Name	Description
	Empty
001	Individuals in the U.S.
002	Partnerships in the U.S.
004	Broker Dealers in the U.S.
005	US Banks in the U.S. - In...
006	Branches and Agencies ...
011	Individuals Outside the U...
012	Partnerships Outside the...
013	Corporations Outside the...
014	Broker Dealers Outside t...
015	Foreign Branches of U.S...
016	Foreign Branches of For...

Selected:

Name	Description

Fig. A6

2. Select the items and move them to the Selected grid one by one by double clicking on one entry at a time. You can also do the multiple select by pressing the Ctrl key and using '**>' button** to move selected entries to the right side of the Lookup screen (Fig. A7).



Available:

Name	Description
	Empty
002	Partnerships in the U.S.
005	US Banks in the U.S. - In...
011	Individuals Outside the U...
013	Corporations Outside the...
015	Foreign Branches of U.S...
016	Foreign Branches of For...
101	U.S.Governments
102	U.S.Governmental Agen...
103	U.S. State and Local Go...
104	Federal Reserve Banks -...
110	Home Country Central B...

Selected:

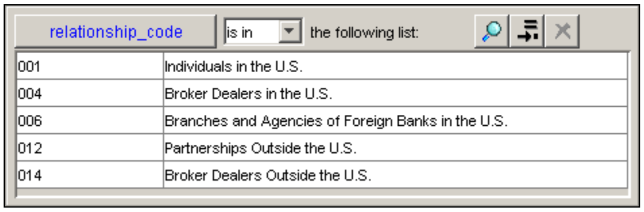
Name	Description
001	Individuals in the U.S.
004	Broker Dealers in the U.S.
006	Branches and Agencies ...
012	Partnerships Outside the ...
014	Broker Dealers Outside t...

Fig. A7

Filter input field and button allow you to narrow down the contents of the left grid using one or several digits of the value or the description common to the desired entries.

If you decide that some selected entries should not be there, you can remove them from the right side of the grid by highlighting them and using '**<**' button to move them back to the left grid. **Note: Clicking on the double arrow button will move the entire contents of a grid.**

3. Click OK to close the Lookup. The selected items appear in the List Condition Pane (Fig. A8).



relationship_code is in the following list:

001	Individuals in the U.S.
004	Broker Dealers in the U.S.
006	Branches and Agencies of Foreign Banks in the U.S.
012	Partnerships Outside the U.S.
014	Broker Dealers Outside the U.S.

Fig. A8

Manual Entry

If you know the codes you need to enter or if there is no lookup available for the selected column, you can enter data manually by inserting rows into the input pane and typing in the values. The system will validate your entries and insert corresponding descriptions.

Editing Data Selection

You can edit the contents of the List Condition window by removing and adding values to the list.

Adding Entries

Add items using the Lookup or Insert functions as described above.

Removing Entries

Remove items by highlighting rows in the List Condition data grid and using the Delete button ☒, activated as the rows get highlighted, to remove the selected row(s).

You can also remove items from the List condition by opening the lookup window, which will show the selected values in the right side grid, and transferring the unwanted entries from the right side grid to the left by highlighting them and using '**<**' button.

Single Value Condition

Single Value condition allows selecting a single numerical value and set up the Logical Operator to establish the margin of acceptance (Fig. A9).

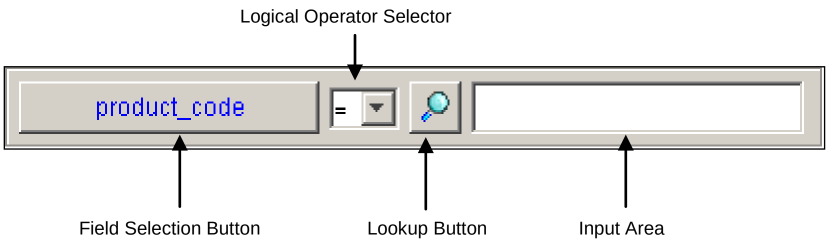


Fig. A9

Select Field Button

Allows access to the lookup listing available columns (data attributes). When selection is made, the button displays the name of the selected field.

Logical Operator Selector

Set on “**=**” by default. You can change it by clicking on Logical Operator Selector arrow button and choosing this option from the list menu (Fig. A10). Refer to the Condition Table for the complete list of the Logical Operators associated with the Singular Concrete condition type.

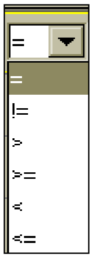


Fig. A10

Lookup Button

This button accesses the lookup with the contents of the specific column. Initially absent this button appears when you select a column.

Input Area

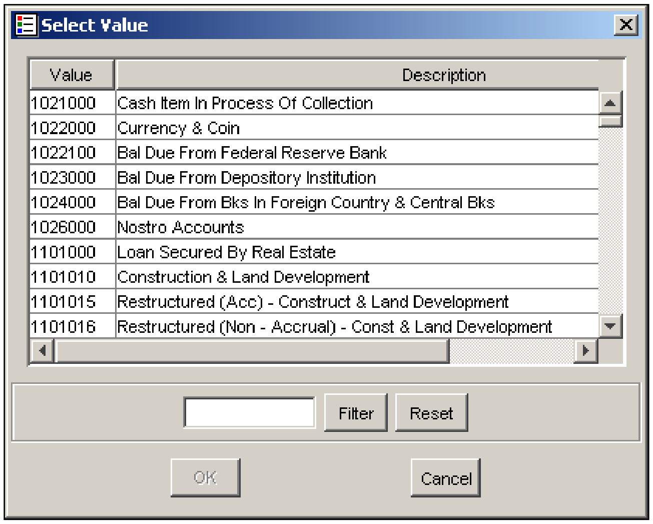
In this area you specify the data type to which Singular (Concrete) Condition will be applied.

Single Value Condition Setup

The column selection for this Condition window is done following the steps outlined above in the Column Selection section of this chapter.

After you made a column selection:

1. Use to open the lookup with the contents of the selected column (Fig. A11).



Value	Description
1021000	Cash Item In Process Of Collection
1022000	Currency & Coin
1022100	Bal Due From Federal Reserve Bank
1023000	Bal Due From Depository Institution
1024000	Bal Due From Bks In Foreign Country & Central Bks
1026000	Nostro Accounts
1101000	Loan Secured By Real Estate
1101010	Construction & Land Development
1101015	Restructured (Acc) - Construct & Land Development
1101016	Restructured (Non - Accrual) - Const & Land Development

Fig. A11

2. Select the value you want by highlighting it and click OK to close the lookup. The selected entry appears in the Condition pane's input field (Fig. A12).

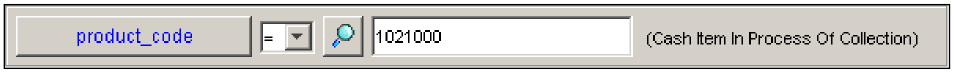

product_code = 1021000 (Cash Item In Process Of Collection)

Fig. A12

3. Select an appropriate Logical operator.

Manual Entry
If you know the code you need to enter or if there is no lookup available for the selected column, you can enter data manually into the input pane field. The system will validate your entry and insert corresponding description.

Wildcard Condition

Allows to select values *similar* to one defined in the window. Use logical operators IS LIKE/IS NOT LIKE to indicate inclusion or exclusion of values from the extracted data (Fig. A13).

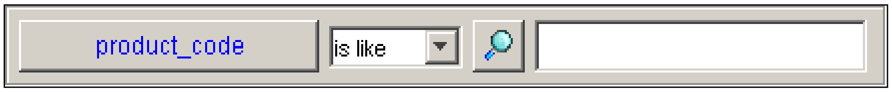

product_code is like

Fig. A13

The column selection for this Condition is done following the steps outlined above in the Column Selection section of this document.

Logical operator selection is limited to 'IS LIKE' and 'IS NOT LIKE'.

After you selected a column, use 🔍 to enter a specific parameter into the input field. You can also enter a value manually. For example:

- Entering A* will extract from the selected column all entries starting with an A;
- Entering *A will extract from the selected column all entries ending with an A;
- Entering *A* will extract from the selected column all entries containing an A.

2 Date Interval Condition

Allows defining a calendar date interval within which you want the system to perform the data selection (Fig. 2-56). When setting up the Date Interval it must be kept in mind that the system will use all days in the internal calendar, including days off and holydays. If you need the system to use Business days only, refer to the Date Interval (Business) section below.

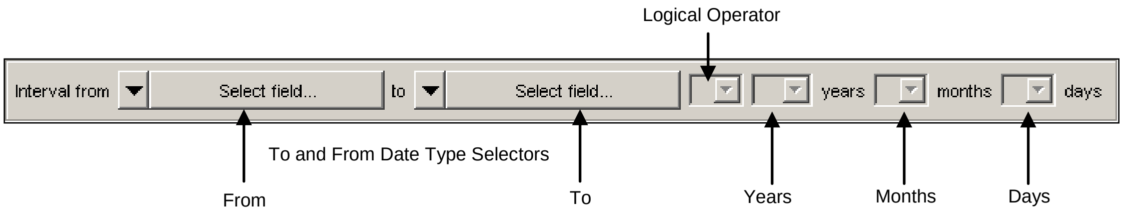


Fig. A14

Date Interval Selectors

There are two Date selectors:

1. “From” date
2. “To” date

Each selector has an Arrow button clicking on which opens the list menu with date types to be defined as the interval margins in setting up the Date Interval condition (Fig. A15).

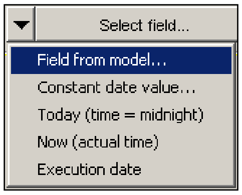


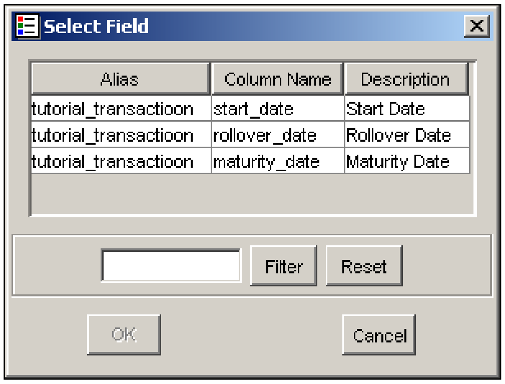
Fig. A15

Field From Model

Selected by default, this option instructs the system to refer to a date-related column of your choice in the Source table, such as Market Date, Issue Date, Maturity date, etc.

To set the margins using Field From Model:

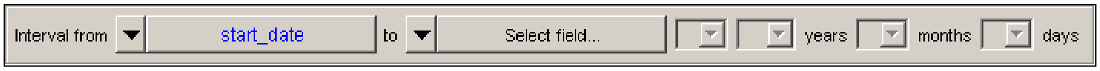
1. Click on the Select Field (From) button. The Lookup appears listing available fields containing Date information (Fig. A16).



Alias	Column Name	Description
tutorial_transactioon	start_date	Start Date
tutorial_transactioon	rollover_date	Rollover Date
tutorial_transactioon	maturity_date	Maturity Date

Fig. A16

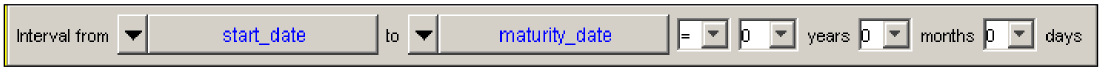
2. Select the required Date column name and click OK to close the lookup. The selected Date column appears on the Select Field button of the ‘From’ margin (Fig. A17).



Interval from start_date to Select field... years months days

Fig. A17

3. To set To margin of the interval using Field From Model selection repeat steps 1 and 2 using Select Field (To) button selecting another Date column for you To margin. The selected Date column appears on the Select Field button of the To margin (Fig. A18).



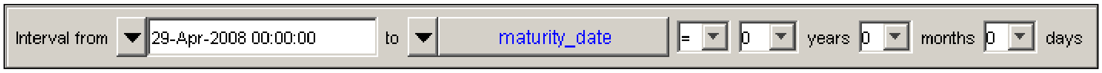
Interval from start_date to maturity_date = 0 years 0 months 0 days

Fig. A18

Note: Defining date margins activate the Logical Operator selector and Date input fields.

Constant Date Value

Selecting Constant Date Value from Date Type selector menu allows you to use a particular date as a margin parameter. By default, when you select Constant from the Date Type selector's list menu, the current date appears in the Margin input field (Fig. A19).



Interval from 29-Apr-2008 00:00:00 to maturity_date = 0 years 0 months 0 days

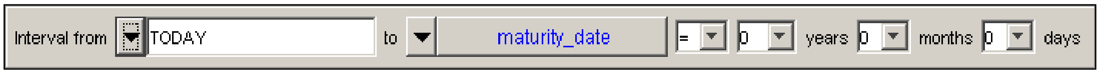
Fig. A19

To change the date, click on the beginning of the entry in the Margin input field and type over the current date. You can use precise time (if relevant to your purpose) by typing hours, minutes, and seconds in the military format.

The above setup will prompt the system to evaluate the time period between the current date (or the date you entered manually and the maturity date.

Today (time=midnight)

Selecting this Time Margin (Fig. A20) allows you to use the current date as a margin parameter.



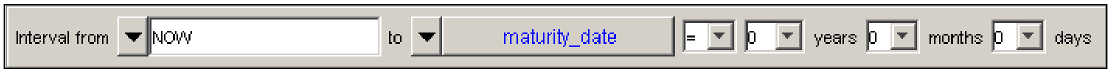
Interval from TODAY to maturity_date = 0 years 0 months 0 days

A20

Set up the time period following the steps described below.

Now

Selecting this Time Margin allows you to use the current date and the time of the day (Fig. A21).



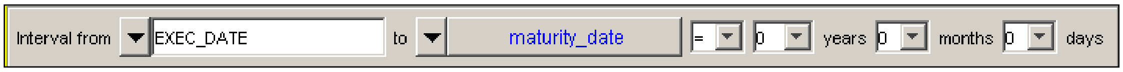
Interval from NOW to maturity_date = 0 years 0 months 0 days

Fig. A21

Set up the time period following the steps described below.

Execution Date

Refers to the date of the current Portfolio Execution. To set the **From** time margin to Execution date, select the latter from the list menu of the Margin Type Selector. EXEC_DATE appears in the From Date input field (Fig. A22).



Interval from EXEC_DATE to maturity_date = 0 years 0 months 0 days

A22

Setting Up Time Period

Next, you have to indicate the number of years, months and/or days the time period must be for the transaction to be included/excluded during a task execution.

The system will allow you to set any combination of years, months and days that might be required by your business needs.

To set the time period:

1. **Set the number of years.** Pressing the arrow button next to the Years input field produces the list menu that goes up and down to 29 years. If this is not enough, you can type in any number you want.

Note: Selecting a negative value for the Date interval margin parameters allows you to reverse To and From margins.

2. **Set the number of months.** Pressing the arrow button next to the Months input field produces the list menu that goes up to 12 months. Select the number you want by clicking on it.

3. **Set the number of days.** Pressing the arrow button next to the Days input field produces the list menu that goes up to 31 days. Scroll down to the number you want and click on it.

For the purpose of our example we will set up the date interval between Execution date and the Maturity date to 1 year.

Logical Operator Selection

An arrow button of the Logical Operator selector opens a menu (Fig. A23) listing logical operators associated with the Date Interval condition.

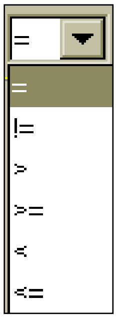


Fig. A23

If ‘=’ the system will select records with 1year to Maturity

If ‘!=’ the system will select records with either more or less than 1year to Maturity

If ‘>’ the system will select records with more than 1year to Maturity

If ‘>=’ the system will select records with the interval larger or equal to 1year to Maturity

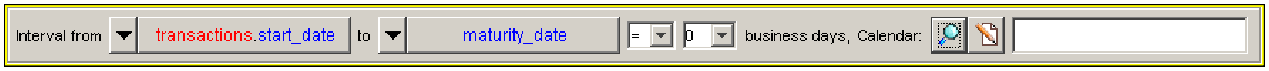
If ‘<’ the system will select records with the less than 1year to Maturity

If ‘<=’ the system will select records with less or equal to 1year to Maturity

Date Interval (Business) Condition

Allows defining date interval within which you want the system to perform the data selection using business days only. The system will refer to the internal calendar for holidays and weekends.

Date Interval (Business) Condition window (Fig. A24) is similar to the Date Interval setup except that in this case you do not use Years and Months, but days only. Indeed, it is difficult to imagine what “the business year” would represent.



Interval from transactions.start_date to maturity_date = 0 business days, Calendar:

Fig. A24

🔎 **Button** opens the lookup that allows you to select the calendar you want the system to use when determining business days.

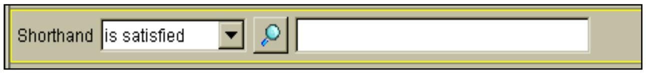
To input the number of business days, either click on the arrow button and select a number from the list menu (from 01 to 64) or simply type the desired number directly into the input field.

Note: Selecting a negative value for the Date interval margin parameters allows you to reverse To and From margins.

The rest of the setup is identical to the regular 2 Date Interval setup.

Shorthand Condition

This Condition (Fig. A25) allows using a preset combination of conditions or a list of values as a criterion for data selection.



Shorthand is satisfied

Fig. A25

For example, you know that for certain purposes a recurring set of currencies should be excluded from a report. It is too cumbersome to enter these currencies into a Condition every time you set up data selection rules. Therefore you create a Shorthand, name it and save it. Now, every time you need to *exclude* these currencies from the data display, you insert a Shorthand Condition and refer to this Shorthand as a single value with IS NOT IN logical operator.

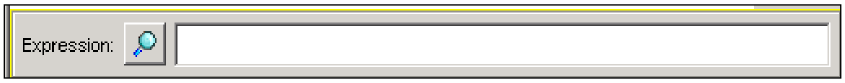
Please refer to the Shorthands section of this manual.

Shorthand Condition operates similar to the Single Value condition, except the logical conditions available are IS IN and IS NOT IN and the Lookup button provides an access to the Shorthand repository and not to the selected column contents.

Use the Lookup to select a shorthand you want to use as a comparison component of the condition.

Freehand Expression Condition

This Condition window (Fig. A26) allows you to input an SQL or mathematical expression to apply to the *entire data source* as a comparable component of a condition.



Expression:

Fig. A26

🔍 button opens a menu (Fig. A27) from where you can select either Column names or a Freehand Macros to include in your statement.

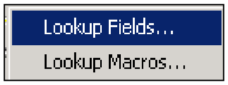


Fig. A27

You can type in any expression statement in the entry field of the Condition window.

Note: This feature is for advanced users only proficient in SQL programming. We strongly advise caution when using this condition. Please consult Axiom Software Labs.

Aggregated Condition

Aggregated condition allows you to set up summarization criteria for combining multiple values from a calculated field into a single number and then comparing this number with a preset comparable value to determine whether the records will be selected or discarded by the system during the Portfolio execution. This complex condition was designed for the use in the Portfolio Definition only and will not work on the analytical screens such as Aggregation Viewer and Filter.

We strongly recommend that you employ this type of condition sparingly not combining it with other conditions in the same node of a portfolio structure.

Note: Setting up the Aggregated condition can be better understood in the context of the Portfolio definition. Please refer to the Portfolio Definition section of this document for better understanding of the Aggregated Condition setup.

Aggregated Condition Pane

The Aggregated Condition pane (Fig. A28) consists of following components:

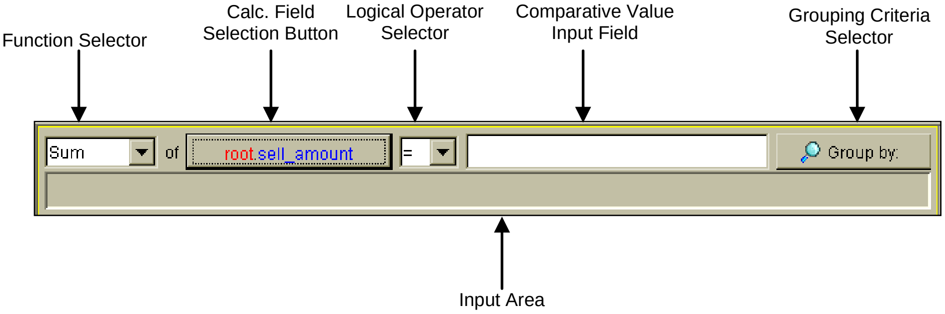

Function Selector
Calc. Field Selection Button
Logical Operator Selector
Comparative Value Input Field
Grouping Criteria Selector
Sum
of
root.sell_amount
=
Group by:
Input Area

Fig. A28

Calculated Field Selection

The Calculated Field selection for this Condition is done with the Select field... button that opens a lookup listing available fields. The numbers in this calculated field will be summarized by the system and evaluated against the Comparative value.

Logical Operator Selection

See the Condition Table for the list of Logical Operators associated with the Aggregated condition (see Logical Conditions, Chapter 2).

Comparative Value

Into this input field you manually enter the number against which you want the system to evaluate the results of the calculations.

Function Selection

Function selection allows you to define the computing procedure you want the system to perform. Pressing the arrow button a menu (Fig. A-29) listing available options.

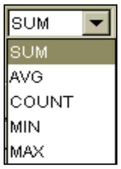


Fig. A29

1. SUMMARY – If SUM is selected, the system will summarize the values in the *calculated field* and compare the sum against the comparative Value.
2. AVERAGE – If Average is selected, the system will calculate the average of the *calculated field* values and compare the result against the comparative value.
3. COUNT – If this function is selected, the system will count how many records are present in the group (see *grouping criteria*) and compare the result against the comparative value.
4. COUNT_DISTINCT – If this function is selected, the system will count how many distinct values of the *calculated field* are present in the group and compare the result against the comparative value.
5. MINIMUM – If this function is selected the system will determine the Minimum value of the *calculated field* and compare it against the comparative value.
6. MAXIMUM – If this function is selected the system will determine the Maximum value of the *calculated field* and compare it against the comparative value.

Grouping Criteria

You can use Group By button to select transaction attributes the system will use for data compartmentalization. Pressing this button opens a lookup with the list of available transaction attributes.

Aggregated Condition Setup

Note: Setting up the Aggregated condition can be understood better in the context of the Portfolio definition. Please refer to the Portfolio Definition section of this document for better understanding of the Aggregated Condition setup.

Reverse Data Processing

It is important to point out, that when the Aggregated Condition is used, the system changes its data processing sequence. Usually the data is processed from top to bottom along the Portfolio hierarchy.

However when it encounters a node with the Aggregated condition, it starts processing data of its children nodes first and then applies Aggregated condition to the resulting data set.

Let us assume we have a source that consists of 35 transactions. In our portfolio setup we want to include a segment that contains transactions for counterparties *cp_a*, *cp_b* and *cp_c*, and only if the Summary value of these transactions equals or exceeds 1 million USD. To accomplish this, you set up a group of nodes:

1. A parent node with the Aggregated condition specifying, which calculated field to *summate* (*purchase_amount*), grouping criteria (*counterparty*), which function to perform (SUM), which logical operator to apply (>=) and the Comparative value (1,000,000).

2. A child node with a List condition in which you specify which counterparties you want the system to isolate in this portion of the portfolio (counter parties *cp_a*, *cp_b* and *cp_c*). Alternatively you can create three child nodes – a node per counterparty in question.

Note: If you do not specify the counterparties and only set up the node with the Aggregated Condition as described above, the system will check every counterparty included in the source.

To setup the Aggregated condition:

1. On the Portfolio panel highlight the node you want to have the Aggregated condition.

2. Click on the Insert Condition option of the Edit pull down menu and select the Aggregated condition from the list. Blank Aggregated Condition window will appear (Fig. A30).

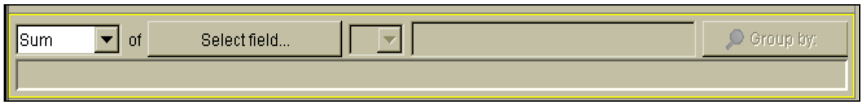


Fig. A30

3. Press Select field... button to open the lookup, listing available calculated fields.

4. Select a calculated field (*purchase_amount*, in our example) and click OK to close the lookup. The name of the selected field is displayed on the Field selection button (Fig. A31).

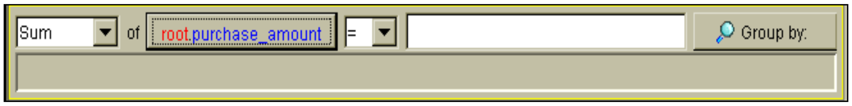


Fig. A31

5. In the Function selector the SUM option is selected by default.

6. Use Logical Operator selector's arrow button to select >= option from the list menu (Fig. A32).

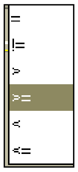


Fig. A32

7. Click on the Comparative Value input field and type in 1000000 (Fig. A33).

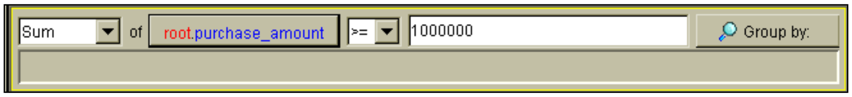
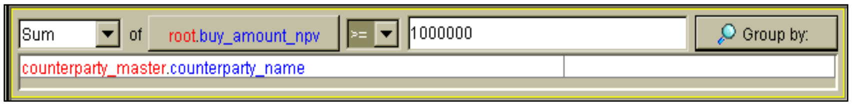


Fig. A33

8. Click on the Group by: button. The multiple selection lookup opens listing available columns that can be used as transactions attributes.

9. Select a column name and click OK to close the lookup. Selected column name appears in the data grid of the Aggregated Condition window (Fig. A34).


Sum of root.buy_amount_npv >= 1000000

counterparty_master.counterparty_name

Fig. A34

Aggregated Condition Execution

As you remember, our sample Aggregated condition setup called for three counterparties (*cp_a, cp_b* and *cp_c*). Transactions involving these counterparties were separated during execution by the List condition of the node which was set up for this purpose as a child to the Aggregated condition node.

At this point the system has not gotten to the Aggregated condition yet and only retrieved transactions based on the conditions in the child node.
As the next step, the system applies the Aggregated condition to the data set resulting from the child node's condition execution. All isolated transactions are divided into 3 sub-segments based on *counterparty_name* as a grouping criterion. In each group the system summates amounts in the selected calculated field and checks if the figure in this column is equal or greater than 1 million. If it is, than all transactions involving this counterparty are accepted.

Multiple Grouping Criteria

You can refine the Aggregated condition setup by selecting several grouping criteria. To expand our example, let us assume that for the selected counterparties *cp_a*, *cp_b* and *cp_c* you would want to isolate only transactions that were conducted in USD and CD.

To accomplish this you would include currency selection criteria in the child node and then include Currency ID into the data grid of the Aggregated Condition window. The system then would create groupings for every counterparty/currency combination and evaluate the results with our comparative value of 1,000,000.

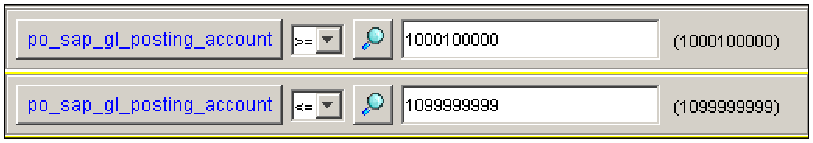
If you want to add specific locations to your Aggregated Condition setup following the above steps, the system will create all possible counterparty/currency/location combinations and so on.

Combining Logical Conditions

AND, **OR** and **NOT** operators allow to combine logical conditions for refined data selection. If you, for example, during the definition of a portfolio node or a shorthand, want to group records by virtue of sharing certain characteristics and differing in others, you would use these operators by combining them with various conditions for data selection.

AND Grouping Operator

AND is a default grouping operator. Every time you group several conditions, the system will insert an [invisible] AND operator between them. The system will select records when all the conditions in the group are satisfied. For example, when processing two single value conditions below (Fig. A53), the system will isolate records falling in the range of indicated SAP GL account code numbers.



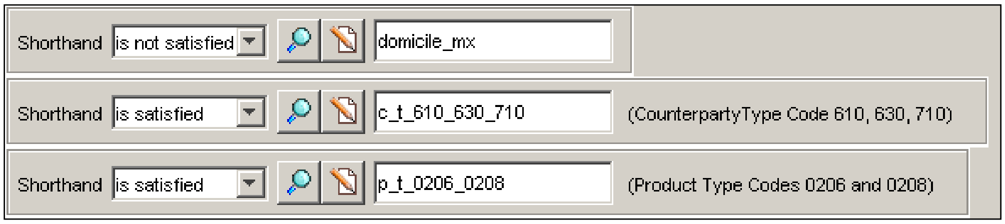
po_sap_gl_posting_account >= 1000100000 (1000100000)

po_sap_gl_posting_account <= 1099999999 (1099999999)

Fig. A35

In other words, in order for a transaction to qualify with the above setup, its SAP GL code should be larger than 1000100000 **AND** smaller than 1099999999.

Another example (Fig. A36) shows a set of conditions tied with the default (invisible) ‘AND’ operator.



Shorthand is not satisfied domicile_mx

Shorthand is satisfied c_t_610_630_710 (CounterpartyType Code 610, 630, 710)

Shorthand is satisfied p_t_0206_0208 (Product Type Codes 0206 and 0208)

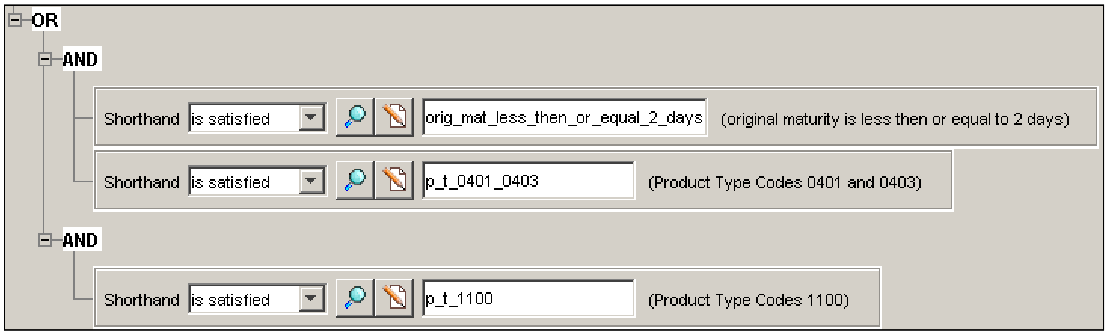
Fig. A36

The above setup will isolate transactions satisfying all three shorthand conditions included in the group.

Note: The ‘AND’ operator becomes visible when used in a combination with other grouping operators.

OR Grouping Operator

The ‘OR’ grouping operator is usually used in conjunction with “AND” and “NOT” operators and instructs the system to select between several conditions (or a groups of conditions) in the setup. Below (Fig. A37) is the setup where the lead-in OR operator indicates that the system would pick records that satisfy either set of conditions preceded by the AND operator.



OR

AND

Shorthand is satisfied orig_mat_less_then_or_equal_2_days (original maturity is less then or equal to 2 days)

Shorthand is satisfied p_t_0401_0403 (Product Type Codes 0401 and 0403)

AND

Shorthand is satisfied p_t_1100 (Product Type Codes 1100)

Fig. A37

NOT Grouping Operator

The ‘NOT’ grouping operator is used rarely for exclusion of values. It allows only one condition to be attached to it.