SMART BB Security Master Adapter Users Guide

**On this page**

* [Prerequisite Software](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Prerequisite_Software)
* [Components](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Components)
* [Installing SMART](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Installing_SMART)
* [Removing SMART](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Removing_SMART)
* [Upgrading an Installed SMART](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Upgrading_an_Installed_SMART)
* [Security Privileges](https://client-connect.iongroup.com/library/content/treasury-management/openlink/operations/smart-bb-security-master-adapter-users-guide/#Security_Privileges)

SMART is a container application which hosts a series of adapters that provide a visual way for users to map OLF business objects to 3rd party data services. Specifically, the Bloomberg Security Master Adapter manages the creation and maintenance of Security Master Instruments where the source is the Bloomberg DataLicense and SAPI products.

In the past, such functionalities were provided by creating plugin programs that implement mapping rules in code. In many ways this type of architecture is quick to implement for simple instruments but hard for complex instruments. Additionally, maintenance of these programs becomes more difficult as more rules are added over time. Since the business logic is hidden in code, the exact capabilities need to be meticulously documented since it is not apparent in any other way.

This application attempts to move the business logic out of the code to an environment where a business user can view the mappings that contain the business rules and even create new rules or change existing rules.

Business rules can quickly become complicated, firstly due to actual complexity, and secondly due to the sheer volume of logic required to make provision for all the different rules and exceptions to rules. Business rules for mappings can roughly be grouped as follows:

* **One-to-One**

Most mappings are easy. The Bloomberg ISSUE\_DT field mostly maps to all types of Bonds’ Start Date. Since most mappings fall in this category, the illusion of simplicity is created.

* **One-to-One with Transformation**

Some data need to be transformed prior to importing it into a Findur instrument. For instance, the Bloomberg value for the FIXED field returns Y or N. These values need to be transformed into the equivalent Findur terms of Fixed or Float. This can be done using lookup tables.

* **If-then-Else**

In some cases, logic is necessary to determine which field from Bloomberg needs to be set on a specific Findur Bond field at run-time. For instance, if we know that a specific bond is an option but we do not know whether it is a Put or Call, we need rules that provide for both cases, but execute only the applicable set of mappings once it is known whether the option is a Put or Call.

The Bloomberg DataLicense Adapter application provides a visual point-and-click user interface that allows mapping rules to be created for all the above scenarios. In addition, it is also possible to define and apply a formula to a retrieved value and create lookup tables.

For scenarios that are even more complex, where there are several layers of “if” statements, some programming is still required. The application provides hooks where custom logic in code can be implemented.

This document is intended to be a guide for using and installing the Bloomberg Security Master Adapter to setup mappings between Bloomberg’s DataLicense product and Findur Bond and Money Market template holding instruments where the intention is to import and maintain these objects in a Findur instance.

The Bloomberg DataLicense Adapter of SMART provides a graphical user interface that can be used to map physical Bloomberg fields to Findur holding instrument fields. For example, mapping Bloomberg’s ID\_CUSIP field to Findur’s Bond instrument’s CUSIP field.

SMART is not intended to be used in an operational workflow, it is an administrative tool. It should be used to set up and test mappings for different types of Bond and Money Market instruments, i.e., fixed rate bonds, float rate bonds, options etc. Operationally, these mappings can then be used by providing a facility for traders to import an instrument, at will, whenever it is noticed that a holding instrument for a specific published instrument does not exist in Findur. A separate component supplied with SMART can be attached to a button within a Findur trading screen to provide this capability.

Diagram

Description automatically generated

Suggested Workflow

The application is a regular Windows application that attaches to a running instance of Findur to get Findur data. It also connects to Bloomberg’s DataLicense web-service to get Bloomberg data. The mappings created by the user maps the data provided by Bloomberg to specific Findur holding instrument fields.

The application provides the following:

* The ability to create, load, edit, and save the mappings containing business rules
* A lookup table for simple transformations
* The ability to apply a formula to an import value
* The ability to apply if-then-else logic
* The ability to combine user input (fixed values) with retrieved values
* The ability to concatenate the values of several fields into one Findur field
* A rule to follow if a mapping experiences an error
* A facility to test mappings and view the outcome without importing the instrument
* Maintain the connection properties required to make the connection to the Bloomberg Service
* A viewer for the data gathered from Bloomberg and set according to various rules on an instrument
* The ability to automatically assign a mapping to the data of a specific returned instrument based on the attributes of that instrument.
  + For instance, if the BB fields **FIXED** returned **Y** and **MARKET\_SECTOR\_DES** returned **Govt**, the instrument is a Fixed Rate Govt Bond
* Explicitly assign issuers thereby negating the need to also map and have lookups for issuers
* Explicitly assign mappings thereby negating the need to have complex rules to automatically assign the mappings
* Manage Bloomberg-specific requirements such as attaching Bloomberg Terminal attributes to a web-service call
  + These attributes are the Bloomberg user number, serial number, and workstation number. Depending on the license of the client, these attributes can be managed on a per user-level or a shared terminal can be defined

Prerequisite Software

SMART is built using .Net 4, thus Microsoft .Net 4 must be installed on the target installation computers.

SMART makes use of the OLF OpenComponent API, which is dependent on the specific Findur installation. It may be necessary, although not always required, to renew the SMART installation upon a Findur upgrade.

SMART is provided in both 32-bit and 64-bit versions.

Components

SMART consists of three related components:

* The SMART Mapping Tool (SMT)
  + Used to create visual mappings and mapping rules
  + Used to set up the properties required for the tool to correctly process
    - Set the credentials and other options for attaching to the Bloomberg service
    - Define lookups to be used during that import process
* Ad-Hoc Security Master Import (ASI)
  + An Openlink *TradeInput* plugin that can be attached to a button on a trade input form to be activated in an ad-hoc manner to import a specific Security Master
* Batch Security Master Import (BSI)
  + A regular Openlink plugin that can be associated with a workflow that takes a CSV file as input to import multiple Security Master objects in a single operation

Installing SMART

SMART Mapping Tool (SMT)

SMART is installed by unzipping its deployment package and merging the resultant *bin* directory with a Findur *bin* directory. SMART must be installed on each installed version of Findur where a SMART user is expected to use this tool.

SMART must be started from within a Findur environment. The easiest way to accomplish this is creation of a button using the *Openlink Central Configuration* window.

Graphical user interface, text, application, email

Description automatically generated

Create a Service to Start SMART

SMART Batch Security Master Import (BSI)

Like the SMT, the BSI executable is available from the OLF bin directory. It can be directly referenced using a button or it can be associated with a workflow or TPM process. If in a workflow, use a *shell* job with the following command *start smart.tradeupload.exe.*

The SMART.TradeUpload.exe also supports command line arguments. For a full list of supported arguments, and a description of each, run SMART.TradeUpload.exe -h on the command line.

A picture containing table

Description automatically generated

Services Manager: Workflow Management Tab

SMART Ad-Hoc Security Master Import (ASI)

Importing the Required Software

The ASI module is the component that can be attached to a button on a trade input screen.

1. Using the *Plugin Library Import* window, choose the **Select Import Libraries**menu.

Graphical user interface, application

Description automatically generated

Plugin Library Import Window

2. Navigate to the *FindurbinSMARTembedded* directory and select the **smartlibs.out** file.

To use this feature, see section 4.5.

Graphical user interface, text, application, Word

Description automatically generated

Embedded Component Directory and Files

3. Click the **Open** button.

Graphical user interface, application, table

Description automatically generated

Plugin Library Import Window

4. Click the **Analyze** button, then the **Save** button.

Setup the TradeInput Button

A screenshot of a computer

Description automatically generated

Add a Button to a Holding Instrument Input Form

Add the **Script** button from the **Buttons** field selection panel on the left and associate *Olf.Risk.Apps.SMARTInstrumentImport* with the **Script Name** property.

Removing SMART

On the servers where SMART is deployed and from within the bin directories of the specific Findur instances, delete the following:

* SMART Directory
* SMART.exe
* SMART.exe.config
* SMART.TradeUpload.exe
* SMART.TradeUpload.exe.config
* Remove the SMART User Defined Security objects
* From within the *OpenComponent Library Window*
  1. Select the Library Open Saved Configuration menu item.
  2. Select all libraries that begin with the word **SMART**. The libraries are shown in the top grid of the window.
  3. Select all the libraries.
  4. Select the **Library** menu item again.
  5. Click Delete.

Upgrading an Installed SMART

Follow the instructions in section 3.1.4, *Removing SMART*, points 1-5, then follow section 3.1.2, *Installing SMART*.

Security Privileges

The following security objects are used by the application and can be configured in the User Defined Security Privileges module:

* **SMART Bloomberg-Open**

Determines whether users can start the Bloomberg adapter in the SMART application.

* **SMART Bloomberg-Edit**

Determines whether a user can save a configuration in the SMART application.

* **SMART Bloomberg-Execute**

Determines whether a user can execute an instrument import in both the SMART application and the embedded component.

* **SMART Bloomberg-Override**

Determines whether a user can provide overridden configuration values at runtime.

* **SMART Manage Configurations**

Allows users to do a global Import/Export of all SMART configurations