

FUSION XL

Structure Explorer User Guide

FUSION XL

FusionXL is a plugin for Microsoft Excel® which provides the ability to browse for, and author registry content in the Fusion Registry

User Guide

This guide provides information on how to use the plugin in order to browse and create Fusion Registry structural metadata

IMPORTANT

This User Guide is no longer maintained. To view the latest version, please refer to the [Metadata Technology Wiki](#).

Contents

1	Overview	3
2	Structure Browsing Features Overview	4
2.1	Connect to Multiple Fusion Registries	4
2.2	Browse, Lookup and Search Structures.....	4
2.3	Recent History.....	4
2.4	Multi Lingual Support.....	4
2.5	Historical Modification	4
2.6	Secure.....	4
2.7	Structure Creation, and Deletion	4
3	Connecting to Fusion Registry.....	5
4	Structure Retrieval.....	6
4.1	Browse Registry	6
4.2	Lookup Structure.....	7
4.3	Search Structure	7
4.4	My Recent History.....	8
5	Structure Action.....	10
5.1	Create Structure.....	10
5.2	Delete Structure	11
5.3	Save Structure.....	12
6	Historical Modifications	13
7	Multilingual Support	14
8	Structure Format	15
8.1	Category Scheme	15
8.2	Codelist & Value List.....	16
8.3	Concept Scheme	16
8.4	Content Constraint.....	16
8.4.1	KeySet - Constraining Series Keys.....	17
8.4.2	Cube Region - Constraining Code Values.....	18
8.5	Data Structure Definition	18
8.5.1	Dimensions Section	18
8.5.2	Attributes Section.....	19
8.5.3	Component Restriction Section	20
8.6	Dataflow	20



8.7	Provision Agreement.....	21
-----	--------------------------	----

1 Overview

FusionXL is a plugin for Microsoft Excel® which, on enabling, will provide the user with the ability to set up connections to one or more Fusion Registries for the retrieval, viewing, and modification of Structural Metadata content.

FusionXL is distributed as a single XLAM file. FusionXL is compatible with any Fusion Registry from version 8.4 onwards. FusionXL provides both structure browsing and maintenance, and from Fusion Registry 9.0 onwards it also supports data discovery. This document covers the structure browsing and maintenance features. The document “FusionXL UserGuide DataExplorer” covers data discovery and retrieval (Fusion Registry installation only).

Information on installing the plugin is provided in the [Setup Guide](#).

2 Structure Browsing Features Overview

2.1 Connect to Multiple Fusion Registries

FusionXL will connect to one or more instances of the Fusion Registry version 8.4 or higher. FusionXL can store up to 5 connections.

2.2 Browse, Lookup and Search Structures

There are three different ways to find structures in the Fusion Registry which are:

- 'Browse Registry' allows the user to view all of the available structures, broken down by structure type, and maintenance agency.
- 'Lookup Structure' allows users to find structures that reference other structures. For example look-up 'Data Structure by Codelist' enables the user to navigate to a particular codelist, and then discover which Data Structures reference the Codelists.
- 'Search Structure' provides a free-text search for structures.

2.3 Recent History

FusionXL stores a list of recently viewed structures, allowing the user to easily retrieve recently viewed structures.

2.4 Multi Lingual Support

FusionXL displays structure names and descriptions in the selected language. FusionXL also makes it easy to add names and descriptions in new languages, by merging uploaded content with existing content.

2.5 Historical Modification

The Registry keeps track of every modification to every structure. FusionXL allows the user to view historical copies of the structure, providing an easy way to roll-back to an old copy if required.

2.6 Secure

Whilst Registry structures are all public, if FusionXL is connected to a secure Fusion Registry, then it will enforce that the user authenticates before structure modifications are possible.

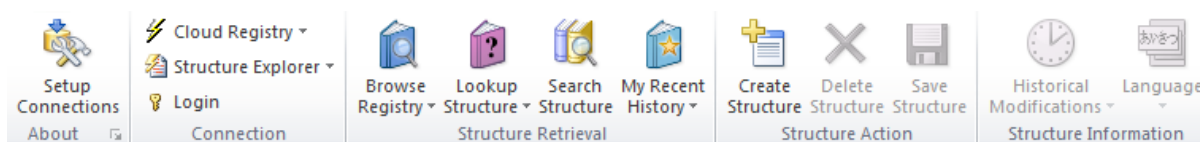
2.7 Structure Creation, and Deletion

FusionXL makes it easy to create new structures within the Fusion Registry, and to delete existing structures. The Registry will ensure all modifications are valid by performing validation checks, before the structure is processed and accepted into the Registry.

3 Connecting to Fusion Registry

To create a connection to a Fusion Registry instance, click on the 'Setup Connections' button. Please refer to the '[SetUp Guide](#)' for further information.

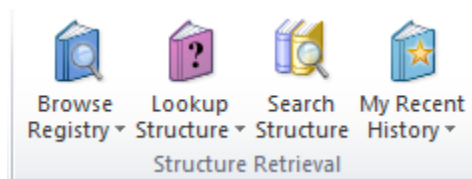
Once a connection has been defined, it is possible to select the connection by alias, and then choose the structure explorer in the 'Connection' section. The structure explorer will then present a number of menu items which can be used to browse the contents of the connected Fusion Registry.



The buttons are grouped into 3 distinct sections: browsing the Registry content (Structure Retrieval); modifying Registry content (Structure Action); and actions relevant to the selected structure (Structure Information).

The next few sections of this document detail the behaviour of each of these groups.

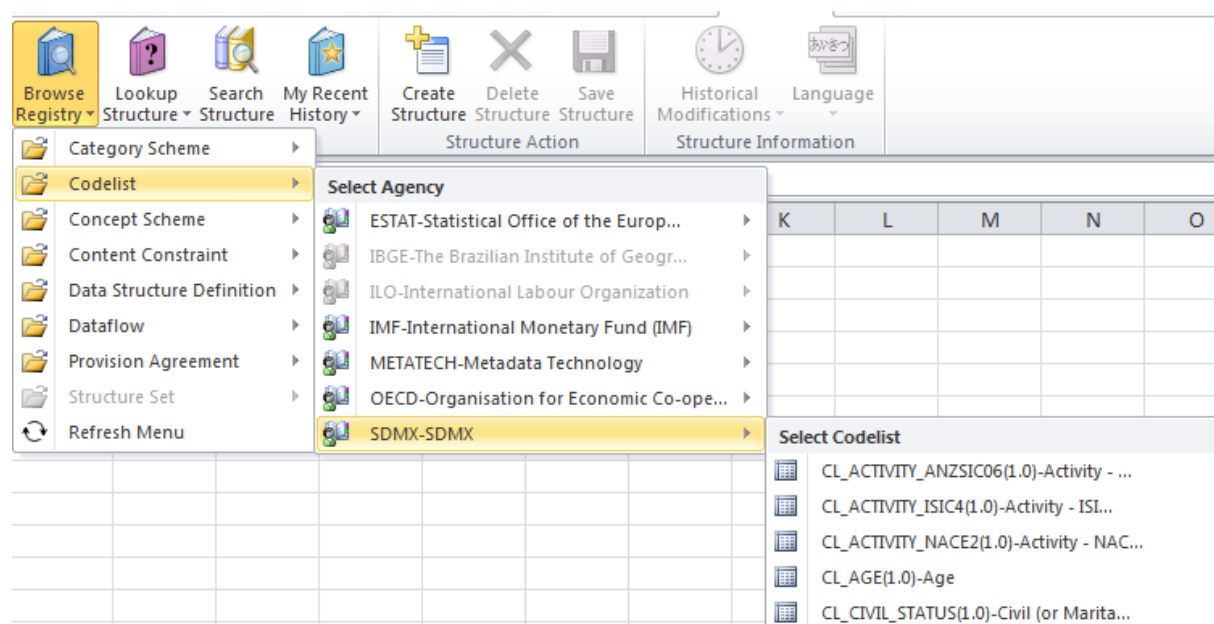
4 Structure Retrieval



There are 4 ways to retrieve structures from the Registry. Regardless of how a structure is retrieved, the retrieval of the structure will result in a new worksheet being opened. Thus, it is possible to have an Excel workbook with multiple worksheets, one for each retrieved structure.

4.1 Browse Registry

This button displays a hierarchical menu of all the structures in the connected Registry, broken down by structure type followed by Agency, as shown below.



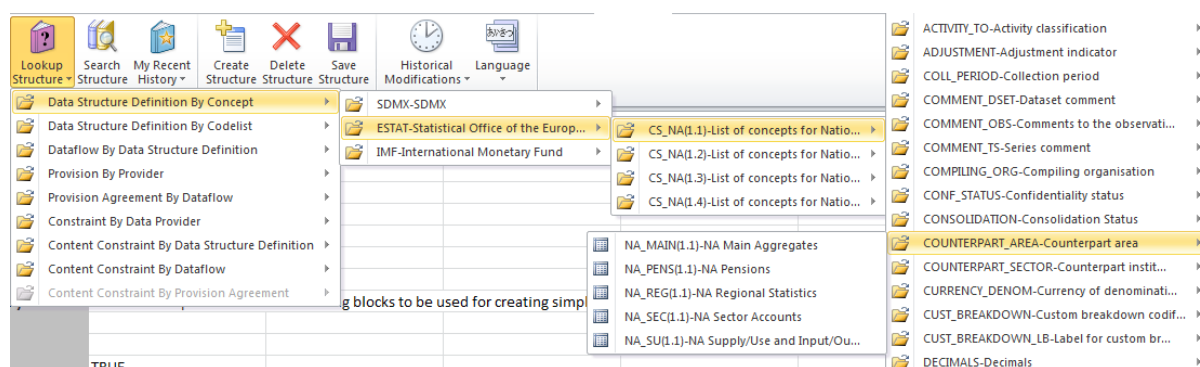
In the above image the CL_AGE(1.0) code list is highlighted for the SDMX Agency. On clicking this menu item, FusionXL will obtain this codelist from the connected Registry, and the resulting structure will be shown in the Excel worksheet, as shown below.

	A	B	C	D	E	F
1	FusionXL Version	1.0				
2	Type	Codelist				
3	Agency Id	SDMX				
4	Id	CL_AGE				
5	Version	1.0				
6	Name (en)	Age				
7	Description (en)	This code list provides a set of building blocks to be used for creating simple or complex code identifiers relating to the concept of "ag				
8	Start Date					
9	End Date					
10	Final	TRUE				
11	Structure URL					
12						
13	Id	Name Locale	Name	Description Locale	Description	Parent
14	Y	en	Year(s)			
15	M	en	Month(s)			
16	W	en	Week(s)			
17	D	en	Day(s)			
18	H	en	Hour(s)			

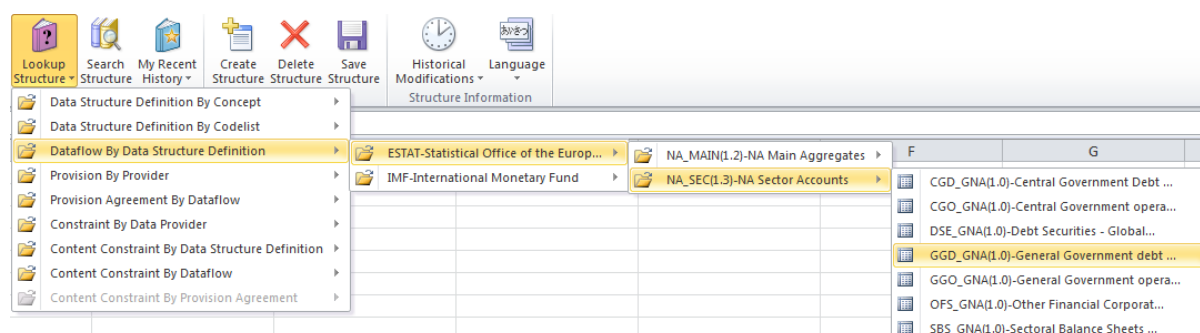


4.2 Lookup Structure

Lookup Structure provides a drop down menu of Registry content where the structures have been organised into folders depending on which structures reference them. An example of a lookup is *Data Structure by Concept*, where the menu structure allows the user to choose the Maintenance Agency of the Concept Scheme, followed by the Concept Scheme, followed by the Concept in the Concept Scheme, the final menu is a list of all the Data Structures which make use of that Concept by referencing it from either a Dimension or an Attribute. This menu structure is shown in the example below.



Another example is *Dataflow by Data Structure Definition (DSD)* where the menu structure is first the owning Agency of the DSD, followed by a list of all the DSDs for the Agency, followed by a list of all the Dataflows that reference the DSD. This menu structure is shown in the example below.

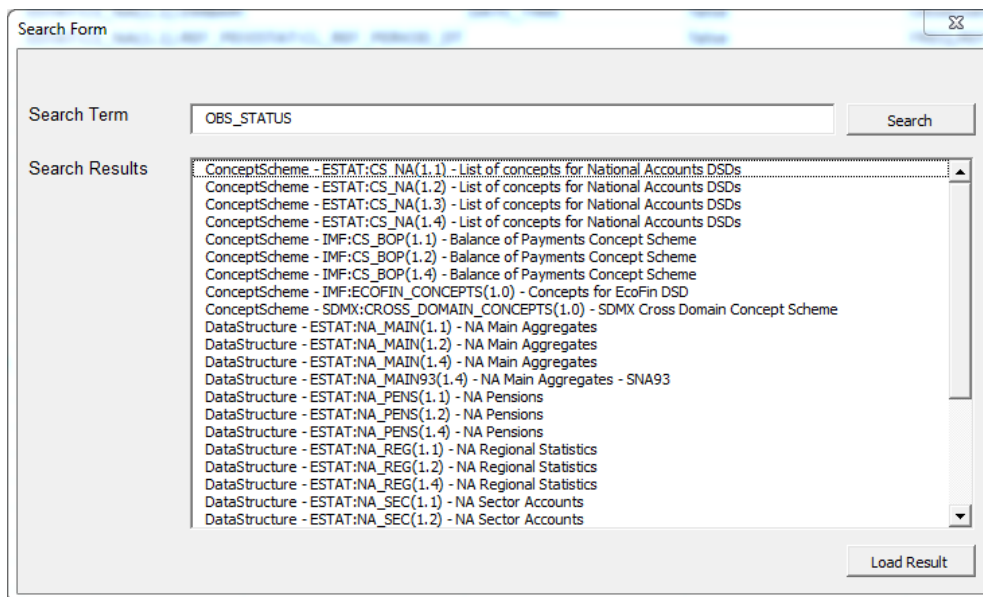


Lookup Structure is a very useful mechanism for quickly finding the desired structure, as well as a useful tool to discover which structures make use of other structures.

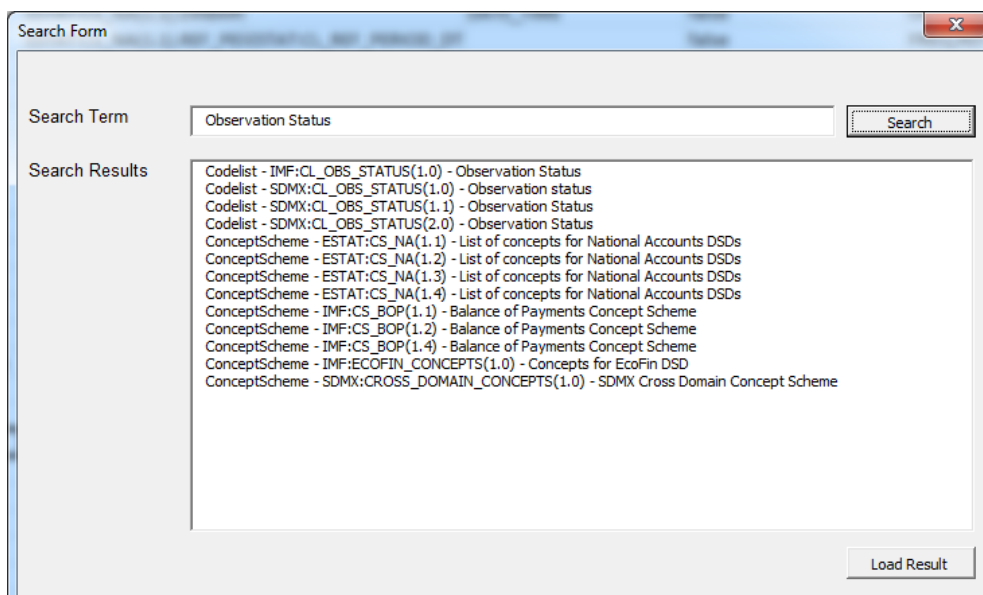
It should be noted that whilst then 'Browse Structures' menu displays all Registry content, 'Lookup Structure' may be showing a subset, depending on the Registry content.

4.3 Search Structure

The Search Structure provides a free text search of Registry content. The search terms are applied to Ids, names, descriptions of all Registry structures. A search on OBS_STATUS for example will bring back the concept schemes that contain the OBS_STATUS concept, and Data Structure Definitions with dimension Ids of OBS_STATS, as shown below.

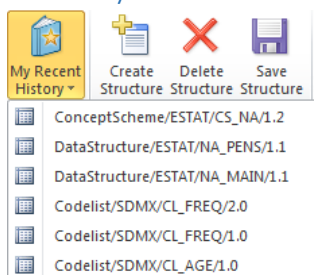


A search for the term Observation Status will bring back structures that contain this name or description.



To load a result, either double-click on a search result, or select the result and click the 'Load Result' button.

4.4 My Recent History



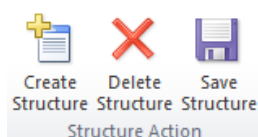
My recent history shows up to the last 10 structures retrieved against the current Registry connection with the most recently retrieved structure shown first in the menu. If the Registry connection is



changed in the Setup, the recent history is updated to show the history of structures retrieved for that connection.

By clicking on a Recent History item, the structure will be retrieved, and the item will be moved to the top of the Recent History menu.

5 Structure Action



Structure Action provides support for the creation, deletion, and modification of structures. Both Delete Structure and Save Structure will only be enabled if a structure has first been retrieved from the Registry. Authentication is performed for any Registry modification, and if authentication is required a login window will be displayed which will enable entry of a username and password.

5.1 Create Structure

The Create Structure menu button results, when clicked, in a pop up window. The new window requires information about the type of structure being created, the Id of the owning Agency, the structure Id, version and structure name. All fields on this form are mandatory. It is important to note that some structures cannot currently be created using FusionXL, in this instance the Fusion Registry Maintenance UI should be used.

Create Structure

Structure Type: Codelist

Owning Agency: SDMX

Structure Id: CL_NEW_CODES

Structure Version: 1.0

Name: My New Codelist

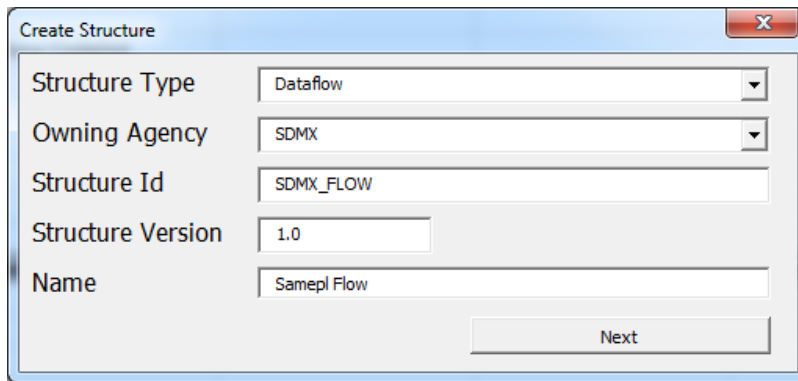
Create Structure

On clicking 'Create Structure' the structure will be created in the Registry, and a copy of the structure will be displayed in the worksheet, as shown below.

	A	B	C	D	E	F
1	FusionXL Version	1.0				
2	Type	Codelist				
3	Agency Id	SDMX				
4	Id	CL_NEW_CODES				
5	Version	1.0				
6	Name (en)	My New Codelist				
7	Description (en)					
8	Start Date					
9	End Date					
10	Final					
11	Structure URL					
12						
13	Id	Name Locale	Name	Description Locale	Description	Parent
14						
15						
16						
17						

For more complex structures, the structure creation is a two-step process. For example, a Dataflow requires a reference to a Data Structure Definition before it can be stored in the Registry. When creating a structure that requires additional information the 'Create Structure' button will turn into a 'Next' button, as shown below.

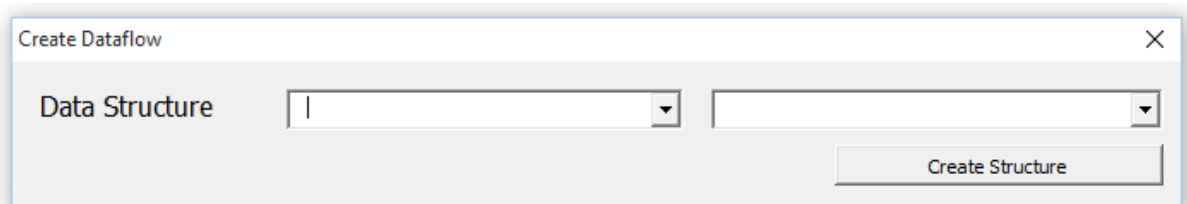
Please note that in the current version of FXL, it is not possible to create Dataflows or Provision Agreements.



The 'Create Structure' dialog box contains the following fields:

- Structure Type:** Dataflow (dropdown)
- Owning Agency:** SDMX (dropdown)
- Structure Id:** SDMX_FLOW (text input)
- Structure Version:** 1.0 (text input)
- Name:** Samepl Flow (text input)
- Next:** A button at the bottom right.

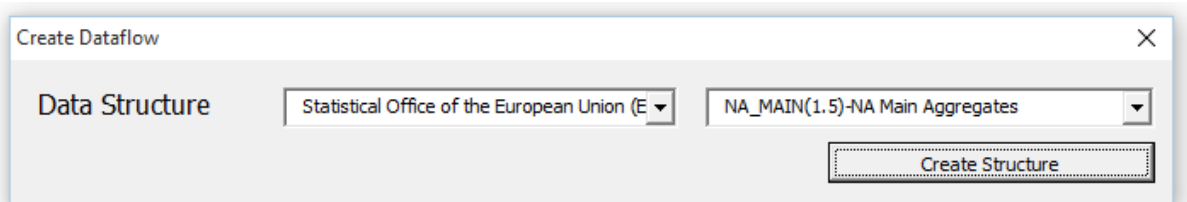
On clicking 'Next' a new window will be displayed to obtain the additional information, as shown below:



The 'Create Dataflow' dialog box contains the following fields:

- Data Structure:** Two empty dropdown menus.
- Create Structure:** A button at the bottom right.

For this example, the Data Structure reference must be defined by first selecting the owning Agency of the Data Structure, followed by the Data Structure. This is shown below:

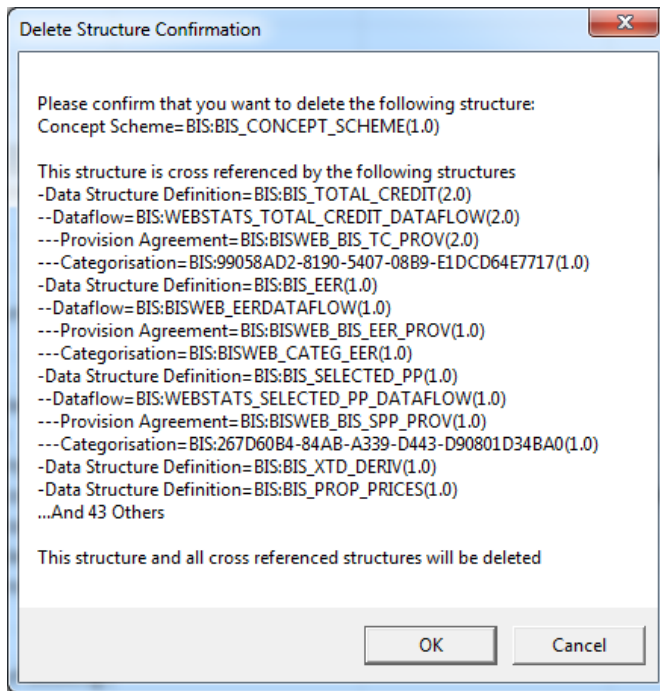


The 'Create Dataflow' dialog box shows the following selections:

- Data Structure:** Statistical Office of the European Union (E) (dropdown)
- Data Structure:** NA_MAIN(1.5)-NA Main Aggregates (dropdown)
- Create Structure:** A button at the bottom right.

5.2 Delete Structure

Delete Structure will only be enabled if a Structure has first been obtained from the Registry. On clicking Delete Structure the Registry will determine which structure(s) cross reference the structure for deletion, and a resulting confirmation window will be displayed.



If the structure is cross referenced, these will be shown in the confirmation window. On confirming the deletion request will be processed, and validation will take place to ensure the user has the correct credentials to perform such as request. If authentication is required then a login window will appear to ask for user details. In keeping with the Registry validation rules, the delete operation will be rejected if any of the structures to be deleted are marked as final.

If the deletion is successful, then the deleted structure and all structures that reference it will have been removed from the Registry. The structure will still be shown in the recent history, but will result in an error if an attempt is made to retrieve the structure.

5.3 Save Structure

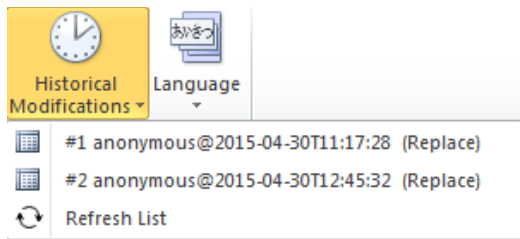
The save structure operation will submit the contents of the worksheet to the Registry to be stored. The save operation is only enabled if a structure has been created via the 'Create Structure' action, or if a structure has been obtained via the 'Structure Retrieval'.

On clicking save, FusionXL will transmit the worksheet data to the Registry for validation, if Authentication is required a login pop up will be displayed requesting user details.

6 Historical Modifications

The Historical Modifications menu item displays a list of all the modifications for the currently selected structure. Each time a structure is saved back to the Registry the modification is stored and backed up. This makes it possible to view, and roll back to any previous instance of the structure.

The Historical Modifications menu displays the username of the user that made the modification (or anonymous if the registry is public and no user was logged in), the date-time of the modification, and the modification action. The Historical Modifications menu is shown below.



It is possible to view a previous version of the structure by selecting from the Historical Modification menu. This will obtain the structure from the Registry, and display it in the workspace.



7 Multilingual Support

The FusionXL plugin will obtain structure names and descriptions in the language specified in the Language drop down. The Language menu will only display the available languages for the currently displayed structure. Where the structure does not have a name/description in the selected language, a different language will be displayed, defaulting to English.

The language is displayed either in brackets, or as a column header, as shown below.

	A	B	C	D	
1	FusionXL Version	1.0			
2	Type	Codelist			
3	Agency Id	SDMX			
4	Id	CL_FREQ			
5	Version	2.0			
6	Name (en)	Frequency			
7	Description (en)	This code list provides a set of values indicating the "frequency" of the data (e.g. weekly, monthly, quarterly). The con			
8	Start Date				
9	End Date				
10	Final	TRUE			
11	Structure URL				
12					
13	Id	Name Locale	Name	Description Locale	Description
14	A	en	Annual	en	To be used for data collected or dissen
15	S	en	Half-yearly, semester	en	To be used for data collected or dissen
16	Q	en	Quarterly	en	To be used for data collected or dissen
17	M	en	Monthly	en	To be used for data collected or dissen
18	W	en	Weekly	en	To be used for data collected or dissen
19	D	en	Daily	en	To be used for data collected or dissen
20	H	en	Hourly	en	To be used for data collected or dissen
21	B	en	Daily – businessweek	en	Similar to "daily", however there are n
22	N	en	Minutely	en	While N denotes "minutely", usually, t

It is possible to add text for a different language by simply changing the locale and entering the new text. In the example below the Frequency Codelist and the Annual Code have both been modified to add a German name.

6	Name (de)	Häufigkeit			
7	Description (en)	This code list provides a set of values indicating the "frequency" of the data (e.g			
8	Start Date				
9	End Date				
10	Final	TRUE			
11	Structure URL				
12					
13	Id	Name Locale	Name	Description Locale	
14	A	de	Jährlich	en	
15	S	en	Half-yearly, semester	en	
16	Q	en	Quarterly	en	

On saving this structure back to the Registry, this Codelist will be merged with the existing Codelist to add in the German names.

The Language drop down will then contain the additional option of German as an available language for the structure. When a different language is selected, the structure will be re-obtained from the Registry in the chosen language, and a new worksheet created.



8 Structure Format

Each Structure Type has a defined format in FusionXL, with the header information in rows 1 to 11, followed by the structure specific details.

Rows 1-11 are shown below:

	A	B
1	FusionXL Version	1.0
2	Type	Codelist
3	Agency Id	SDMX
4	Id	CL_NEW_CODES
5	Version	1.0
6	Name (en)	My New Codelist
7	Description (en)	
8	Start Date	
9	End Date	
10	Final	
11	Structure URL	

Row 1 (FusionXL Version) is information about the version of the FusionXL **communication protocol** (note this is not related to the version of the FusionXL plugin) , the Registry uses this information to know how to process the message. This value is generated by the Registry and should not be modified.

Row 2 (Type) determines the type of structure that the worksheet is defining. This value is generated by the Registry and should not be modified.

Rows 3-11 define information about the structure, and can all be modified as appropriate.

The layout for the remainder of the worksheet depends on the structure being created/modified.

The format for each structure type is described in the following sections, the rows 1-11 for each structure type are consistent, with the exception of cell B2, which denotes the Type of structure being described. In each of the follow sections, the value for Type is given.

8.1 Category Scheme

Type: CategoryScheme

A Category Scheme is a hierarchical scheme of Categories. Each category has a mandatory id and name, and an optional description.

13	Id	Name Locale	Name	Description Locale	Description
14	1	en	Demographic and social st	en	Covers work in population
15	1.1	en	Population and migration		
16	1.2	en	Labour	en	Covers statistics on labour
17	1.3	en	Education	en	Includes educational partic
18	1.4	en	Health		
19	1.5	en	Income and consumption		
20	1.6	en	Social protection		
21	1.7	en	Human settlements and h		
22	1.8	en	Justice and crime		
23	1.9	en	Culture		
24	1.10	en	Political and other commu		
25	1.11	en	Time use		
26	2	en	Economic statistics		

The Category Id can contain a period ‘.’ separator to indicate parentage. In the example above category with id ‘1.1’ is a child of category with id ‘1’. A Category may contain more than one ancestor,



for example '1.1.1' would be a valid id, denoting category '1.1' as the immediate parent, and category '1' as the grandparent.

8.2 Codelist & Value List

Type: Codelist & Value List

A Codelist is represented as a flat list of Codes, where each Code has a mandatory Id and Name, with an optional Description and Parent. The Parent column references the Parent Code by Id.

13	Id	Name Locale	Name	Description Locale	Description	Parent
14	A	en	Agriculture, forestry and fi			
15	A01	en	Crop and animal productio			A
16	A011	en	Growing of non-perennial			A01
17	A0111	en	Growing of cereals (excep			A011
18	A0112	en	Growing of rice			A011
19	A0113	en	Growing of vegetables and			A011
20	A0114	en	Growing of sugar cane			A011
21	A0115	en	Growing of tobacco			A011
22	A0116	en	Growing of fibre crops			A011
23	A0119	en	Growing of other non-per			A011

A Codelist can be represented as a hierarchy, but unlike a Category Scheme, the Code Id must not contain a period separator, and each Id must be unique. The parent code can be defined in the Parent column.

The reason for this difference is that SDMX represents a Category Scheme as an actual hierarchy, which allows for Categories at each level of the hierarchy to share the same id (if the parent is different). A Codelist is represented as a flat list, ids must be unique, and parent Codes are referenced by id.

8.3 Concept Scheme

Type: ConceptScheme

A Concept Scheme is represented as a flat list of Concepts, where each Concept has a mandatory Id and Name, with an optional Description and Parent. The Parent column references the Parent Concept by Id.

13	Id	Name Locale	Name	Description Locale	Description	Parent
14	ACCESSIBILITY	en	Accessibility	en	The ease and the conditions under which statistical i	
15	ACCURACY	en	Accuracy	en	Closeness of computations or estimates to the exact	
16	ACCURACY_OVERALL	en	Accuracy - overall	en	Assessment of accuracy, li ACCURACY	
17	NONSAMPLING_ERR	en	Accuracy - non-sampling e	en	Error in sample estimates ACCURACY	
18	SAMPLING_ERR	en	Accuracy - sampling error	en	That part of the differenceACCURACY	
19	ADJUSTMENT	en	Adjustment	en	The set of procedures employed to modify statistical	
20	ADJUST_CODED	en	Adjustment - coded	en	Type of adjustment used, ADJUSTMENT	
21	ADJUST_DETAIL	en	Adjustment - detail	en	Textual description of the ADJUSTMENT	
22	AGE	en	Age	en	The length of time that a person has lived or a thing t	
23	BASE_PER	en	Base Period	en	The period of time used as the base of an index num	
24	CIVIL_STATUS	en	Civil status	en	Legal, conjugal status of each individual in relation tc	
25	CLARITY	en	Clarity	en	The extent to which easily comprehensible metadata	
26	CLASS_SYSTEM	en	Classification system	en	Arrangement or division of objects into groups basec	
27	COHERENCE	en	Coherence	en	Adequacy of statistics to be combined in different w	
28	COHER_X_DOM	en	Coherence - cross-domain	en	Extent to which statistics a COHERENCE	

8.4 Content Constraint

Type: ContentConstraint



A Content Constraint, which is used to define either restrictions on allowable data, or what data are present, come in two flavours: a Series Constraint (**KeySet**) which defines one or more Series Keys or partial keys; and a **Cube Region** Constraint which defines one or more Code Ids per constrained Dimension/Attribute.

A Content Constraint may also be attached to a Data Structure, Dataflow, Provision Agreement, or Data Provider.

The following image shows the high-level description of a Content Constraint in terms of which structure(s) it attaches to, whether it is defining allowable data, or actual data, and which type of constraint it is.

13	Attachment Structure	DataSet	
14	Attachment(s)	METATECH:LFS_SOE_EXTERNAL(1.0)	
15	Defining Data Present	false	
16	Constraint Type	KeySet	

Row 13 defines the type of structure that is being constrained, the allowable values are:

- DataSet
- Dataflow
- ProvisionAgreement
- DataProvider

Row 14 references the actual structure(s) of the given type that are being constrained. The reference is broken down into the following parts:

AgencyId:Id (version)

If the Constraint is attaching to multiple structures of the given type, then the attachments are given as a comma separated list, for example:

METATECH,LFS_SOE_EXTERNAL(1.0), METATECH,LFS_SOE_INTERNAL(1.0)

Row 15 defines if the constraint is defining data that is known to be present (**true**) or if it is being used to restrict allowable content (**false**).

Row 16 defines the type of Content Constraint and can be set to **KeySet** or **Cube Region**.

Row 18 onwards are used to specify the constrained Series Keys or Code Ids, depending on the Constraint Type, these are described in the following sections.

8.4.1 KeySet - Constraining Series Keys

A KeySet Constraint defines the allowed or restricted content by defining it in terms of the series key. A series key comprises of a Code Id per Dimension. Wildcarding can be used by specifying the Code Id as '*'. Each Series Key is defined as either being Included (present/allowed) or Excluded (not present/not allowed).

An example of a KeySet constraint consisting of three Included series is shown below.

18	KeySet						
19	Dimension Id	FREQ	REF_AREA	SEX	EDUCATION_LEVEL	OCCUPATION	LFS_MEASURE
20	Included	A	UK	M	UPS	MGR	UEMPLOY
21	Included	A	EU	*	*	*	*
22	Included	A	EU27	T	UPS	MGR	UEMPLOY



8.4.2 Cube Region - Constraining Code Values

A Cube Region Constraint defines the allowed or restricted content by defining it in terms of Code Ids per Component (Dimension or Attribute). The Cube Region only needs to specify the Components it is restricting, so it can be a subset of the DSD's components. If the constraint is attached to multiple DSDs (either directly, or indirectly) then the Component Id must be a component which is shared amongst all DSDs.

Each Component which is restricted must define one or more code ids, which should be marked as being Included (present/allowed) or Excluded (not present/not allowed).

An example of a Cube Region constraint consisting of three constrained Component Ids is shown below.

18	Cube Region			
19	Component Id	AGE	GEO	OBS_STATUS
20	Included/Excluded	Included	Included	Excluded
21	Code Id(s)	Y15-19	UK	e
22		Y15-24		f
23		Y15-39		i
24				n
25				p
26				r
27				s
28				u
29				z

8.5 Data Structure Definition

Type: DataStructure

A Data Structure Definition is made up of three sections:

Dimensions

Attributes

Component Restrictions

Each section is described in the following sections.

8.5.1 Dimensions Section

The Dimensions section contains a flat list of dimensions, each with a mandatory Id, Concept Reference, and Dimension Type. Each Dimension may optionally reference a Codelist and define a Text Type.

Concept references are in the format:

AgencyId:SchemeId(SchemeVersion).ConceptId

Codelist reference are in the format:

AgencyId:CodelistId(Version)

The allowable Text Types are given in Annex 1.



Dimension Types can be any of the following values:

PrimaryMeasure – There must be a PrimaryMeasure Dimension, and there must not be more than one. This dimension must be un-coded and have an Id of OBS_VALUE

Dimension

MeasureDimension

TimeDimension – There must only be one TimeDimension, and it must be un-coded. It is recommended that the TimeDimension is the last Dimension

An example of Dimensions is shown in the image below:

	A	B	C	D	E	F	G	H
13	Dimension Id	Concept	Codelist	Text Type	Dimension Type	Group1	Group2	SiblingGroup
14	OBS_VALUE	ESTAT:CS_NA(1.1).OBS_VA		DOUBLE	PrimaryMeasure			
15	FREQ	ESTAT:CS_NA(1.1).FREQ	SDMX:CL_FREQ(1.0)	STRING	Dimension	Y	Y	
16	ADJUSTMENT	ESTAT:CS_NA(1.1).ADJUST	ESTAT:CL_ADJUSTMENT(1.	STRING	Dimension	Y	Y	Y
17	REF_AREA	ESTAT:CS_NA(1.1).REF_AR	IMF:CL_AREA(1.1)	STRING	Dimension	Y	Y	Y
18	COUNTERPART_AREA	ESTAT:CS_NA(1.1).COUNT	IMF:CL_AREA(1.1)	STRING	Dimension	Y		Y
19	REF_SECTOR	ESTAT:CS_NA(1.1).REF_SE	ESTAT:CL_SECTOR(1.1)	STRING	Dimension	Y		Y
20	COUNTERPART_SECTOR	ESTAT:CS_NA(1.1).COUNT	ESTAT:CL_SECTOR(1.1)	STRING	Dimension	Y		Y
21	ACCOUNTING_ENTRY	ESTAT:CS_NA(1.1).ACCOU	IMF:CL_ACCOUNT_ENTRY(STRING	Dimension	Y	Y	Y
22	STO	ESTAT:CS_NA(1.1).STO	ESTAT:CL_NA_STO(1.1)	STRING	Dimension	Y		Y
23	INSTR_ASSET	ESTAT:CS_NA(1.1).INSTR_	ESTAT:CL_INSTR_ASSET(1.	STRING	Dimension	Y		Y
24	ACTIVITY	ESTAT:CS_NA(1.1).ACTIVIT	ESTAT:CL_ACTIVITY(1.1)	STRING	Dimension	Y	Y	Y
25	EXPENDITURE	ESTAT:CS_NA(1.1).EXPEND	ESTAT:CL_COICOP(1.1)	STRING	Dimension	Y		Y
26	UNIT_MEASURE	ESTAT:CS_NA(1.1).UNIT_M	IMF:CL_UNIT(1.1)	STRING	Dimension	Y		Y
27	PRICES	ESTAT:CS_NA(1.1).PRICES	ESTAT:CL_NA_PRICES(1.1)	STRING	Dimension	Y		Y
28	TRANSFORMATION	ESTAT:CS_NA(1.1).TRANSF	ESTAT:CL_TRANSFORMATI	STRING	Dimension	Y		Y
29	TIME_PERIOD	ESTAT:CS_NA(1.1).TIME_PI		OBSERVATION	TimeDimension			

The columns **F-G** define dimension groupings, where dimensions included in the group are depicted by the relevant cell having the content 'Y'. The alias for each grouping is the column header, so in this example the groups 'Group1', 'Group2', and 'SiblingGroup' exist. Groups can be referenced by attributes, as described in the following section. If a group alias starts with the word 'Group' and is followed by a numerical value (e.g. 'Group1') then the generated attribute will reference all of the dimensions included in the group by Id (this is the equivalent SDMX attachment type of 'Dimension Group'). If the group alias is anything else, then the generated Data Structure will include an explicit Group, with the id being set to the group alias. Any Dimensions that reference an explicit group will reference the Group by Id in the generated Data Structure Definition.

8.5.2 Attributes Section

The Attributes section contains a flat list of Attributes, each with a mandatory Id and Concept Reference. An Attribute must also define if it is Mandatory and what it is attached to. Each Attribute may optionally reference a Codelist and define a Text Type.

Concept references are in the format:

AgencyId:SchemeId(SchemeVersion).ConceptId

Codelist reference are in the format:

AgencyId:CodelistId(Version)



Attribute Attachment must be one of the following:

- Dataset
- Observation
- Group Alias (as defined in Dimensions Section)

An example of Attributes is shown in the image below:

Attribute Id	Concept	Codelist	Text Type	Mandatory	Attribute Attachment
OBS_STATUS	ESTAT:CS_NA(1.1).OBS_ST	SDMX:CL_OBS_STATUS(1.1)	STRING	true	Observation
CONF_STATUS	ESTAT:CS_NA(1.1).CONF_S	SDMX:CL_CONF_STATUS(1.1)	STRING	true	Observation
COMMENT_OBS	ESTAT:CS_NA(1.1).COMME		STRING	false	Observation
EMBARGO_DATE	ESTAT:CS_NA(1.1).EMBARC		DATE_TIME	false	Observation
REF_PERIOD_DETAIL	ESTAT:CS_NA(1.1).REF_PE	ESTAT:CL_REF_PERIOD_DT	STRING	false	Group1
TIME_FORMAT	ESTAT:CS_NA(1.1).TIME_F	SDMX:CL_TIME_FORMAT(1.1)	STRING	false	Group1
TIME_PER_COLLECT	ESTAT:CS_NA(1.1).TIME_P	IMF:CL_TIME_COLLECT(1.0)	STRING	false	Group1
REF_YEAR_PRICE	ESTAT:CS_NA(1.1).REF_YE		INTEGER	false	Group1
DECIMALS	ESTAT:CS_NA(1.1).DECIMA	SDMX:CL_DECIMALS(1.0)	STRING	true	Group1
TABLE_IDENTIFIER	ESTAT:CS_NA(1.1).TABLE_I	ESTAT:CL_NA_TABLEID(1.0)	STRING	false	Group1
TITLE	ESTAT:CS_NA(1.1).TITLE		STRING	false	Group1
UNIT_MULT	ESTAT:CS_NA(1.1).UNIT_M	SDMX:CL_UNIT_MULT(1.0)	STRING	true	Group1
LAST_UPDATE	ESTAT:CS_NA(1.1).LAST_UI		DATE_TIME	false	Group1
COMPILING_ORG	ESTAT:CS_NA(1.1).COMPIL	IMF:CL_ORGANISATION(1.1)	STRING	false	Group1
COMMENT_DSET	ESTAT:CS_NA(1.1).COMME		STRING	false	Dataset
COMMENT_TS	ESTAT:CS_NA(1.1).COMME		STRING	false	Group1

8.5.3 Component Restriction Section

Each Component (Dimension or Attribute) may additionally have added restrictions on their content. This would usually not be the case for a Coded Component, as the Codelist provides the list of allowable enumerated content.

Each Component that has additional restrictions should be listed in this section, along with the value for each restriction. An example is shown below.

Component Restrictions	Min Length	Max Length	Start Value	End Value	Interval	Time Interval	Decimals	Pattern	Sequence
FREQ	1	1							
ADJUSTMENT	1	2							
REF_AREA	2	4							
COUNTERPART_AREA	2	4							
REF_SECTOR	2	7							
COUNTERPART_SECTOR	2	7							
ACCOUNTING_ENTRY	1	2							
STO	1	8							
INSTR_ASSET	1	7							
ACTIVITY	1	9							

8.6 Dataflow

Type: Dataflow

A Dataflow simply references a single Data Structure Definition, the reference is by:

AgencyId:Id(Version)

An example is shown below.

12		
13	Data Structure	METATECH:LFS_STRUCTURE1(1.0)
14		



8.7 Provision Agreement

Type: ProvisionAgreement

A Provision Agreement references a Dataflow and a Data Provider.

The Dataflow reference takes the following format:

AgencyId:Id(Version)

The Data Provider reference takes the following format:

AgencyId:DATA_PROVIDERS(1.0).ProviderId

An example is shown below.

12			
13	Data Flow	METATECH.ESAT:cens_01neisco(1.0)	
14	Data Provider	METATECH:DATA_PROVIDERS(1.0).AT-NSI	
15			

