SDMX Constructor: User Manual

International Labour Organization: Department of Statistics

2023-06-12

Contents

Preface

Welcome to the **ILO's SDMX Constructor User Manual!**

SDMX Constructor[1] is a powerful desktop software tool that helps users model aggregate data per the SDMX standards [2] . It eases generating and editing SDMX artefacts and ultimately supports data availability and access through SDMX-compliant data portals (such as ones built with .Stat Suite[3]).

SDMX Constructor is one of the tools that comprise the ILO SDMX toolkit [4], alongside SMART (Statistical Metadata-driven Analysis and Reporting Tool) and the SDMX Excel Add-in.

This user manual for the SDMX Constructor provides step-by-step instructions on using the tool in a user-friendly and accessible manner. It provides an in-depth understanding of SDMX Constructor's features and functionalities. It is an essential resource for anyone using the tool to manage and share data following the SDMX standards.

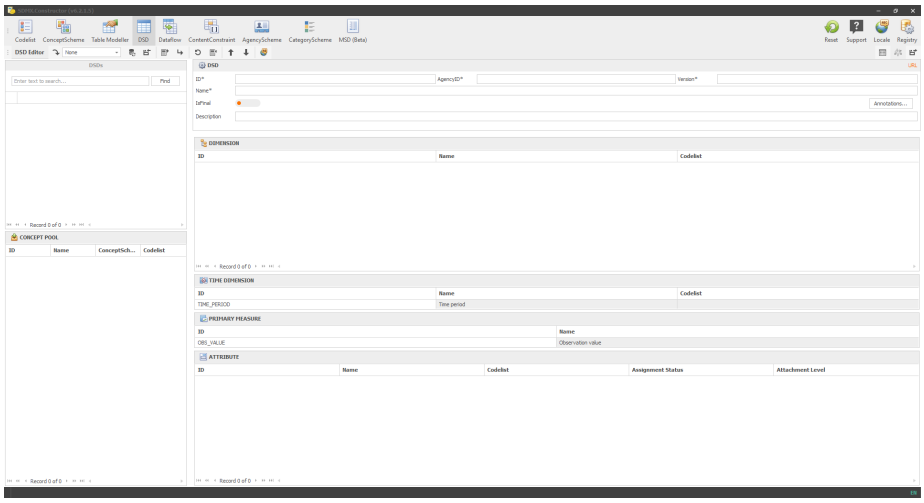
Below is a screenshot of the tool's landing page as an example of its user interface.

[1] Previously known as DSD Constructor, this software was used for creating and editing Data Structure Definitions (DSDs) and their related artefacts such as concept schemes and code lists. Since then, the software has undergone significant improvements and additions, resulting in a name change to SDMX Constructor. The new name reflects the software's expanded capabilities and makes it more relevant and recognizable to its target audience (https://ilostat.github.io/dsdc/).

[2] SDMX has also been published as an ISO International Standard. https://www.iso.org/obp/ui/fr/#iso:std:iso:17369:ed-1:v1:en and https://sdmx.org/?page_id=5008

[3] To learn more about the .Stat Suite platform please refer to the documentation here: https://siscc.org/stat-suite/. The SDMX Constructor has been optimised to seamlessly integrate with the .Stat Suite platform, as its back-end client.

[4] The ILO offers several tools for Statistical Data and Metadata eXchange (SDMX). https://ilostat.ilo.org/resources/sdmx-tools/



Click here to enlarge the image

Audience and use cases

The SDMX Constructor is designed to meet the needs of a wide range of users, from beginners who are new to SDMX to data managers who need advanced capabilities for managing large and complex datasets. In addition to these user types, there may be other groups of users with specific requirements or use cases. This user manual provides guidance and instructions for using the SDMX Constructor, focusing on the needs of these various user types.

SDMX beginners: These users are new to SDMX and want to use the SDMX Constructor to view, edit, and create new SDMX structural artefacts. They may need assistance understanding the SDMX concepts, terminology, and the software's user interface.

For SDMX beginners, this manual explains using the SDMX Constructor to access, view, edit, and create SDMX structural artefacts from SDMX registries. While one can find information on SDMX concepts and terminology from other sources, this manual focuses explicitly on the SDMX Constructor. The SDMX beginners may be keen to know how to view and access SDMX artefacts from SDMX registries, as well as how to connect to an SDMX registry through the SDMX Constructor.

Data managers: These users are likely familiar with the SDMX concepts and terminology and may require information about advanced functionalities offered by the SDMX Constructor. For example, they may want to use the SDMX Constructor as a backend client to manage SDMX artefacts for the .Stat Data Lifecycle Manager (DLM). For such cases, they may also use the SDMX

Constructor to build the initial structural metadata when creating a new .Stat Suite instance.

There are several topics that data managers may be interested in learning when it comes to creating SDMX artefacts from scratch. Firstly, they may want to start by setting up a registry as a local folder. Next, they can prepare inputs and create several artefacts, including the AgencyScheme, ConceptScheme, and Codelist and the DSD, Dataflow, ContentConstraint, and CategoryScheme. After creating these artefacts, they may want to learn how to upload the XML file to the Data Lifecycle Manager (DLM). Additionally, they may want to know how to access SDMX artefacts from SDMX registries and connect to an SDMX registry for editing SDMX artefacts directly in the DLM.

SDMX metadata managers: These users manage SDMX artefacts and ensure their accuracy and consistency. They may use the SDMX Constructor to model data and modify SDMX structural artefacts, including translating SDMX artefacts in various languages, managing annotations and creating Metadata Structure Definition (MSD).

Scope and assumptions

This comprehensive manual provides a step-by-step guide on getting started with the software, including installing it and creating and managing SDMX artefacts. The manual includes a detailed overview of the tool's user interfaces, including menu items, navigation, and other essential features. The manual does not, however, delve into the intricacies of the SDMX standard itself. Describing the SDMX standard is beyond the scope of this manual, as it is an extensive and complex topic requiring more in-depth discussion and is available elsewhere. Instead, the manual assumes that readers are familiar with the SDMX standard and focuses on explaining how to use the tool within that context.

Overview

The chapters in this manual build up toward providing comprehensive guidance that covers topics users need to know to get started with the application, understand its navigation, and use its features and functionalities to model data effectively. The manual includes the following chapters.

- Chapter 1: Benefits of SDMX Constructor
- Chapter 2: Getting Started
- Chapter 3: User Interface
- Chapter 4: Using SDMX Constructor
- Chapter 5: Special Topics

Contact information

For more information and to seek basic technical assistance or support on the tool, please contact the International Labour Organization - Department of Statistics at sdmx.support@ilo.org.

Chapter 1

Benefits of SDMX Constructor

SDMX Constructor is a software tool that offers a range of features and functionalities designed to streamline and optimise the SDMX workflow, making it an essential tool for statistical organisations, data providers, and other stakeholders involved in the production and dissemination of statistical data.

One of the key benefits of SDMX Constructor is that it allows users to access and query SDMX structural artefacts from SDMX registries (through APIs in the background) in a desktop environment. This makes it easy for users to quickly and easily search for and view essential components of SDMX structural metadata, such as Data Structure Definitions (DSDs) and dataflows.

In addition to its API capabilities, SDMX Constructor offers two other essential features. The first is that it helps users model data following the SDMX standards. For example, SDMX Constructor makes it easy for users to define data structures and create data flows following SDMX standards. This is essential for data providers who need easy-to-use tools to ensure that their data can be shared and used by others in a consistent and standardised way.

The second essential feature of SDMX Constructor is that it enables users to create and edit SDMX artefacts in a user-friendly environment. This includes creating and editing code lists, concept schemes, DSDs, dataflows, content constraints, category schemes and Metadata Structure Definitions (MSDs)[1]. SDMX Constructor provides a user-friendly interface that makes it easy for users to create and modify these artefacts without needing to be an expert in SDMX.

[1]These SDMX terms are detailed in the SDMX Glossary, available on the official SDMX website. The SDMX Glossary is a comprehensive reference guide that provides definitions and explanations for all the key terms and concepts used in SDMX. You can access the SDMX Glossary by visiting the following link: https://sdmx.org/?sdmx_news=sdmx-glossary.

Finally, SDMX Constructor supports data availability and access through online data portals. Using SDMX Constructor, data providers can ensure their data is available and accessible through online data portals built on SDMX standards. This allows data users to easily access and use the data they need for their research, analysis, and other activities.

Chapter 2

Getting Started

In this chapter, you will find information about the installation process of SDMX Constructor in a Windows environment. Specifically, it will cover the system requirements necessary to install the software, the steps required to download the executable file and the subsequent installation process. The chapter will also explain the software update process, providing practical guidance on keeping your SDMX Constructor current.

2.1 System requirements

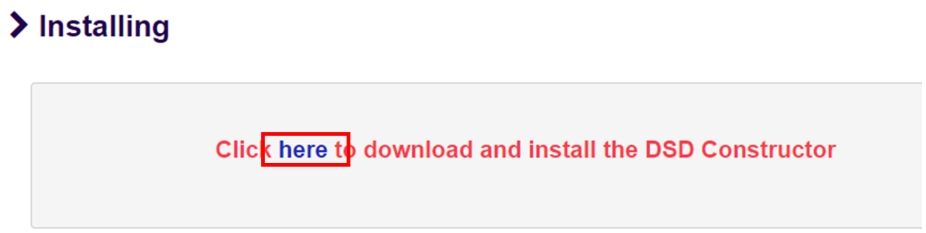
Before installing any desktop software, ensuring the computer meets the necessary system requirements is essential. These requirements can include hardware specifications like processor speed, memory, and storage capacity, as well as software prerequisites like operating system version and other dependencies. Find below the minimum system requirements for installing and running SDMX Constructor.

- Windows 7 or later in both 32- and 64-bit environments
- .NET 4.5.2 or later with all updates and patches
- 200MB disk space for the software

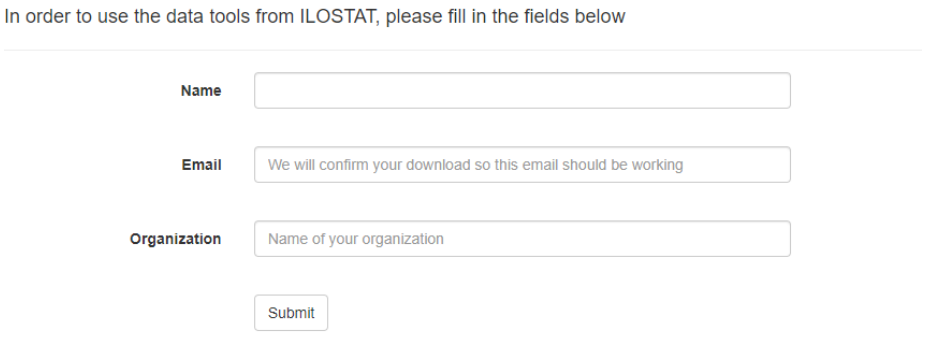
2.2 Installation

Step 1: Visit the site https://ilostat.github.io/dsdc/ and click on the INSTALL menu option (as shown below).

Step 2: In the resulting page, click on "here" (as shown below)



Step 3: An interface (as shown below) will open, requiring you to enter a few details (your name, your email, and the name of your organisation (if applicable)).



Step 4: The download will start when you fill in the form and submit the information. The software installer file, named "ILOSTAT_DSD_Constructor_Setup.exe", will be downloaded.

Step 5: Double-click on this file to begin the installation process. Select 'Install' on the security message and wait for the installation process to complete.

2.3 Software updates

The SDMX Constructor updates itself to the latest version (if available) using the ClickOnce Deployment technique. This method ensures a streamlined installation process with minimal user involvement. An internet connection is required when launching the application to initiate the update. The software will check for available updates and automatically download and install them in the background without requiring any input from the user. This feature helps to ensure that users always have access to the latest version of the software, with the latest bug fixes, performance improvements, and new features.

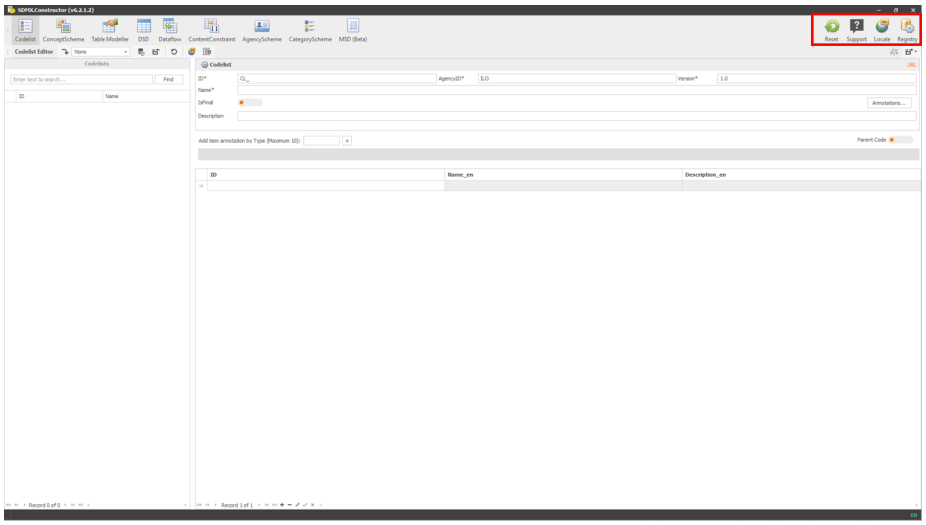
Chapter 3

User Interface

This chapter will provide you with a comprehensive guide on the main aspects of the SDMX Constructor's user interface. It will cover three main topics. The first topic will be an overview of the user interface, including a detailed explanation of the various menu options and the working area. The second topic will focus on the inputs and outputs of SDMX Constructor. It will cover the different input and output options available. The third topic will provide an overview of the translation functionality in SDMX Constructor.

3.1 General Functions Menu

In the top right corner (as highlighted below), the first group of buttons (General Functions) shows the items applicable to the whole tool. They include Reset, Support, Locale and Registry. See Table 3.1 for a brief overview of the menu items.



Click here to enlarge the image

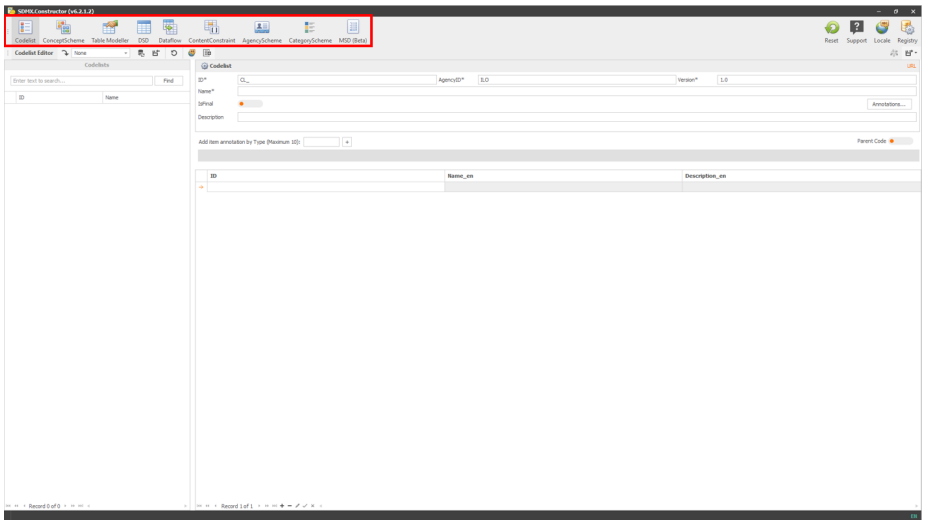
Table 3.1: A bird's-eye view of the menu items in the top right corner (General Functions)

Menu item	What to expect: A bird's-eye view of General Functions
Reset	The Reset button resets the tool to its default settings. It removes all the inputs and initiates a fresh start.
Support	The Support button launches the default email application on a computer, with a new email message addressed to the support team for the tool, ready to be composed and sent.
Locale	The Support button launches the default email application on a computer, with a new email message addressed to the support team for the tool, ready to be composed and sent.

Menu item	What to expect: A bird's-eye view of General Functions
Registry	The Registry button provides several options for users to configure their settings. These are grouped in the following tabs: SDMX Registry, Google API, DeepL and Proxy. In the SDMX Registry tab, users can specify the connection details of the SDMX registry (either a local folder or a local instance (localhost) or an online platform (with SDMX-compliant API). Also, using connection credentials, users can push SDMX artefact to the .Stat Suite through the 'DLM Connection' option available in the SDMX Registry tab. Using the Google API and DeepL tabs, users can enter authentication credentials for automated translation using Google Translation or DeepL API. The Proxy tab allows specifying the proxy settings for the internet connection if a proxy is needed.

3.2 Editors Menu

The second group of menu items (Editors) shows the options in the top left corner (as highlighted below). They include Codelist, ConceptScheme, Table Modeller, DSD, Dataflow, ConceptConstraint, AgencyScheme, CategoryScheme and MSD. Each menu item is an entry point for creating and editing a specific artefact. Clicking on any of the options will reveal more particular options below. See Table 3.2 for a brief overview of the menu items (Editors).



Click here to enlarge the image

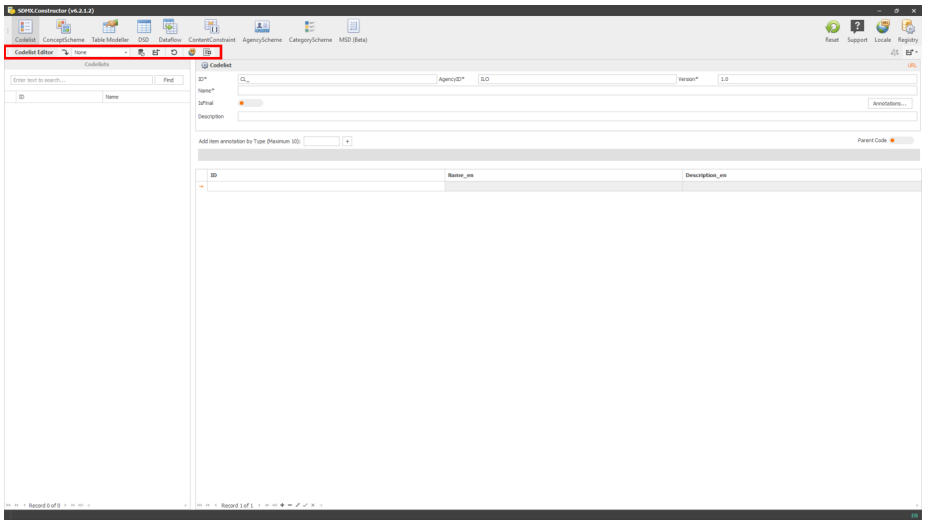
Table 3.2: A bird's-eye view of the menu items in the top left corner (Editors)

Menu item	What to expect: A bird's-eye view of the menu items in the top left corner (Editors)
Codelist	The Codelist button shows an interface to create or edit the list of Concepts (dimensions/attributes of a modelled dataset), each having a list of possible values or codes. The codes have unique IDs and (multilingual) labels and may have other information such as order and hierarchy.
ConceptScheme	The ConceptScheme button shows an interface to create or edit the Concepts (dimensions/attributes of a modelled dataset). Each concept has an ID and (multilingual) labels and may have other information.
Table Modeller	The Table Modeller button shows the options to generate the SDMX artefacts by recreating the statistical table using the Concepts (dimensions/attributes of a modelled data set) with an intuitive user interface (drag and drop).
DSD	The DSD button abbreviating Data Structure Definition shows an interface to create or edit the structure of a modelled dataset. It specifies the dimensions, attributes and their possible values or codes.

Menu item	What to expect: A bird's-eye view of the menu items in the top left corner (Editors)
Dataflow	The Dataflow button shows an interface to create or edit a subset and representation of a DSD and describes which dimensions to show on rows or columns.
ConceptConstraint	The ConceptConstraint button shows an interface to create or edit combinations of codes that are allowed for various artefacts of the dataflows.
AgencyScheme	The AgencyScheme button shows an interface to create or edit a hierarchical list of agencies or organisations. Each agency or organisation is identified by a unique code (ID), a name, and other descriptive information.
CategoryScheme	The CategoryScheme button shows an interface to create or edit a CategoryScheme, which is essentially a list of categories organised in a hierarchical structure. Each category is identified by a unique code (ID), a name, and other descriptive information.
MSD	The MSD button abbreviating Metadata Structure Definition shows an interface describing the structure of reference metadata associated with a data set. It specifies the elements and attributes that can be used to represent metadata concepts and the relationships between them.

3.3 Editor Ribbon Menu

The third group of menu items (Editor Ribbon) are in the top left corner and below the second group of menu items (as highlighted below).



Click here to enlarge the image

It contains icons that represent the most commonly used features of the selected options above, and hovering over each icon will reveal a tooltip that describes its function. The options available in this group of menu items change according to the menu item selected on the top menu.

Below are the icons representing the menu items (Editor Ribbon) and their functions. Users can change the size of these icons by right-clicking on any of them, selecting Customize, and selecting the appropriate setting in Options.

Table 3.3: A bird's-eye view of the menu items in the Editor Ribbon

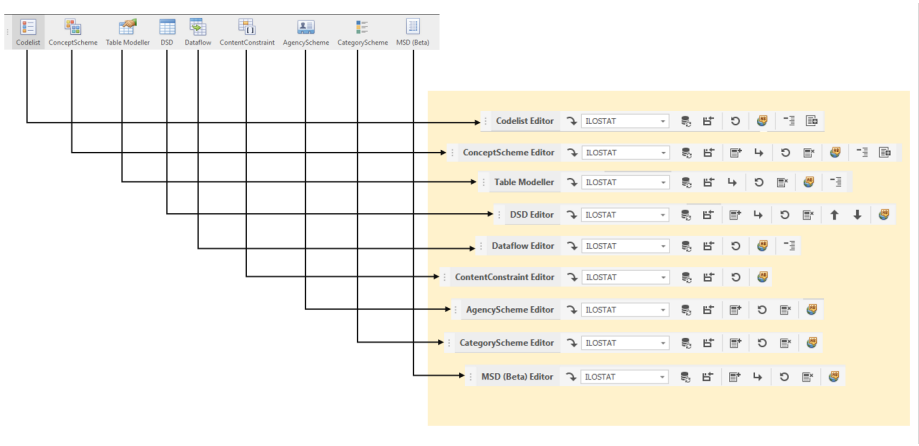
Menu item	Names and functions
	'Load from Registry' is the registry selector. It allows users to select from a dropdown list to load a registry. This registry could be a local folder or a local instance (localhost), or an online platform (with SDMX-compliant API).
	Refresh the registry.
	Import. Allows to import an SDMX XML file locally.
	Reset the current Editor inputs.

Menu item	Names and functions
	Translation Service. Provide inline translations for names and descriptions via translation API or manually if needed.
	Collapse DSDs.
	Bulk load. Allows to upload multiple items (if available) by copying and pasting.
	Add New Concept (e.g., Agency or Category). Depending on which Editor is active, e.g., it can help add a new agency if it is for AgencyScheme.
	Load Concept.
	Delete Selected Concepts.
	Move Up.
	Move Down.

If a user chooses “Codelist” from the top menu, the corresponding menu items in the third group will appear as “Load from registry” (from the dropdown menu), “Refresh the registry”, “Import”, “Reset the current Editor inputs”, “Translation Service”, and “Bulk load”.

However, suppose a user selects “ConceptScheme” from the top menu instead. In that case, the user will see three additional options: “Add New Concept”, “Load Concept”, and “Delete Selected Concepts” relevant to the chosen menu item at the top, i.e., “ConceptScheme”.

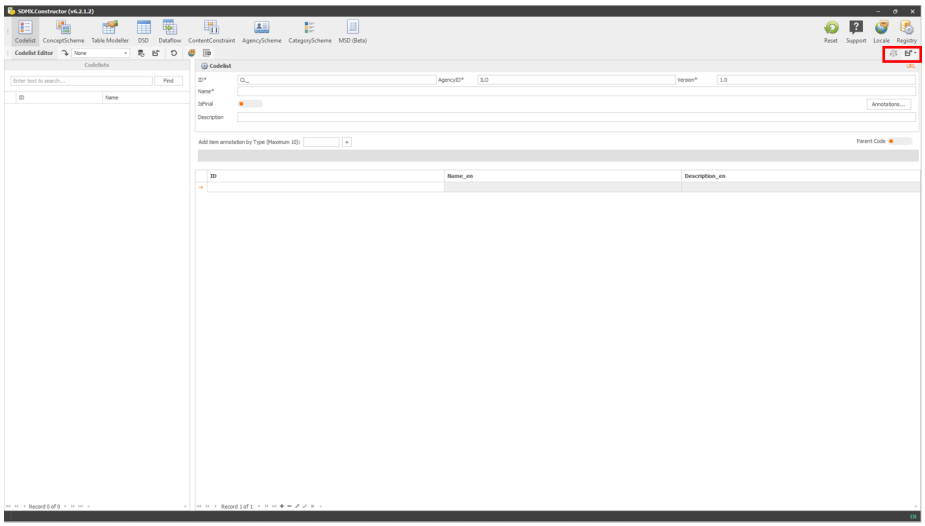
All options available within this menu group per the item selected on the top menu are listed below.



Click here to enlarge the image

3.4 Preview and Export Menu

The fourth menu item group (Preview and Export) is in the top right corner and below the first group of menu items (as highlighted below).



Click here to enlarge the image

The options available in this group of menu items are related to the preview and export functionalities of the tool and change according to the menu item selected on the top menu (the second group of menu items (Editors)).

Below are the icons representing the menu items, along with their features. Users can change the size of these icons by right-clicking on any of them, selecting Customize, and selecting the appropriate setting in Options.

Table 3.4: A bird's-eye view of the menu items in the Editor Ribbon

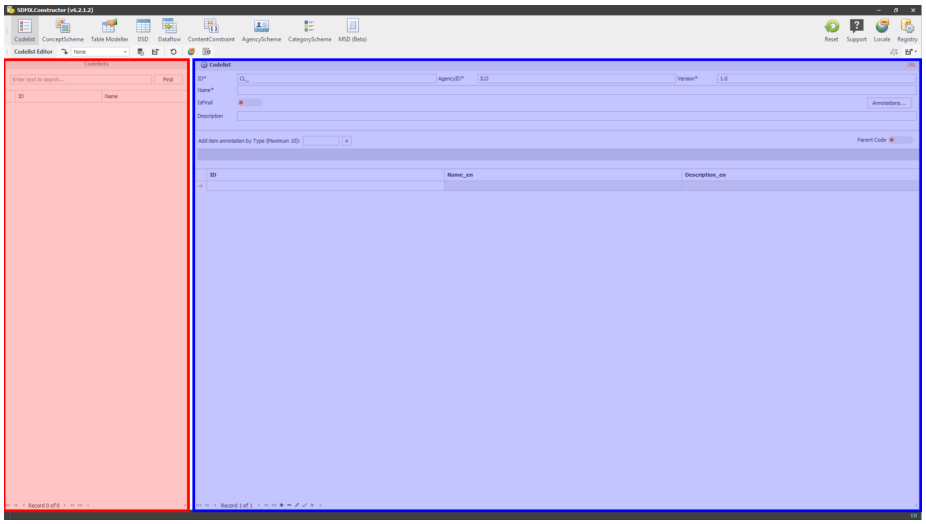
Menu item	Names and functions
	This is the Export button. This button saves the SDMX artefact in a local folder. In the case of saving the following artefacts: Concept Scheme, Table Modeller and Category Scheme, there are two options available: 'Save with descendants' and 'Save without descendants'. For saving a Code list, the following options are available: .Stat v7 Dim, CSV and SDMX-ML. For the following artefacts: DSD, Dataflow, ContentConstraint, Agency Scheme, and MSD, there are no variations in the option for saving.
 or 	The 'Push to DLM' (DLM is short for Data Lifecycle Manager of the .Stat Suite) button becomes active (appears with colours) when the SDMX Consctuctor is connected with a DLM instance of a .Stat Suite. Note that credentials are required to upload structures (push) to a registry. Two options are available for the following artefacts: ConceptScheme, Table Modeller and CategoryScheme: 'Push with descendants' and 'Push without descendants'. In the case of the following artefacts: Codelist, DSD, Dataflow, ContentConstraint, AgencyScheme, and MSD, there are no variations in the option for saving.
	Structure Viewer. It lets you preview the tabulation and download the MS Excel and CSV templates. It is available for DSD and Dataflow.

3.5 Working area

The following section explains the working area of the tool.

Below the menu items on top, the user interface appears divided into two main parts. First, the left pane on the interface serves as a space to hold the artefacts' list. The other side is designated to showcase relevant details (both sides are

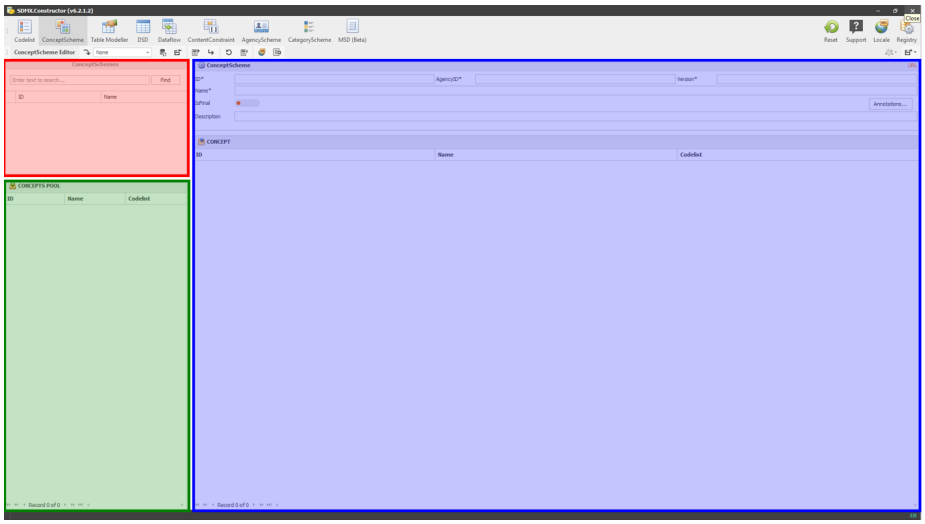
highlighted in different colours in the screenshot below). Selecting (by double-clicking) an artefact from the left pane prompts its details to appear on the screen's right-side pane. This layout is available for Codelist, Dataflow and ContentConstraint.



Click here to enlarge the image

For ConceptScheme, Table Modeller, DSD, AgencyScheme, CategoryScheme, and MSD, an additional pane also appears in the user interface (below the placeholder for listing the artefacts - (as highlighted below)), which acts as a staging area (or a pool). It becomes a CONCEPTS POOL for ConceptScheme, Table Modeller, DSD and MSD options. And it turns into an AGENCY POOL for AgencyScheme and a CATEGORY POOL for CategoryScheme options.

The users can drag the artefacts back and forth between the pool area and the pane on the right.



Click here to enlarge the image

3.6 Input and output methods

Users can input information into SDMX Constructor in several ways, depending on the specific scenarios. The following are the options.

- Typing and copy-pasting: Users can enter information by typing text or copying and pasting into fields in the forms using a keyboard.
- Clicking: Users can interact with buttons, links, menus, or other graphical elements by clicking with a mouse, touchpad, or touchscreen. For example, double-clicking an artefact from the left pane load its details on the screen's right-side pane.
- Dragging and dropping: Users can manipulate objects or select multiple items by dragging and dropping them with a mouse, touchpad, or touchscreen.
- Loading from online SDMX Registry: Users can access pre-prepared SDMX artefacts from many default online SDMX registries. The input method allows users to retrieve information from the SDMX registry.

SDMX Constructor presents information to users in a standardised and user-friendly way, using a combination of graphical elements, form-based input and validation and error messages.

- Graphical interface: The SDMX Constructor uses a graphical user interface (GUI) that allows users to interact with the metadata components using visual elements like buttons, menus, and icons. This makes it easy for

users to navigate and manipulate the metadata and get a quick overview of the SDMX artefacts.
- Form-based input: The SDMX Constructor uses form-based input screens to allow users to enter information about the metadata components. The forms are designed to be intuitive and easy to use, with clear labels, drop-down menus, and text boxes that guide users through the input process.
- Validation and error messages: The SDMX Constructor validates the metadata components to ensure that they conform to the SDMX standard and are consistent with other components. If there are errors or inconsistencies, the tool provides error messages that help users identify and correct the issues.

3.7 Translation

SDMX Constructor allows users to manage multilingual metadata components easily. Users can choose from automatic or manual translation methods and edit translations using translation tables. The tool simplifies the translation process with standardised language codes and automation and helps users achieve accurate, high-quality translations.

Here are some tool features supporting the translation of SDMX artefacts:

- Translation into three languages: The tool supports translating content into three languages of your choice. It helps by following two automatic methods, but you can always do it manually if required.
- Automatic translation methods: Using the Google Translate API key, you can connect with Google's translation services. Another automated way is using DeepL. The SDMX Constructor allows programmatic access to DeepL's machine translation technology, bringing high-quality translation capabilities directly to the tool.
- Translation tables: Using these API keys, the SDMX Constructor provides translation tables, which allow users to edit suggested translations of metadata components in different languages. You can enter translations for different language codes, and the tool will automatically populate the translations in the appropriate fields.
- Standardised language codes: The SDMX Constructor uses standardised language codes; users can select from a list of pre-defined language codes.

Chapter 4

Using SDMX Constructor

Welcome to the chapter on using SDMX Constructor! This chapter will cover three key topics that will help you make the most of this powerful tool.

The first topic we will cover is how to use SDMX Constructor to access SDMX artefacts from SDMX registries. This will include step-by-step instructions on how to use the tool to connect to a registry, browse its contents, and download the artefacts you need. Sections 4.1 and 4.2 will cover this topic.

Next, we will dive into how to use SDMX Constructor to create new SDMX artefacts from scratch. Whether you need to create a new ConceptScheme, code list or DSD, SDMX Constructor provides a simple and intuitive interface to help you get the job done. Sections 4.3 to 4.7 will cover this topic.

Finally, we will explore how to use SDMX Constructor to work with .Stat Suite. This powerful platform is designed to help you analyse, visualise, and disseminate statistical data, and SDMX Constructor is the perfect complement to help you access and work with the SDMX artefacts you need. Sections 4.8 and 4.9 will cover this topic.

By the end of this chapter, you will have a solid understanding of how to use SDMX Constructor to access, create, and work with SDMX artefacts, and you will be well on your way to becoming an expert in this powerful tool. So, let's get started!

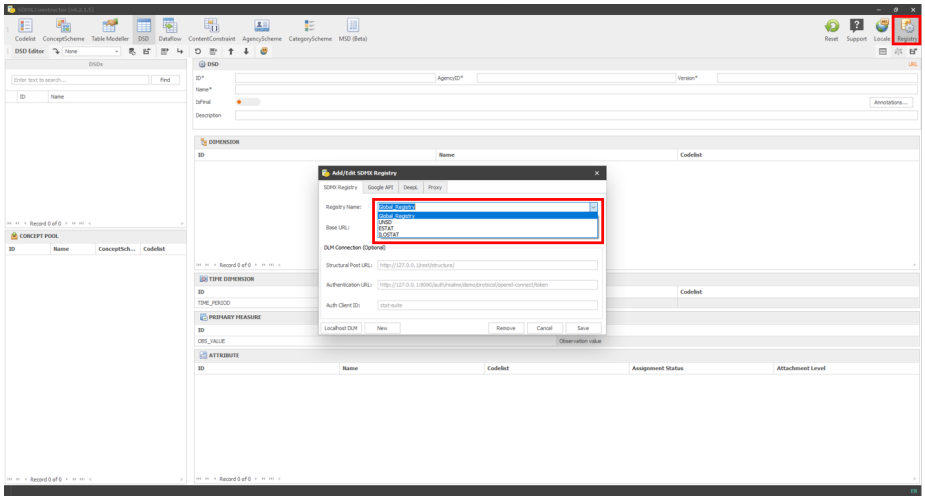
4.1 Accessing SDMX artefacts from registries

In this section, we will walk you through the process of using SDMX Constructor on your computer to access and view the SDMX artefacts from the SDMX registries. This will enable you to easily browse and download the artefacts you

need, for example, from the default registries already available in the SDMX Constructor.

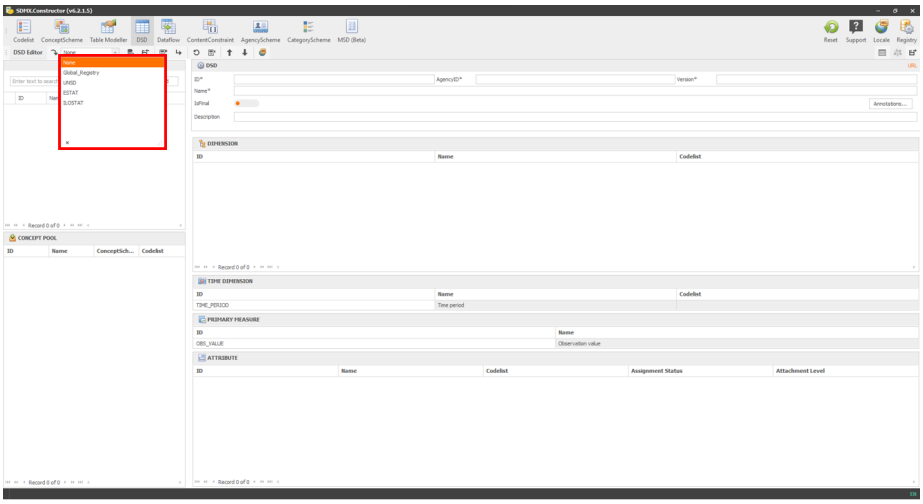
Default SDMX registries

You can use the SDMX Constructor on your computer to access and view the SDMX artefacts from the SDMX registries. By default, SDMX Constructor offers the following registries to access SDMX artefacts: SDMX Global Registry: (https://registry.sdmx.org/), United Nations Statistics Division (UNSD): (https://data.un.org/WS), the Italian National Institute of Statistics (ESTAT) and the ILO Department of Statistics (ILOSTAT): (https://www.ilo.org/sdmx/index.html). You can view these by going to the Registry button and opening the Registry Name dropdown in the SDMX Registry tab, as shown below.



Click here to enlarge the image

As shown below, select a registry from the dropdown option to load the artefacts.

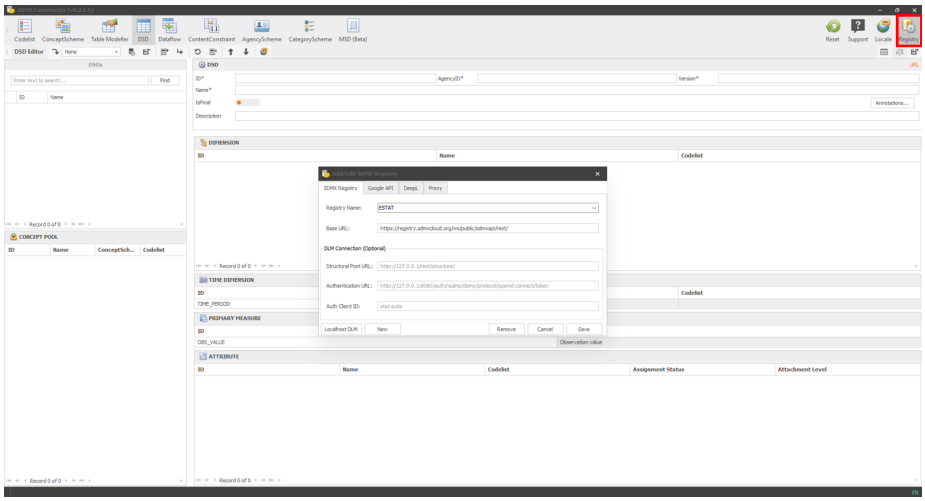


Click here to enlarge the image

How to add a new registry to the list

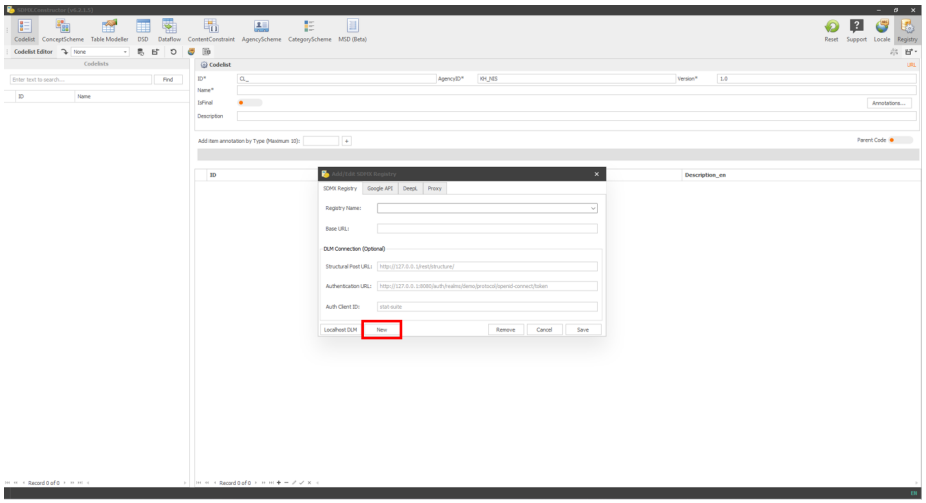
You can also use the SDMX Constructor to access and view the SDMX artefacts from additional SDMX registries. You can do that by following these steps.

Click on the Registry button as shown below. It will open up a pop-up window and show the SDMX Registry tab.



Click here to enlarge the image

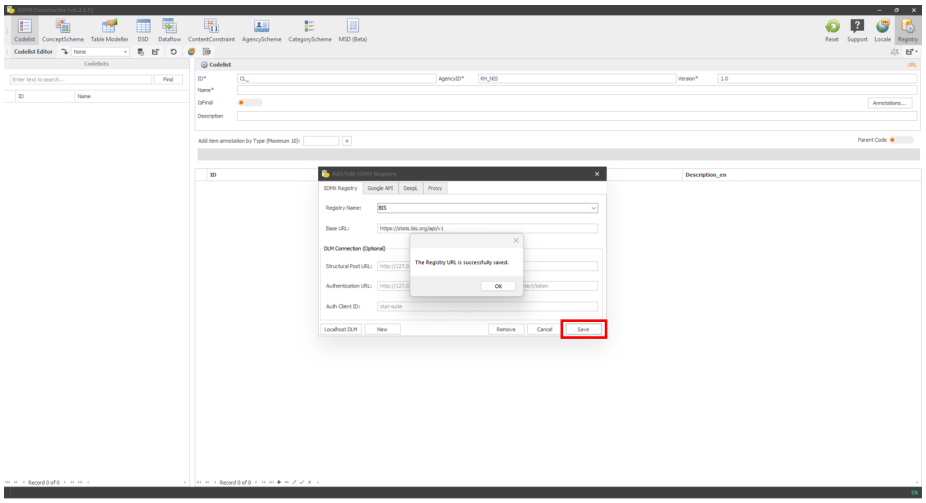
Clicking the New button will clear the two fields: Registry Name and Base URL.



Click here to enlarge the image

To illustrate further with an example, let's take the case of the Bank for International Settlements (BIS) SDMX registry here: https://stats.bis.org/api-doc/v1

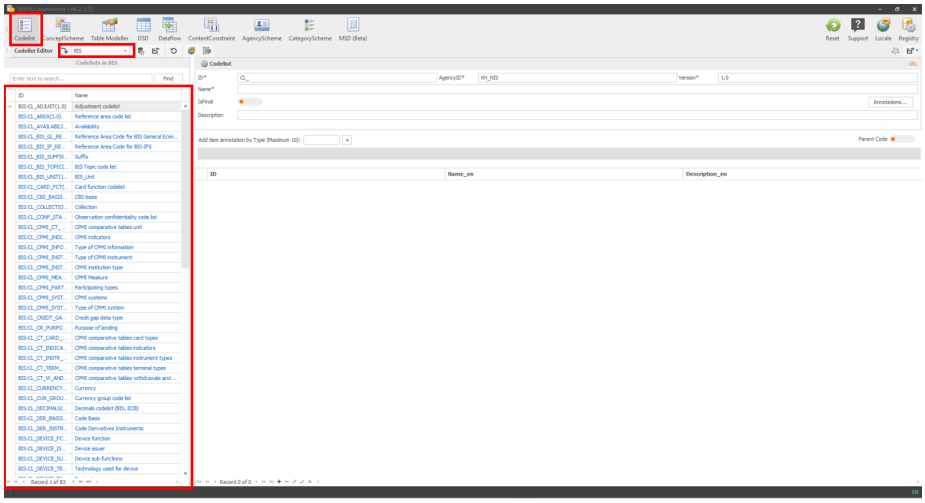
In the fields 'Registry Name' and 'Base URL', fill in the required details. For the Registry Name, let's write BIS, and for the Base URL, enter https://stats.bis.org/api-doc/v1. Pressing the Save button will result in a confirmation message indicating that the Registry URL is successfully saved, as shown below.



Click here to enlarge the image

Click on OK to close the message.

Now, we can see the SDMX artefacts from the BIS SDMX registry. For instance, you can now click Codelist and select BIS from the 'Load from Registry' option. This action will load the code lists from the BIS SDMX registry, as shown below.



Click here to enlarge the image

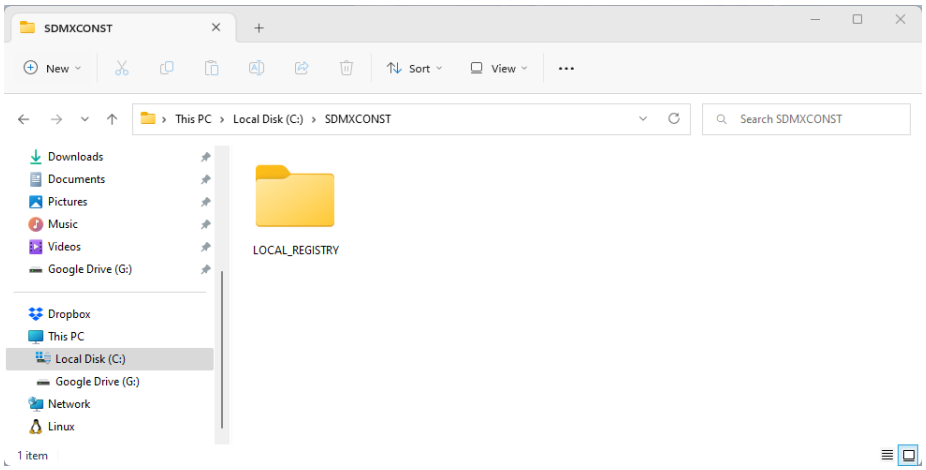
By accessing the default registries or adding new ones, you can easily access SDMX artefacts from registries, making data retrieval and analysis more efficient and effective.

4.2 Setting up a registry as a local folder

You can use the SDMX Constructor to create SDMX artefacts on your computer without a complicated setup. You can start using it with a local folder on your computer. Following are the steps to set up a registry in a computer's local folder.

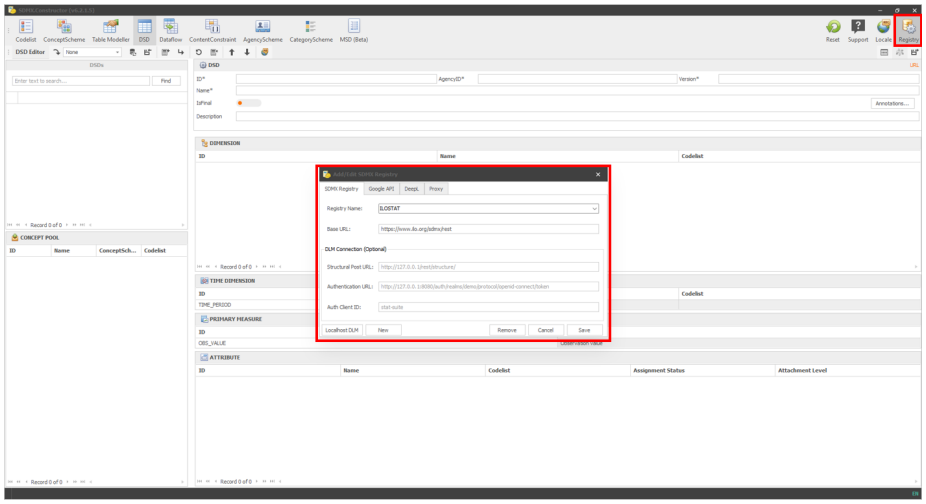
- On your computer, create a folder, and let's call it LOCAL_REGISTRY [1]. The screenshot below shows that the folder is created within the C drive.

[1] You can name the folder as you prefer; however, there are some good practices which you can follow. For example, consider using a naming convention like "My_Registry" or "XXX_Reg". Use "LOCAL_XXX" as a consistent naming convention for all local registries if you expect multiple registries, with XXX as a placeholder for differentiation. Also, the folder's name can be different from the registry's name as long as it's clear which set of artefacts it contains. Keep in mind that you can create multiple local registries in different folders.



Click here to enlarge the image

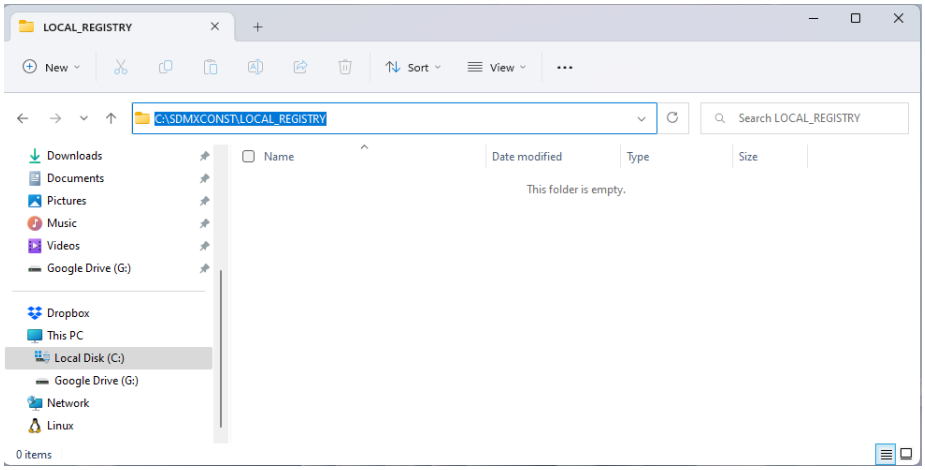
- Start the SDMX Constructor.
- Click on the Registry button on the SDMX Constructor. It will open a pop-up window showing the default entries in the SDMX Registry tab.



Click here to enlarge the image

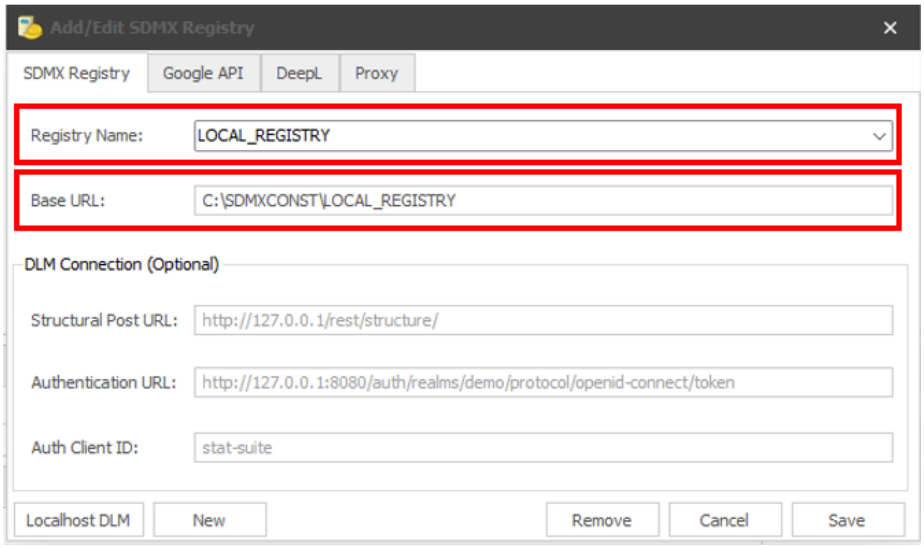
- In the pop-up window, there are only two fields where we need to make changes: Registry Name and Base URL. You can click the 'New' button (which will clear the fields) or type directly within the fields.

- For Registry Name, please type the name of the folder we created before LOCAL_REGISTRY.
- For the Base URL, get the path of the folder (shown below).



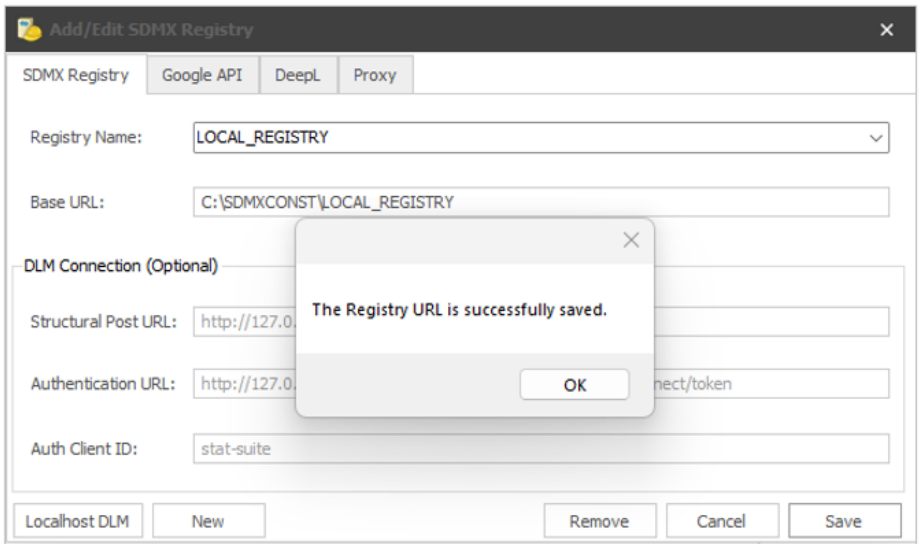
Click here to enlarge the image

- After entries in two fields, the pop-up window will look like this:



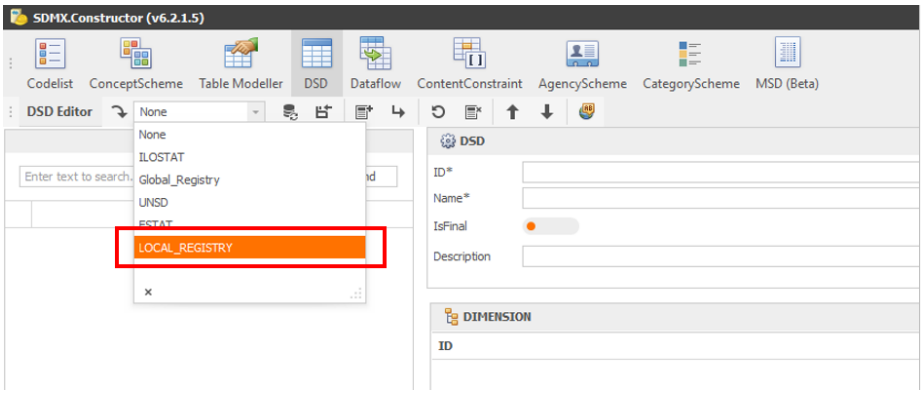
Click here to enlarge the image

- Hit Save. It will generate a confirmation message, as shown below.



Click here to enlarge the image

- Press OK to confirm. The pop-up windows will go away.
- To confirm if the folder is accessible from the tool, in the Editor Ribbon area, if you go to the 'Load from registry' option, you will see the LOCAL_REGISTRY in the dropdown options.



Click here to enlarge the image

- The setup with a local folder is complete.

4.3 Preparing inputs

To demonstrate the key functionalities of the SDMX Constructor, it is helpful to have a dummy dataset modelled appropriately.

Imagine a country called Demoland and its National Statistical Office (NSO), Demoland NSO. Imagine the Demoland NSO is creating SDMX artefacts for its data in the following two tables.

Table 4.1: Unemployment Rate by sex and region

Region	2015	2015	2015	2016	2016	2016
	Men	**Women**	**Both Sexes**	**Men**	**Women**	**Both Sexes**
Cities and Towns	14.4	20.7	17.6	14.4	21	17.2
Urban Villages	27.8	30.3	29.1	28.2	29.4	29.1
Rural Areas	26.2	29.6	28	26.2	29.5	27.1
Total	24	27.9	26	24	27.9	26

Table 4.2: Population outside the labour force by sex and age group (2022)

Age Group	Men	Women	Both Sexes
15-24	130,088	133,770	263,856
25-34	33,165	60,240	93,405
35-44	21,970	38,944	60,914
45-54	18,631	36,403	55,034
55-64	26,821	44,010	70,831
65+	45,323	78,725	124,048
Total	275,997	392,092	668,088

After modelling the data in both tables, we have the following ConceptScheme and Codelist arranged below. Note that the order of the columns in the following tables is per the SDMX Constructor's bulk load default templates (ConceptScheme and Codelist).

ConceptScheme:

Table 4.3: ConceptScheme

Concept ID	Concept Name	Description	Codelist ID
INDICATOR	Indicator	Refers to statistical measure describing a particular aspect of a social, economic or environmental phenomenon	CL_INDICATOR
UNIT_MEASURE	Unit of measure	Unit in which the observation values are expressed	CL_UNIT_MEASURE
SEX	Sex	State of being male or female	CL_SEX
GEO	Geographic area	Refers to Urban or Rural locations	CL_GEO
TIME_PERIOD	Time period	Timespan or point in time to which the observation refers	CL_TIME_PERIOD
FREQ	Frequency	Time interval at which observations occur over a given time period	CL_FREQ
AGE	Age groups	Length of time that an entity has lived or existed	CL_AGE
OBS_VALUE	Observation values	Refers to data or observation values	CL_OBS_VALUE

Codelist:

Table 4.4: Codelist

Codelist ID	Concept Name	Code ID	Code Name
CL_INDICATOR	Indicator	UNER	Unemployment Rate
CL_INDICATOR	Indicator	POLF	Population outside the labour force
CL_UNIT_MEASURE	Unit of measure	RT	Rate
CL_UNIT_MEASURE	Unit of measure	PS	Persons
CL_SEX	Sex	M	Men
CL_SEX	Sex	F	Women
CL_SEX	Sex	_T	Both Sexes (Total)
CL_GEO	Geographical area	M	Cities and Towns (Metropolitan Area)
CL_GEO	Geographical area	U	Urban Villages
CL_GEO	Geographical area	R	Rural Areas
CL_GEO	Geographical area	_T	Total

Codelist ID	Concept Name	Code ID	Code Name
CL_FREQ	Frequency	A	Annual
CL_AGE	Age groups	15T24	15-24
CL_AGE	Age groups	25T34	25-34
CL_AGE	Age groups	35T44	35-44
CL_AGE	Age groups	45T54	45-54
CL_AGE	Age groups	55T64	55-64
CL_AGE	Age groups	GE65	65+
CL_AGE	Age groups	_T	Total

We will use these dummy datasets in this user manual to illustrate some critical functionalities of SDMX Constructor.

4.4 Creating AgencyScheme

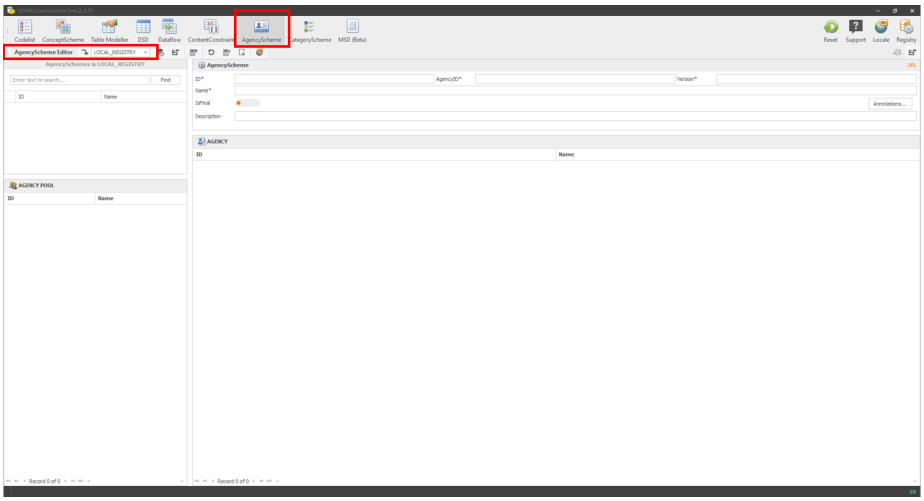
In the context of SDMX, an AgencyScheme is a type of structural metadata that provides information about the organisations or agencies responsible for producing or disseminating data. It defines the different roles these organisations play and assigns unique identifiers to them, allowing them to be easily referenced and identified within the SDMX framework.

The AgencyScheme is an important component of the SDMX infrastructure, as it enables data users to understand better the context of the data they are working with. It allows them to identify who produced the data, what their role was in the data production process, and where to go for more information.

For example, an AgencyScheme may include information on national statistical organisations, international organisations, or other data-producing entities. Each organisation or agency would be assigned a unique identifier and given a defined role within the AgencyScheme. Using the AgencyScheme allows data users to easily navigate the complex world of data production and dissemination, ensuring that they are working with accurate and reliable information.

To create an AgencyScheme, you can use the SDMX Constructor by following these steps:

- Start the SDMX Constructor, click the AgencyScheme button on top, and select the folder we created before, LOCAL_REGISTRY, from the AgencyScheme Editor's 'Load from registry' dropdown option, as shown below.

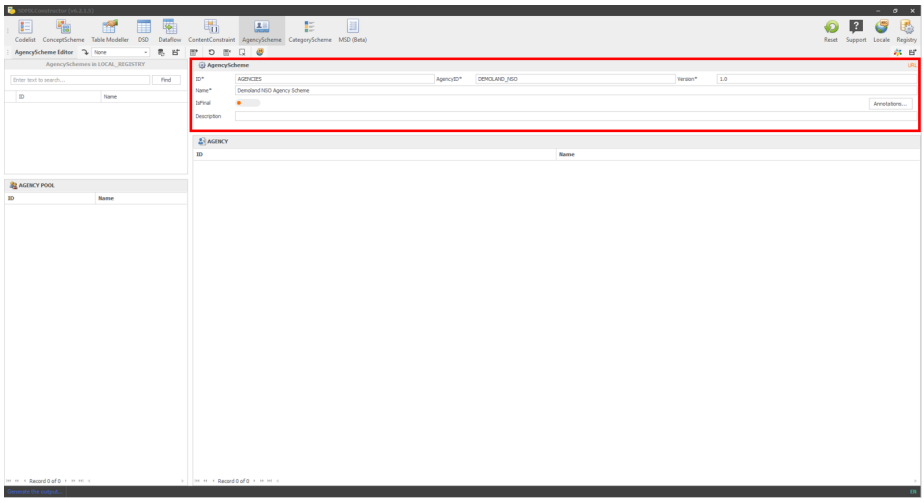


Click here to enlarge the image

- Create an AgencyScheme by entering the ID, AgencyID, Version and Name in the fields as shown below. For this exercise, we will have only Demoland NSO as an agency (all data are from Demoland NSO). We will do it in two stages. First, we will enter the properties for the AgencyScheme and then create the agency.

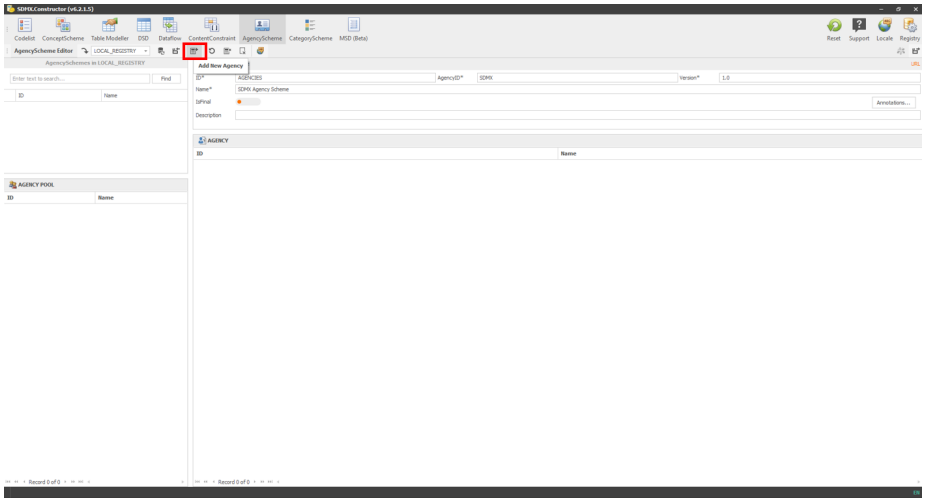
  Note on IDs and versions: In the context of SDMX, it is important to note that the identifiers for all SDMX artefacts follow a specific structure. These identifiers consist of a triplet composed of the "Agency ID," the "ID," and the "Version." It is worth emphasising that this naming convention applies not only to the AgencyScheme but also to other SDMX artefacts. However, unlike other SDMX artefacts that may undergo revisions and updates, the AgencyScheme is a foundational component within the SDMX framework. It defines the agencies or organisations involved in statistical data and metadata exchange, along with their unique identifiers (Agency ID). While the AgencyScheme identifier still follows the triplet structure of "Agency ID, ID, and Version," the version component remains fixed at 1.0. Also, note that a global guideline, 'Guideline for CL_ORGANIZATION', describes a method to build and maintain code lists for "organisations" broadly. https://sdmx.org/wp-content/uploads/CL_ORGANISATION-1.0_April_2021.docx.

- For ID, enter AGENCIES; For AgencyID, enter Demoland_NSO; for version, enter 1.0; and for the Name, enter Demoland NSO Agency Scheme as shown below.



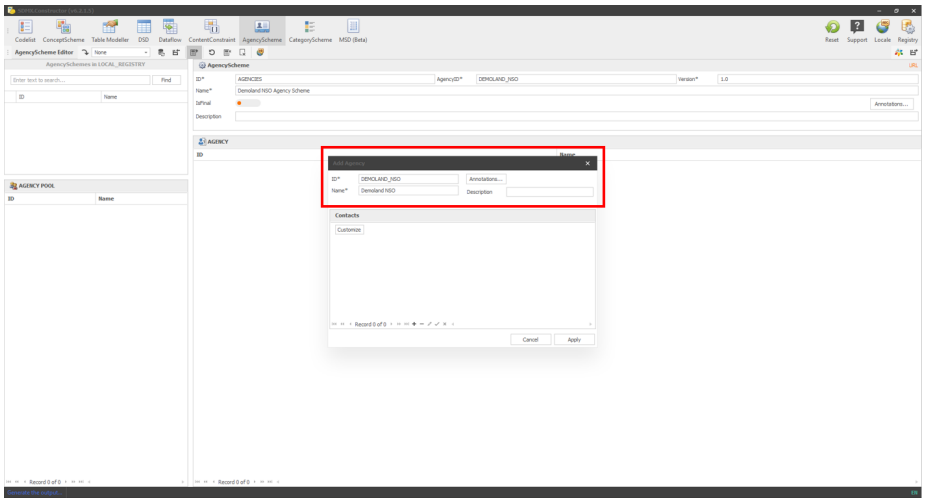
Click here to enlarge the image

- Now, we create the agency (as part of the AgencyScheme) by clicking on 'Add New Agency' as shown below.



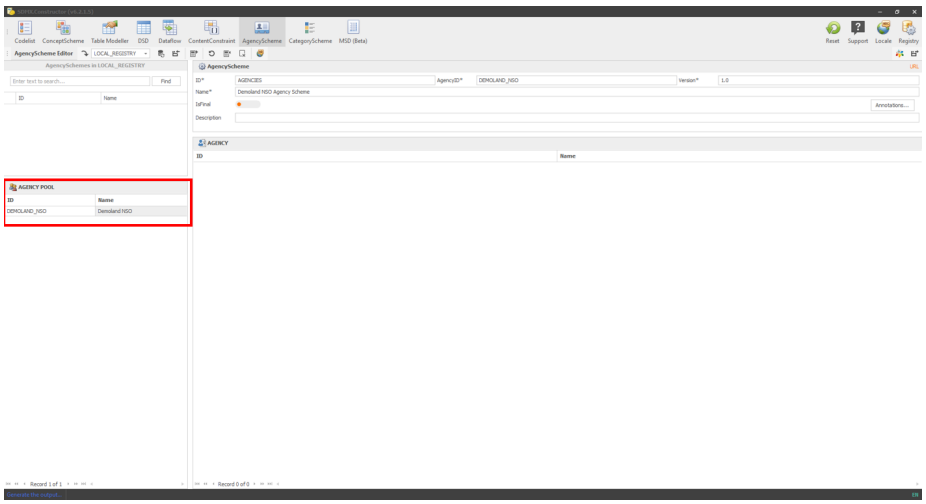
Click here to enlarge the image

- Clicking on 'Add New Agency' will open a pop-up window. We create the Demoland NSO agency by entering DEMOLAND_NSO in the ID and Demoland NSO in the Name field in the Add Agency pop-up and clicking Apply (as shown below).



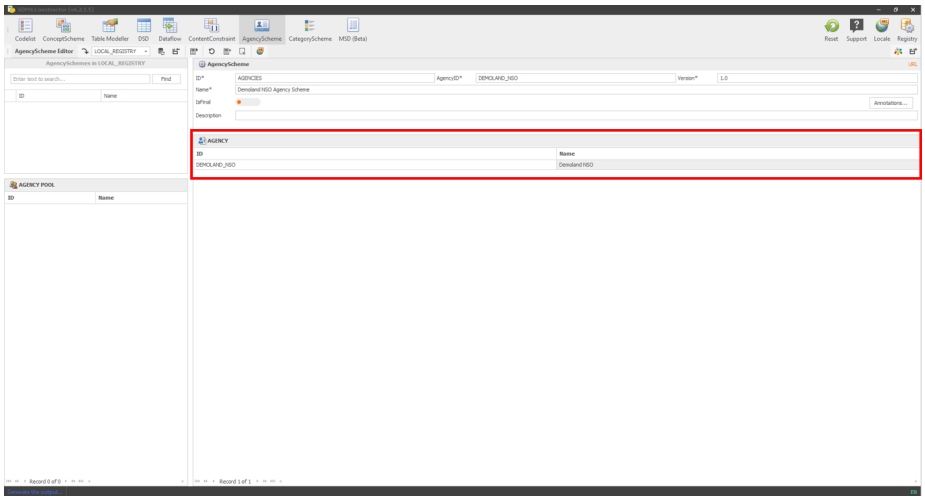
Click here to enlarge the image

- Once we finished creating the agency, it would look like the following (the agency will be in the AGENCY POOL).



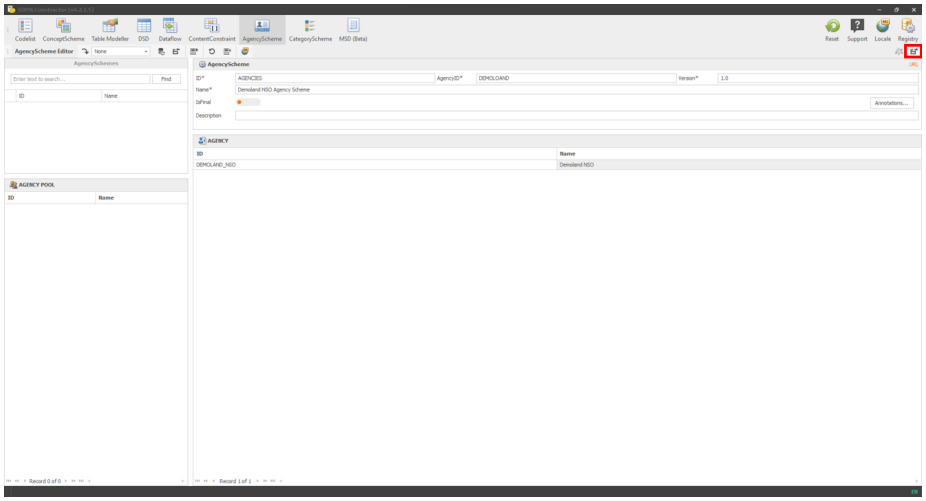
Click here to enlarge the image

- Move the agency from the AGENCY POOL to the AGENCY space on the right pane by selecting and dragging it. After the move, it would look like the following.



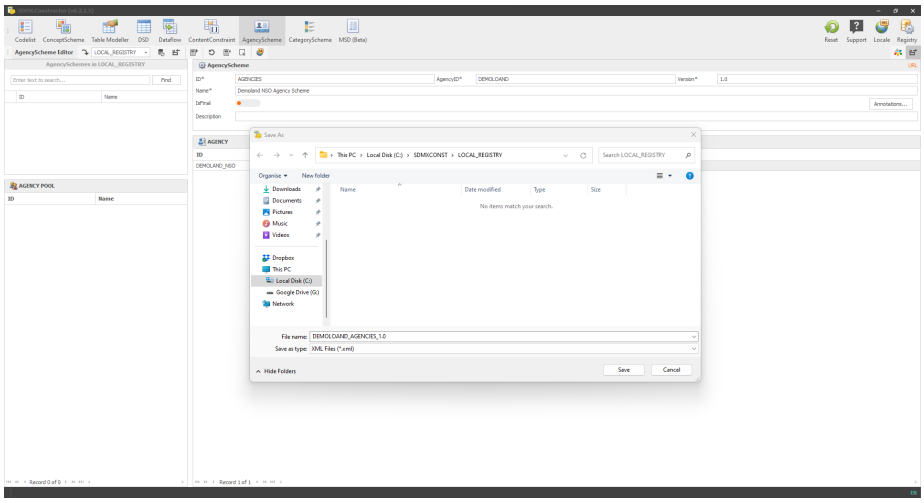
Click here to enlarge the image

- Click the Export button highlighted below to save the agency scheme in the folder.



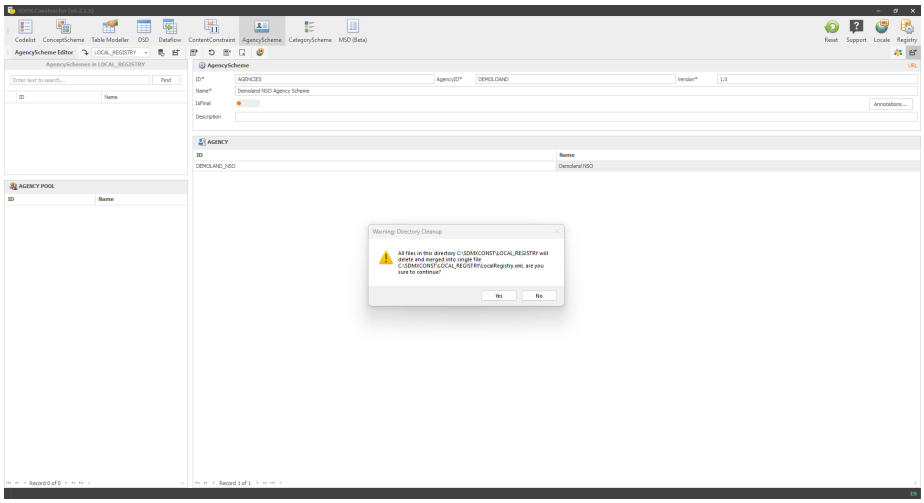
Click here to enlarge the image

- As shown below, a pop-up window will open to confirm the saving location.



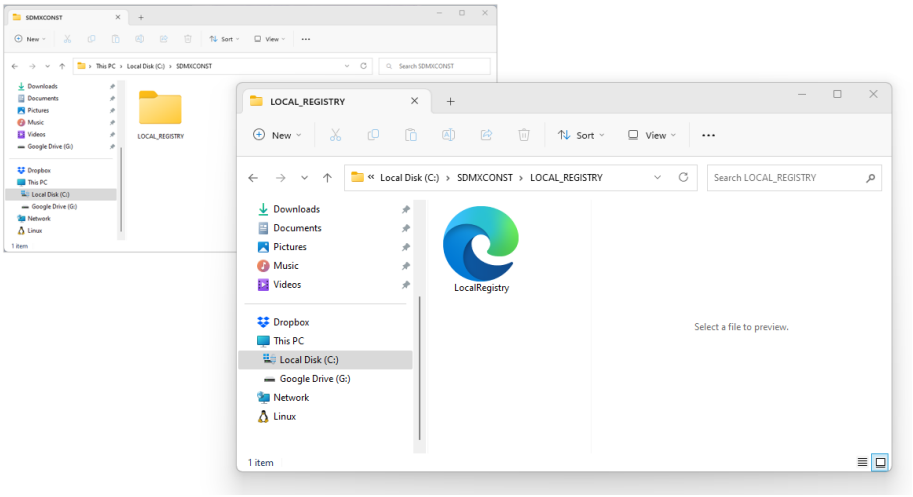
Click here to enlarge the image

- Clicking on Save will prompt another message (“All files in this directory (path) with delete and merged into a single file (path) are you sure to continue?”), as shown below.



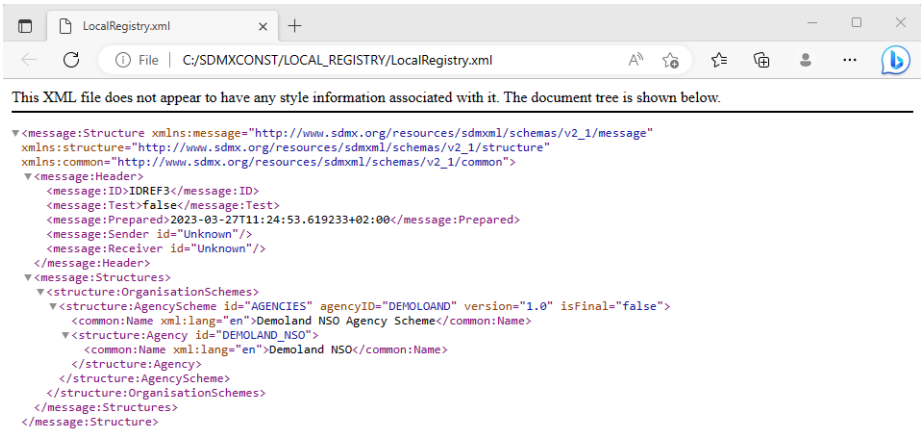
Click here to enlarge the image

- Clicking on Yes will create and save an XML file in the folder we created before, as shown below.



Click here to enlarge the image

- Opening the file (LocalRegistry.xml) will show the agency scheme, as shown in the image below.



Click here to enlarge the image

4.5 Creating ConceptScheme & Codelist

In the context of SDMX, a ConceptScheme refers to a structured representation of concepts used to classify and describe statistical data and metadata. It serves

as a framework or taxonomy for organising and categorising these concepts within the SDMX framework.

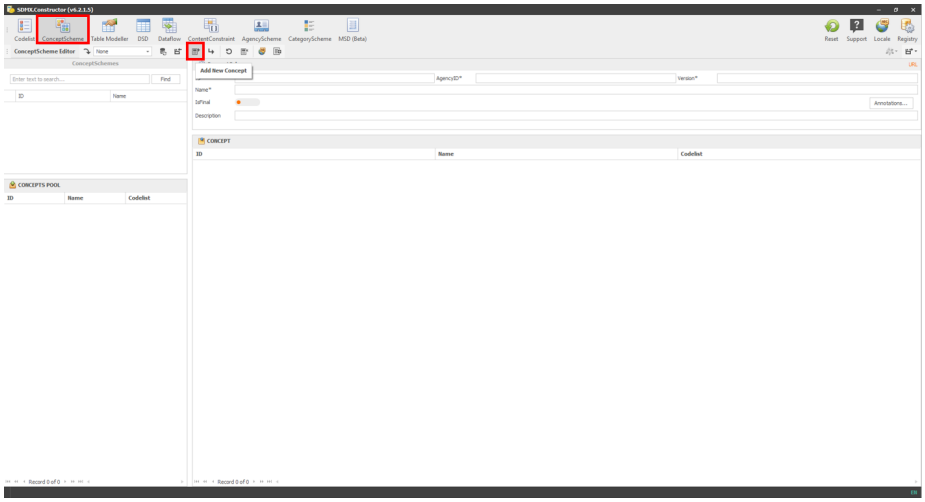
A ConceptScheme provides a standardised and harmonised way of defining relevant concepts to a particular domain or statistical subject area. It helps ensure consistency in interpreting and using these concepts across different datasets and statistical systems.

When Concepts represent a statistical categorical variable (or qualitative variable), they can take on one of a limited and fixed number of possible values. In such a case, the concept is "enumerated", and its representation is given by a Codelist that contains the valid values that the concept can take on and its meaning. For example, in a dataset on population demographics, the code list for the concept of sex might include codes such as Men, Women, and Both Sexes (Total).

Create Codelist: The "one by one" approach.

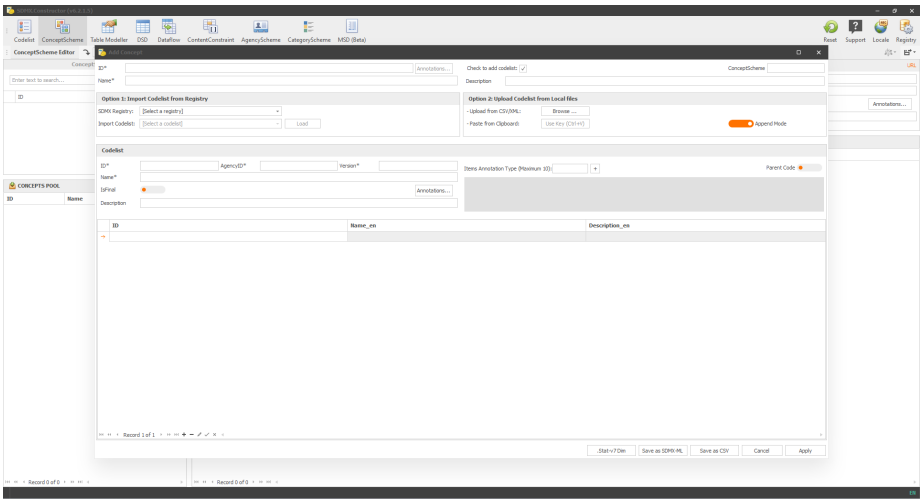
We can create a Concept Scheme in SDMX Consctuctor by following the steps described below. This section describes the one-by-one entry approach that is also possible, though the subsequent section illustrates the bulk load feature in detail.

- Click on the ConceptScheme button in the Editor menu. Then click the 'Add New Concept' button in the Editor Ribbon menu, as shown below.



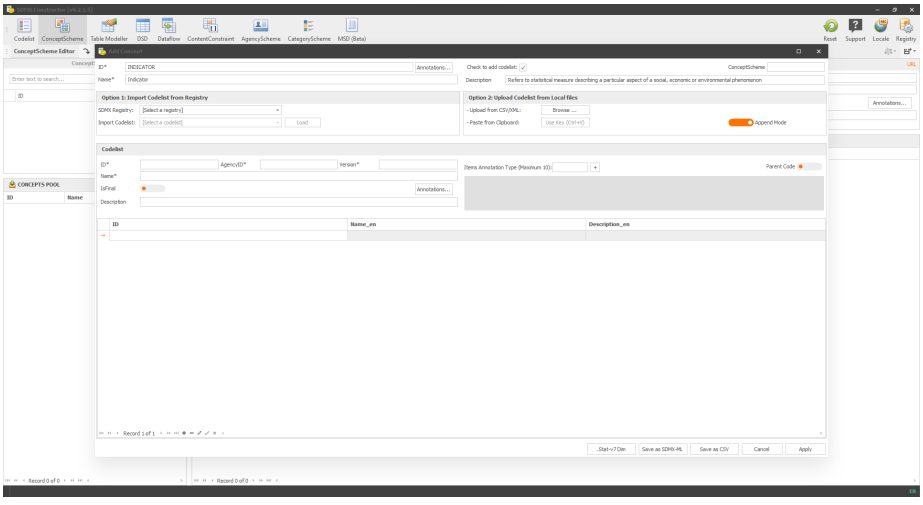
Click here to enlarge the image

- The resulting pop-up window would look like the following.



Click here to enlarge the image

- Provide the details as per the template. As an example, we choose the concept of 'indicator'. Hence we could use ID = INDICATOR and Name = Indicator. For Description, we could use 'Refers to statistical measure describing a particular aspect of a social, economic or environmental phenomenon'. We could keep the checkmark to add the code list (this would allow us to add the code list in the same interface), as shown below.

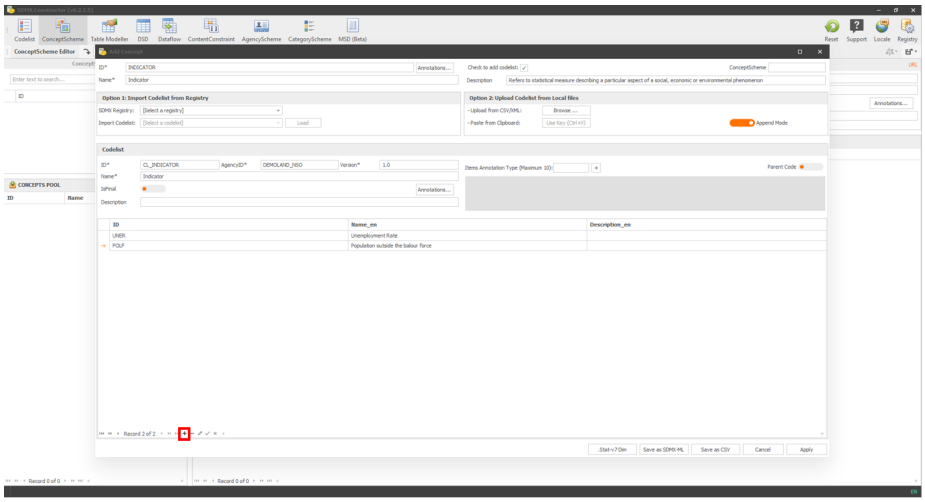


Click here to enlarge the image

- Now, we have three options to add the code list. Option 1 is to 'Import Codelist from Registry', Option 2 is to 'Upload Codelist from Local files', and below these options, as you can see in the image below, is the option to create a code list from scratch.

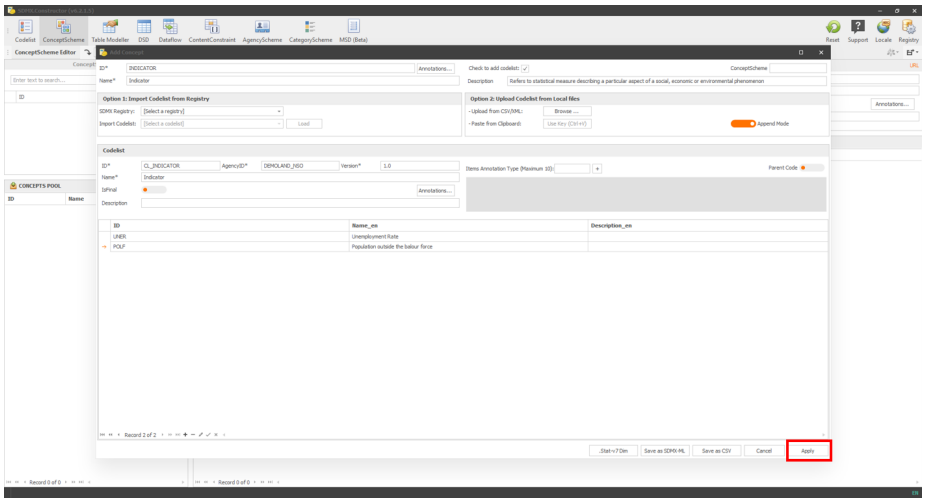
- We create the code list from scratch in the next step. We could use ID = CL_INDICATOR, Agency ID = DEMOLAND_NSO, Version = 1.0 and Name = Indicator

- We add two items to the code list for illustration purposes. ID = UNER, Name: Unemployment Rate and ID = POLF, Name = Population outside the labour force in two rows. Use the Append or + button to add multiple rows, as highlighted below.



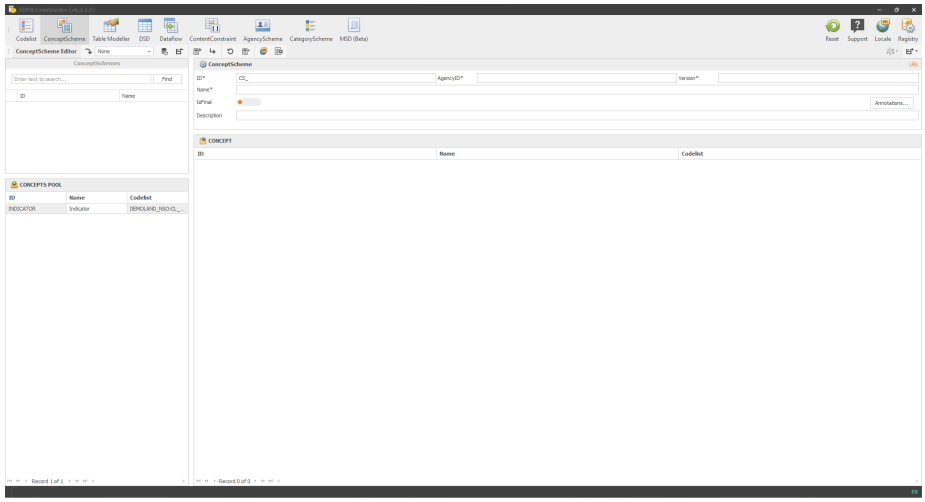
Click here to enlarge the image

- Click the Apply button (as highlighted below).



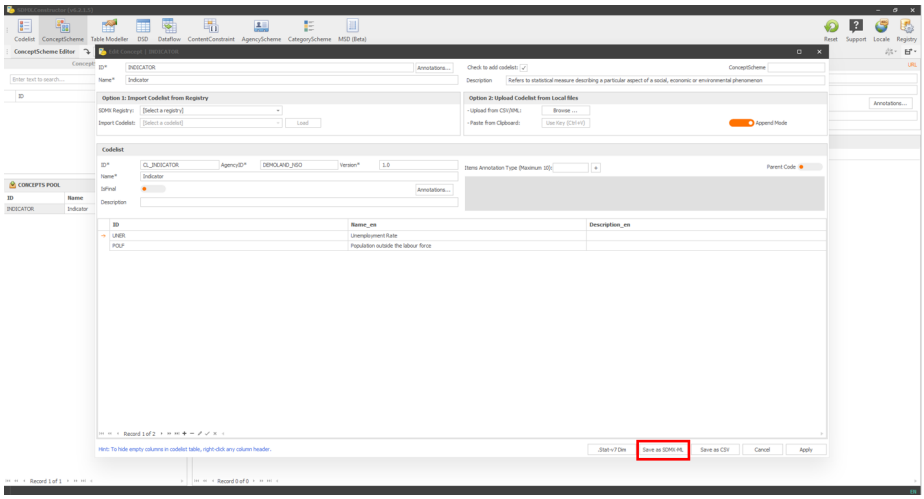
Click here to enlarge the image

- After clicking the Apply button, we will see the concept added in the CONCEPT POOL pane, as shown below.



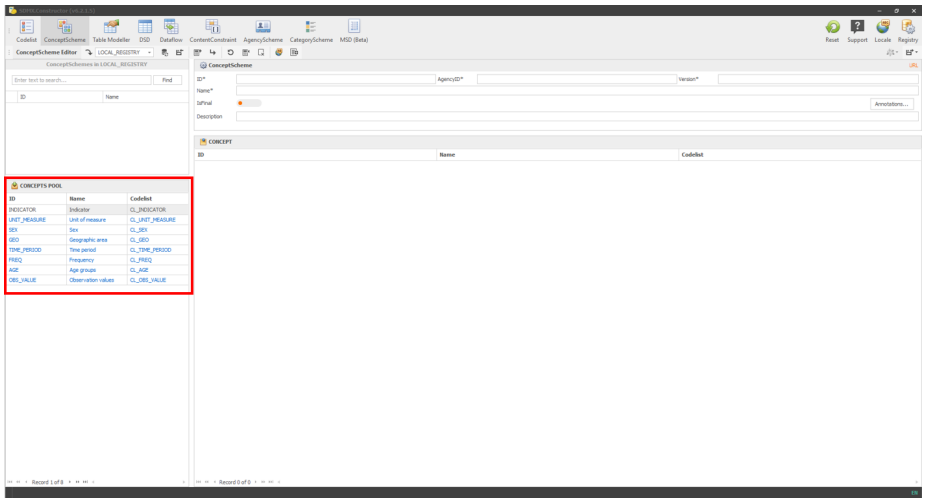
Click here to enlarge the image

- Double-clicking on the concept scheme will open up the pop-up window, as shown below. This time, clicking 'Save as SDMX - ML' (other options are also available), as shown below, will start the standard process of saving the artefact in a local folder for later use.



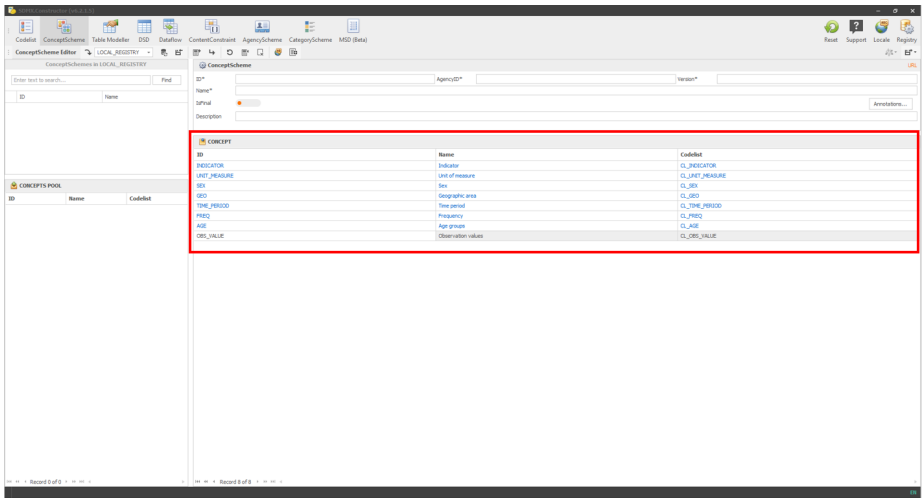
Click here to enlarge the image

- After creating all the concepts and related code lists, you can see all the concepts in the CONCEPT POOL, as shown below.



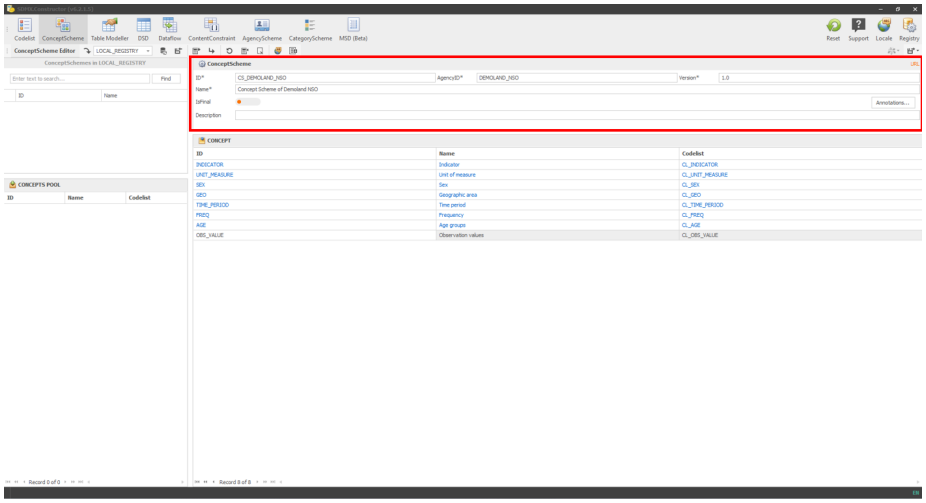
Click here to enlarge the image

- Move all the concepts from the CONCEPT POOL to the CONCEPT pane by selecting all (ctrl + a), then dragging and dropping. After the move, it would look like the following.



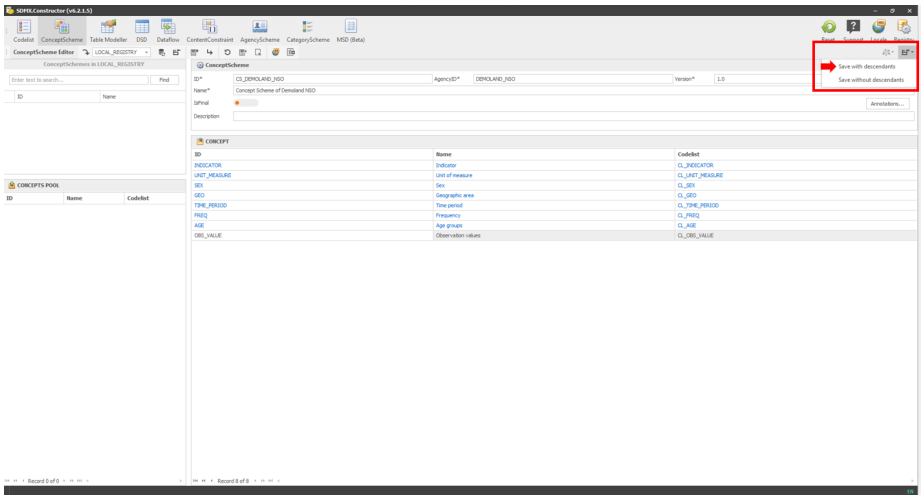
Click here to enlarge the image

- After moving the concepts, enter the details: (ID: CS_DEMOLAND_NSO, AgencyID: DEMOLAND_NSO, Version: 1.0, and Name: Concept Scheme of Demoland NSO) for the ConceptScheme, as shown below, to save the ConceptScheme.



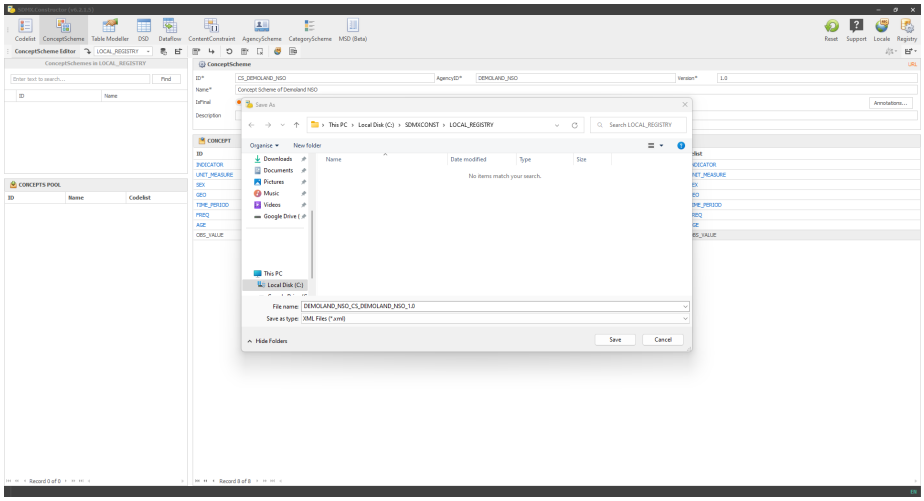
Click here to enlarge the image

- Then, click the 'Save with descendants' from the save option as shown below. This option, 'Save with descendants,' will save the concept scheme with the code list.



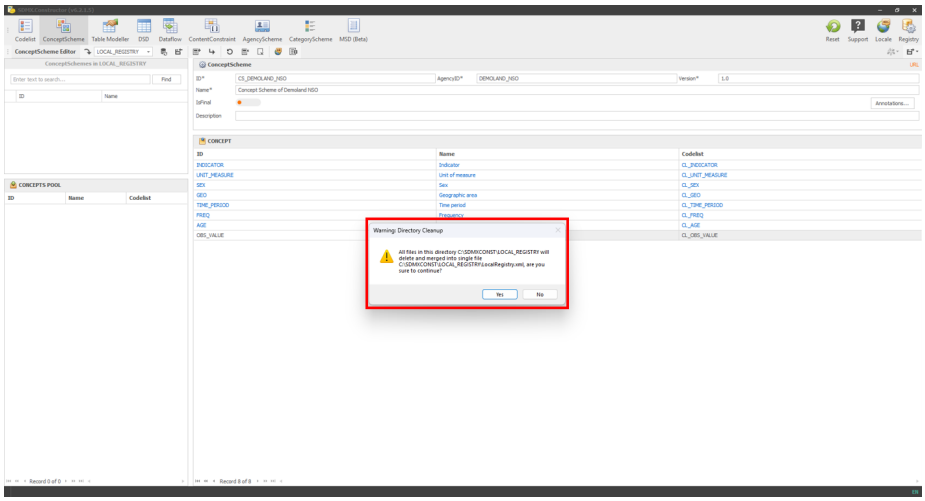
Click here to enlarge the image

- A pop-up message will ask to save the XML file in the folder (LOCAL_REGISTRY) we created before. Click on Save to save the file.



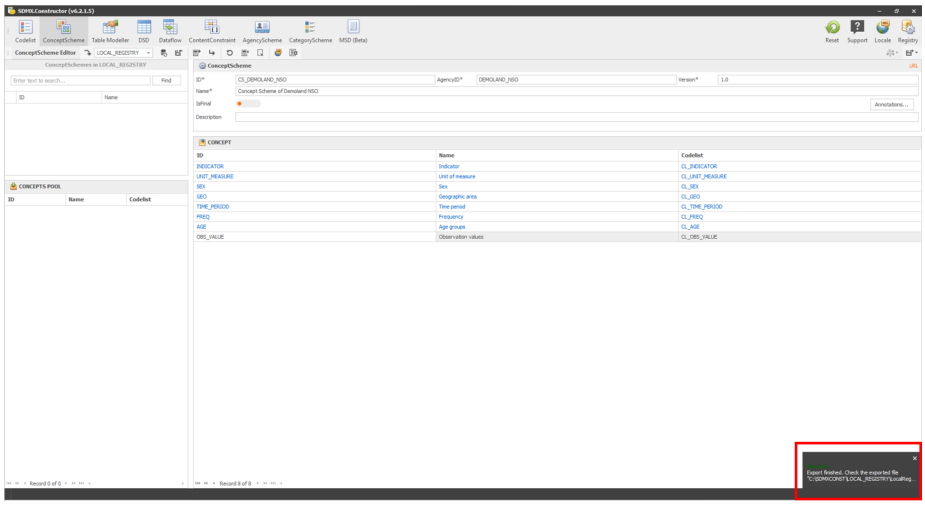
Click here to enlarge the image

- After clicking Save, the tool will ask the question to merge files. Select Yes.



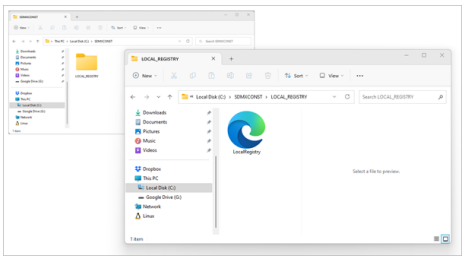
Click here to enlarge the image

- A confirmation message will appear briefly at the bottom right corner, as shown below.



Click here to enlarge the image

- After that, if you go to the file's location, you will see the XML file created, as shown below.

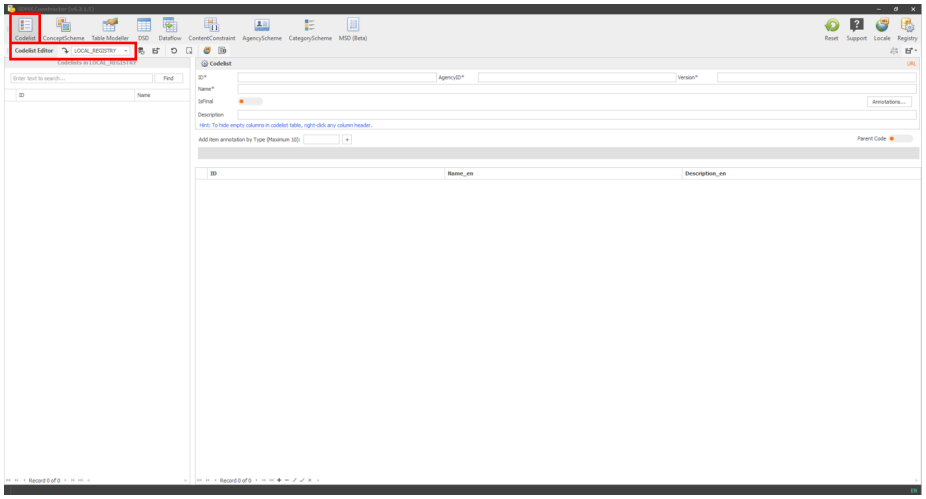


Click here to enlarge the image

Create Codelist: Bulk load.

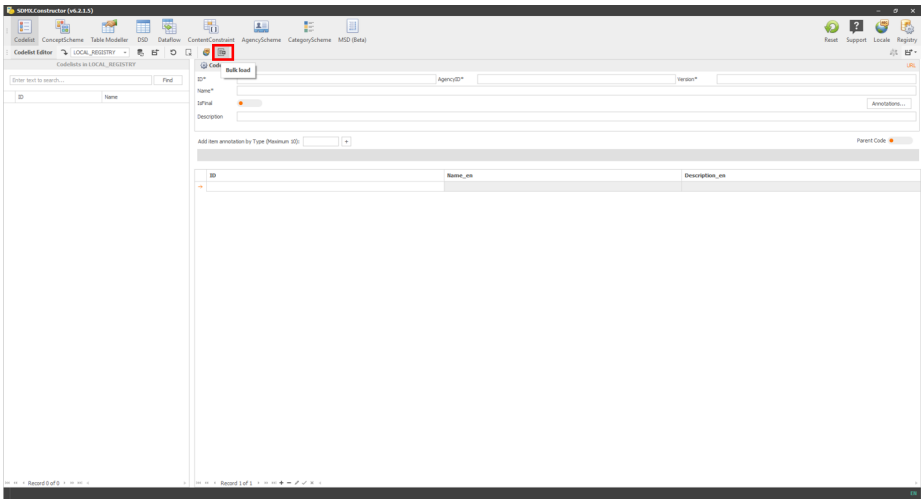
To create multiple Codelists and ConceptSchemes at once, you can use the SDMX Constructor by following these steps. First, upload the Code List and then the ConceptScheme. By completing these steps in sequence, you can create both artefacts in one go.

- Click on the Codelist button on top and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the Codelist Editor's 'Load from registry' dropdown option, as shown below.



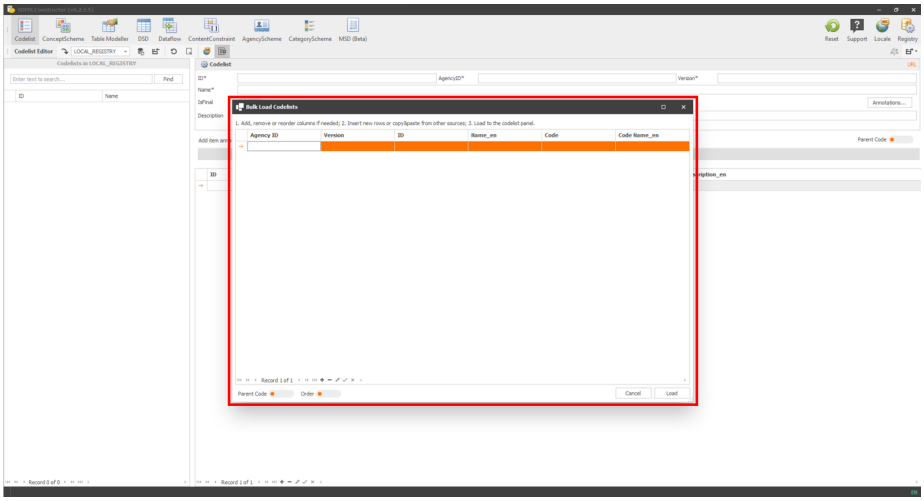
Click here to enlarge the image

- Click on the Bulk load button as shown below.



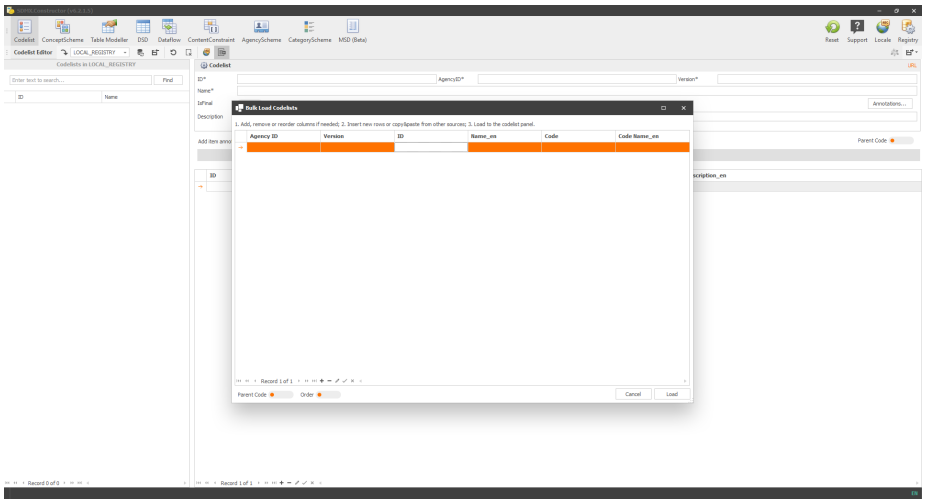
Click here to enlarge the image

- It will open up a pop-up window, as shown below.



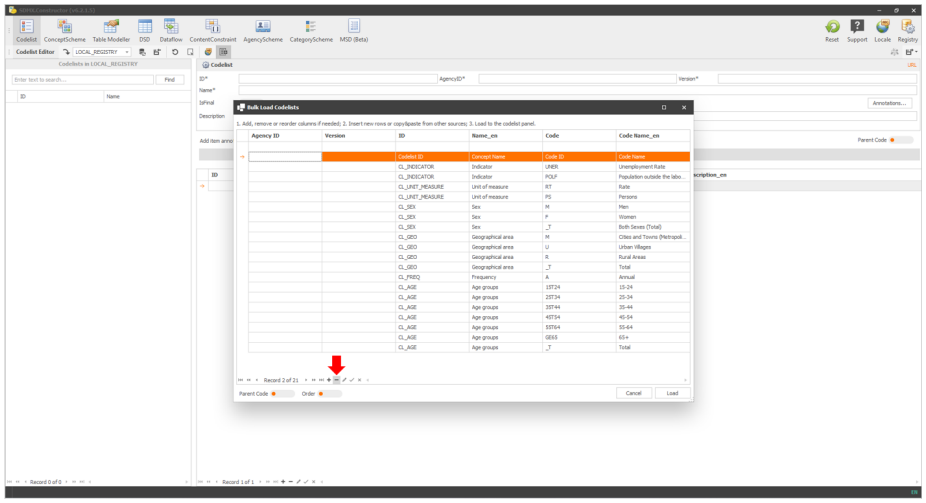
Click here to enlarge the image

- Copy the Codelist table (Table: 4.4) we prepared before and paste its contents here. Before pasting, remember to click on the ID column (shown in white in the image below). Then, select the entire row (by clicking on the little arrow (pointing at the right) at the beginning of the rows).



Click here to enlarge the image

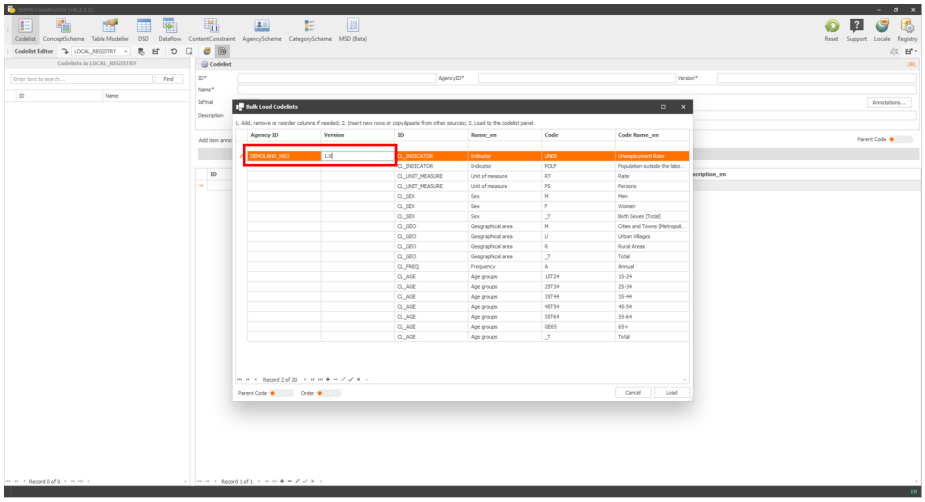
- After pasting, remember to delete the header row. You can do this by selecting the entire row (by clicking again the little arrow pointing at the right at the beginning of the rows) and clicking the button ("-") below, as indicated by a downward red arrow.



Click here to enlarge the image

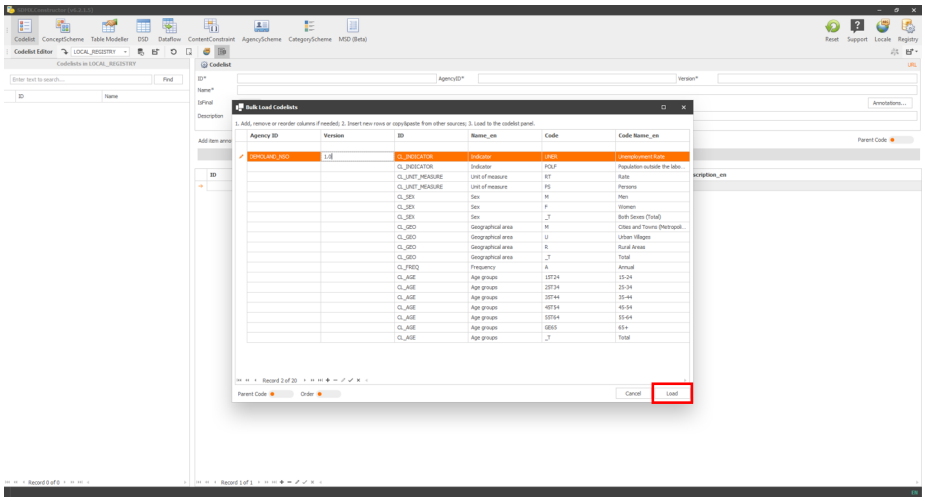
- Enter the Agency ID and Version on the top row as DEMOLAND_NSO and 1.0, respectively, as shown below. If only the top row contains entries

(DEMOLAND_NSO and 1.0) and the rest is empty, it implies that the Agency ID and Version are repeated for each row.



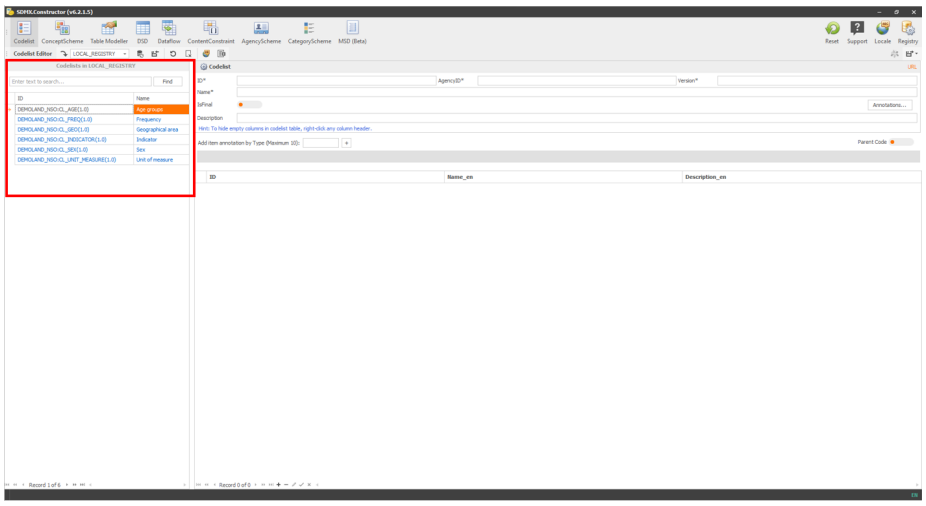
Click here to enlarge the image

- Click on Load, as shown below.



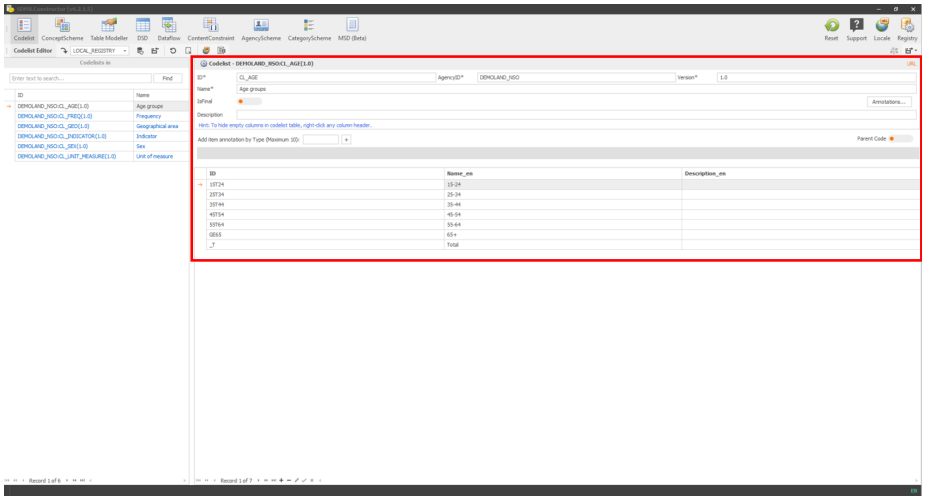
Click here to enlarge the image

- After the loading, this is how it would look (as shown below).



Click here to enlarge the image

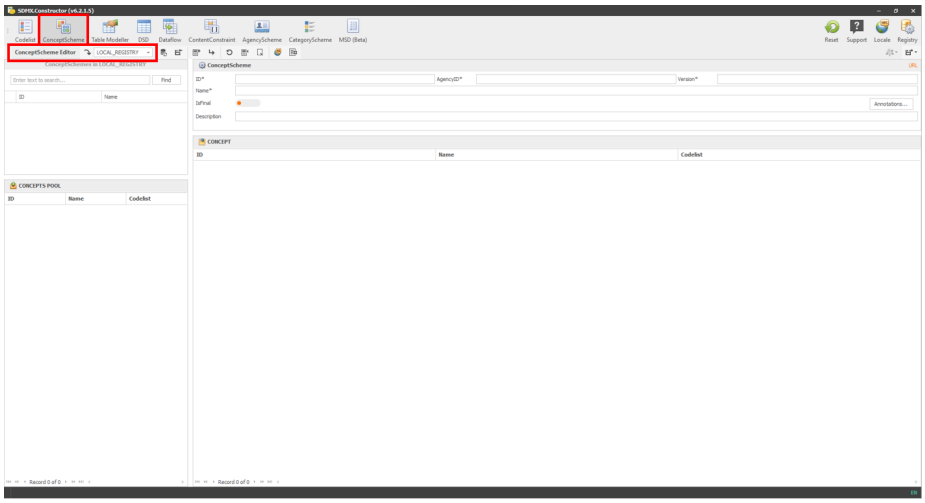
- Clicking on any item on this list will show the details on the right pane, as shown below.



Click here to enlarge the image

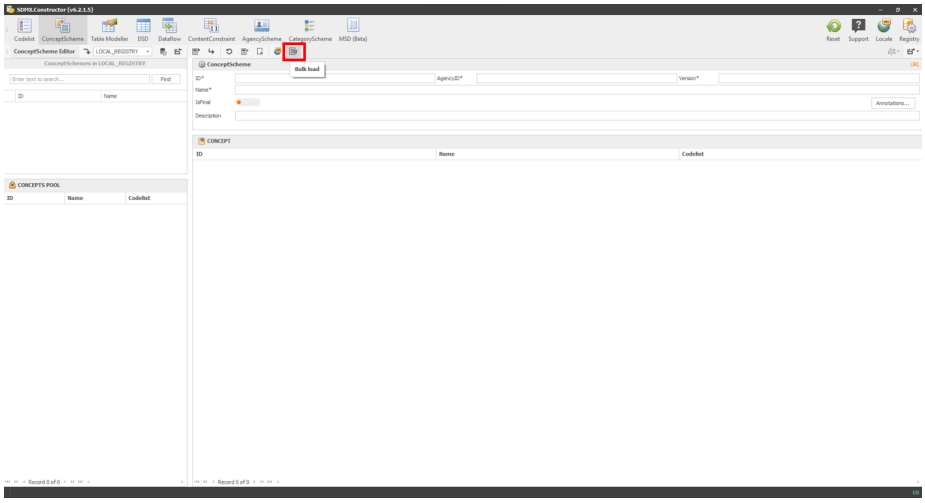
Create ConceptScheme:

- Click on the ConceptScheme button on top and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the ConceptScheme Editor's 'Load from registry' dropdown option, as shown below.



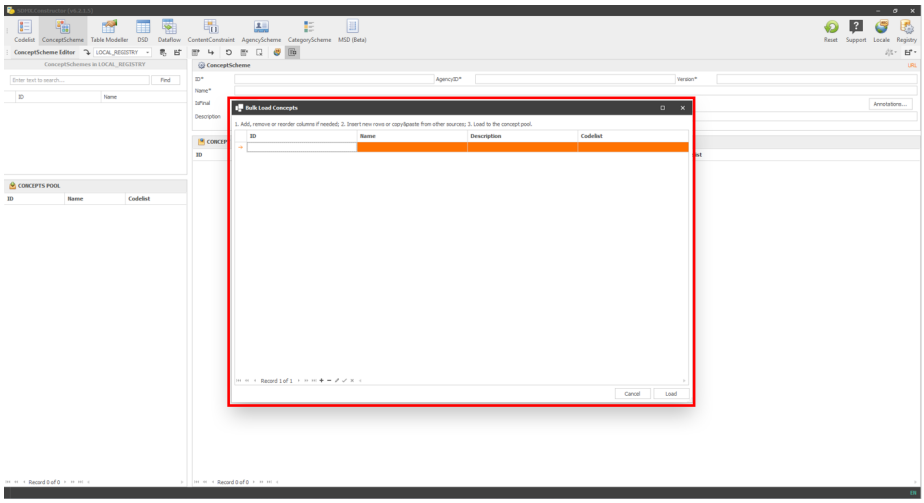
Click here to enlarge the image

- Click on the Bulk load button, as shown below.



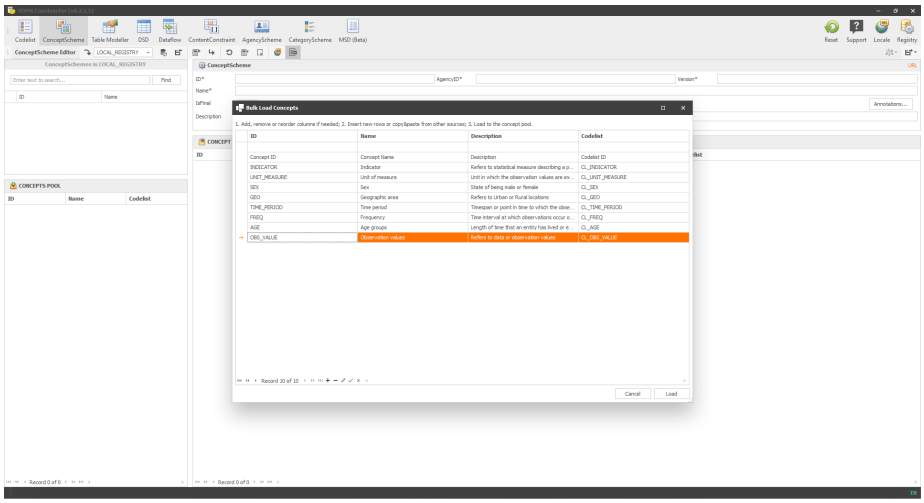
Click here to enlarge the image

- It will open up a pop-up window, as shown below.



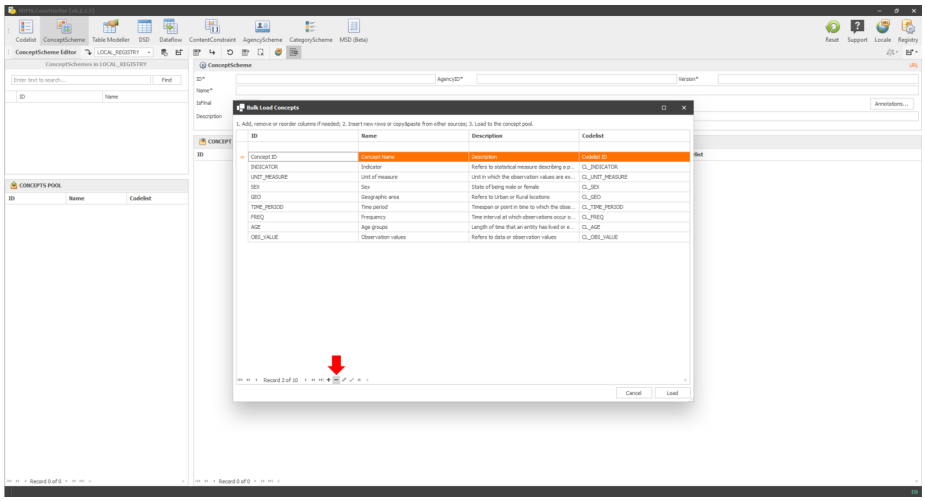
Click here to enlarge the image

- Copy the ConceptScheme table (Table: 4.3) we prepared before and paste its contents here. Before pasting, remember to click on the ID column and select the entire row (by clicking on the little arrow at the beginning of the rows).



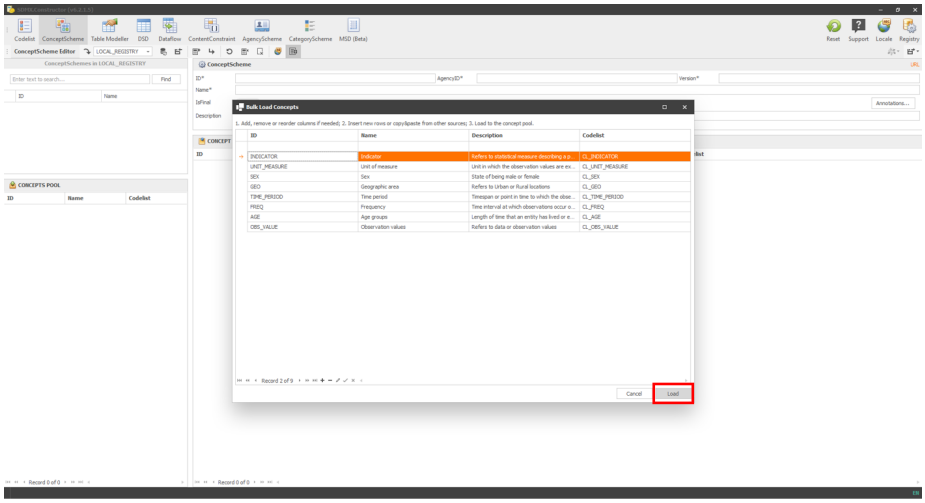
Click here to enlarge the image

- After pasting, remember to delete the header row by selecting the entire row and using the button ("-") below, as indicated by a downward red arrow.



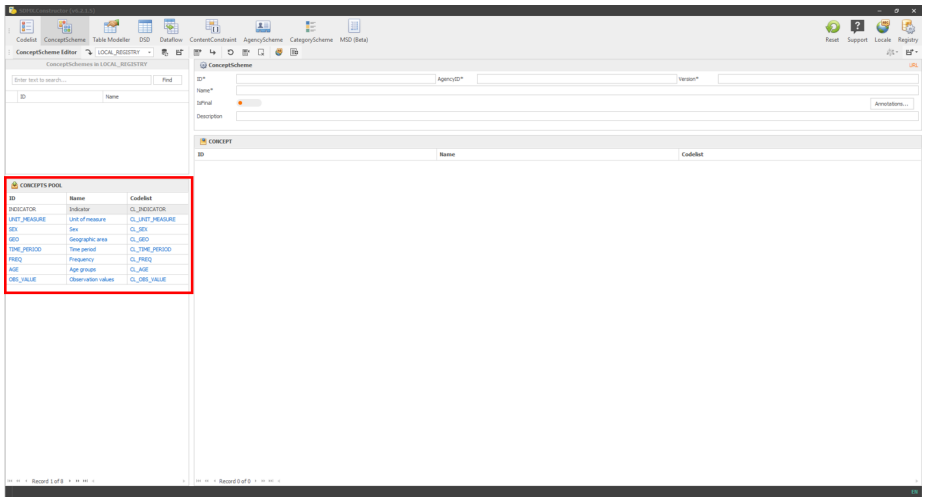
Click here to enlarge the image

- Click on Load, as shown below.



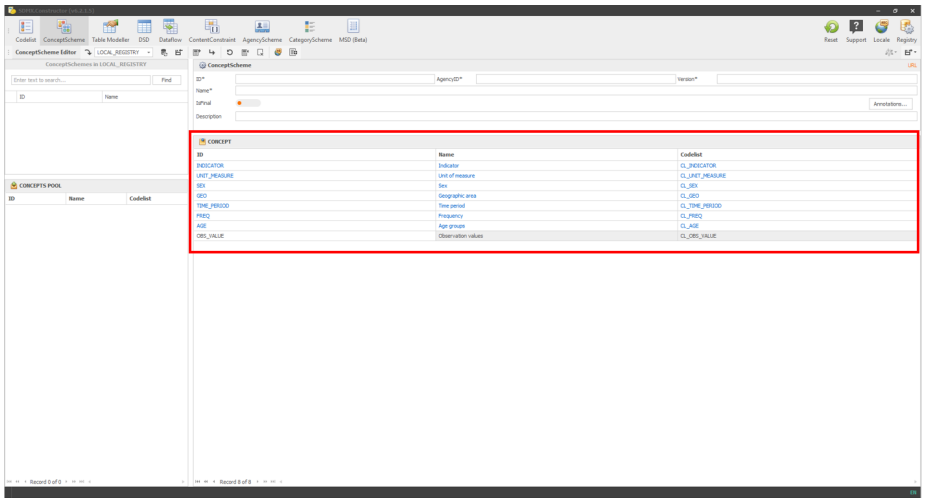
Click here to enlarge the image

- After loading, the concepts will be visible in the CONCEPT POOL, as shown below.



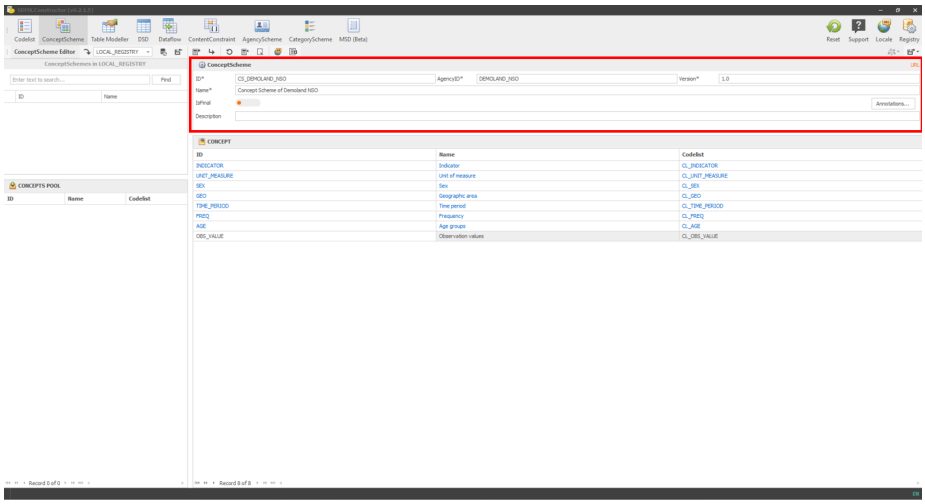
Click here to enlarge the image

- Move all the concepts from the CONCEPT POOL to the CONCEPT pane by selecting all (ctrl + a), then dragging and dropping. After the move, it would look like the following.



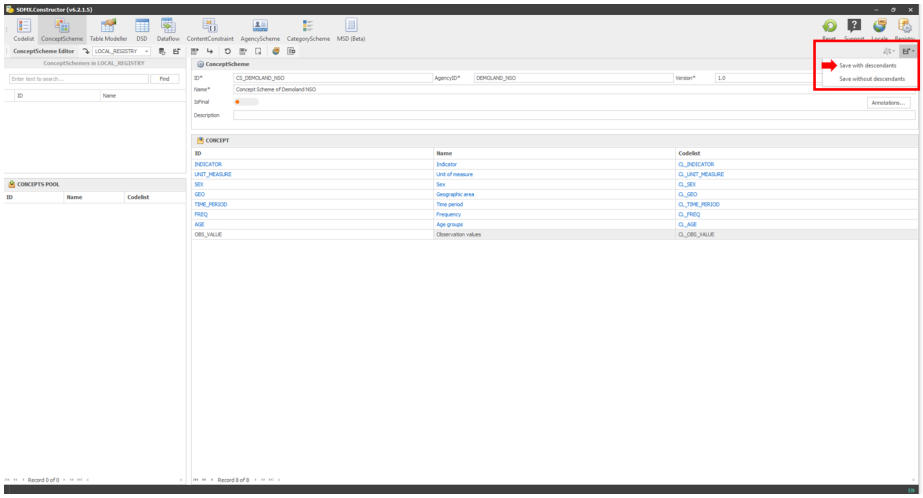
Click here to enlarge the image

- After moving the concepts, enter the details: (ID: CS_DEMOLAND_NSO, AgencyID: DEMOLAND_NSO, Version: 1.0, and Name: Concept Scheme of Demoland NSO) for the ConceptScheme, as shown below.



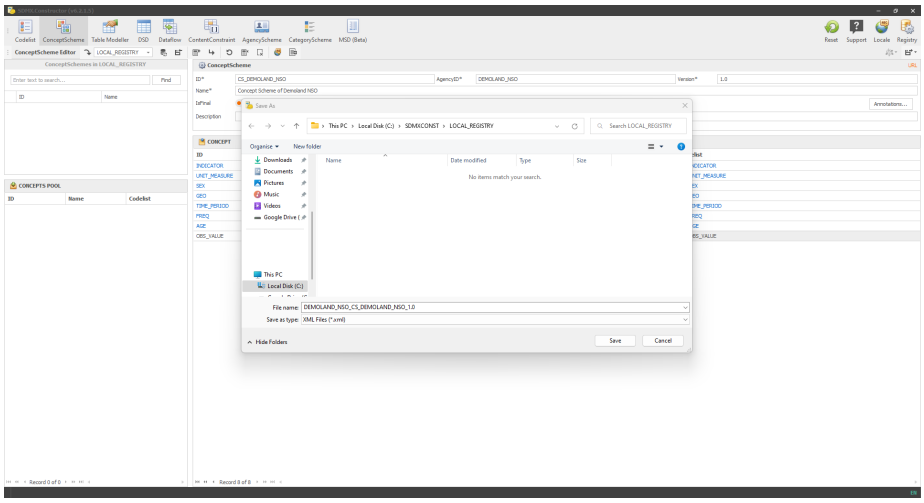
Click here to enlarge the image

- Then, click the 'Save with descendants' from the save option as shown below. This option, 'Save with descendants,' will save the concept scheme with the codelist.



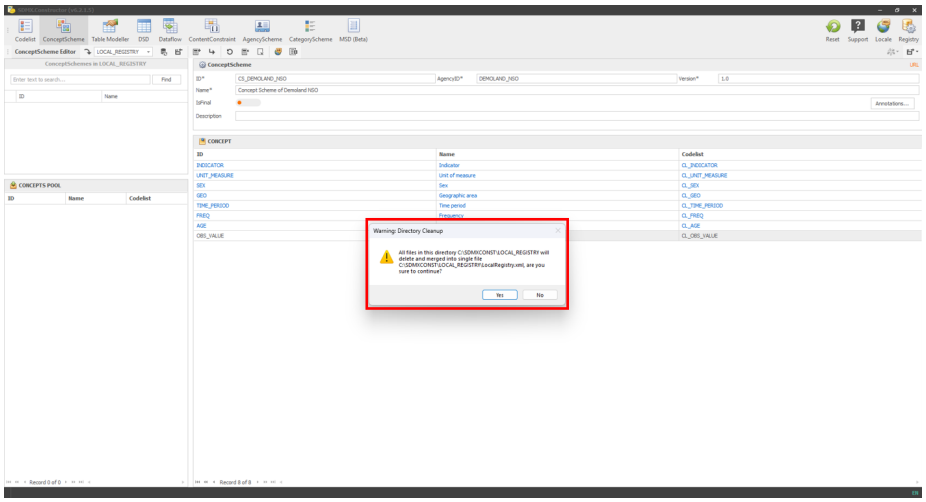
Click here to enlarge the image

- A pop-up message will ask to save the XML file in the folder (LOCAL_REGISTRY) we created before. Click on Save to save the file.



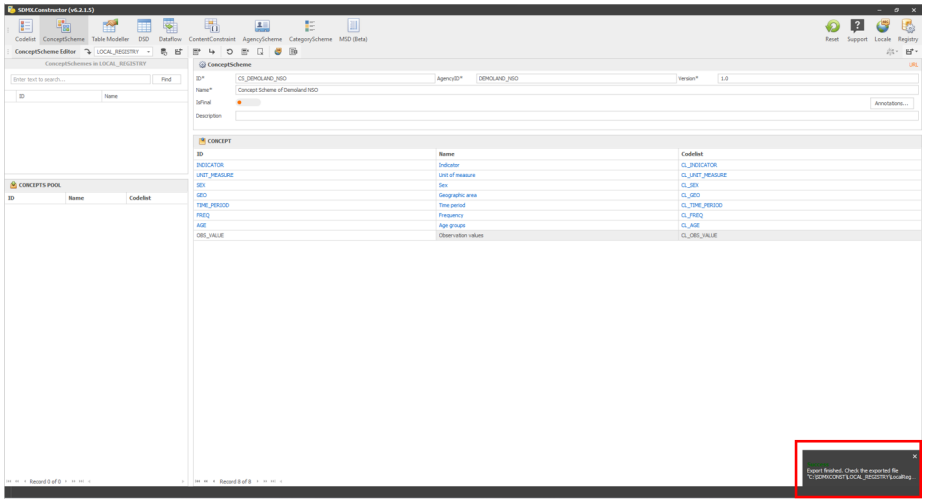
Click here to enlarge the image

- After clicking Save, the tool will ask the question to merge files. Select Yes.



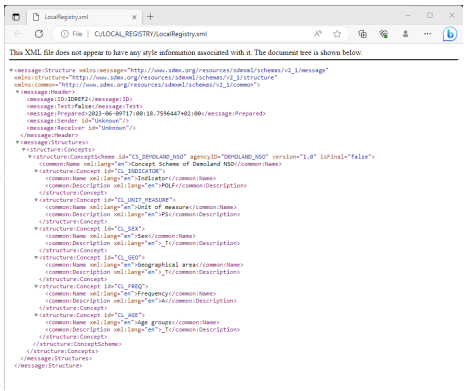
Click here to enlarge the image

- A confirmation message will appear for a short time at the bottom right corner, as shown below.



Click here to enlarge the image

- After that, if you go to the file's location, you will see the XML file created, as shown below.



Click here to enlarge the image

- Opening the XML file will show the details containing, AgencyScheme, ConceptScheme and Codelists.

4.6 Creating DSD, ContentConstraint and Dataflow

In SDMX, a Data Structure Definition (DSD) is used to organise data in a specific format. Represented as a cube, a DSD is made up of dimensions and attributes. A Dataflow, on the other hand, represents a specific portion or filtered view of the cube that the DSD represents. A Content Constraint specifies the permitted code list and codes for a particular Dataflow.

In SDMX Constructor, we can create DSDs, Dataflows and Concent Constraints either by "one by one" approach or in one go using the Table Modeller functionality.

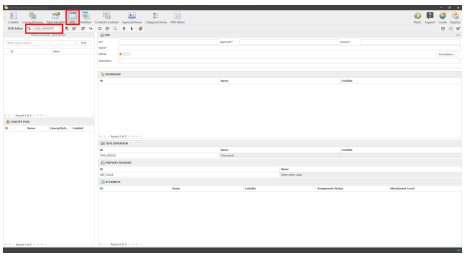
We will first see how to apply the "one by one" approach and, subsequently, the Table Modeller functionality.

The "one by one" approach

Creating DSD

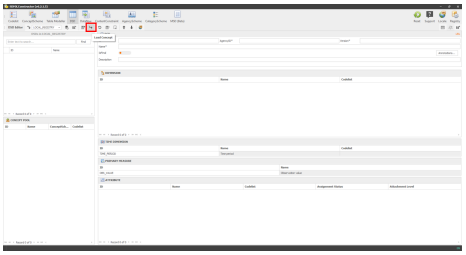
The following steps assume you already have a ConceptScheme created and the corresponding SDMX-ML (XML) file is saved on your computer.

- Start the SDMX Constructor. Click on the DSD button in the Editor menu and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the 'Load from registry' dropdown option, as shown below.



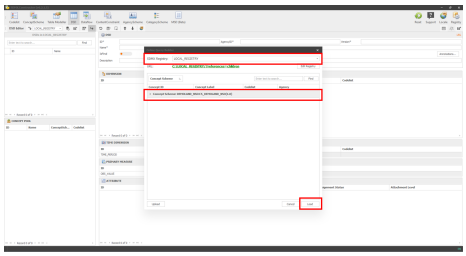
Click here to enlarge the image

- Click on the 'Load Concepts' in the Editor Ribbon menu, as shown below.



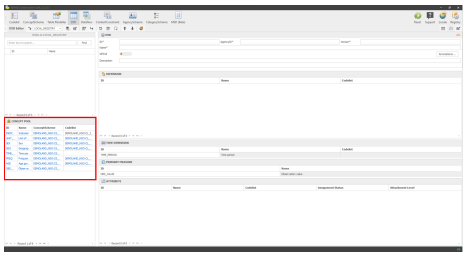
Click here to enlarge the image

- In the resulting window, as shown below, select the SDMX Registry (which will be LOCAL_REGISTRY in this case). Then click on Load, as shown below.



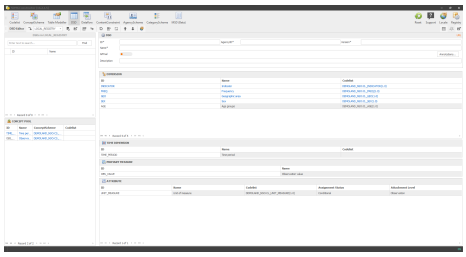
Click here to enlarge the image

The concepts will be visible in the CONCEPT POOL, as shown below.



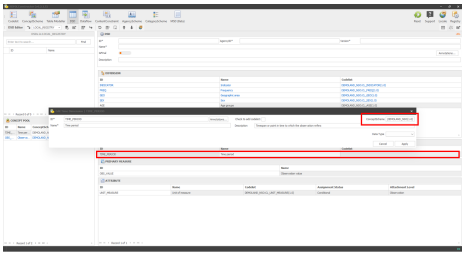
Click here to enlarge the image

- Drag and drop the concepts from the CONCEPT POOL to either the DIMENSION or the ATTRIBUTE pane on the right. For our example data, except for the Unit of Measurement, which will go to the ATTRIBUTE pane, all other concepts will be in the DIMENSION pane, as shown below.



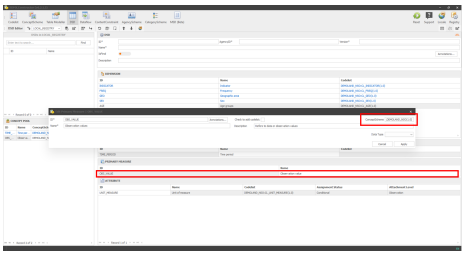
Click here to enlarge the image

- Note that by default, SDMX Constructor has preselected TIME DIMENSION and PRIMARY MEASURE on the right pane. However, ensure that the ConceptScheme fields in both are set to the same (DEMOLAND_NSO:CS_DEMOLAND_NSO(1.0)) as other dimensions. Double-clicking the TIME DIMENSION will open up the pop-up window where it can be specified, as shown below. Click Apply.



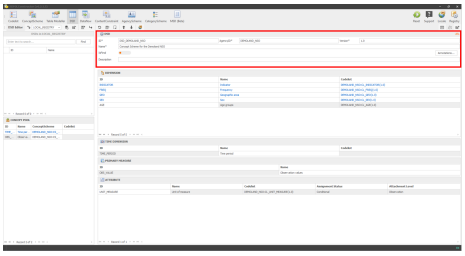
Click here to enlarge the image

- Repeat the same for the PRIMARY MEASURE as shown below. Click Apply.



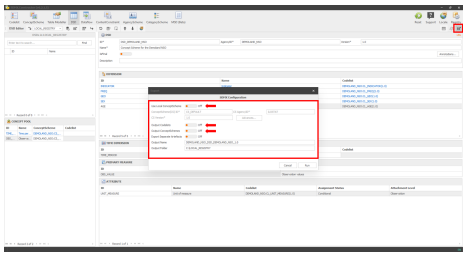
Click here to enlarge the image

- Fill in the details for the DSD (ID = DSD_DEMOLAND_NSO, AgencyID = DEMOLAND_NSO, Version = 1.0 and Name = Concept Scheme for the Demoland NSO) as shown below.



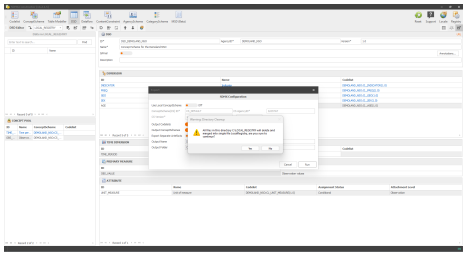
Click here to enlarge the image

- Clicking on the Export/Save button, as shown below, will open up a pop-up window where you can make the following adjustments: Use Local ConceptScheme - turn it off; output Codelists - turn it off and Output ConceptSchemes - turn it off.



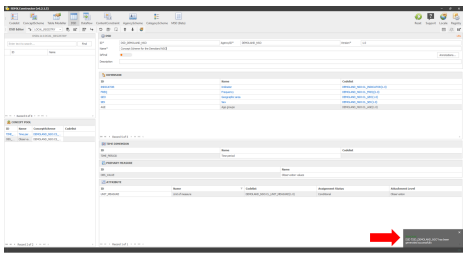
Click here to enlarge the image

- Clicking on Run will show the following message. Click on Yes.



Click here to enlarge the image

- You will see a message confirming the creation of the DSD, as shown below.

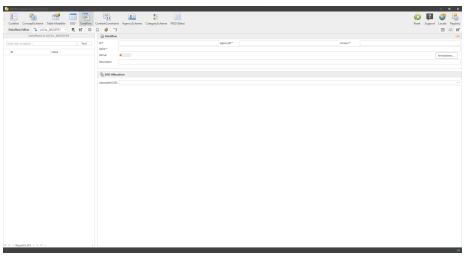


Click here to enlarge the image

- After that, you will see the XML file created if you go to the file's location.

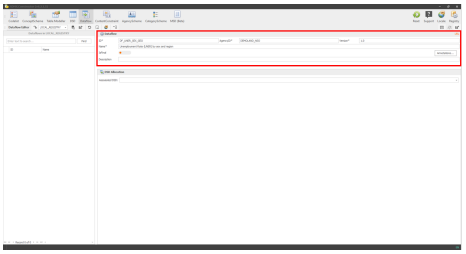
Creating Dataflow

- Click on the Dataflow button in the Editor menu and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the 'Load from registry' dropdown option, as shown below.



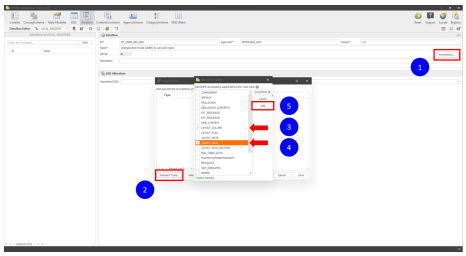
Click here to enlarge the image

- For this example of Dataflow, we will select 'Table 4.1: Unemployment Rate by sex and region'. We will first enter the details (ID = DF_UNER_SEX_GEO, AgencyID = DEMOLAND_NSO, Version = 1.0 and Name: Unemployment Rate (UNER) by sex and region) as shown below.



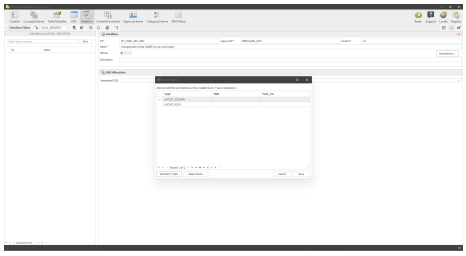
Click here to enlarge the image

- Then click on the Annotation button. After that, click on Standard Types in the pop-up window. Then select the LAYOUT_COLUMN and LAYOUT_ROW from the list. Finally, click on the Add button, as shown below.



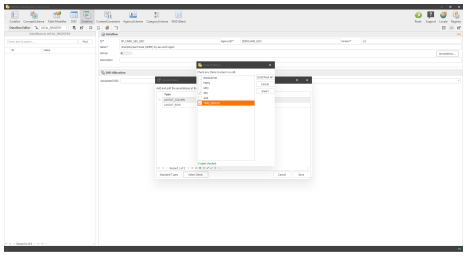
Click here to enlarge the image

- You will see the following window.



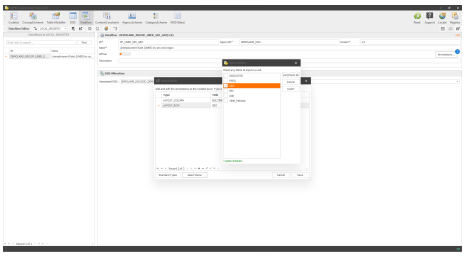
Click here to enlarge the image

- Select the LAYOUT_COLUMN. Then click in the Title column field. Then click on the Select Items button. Then following the layout scheme of Table 4.1, select the items for LAYOUT_COLUMN: as SEX and TIME_PERIOD. Finally, click on the Insert button.



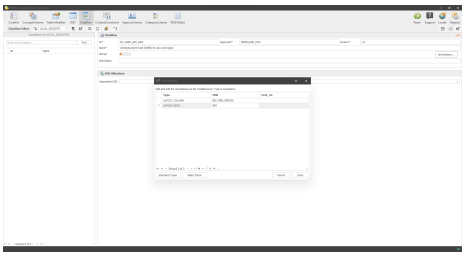
Click here to enlarge the image

- Repeat the same for the LAYOUT_ROW. Click inside the corresponding field of the column 'Title', then click on the Select Items button. Then select GEO (as per Table 4.1) and click on the Insert button.



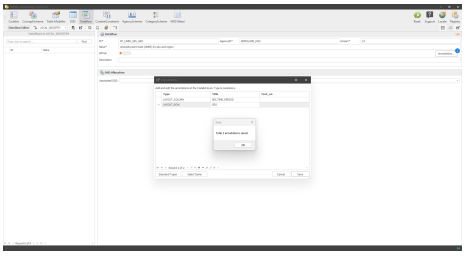
Click here to enlarge the image

- This is how it would look after both entries.



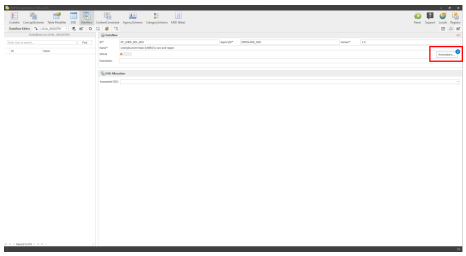
Click here to enlarge the image

- Clicking on the Save button will result in the following confirmation.



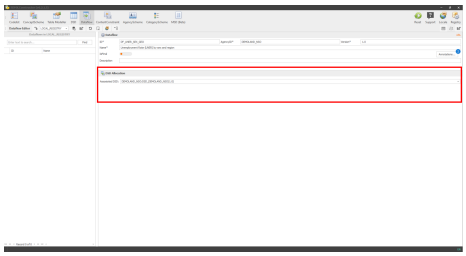
Click here to enlarge the image

- Click OK. It will now show the 2 annotation notifications as highlighted below.



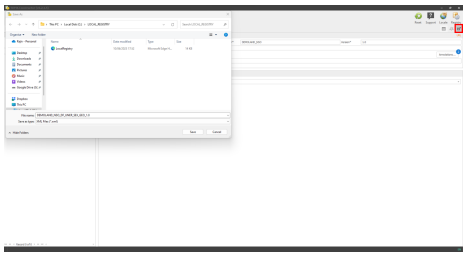
Click here to enlarge the image

- Now, in the DSD Allocation pane, in the Associated DSD dropdown, select the DSD on which this Dataflow will be based. We selected the DSD we created before - DEMOLAND_NSO:DSD_DEMOLAND_NSO(1.0).



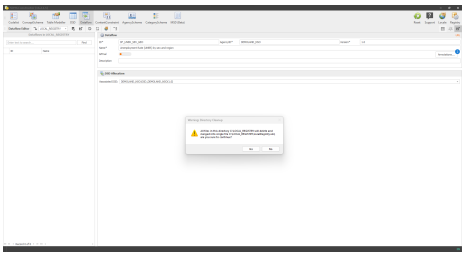
Click here to enlarge the image

- Now click on the Export/Save button, as shown below and click on Save.



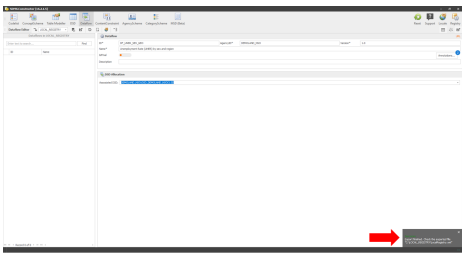
Click here to enlarge the image

- It will show a message as shown below.



Click here to enlarge the image

- Clicking on Yes will show a message indicating that the XML file (containing the Dataflow) has been created.

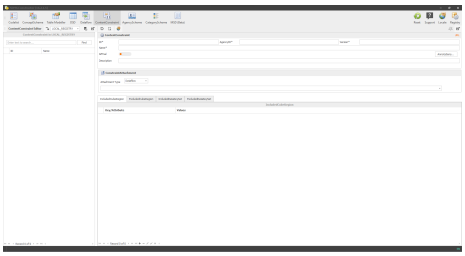


Click here to enlarge the image

- After that, you will see the XML file created if you go to the file's location.

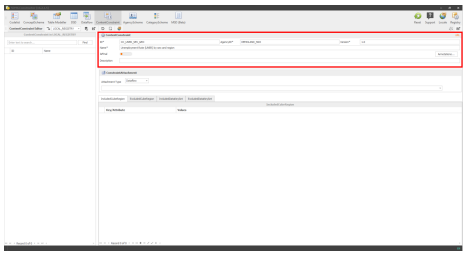
Creating ContentConstraint

- Click on the ContentConstraint button in the Editor menu and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the 'Load from registry' dropdown option, as shown below.



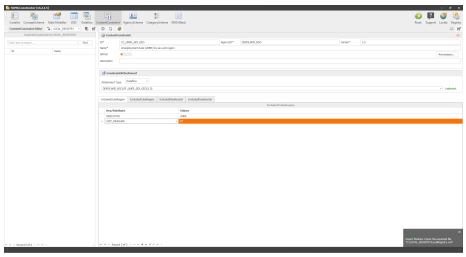
Click here to enlarge the image

- Fill in the details for the ContentConstraint (ID = CC_UNER_SEX_GEO, AgencyID = DEMOLAND_NSO, Version = 1.0 and Name = Unemployment Rate (UNER) by sex and region) as shown below.



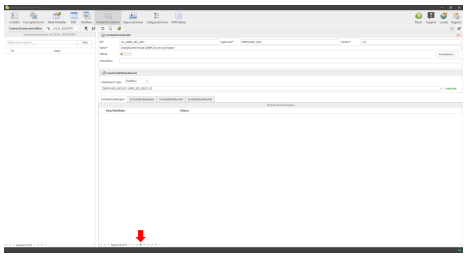
Click here to enlarge the image

- Then in the ConstraintAttachment pane, in the Attachment Type, select the Dataflow option and from the dropdown list underneath, select the relevant dataflow as shown below.



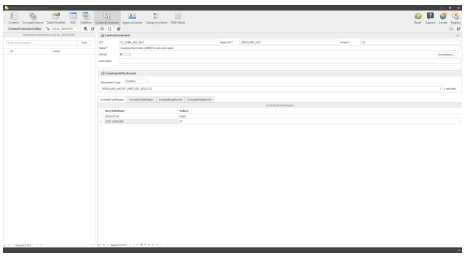
Click here to enlarge the image

- Then click the append (or +) button to add the required rows.



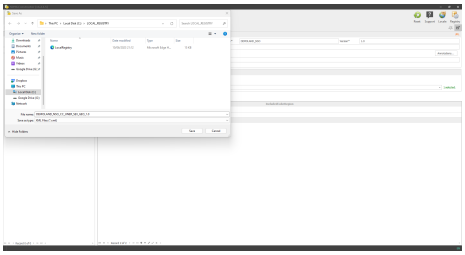
Click here to enlarge the image

- We add two rows. In the first row, we select the concept 'Indicator' and the value as UNER (for Unemployment Rate). In the second row, we select the concept 'UNIT_Measure' (for Unit of Measurement) and the value as RT (for Rate), as shown below.



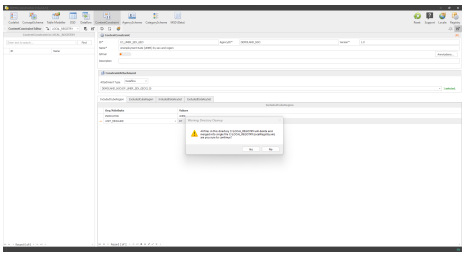
Click here to enlarge the image

- Now click on the Export/Save button, as shown below and click on Save in the resulting window.



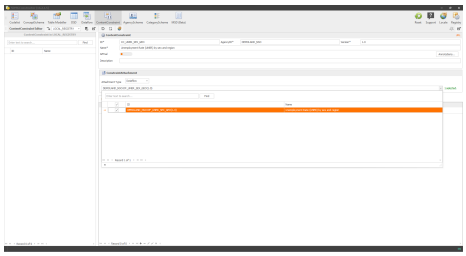
Click here to enlarge the image

- It will show a message as shown below.



Click here to enlarge the image

Clicking on Yes will show a message in the bottom right-hand corner indicating that the XML file is created.



Click here to enlarge the image

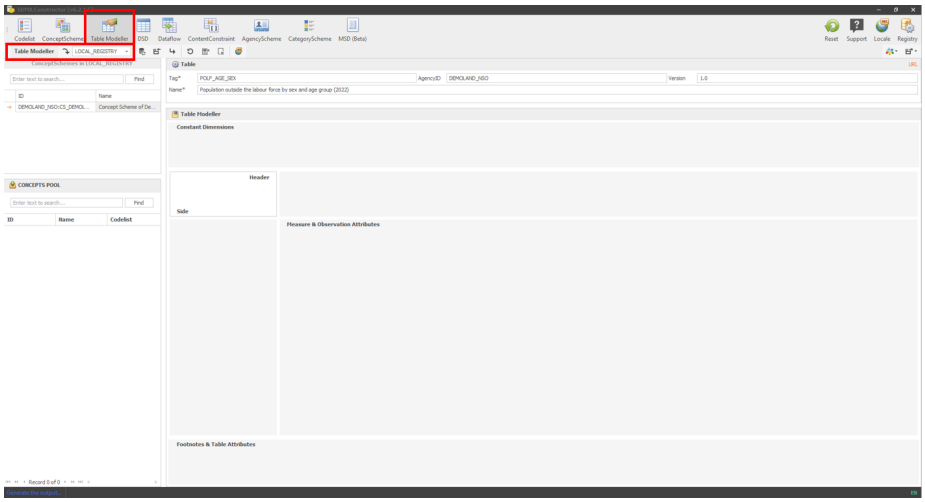
After that, you will see the XML file created if you go to the file's location.

The Table Modeller approach

In SDMX Constructor, the Table Modeller feature (Under special topics, the functionality of the Table Modeller is detailed) provides an intuitive user interface for designing statistical tables and generating the corresponding SDMX artefacts in one go.

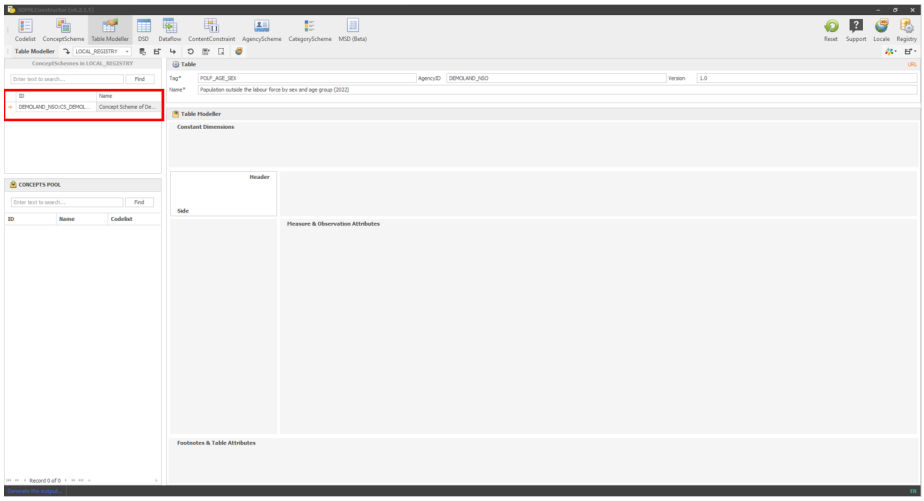
Recalling our two initial tables (Table: 4.1 and Table: 4.2), we will now generate DSDs, ContentConstraints and Dataflows. We can create all these artefacts through the Table Modeller option in the SDMX Constructor.

- Click on the Table Modeller button on top and ensure that the folder we created before, LOCAL_REGISTRY, is selected from the Table Modeller Editor's 'Load from registry' dropdown option, as shown below.



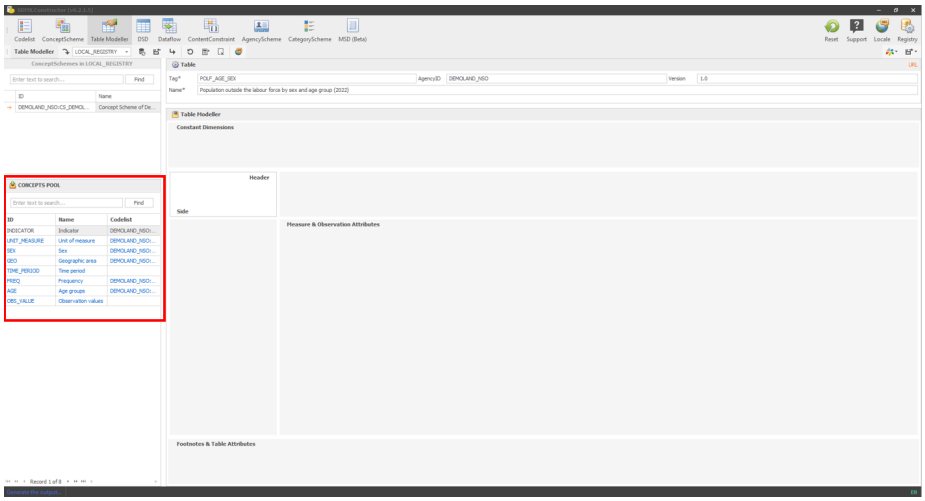
Click here to enlarge the image

- You will notice the concept scheme we created before, as highlighted below.



Click here to enlarge the image

- Double-clicking the concept scheme will move it to the CONCEPT POOL, as shown below.

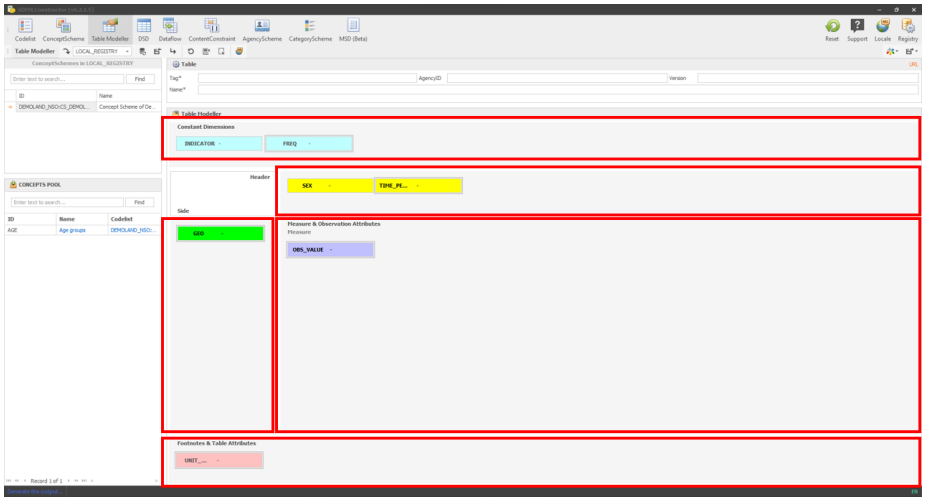


Click here to enlarge the image

- Now, you can select the concepts in the CONCEPT POOL, then drag and drop them into various spaces (Constant Dimensions, Header, Side,

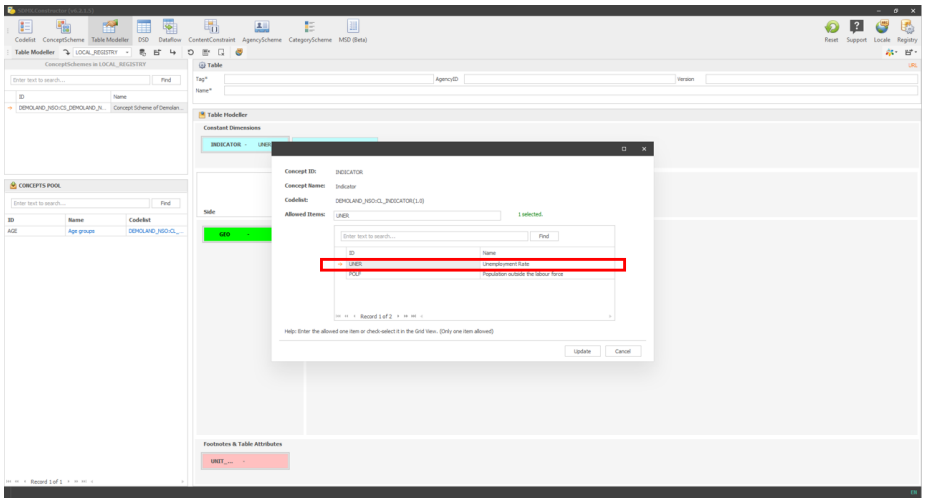
Measures & Observation Attributes and Footnotes & Table Attributes) on the right pane. The logic driving where to drop each concept comes from our original tables (Table: 4.1 and Table: 4.2).

- For Table 4.1 (Unemployment Rate by sex and region), the following distribution will be representative.



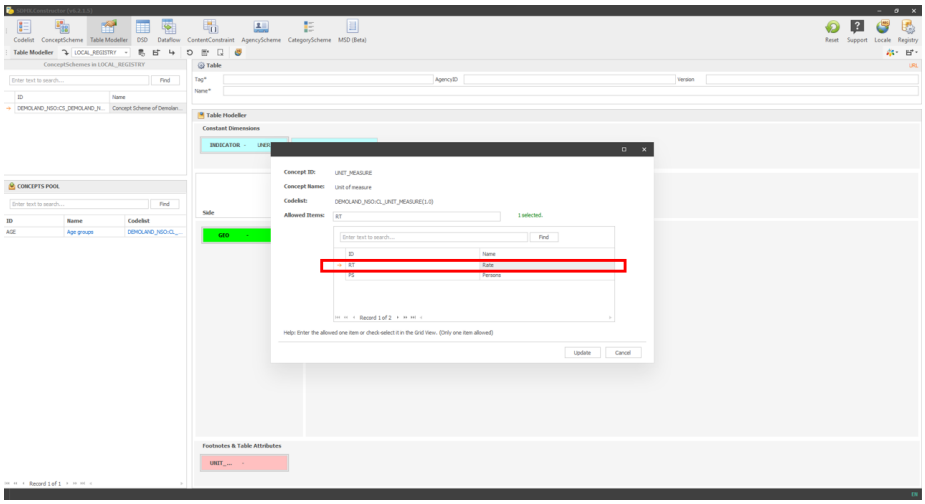
Click here to enlarge the image

- We can now apply two constraints: one for the indicator's name and the other for the unit of measure. The indicator's name will be the Unemployment Rate, which we can select by double-clicking the moved indicator concept as shown below.



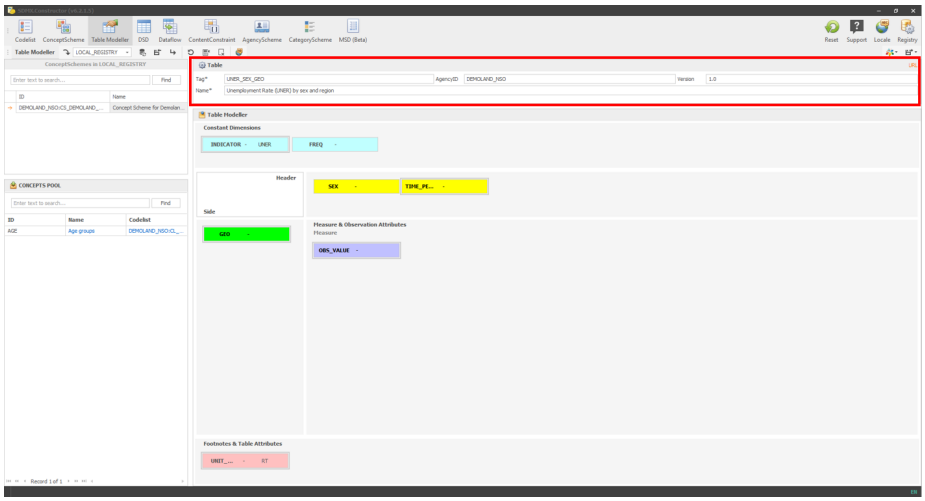
Click here to enlarge the image

- Another constraint would be the unit of measure. Because it is ‘rate’ (for Table 4.1), double-clicking the moved unit of measure concept would offer the option to select the rate. Select the Rate as shown below.



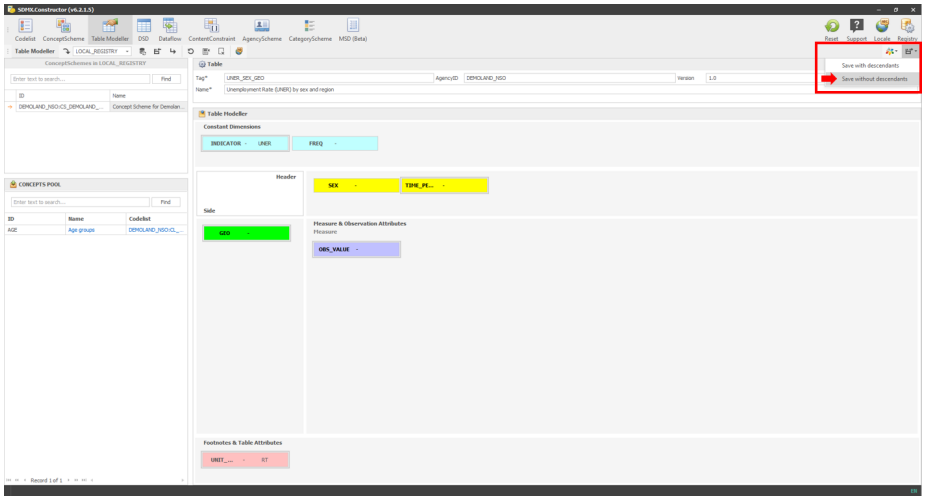
Click here to enlarge the image

- Now we add the table information as shown below.



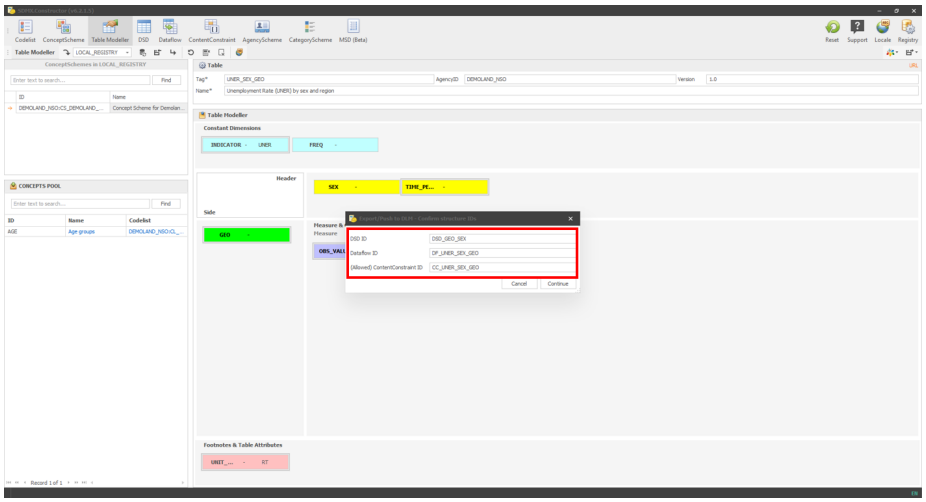
Click here to enlarge the image

- Then we will hit save (select 'Save without descendants'), as shown below.



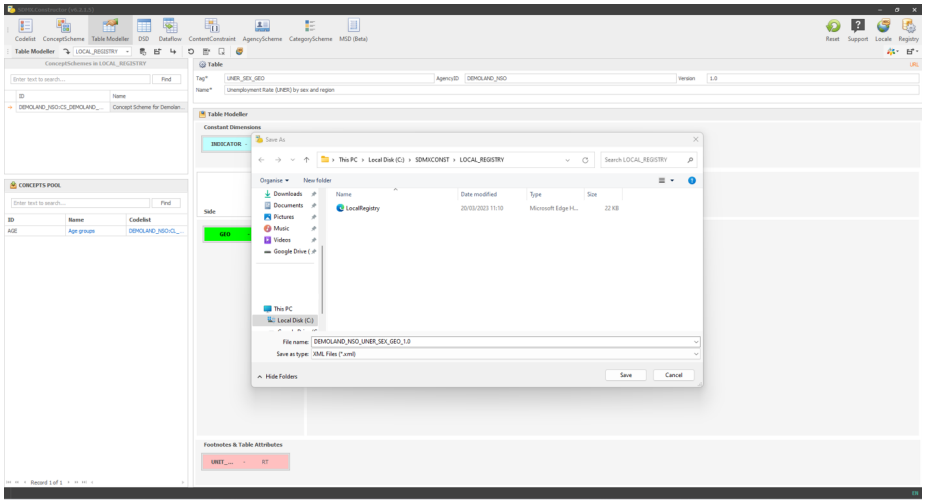
Click here to enlarge the image

- After saving, it will show a message as shown below.



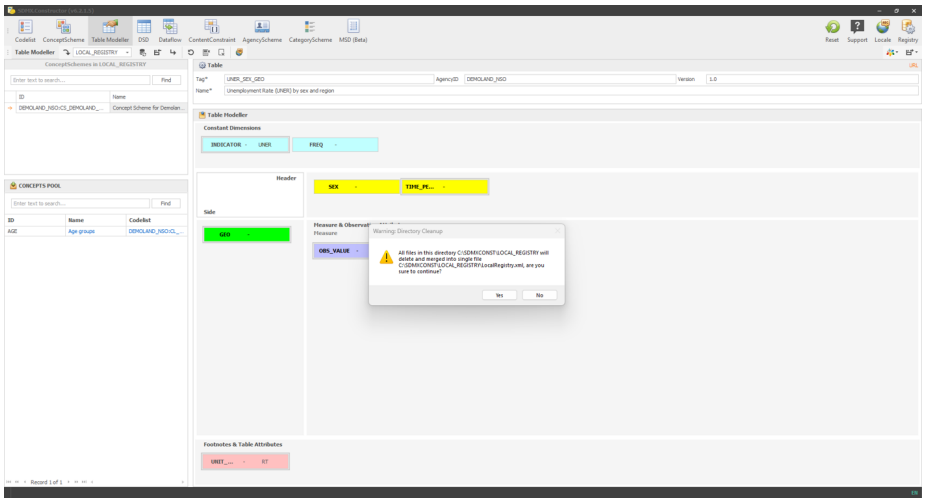
Click here to enlarge the image

- Clicking on Continue will result in a pop-up asking to save the file, as shown below.



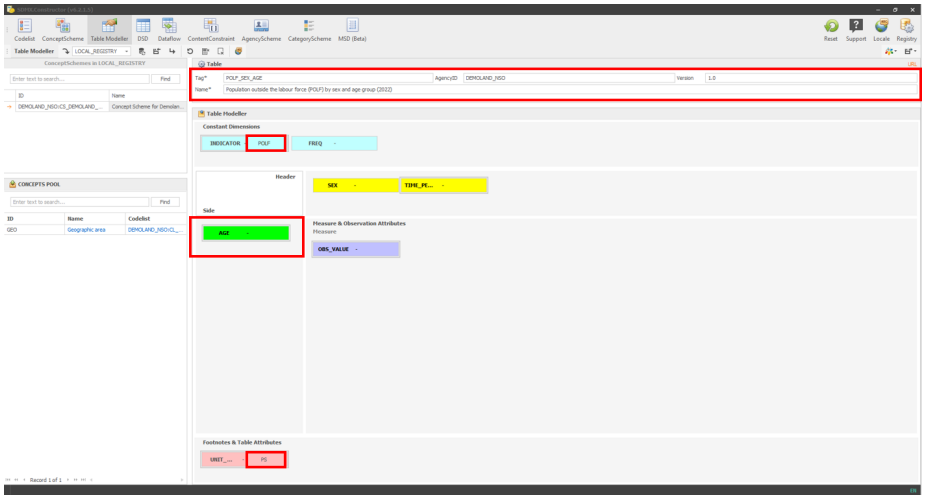
Click here to enlarge the image

- Clicking Save will show this message and ask if the files should be merged.



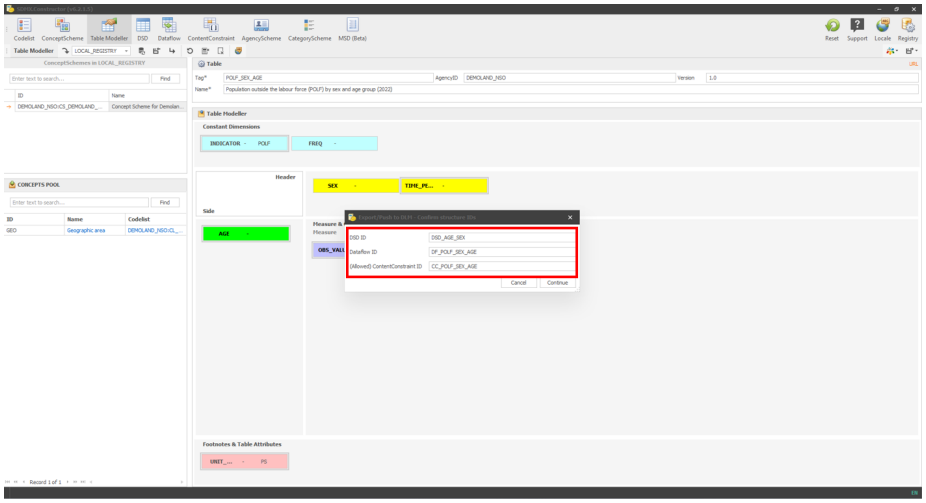
Click here to enlarge the image

- Clicking on Yes will merge the file.
- We will repeat the process for Table 4.2 (Population outside the labour force by sex and age group (2022)). We can do that by moving Age to the 'Side' space (and removing the Geo back to the CONCEPT POOL), changing the constraints for the indicator to 'Population outside the labour force' and unit of measure to 'Persons', and updating the table information as shown below.



Click here to enlarge the image

- After hitting the save (Save without descendants) button, the message will read like the one below.



Click here to enlarge the image

- Proceed with 'Continue', 'Save', and 'Yes' for merge.
- We should see two DSDs, two content constraints and two data flows. `Remember, dataflows are the filtered view of the DSDs.`

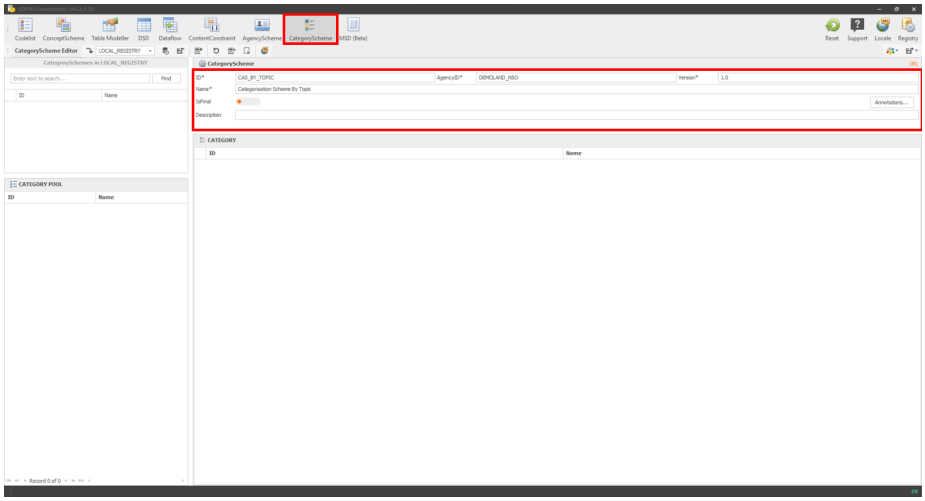
4.7 Creating CategoryScheme

A CategoryScheme groups similar objects based on shared characteristics, providing a hierarchical structure that helps classify and organise data and metadata meaningfully.

It is crucial to distinguish between topics and specific indicators within those topics. For instance, in this user manual, we have used two indicators as examples: 'Unemployment Rate by sex and region' and 'Population outside the labour force by sex and age group.' Although these indicators focus on different aspects, they both fall under the broader topic of the Labour force.

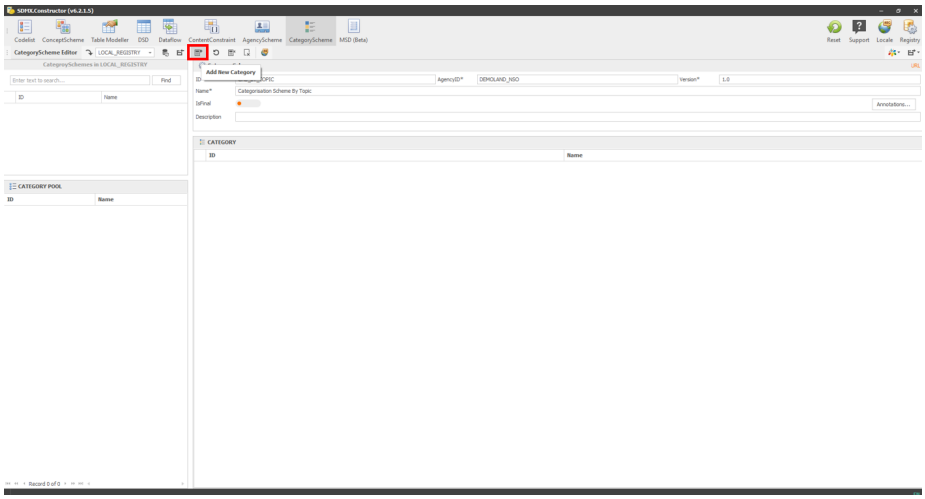
Referencing our two initial tables (Table: 4.1 and Table: 4.2), we will create a CategoryScheme in the SDMX Constructor. The categorisation would be by the topic 'Labour force'.

- First, we will click the CategoryScheme button and enter the properties (ID: CAS_BY_TOPIC, AgencyID: DEMOLAND_NSO, Version: 1.0 and Name: Categorisation Scheme By Topic) as shown below.



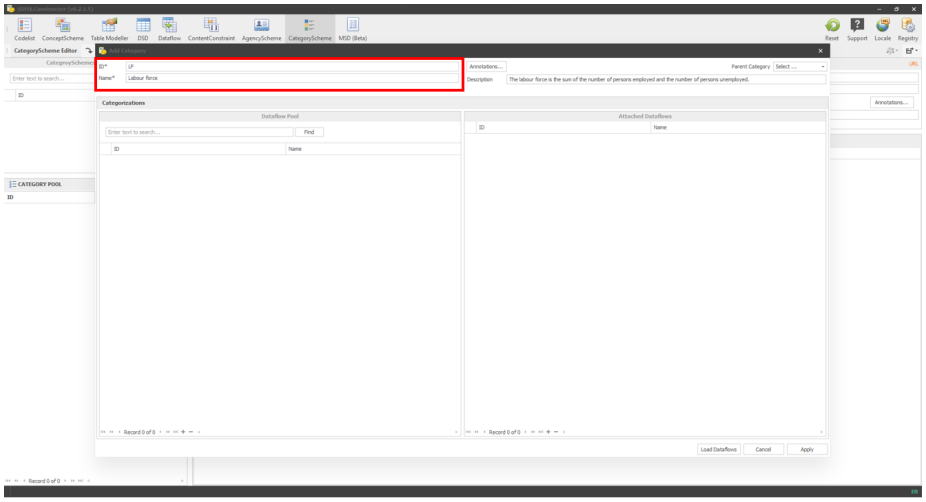
Click here to enlarge the image

- Click on “Add New Category”, as shown below.



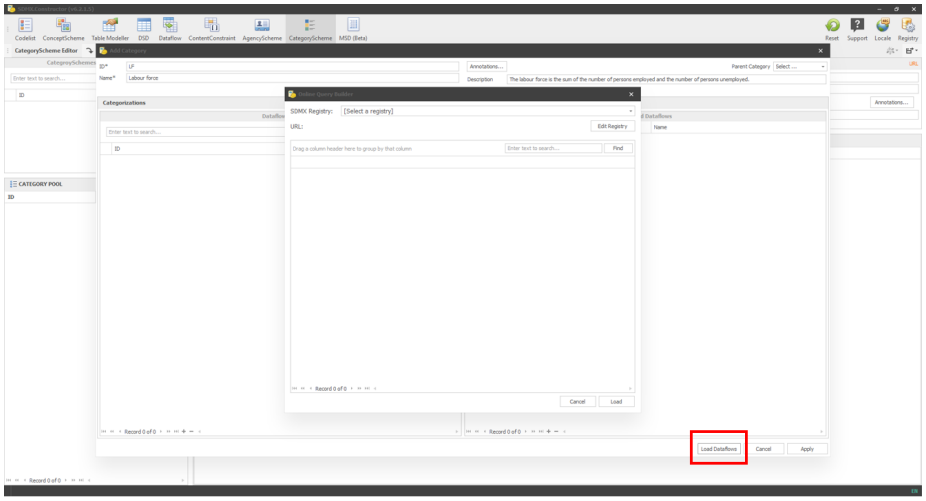
Click here to enlarge the image

- It will open a pop-up window, as shown below. Enter the details: ID: LF (shorthand for Labour force), Name: Labour force and Description: The labour force is the sum of the number of persons employed and the number of persons unemployed.



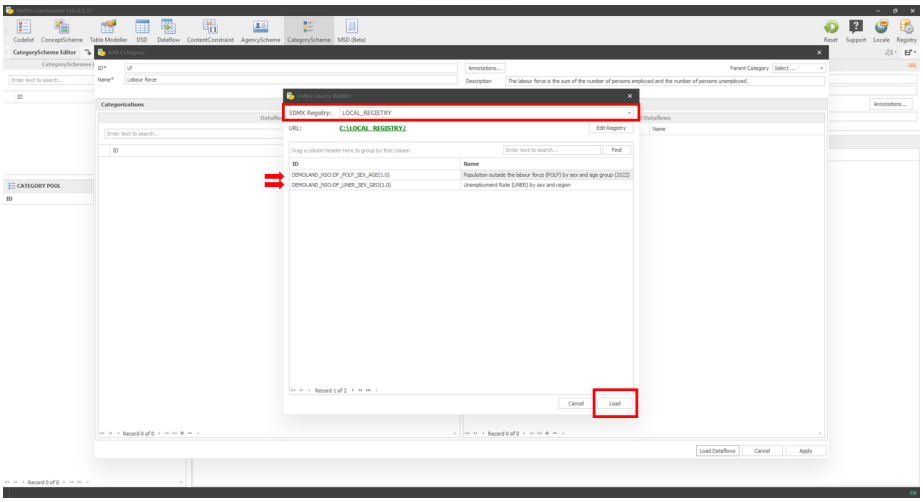
Click here to enlarge the image

- Click on the Load "Dataflows" button, as shown below. It will open another pop-up window to select the dataflows through the Registry.



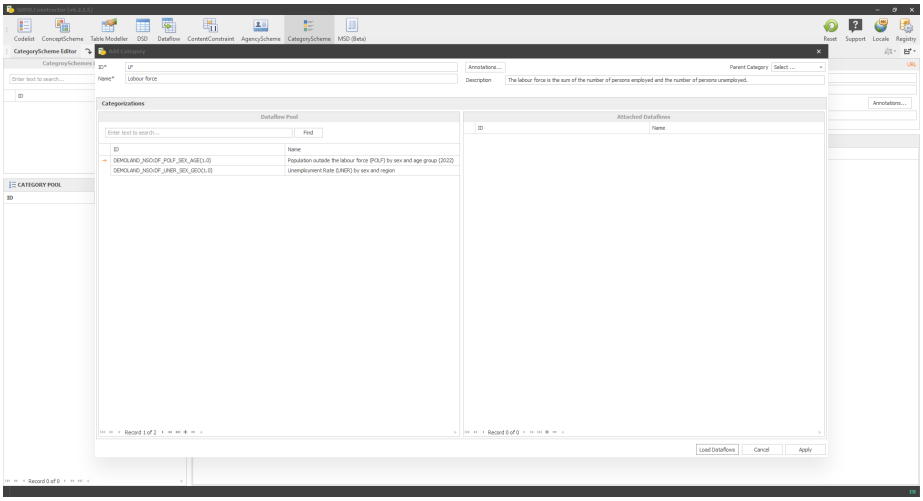
Click here to enlarge the image

- Select the Registry (LOCAL_REGISTRY), then the dataflows, and press Apply as shown below.



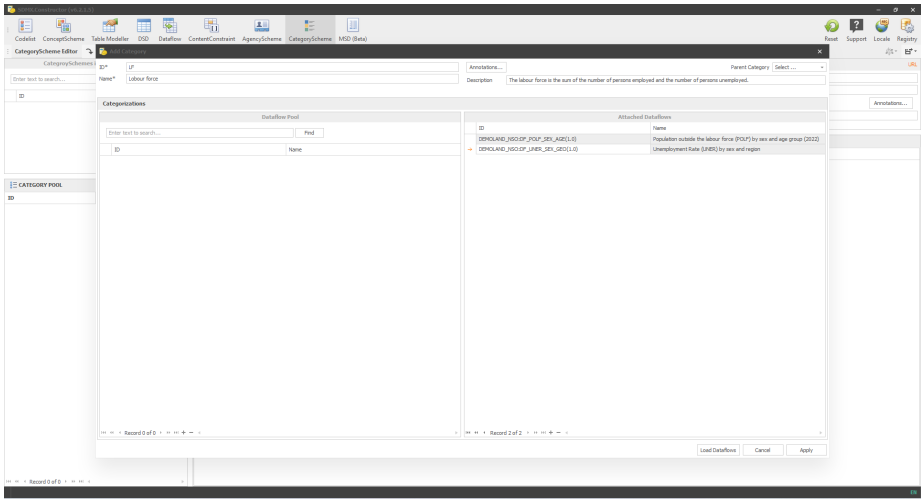
Click here to enlarge the image

- The resulting window will show the two dataflows in the dataflow pool, as shown below.



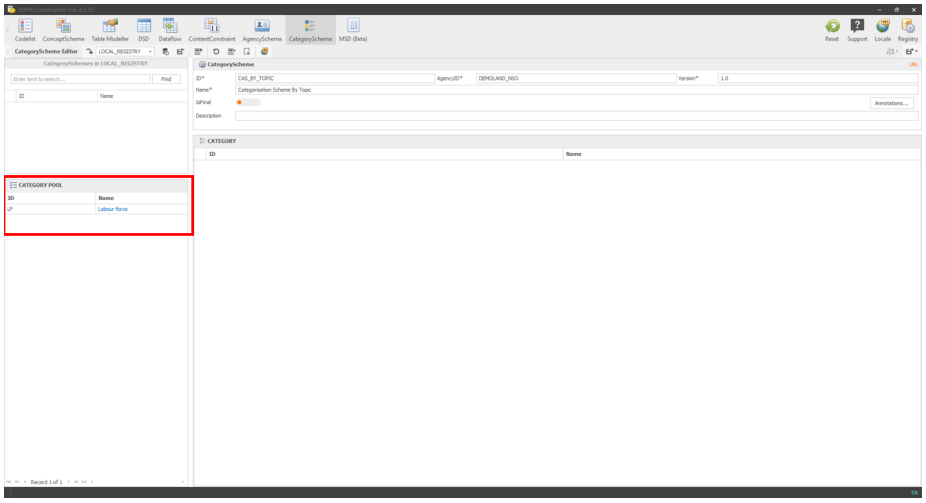
Click here to enlarge the image

- Select both the dataflows and move (by dragging and dropping) to the Attached Dataflows pane on the right, as shown below. Click on Apply.



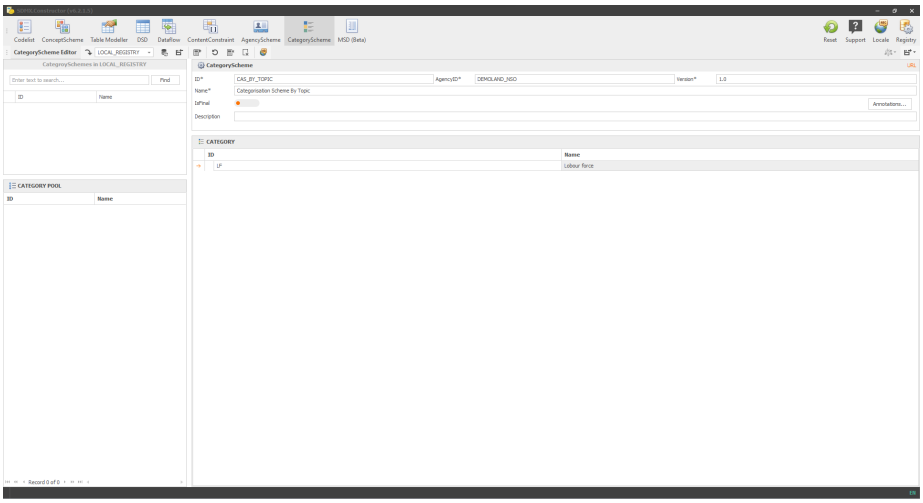
Click here to enlarge the image

- Clicking "Apply" in the previous step will take you to the following window, showing the entry into the CATEGORY POOL.



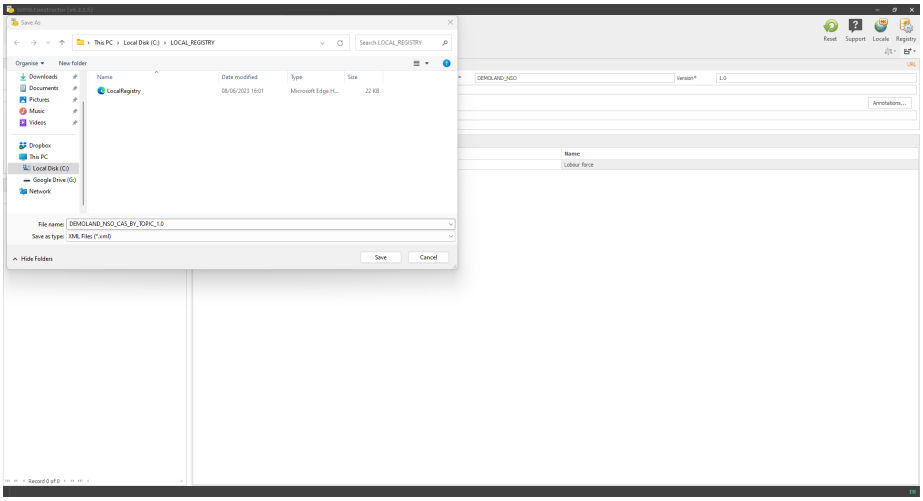
Click here to enlarge the image

- Move it to the right pane, as shown below.



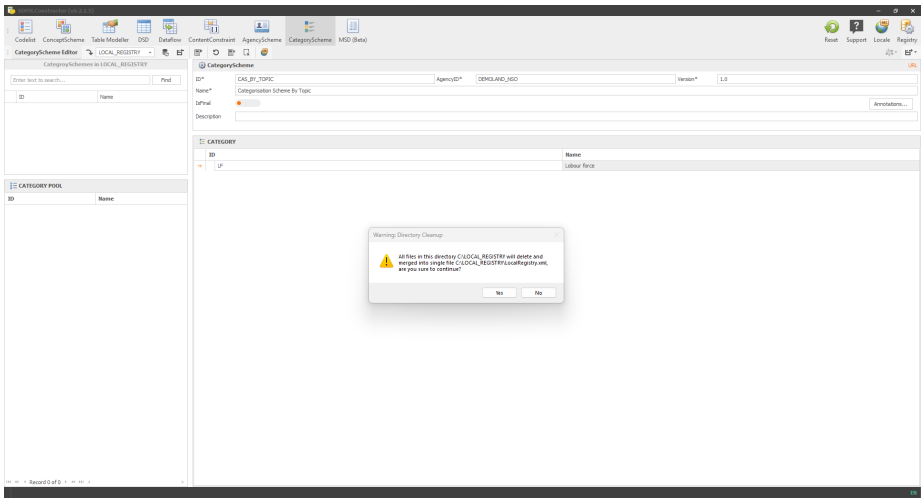
Click here to enlarge the image

- Then hit Save (Save without descendants), and it will show the pop-up windows to ensure the location where it saves the file, as shown below.



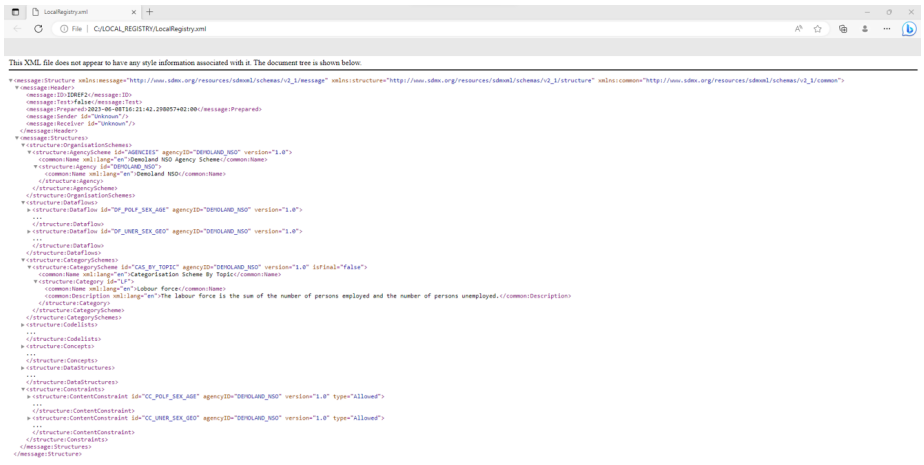
Click here to enlarge the image

- Clicking on Save will ask the confirmation to merge the file. Select Yes.



Click here to enlarge the image

- The XML file (by going to the location of the folder we created before and opening the XML file) would be like the one below.



Click here to enlarge the image

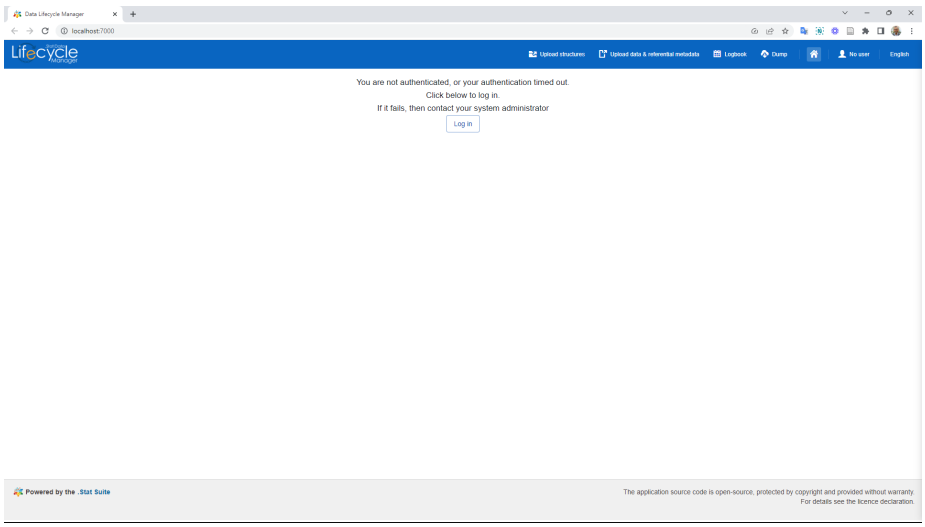
4.8 Uploading XML file to the DLM

You can upload the XML file containing the SDMX artefacts to the .Stat Data Lifecycle Manager (DLM). The DLM could be on the cloud or local computer

(Localhost). Here is one case illustrated where we upload the XML file we created and saved using the SDMX Constructor in a folder to a locally hosted instance of the DLM.

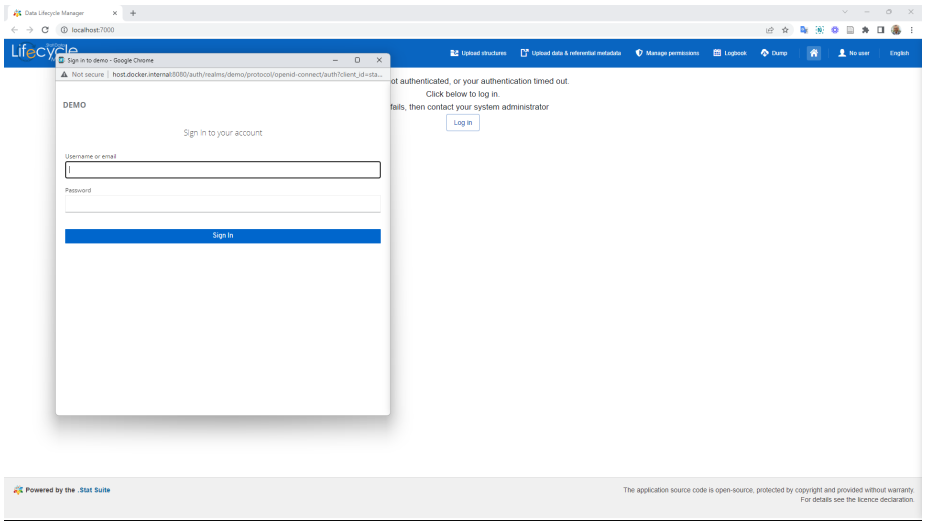
> **Note on artefact versioning**: Artefact versioning is crucial for managing SDMX artefacts, as it prevents conflicts when uploading structures to the DLM. This is especially important when multiple users work on the same artefact simultaneously or when updating artefacts over time. You may recall the usual place (in the right corner of the respective windows) to enter the version number in the user interface for creating or modifying artefacts in the SDMX Constructor.

- Start the DLM. Configured as localhost, it will look like the following.



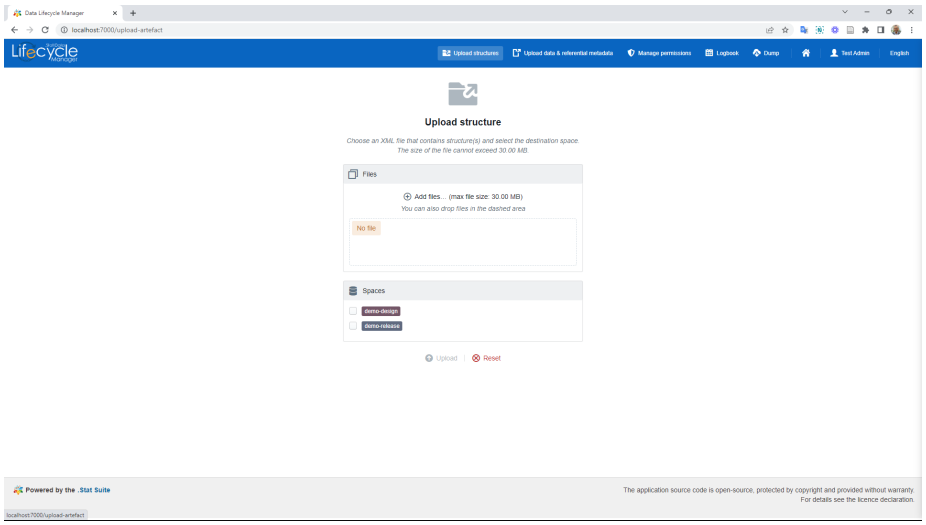
Click here to enlarge the image

- Login using your credentials.



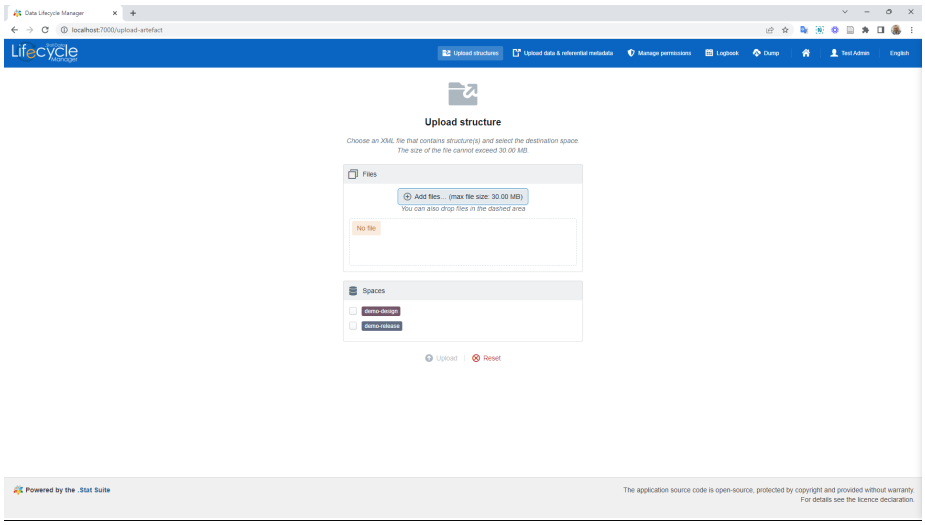
Click here to enlarge the image

- Click on the 'Upload structures' button as shown below:



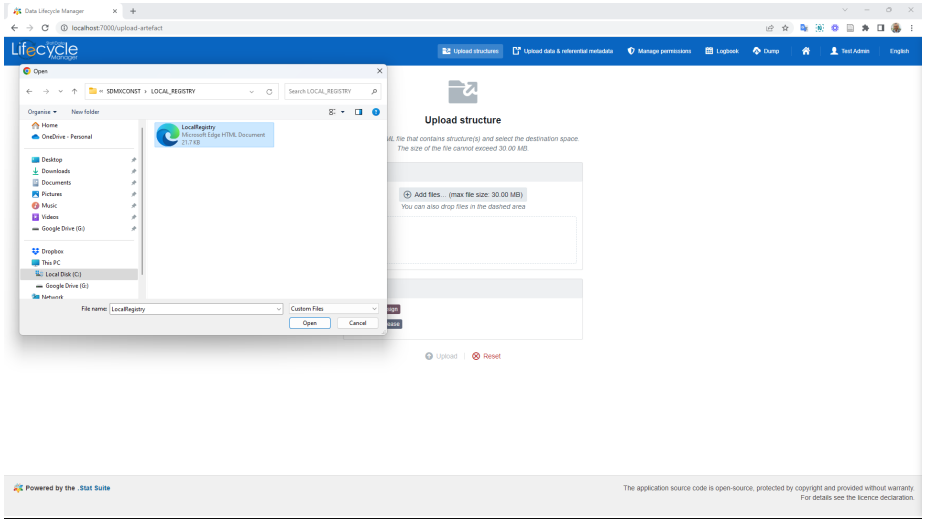
Click here to enlarge the image

- Click on 'Add files' as shown below.



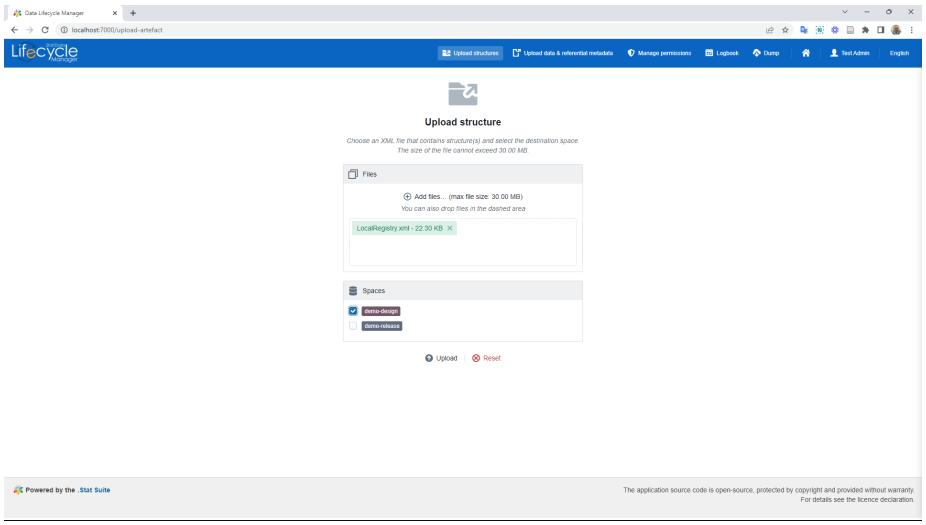
Click here to enlarge the image

- Select the XML file we created before from the folder as shown below. (Note that the LocalRegistry.xml file can be uploaded at once and may contain multiple artefacts.)



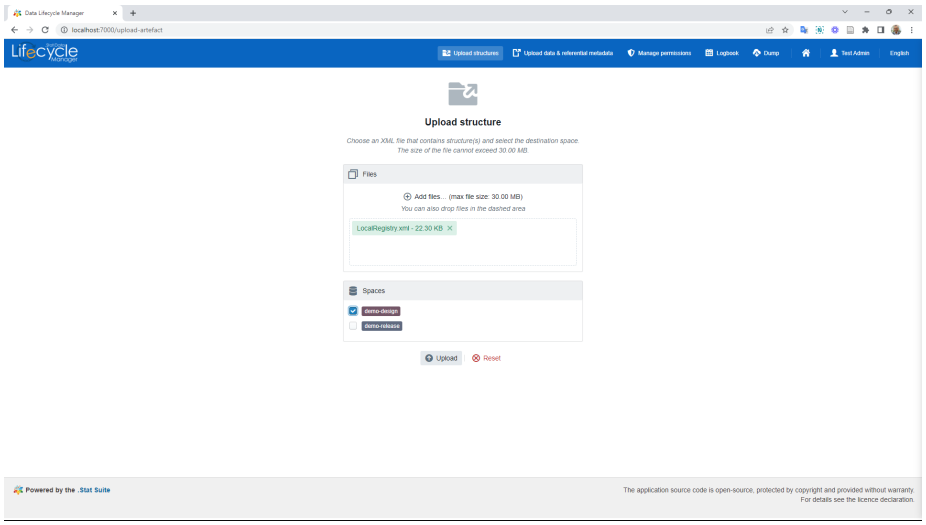
Click here to enlarge the image

- After adding the file, select the ‘demo-design’ space, as shown below.



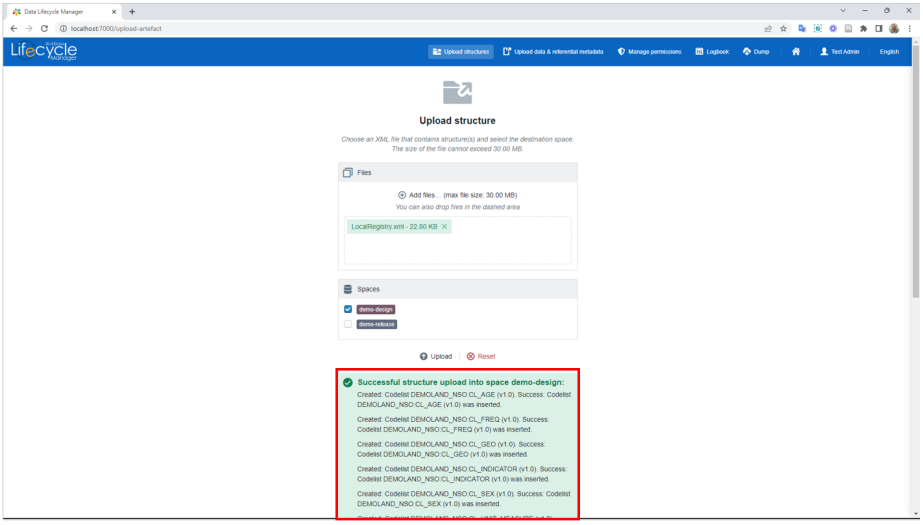
Click here to enlarge the image

- Click upload as shown below.



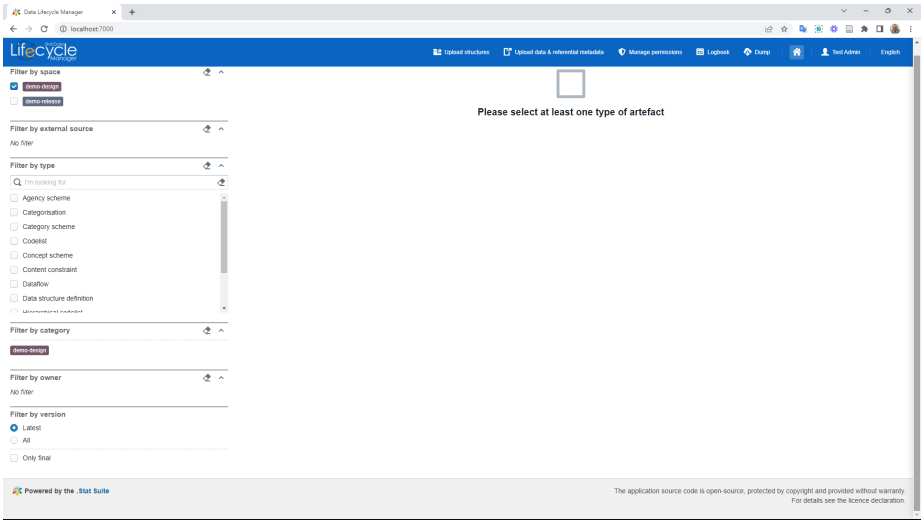
Click here to enlarge the image

- As shown below, the green background and the message will indicate the successful XML file upload.



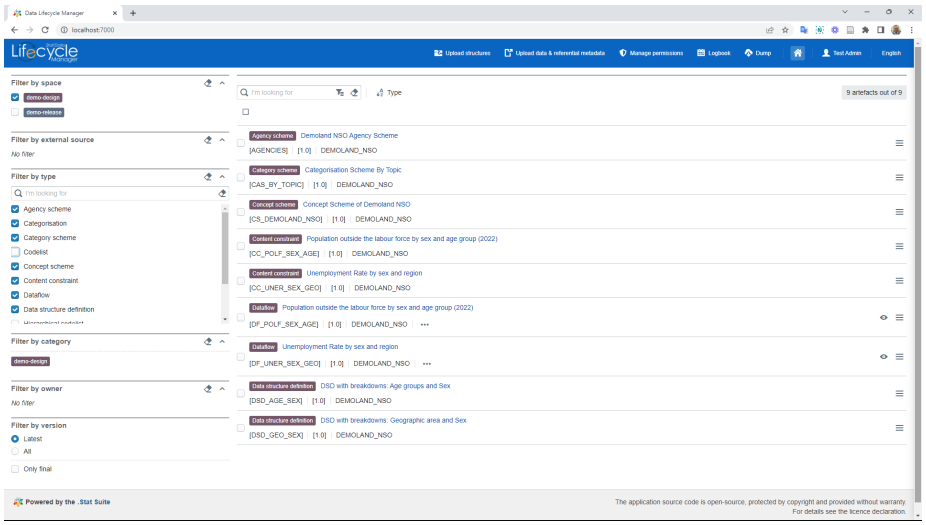
Click here to enlarge the image

- Clicking the Home icon will take you to the following interface.



Click here to enlarge the image

- Here, selecting the options on the left navigation panel will show the artefacts on the main pane, as shown below.

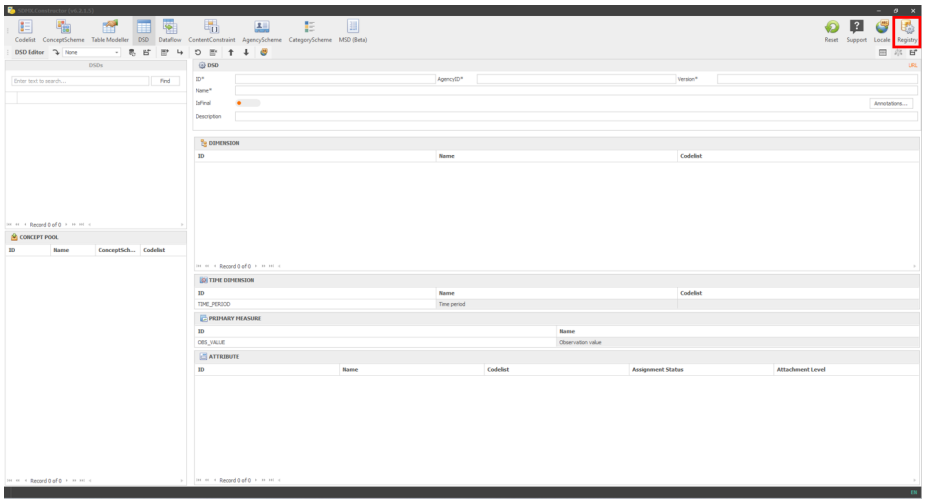


Click here to enlarge the image

4.9 Connecting to an SDMX registry

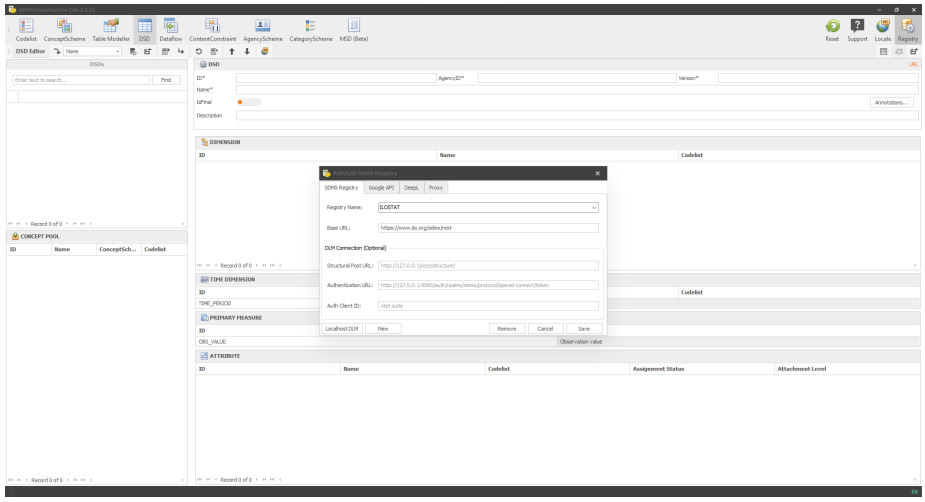
SDMX Constructor can also connect to a new SDMX registry (note: DLM is not a registry; .Stat Suite is. The DLM is composed of a set of back-office modules as one of the major components of the .Stat Suite that combines all data lifecycle management activities into one user interface.) directly, and users can directly pull or push (perhaps after editing) (with authentication credentials) artefacts from the SDMX Constructor.

- After launching the SDMX Constructor, click on the Registry, as shown below.



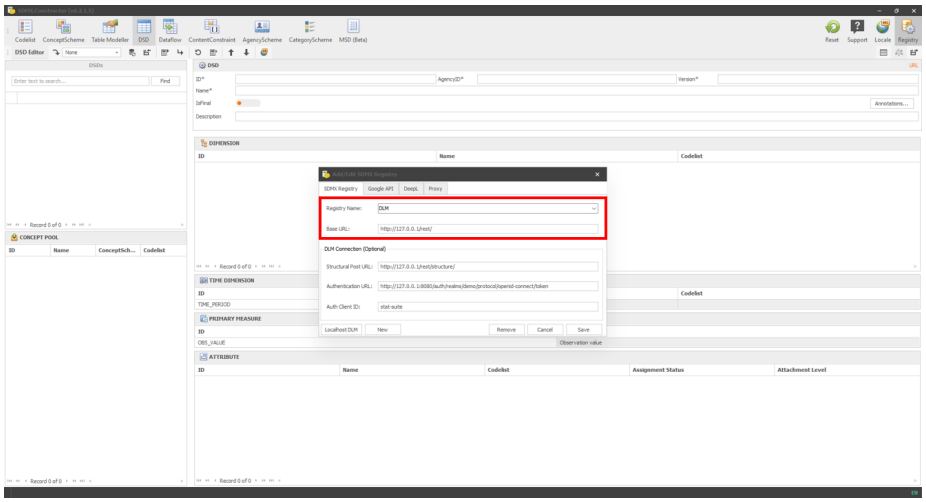
Click here to enlarge the image

- The default value you can see is as follows.



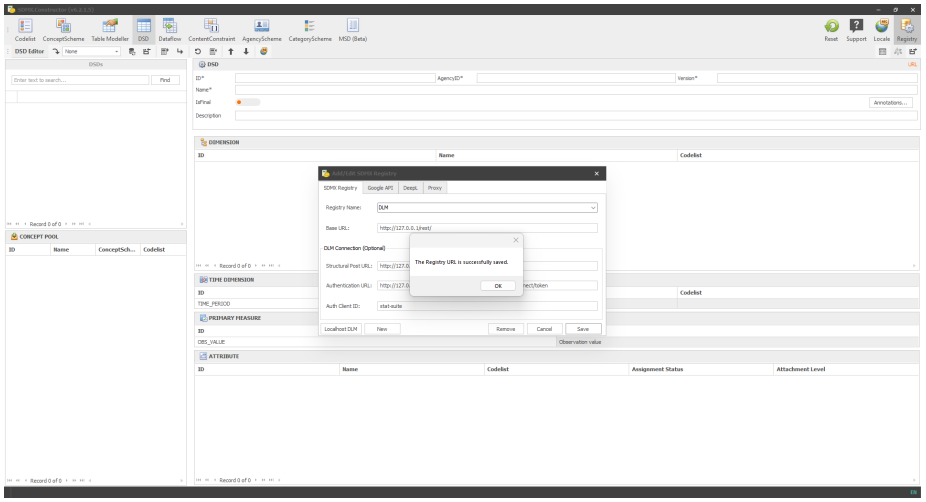
Click here to enlarge the image

- Change the default values for Registry Name and Base URL. For Registry Name, you can choose a name of your choice. For example, Registry Name could be DLM, and for the Base URL, the localhost configuration is already specified (by default, it will be http://127.0.0.1/rest/).



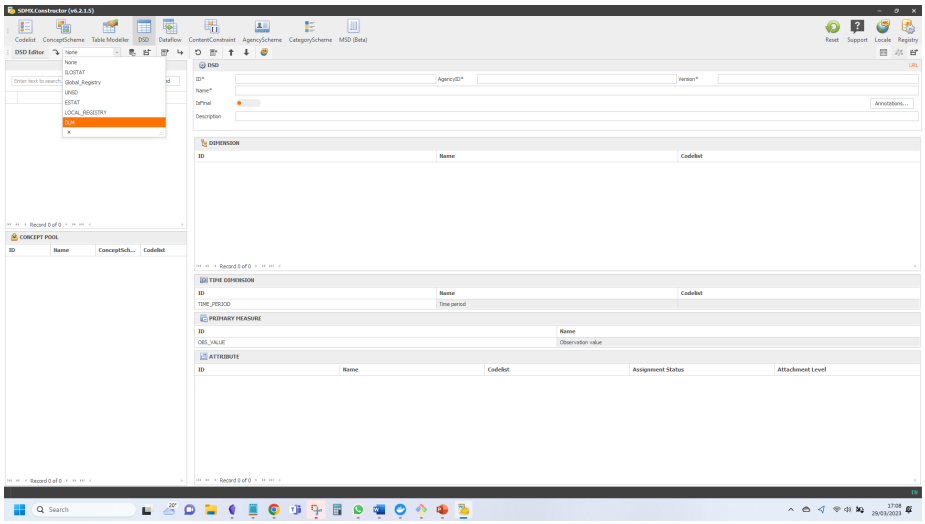
Click here to enlarge the image

- Clicking on Save will show the message shown below. Click on OK.



Click here to enlarge the image

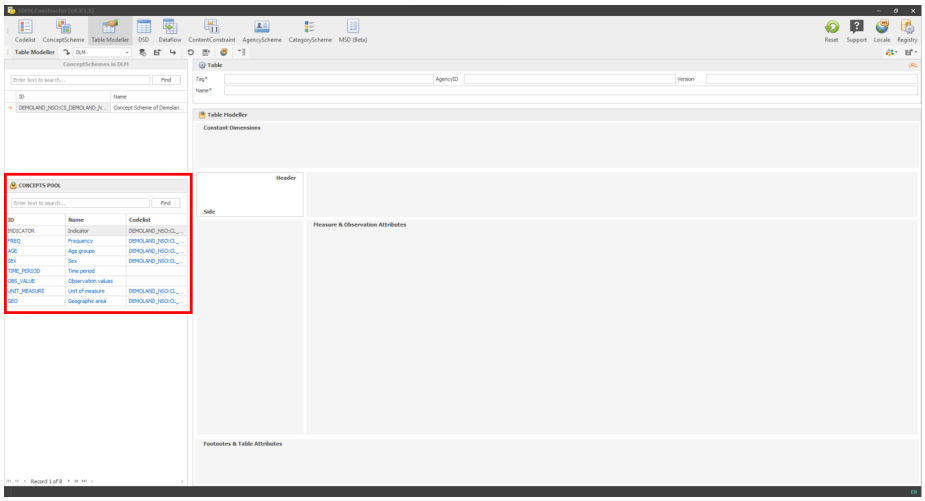
- Once the Registry is set, select it from the 'Load from registry' option, as shown below.



Click here to enlarge the image

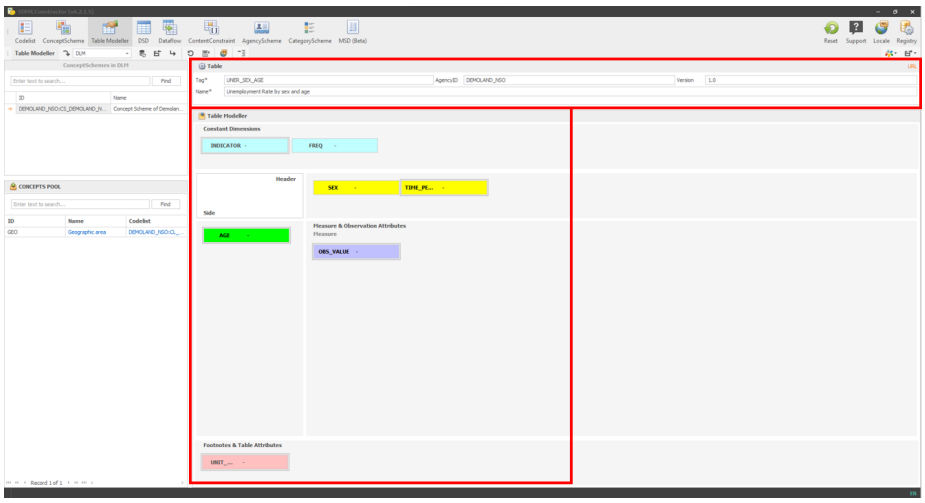
Doing this syncs the artefacts. For instance, you can see the dataflows in SDMX Constructor populated from the DLM. However, pushing any new entry (in this case, let's say, a new dataflow) to DLM would require the credentials for the DLM.

- From the concepts obtained from the DLM, as shown below, let's create a new data flow.



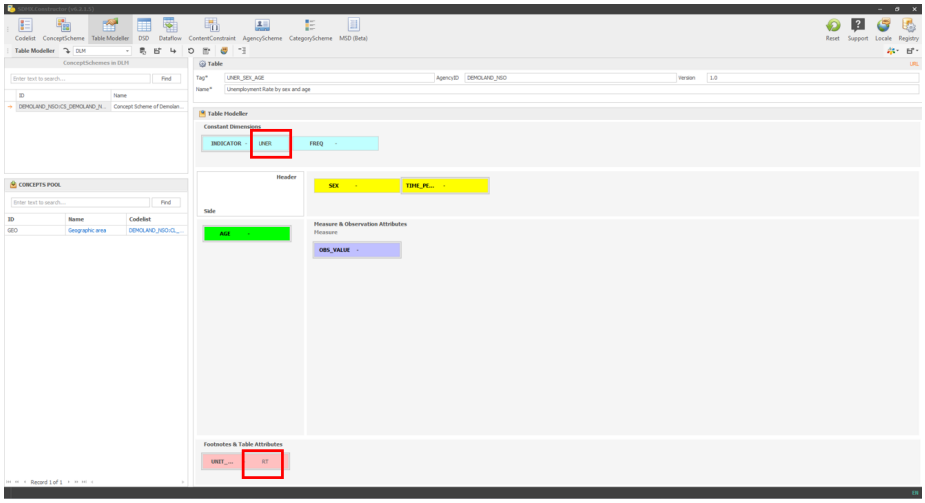
Click here to enlarge the image

- Move the concepts around and enter table details as shown below.



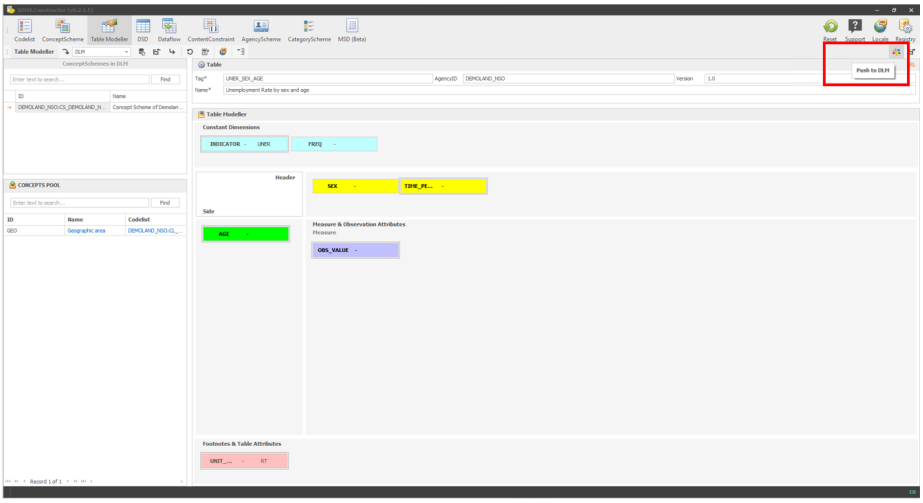
Click here to enlarge the image

- Apply the content constraints as shown below.



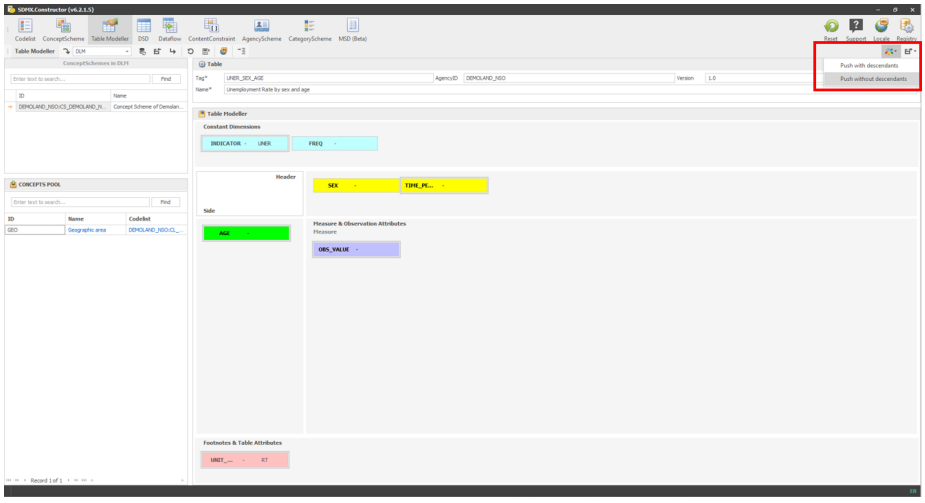
Click here to enlarge the image

- Then click on the Push to DLM button, as shown below.



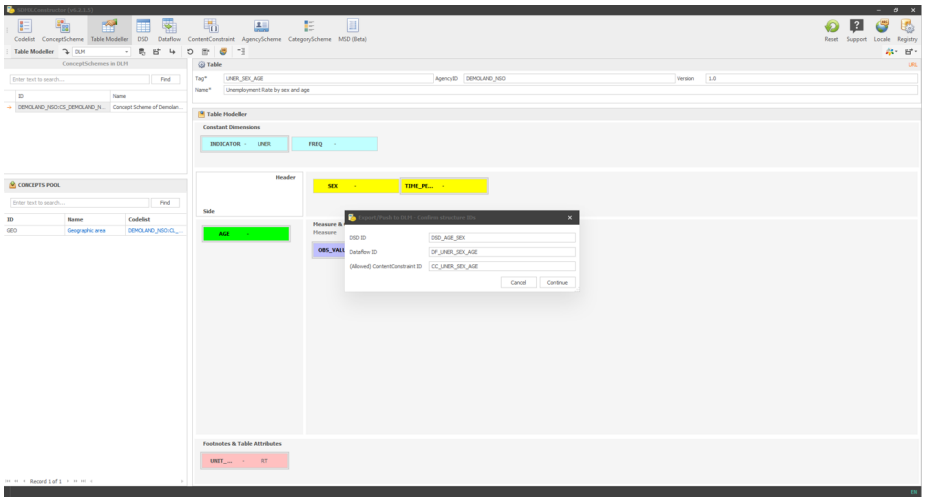
Click here to enlarge the image

- Select 'Push without descendants'.



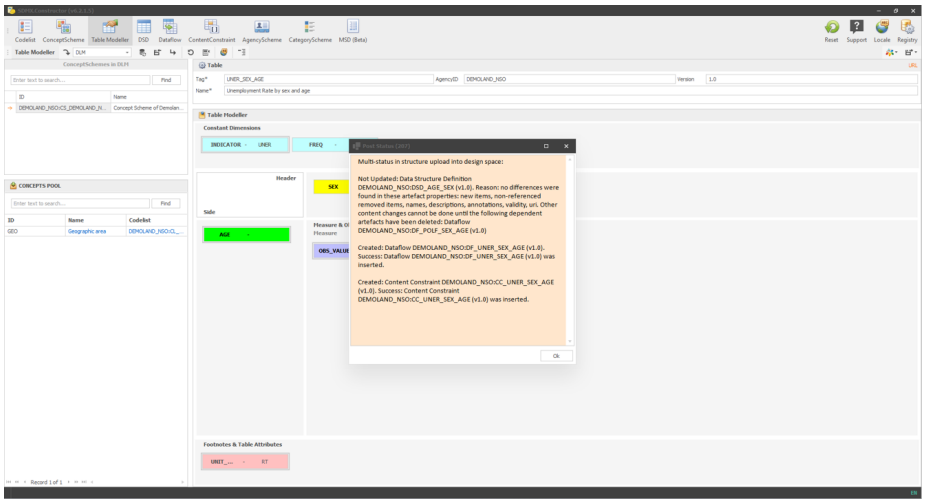
Click here to enlarge the image

- After entering the credentials for the DLM, push, and you will see a message like the one below.



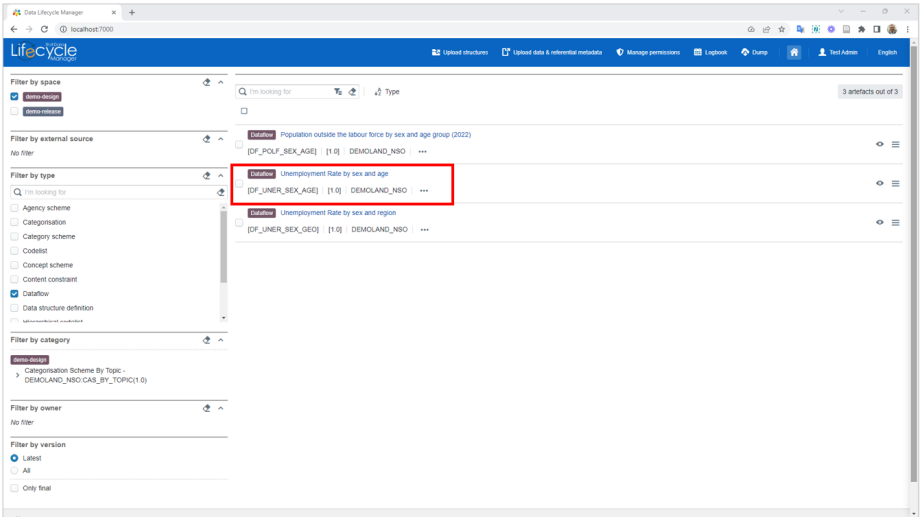
Click here to enlarge the image

- After you have pressed the Continue button, you will see the following. The message mentions what has changed and what has not. In this case, it notes no DSD change, but one data flow and a content constraint have been created.



Click here to enlarge the image

- Click on OK and go to the DLM. In the DLM, clicking on the dataflow would show the additional dataflow added directly through the SDMX Constructor to the DLM, as shown below.



Click here to enlarge the image

Chapter 5

Special Topics

Welcome to the special topics chapter of the SDMX Constructor user manual. In this chapter, we will delve into three topics that require additional explanations. These topics are Annotations, Table Modeller and Translations using Google and Deepl APIs. By the end of this chapter, you will have a better understanding of how to utilise these features within the SDMX Constructor software tool. Let's get started!

5.1 Annotations

The SDMX Constructor is a powerful tool that supports various types of SDMX artefact annotations, similar to what the .Stat Suite offers (see SDMX annotations supported by the .Stat Suite here: https://sis-cc.gitlab.io/dotstatsuite-documentation/using-de/sdmx-annotations/).

Annotations in the SDMX framework are constructs that allow the user or organisation-specific metadata to be included. These annotations can be applied to a wide range of SDMX structural metadata artefacts, offering flexibility and customisation options.

The main advantage of annotation functionality within the SDMX Constructor is its adaptability. This flexibility is highly advantageous, especially for applications such as dissemination tools where custom metadata is crucial.

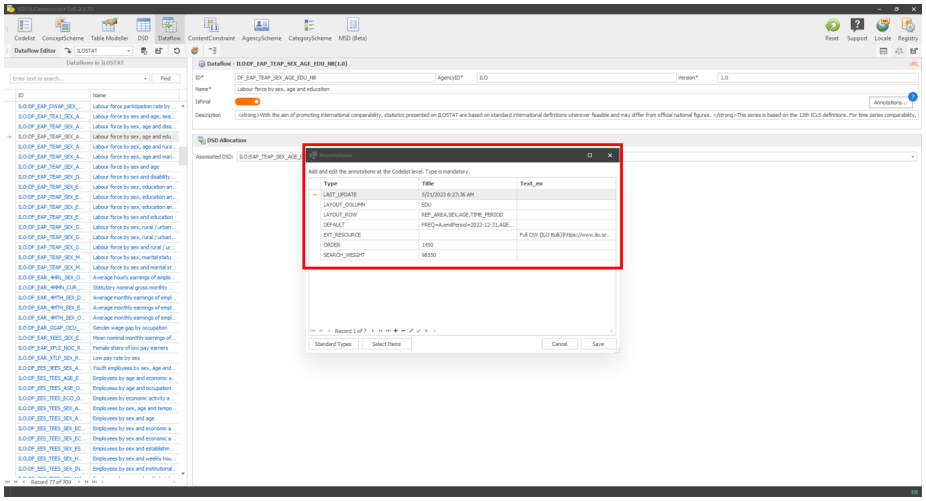
By leveraging the SDMX Constructor's support for annotations, users can enrich their data with additional information that goes beyond the standardised SDMX elements. These annotations can include context-specific details, explanations, or additional attributes relevant to the specific use case.

A walkthrough

For instance, an organisation may annotate a dataset with details about the dimensions to be presented in rows and columns of a presentation table or specify the default filters in the user interface of the dissemination interface. These annotations provide valuable context, enhance the understanding and interpretation of the data, and improve the dissemination tool's efficiency.

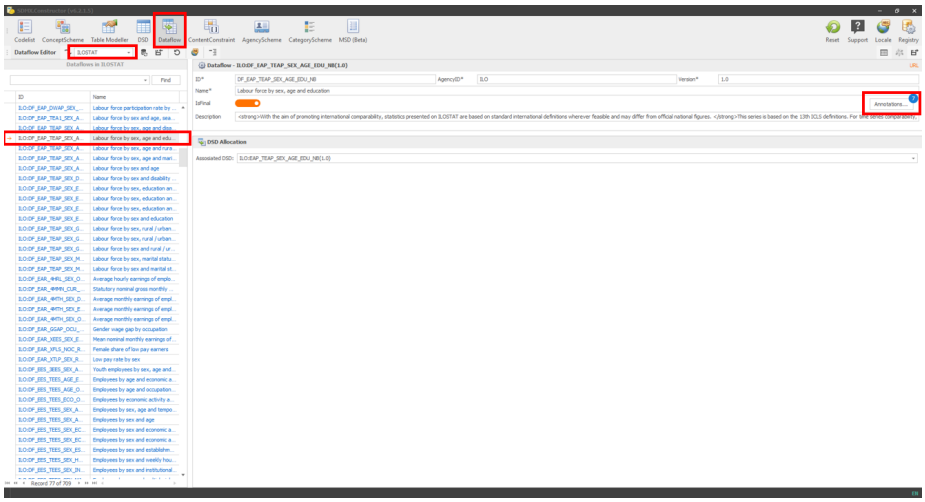
Take an example within the ILOSTAT (https://data.ilo.org/), which is built using .Stat Suite. You can choose the following indicator - Labour force by sex, age and education - here. Note that in the resulting display, there are default filters applied (for example, for 'Frequency', it is Annual; for 'Sex', it is Total; and for 'Age', it is Youth and Adults (15+). Also, in the 'Table' view, you may notice that the country names (Reference area) and years (Time period) are in the rows and education levels are in the columns of the table. Such a specific display/interface is a result of the annotation settings.

You can use the SDMX Constructor (as a back-end tool for the .Stat Suite) to specify such behaviour of loading and displaying data through its annotation functionality. In the SDMX Constructor, annotations are available through all Editor menu items such as Codelist, ConceptScheme, DSD, Dataflow, Concept-Constraint, AgencyScheme, CategoryScheme and MSD. We will see below how it works for the Dataflow as an example. It works similarly for other artefacts. As shown below, Annotations have three main components: Type, Title and Text (in multiple languages - if applicable).



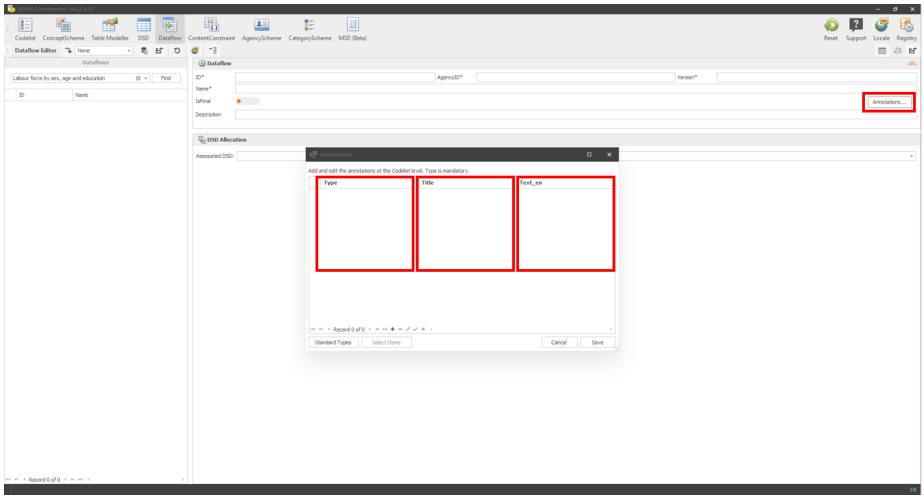
Click here to enlarge the image

By clicking on 'Standard Types' on the pop-up, one can choose from the SDMX annotations supported by the .Stat Suite.



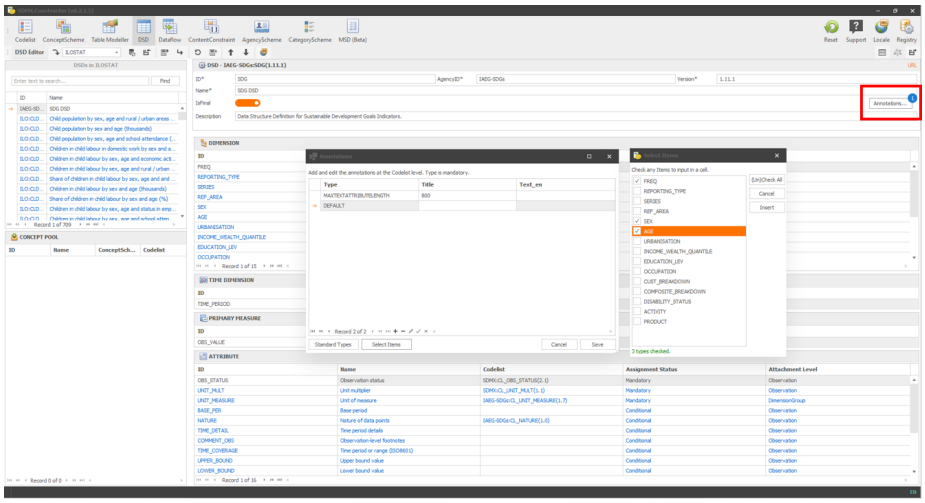
Click here to enlarge the image

Once checked/selected and added (for example, in the below image, Default, Layout_Column and Layout_Row are added), one can save the annotation by clicking the Save button.



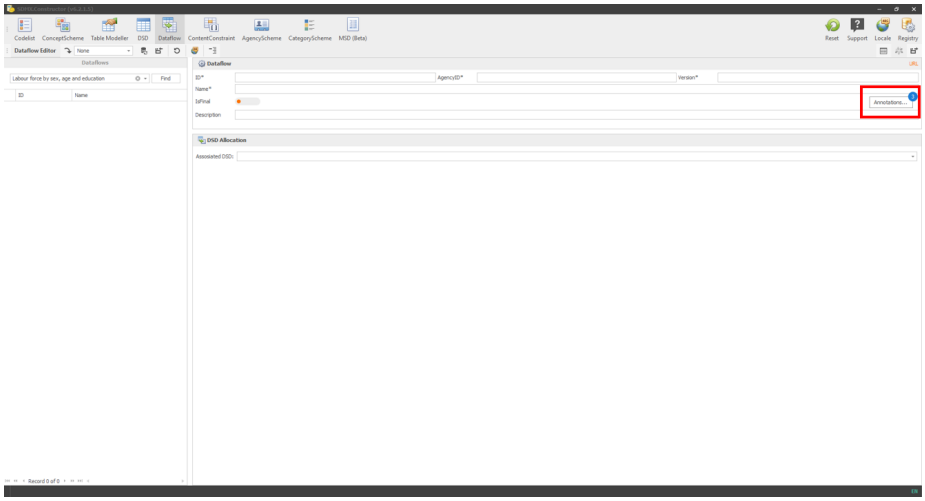
Click here to enlarge the image

Once saved, one can see the number of annotations displayed on the annotation button (as shown below).



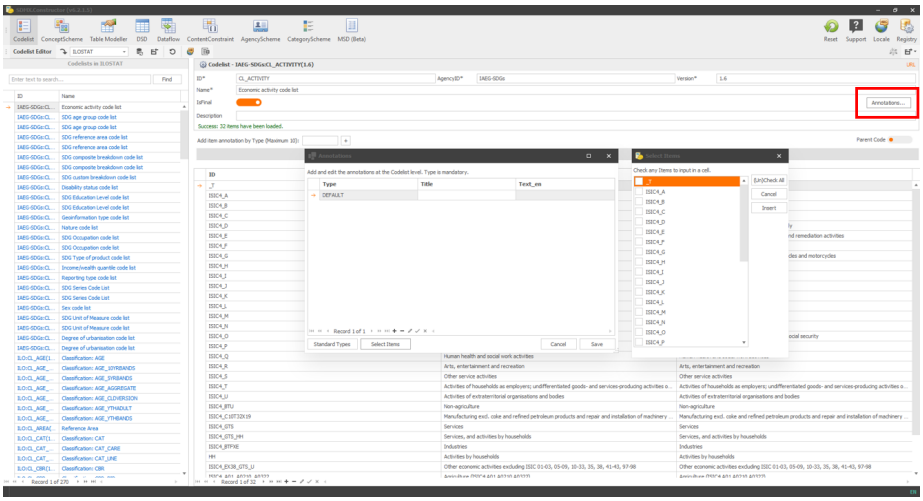
Click here to enlarge the image

If we load the ILOSTAT registry in the SDMX Constructor and select the Dataflow for the Labour force by sex, age and education, we see in-fact 7 annotations (as shown below).



Click here to enlarge the image

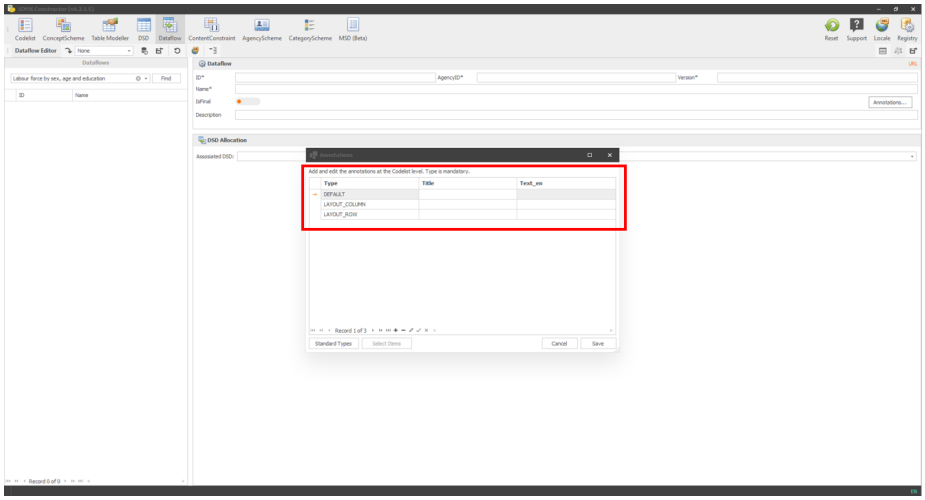
Clicking on the Annotations button will show the preselected Types, Titles and Text (in English) relevant to this Dataflow as shown below.



Click here to enlarge the image

Titles

To manually assign items in the 'Title' column of the annotation pop-up window, click on the desired row and click the 'Select Items' button. From there, you can choose from the available options, such as EDU for the row: LAYOUT_COLUMN for the 'Labour force by sex, age, and education' as shown below.

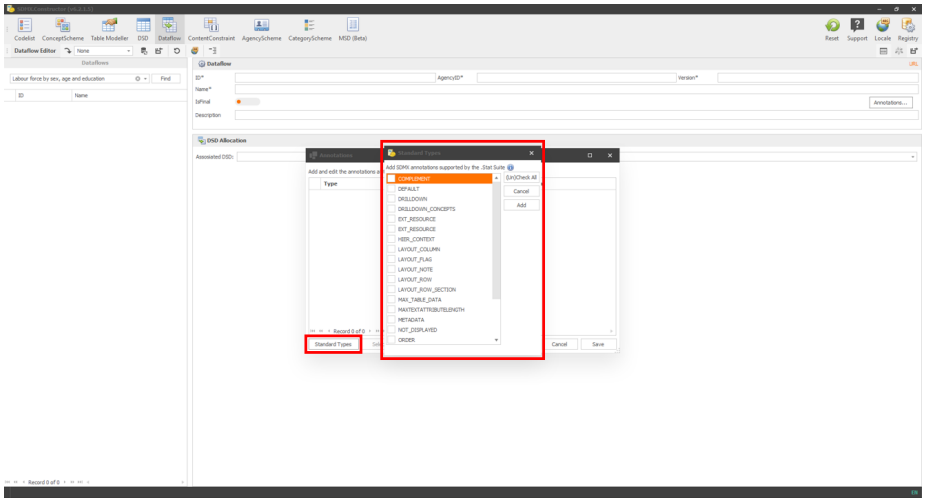


Click here to enlarge the image

Please be aware that the "Selected Items" button will become active under the circumstance: 1) When a registry is chosen, which permits the potential

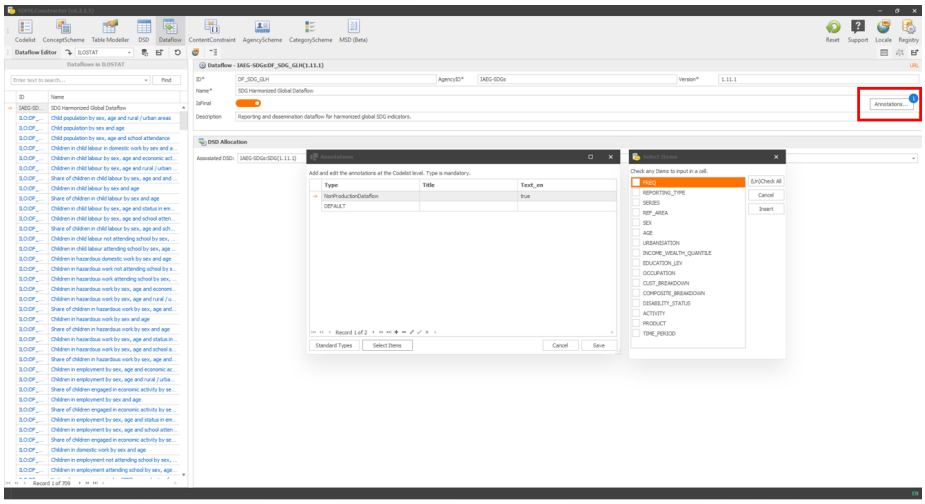
cross-referencing of artefacts (for example, selecting ILOSTAT), and 2) In the situations listed below:

1. Codelist Editor: The list of code items of the currently selected code list will be provided for selections. For example, in the image below, the first code list CL_ACTIVITY is double-clicked and then Annotations. After choosing items from 'Standard Types', one can choose Titles from the 'Select Items' button.



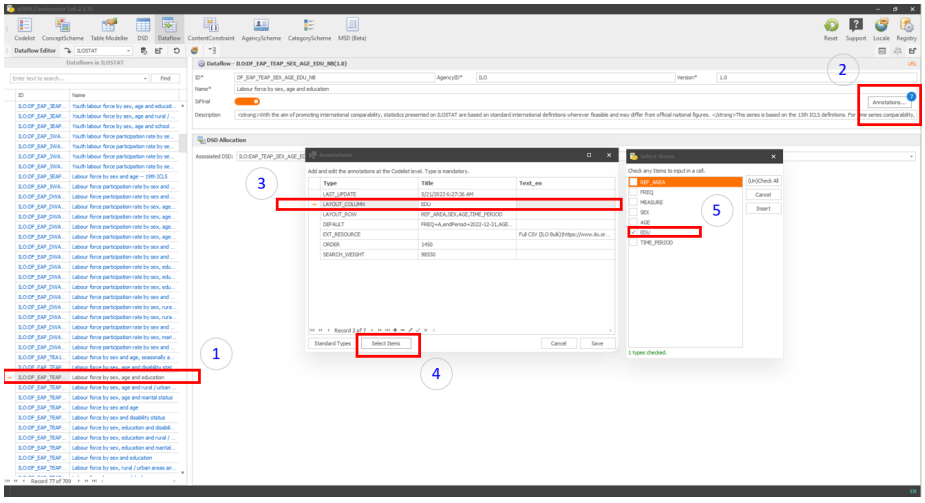
Click here to enlarge the image

2. ConceptScheme Editor: When you double-click a concept with a code list. The list of code items of the currently selected concept will be provided for selection. Clicking on the Annotations, as shown below, will offer the opportunity to select 'Standard Types' and Titles from the 'Select Items' button.



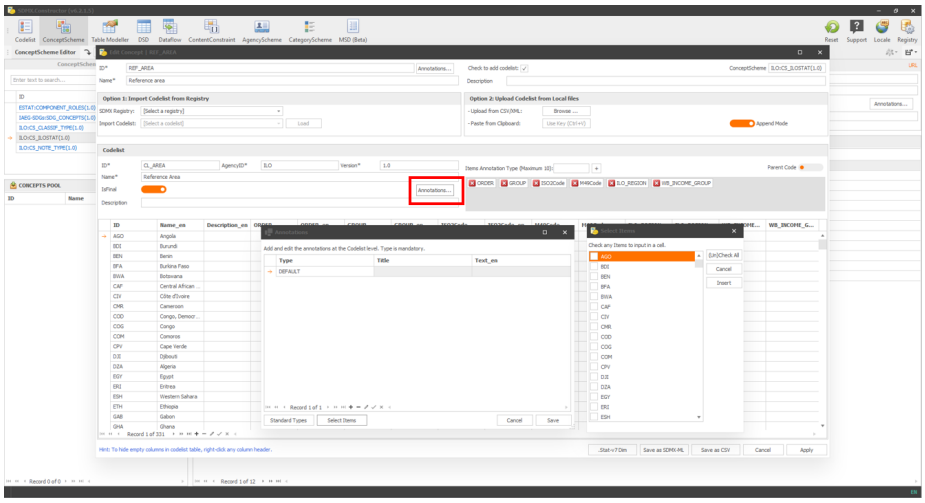
Click here to enlarge the image

3. DSD Editor: The list of Dimensions of the currently selected DSD will be available for selection. Clicking on the Annotations, as shown below, will offer the opportunity to select 'Standard Types' and Titles from the 'Select Items' button.



Click here to enlarge the image

4. Dataflow Editor: The list of dimensions of the selected dataflow (the implied DSD) will be available for selection. Clicking on the Annotations, as shown below, will offer the opportunity to select 'Standard Types' and Titles from the 'Select Items' button.



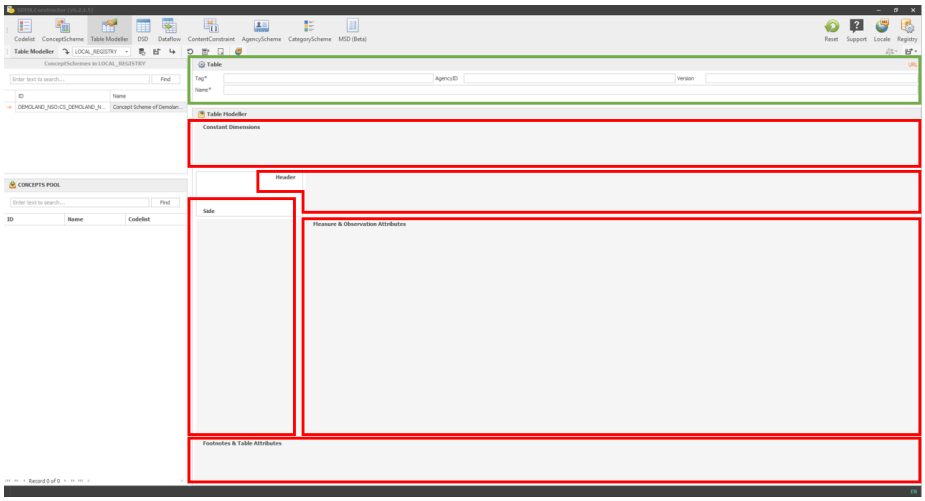
Click here to enlarge the image

Annotations supported by the SDMX Constructor serve as a powerful tool for driving the display of data. By leveraging annotations' flexibility and customisation options, users can enhance the user experience and improve data interpretation. The SDMX Constructor's annotation support is valuable for users seeking to make their data more meaningful, informative, and visually engaging.

5.2 Table Modeller

The Table Modeller functionality supports designing statistical tables using an intuitive user interface that generates the SDMX artefacts that model it. The interface is designed to use statistical terms to name objects, hiding the SDMX artefact names wherever possible, making it user-friendly and accessible.

By understanding the user interface (as shown below), the users of the Table Modeller functionality in SDMX Constructor can easily create statistical tables that generate the SDMX artefacts. Note that the table is a view of a container, which defines how the data will be stored.



Click here to enlarge the image

The placeholders provided in the Table Modeller interface help users understand and include different concepts in the table and create well-defined SDMX artefacts.

A table in SDMX parlance represents a Dataflow, has a title, and comprises concepts (SDMX CONCEPTS) that take different roles in the table. The Table Modeller interface has various placeholders for the concepts, which are:

- **Constant Dimensions:** A placeholder for SDMX CONCEPT type DIMENSIONS that remain constant across the table, such as indicator or measure and frequency.
- **Header:** A placeholder for SDMX CONCEPT type DIMENSIONS that appear at the top of the table, such as time-period.
- **Side:** A placeholder for SDMX CONCEPT type DIMENSIONS that appear on the left-hand side of the table. It can contain dimensions such as Age or Region (urban/rural).
- **Measure & Observation Attributes:** A placeholder for observation values (or primary measure) and SDMX CONCEPT type ATTRIBUTES attached at the observation level, such as DECIMALS.
- **Footnotes & Table Attributes:** A placeholder for SDMX CONCEPT type ATTRIBUTES such as 'free text' notes and 'source' attached at the dataset level.

Using the Table Modeller

- You can “drag-and-drop” concepts from the CONCEPT POOL on the left panel to the different areas/spaces/placeholders on the right panel.
- The concepts dropped in the “Side”, “Header”, and “Measure & Observation Attributes” areas will become DIMENSIONS. Note that the relative position of the concept in the area is relevant.
- Concepts dropped in “Constant Dimensions” will also be DIMENSIONS, but the user will have to select one (and only one) item from each DIMENSION’S Codelist, which will be assigned to it (as ContentConstraint).
- Concepts dropped in the “Measure & Observation Attributes” area, and the “Footnotes & Table Attributes” area will become ATTRIBUTES, the former attached at the Observation level and the latter at the Dataset level.
- The outputs of the Table Modeller are the artefacts needed to represent in SDMX the table design.

5.3 Translations using Google API/DeepL

We’re working hard on finishing this section. Stay tuned for updates.