HARMOSEARCH
the future of information services

FP7-SME-1
Project no. 262289

HARMOSEARCH

Harmonised Semantic Meta-Search in
Distributed Heterogeneous Databases

SEVENTH FRAMEWORK PROGRAMME

D6.4 V1.0
Manual for the Mapping Tool

Due date of deliverable: 2013-02-28
Actual submission date: 2013-02-28

Start date of project: 2010-12-01 Duration: 27 month

Project funded by the European Commission within the Seventh Framework Programme

Dissemination Level		
PU	Public	X
PP	Restricted to other participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

HARMOSEARCH
the future of information services

PROJECT ACRONYM: **HARMOSEARCH**

Project Title: Harmonised Semantic Meta-Search in Distributed Heterogeneous Databases

Grant Agreement: 262289

Starting date: December 2010 **Ending date:** February 2013

Deliverable Number: D6.4, Version 1.0

Title of the Deliverable: Manual for the Mapping Tool

Lead Beneficiary: Promoter

Task / WP related to the Deliverable: WP 6

Type (Internal or Restricted or Public): Public

Author(s): Albert Rainer, Thomas Motal, Inés Matres, Claudio Prandoni, Marlis Valentini, David Faveur, Beatrix Lehmann

Partner(s) Contributing: TU Wien, Promoter, SPK, Afidium, Museumsmedien

Contractual Date of Delivery to the CEC: February 28th 2013

Actual Date of Delivery to the CEC: February 28th 2013

PROJECT CO-ORDINATOR

Company name:	[X+O]
Name of representative:	Manfred Hackl
Address:	Siebensterngasse 4/22, A-1070 Vienna, Austria
Phone number:	+43-676-842755-100
Fax number:	+43-676-842755-599
E-mail:	manfred.hackl@xpluso.com
Project WEB site address:	www.harmosearch.org

TABLE OF CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

The document at hands represents the manual of the HarmoSearch Mapping Tool. The manual serves as reference and is written for domain experts that have little or no experience in creating a mapping. The tools main functionality is thereby explained by means of a concrete business scenario which will be used throughout the manual.

1.2 DEFINITION OF TERMS AND ABBREVIATIONS

In the following terms and abbreviations use throughout the whole document are listed and shortly described:

- **Domain-expert:** An experienced professional having a deep understanding about a certain problem domain. Usually domain-experts have only little or no technical expertise in creating a proper mapping. Domain-experts are the focus group the HarmoSearch mapping tool has been developed for.
- **XML:** The Extensive Markup Language (XML) is a markup language standardized by the W3C (World Wide Web Consortium). Best known for its ability to define machine-readable documents it is widely used in the software domain.
- **Mapping resource:** A mapping resource denotes to files that are necessary for the creation of a mapping. Typical mapping resources are: XML schemas, mapping filters, mapping patterns, and mapping files.
- **XML schemas:** A formal and structured description of terms and constraints that represent a specific data schema, e.g., Harmonise ontology.
- **Mapping filter:** A mapping filter is used to limit the number of concepts of a schema or ontology that are presented to the user to a pre-defined subset.
- **Mapping pattern:** Mapping patterns are reusable concepts that encapsulate particular rules on how a given concept has to be mapped on a syntactical level. For instance date-time conversions from one date-time format to another format and vice-versa. Mapping patterns as used by the HarmoSearch mapping tool are implemented XSLT templates.
- **Input files:** Input files are used during the transformation process and are typically XML files that contain instance data that follow the structure of a particular XML schema.
- **Instance data:** Concrete information constrained to the requirements of a given data schema. Most common are XML-based instance data files. These files are used to transfer information from a data source to a data sink in a machine-readable form (usually XML-based).
- **Mapping file:** A mapping file formalizes the mapping between corresponding elements. A mapping links elements of a source schema or ontology to another element in a target schema. Elements are usually distinguished in simple and complex elements. Simple elements are for example name fields whereas complex elements consist of multiple sub-elements, e.g., a person has a first and a last name.
- **Mapping:** A mapping specifies a relationship between a source and a target element. These relationships may be simple or complex. The former denotes

to single elements that can be mapped one-to-one to another single element. The latter addresses elements that are hierarchically ordered. A typical example would be a person's name, which can contain of a first name and a last name.

- **Data mapping:** A data mapping links elements in a source schema to elements in a target schema.
- **Value mapping:** Similar to a data mapping a value mapping interlinks two elements from different data sources. In case of a value mapping these data sources are lists of fixed values, so called reference lists.
- **Reference lists:** A reference list is a collection of data that represents dedicated domain values, e.g., types of exhibitions. In many cases such information is stored within the local database and identified by means of a unique id. However, this id is usually unique and thus needs to be mapped to the proper value in the target list, e.g., id 123 used for art of the 19's century has to be mapped to 'events after 1800' in the Harmonise ontology.
- **Eclipse IDE:** Eclipse is a free and popular development environment. It is shipped with a flexible plug-in mechanism that allows developers to build their own domain-specific applications.
- **Workspace:** An Eclipse-related term that refers to a location on the local hard disk which is used to store different types of resources. Resources in this context are: projects, folders, or files.
- **XSLT:** XSLT (Extensible Stylesheet Language) is a popular XML-based transformation language. It is widely used to transform XML documents to another XML-based document, e.g., XML to HTML transformations. XSLT transformations are specified declaratively using a template mechanism allowing an element-wise code transformation.
- **Mapping project:** A mapping project serves as central repository. A mapping project may contain an arbitrary number of files and folders. Per default a mapping project has 6 folders: Filter, Input, Mapping, Output, Pattern, and Schema. Each folder contains specific types of mapping resources. For example the Schema folder contains XML schema files whereas the Pattern folder contains XSLT-based mapping patterns.
- **Mapping stub:** A mapping stub is a pre-configured mapping project already including relevant mapping resources narrowed to the specific needs of a particular mapping scenario. For example a mapping stub dedicated to realize mappings from an arbitrary source schema to the Harmonise ontology. Such a project would already contain the necessary mapping files, a mapping filter constraining the elements to relevant sub-sets, as well as an initialized mapping file where the Harmonise schema has already been set as a target schema. Creating and distributing a mapping stub is of special interest for data integrators who want to leverage their services.
- **Update site:** An update site is a remote repository that stores and manages software used or ready to be installed for an Eclipse application. Using an update site allows Eclipse-based applications to update or install new software from the repository under consideration, e.g., updating to a newer version or extending the basic functionality by a third party plug-in.
- **Mapping Tool:** The mapping tool is a standalone application that supports a user with little technical knowledge in creating visually the necessary mapping definitions from the data model of a Harmonise participant to the

one of Harmonise and vice-versa. It consists of a graphical User Interface to show and manipulate mappings, a pluggable set of algorithms to support automatic mappings, a generator to create mapping artifacts, and an infrastructure in order to manage a mapping project.

- **Harmonise:** Name of the existing technological solution. The current version is Harmonise 2.0, which includes the Harmonise Ontology, Harmonise Service Centre and the Harmonise Portal.
- **Matching:** The process of finding correspondences between elements of different schemata or ontologies.
- **Mapping Variable:** A mapping variable is usually used for specifying parameters of often repetitive mapping patterns, e.g., mapping of different languages.
- **SPARQL:** An open and standardized format for specifying queries operating on RDF data. It is standardized by the W3C and has been released as an official W3C recommendation. A SPARQL query can contain triple patterns, conjunctions, disjunctions, and optional patterns. SPARQL is the de-facto query language standard for databases able to retrieve and manipulate data stored in the RDF format.

1.3 RELATIONSHIP WITH OTHER DOCUMENTS

Initial inputs for this document are deliverable D2.1 Use Case Specification and D2.2 Architectural Design. D2.1 defines the main functionalities and requirements whereas D2.2 is used to describe the basic architecture of the HarmoSearch system and the Harmonise system. Additional input has been gained from D6.1 First running prototype with main functionalities integrated. D6.1 defines the basic functionalities of the HarmoSearch Mapping Tool as well as concepts and techniques used for the implementation of the tool.

1.4 STRUCTURE OF THE DOCUMENT

The document is structured as follows:

- Section 2 provides an overview about the HarmoSearch Mapping Tool and its main components.
- Section 3 describes how the HarmoSearch Mapping Tool can be downloaded and installed to a Windows operating system. Furthermore, the section provides an overview about the basic technical requirements that need to be fulfilled in order to install and run the HarmoSearch Mapping Tool.
- Section 4 is the most relevant section for novice users who have no or less knowledge with regards to the definition of certain data mappings. Thereby, the section introduces the main principles and concepts as implemented and supported by the HarmoSearch mapping tool based on a concrete business scenario taken from the domain of cultural heritage. Following the tutorial the user gains insight about the basic mapping procedure and how a data mapping is created and adapted using the HarmoSearch Mapping Tool.
- Section 5 complements Section 4 and introduces additional concepts and building blocks. For example how an existing mapping solution can be exported and shared among different stakeholders. Furthermore, Section 5

addresses the highly complex mapping of reference lists (value mapping) explaining the common procedure on how to define a value mapping and how it is combined with an existing data mapping.

- Section 6 represents the tool reference mainly focused on the user interface explaining the content of the main menu and toolbar as well as other GUI (graphical user interface) in particular wizards, views, and editors.
- Section 7 provides details on the technical requirements, e.g., supported operating systems, required Java version.

2 OVERVIEW

HarmonSearch uses an Interlingua approach, that is, the data format of data sources (Sender) and data sinks (Receiver) are mapped to an intermediate data format. The concepts for this intermediate format are provided by the Harmonise ontology. This approach can improve understanding of common and shared concepts and thus reduce the risk of misunderstandings between data sources and data sinks in a data integration project. Instance data sent by or retrieved from the data source is first translated to this intermediary format and, in a second step, is translated to the format of the data sink.

The HarmoSearch mapping tool has been implemented as a stand-alone application. Its aim is to provide a simple and intuitive tool for creating and managing complex mapping solutions.

Figure 1 shows the main components of the HarmoSearch Mapping Tool's user interface:

1. Main Menu
2. Toolbar
3. Mapping Editor
4. Views

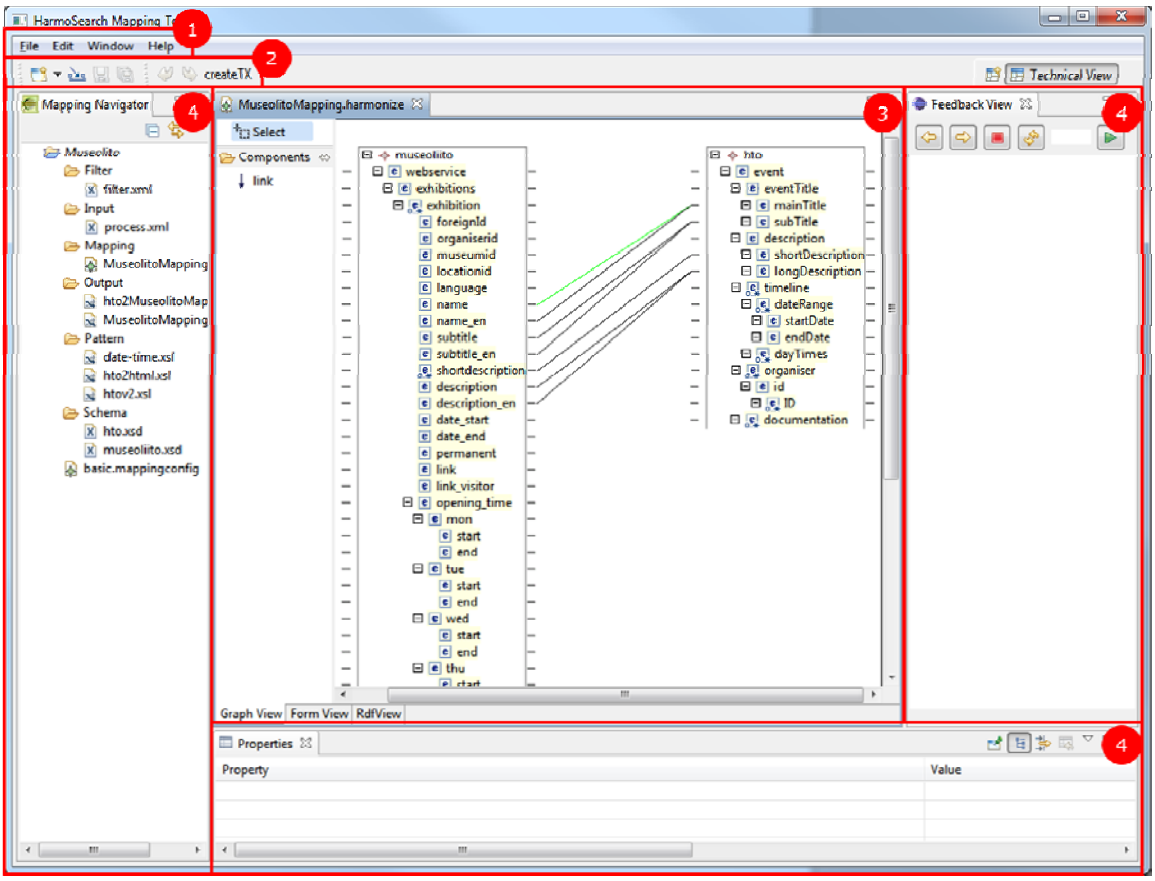


Figure 1: User interface of the HarmoSearch Mapping Tool

Main Menu

The *main menu* is located at the top of the HarmoSearch mapping tool and provides basic actions the user is able to perform, e.g., create a new mapping project, open the integrated help index, basic file operations such as copy and paste.

Toolbar

The *toolbar* is placed below the *main menu* and (similar to the main menu) provides access to basic activities a user has to perform during the mapping process. In contrast to the main menu, the toolbar is constrained to the main tasks, e.g., import a resource file, save, undo or redo actions.

Mapping Editor

The *mapping editor* is placed in the editor pane (center of the user interface) supporting the user by creating a mapping specification graphically. The mapping editor is thereby, separated into a graph-based and a form-based editor. Both complement each other and can be used together or separately for creating valid mapping specifications.

Views

Views are used to provide alternative presentations as well as ways to navigate the information, e.g., organizing the mapping resources.

The user interface of the HarmoSearch Mapping Tool is composed of the following main views:

- The **Mapping Navigator** view is used to visualize and manipulate existing mapping projects. Thereby it provides a contextual menu allowing the user to manipulate projects and resources, e.g., rename, delete, import, and create.
- The purpose of the **Properties view** is to display properties and events of a selected resource and provides editable text fields, lists and other controls to allowing editing properties and events.
- The **Feedback view** is used to display the result of a mapping transformation.

3 INSTALLING THE HARMOSEARCH MAPPING TOOL

This section describes how the HarmoSearch mapping tool can be installed. Installing the HarmoSearch mapping tool requires three basic steps:

1. You have a Windows operating system (x86 or x64)
2. Ensure that the Java Runtime Environment (JRE) version 1.6 or later is installed on your system. The JRE can be downloaded from the Oracle website under http://www.java.com/en/download/manual.jsp
3. Download the most recent version of the HarmoSearch mapping tool from the HarmoSearch portal. The download is available for registered users under http://harmosearchdev.harmonet.org/mappingstore

The remainder of this chapter explains how to accomplish the installation of the HarmoSearch mapping project under Windows 7 (64-bit version):

- Check the installed JRE version
- Download the mapping tool
- Install the mapping tool
- Start the mapping tool

3.1 CHECK THE INSTALLED JRE VERSION

To check if the JRE is properly installed in Windows 7 click on the Windows Start button, type “cmd” into the text box and hit enter (cf. Figure 2). Within the upcoming command box type the following command: “Java –version”. Check the version number and make sure it is something like 1.6.x (cf. Figure 3).

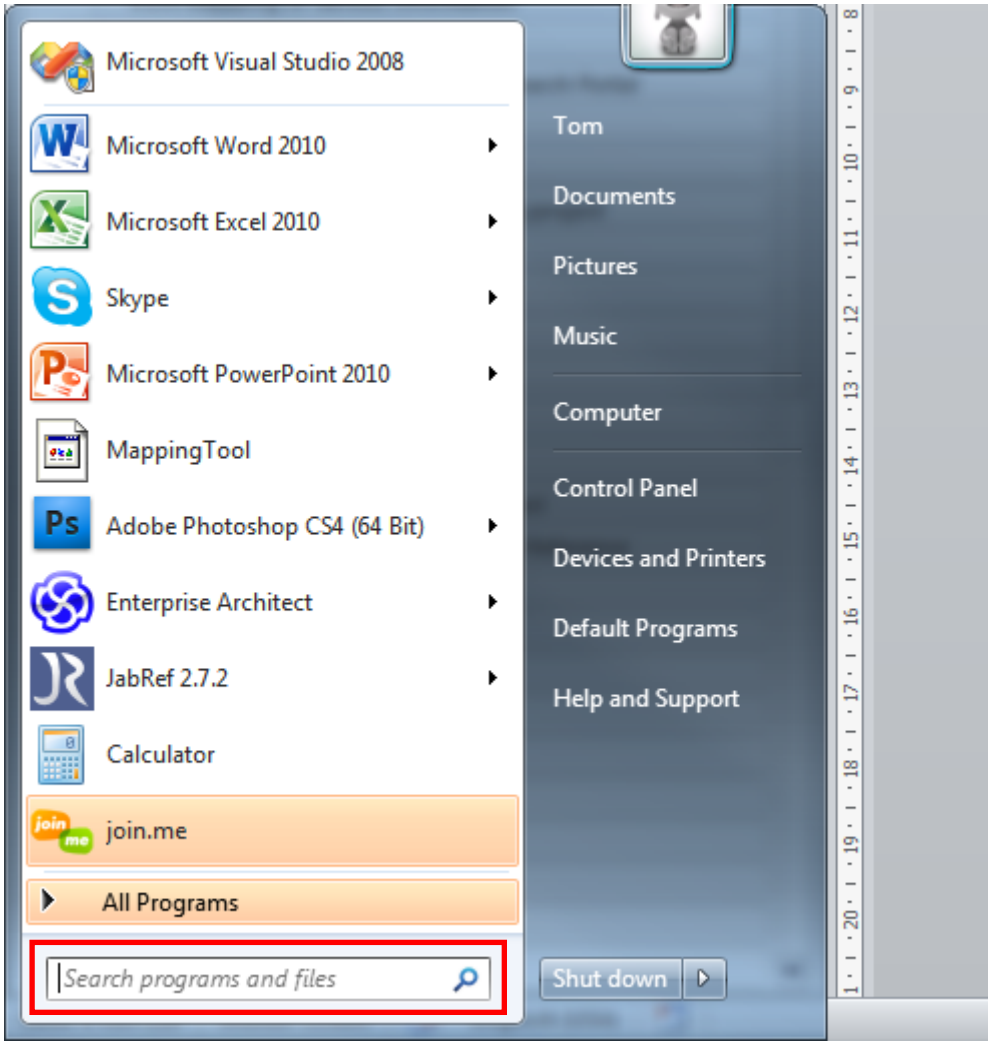


Figure 2: Using the search field for opening a new command window in Microsoft Windows 7

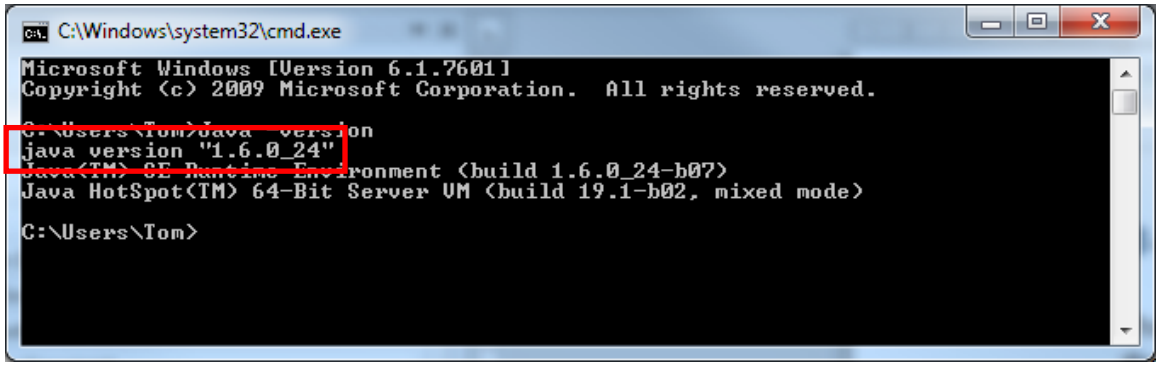


Figure 3: Checking the installed JRE version

If the test fails you have to download and install the most recent version of the JRE. The JRE is provided by Oracle and can be downloaded for free using the following link: http://www.java.com/en/download/manual.jsp

3.2 DOWNLOAD THE MAPPING TOOL

The most recent version of the mapping tool can be downloaded from the HarmoSearch portal. In order to download the mapping tool do the following:

- Open your Web-browser and surf to http://harmosearchdev.harmonet.org/mappingstore
- Login with your credentials (username and password)
- If you are not automatically redirected to the Mapping Store, click on Configurations in the main menu and click the Mapping Store sub-menu
- Download the proper installer depending on your operation system, i.e., 32-bit or 64-bit:
  - MappingToolSetup_x86_*.exe
  - MappingToolSetup_x64_*.exe
- Optionally download the resources for the basic tutorial in Section 4

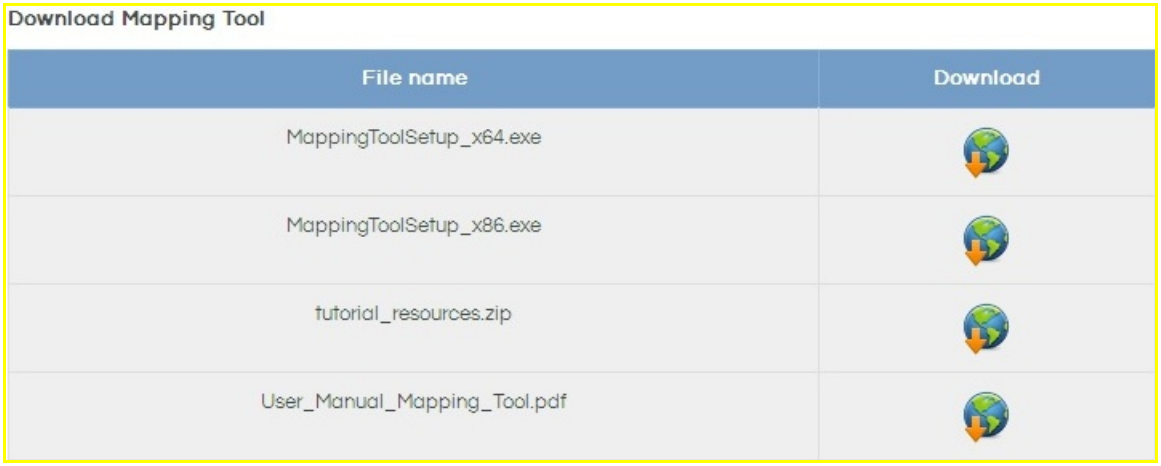


Figure 4: Download Mapping Tool from HarmoSearch portal

To check if your system runs in 32-bit or 64-bit mode in Microsoft Windows 7 open the Windows start menu by clicking the Start button. Locate Computer (cf. Figure 5) on the right hand side of the Windows start menu and select Properties from the popup window (right-click on Computer) to open the System window. The System window shows basic information about your computer. Locate System type in the window to check the operational type of your system. In case of a 64-bit operating system (as shown in Figure 6) you will have to download the 64-bit version of the installer (MappingToolSetup_x64_*.exe) otherwise the 32-bit version of the installer (MappingToolSetup_x86_*.exe).

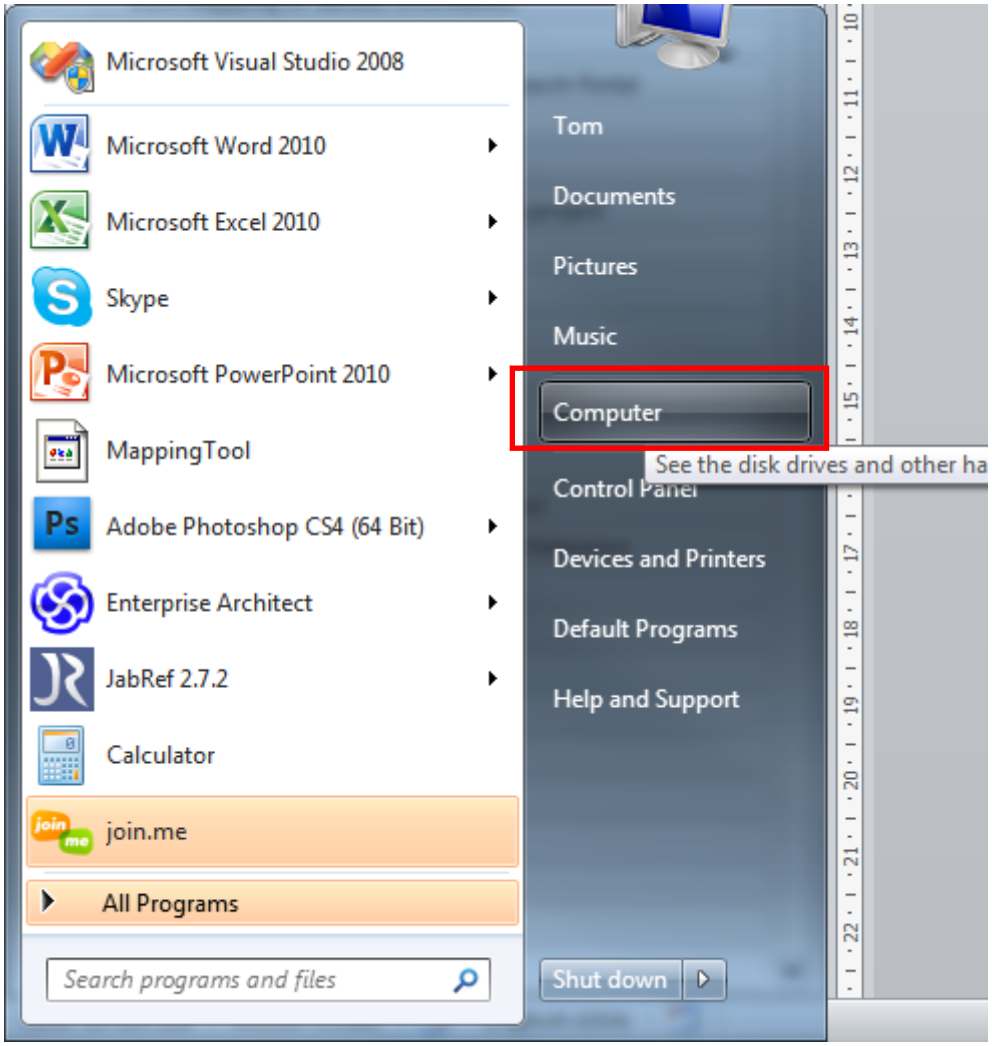


Figure 5: Opening the system properties in Windows 7

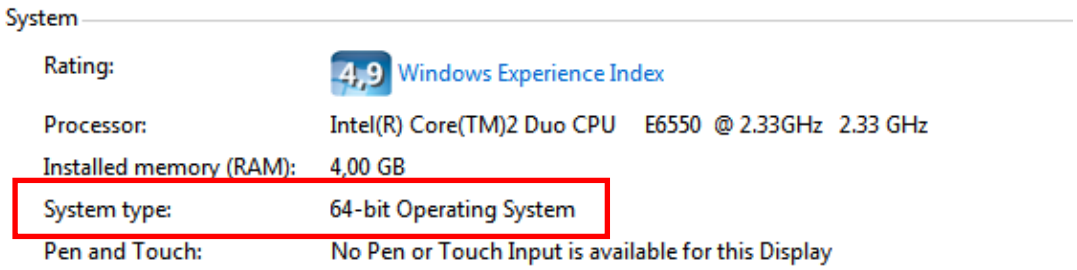


Figure 6: Finding the system type property in Windows 7

After the download is completed double-click the installer and run the setup. Follow the installer's instructions and agree the license agreements, choose a proper install location of your choice and wait until the installation process has been finished. You will find a new folder in the location you provided during the installation procedure and in addition a desktop shortcut pointing to the executable of the HarmoSearch mapping tool

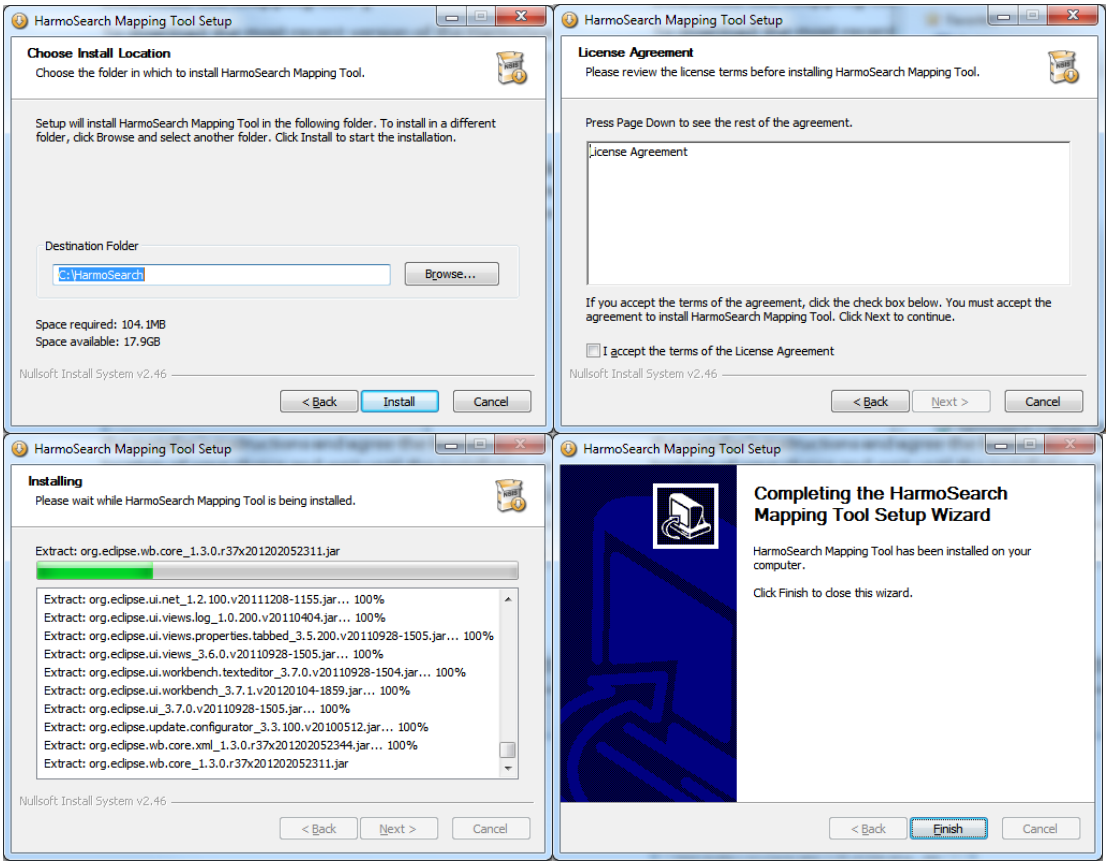


Figure 7: Individual installation steps

3.3 START THE MAPPING TOOL

To start the HarmoSearch mapping tool simply double-click the desktop-shortcut labeled Mapping Tool.

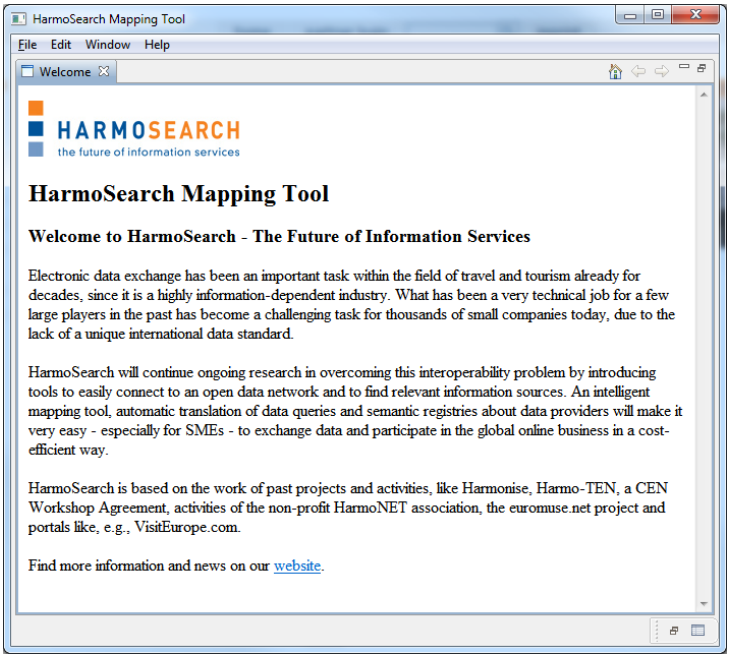


Figure 8: The Welcome screen of the HarmoSearch Mapping Tool

4 BASIC TUTORIAL

This section will introduce the main steps necessary for creating, specifying and testing a proper mapping using the HarmoSearch mapping tool. Thereby, a concrete business scenario serves as a reference covering the complexity of the underlying (suggested) mapping procedure. The workflow a user has to undergo can be described by four certain steps:

1. **Create a new mapping project:** In a first step the user has to create a new mapping project. A mapping project serves as a central repository which contains all resources that are necessary in order to create a proper mapping. Resources in this context denote to certain types of files, such as schema files, mapping pattern files, mapping files, mapping filter files, and input data files. More information concerning a mapping project and the different types of mapping resources can be found in Section 1.2.
2. **Setting up the mapping project:** In the second step the necessary resources that support the development of a proper mapping definition must be added to the newly created mapping project. Typically the user has to add at least two schema files representing structural information about the source and the target data, one or more mapping patterns, which will be used to define the actual mappings, a transformation configuration file dealing with the basic settings that drive the transformation procedure, and a mapping filter, which will guide the user towards a proper mapping definition.
3. **Setting up the mapping file:** Before a mapping can be created it will be necessary to specify the source respectively the target schema. This is done directly in the mapping file using the mapping editor. After initializing these basic requirements the user is able to define the mapping definition.
4. **Creating a proper mapping definition:** After creating and setting up the mapping project and the mapping file the actual mapping can be created. The mapping is done using the HarmoSearch mapping editor graphically supporting the user in creating proper mapping links. Simplified a mapping link represent a connection between a source and a target element. The relationship also defines the mapping pattern that will be used for creating proper mappings during the transformation procedure.
5. **Running the transformation:** Running the transformation procedure is actually the last and final step of the usual workflow. The transformation is thereby fully automated creating certain output files according to the process specification. Note, however, the output is restricted to XML-based files, such as XML or HTML. The main advantage herby is that the final outcome can be visualized and thus, provides immediate feedback to the user.

Putting the described workflow in context a real business scenario from the cultural heritage domain will be applied. The scenario deals with the complex task of mapping event data from a given database schema to the Harmonise ontology. Events in this context represent distinct information about exhibitions. The museum under consideration is thereby, the finish museum Museoliitto.

Overall aim of the tutorial:

- Leverage a common understanding about the main functionalities and principles of the HarmoSearch mapping tool
- Apply the HarmoSearch mapping process to a concrete business scenario in the cultural heritage domain.

The tutorial makes use of the following components:

Below the main components relevant for the tutorial are listed below. Sections providing more details are written in parentheses next to the component.

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Feedback View (see Section 6.3.2)
- Mapping Navigator View (see Section 6.3.1)
- Menus and toolbars (see Section 6.1 and Section 6.2.1)
- Installation and import wizards (see Section 6.5)

Files used in the tutorial:

Realizing the business scenario at hand requires a couple of different resources. These resources are listed in the table below. Thereby, each resource file is described by its name, purpose and type.

File Name	Purpose	Type
filter.xml	The filter file used to structure the mapping process into certain steps, e.g., overall information about an event. The elements specified by the filter can be seen as requirements or recommendations for creating proper mappings for a specific domain, e.g., mapping of events in the cultural heritage domain.	Filter
museoliitto.xml, process.xml	Input data that will be used during the transformation procedure. The process specification (e.g., process.xml) defines several parameters of the transformation process, e.g., input and output. The Museoliitto.xml file represents certain input data that can be mapped using the newly created data mapping. Consequently, the mapping can be evaluated by the	Input

	user using real business data.	
data-time.xsl, Euromuse_RL_langauges.xslt, eurmouse3html.xsl, HTO_Euromuse.xslt, htov2-1.xsl, htov2-1-extension.xsl, pure-xslt-impl.xsl	Pattern files that are used to describe the actual mapping between a source and a target element.	Pattern
hto.xsd, museoliitto.xsd	The source and target schemas representing the data structure of the Museoliitto museum and the Harmonise ontology.	Schema
museoliitto_mapping.harmonize	The actual mapping that defines the relationships between elements in the Museoliitto source data schema and elements in the Harmonise ontology.	Mapping

All files are available on the HarmoSearch platform and can be downloaded as an archive file (Zip). For more information see Section 3.2.

4.1 CREATE A NEW MAPPING PROJECT

This section addresses the tasks necessary to create a new mapping project that only contains an empty mapping file (no mappings defined). A mapping project serves as central repository containing all necessary resources for creating and managing data mappings (see Section 1.2) for more details on individual resource types). As such a mapping project can also be easily passed to third parties. Data integrators for example could use this as an opportunity to save their customers valuable time by providing a pre-configured mapping project that already contains the majority of necessary resources, e.g., mapping patterns, target schema file, pre-configured process configuration. The only thing the customer has to do in such a case is adding his own data schema as well as potential data instances.

When creating a mapping project two things have to be considered:

1. Similar to a file system mapping projects must have a unique name otherwise it will not be possible creating the mapping project in the current workspace (location on the local hard-disk).
2. A name for the mapping file that will be created by default. Changing the name is optional but recommended.
3. A proper selection of the character encoding. Per default this option is set to UTF-8 which should fit common needs. However, in case the mapping contains special characters it might be necessary to change the encoding type.

In the following a step-by-step guidance illustrates how to create a new mapping project for the Museoliitto business scenario described in Section 4.

Step-by-step guidance:

1. Call the Mapping Project Wizard by clicking **File→New→Mapping Project**
2. Enter Museoliitto into the **project name** text field
3. Enter **museoliitto_mapping.harmonize** into the **file name** text field
    a. **Note:** This step is optional but recommended in order to add additional meaning to the data mapping
4. Do not change the **XML encoding** (use default value **UTF-8**)
    a. **Note:** Additional information on supported character encodings is provided in Section 6.5.1.

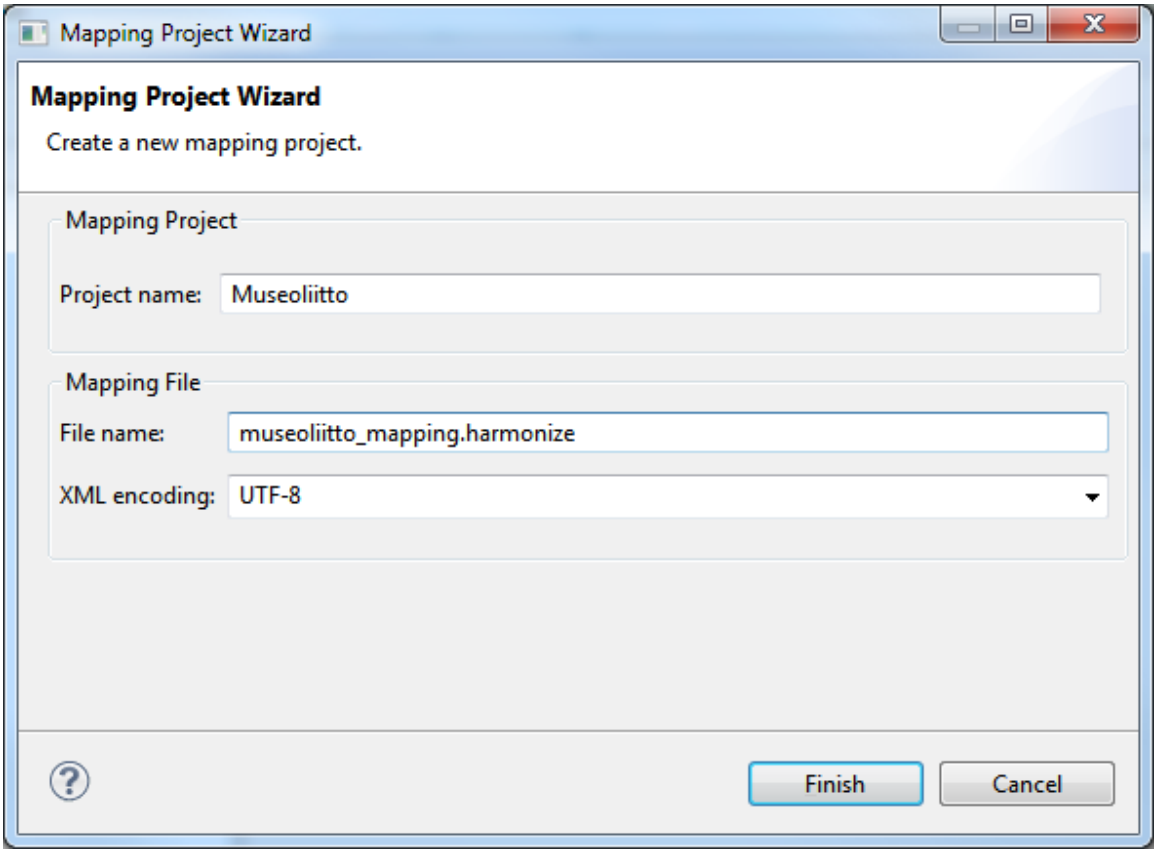


Figure 9: Mapping Project Wizard

5. Click **Finish**

The newly created Museoliitto mapping project is added to the current workspace and displayed in the Mapping Navigator View (cf. Figure 10).

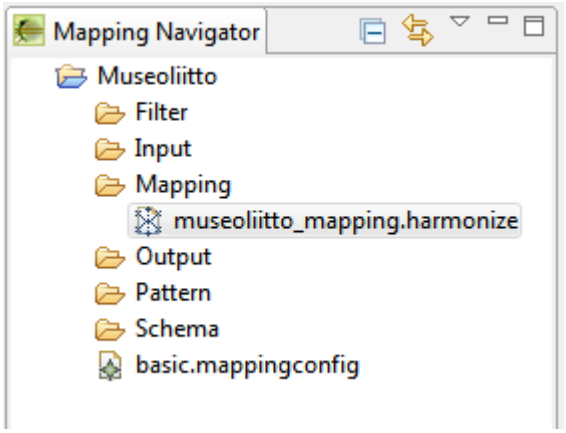


Figure 10: The Museoliitto mapping project is added to the current workspace and displayed in the Mapping Navigator View

Additional information:

- Mapping Project Wizard (see Section 6.5.1)
- Mapping Navigator View (see Section 6.3.1)

4.2 SETTING UP THE MAPPING PROJECT

After creating a new mapping project the required resources necessary for creating the mapping between Museoliitto and Harmonise must be added. For example:

- the source and target schemas representing the data structure
- a mapping filter proposing a structured guideline for the mapping of recommended elements in the source data schema
- a collection of mapping patterns and
- a collection of several input files such as instance data collected from the Museoliitto database or the process transformation configuration

The purpose of this step is to initialize and configure the mapping project and its resources in order to create a concrete mapping. Thereby, the above listed resources have to be imported to the mapping project. This is done using several wizard-driven dialogs allowing the user to import certain files to the mapping project.

For demonstration purposes the section has been divided into the following sub-section:

1. **Add schema files** - This section describes how schema files are added to the Museoliitto mapping project.
2. **Add a mapping filter** - Deals with importing a mapping filter to the mapping project using the Import Filter File Wizard.
3. **Add pattern files** – explains how to add a collection of properly defined mapping patterns can be added to the mapping project.
4. **Add input files** – deals with the import of certain files such as instance data files or process configurations. These files are mainly used during the transformation procedure.

Each section will be described in more detail by means of a step-by-step user guidance. At the end of this section the Museoliitto mapping project will contain all necessary files that are needed in order to begin creating the concrete data mapping.

4.2.1 Add schema files

In this section the import of the Museoliitto and Harmonise data schema will be described. A schema is used to specify the actual content and hierarchy of the source and target data elements. Thus, schemas are the basis on which a mapping operates.

Importing schema files is done via the Import Bundled Files Resource Wizard. The wizard is able to import single or multiple schema files into an existing mapping project. Triggering the import copies the schema file(s) from the specified location to the Schema folder of the mapping project.

The user has to provide the following information:

- Path to a schema file (or bundle) on the local hard-disk
- The target mapping project

As mentioned the wizard is able to add multiple schema files at once, which is more efficient than importing each schema file separately. The basic requirements for importing multiple files at once are that the files to be imported are archived using a Zip file (zipping resources can be easily done by using the Windows Zip functionality, which will not be described here in more detail).

In the following the guideline on how to import schema files will use the multiple file import option. The single file import, however, follows the same pattern. Thus, the provided guidelines also apply for the single file import.

Step-by-step guidance:

1. Call the Import Wizard from the main menu by clicking **File→Import…**
2. Expand **HarmoSearch Wizards**
3. Select **Import Schema File**

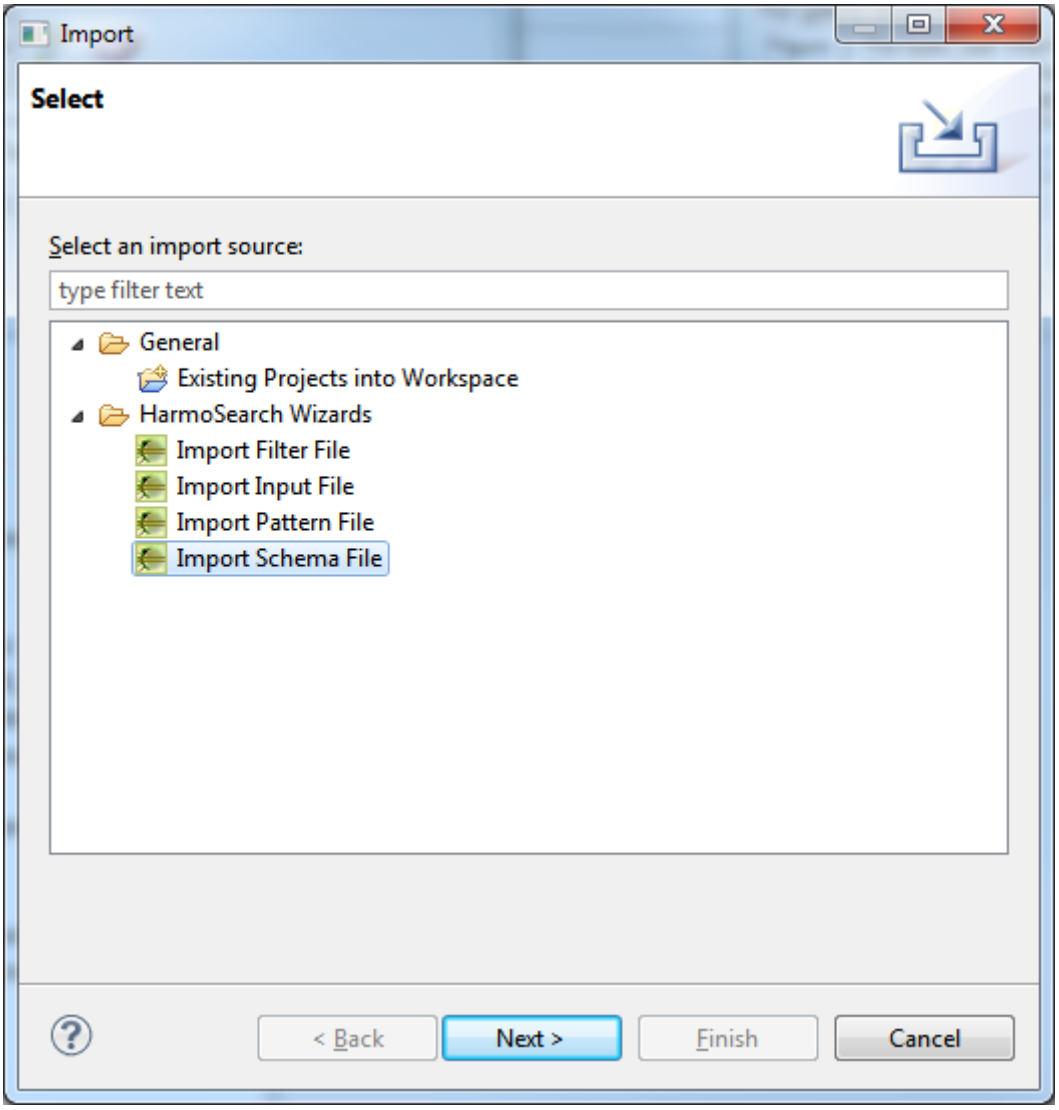


Figure 11: Select the Import Schema File wizard in the Import Wizard dialog

4. Click **Next**
5. Select **Choose a bundle of files packed in a ZIP file**
   a. Alternatively click **Choose an existing file from local source** if you want to import the schema files separately
6. Click **Browse** and locate the **schemas.zip** file on your local hard-disk
7. Press **Select All** to check the listed schema files in case the **hto.xsd** and the **museoliitto.xsd** files are not checked.
8. Select the **Museoliitto** mapping project as **target project** from the list below

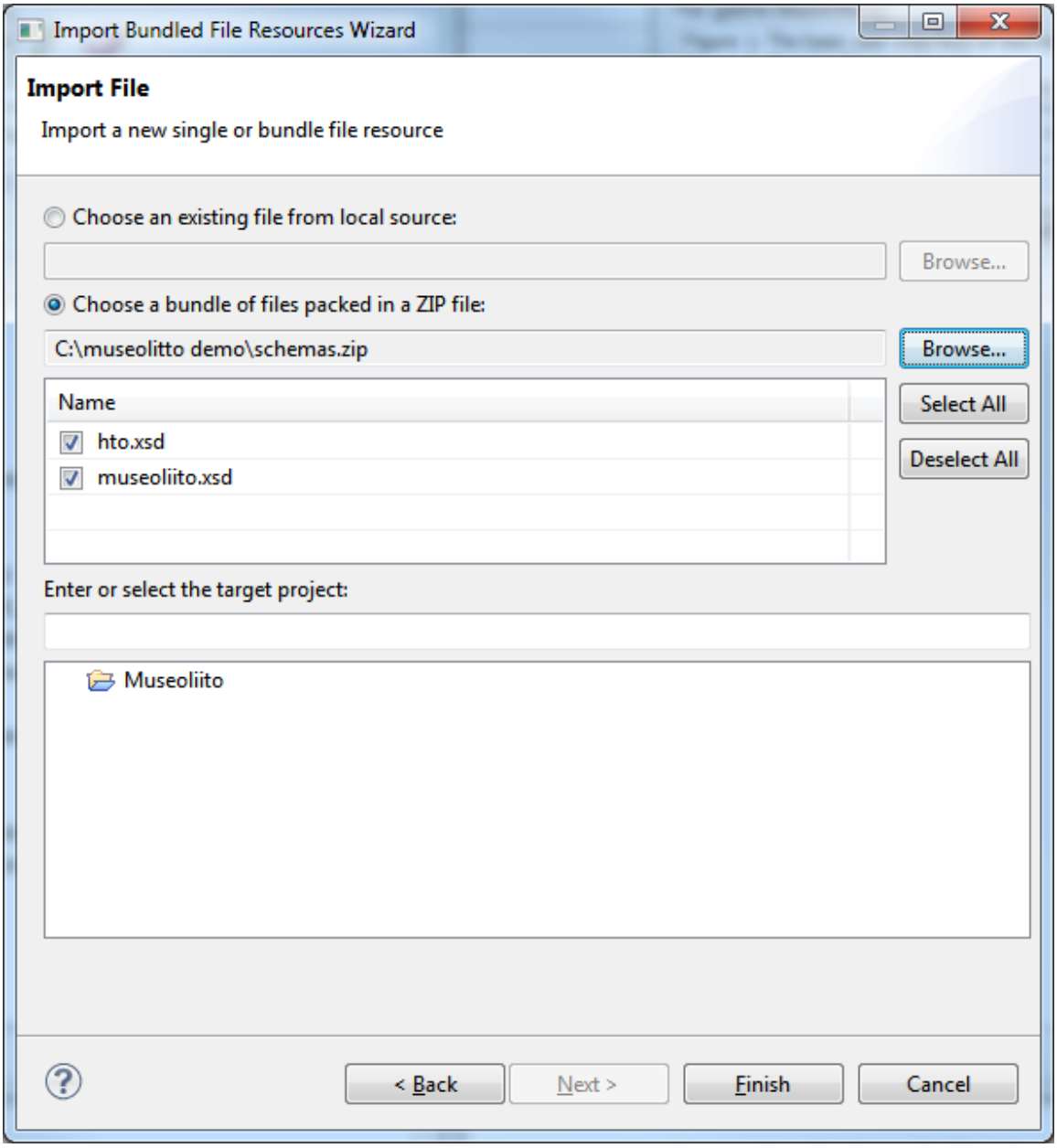


Figure 12: Importing the two schema files hto.xsd and Museoliitto.xsd

9. Click **Finish**

The selected schema files hto.xsd and museoliitto.xsd are imported to the Schema folder of the Museoliitto mapping project. The result is shown in the Mapping Navigator View in Figure 13.

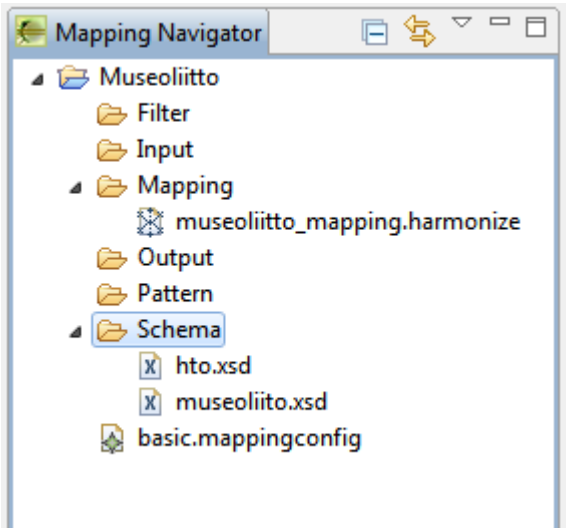


Figure 13: The schemas hto.xsd and museoliitto.xsd in the Museoliitto Schema folder

Additional information:

- Import Bundled Files Resource Wizard (see Section 6.5.2)
- Mapping Navigator View (see Section 6.3.1)

4.2.2 Add a mapping filter

In a next step a mapping filter is added to the mapping project. A mapping filter provides guidance for the user suggesting mandatory data elements in the target schema. Although not mandatory it is recommended using one since a filter allows creating a mapping in a structure way.

Importing a mapping filter is done by using the Import File Resources Wizard. The wizard copies a filter file from the local hard-disk into the Filter folder of a mapping project. Note, the mapping project must exist in the current workspace.

The following user input is required:

- Path to the filter file on the local hard-disk
- The target mapping project

Step-by-step guidance:

1. Call the Import Filter Wizard from the main menu by clicking **File→Import…**
2. Expand **HarmoSearch Wizards**
3. Select **Import Filter File**

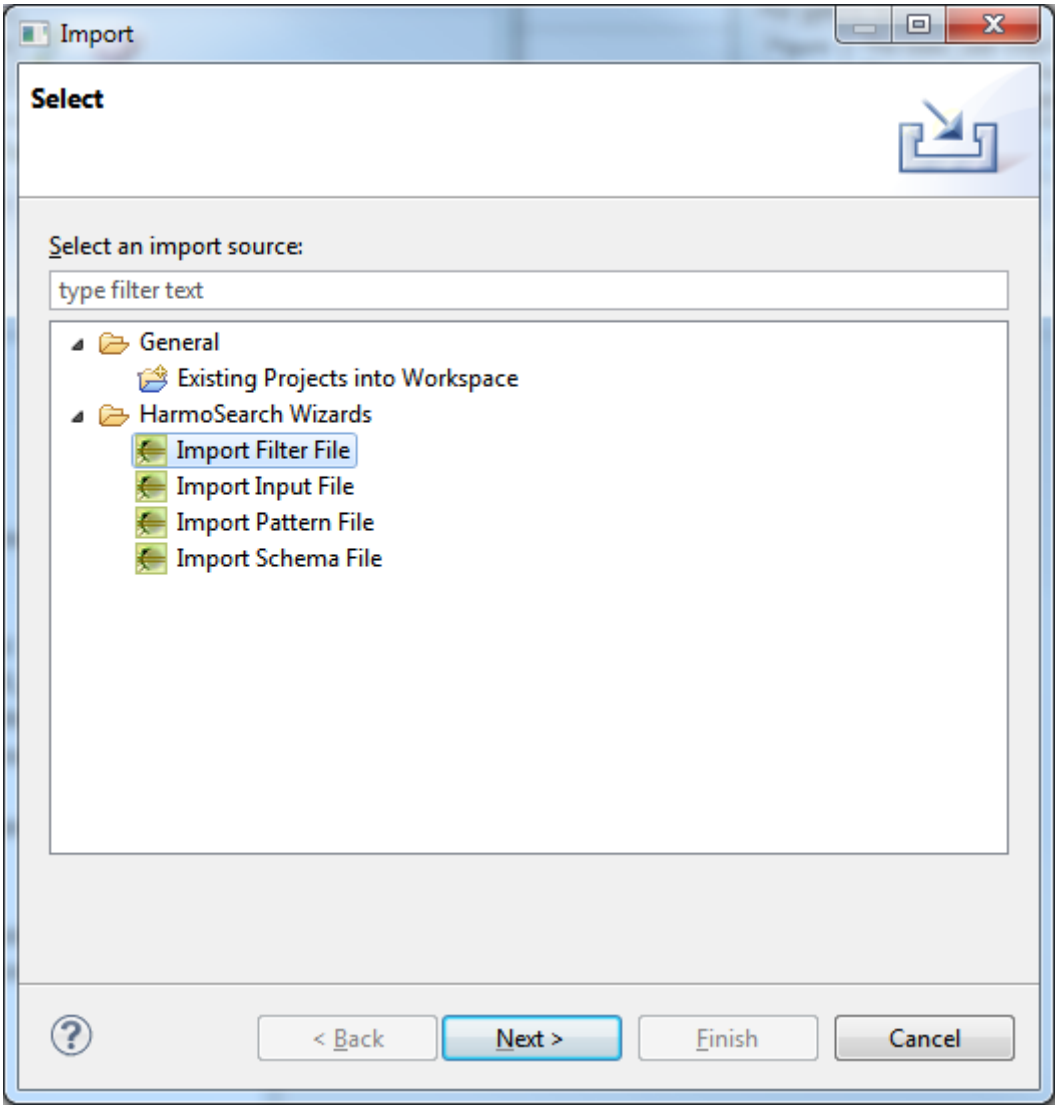


Figure 14: Select Import Filter File in the Import Wizard dialog

4. Click **Next**
5. Click **Browse** and locate **filter.xml** on the local hard-disk
6. Select the **Museoliitto** mapping project from the workspace below

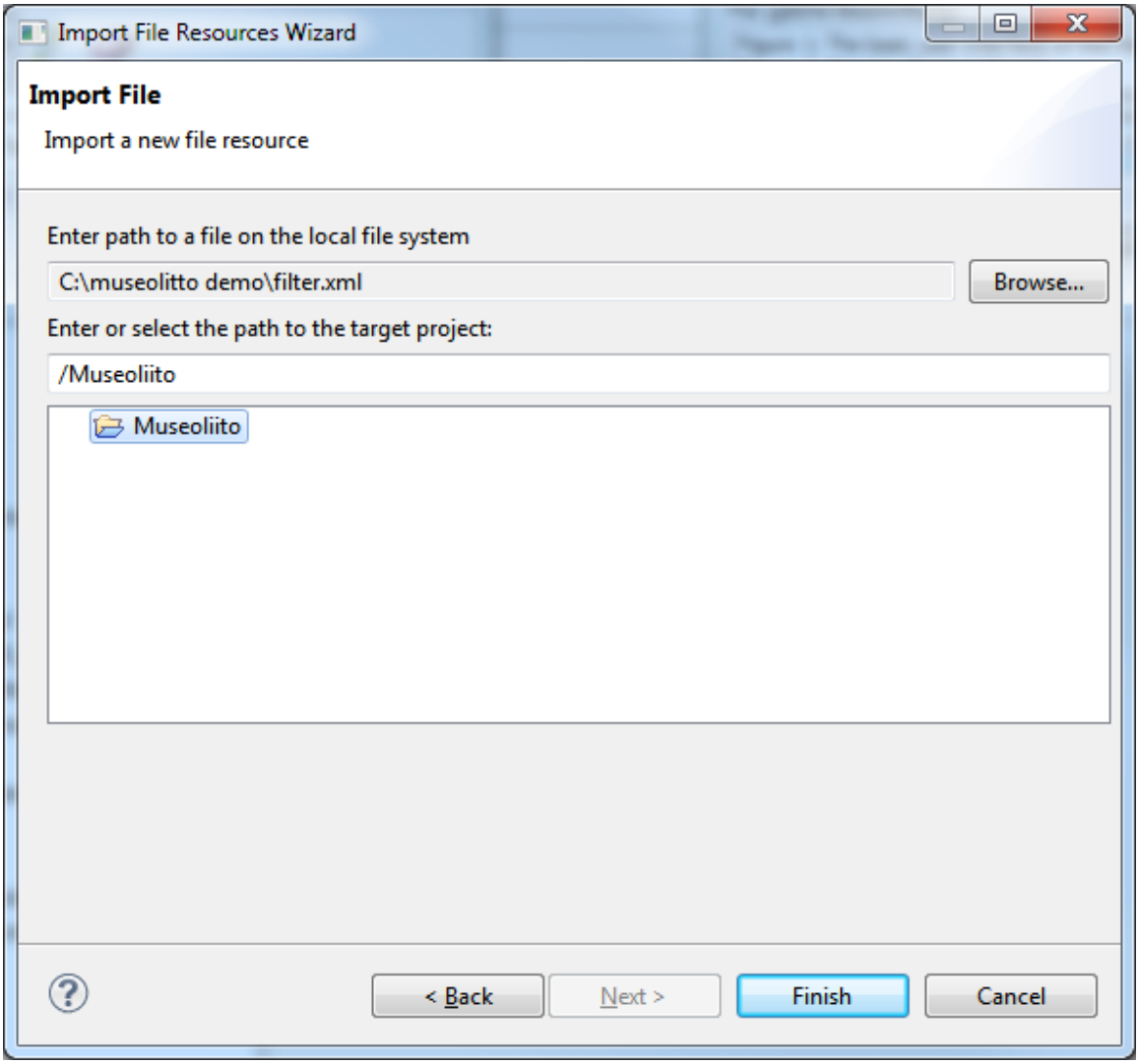


Figure 15: Importing a filter file into the Museoliitto mapping project

7. Click **Finish**

As a result filter.xml is added to the Filter folder of the Museoliitto mapping project as illustrated in Figure 16.

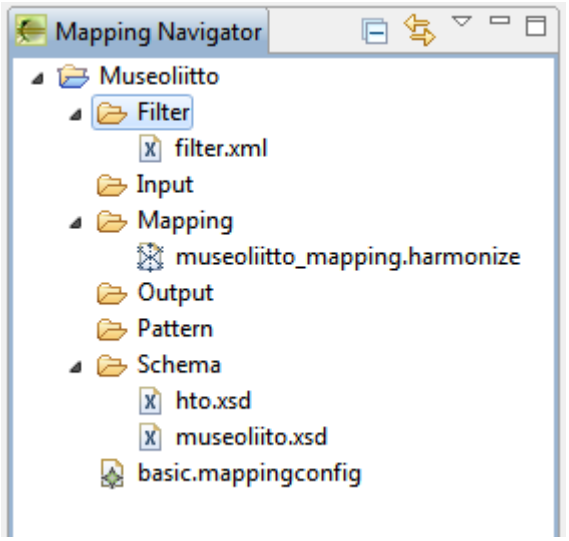


Figure 16: The filter file filter.xml in the Museoliitto Filter folder

Additional information:

- Import File Resources Wizard (see Section 6.5.3)
- Mapping Navigator View (see Section 6.3.1)

4.2.3 Add pattern files

Another resource which may be part of a mapping project are pattern files. Pattern files are reusable building blocks that govern the translation from one mapping element to another mapping element. As such patterns are reusable and can be used in multiple mapping projects.

Importing pattern files is done via the Import Bundled Files Resource Wizard. Again, the wizard is able to import either a single file or multiple files. Triggering the import copies the pattern file(s) from the specified location to the Pattern folder of the mapping project.

The following user information is required:

- Path to the pattern file (or bundle) on the local hard-disk
- The target mapping project

Similar to the schema import guidelines presented in Section 4.2.1 the step-by-step guidance in this section will also use the multi-file import approach. Following the guideline will add the following pattern files to the Museoliitto mapping project:

- **date-time.xsl** – maps different date formats to the Harmonise date format
- **Euromuse_RL_languages.xslt** – maps different Euromuse related elements to the Harmonise ontology
- **euromuse2html.xsl** – a transformation pattern allowing HTML output
- **HTO_Euromuse.xslt** – transformation pattern for mapping Harmonise to Euromuse
- **htov2-1.xsl** – additional patterns addressing basic mappings of Harmonise elements
- **htov2-1-extension.xsl** – extended patterns contributing to the htov2-1-xsl mapping patterns

Step-by-step guidance:

1. Call the Import Wizard from the main menu by clicking **File→Import…**
2. Expand **HarmoSearch Wizards**
3. Select **Import Pattern File**

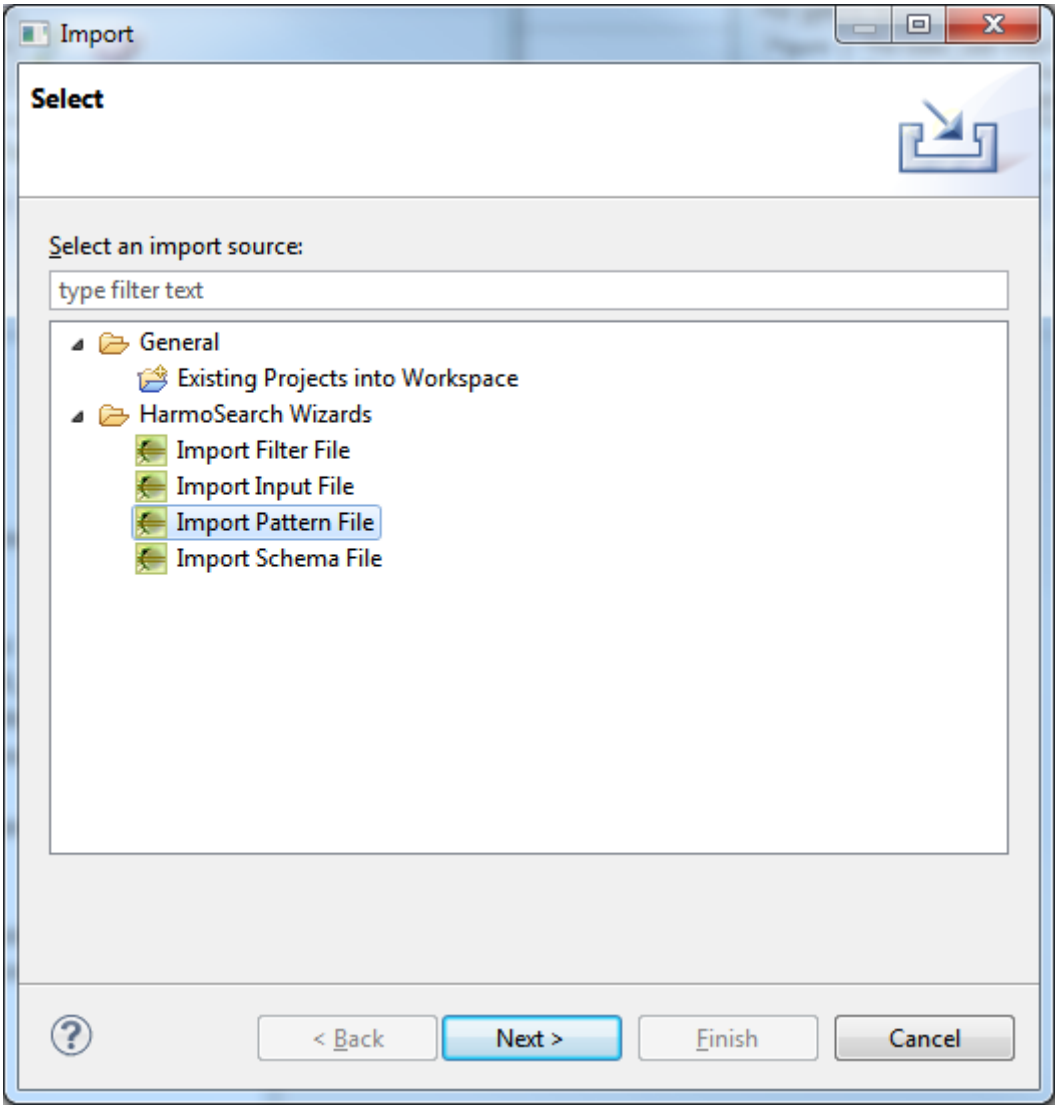


Figure 17: Select Import Filter File in the Import Wizard dialog

4. Click **Next**
5. Select **Choose a bundle of files packed in a ZIP file**
   a. Alternatively click **Choose an existing file from local source** if the import process should only import a single pattern file.
6. Click **Browse** and locate the **patterns.zip** file on the local hard-disk
7. Press **Select All** to check the listed pattern files if not done already
8. Select the **Museoliitto** mapping project as **target project** from the list below

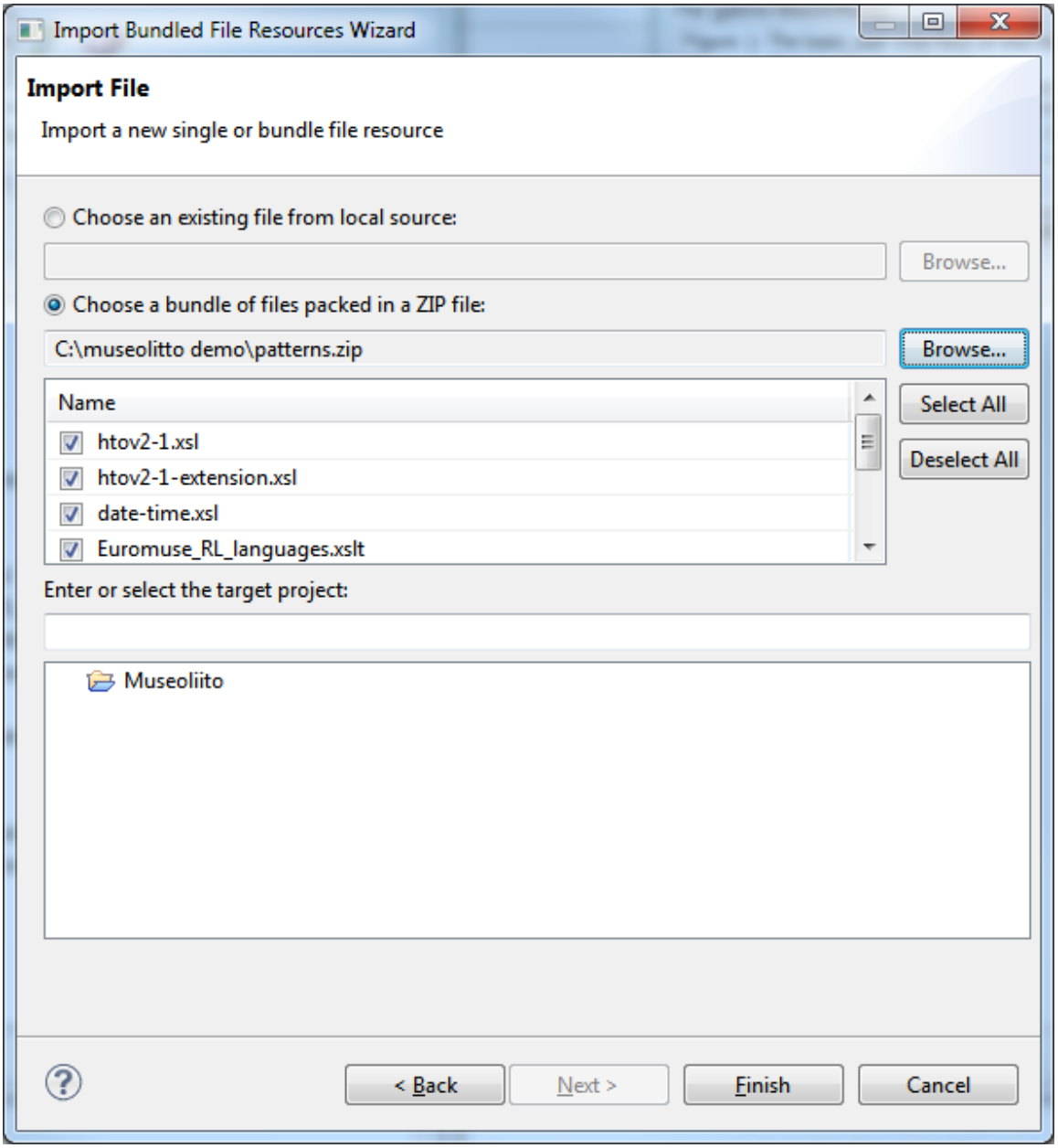


Figure 18: Importing pattern files to the Museoliitto mapping project

9. Click **Finish**

As shown in Figure 19 the selected pattern files are added to the Pattern folder of the Museoliitto mapping project and displayed in the Mapping Navigator View.

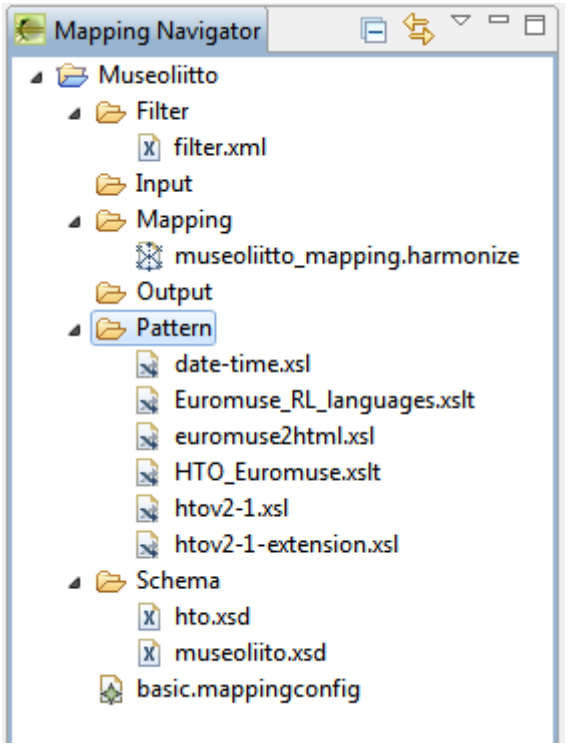


Figure 19: The selected patterns added to the Pattern folder of the Museoliitto mapping project

Additional information:

- Import Bundled Files Resource Wizard (see Section 6.5.2)
- Mapping Navigator View (see Section 6.3.1)

4.2.4 Add input files

This section shortly explains how to import certain input files to the Museoliitto mapping project. Input files are resources relevant for the transformation procedure (e.g., concrete instance data) or the value mapping (e.g., reference lists).

Similar to schema and pattern files input files can be imported separately or via a Zip archive. For convenience reasons the guideline below will use multiple-file import again.

The following files will be added to the Museoliitto mapping project:

- **museoliitto.xml** – instance data that allows the transformation to operate on real-world instances.
- **process.xml** – the configuration file for the transformation process. It allows adding input and defining output parameters for the transformation procedure.

After completing the guidance below both files will be added to the Input folder of the Museoliitto mapping project.

Step-by-step guidance:

1. Call the Import Wizard from the main menu by clicking **File→Import…** or click  from the main toolbar
2. Expand **HarmoSearch Wizards** select **Import Input File**

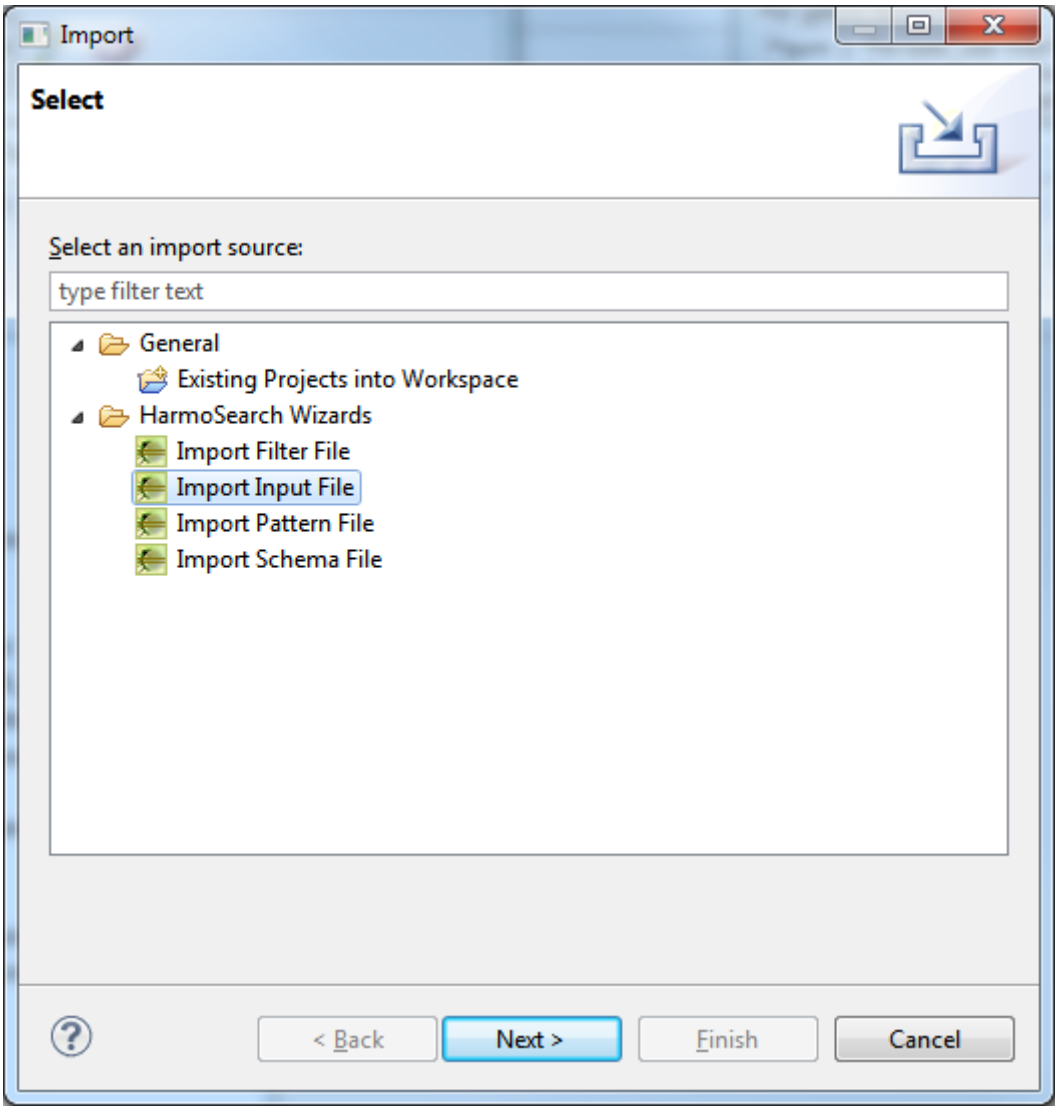


Figure 20: Select Import Input File from the Import Wizard dialog

3. Click **Next**
4. Since multiple input files have to be imported select **Choose a bundle of files packed in a ZIP file**
    a. Alternatively click **Choose an existing file from local source** if only a single file needs to be imported
5. Click **Browse** and locate **input.zip** on the local hard-disk. The Zip file includes all of the required input files
6. Check all files listed by pressing the **Select All** button if not done already
7. Select the **Museoliitto** mapping project as **target project**

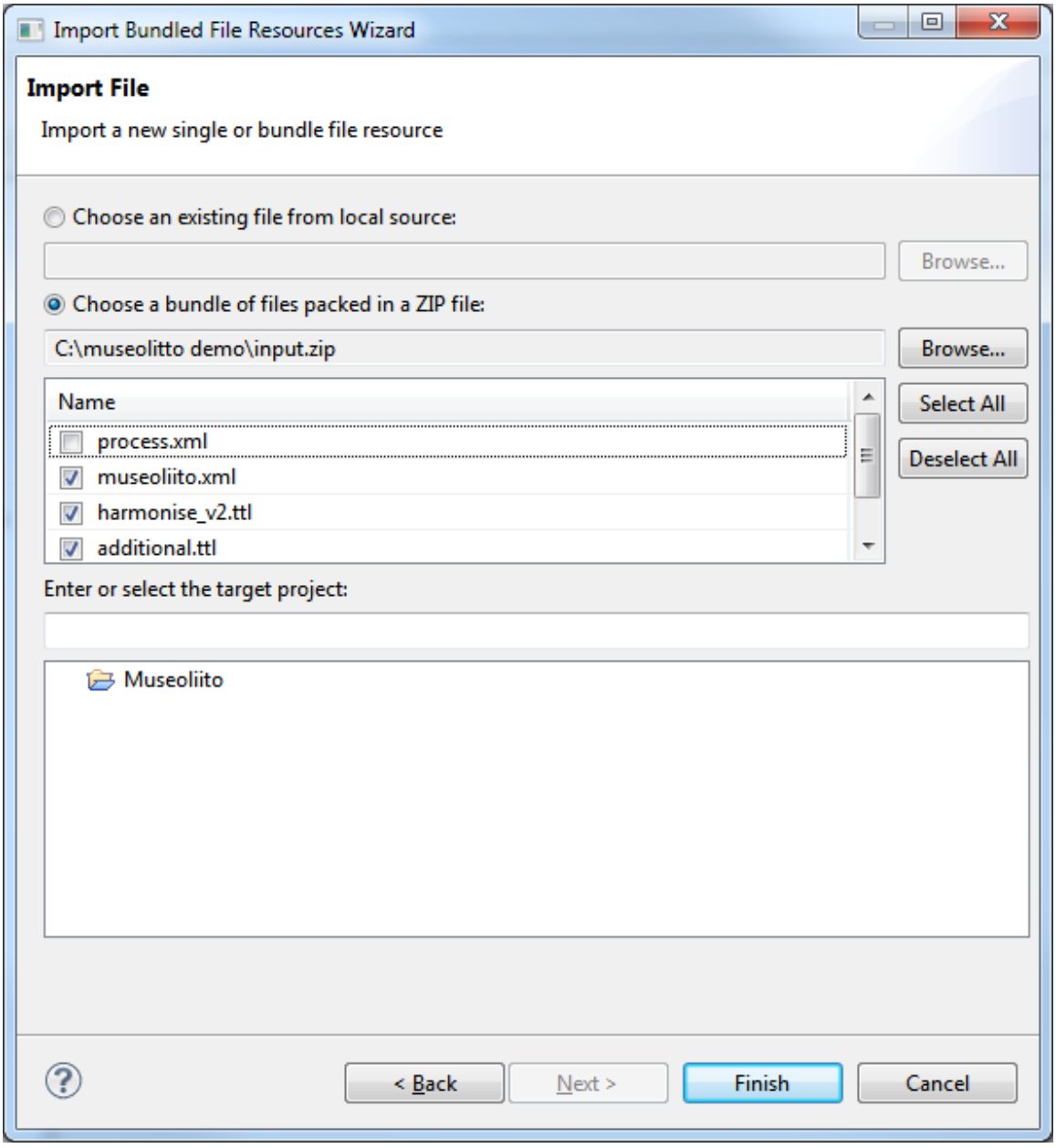


Figure 21: Importing input files process.xml and museoliitto.xml into the Museoliitto mapping project

8. Click **Finish**

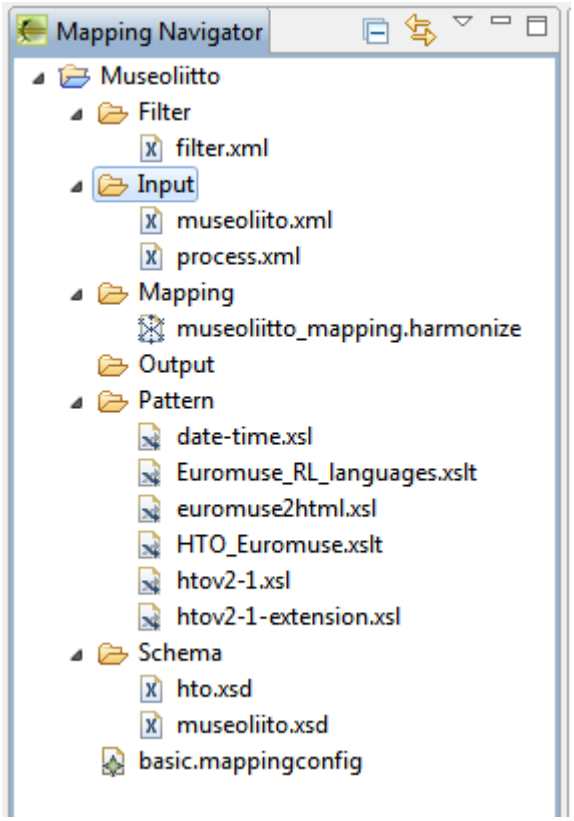


Figure 22: The instance data museoliitto.xml and the reference lists additional.ttl and harmonise_v2.ttl have been added to the Input folder

After the import has finished the Input folder of the Museoliitto mapping project contains two new files: museoliitto.xml and process.xml.

Additional information:

- Import Bundled Files Resource Wizard (see Section 6.5.2)
- Mapping Navigator View (see Section 6.3.1)

4.3 SETTING UP THE MAPPING FILE

This section describes how the corresponding source and target schemas are added to the Museoliitto mapping file. Configuring the mapping file is necessary in order to allow proper mapping configurations.

Per default the mapping file (which has been added to the mapping project at the time it was created in the workspace) is empty. However, in order to create a mapping at least one source and one target schema have to be added.

Coined to the Museoliitto scenario two schemas have to be added: hto.xsd as target schema and museoliitto.xsd as source schema. Adding the schemas is done via the mapping editor, which allows adding proper schemas by simply drag-and-drop the corresponding schema file from the mapping project onto the editor's pane. To do so the user has to perform the following activities:

- Identify the source respectively target schema in the mapping project
- Drag-and-drop the schemas onto the mapping editor
- Select the root node (top-level element of the hierarchy) for each schema

A detailed guideline on how to accomplish the described steps is provided below.

Step-by-step guidance:

1. Expand the **Mapping** folder in the **Museoliitto** mapping project
2. Double-click the mapping file **museoliitto_mapping.harmonize** which opens the HarmoSearch Mapping Editor in the editor pane.
3. Click on the Graph View tab at the bottom of the editor
4. Expand the Schema folder in the Museoliitto mapping project
5. Drag-and-drop the hto.xsd (Harmonise schema) onto the editor's modeling canvas to define hto.xsd as target schema.

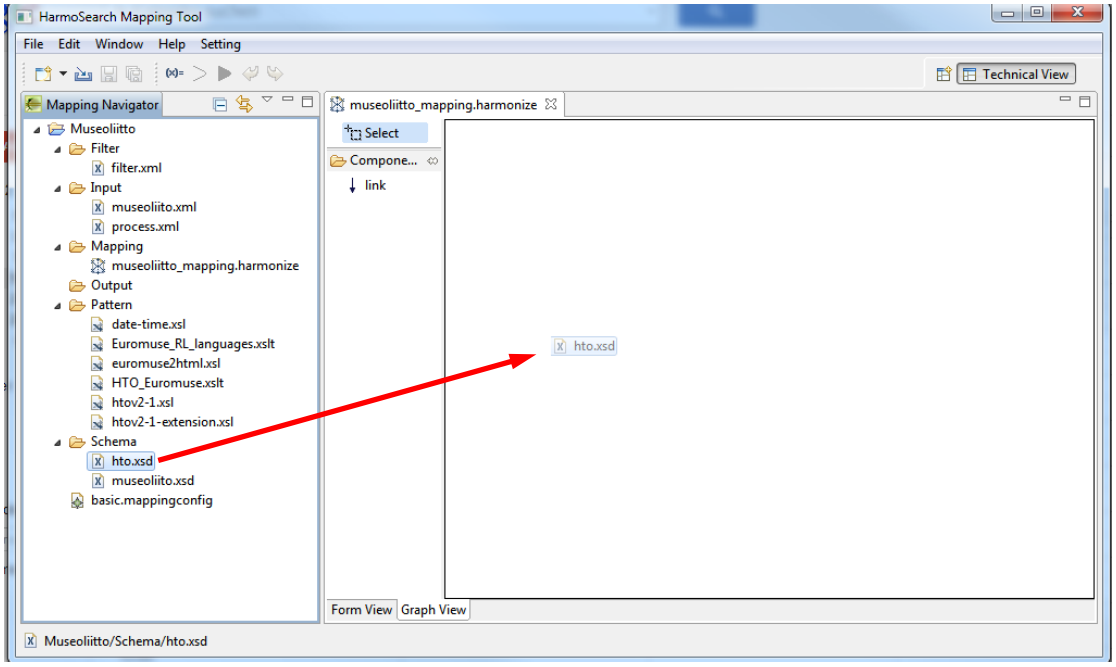


Figure 23: Defining hto.xsd as target schema

6. Since the mapping aims to describe event data select **event** from the drop-down box in the Select Root Element dialog.
7. Click **OK** to add the Harmonise schema to the mapping file. The Harmonise schema is visualized as a tree.

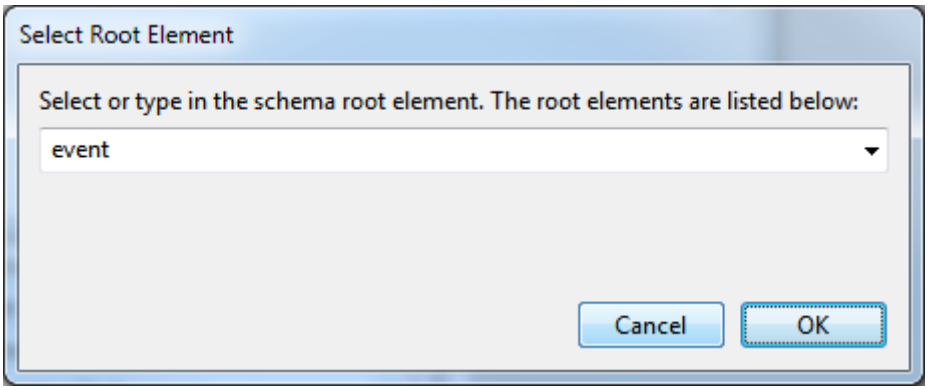


Figure 24: Select event as root element

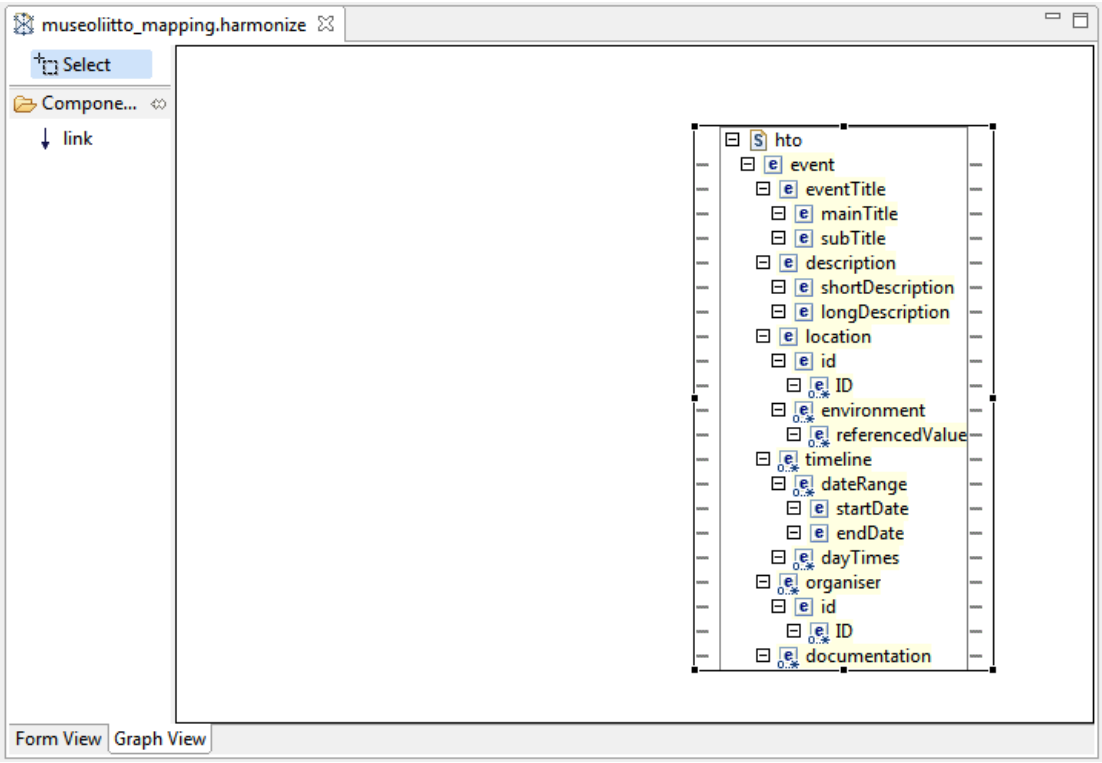
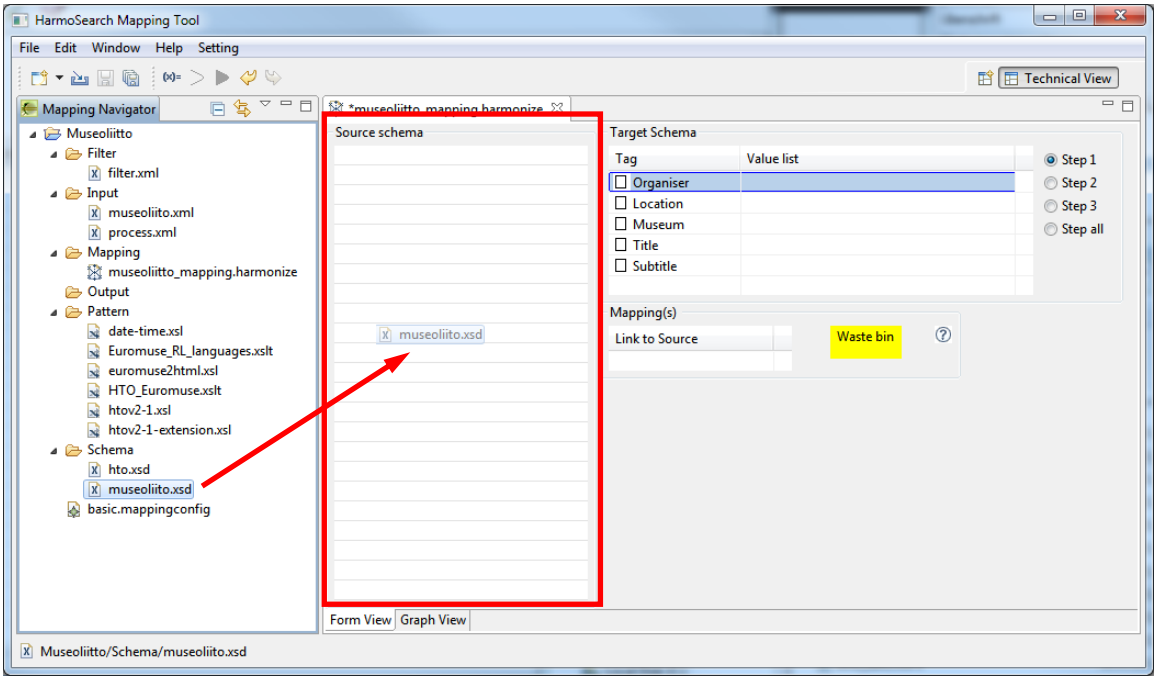


Figure 25: The hto schema visualized as a tree

8. Save your changes by clicking **File→Save** in the main menu or from the toolbar
9. Click on the **Form View** tab at the bottom of the editor
10. Drag-and-drop **museoliitto.xsd** from the Mapping Navigator View onto the **Source Schema area**. The Source Schema area is located on the left hand side of the HarmoSearch Mapping Editor.



11. Select **webservice** from the drop-down box in the **Select Root Element dialog**

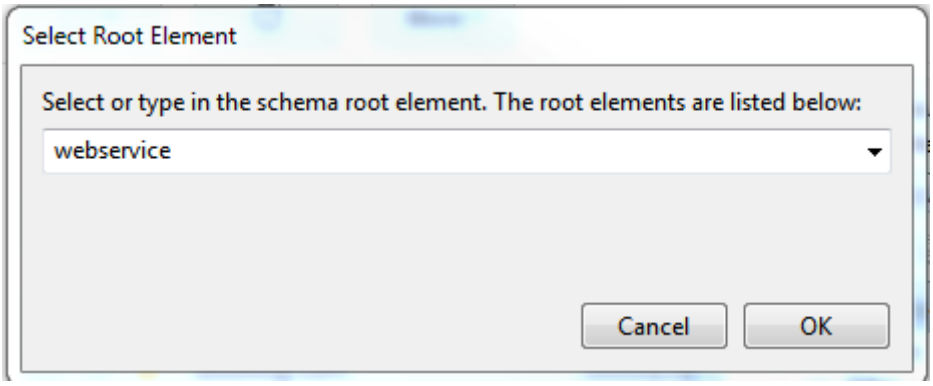


Figure 26: Select webservice as the root element

12. Click **OK** for importing the **museoliitto.xsd** file into the editor. The structure of the schema is visualized by means of a tree having **webservice** set as the root node. Expand the tree for gaining a better overview.

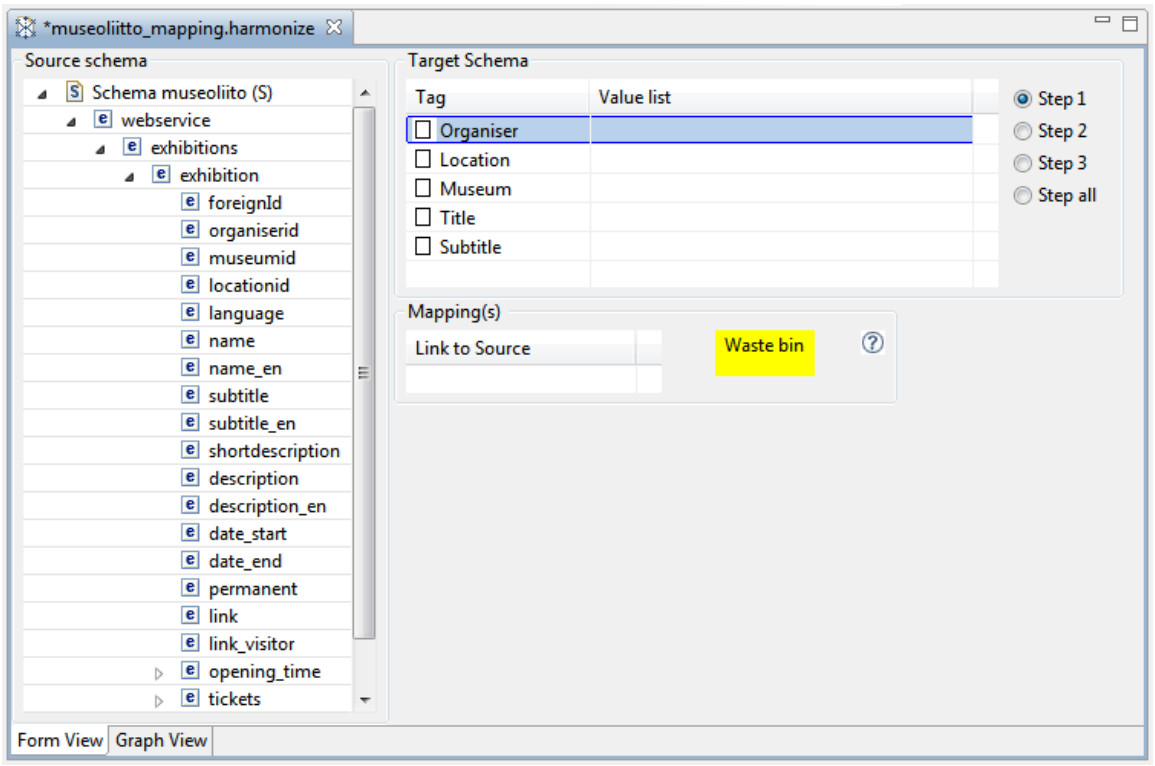


Figure 27: The museoliitto.xsd source schema has been added to the editor

Additional Information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)

4.4 CREATE THE DATA MAPPING

In this section the actual data mapping will be described. Creating a mapping using the HarmoSearch mapping tool is done either fully graphically or form-based. Accordingly, the HarmoSearch mapping editor has two distinct views:

- Form View and
- Graph View

Both approaches have significant advantages and disadvantages. Our experience showed, however, that beginners feel more confident with the form-based approach whereas more experienced users prefer the graphical approach. As mentioned at the beginning of this chapter the HarmoSearch mapping tool supports guidance for novice users who might have no or limited experience with data mappings. Thus, the following data mapping is described using the form-based approach. More information on both views is provided in Section 6.4.1.

One main advantage of the form-based approach is the integrated guidance support using a dedicated mapping filter. Simplified a mapping filter provides a structured way a user can follow when creating a data mapping. Thereby, the filter recommends certain elements as mapping targets to the user. The HarmoSearch mapping tool does not mandate using a mapping filter in combination with the form-based approach. However, experience showed that a structured and guided way on how to map elements from a given target schema is valuable and highly welcomed by domain experts since more time and cost efficient.

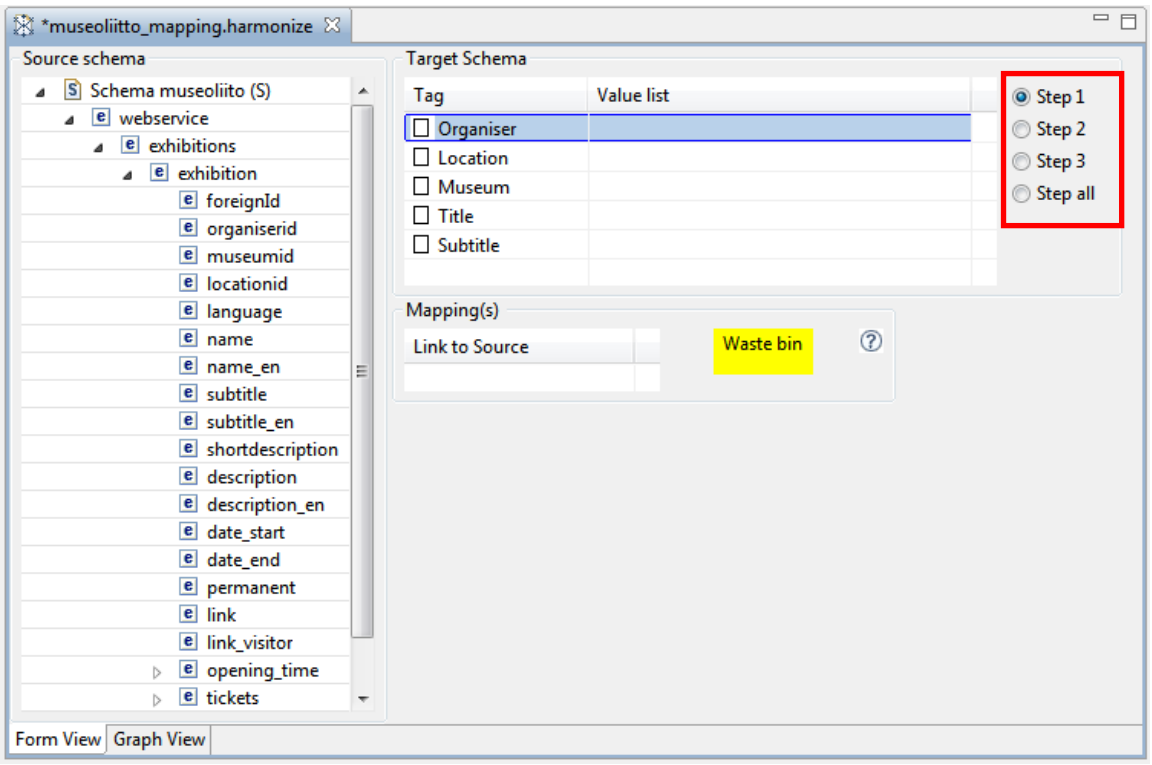


Figure 28: The mapping filter is fully integrated and supported by the Form View

The mapping filter is fully integrated into the HarmoSearch mapping editor. The blocks (in the following referred as steps) are visualized by means of a radio list at the left hand side of the Form View (cf. Figure 28) in descending order.

The tutorial at hand provides a pre-configured mapping filter. The filter consists of three basic steps that identify Harmonise elements typically used to describe events in the cultural heritage domain. The elements clustered for each step are listed below:

Step name:	**Synopsis:**	**Elements:**
Basic information	Clusters the basic information describing an event in the cultural heritage domain. For example information about the organizer or the event's title.	• organiser • location • main title • sub title
Additional information	Addresses additional information that might be of interest to potential visitors, e.g., a description of the event under consideration or the start and end dates.	• long description • short description • start date • end date • opening hours
Rich meta-data	Concerns rich meta-data such as media files, links or pictures.	• multimedia item • image item

The remainder of this section describes the tasks and activities that are necessary for defining the data mapping according to the suggested filter steps. At the end of this section the mapping will be defined and ready to be transformed and tested.

4.4.1 Mapping of basic event information

The first step suggested by the mapping filter addresses basic information that are usually used to describe a cultural event, e.g., name of the organizer, location of the event, or the event's tile.

The following elements are listed as potential mapping candidates:

The reminder of this section provides a step-by-step guidance on how to create and configure the actual mapping links between recommended target elements in the Harmonise schema and the proper source elements in the Museoliitto schema as listed below:

Filter (Harmonise schema)	Museoliitto schema
Organiser	organiserid
Location	locationid
Museum	museumid
Title	• name • name_en
Subtitle	• subtitle • subtitle_en

The first 3 elements organiserid, locationid, and museumid in the Museoliitto schema are unique identifiers. Thus, they are mapped to ID elements in the Harmonise ontology. The pattern of choice is thereby the hto:ReferencedValue mapping pattern. It transforms a given value into a proper ID element as specified by the Harmonise ontology.

The remaining elements (name, name_en, subtitle, and subtitle_en) are all language text elements. They are mapped using the hto:languageText mapping pattern. The pattern has two parameters: text and language. The text parameter represents the actual text content whereas the language parameter specifies the text's language, e.g., English (en) for description_en or Finish (fi) for the shortdescription.

Step-by-step guidance:

1. Expand the **Mapping** folder in the **Museoliitto mapping project** and double-click the **museoliitto_mapping.harmonize** mapping file (if not already opened in the editor pane).
2. In the HarmoSearch Mapping Editor expand the **Source Schema**
3. Click on **Step 1** and select **Organiser** from the **Target Schema list**
4. Find the **organiserid** element in the **Source Schema** tree, drag it onto the **Mapping(s) list** and **click** it once. Notice the black square next to the **organiserid** in the Mappings list shows that the mapping has been defined but not configured.
5. In the **Mapping Details** section select **hto:ReferencedValue** from the mapping pattern list.
6. Enter **'.'** for the **value** parameter to denote that the current value of the organiserid element will also be set to the target element. For example, if the organiserid is set to the value of 123 the organiser id after the transformation will also remain 123 (but represented using the Harmonise ontology).
7. Click **Confirm** to persist the mapping. As result a **green box** appears in the **upper left** corner of the **Organiser** element in the Target Schema. In addition the **organiserid** in the **Mapping(s)** list changed its color from black to **green**.

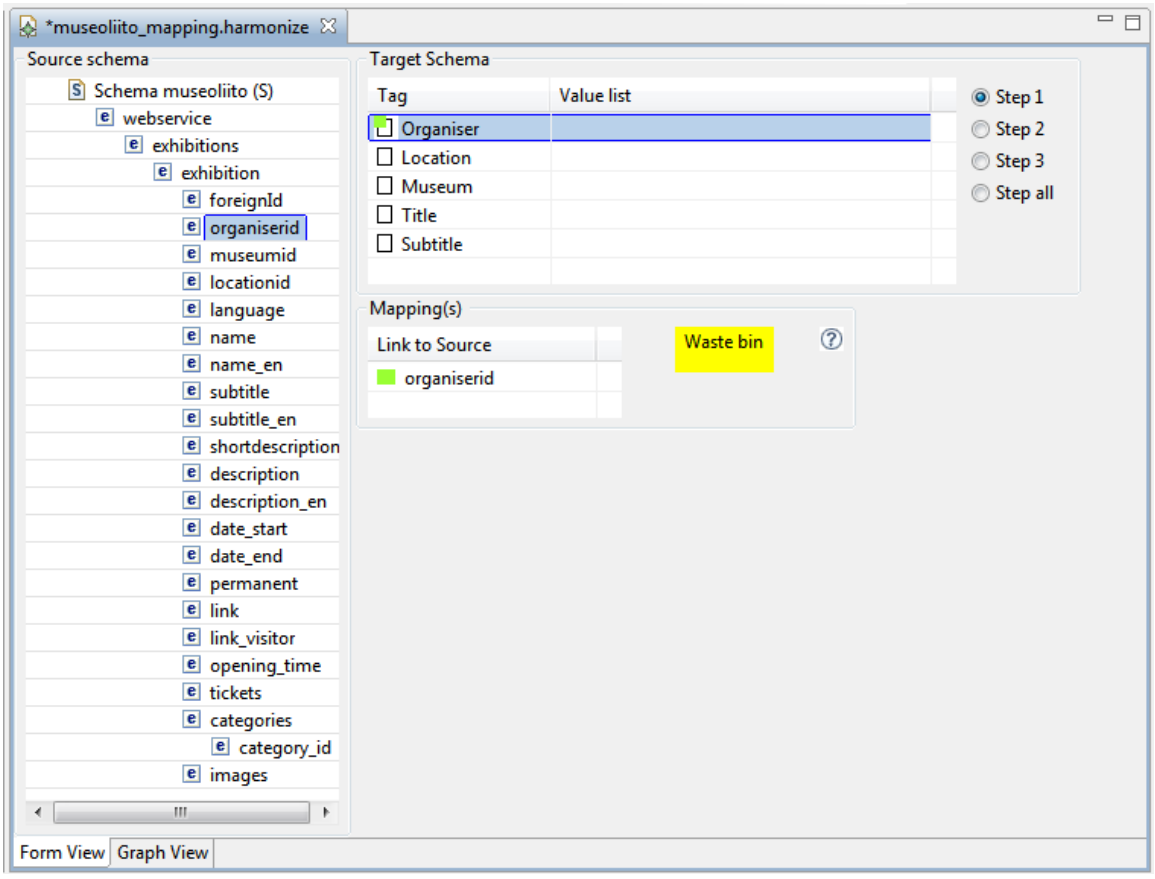


Figure 29: Finalized mapping for the target element Organiser

8. Next select **Location** from the **Target Schema** list
9. Find the **locationid** element in the **Source Schema**, drag it onto the **Mapping(s)** list and **click** it once.
10. Again select the **hto:ReferencedValue** mapping pattern from the Mapping Again the pattern should inherit the value of the original source element. Thus, enter **'.'** Into the text box for the **value** parameter.
11. Click **Confirm** to persist the configuration.
12. Select the **Museum** element from the **Target Schema** list.
13. Find **museumid** in the **Source Schema**, drag it onto the **Mapping(s)** list and click it once.
14. Select **hto:ReferencedValue** from the mapping pattern list and enter **'.'** For the **value** parameter (same as for locationid and organiserid).
15. Click **Confirm** to persist the configuration

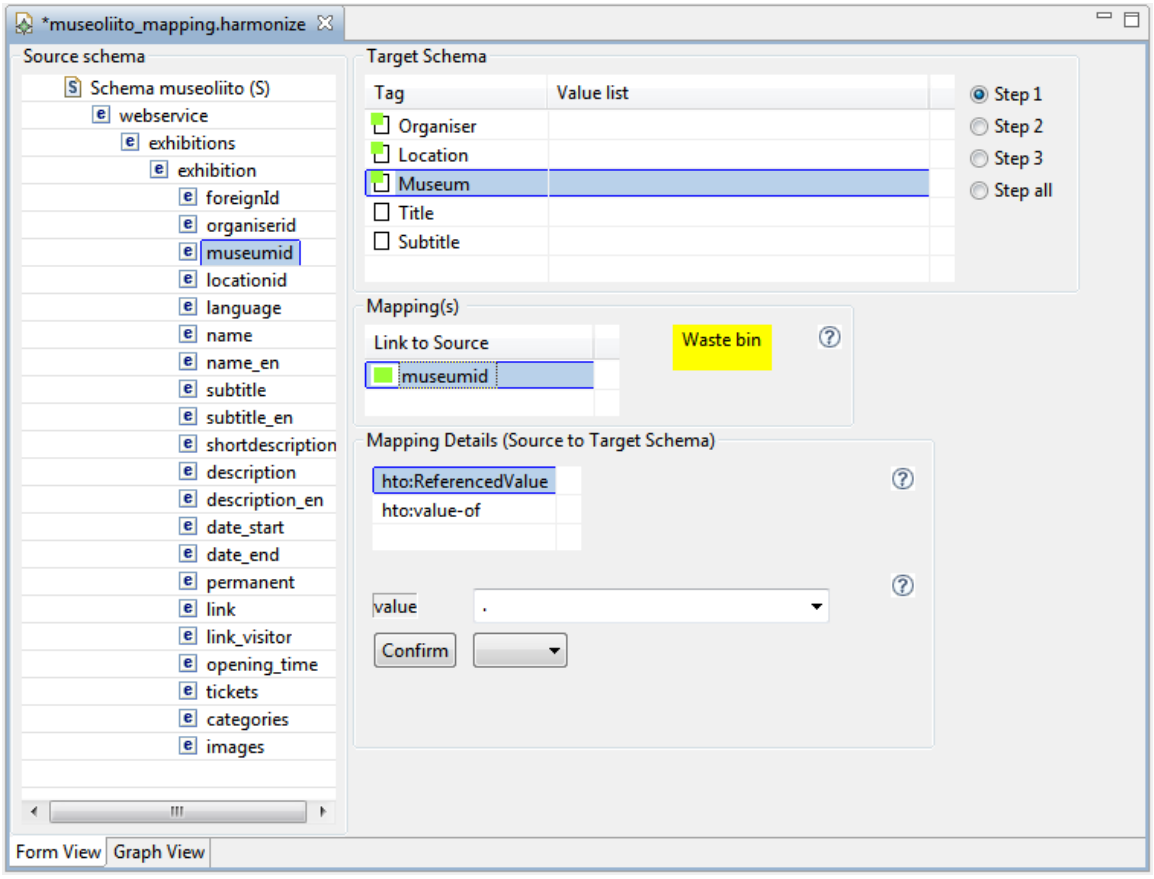


Figure 30: Confirmed mapping for the Museum element

16. Select **Title** in the **Target Schema** list
17. The **Title** element has **two** proper elements in the Museoliitto schema: **name** and **name_en**. Drag and drop both elements onto the **Mapping(s)** list.
18. Select the **name** element in the **Mapping(s)** list.
19. Since a title has to be mapped to a **language text element** in the Harmonise ontology select the **hto:languageText** pattern from the pattern list. The language text pattern has two parameters: **text** and **language**. The

text parameter inherits the value of the original source element and the language parameter has to be set to Finish (the native language Museoliitto describes its events). Accordingly enter **'.'** For the **text** and **fi** for the **language** parameter from the drop-down box. Finally accept the mapping by clicking **Confirm**.

20. **Repeat step 19** for the **name_en** element in the **Mapping(s)** list. This time, however, enter **en** for the **language** parameter. **Text** stays with **'.'**. Again, accept the mapping by clicking **Confirm**.
21. The last element **Subtitle** is mapped to the **subtitle** element in the **Museoliitto Source Schema**. Locate and drag the **subtitle** and the **subtitle_en** element from the tree onto the **Mapping(s)** list.
22. First select **subtitle** from the **Mappings(s)** list and choose the **hto:languageText** pattern from the **Mapping Details** list.
23. Enter **'.'** for the **value** and again **fi** for the **language** parameter
24. Confirm your mapping by clicking the **Confirm** button
25. Select **subtitle_en** from the **Mapping(s)** list, select **hto:languageText** as mapping pattern
26. Enter **'.'** for the **value** and **en** for the **language** parameter.
27. Confirm your mapping by clicking the **Confirm** button

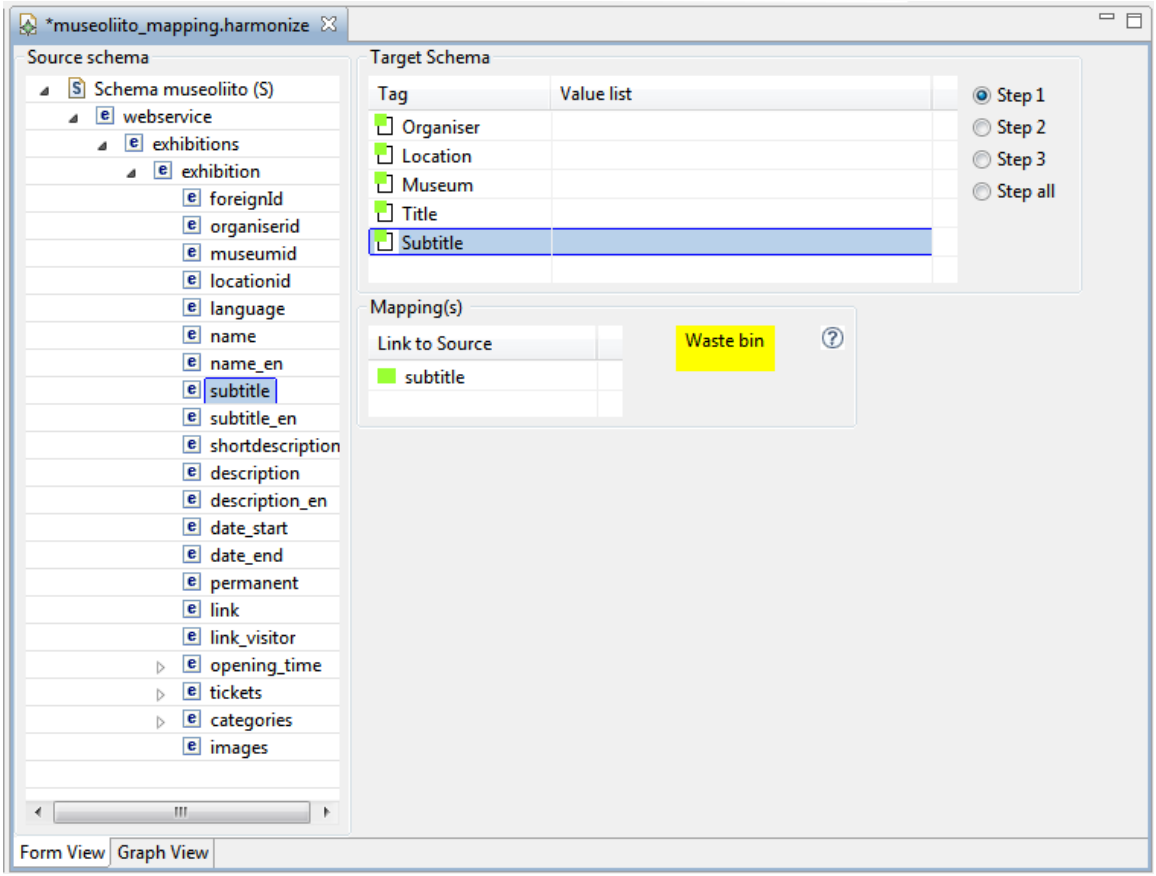


Figure 31: Final mapping for Step 1 (Basic Information)

After completing the mappings all of the listed target elements should have turned green showing that a mapping has been created and configured.

Additional Information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)

4.4.2 Mapping of additional event information

The second step of the mapping filter assembles additional information such as opening hours or short and long descriptions of a cultural event.

The remainder of this section provides a step-by-step guidance on how to create and configure the actual mapping links between the recommended target elements in the Harmonise schema and proper source elements in the Museoliitto schema. The mapping candidates are listed in the table below:

Filter (Harmonise schema)	Museoliitto schema
longDescription	• description • description_en
shortDescription	shortdescription
Begin Date	date_start
End Date	date_end
Day Times	opening_time

The description elements (description, description_en, shortdescription) are simple text elements and are mapped to language text elements in the Harmonise ontology. The mapping pattern focused on the creation of language text elements is the hto:languageText pattern. The pattern has two parameters: text and language. The text parameter represents the actual text content whereas the language parameter specifies the text's language, e.g., English (en) for description_en or Finish (fi) for the shortdescription.

The date time (date_start, date_end) elements in the Museoliitto schema are proper candidates for date elements in the Harmonise ontology. The proper mapping pattern for mapping the data format used by Museoliitto is hto:yyyymmdd-date-to-hto-date mapping. The pattern has only a single parameter called date used to specify the actual date to be transformed, e.g., 20120215.

The last element in the list is the opening_time element, which is mapped to a Harmonise day time element, which is accomplished by using the hto:DayTimes mapping pattern. The pattern has 3 parameters: day, begin, and end. The first parameter is self-explanatory. The latter two, however, specify the actual start and end times, e.g., 9:00 to 19:00.

Step-by-step guidance:

1. Click on **Step 2** in order to narrow the focus on elements describing additional event information
2. Select **longDescription** from the **Target Schema** list.
3. Locate **description** and **description_en** in the **Source Schema** and drag both elements onto the **Mapping(s)** list.
4. First select **description** from the **Mapping(s)** list.
5. **Description** is mapped to a Harmonise **language text** element. Thus, select the **hto:languageText** mapping pattern.
6. Enter **'.'** for the **text** parameter to get sure the value for the target element will be inherited from the source element during the transformation. Enter **fi** for Finish in the **language** parameter text box.
7. Accept the mapping by clicking the **Confirm** button
8. Repeat **step 4** and **5** for the **description_en** element but enter **en** for the **language** parameter.
9. Accept the mapping and press **Confirm**

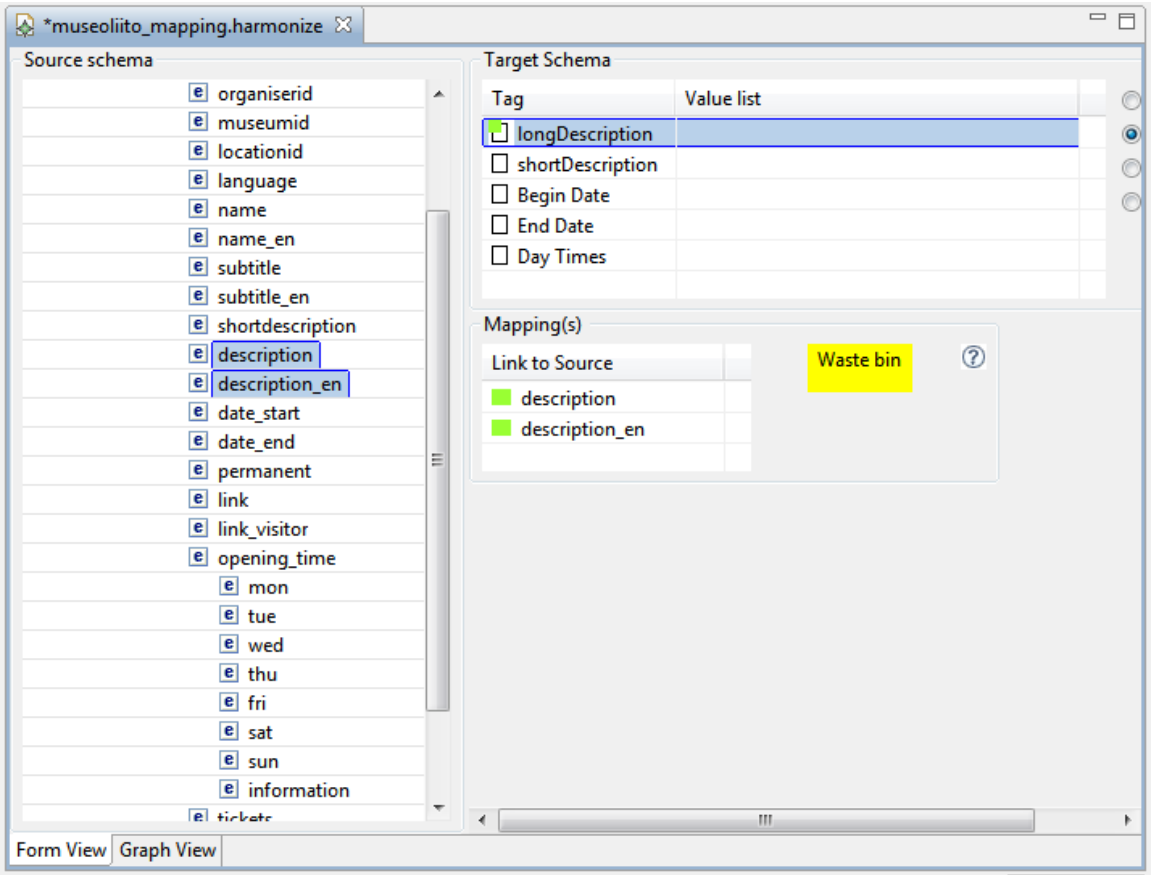


Figure 32: Confirmed mapping for longDescription

10. Select **shortDescription** from the **Target Schema** list
11. Find **shortdescription** in the **Source Schema** tree and drag it onto the **Mapping(s)** list and select it

12. Similar to the description element **shortdescription** is mapped to a **language text element**. Thus, repeat **step 4** and **step 5**. Enter **'.'** For the **text** and **fi** for the **language** parameter.
13. Confirm your mapping by clicking **Confirm** in the **Mapping Details** section
14. Select **Begin Date** in the **Target Schema** list
15. Find **date_start** in the **Source Schema** tree and drag it onto the **Mapping(s)** list
16. The **date_start** element is mapped to a Harmonise **date** element. The corresponding mapping pattern is the **hto:yyyymmdd-to-hto-date** pattern. Click on it and make sure the value for the **date** parameter is set to **'.'** Since the original value should be inherited during the transformation.
17. Confirm your mapping by clicking the **Confirm** button in the **Mapping Details** section
18. Select **End Date** from the **Target Schema** list
19. Locate **data_end** in the **Source Schema** list and drag it onto the **Mapping(s)** list
20. Repeat **step 15** and confirm your mapping by clicking the **Confirm** button in the **Mapping Details** section
21. Select **Day Times** in the **Target Schema** list
22. Locate and expand the **opening_time** element in the **Source Schema** tree
23. Drag and drop **mon**, **tue**, **wed**, **thu**, **fri**, **sat**, and **sun** onto the **Mapping(s)** list.
24. Select **mon** from the **Mapping(s)** list and map it to a Harmonise **day time element**. The corresponding mapping pattern is called **hto:DayTimes**. **htoDayTimes** has three parameters: **day**, **begin**, and **end**. Enter **'.'** for the **day** parameter, select **start** from the drop-box for the **begin parameter**, and finally **end** from the drop-box for the **end parameter**. Again the '.' value guarantees that the original value will be kept during the transformation.
25. Confirm your mapping by clicking the **Confirm** button
26. Repeat **step 23** and **24** for the remaining days (**tue**, **wed**, **thu**, **fri**, **sat**, and **sun**).

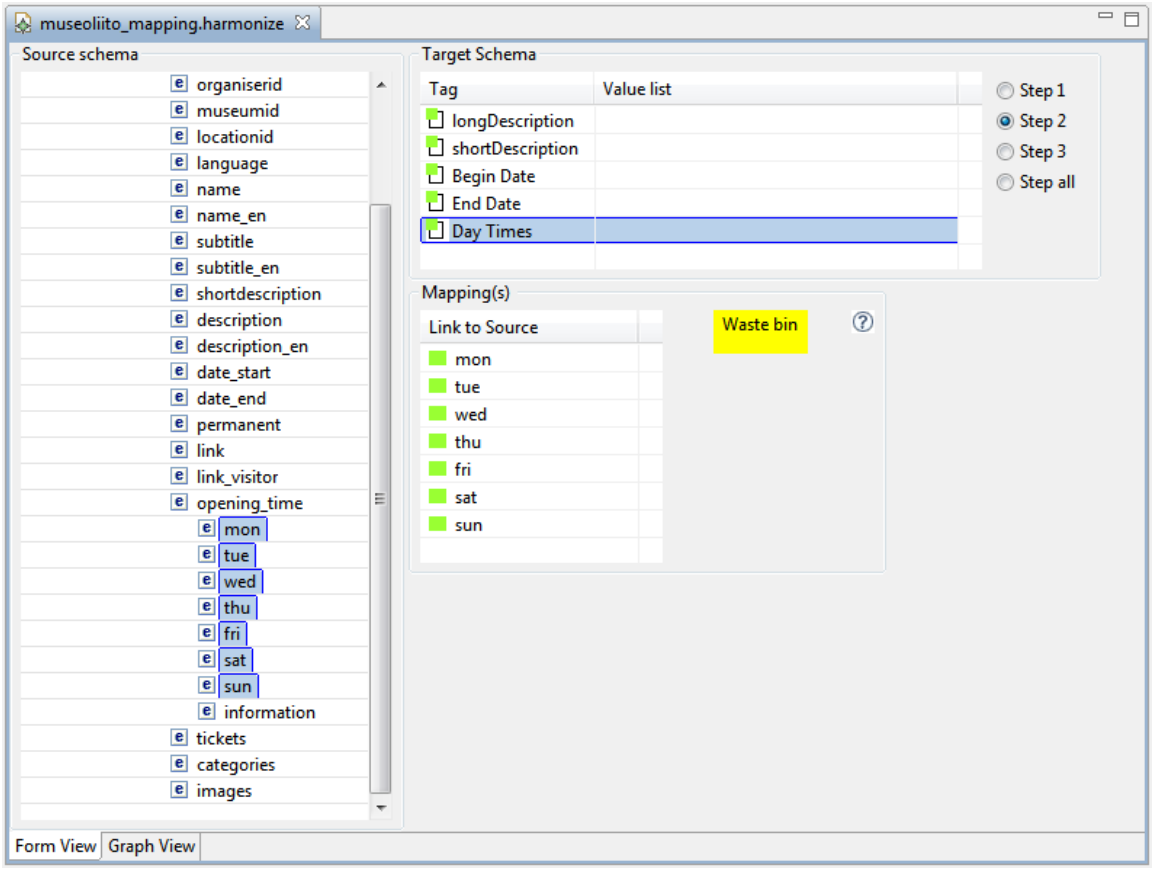


Figure 33: Final mapping for the Day Time element

Additional information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)

4.4.3 Mapping of rich event meta-data

The final step, as defined by the mapping filter, addresses the mapping of rich meta-data, such as images or links.

Filter (Harmonise schema)	Museoliitto schema
Link and images	• link • link_visitor • images

Link and link_visitor are common HTML links and will be mapped to a Harmonise multimedia item. The multimedia item has three distinct parameters: url, mmiType, and language.

The images element will be mapped to a Harmonise image item using the hto:imageItem mapping pattern. The pattern has four parameters: file, mmiType, description, copyright, and language. For more information on the Harmonise image element see the Harmonise Ontology Manual.

Note, the mapping of the categories element is described in Section 5.1 and will not be further explained here.

The remainder of this section provides a step-by-step guidance on how to create and configure the actual mapping links between the recommended target elements in the Harmonise schema and the proper source elements in the Museoliitto schema.

Step-by-step guidance:

1. Select **Link and Images** from the **Target Schema** list
2. Add the source elements **link**, **link_visitor**, and **images** to the **Mapping(s)** list
3. **Link** and **link_visitor** are mapped to Harmonise **MultiMedia** elements whereas the **images** element is mapped to an Harmonise **ImageItem** element.
4. Click on **link** in the **Mapping(s)** list and select the **hto:MultiMediaItem** mapping pattern from the **Mapping Details** list. The pattern has three parameters: **url**, **mmiType**, and **language**. Enter **'.'** for each pattern and click **Confirm**.

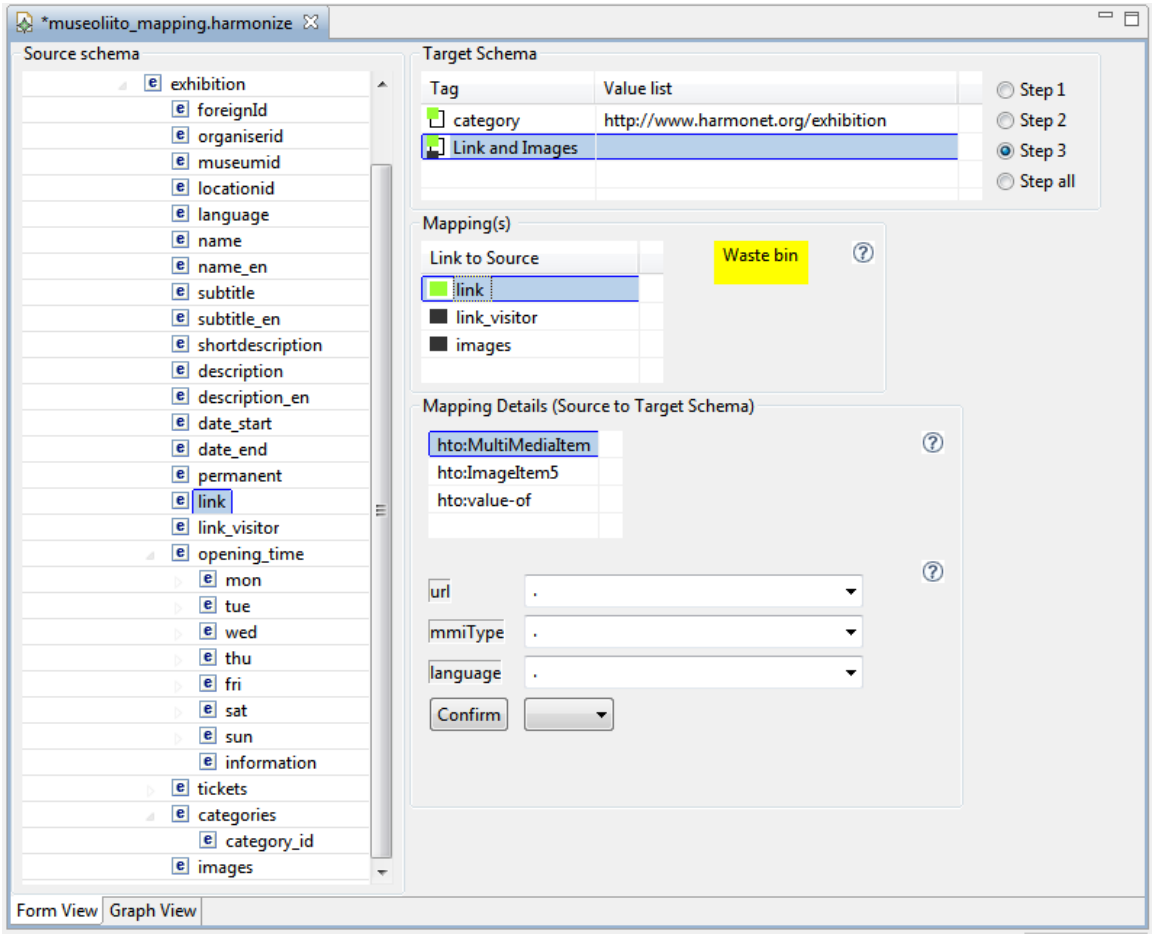


Figure 34: Mapping of link and link_visitor to Harmonise multi-media item elements

5. Select **link_visitor** and repeat **step 11**.
6. Next select the **images** element from the **Mapping(s)** list
7. Select **hto:ImageItem5** from the **Mapping Details** section. The **hto:ImageItem** mapping pattern has 4 parameters: **file**, **mmiType**, **description**, **copyright**, and **language**. Enter **'.'** for each **parameter** and click **Confirm**

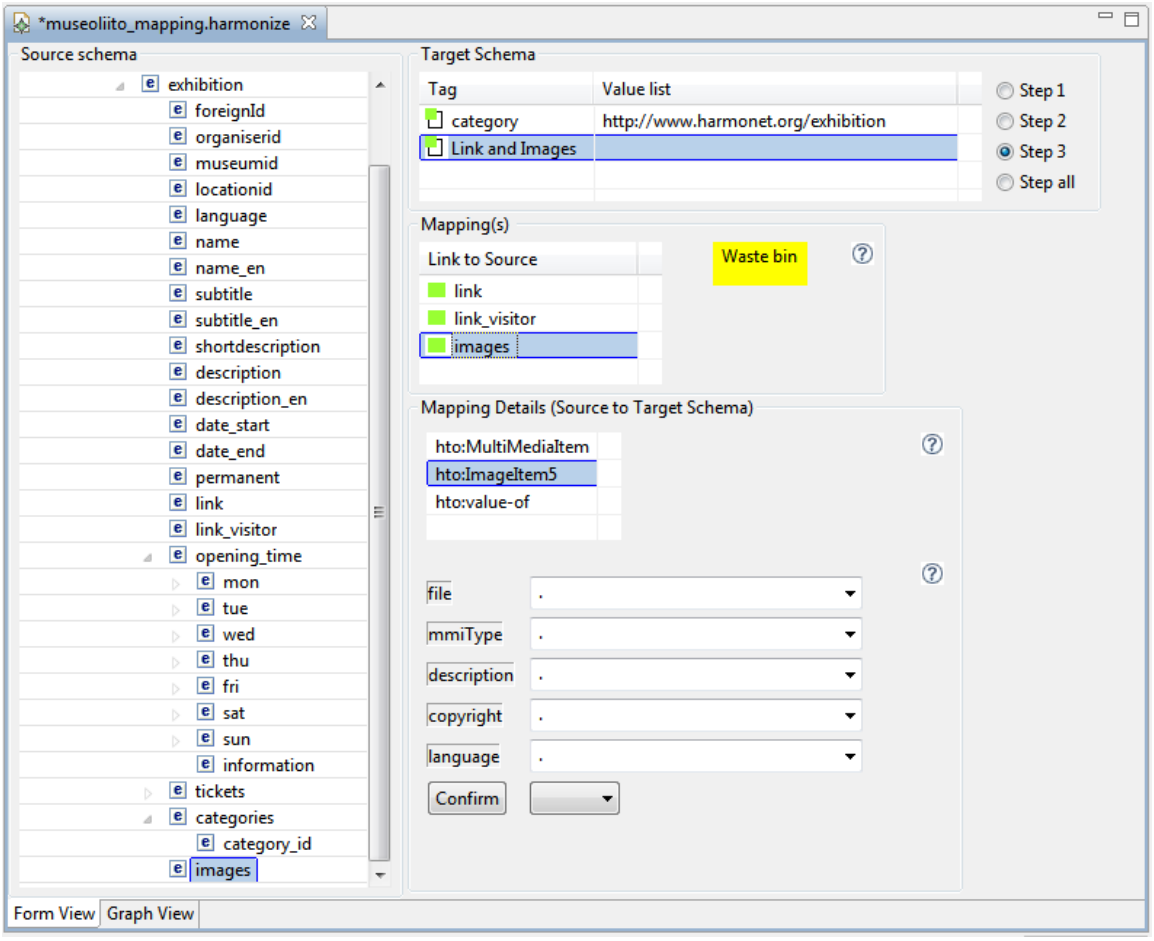


Figure 35: Mapping the images element

Additional information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)
- Value Mapping Editor (see Section 6.4.2)

4.5 RUNNING THE TRANSFORMATION

This section details the tasks and activities necessary to trigger the transformation of the finalized mapping. The main activities that have to be performed by the user are:

- Adapt the process.xml to include instance data and transformation patterns
- Run the actual transformation and create the corresponding output files. Output files are stored in the mapping project's Output folder.
- Visualize the output in the Feedback View.

Step-by-step guidance:

1. Open the **process.xml** file located in the **Input** folder of the **Museoliitto** mapping project.
2. Find the **<In>** tag, replace the text between **<In>** and **</In>** with **Input/museoliitto.xml**. This will set the transformation input to the actual instance data which is provided by the **museoliitto.xml** file.
3. Save your changes by clicking **File→Save** in the main menu or the button in the main toolbar.
4. Close the **process.xml** file by clicking in the editor tab.
5. Expand the **Mapping** folder in the **Museoliitto** mapping project and open the **museoliitto_mapping.harmonize** mapping file by a **double-click**.
6. Click on **Step all** to display all recommended Target Schema elements

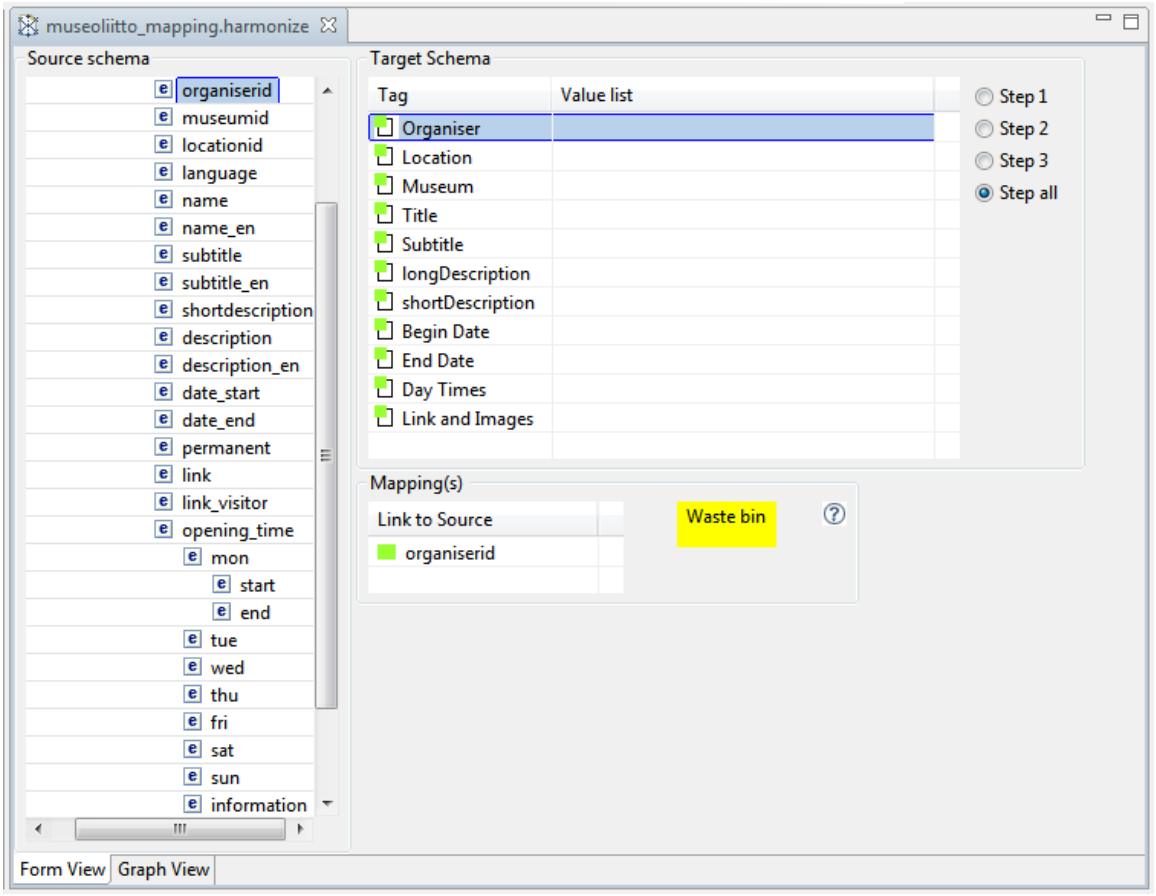


Figure 36: List of target schema elements as recommended by the provided mapping filter

7. Click on the (yellow) button  in order to run the transformation. The generated output is automatically saved in **Output** folder of the **Museoliitto** mapping project.

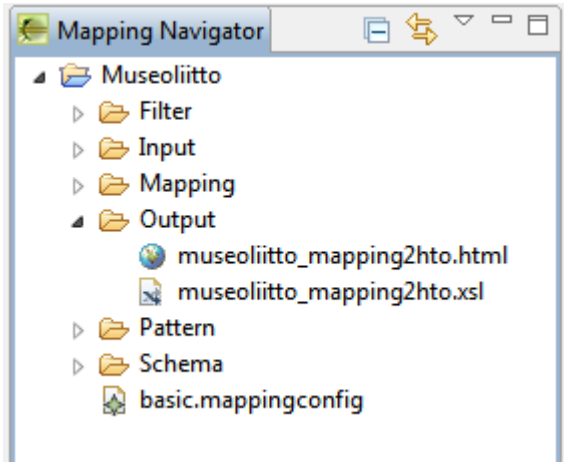


Figure 37: The generated output files

8. In case the Feedback View has not opened yet click **Window→Open View→Feedback View** and review the result in the Feedback View

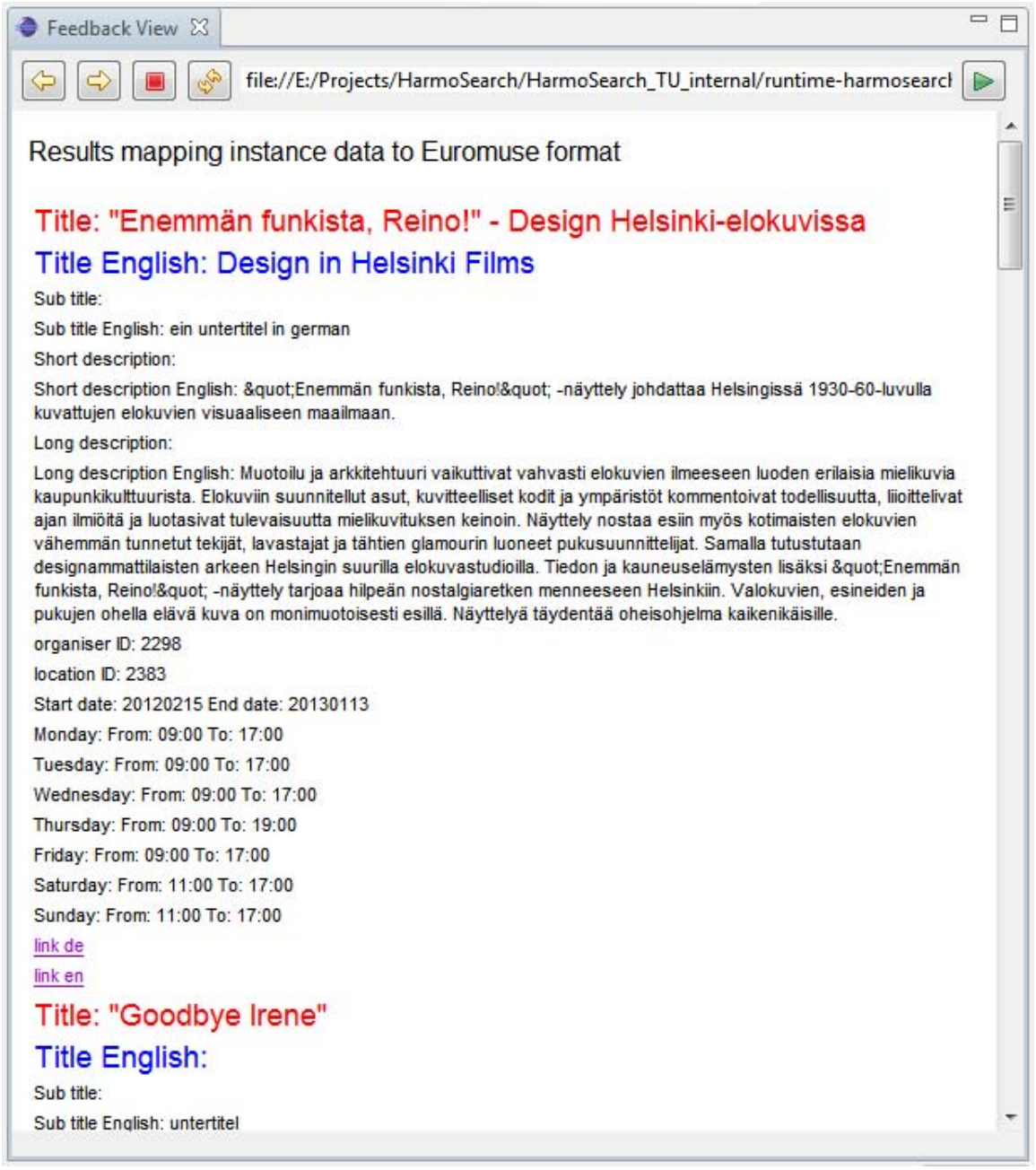


Figure 38: Final transformation result based on the input-data provided by the museoliitto.xml file

Additional information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)
- Feedback View (see Section 6.3.2)

4.6 UPLOAD THE MAPPINGS TO THE HARMOSEARCH PORTAL

This section explains how the mappings can be directly uploaded to the HarmoSearch portal from the mapping tool itself, instead of doing it manually through the Mapping Store user interface of the portal.

Step-by-step guidance:

1. You must have completed the mapping and validated it, namely generated the output by clicking on the (yellow) button  that runs the transformation.

2. Click on **Setting** and select **Upload to Portal**

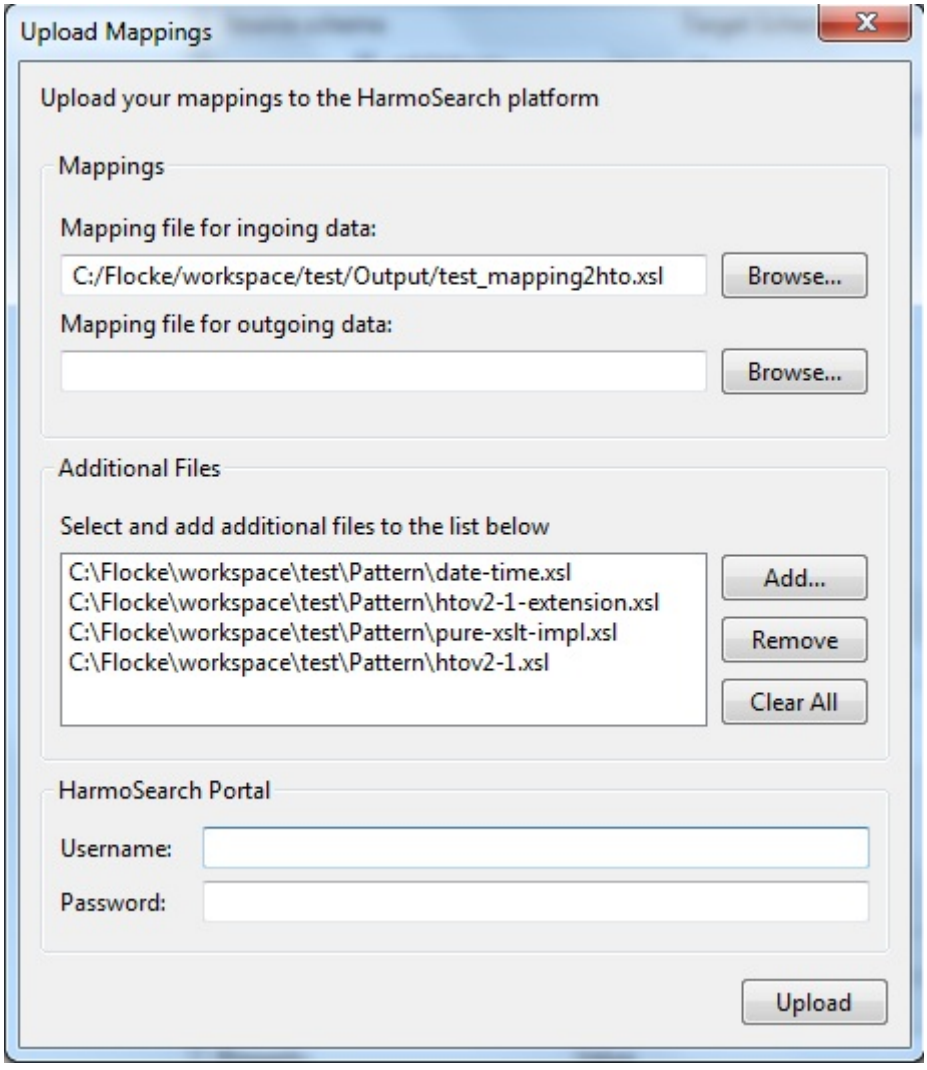


Figure 39: Upload to HarmoSearch portal Dialog

3. The mapping file is already pre-selected, but you can also select it by clicking on **Browse**

4. Additional files can be uploaded together with the mapping file, if needed. It is the case of the patterns that are used by the mapping file. On the HarmoSearch portal they correspond to the term *support libraries.*

5. Insert the username and password of your account on the HarmoSearch portal (you must have a valid user account in order to upload your own mapping)

6. Click on **Upload**

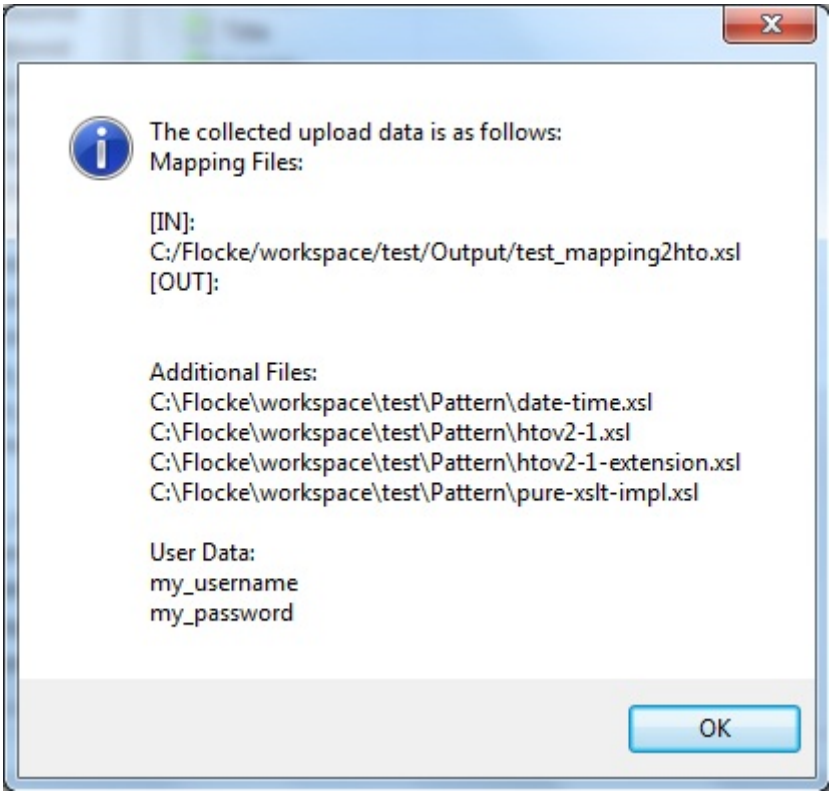


Figure 40: Confirm upload mapping to portal

7. Confirm the upload by clicking **OK**

5 HARMOSEARCH MAPPING TOOL GUIDELINES

This section deals with common tasks and functions a user will encounter when using the HarmoSearch Mapping Tool. The content complements the tutorial introduced in the previous chapter providing additional information about:

- Mapping of reference lists
- Defining mapping variables
- Importing an existing mapping project
- Export an existing mapping project
- Install a third party plug-in
- Update the HarmoSearch mapping tool

5.1 MAPPING OF REFERENCE LISTS

This section explains how dedicated reference values can be mapped and combined with an existing data mapping. Reference values are certain types of information that refer to elements that are not part of the data schema respectively the instance data. Instead these elements refer to data pieces that for example only exist in the database of an individual customer. However, these elements have to be provided in order to connect these elements with the information provided by the data schema.

A typical example of reference values in the domain of cultural heritage is the categorization of a cultural event, e.g., modern art or classic. These classifications depend on the individual museum and (usually) do not follow an official standard. However, in order to map cultural events (as described in the Museoliitto mapping scenario for example) it becomes necessary to map these custom values and integrate them into the data mapping. Otherwise important information will be missing and the mapping does not lead to the result as it is expected by the domain expert.

Due to these specific characteristics the HarmoSearch Mapping Tool approach separates the mapping of reference lists (value mapping) and data schemas into two distinct parts. Thus, a value mapping can be done independent of the data mapping. Another advantage of this approach is that one and the same value mapping can be used for multiple data mappings. Hence, the effort the user has to spent in terms of time for creating a mapping solution can be significantly reduced.

Creating a value mapping is supported by using the HarmoSearch Value Mapping editor. The Value Mapping editor is able to integrate an arbitrary number of reference lists, which can be added to either as source or target lists. Reference lists can be imported to the editor either from a local or a remote data source. Using the remote approach is especially valuable in case the reference values have to be mapped to standardize solutions that are published online. However, in both cases the reference lists have to comply with the Skosify semantic web standard. Once the reference lists have been imported to the editor the user is able to define the final mapping. The mapping file as created by the HarmoSearch Mapping Tool can then be combined with the data mapping.

The tasks a user has to conduct when creating a value mapping can be summarized as follows:

1. Create a new value mapping file and add it to an existing mapping project
2. Setup the value mapping file by adding the source respectively target reference lists to the value mapping editor.
3. Define the mapping links between the different concepts
4. Combine the data and the value mapping

The following sub-sections details the above described steps and provide an overview about requirements and steps necessary for creating and managing a value mapping. Note, the described guidelines complement the Museoliitto business case. Thus, a fully specified mapping project as it was created and configured in the previous sections will be required. Beside these basic requirements access to the

HarmoSearch portal has to be given. This will be necessary since the value mapping file needs to be uploaded to the HarmoSearch portal.

Note: The described tasks require experience and to some extent, technical knowledge about XML and XSLT. Thus, novice users or users with limited technological knowledge in the mentioned technologies should continue with Section 5.2.

5.1.1 Creating a value mapping file

This section describes how value mappings (mapping definitions dealing with the mapping of reference values) can be added to an existing mapping project.

The following step-by-step guidance describes the tasks and activities necessary to create a new value mapping file in the Museoliitto mapping project. The creation of a value mapping file is done via the Value Mapping File Wizard.

Step-by-step guidance:

1. Call the Value Mapping File Wizard from the main menu by clicking **File→New→Value Mapping File**
2. Select the **Mapping** folder of the **Museoliitto** mapping project
3. Enter **museoliitto_value_mapping.values** into the file name text box

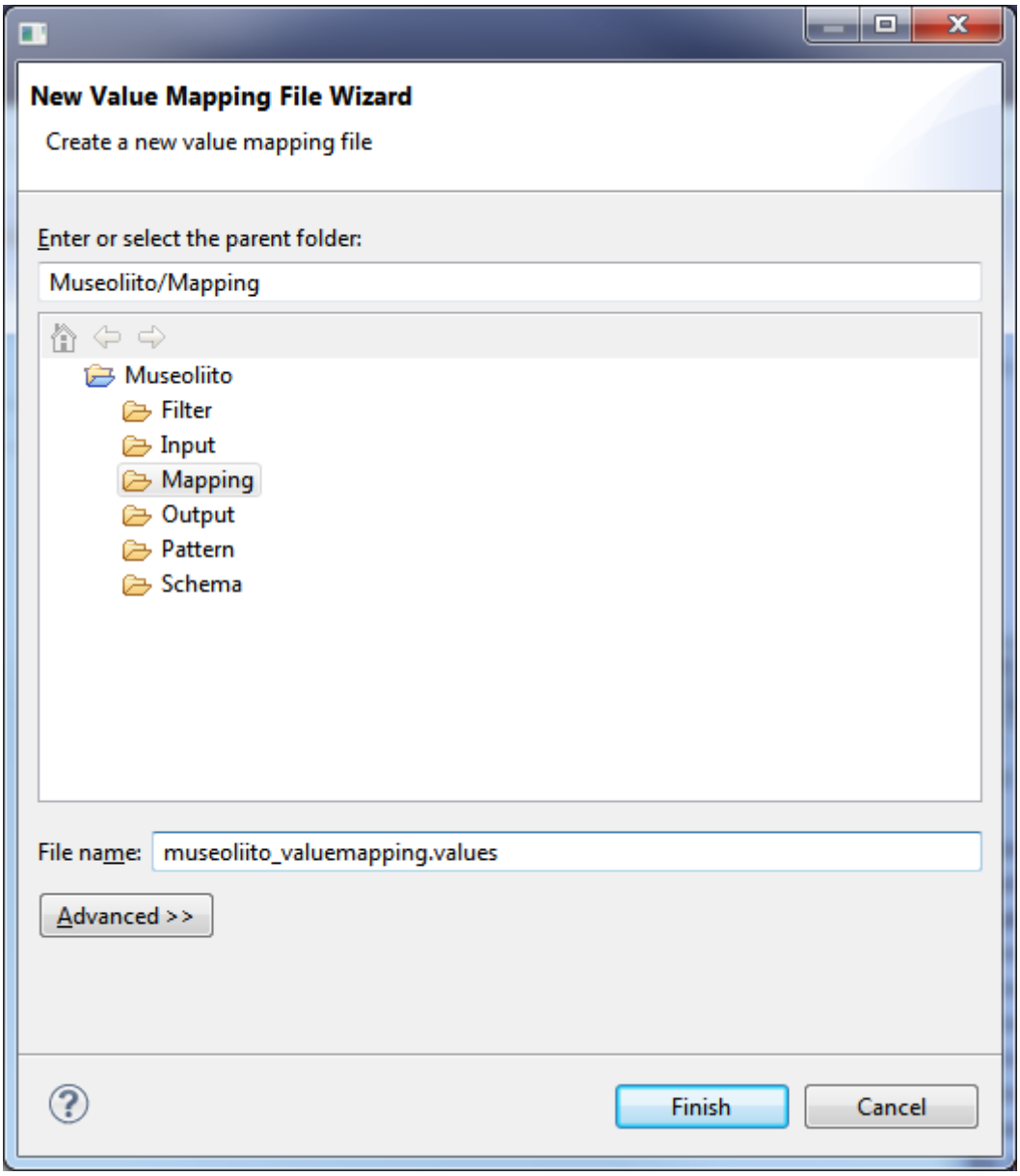


Figure 41: Creating the museoliitto_valuemapping.values file

4. Click **Finish** and check the **Mapping** folder in the Mapping Navigator View. It now contains two files: **museoliitto_mapping.harmonize** and **museoliitto_valuemapping.values**

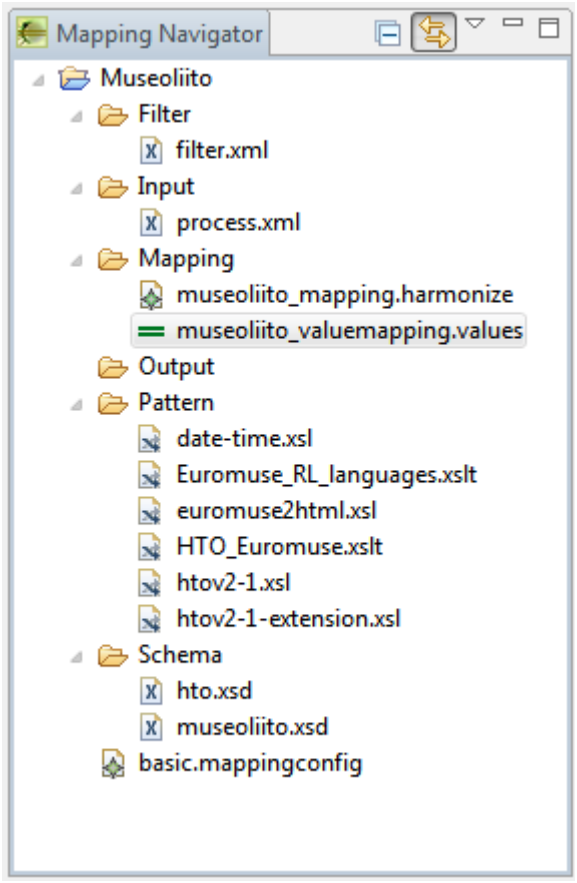


Figure 42: The value mapping file museoliitto_valuemapping.values is added to the Mapping folder

Additional information:

- Value Mapping File Wizard (see Section 6.5.5)
- Mapping Navigator View (see Section 6.3.1)

5.1.2 Defining a value mapping

The following section deals with the tasks and activities that are necessary in order to define the value mapping for our example:

1. Import the reference lists as input files to the mapping project. This step is optional, however, in order to provide a sealed solution the guideline below will include this step.
2. Locate and add the source and target reference lists (Museoliitto and Harmonise)
3. Identify potential mapping candidates in the Museoliitto reference list
4. Create a new mapping link mapping a concept from the Museoliitto source reference list to a concept of the Harmonise target reference list

Step-by-step guidance

1. **Import** the **reference lists** intot he mapping project using the Import Input File wizard (see Section 4.2.4 for more details). Select the **referencelists.zip** file as data source.
    a. As written above this step is not mandatory. The reference lists do not have to exist in the mapping project. However, for the sake of completeness the reference lists are imported.

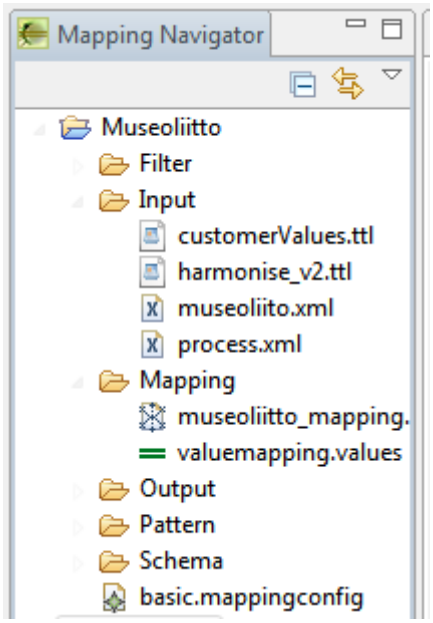


Figure 43: Value lists imoprted to the Input folder

2. Expand the **Mapping** folder of the **Museoliitto** mapping project in the Mapping Navigator
3. Double-click the value mapping file **museoliitto_valuemapping.values** and open the Value Mapping Editor

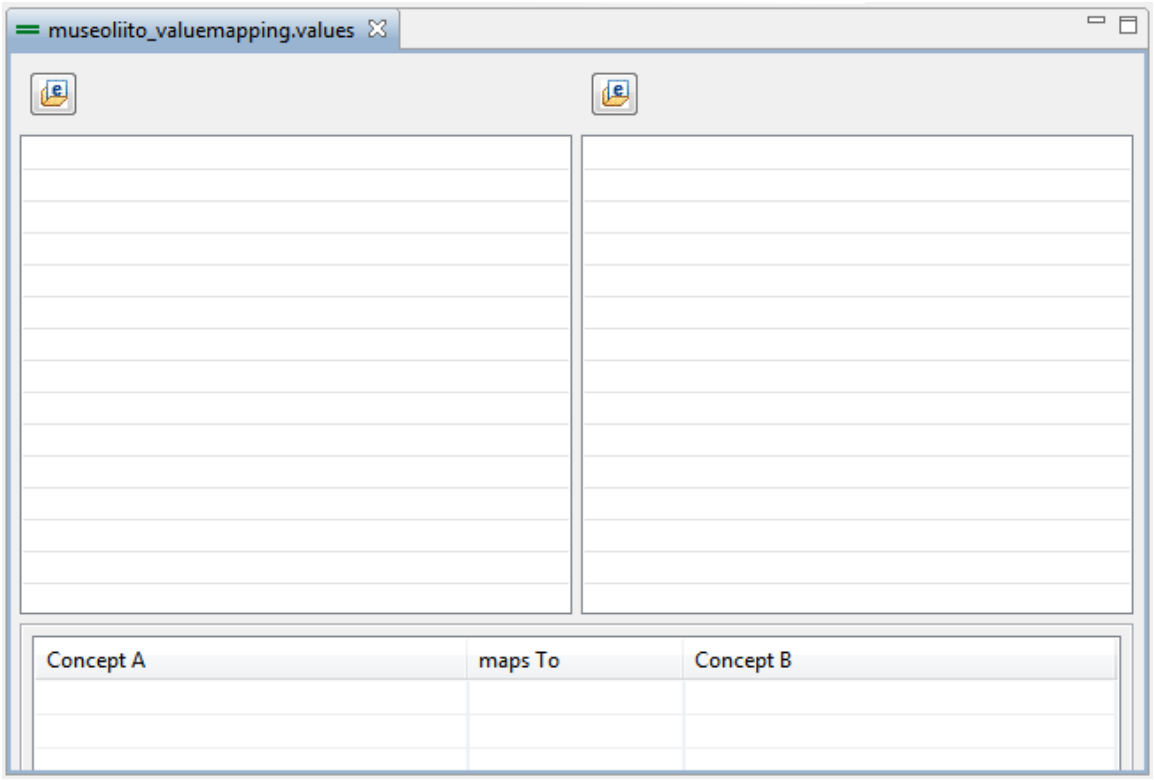


Figure 44: The empty value mapping

4. Importing the **target** reference list is done by clicking the  **button** on the right hand side of the Value Mapping Editor
5. Click **Add Local…**
6. Locate the **harmonise_v2.ttl** reference list file in your local file system and click **Open**

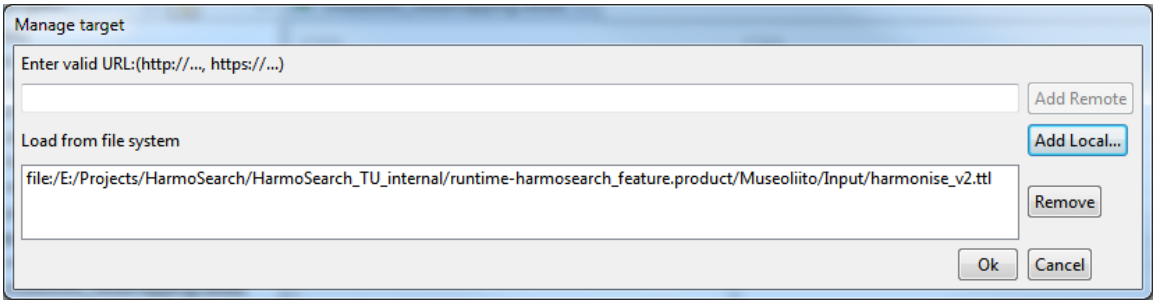


Figure 45: Add the harmonise_v2.ttl reference list as mapping target

7. Click **OK** to add the **harmonise_v2.ttl** reference value list to the Value Mapping Editor
8. Import the **source** reference list by clicking the  button on the left hand side of the Value Mapping Editor
9. Click **Add Local…**

10. Locate the **customerValues.ttl** reference list file in your local file system and click **Open**

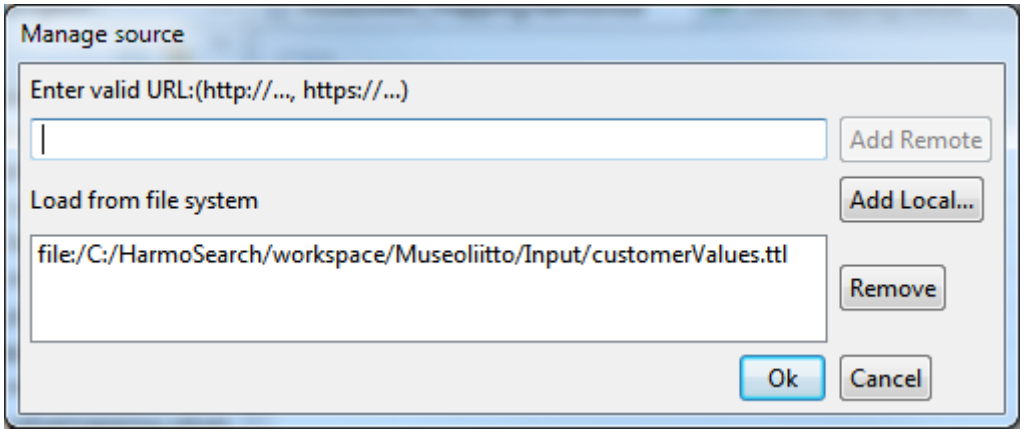


Figure 46: Add the customerValues.ttl reference list as mapping source

11. Click **OK** to add the **customerValues.ttl** reference value list to the Value Mapping Editor
12. Expand the **kuenste** reference list **http://www.tu.at/kuenste** which has been added to the **source area** on the left hand side of the Value Mapping Editor
13. Select the **Tanz** concept
14. Expand the **arts** reference list **http://www.harmonet.org/arts** which has been added to the **target area** on the right hand side of the Value Mapping Editor
15. Find and click the concept labeled **dance**
16. **Left-click** the **dance** concept and select **Create Relation** to add a new mapping link. The mapping link will be displayed in the list below the source respectively target area of the Value Mapping Editor.
17. **Save** your changes by clicking **File→Save** or the  button in the main toolbar

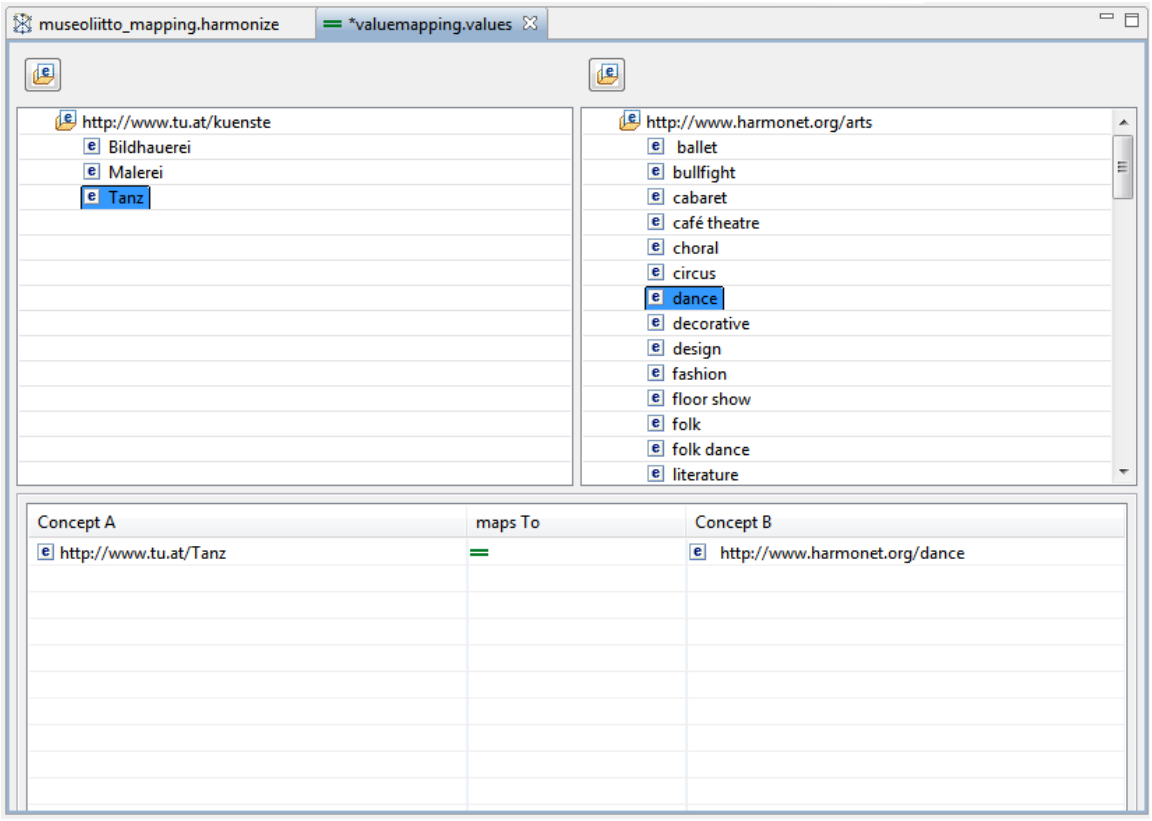


Figure 47: Mapping Tanz to dance

Additional information:

- Value Mapping Editor (see Section 6.4.2)
- Mapping Navigator View (see Section 6.3.1)

5.1.3 Combine value and data mapping

In this section the previously created value mapping will be integrated with the data mapping.

The integration is demonstrated by mapping the category element to the value mapping. This is done by using the hto:SKOSCategory mapping pattern. The pattern has two basic parameters: UniqueID and value. The former is used to identify the correct concept whereas the latter specifies the concrete element that holds the respective value (which is in the end the mapping candidate). Coined to the Museolliito business case the UniqueID will be set to 'http://www.tu.at/' and the value parameter is set to the category_id (sub-element of categories).

After the mapping has been defined additional changes have to be made in order to allow a seamless transformation on the HarmoSearch portal. Mandatory for the generation of adequate transformation code is the availability of (1) a SPARQL endpoint and (2) the transformation output and patterns have to be uploaded to the HarmoSearch portal. Thus the user needs to know two mandatory things:

1. The address (URL) to the HarmoSearch SPARQL endpoint
2. The log-in credentials in order to upload the generated files to the portal

The following guideline details the necessary tasks and provides further details on how to integrate the value mapping into the data mapping. Following the guideline requires explicit knowledge about XML and XSLT. Thus, beginners who feel not confident with the mentioned technologies may continue with Section 5.2.

Step-by-step guideline:

1. Double-click the **museoliitto_mapping.harmonize** mapping file
2. Select **Step 3**
3. Select **category** in the **Target Schema** list.
4. Find the **categories** element in the Museoliitto source schema
5. Drag it onto the **Link to Source** list in **the Mapping(s)** section
6. Select **hto:SKOSCategory** listed in the **Mapping Details** section
7. Type '**http://www.tu.at/**' (**including** the ') into the text field of the **UniqueID** parameter
8. Choose **category_id** from the drop-down box of the **value** parameter
    a. Note, the category_id is a sub element of categories and contains the actual value that has to be mapped. Thus, it must be specified as an input parameter for the hto:SKOSCategory mapping pattern.
9. Confirm your mapping by clicking the **Confirm** button.

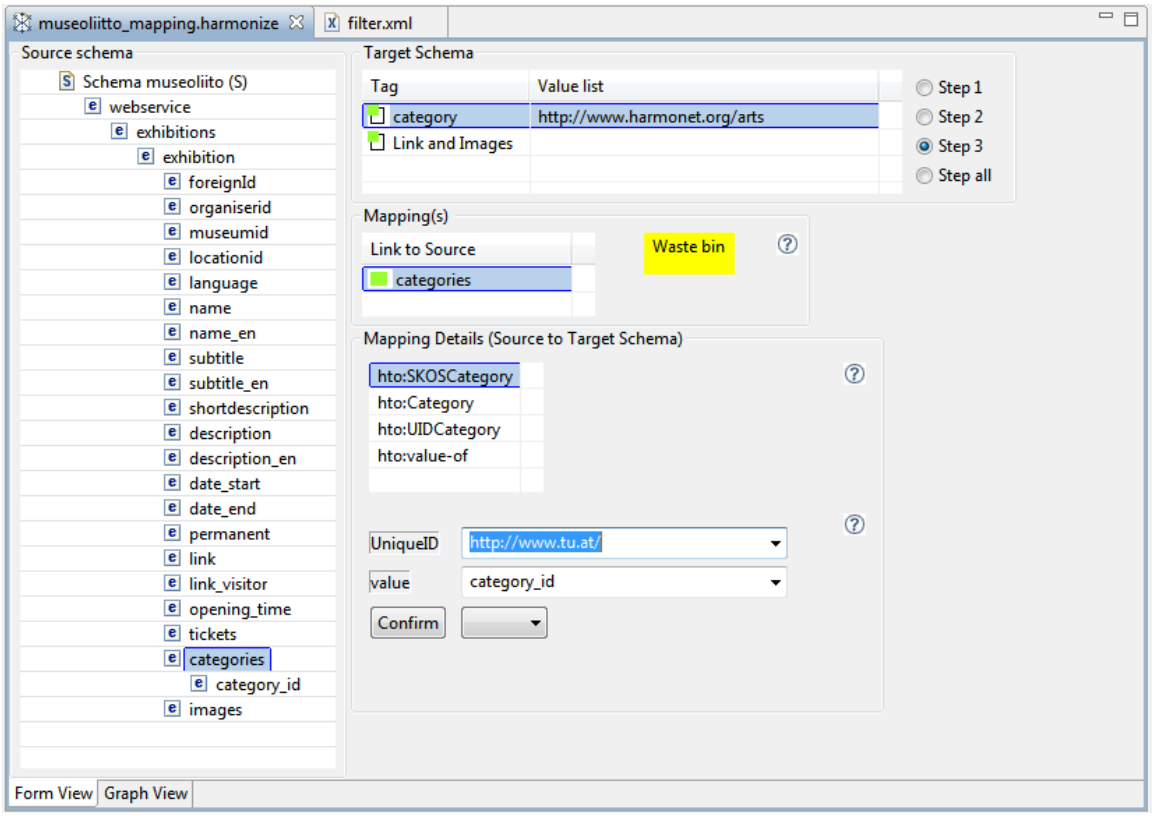


Figure 48: Confirmed value mapping for the category element

10. **Save** your changes by clicking **File→Save** or the  in the toolbar
11. **Right-click** on the value mapping file **museoliitto_valuemapping.values** and select **Copy** from the pop-up menu
12. .**Right-click** the **Output** folder and select **Paste** from the pop-up menu
13. **Right-click museoliitto_valuemapping.values** (Output folder) and select **Rename** from the pop-up menu

14. Keep the file name but **replace .values** with **.ttl**. Do not forget the '.' (without '')! Not renaming the file will result in an error when triggering the transformation procedure on the HarmoSearch portal.
15. Expand the **Pattern** folder in the Museoliitto mapping project
16. Double-click **hto2-1-extension.xsl**
17. In the XML editor locate the **hto:SKOSCategory** mapping pattern by searching for **<xsl:template name="hto:SKOSCategory">**
18. Find the **endpoint** variable **<xsl:variable name="endpoint">** and provide the valid **connection string** under '' pointing to the sparql endpoint. For HarmoSearch, as the endpoint is located on the same machine, you can keep 'http://localhost:3030/sparql' **IMPORTANT:** Make sure the provided URL is encapsulated by '
    a. If you have installed Fuseki the default URI is http://localhost:3030/sparql
19. Find the **query** variable **<xsl:variable name="query">** and find the code peace starting with **SELECT * from <**SOME_URL**>**
20. Replace the SOME_URL fragments with the **URL** pointing to the **value mapping file.** For the HarmoSearch portal you can keep: http://localhost:3030/<values_file_name>.ttl.
21. Click **File→Save** in the main menu to save the changes in the mapping pattern
22. **Double-click** the **museoliitto_mapping.harmonize** mapping file
23. Re-create the transformation output by clicking the **Create and run Transformation** (yellow) button  in the toolbar
24. Close the **Feedback View**
25. **Upload** the output files to the HarmoSearch portal (see Section **Fehler! Verweisquelle konnte nicht gefunden werden.**)

Additional information:

- Value Mapping Editor (see Section 6.4.2)
- Mapping Navigator View (see Section 6.3.1)
- HarmoSearch Mapping Editor (see Section 6.4.1)

5.2 DEFINING MAPPING VARIABLES

This section describes how mapping pattern parameters can be stored and reused during the creation of a data mapping, which is accomplished via the concept of mapping variables.

The main purpose of a mapping variable is to reduce the manual effort for the user when applying a mapping pattern. The mapping pattern, as described earlier, is used to specify how a given source element has to be transformed to match the selected target element. Therefore a mapping pattern usually expects certain input parameters. The parameters represent the actual value that will be used during the transformation and generation procedure. Evidently, specifying the parameter values has to be done manually by the user. Since this, of course depending on the complexity and the number of parameters, can take a serious amount of time it makes sense to reuse the parameter values for similar elements once they have been specified. This will save the user time and at the same time decreases the risk of failure.

In order to provide a reusable way the HarmoSearch mapping tool supports the use of so called mapping variables. From a conceptual point of view a mapping variable is a simple key-value pair representing the variable's name (key) and value. The name of a mapping variable is used to identify the variable in the drop-down box that is used to specify the mapping parameters in the Mapping Details section of the HarmoSearch mapping editor. Once created a mapping variable can be used and applied to any mapping parameter. For identification purposes a mapping variable is always tagged by means of a $ pre-fix.

The remainder of this section complements the basic tutorial which has been introduced in Section 4. The provided step-by-step guidance describes the following:

1. How a mapping variables can be defined and
2. How they can be used as input parameters for a dedicated mapping pattern

As stated above the guideline builds upon the Museoliitto mapping. Thus, the Museoliitto mapping project must exist in the current workspace. Furthermore, it assumes that the configuration steps described in Section 4.2, 4.3, and 4.4 have been accomplished.

Step-by-step guidance

1. Expand the **Mapping** folder of the **Museoliitto** mapping project in the Mapping Navigator view
2. Double-click **museoliito_mapping.harmonize** in order to open the mapping editor
3. Click **Setting→XSL variables** in the main menu or (x)= from the toolbar
4. **Right-click** on the list of variables and select **Insert New Variable**
5. Click on the newly created variable in the **name column** and type **lang_en** into the text field
6. Press **Enter**
7. Click on the **value column** of the newly created variable and type **en** into the text field
8. Press **Enter**
9. Repeat **step 4** to **8** but enter **lang_fi** as **name** and **fi** as **value**

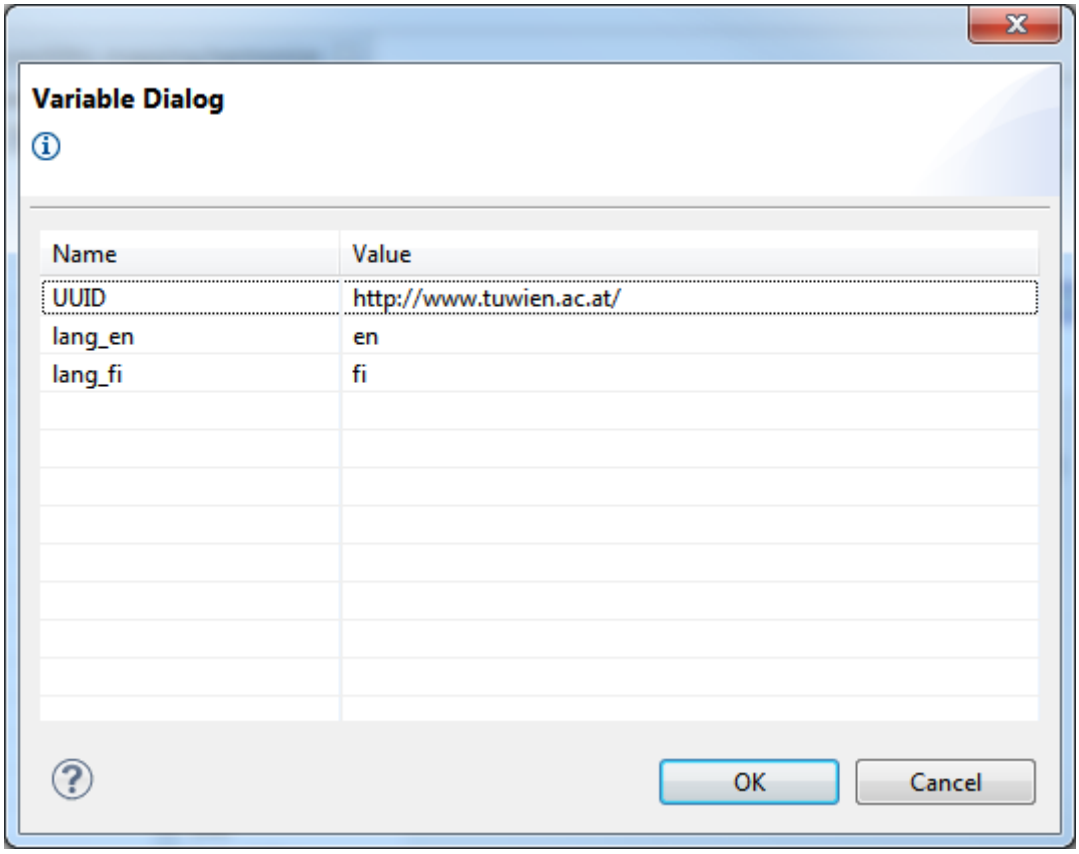


Figure 49: Adding mapping variables using the Variable Dialog

10. Click **OK** to close the Variable Dialog
11. Click **File→Save** in the main menu or click in the toolbar

12. In the mapping editor select **Step 1** from the radio box group in the **Target Schema** section
13. Select **Title** element in the **Target Schema** section
14. Select **name** in the **Mapping(s)** area in order to visualize the **Mapping Details** section
    a. The mapping details are already configured, thus, the hto:languageText mapping pattern has already been selected
15. Select **$lang_fi** from the drop-down box of the **language** parameter
    a. As mentioned before the name element in the Museoliitto data schema denotes to the original language the event is described (in this case Finish)

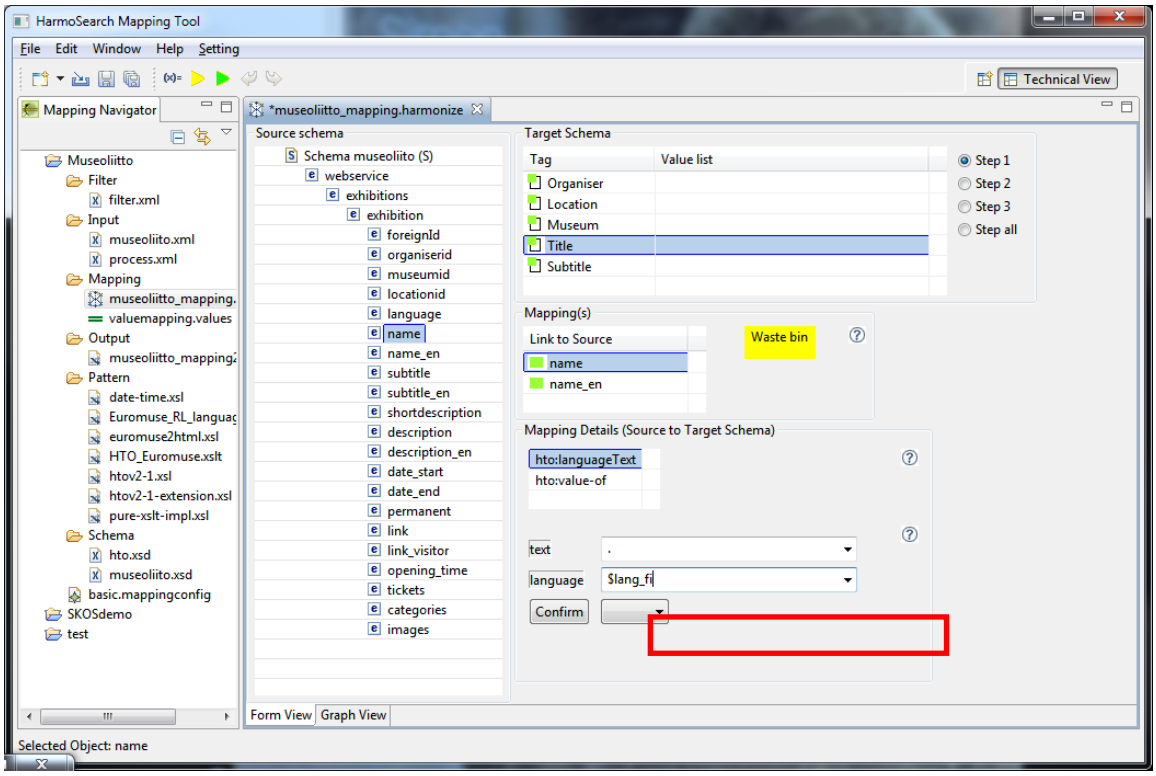


Figure 50: Selecting the $lang_fi variable

16. Click **Confirm** to persist the changes
17. Select **name_en** from the **Mapping(s)** section
18. Select **$lang_en** from the drop-down box near the **language** parameter

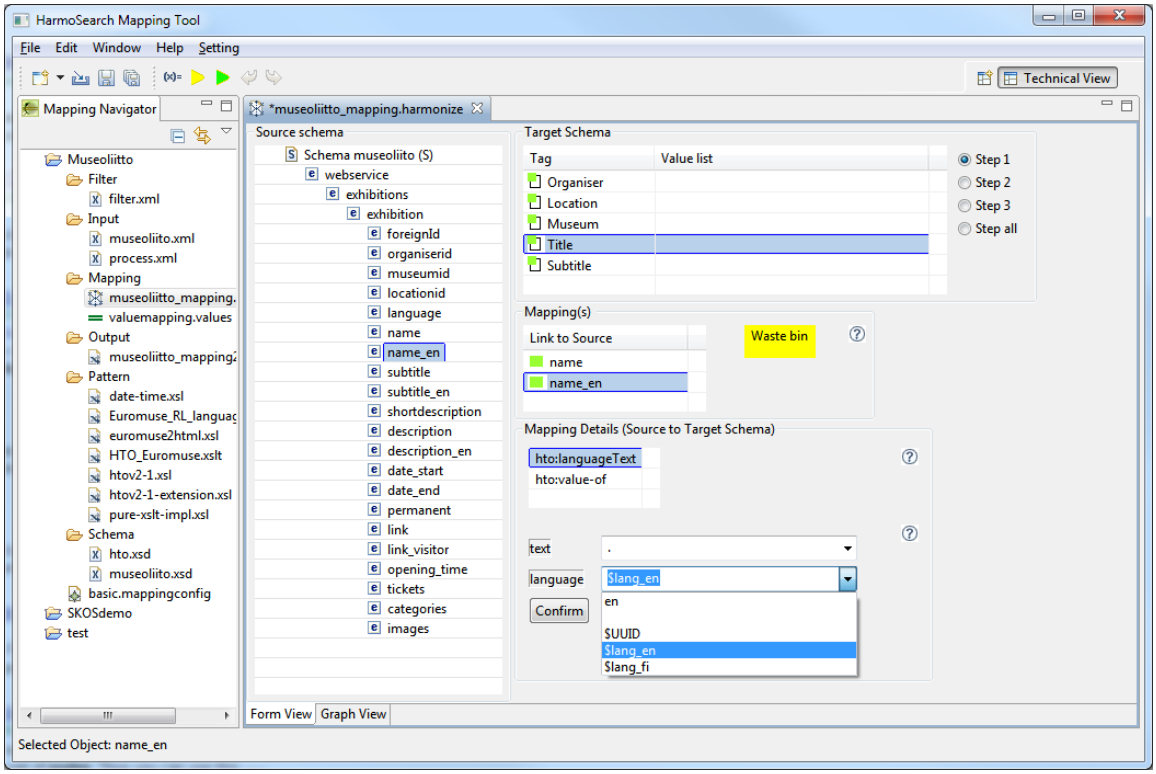


Figure 51: Selecting the $lang_en variable from the drop-down box

19. Click **Confirm** to persist the changes

Additional Information:

- HarmoSearch Mapping Editor (see Section 6.4.1)
- Mapping Navigator View (see Section 6.3.1)

5.3 IMPORTING AN EXISTING MAPPING PROJECT

The HarmoSearch mapping tool provides the ability to reuse existing mapping projects. Projects can be imported as Zip file or from a folder on the local hard-disk.

Reusing existing solutions is of special interest to data integrators who may provide already pre-configured mapping projects their customers may use as an initial stub. The stub project will include all necessary resources and configurations that otherwise the customer had to do. For example:

- A pre-configured mapping file already including the data provider's schema as mapping target
- A mapping filter that provides structured information of recommended target mappings specifically focused on the data integrators schema.
- A pre-configured process specification that defines which output should be generated, e.g., pure XML or HTML code
- Specific sets of mapping patterns defining how distinct elements can be mapped to the data integrator's data schema

The following provides a short step-by-step guidance exemplifying how an import of existing mapping projects can be accomplished. Thereby, the guideline will built upon an existing mapping project that provides an initial mapping stub for the previously described Museoliitto mapping. The mapping project (provided as Zip file) contains the following resources:

- A set of dedicated mapping patterns
- The Harmonise schema file, which has also been registered as target schema in the mapping file
- A pre-configured process configuration
- A mapping filter focused on recommending elements usually used to represent cultural events using the Harmonise ontology

At the end of the guideline a new mapping project will be available in the current workspace.

Step-by-step guidance:

1. Start the HarmoSearch mapping tool. Per default the current workspace should be empty (in case you have not already created or imported other mapping projects)
2. Call the Import wizard from the main menu by clicking **File→Import** in the main menu or from the main toolbar

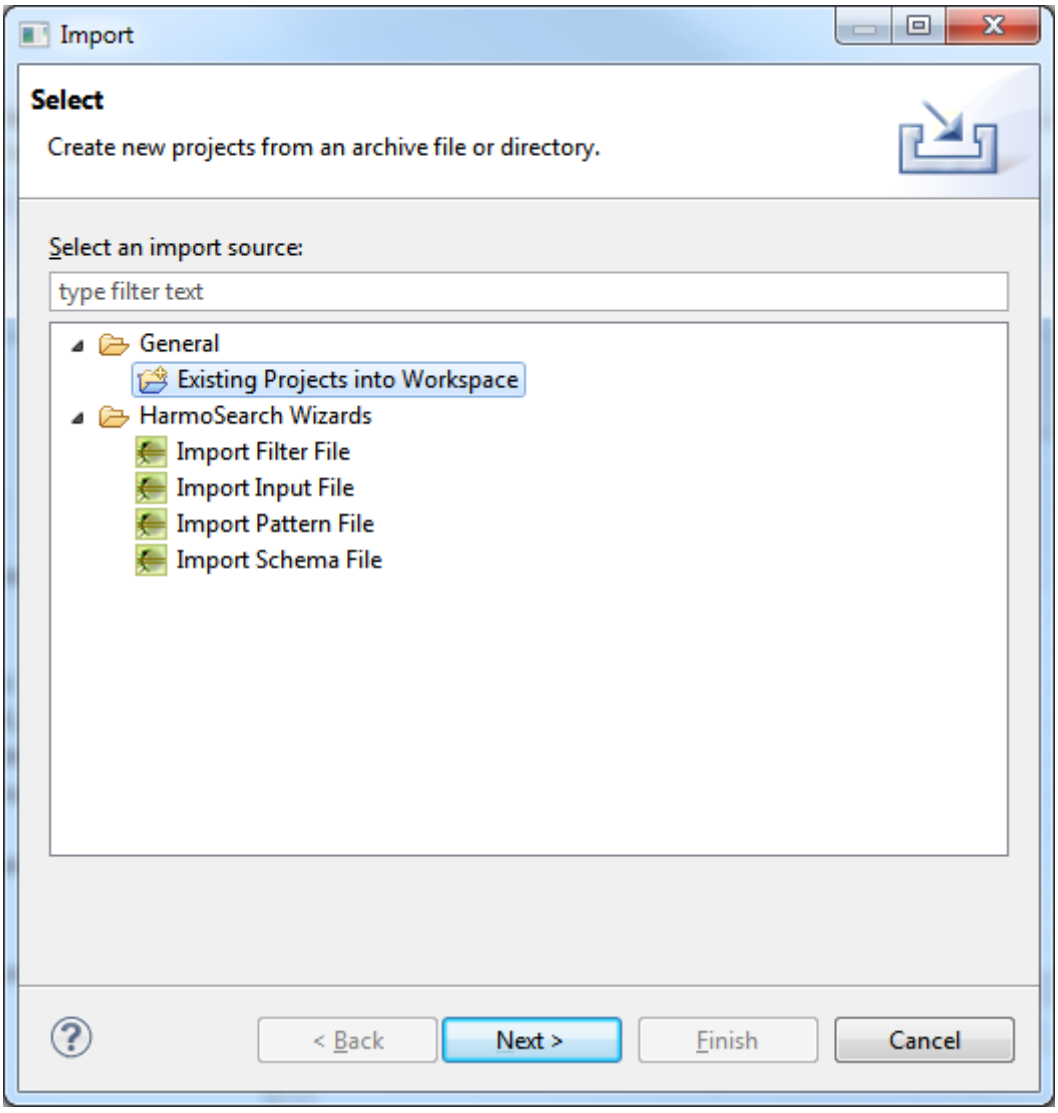


Figure 52: Select Existing Project into Workspace from the Import Wizard dialog

3. The Import wizard dialog lists all available import wizards. Expand General and select **Existing Projects into Workspace**
4. The mapping project stub has been provided as Zip archive. Thus, select **Select archive file** and click on the **Browse…** button.
    a. Alternatively, projects may also be imported directly from the file system. In such a case select **Select root directory** and locate the proper folder containing the target project(s) on your local file system.
5. Locate the archive file containing the mapping project stub and click Open

6. The Zip will be scanned and all available projects are listed. Check the **Museoliitto** project and uncheck (if necessary) **Add project to workingsets**.

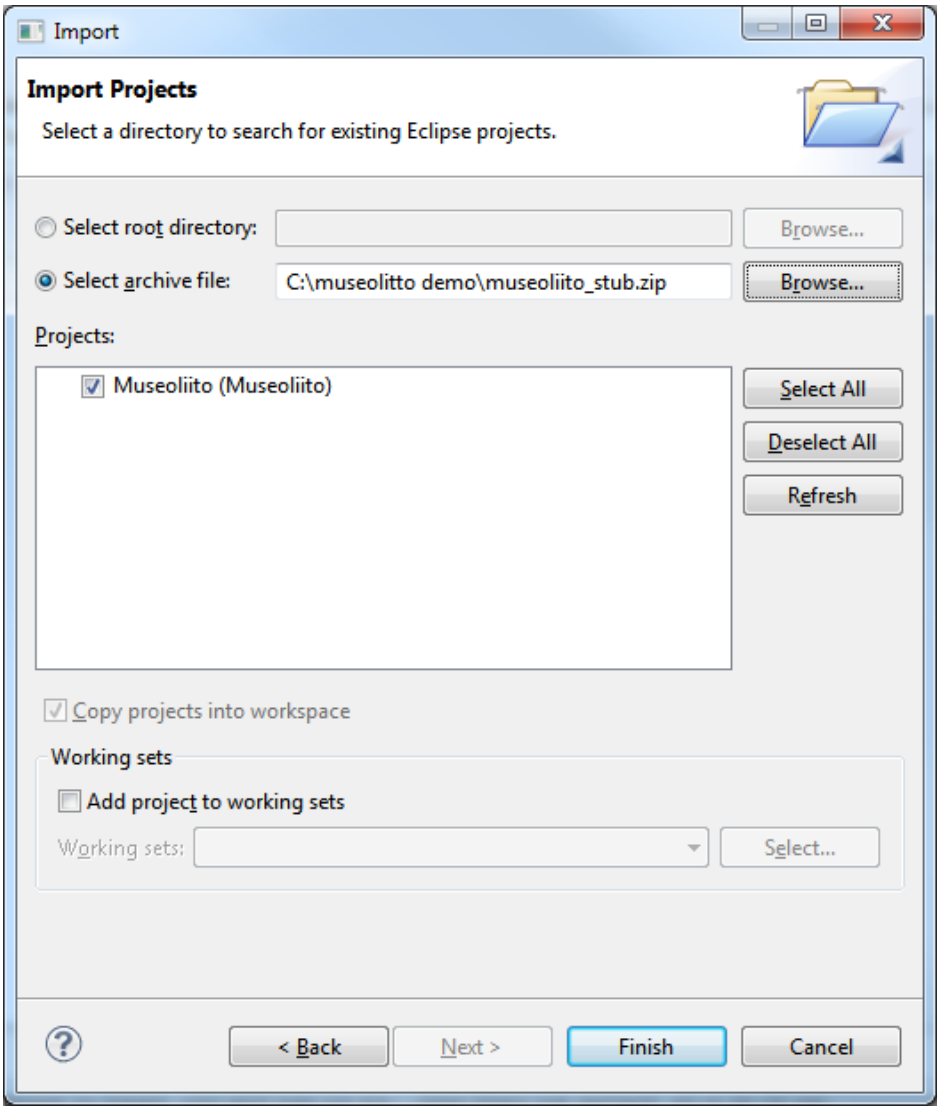
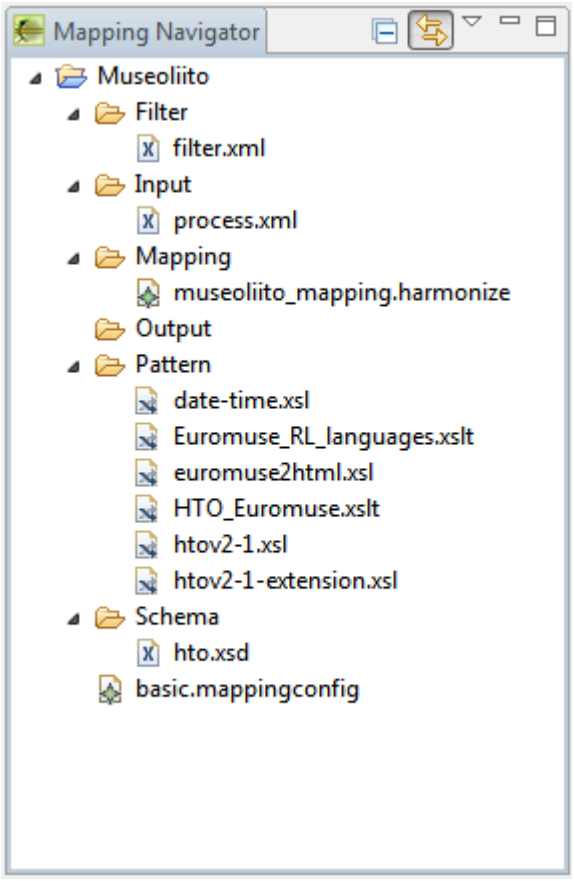


Figure 53: Importing the Museoliitto mapping project stub to the current workspace

7. Click **Finish** to trigger the import procedure that unzips and copies the project into the current workspace.
8. Expand the **Museoliitto** mapping project and its **folders** to gain an overview about the resources that have already been added.



Additional information:

- Mapping Navigator View (see Section 6.3.1)
- Import Wizard (see Section 6.5.7)

5.4 EXPORT AN EXISTING MAPPING PROJECT

Mapping projects that exist in the current workspace can be exported as an archive file. The main advantage of exporting a mapping project as an archive file is that it becomes possible to provide a so called mapping stub to customers (see Section 5.1).

The remainder of this section provides a short step-by-step guidance exemplifying how to export and pack the contents of an existing mapping project into a Zip archive. Again, the previously created Museoliitto mapping project will serve as an example.

The final result will be a Zip archive including the Museoliitto mapping project and its associated resources, e.g., mapping file, pattern files, input files.

Step-by-step guidance:

1. Click on **File→Export** and select Archive File from the Export wizard

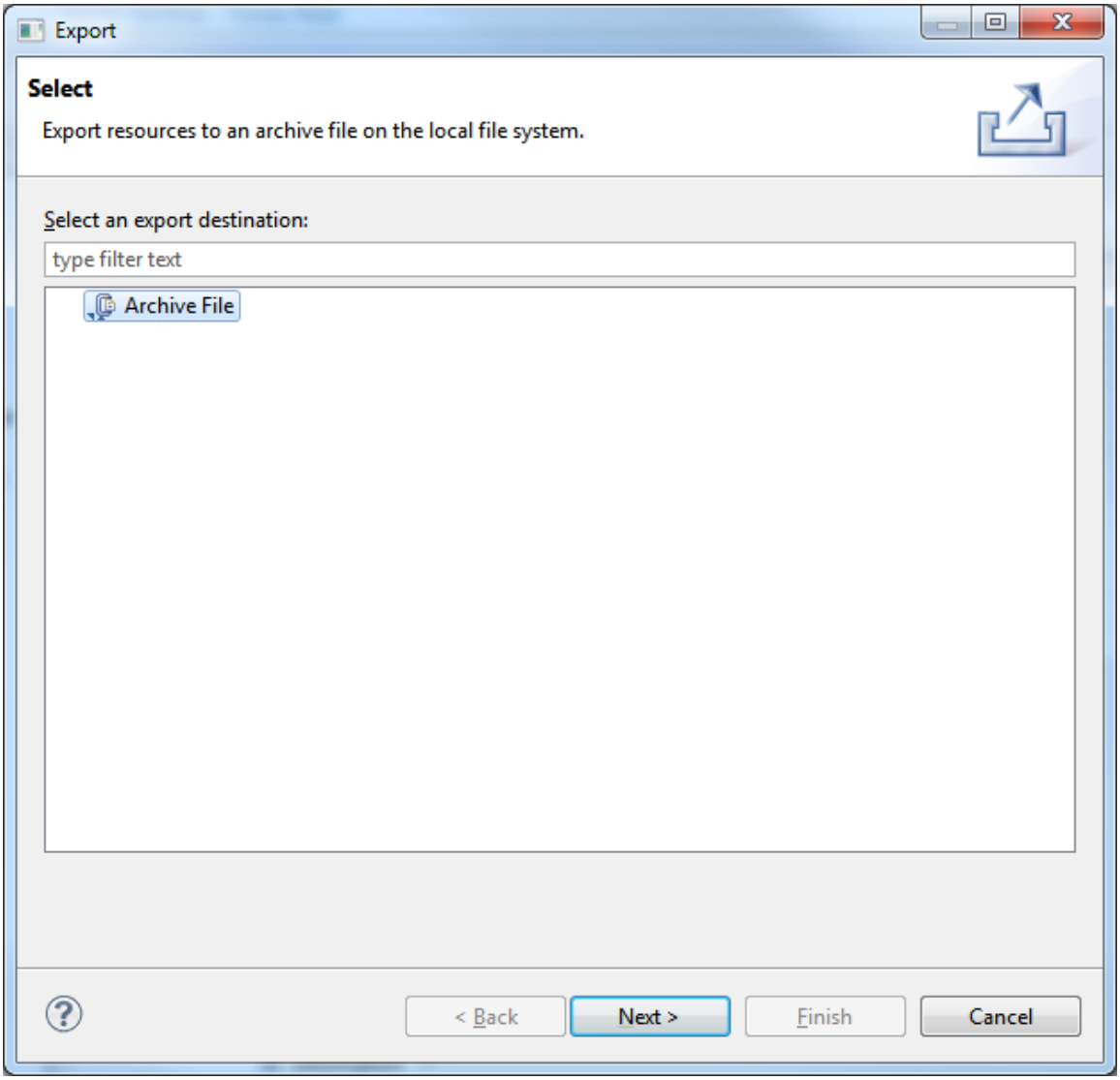


Figure 54: The Export wizard

2. Check the **Museoliitto** mapping project and its associated files
3. Click the **Browse** button and select the location the resulting Zip file will be stored. Furthermore, provide a proper **file name**, e.g., **museoliitto_stub.zip**
4. In the **Options** section select Save in zip format and **Create directory structure for files**.
    a. Optionally you can decide not to create the whole directory structure. If so select **Create only selected directories**.
    b. Optionally you can change the file format to **tar**. To do so select **Save in tar format**.
5. Click **Finish**

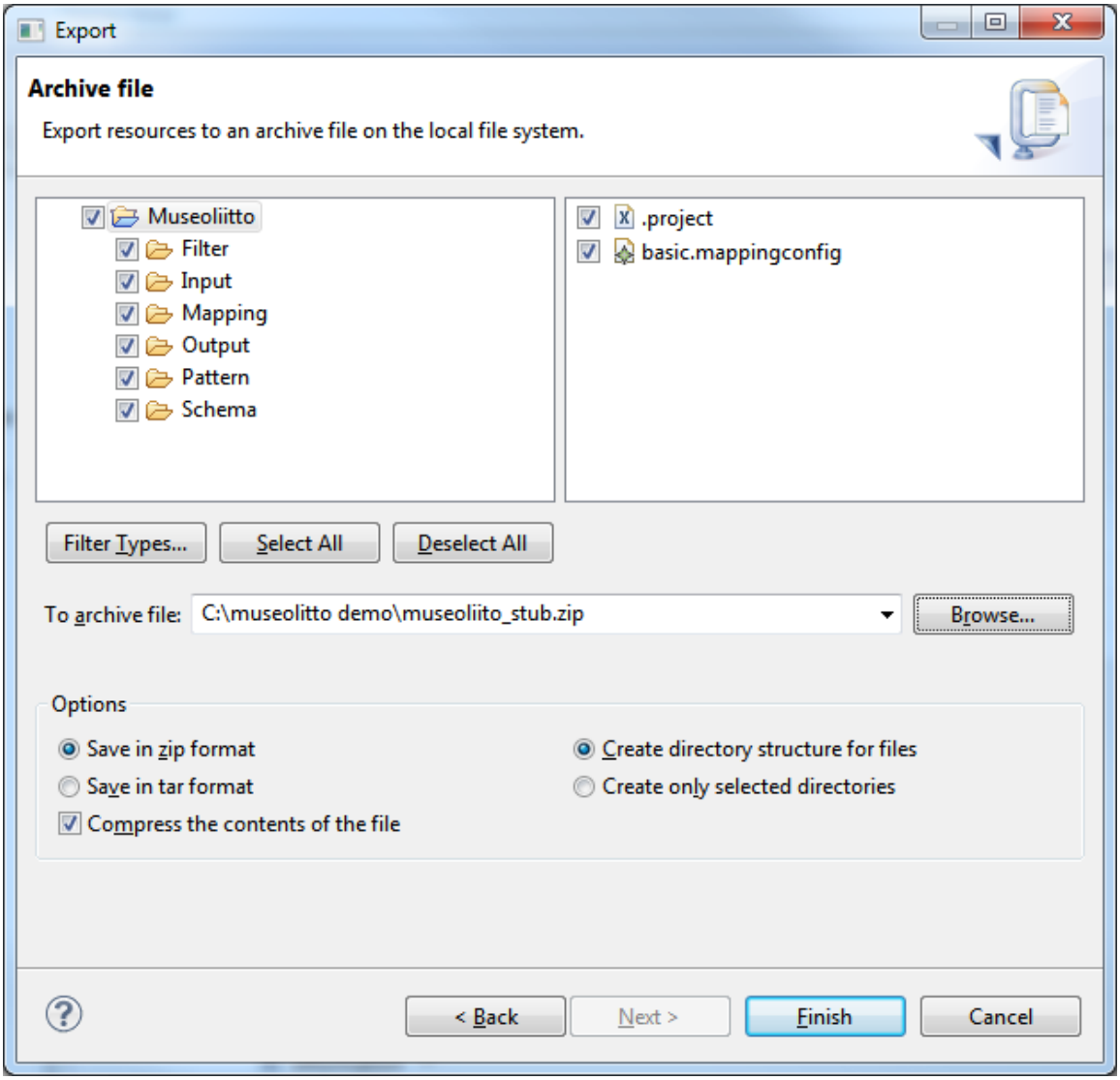


Figure 55: Exporting the Museoliitto mapping project as an archive file

Additional information:

- Mapping Navigator View (see Section 6.3.1)
- Export Wizard (see Section 6.5.6)

5.5 INSTALL A THIRD PARTY PLUGIN

This section shortly describes how to extend the HarmoSearch mapping tool using a flexible and light-weight update and install mechanism. The mechanism allows downloading and installing new software components for the HarmoSearch mapping tool in a user-friendly and minimal invasive way.

Software can be installed either via an online or offline repository. The latter denotes to folders or archives (e.g., Zip or Tar files) located on the user's hard-disk. The common and usual way, however, for installing new software is the online approach.

Step-by-step Guidance:

1. Open the Install Wizard from the file menu by clicking **Help→Install New Software…**
2. Add a new repository by clicking **Add…**
3. Name the new repository and enter the URL of the repository into the location text field
    - Alternative: Set path to local folder by clicking the **Local…** button
    - Alternative: Set path to archive file by clicking the **Archive…** button

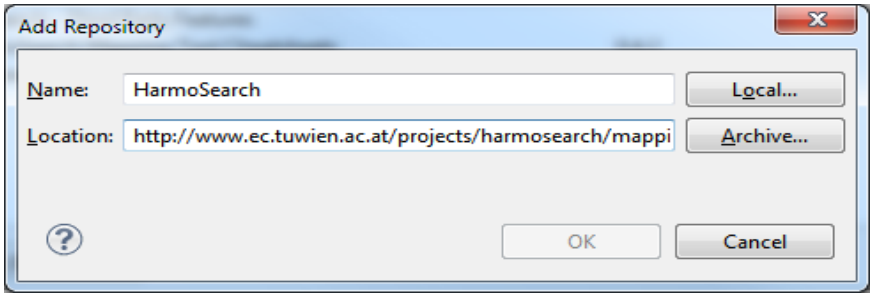


Figure 56: The Add Repository dialog pointing to the HarmoSearch online repository

4. Click **OK**
5. Check the following boxes:
    - **Show only the latest versions of available software**
    - **Group items per category**
    - **Contact all update sites during install to find required software**
6. Select the plugin to install
7. Complete the installation by clicking on **Next**, accepting the license and restarting the tool

Example (install Cheatsheets):

1. Select the **All Available Sites** from the drop-down box (one example repository is already pre-loaded)
2. Expand **HarmoSearch – Additions**
3. Check **HarmoSearch Mapping Tool Cheatsheets**

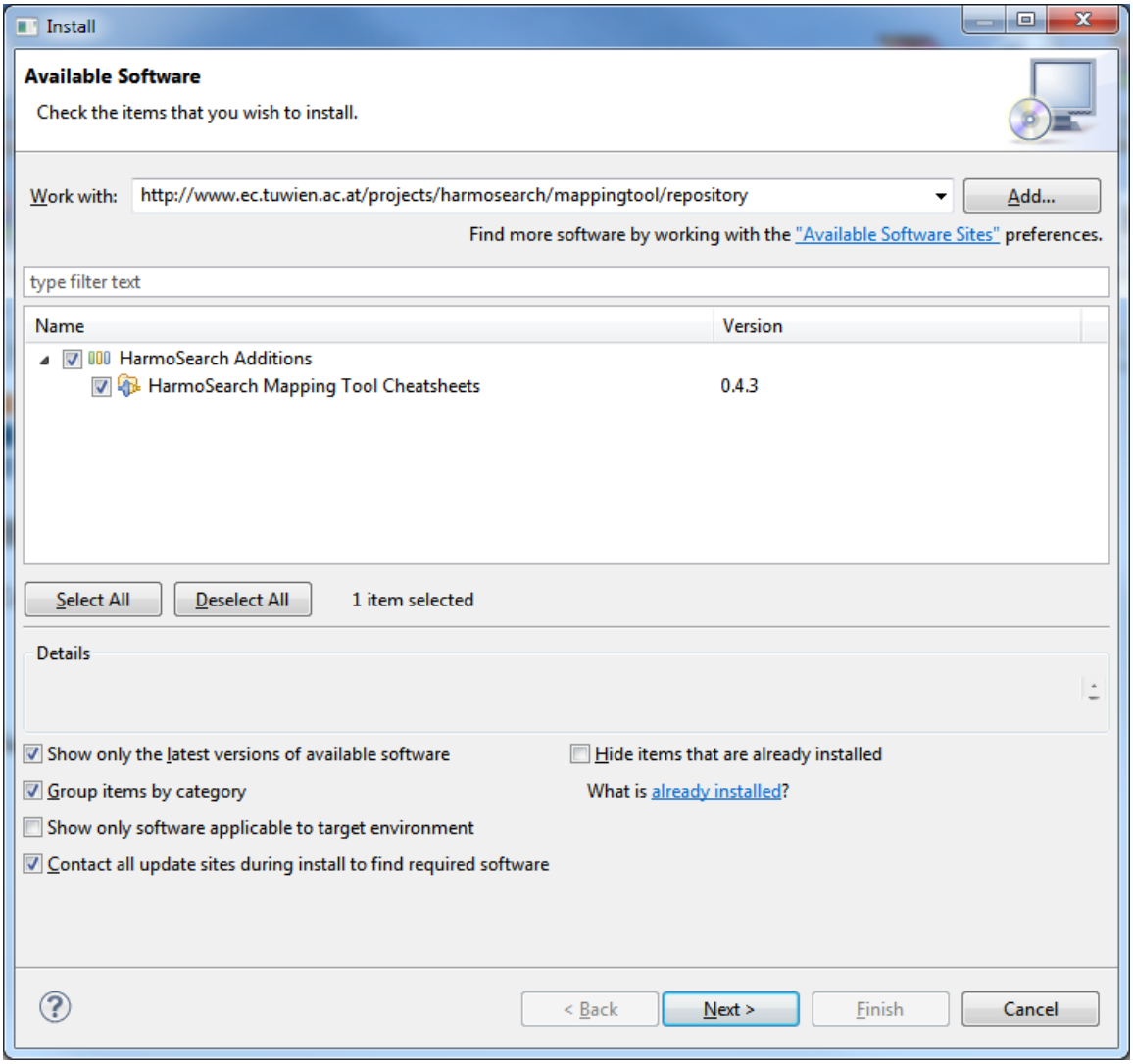


Figure 57: Select the cheatsheet plugin from the HarmoSarch repository

4. Click **Next**

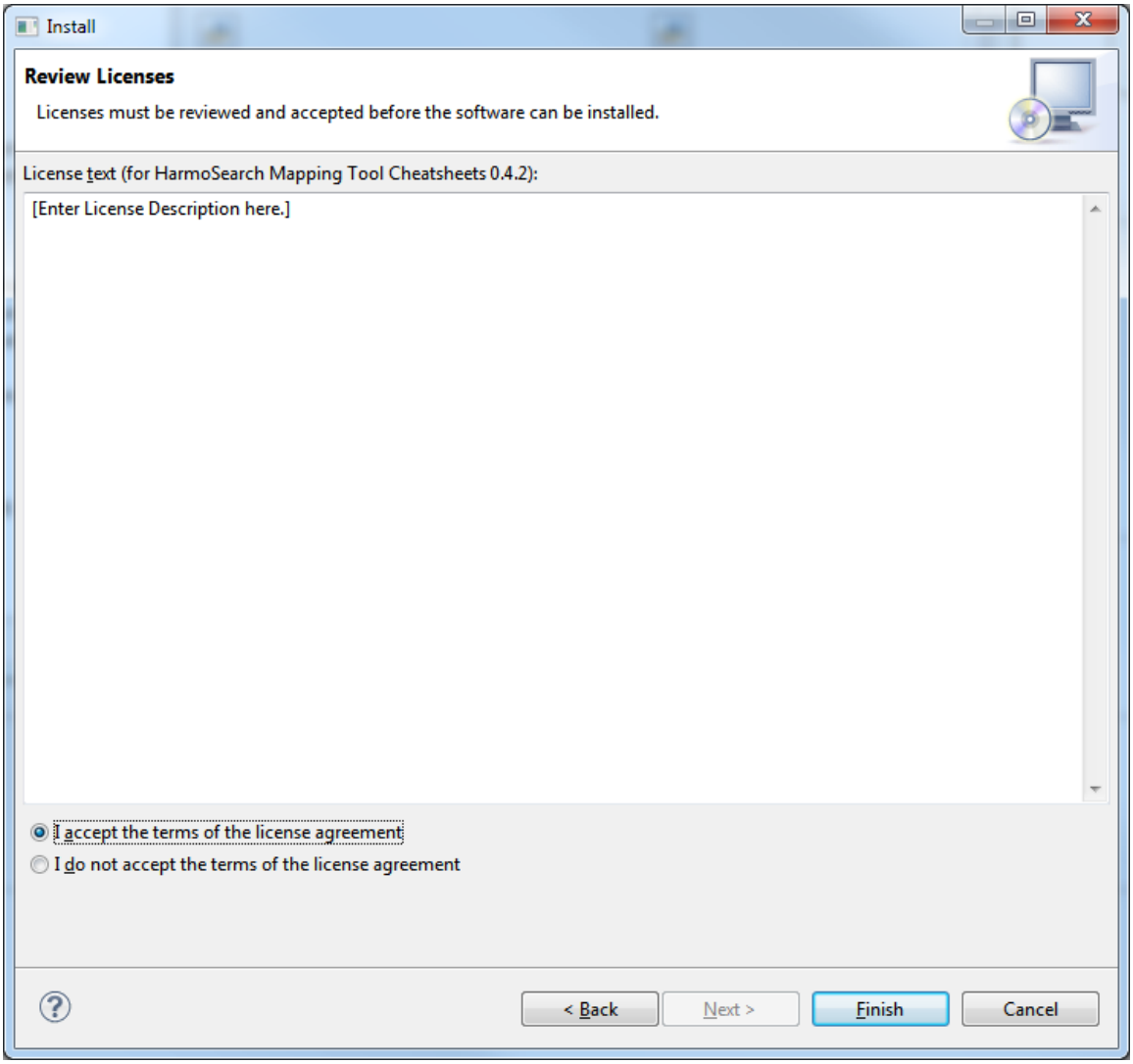


Figure 58: Accept the license agreement

5. Click **Finish** to start the download and the installation
6. Click **OK** in the Security Warning dialog
7. Click **Restart Now**

After restarting the mapping tool the new software can be used. In case of the cheatsheets plugin a new menu item has been added to the Help menu as shown in Figure 59.

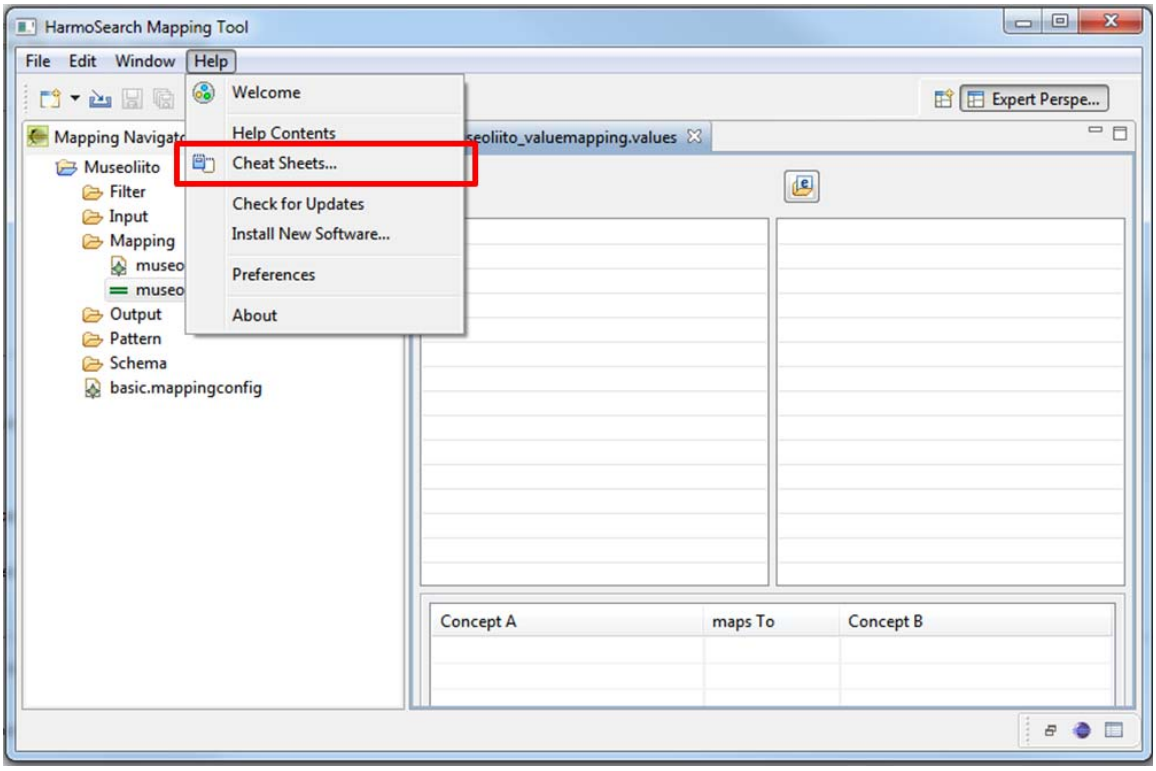


Figure 59: Installed cheatsheet support

5.6 UPDATE THE HARMOSEARCH MAPPING TOOL

The HarmoSearch mapping tool facilitates functionality for updating installed software components in the system.

Updates are identified and installed using the Available Updates dialog. The dialog lists available updates for the installed software components, which can be individually selected by the user. Selected components are automatically updated. Therefore, the wizard downloads and installs the required sources fully automated.

Performing an update requires a stable Internet connection.

Step-by-step guidance:

1. Open the Available Updates wizard by clicking **Help→Check for Updates**
2. Review the list of available updates
    a. If no updates are available at the moment a proper message dialog is shown
3. Check the proposed updates (**HarmoSearch Mapping Tool**) you want to install and uncheck those you do not want to install

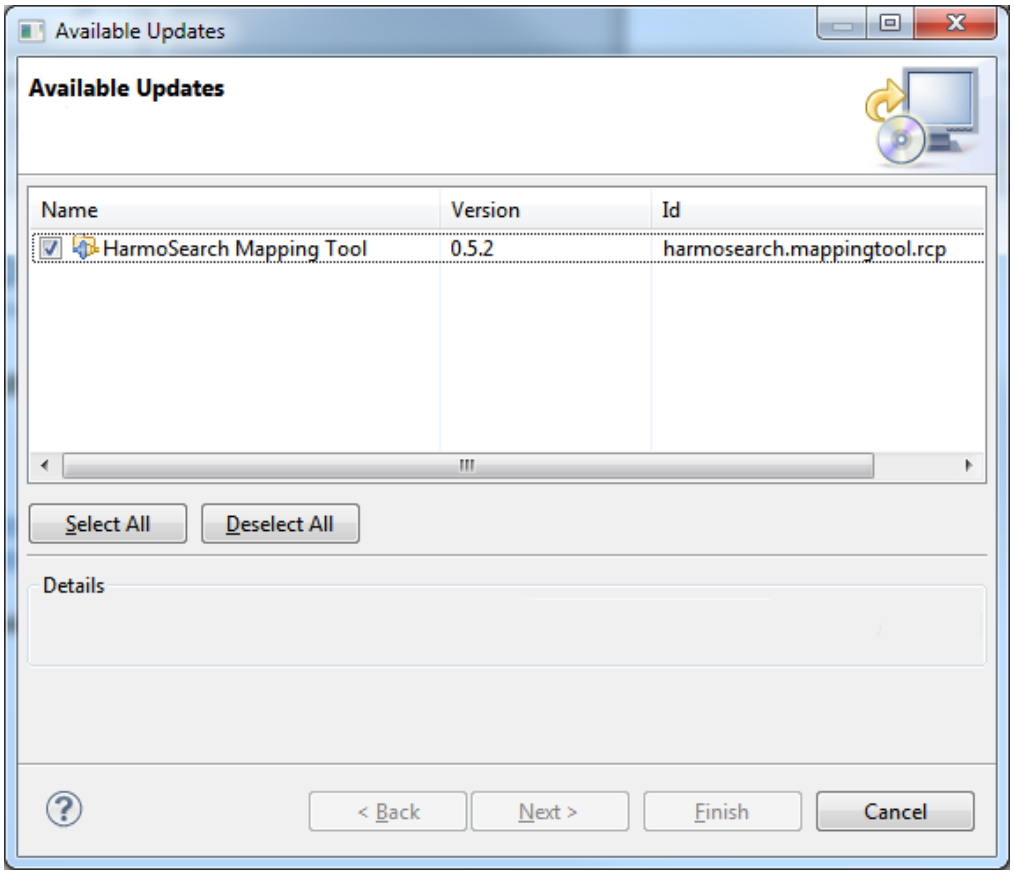


Figure 60: Updating the HarmoSearch mapping tool

4. Click **Next**
5. Review the **license agreements** for the plugins and click **I accept the terms in the license agreement** in case you agree
    a. **Stop** the installation in case you **do not agree** with the license agreements
    b. Only shown if one of the plugins has a license agreement
6. Click **Finish**
7. Click **Yes** to restart the mapping tool

6 HARMOSEARCH MAPPING TOOL USER INTERFACE REFERENCE

The following section lists all menus, toolbars, menu options, and toolbar options in the HarmoSearch mapping tool. Furthermore, each menu respectively option is described in detail.

6.1 MENUS

The following section lists and shortly describes all menus and menu options in the HarmoSearch mapping tool.

6.1.1 File

Name	Function
New	Create a new mapping project, value mapping project, value mapping file, common file, or folder
Open File	Opens a file from the file system in an editor
Save	Save the content of the current editor. The menu item is disabled if the editor does not contain unsaved changes
Save As	Save the content of the current editor under a new name
Save All	Save the content of all editors with unsaved changes. The menu item is disabled if no editor contains unsaved changes
Move	Moves a resource to a new location within the workspace, e.g., another folder or project
Rename	Renames a resource
Refresh	Refreshes the content of the selected element with the local file system. In case no particular resource has been selected all projects in the workspace are refreshed
Print	Prints the content of the current editor. This menu item is enabled when an editor has the focus
Import	Opens the import wizard dialog
Export	Opens the export wizard dialog
Properties	Opens the property page of the selected element
Exit	Exit the HarmoSearch mapping tool

6.1.2 Edit

Name	Function
Undo	Revert the last change in the editor
Redo	Revert an undone change
Cut	Copies the currently selected text or element to the clipboard and removes the element. On element, the remove is not performed before the clipboard is pasted
Copy	Copies the currently selected text or element to the clipboard
Paste	Paste the current content as text to the editor, or as a siblingor child element to the currently selected element
Delete	Delete the current text or element selection
Find/Replace	Open the Find/Replace dialog. Works only within an editor

6.1.3 Window

Name	Function
Open View	Opens a specific view, i.e.,

6.1.4 Help

Name	Function
Welcome	Opens the welcome content
Help Contents	Displays the help contents in a help windows or external browser. The help contents contain help books, topics, and information related to the Workbench and installed features.
Check for Updates	Checks for updates to the installed software

Install New Software	Allows the user to download and install new software
Preferences	Opens the HarmoSearch mapping tool preferences dialog
About	Displays information about the HarmoSearch mapping tool RCP, installed features, and available plug-ins

6.1.5 Setting

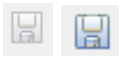
Note: becomes visible as soon as a mapping is complete and validated (a transformation was ran successfully).

Name	Function
XSL Variables	Opens the Variable dialog
XSL Generation ON / OFF	Enables or disables the generation of output XSL files
Upload to Portal	Opens the Upload Mappings dialog
Mapping Direction	Sets the mapping direction flag to one of the following values: • IN – ingoing only • OUT – outgoing only • INOUT – both direction, ingoing and outgoing

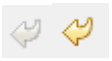
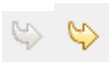
6.2 TOOLBARS

In the following section toolbars and their corresponding menu items are listed and shortly explained.

6.2.1 Main Toolbar

Name	Function	Icon
New	Allows to create new mapping projects, value mapping projects, files, and folders.	
Import	Opens the import wizard dialog	
Save	Save the content of the current editor. The menu item is disabled if the editor does not contain unsaved changes	
Save All	Save the content of all editors with unsaved changes. The menu item is disabled if no editor contains unsaved changes	

6.2.2 Mapping Editor Toolbar

Name	Function	Icon
Set XSL Variables	Opens the Variable Dialog	
Create and Run Transformation	Run XSL transformation and generate output files. Disabled when XSL generation has been set to OFF	 (yellow)
Run Transformation	Run XSL transformation and do not generate output files. No effect in case no transformation file exists.	 (green)
Undo	Revert the last change in the editor. Disabled in case no changes have happened.	
Redo	Revert an undone change. Disabled in case no undone change has been reported.	

6.3 VIEWS

The following section lists and shortly describes all views provided by the HarmoSearch mapping tool.

6.3.1 Mapping Navigator View

The Mapping Navigator View is per default shown on the left hand side of the HarmoSearch Mapping Tool. It provides a hierarchical and structured view on all mapping and value mapping projects that exist in the current workspace. From the Mapping Navigator View you can open, edit or select resources for operations, e.g., exporting, open a mapping file.

Right-click on any resource in the Mapping Navigator View to open a pop-up menu that allows you to perform certain operations, e.g., copy, move, and create a new resource.

Furthermore, the Mapping Navigator View has an additional toolbar. The toolbar supports the following functionality:

- Collapse the tree expansion state of all resources present in the Mapping Navigator View
- Link selections in the Mapping Navigator View to the active editor. When this option is selected, changing the active editor will automatically update the Mapping Navigator View's selection to the resource being edited.
- Display a menu that provides certain menu items that allow the user to sort or filter the contents of the Mapping Navigator View.

The figure below displays the structure of the Museoliitto mapping project. Depending on the resource type the Mapping Navigator View uses different icons, e.g., schema file or folder icons. The icons are always resource type dependent and may vary.

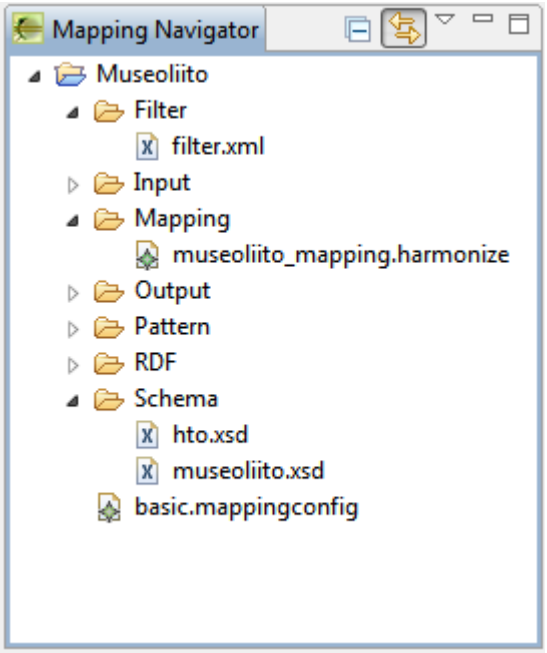


Figure 61: The Museoliitto mapping project displayed in the Mapping Navigator View

To open the Problems View click **Window→Open View→Mapping Navigator**

6.3.2 Feedback View

The Feedback View is used to provide visual feedback to the user after performing a XSL transformation. The Feedback View is able to display Web-related content such as HTML or XML files. Note, that the content depends on the transformation result. The view is per default hidden and automatically displayed as soon as a XSL transformation has been triggered.

The Feedback View has an additional toolbar similar to a standard Web browser. The following operations are supported:

- Back
- Forward
- Stop – Stops all activities
- Refresh – Refreshes the content of the Feedback View
- Open – opens the provided file in the Feedback View

The figure below shows the transformation output of the Museoliitto mapping. The output is thereby provided by means of a HTML file.

Figure 62: Transformation result displayed in the Feedback View

To open the Problems View click **Window→Open View→Feedback View**

6.3.3 Error Log View

The Error Log View is used to log all errors and warnings that are logged by plug-ins and stored in the log file. The log file can be found in the .metadata subdirectory of the workspace. Below the Error Log View shows recently captured exceptions.

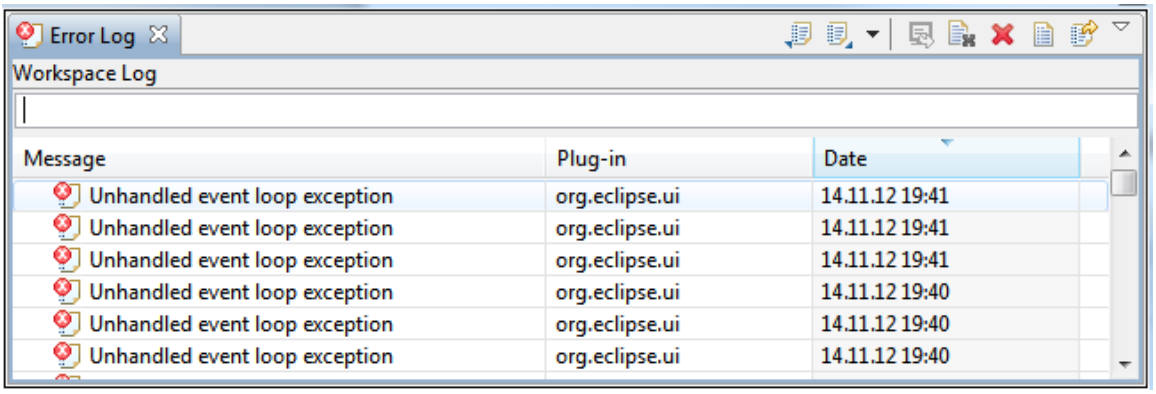


Figure 63: Log View displaying recent error messages

Events listed in the Error Log View can be sorted by certain fields, i.e., Message, Plug-in ID, or Date in ascending or descending order. To sort the provided data click on the column header of the corresponding field. An arrow in the column indicates descending (down arrow) or ascending order (up arrow).

The toolbar on the upper left hand side of the Error Log View allows advanced operations, e.g., clear and delete logs, import and export logs, open a log, or restoring a log.

To open the Problems View click **Window→Open View→Log View**

6.3.4 Problems View

The Problems View displays tool-generated errors, warnings, or information associated with a particular resource. Typically these messages are generated by builders, e.g., saving a file with syntax errors. Corresponding messages are automatically logged in the Problems View.

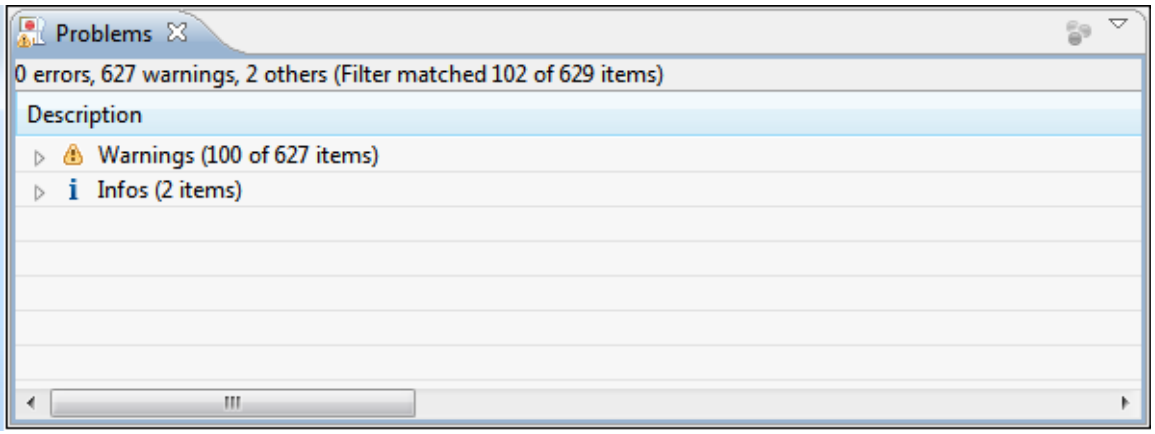


Figure 64: The Problems View showing tool-generated information messages

To open the Problems View click **Window→Open View→Problem View**

6.3.5 Properties View

The Properties View is used to display property names and basic properties of a selected resource, e.g., a mapping file or a schema element in the HarmoSearch Graph Editor:

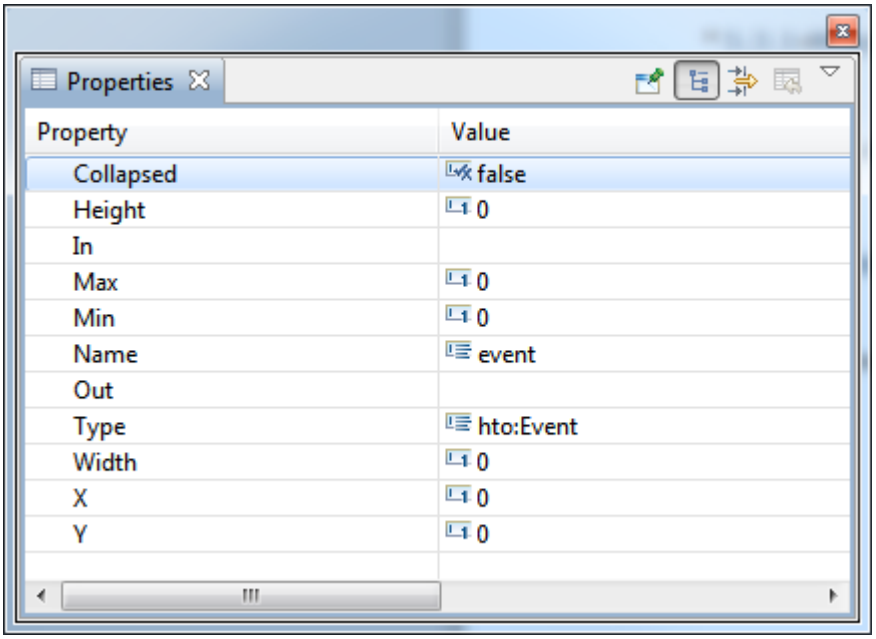


Figure 65: Properties of a schema element

The toolbar buttons on the upper right hand side allow the user to toggle whether to display properties by category and whether to filter advanced properties. Furthermore, another toolbar button allows restoring the selected property to its default value.

To open the Properties View click **Window→Open View→Properties**

6.3.6 Outline View

The Outline View displays and lists structural elements of a structured file or model currently opened in the editor area. Please note, that the contents displayed in the Outline View are always editor specific.

The example below shows the structure of a HarmoSearch mapping file, representing schemas and links.

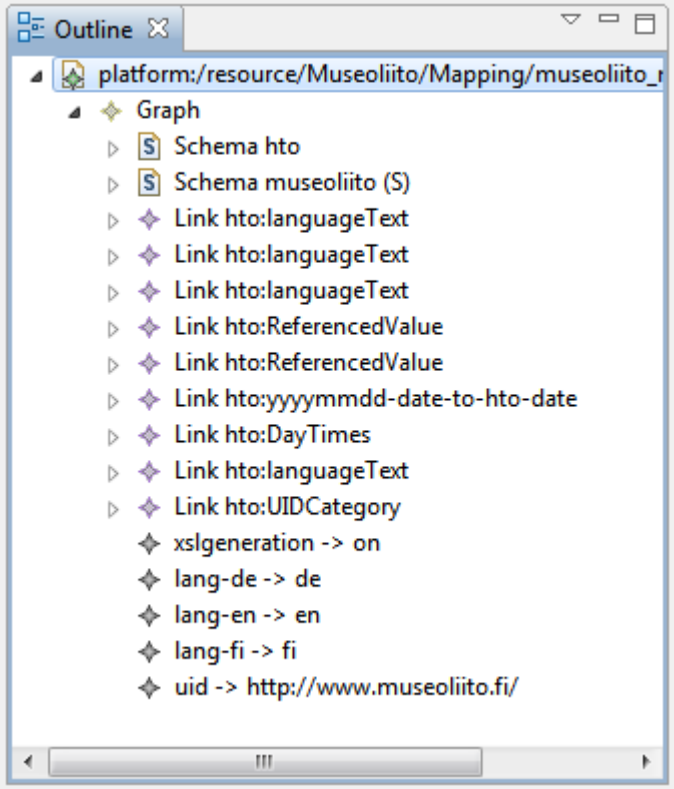


Figure 66: Outline View displaying structural elements of a mapping file

To open the Outline View in the HarmoSearch mapping tool click **Window→Open View→Content Outline**

6.4 EDITORS

The following section lists and shortly describes all editors provided by the HarmoSearch mapping tool.

The HarmoSearch mapping tool comprises the following editors:

- HarmoSearch Mapping Editor
- Value Mapping Editor
- XML Editor

6.4.1 HarmoSearch Mapping Editor

The HarmoSearch Mapping Editor is used to operate and manipulate a HarmoSearch mapping model. The mapping model is a hierarchical structured graph that consists of schemas, schema elements and links. Links are used to define a relation between schema elements from one schema to schema elements of another schema.

The HarmoSearch Mapping Editor supports two different ways for creating and changing the mapping model: graph-based and form-based. Both approaches are encapsulated by separate views and complement each other:

- Graph View
- Forms View

Graph View

The Graph View (cf., Figure 67) can be used to create or change a mapping model fully graphically. The main features of the editor are:

- A graphical modeling approach
- Intuitive visual feedback highlighting the current status of a mapping link
- Light-weight configuration via a dialog-based approach
- Simple drag-and-drop approach for loading schemas to the editor

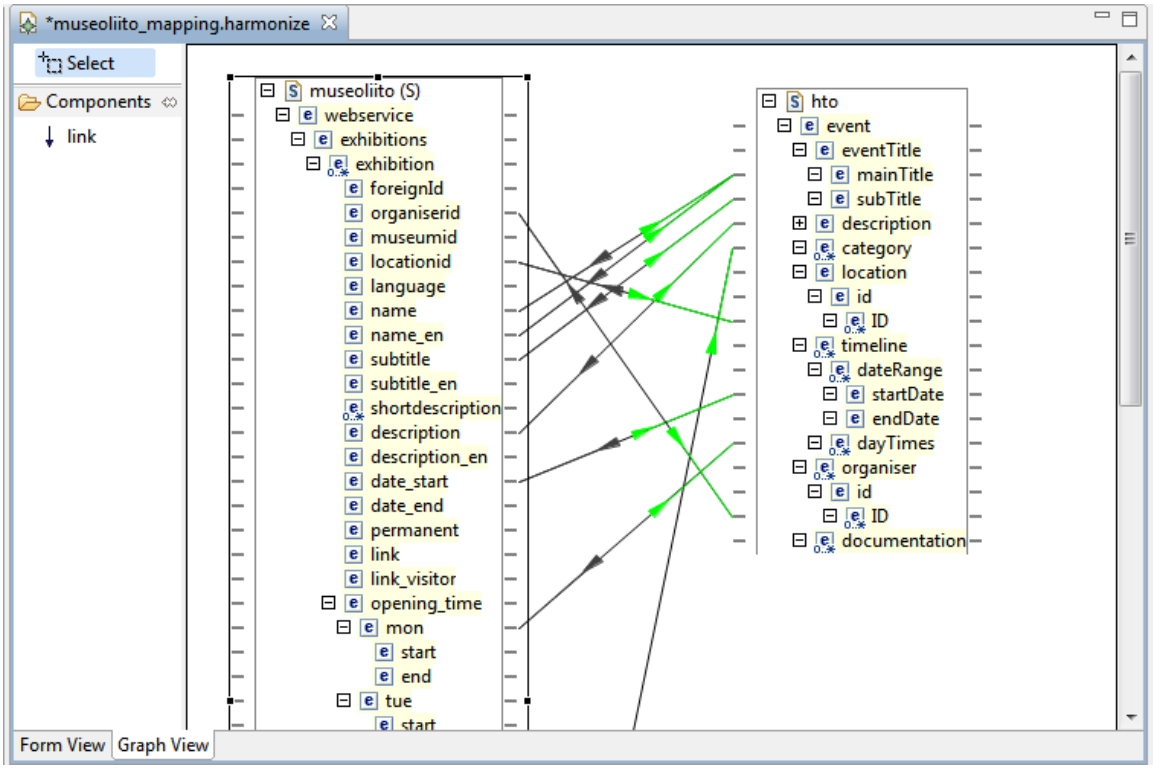


Figure 67: The Graph View editor displaying the Museoliitto to Harmonise mapping

The Graph View provides a modeling canvas and a list of modeling elements. The canvas is shown on the right hand side of Figure 67. It is used to load display schemas and mapping links. Schemas are represented as trees and reflect the structure of the underlying XML schema hierarchically. Links are displayed as simple lines connecting elements from one schema with elements of another schema. Links are added to the mapping model by simply click on the proper source and target elements on the modeling canvas. Therefore, the user has to enter the link mode, which is done by clicking on the link modeling element located on the left hand side of the Graph View (cf., Figure 67).

Mappings are configured by using a light-weight dialog which pops up when clicking a mapping link. The dialog shows detailed information about the configured mapping setting:

1. The target mapping element
2. A list of available mapping patterns
3. List of required mapping parameters (depend on the selected mapping pattern)

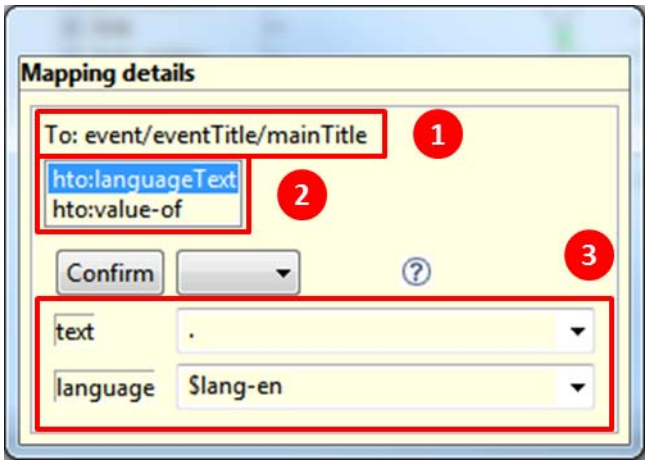


Figure 68: The Mapping Details Dialog

The actual mapping settings are persisted by clicking Confirm. In addition the user may specify the current status of the mapping. Setting the status of a mapping also affects the graphical representation of the corresponding mapping link. The table below illustrates the available mapping states and their implications on the mapping link under consideration.

Status	Functionality	Implication
OK	Denotes a finalized mapping.	Green mapping link
TODO	Used for mappings that cannot be solved by the domain expert, e.g., unclear semantics of the target element or missing pattern support.	Red mapping link
Reset	Resets the mapping details	Black mapping link

Forms View

The Forms View (cf., Figure 69) provides an alternative view on the mapping model. Similar to the Graph View it is used to create and maintain a mapping between a source and a target schema. The main functions of the view are as follows:

- Mapping filter base guidance
- Intuitive visual feedback highlighting the current status of a mapping
- Simple drag-and-drop approach for loading the source schema to the editor
- Light-weight form-based mapping approach

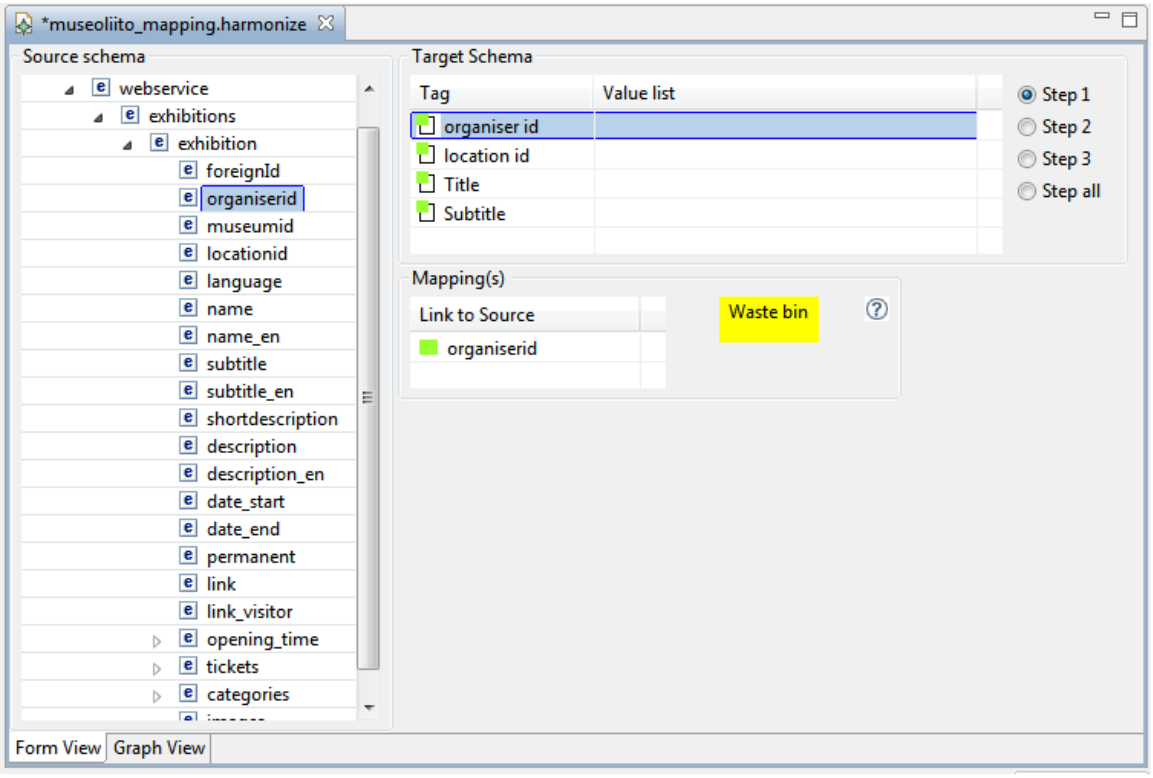


Figure 69: The Forms View editor displaying the Museoliitto to Harmonise mapping

As shown in Figure 69 the Forms View is separated into two main areas: the source schema and the target schema. The source schema area displays the source data schema's structure as a tree whereas the target schema area shows dedicated information about target elements and mappings.

Mapping a source element to a specific target element is simply done by dragging the source element onto the Mappings list for the dedicated target element. The number of mappings per element is thereby not restricted. Accordingly, a single target element may be interlinked to multiple source elements.

Similar to the Graph View the Forms View facilitates functionality for specifying the mapping details for a link.

6.4.2 Value Mapping Editor

The Value Mapping Editor is used to define mappings between elements of a source reference list and elements of a target reference list. Similar to the HarmoSearch Mapping Editor the Value Mapping Editor allows defining certain links between two elements.

The Value Mapping Editor has the following main functionalities:

- Conceptual mapping of reference lists
- Intelligent user feedback in case a concept or reference list has changed

The editor is composed of three main parts:

- The source area which lists all available source reference lists and their associated elements
- The target area, which lists all available target reference lists and their associated elements.
- The mapping area allowing to map a source reference element to a dedicated target reference element

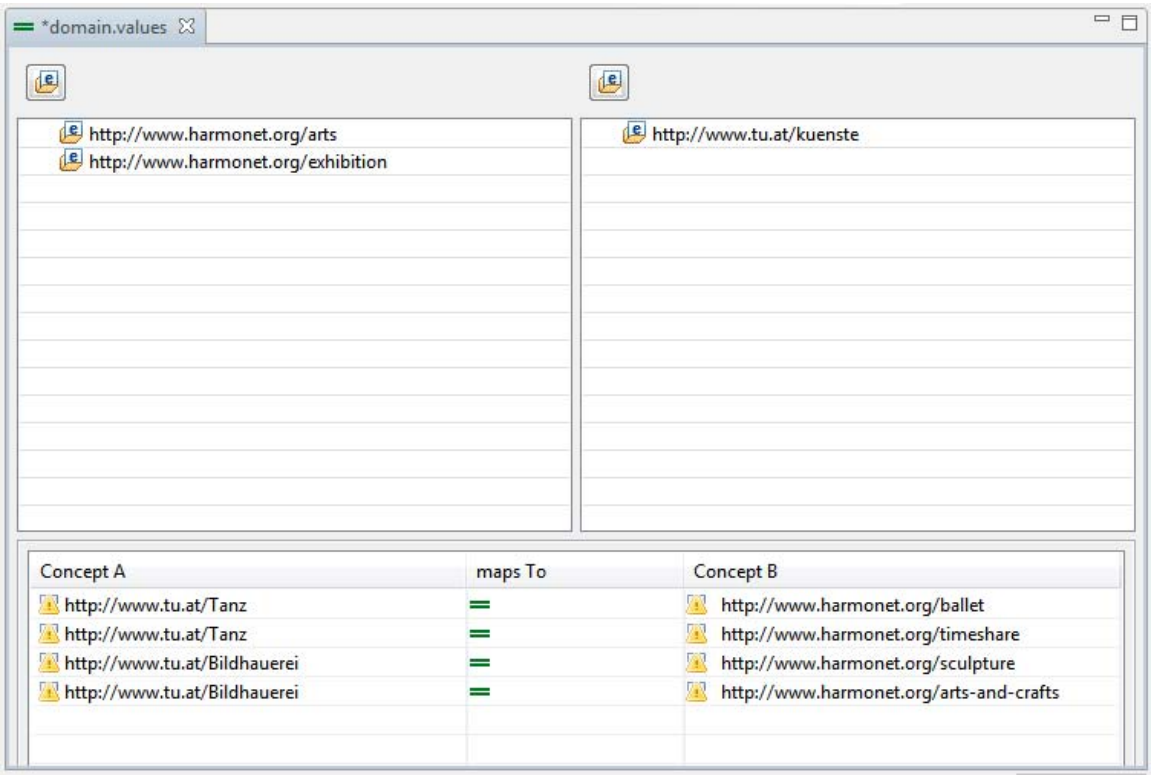


Figure 70: The Value Mapping Editor displaying a common value mapping

6.4.3 XML Editor

The XML Editor is used to create and view XML files supporting the following main functions:

- Syntax highlighting for XML-based documents
- Editing and designing XML-based documents

The XML Editor is automatically called in case the user wants to open an XML or XML Schema file.

The XML Editor has two main views:

- Source View
- Design View

In the following both views are presented in more detail.

Source View

The Source View (cf., Figure 71) is used to work directly with a file's source code. The underlying file is thereby loaded into the editor. Its content is presented syntactically highlighted in the editor.

The features of the Source View Editor can be summarized as follows:

- Syntax highlighting
- Content assist providing a list of acceptable continuations depending on where the cursor is located in the XML file
- Smart user assistance by using double clicking features. For example, if the cursor is placed in an attribute value, a single double-click selects the value, another double click selects the value and the associated attribute, and a third double click selects the entire tag.

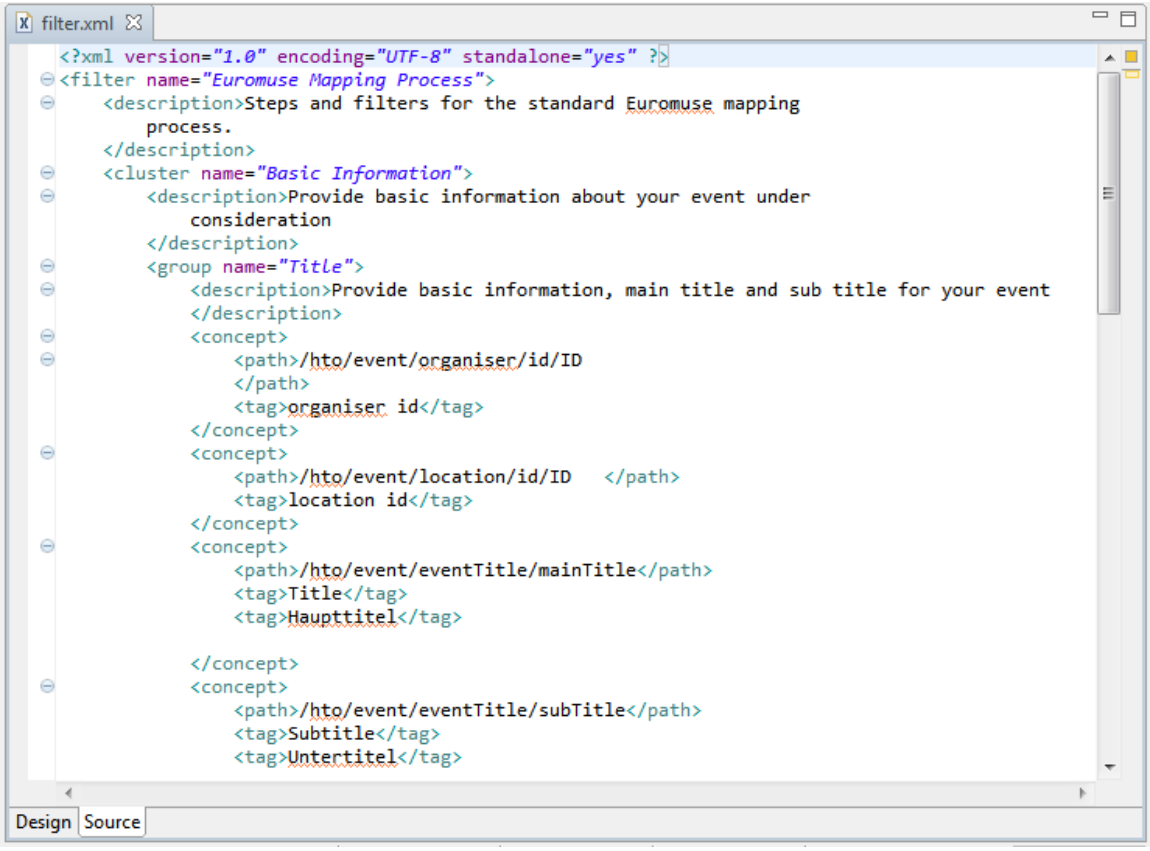


Figure 71: Source view of the XML Editor

Design View

The Design View (cf., Figure 72) is the second view supported by the XML Editor. Its purpose is to provide an easy to use approach for creating and editing XML files without writing XML-code by hand. This is especially helpful for users who are new to XML or need to do form-oriented editing.

The following functionality is provided by the Design View:

- The XML file's content is represented graphically, both and simultaneously as a table and a tree enabling an easier navigation and editing experience for the user.
- Content and attribute values can be edited directly in the table cells. In addition pop-up menus on the tree nodes give alterantives that are valid for that location.

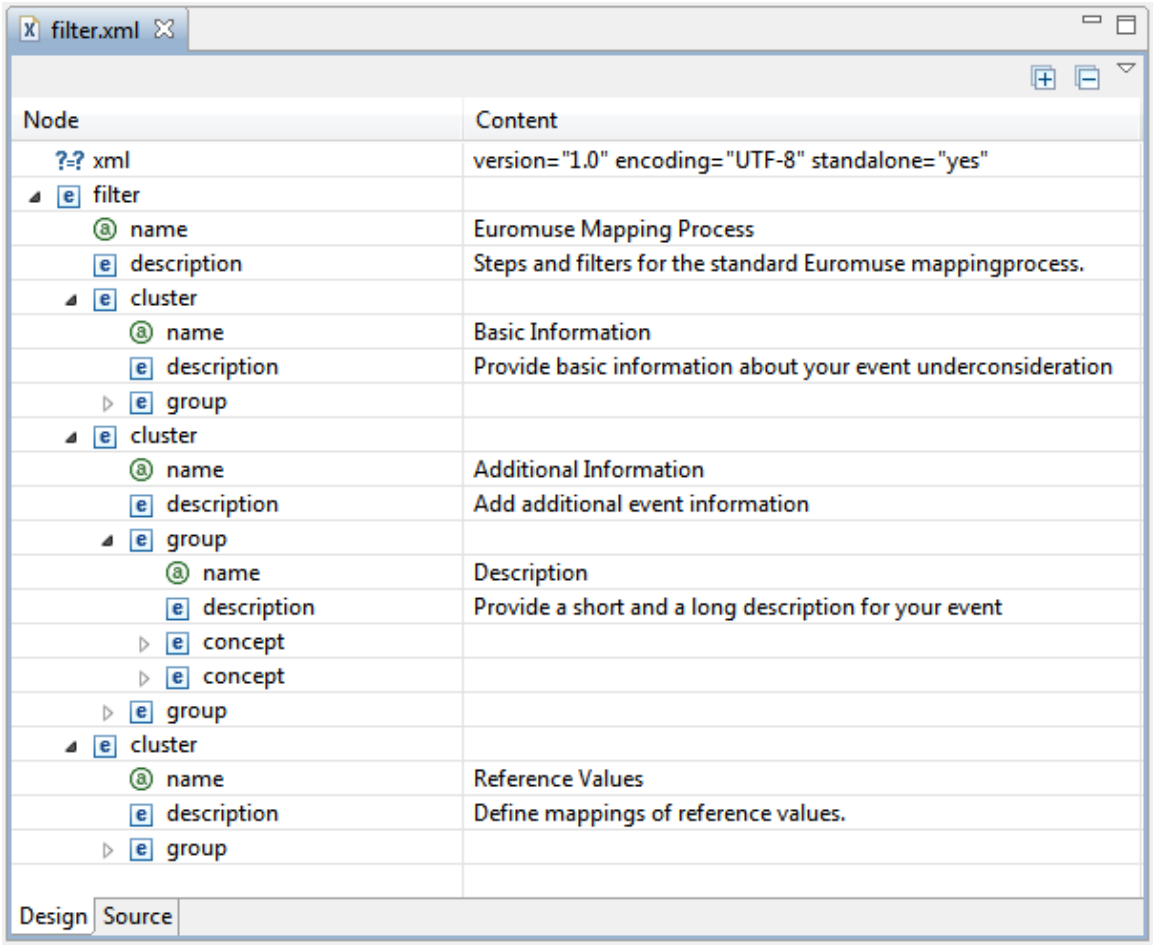


Figure 72: Design view of the XML Editor

6.5 WIZARDS

The following section lists all software wizards that are supported by the HarmoSearch Mapping Tool and shortly describes their individual purpose and functionality. Wizards are used to lead the user step-wise through a well-defined process, e.g., creating a mapping process. Furthermore, each wizard has a short step-by-step guidance section explaining how to use the wizard in practice.

The following wizards are integrated and supported by the HarmoSearch mapping tool:

- Mapping Project Wizard
- Import Bundled Files Resource Wizard
- Import File Resource Wizard
- Value Mapping Project Wizard
- Value Mapping File Wizard
- Export Wizard
- Import Wizard

6.5.1 Mapping Project Wizard

The Mapping Project Wizard (cf., Figure 73) is used to create a basic mapping project in the current workspace. Creating a project requires:

- A valid project name. The provided name is used as ID and had wo be unique in the current workspace. Otherwise an error message will be displayed in the dialog window.
- A proper name for the mapping file name which is created and added to the mapping project.
- Proper value for the file encoding

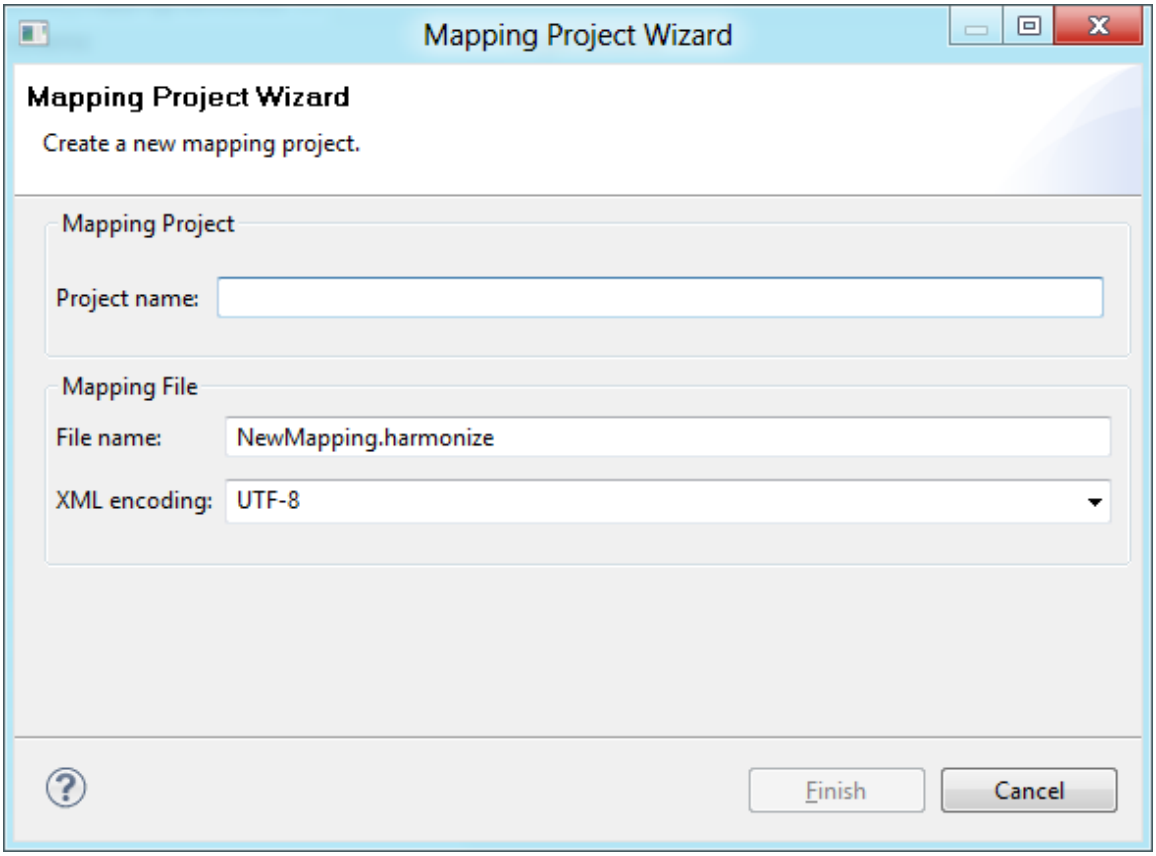


Figure 73: Creating a new mapping project using the Mapping Project Wizard

The Mapping Project Wizard has the following options:

Name	Function	Default Value
Project Name	Name of the project. The name has to be unique in the workspace. If not a proper error message will be displayed.	Blank
File Name	The name of the mapping file name which is per default added to the mapping project.	NewMapping.harmonize
XML Encoding	The encoding of the mapping file. Select the supported encoding via the drop-down list. The following standards are supported: • UTF-8 • UTF-16 • UTF-16LE • UTF-16BE • ISO-8859-1 • ASCII	UTF-8

Step-by-step guidance

1. Open the **Mapping Project Wizard** by clicking **File→New→Mapping Project**
2. Enter a valid **project name**
3. Optionally change the **name** for the **mapping file**
4. Choose the **XML encoding** of your choice from the **drop-down list**
5. Click **Finish**

6.5.2 Import Bundled Files Resource Wizard

The Import Bundle Files Resources Wizard (cf., Figure 74) is used to import single or multiple files of the same type (e.g., schema files or pattern files) into a mapping project. The mapping project has to exist in the current workspace. Single files are imported by pointing to the file location on the local file system whether the multi file functionality takes an archive file as data source. The archive file has to be in Zip format. Included files are listed and can be checked by the user (marked to be imported). Important to note is that whether single or multiple file import has been chosen the original file(s) stay in place. Instead the selected file(s) will be copies and added to the selected mapping project (both, physically on the hard-disk as well as in the current workspace).

The Import Bundled Files Resources Wizard is used for importing:

- Schema files
- Pattern files
- Input data files, e.g., data instances or transformation process specifications

Depending on the type of file to be imported the Import File Resources Wizard is synonymously used with:

- Schema Import Wizard
- Pattern Import Wizard
- Filter Import Wizard
- Input File Import Wizard

The tasks necessary to call the corresponding Import File Resources Wizard is the same for all the above mentioned wizards.

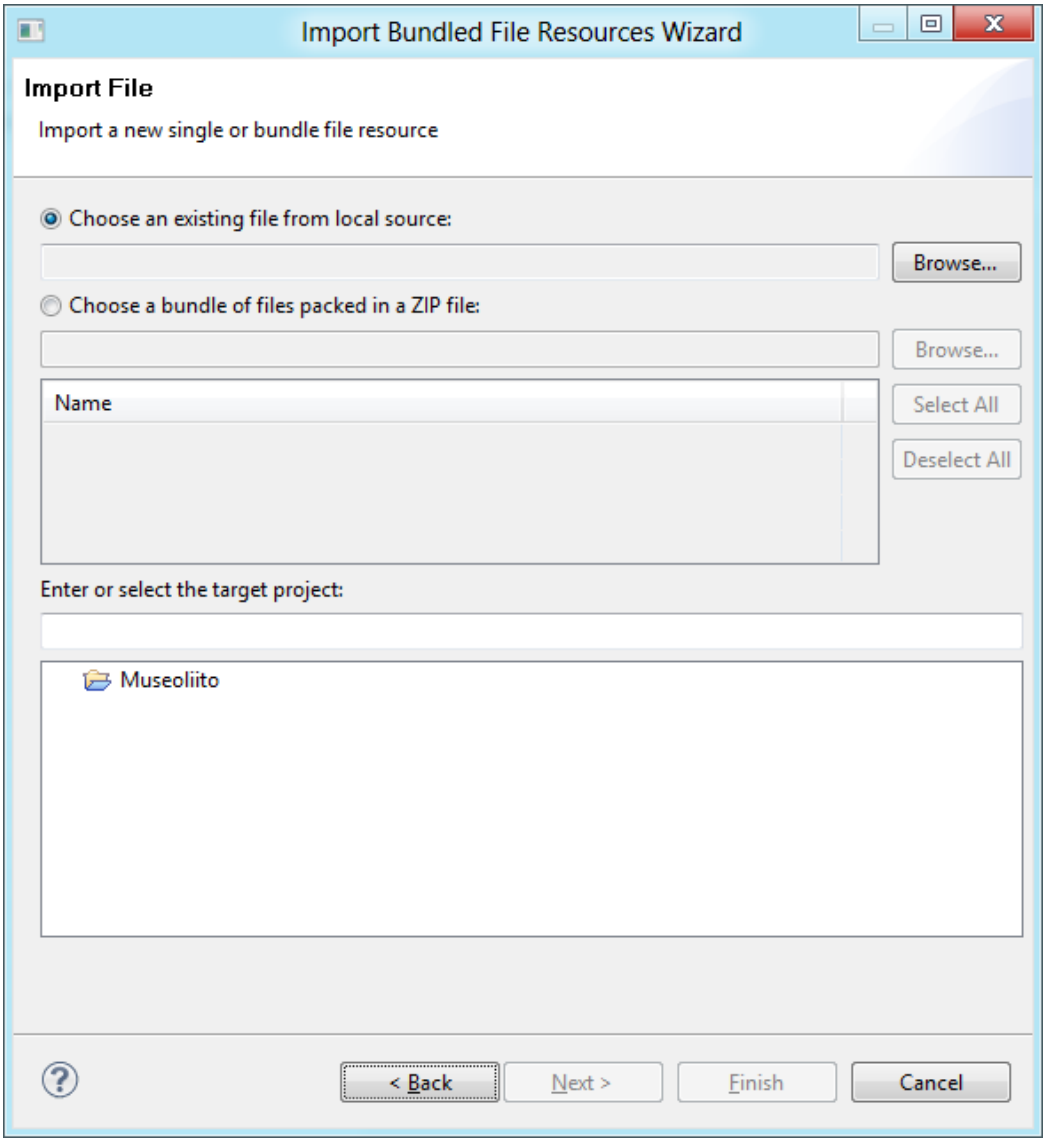


Figure 74: The Import Bundled File Resources Wizard

The Import Bundle Files Resources Wizard has the following options:

Name	Function	Default Value
Choose an Existing File from Local Source	A single file which is the actual target of import. The file has to exist on the local hard-disk. Point to the location of the file by using the Browse button.	Selected, Blank
Choose a Bundle of Files Packed in a Zip File	A Zip file serving as data source. The Zip is scanned and potential files are listed in the list below. Click Browse to locate the Zip archive on your local hard disk	Not selected, Blank
Select All	Select all listed files	
Deselect All	Deselect all selected files	
Enter or Select the Target Project	The target project to which the file under consideration will be imported. The project has to exist in the current workspace. Existing projects are listed below and can be easily selected by clicking. Alternatively provide the path in the text box.	Blank

Step-by-step guidance

6. Open the **Import Bundle Files Resources Wizard** by clicking **File→Import**
7. Expand **HarmoSearch Wizards**
8. Click on **Import Input File** and press **Next**
    a. Depending on the resource type choose alternatively **Import Pattern File** or **Import Schema File**
9. Select **Choose a bundle of files packed in a ZIP file**
    a. Alternatively choose **Choose an existing file from local source** in case you only want to import a single file. In this case you can skip task 5 and continue with task 6.
10. Check all files you want to import
11. Select the **target project** from the current workspace
12. Click **Finish**

6.5.3 Import File Resources Wizard

The Import File Resources Wizard (cf., Figure 75) is used to import a specific file to a mapping project. The file has to exist on the local file system as well as the target project which has to exist in the current workspace. The workspace is listed in a tree and shows all existing mapping projects. The source file which is the actual import target can be located in the file system by clicking the Browse button.

The Import File Resources Wizard is mainly used for importing filter files which usually exist once per mapping project.

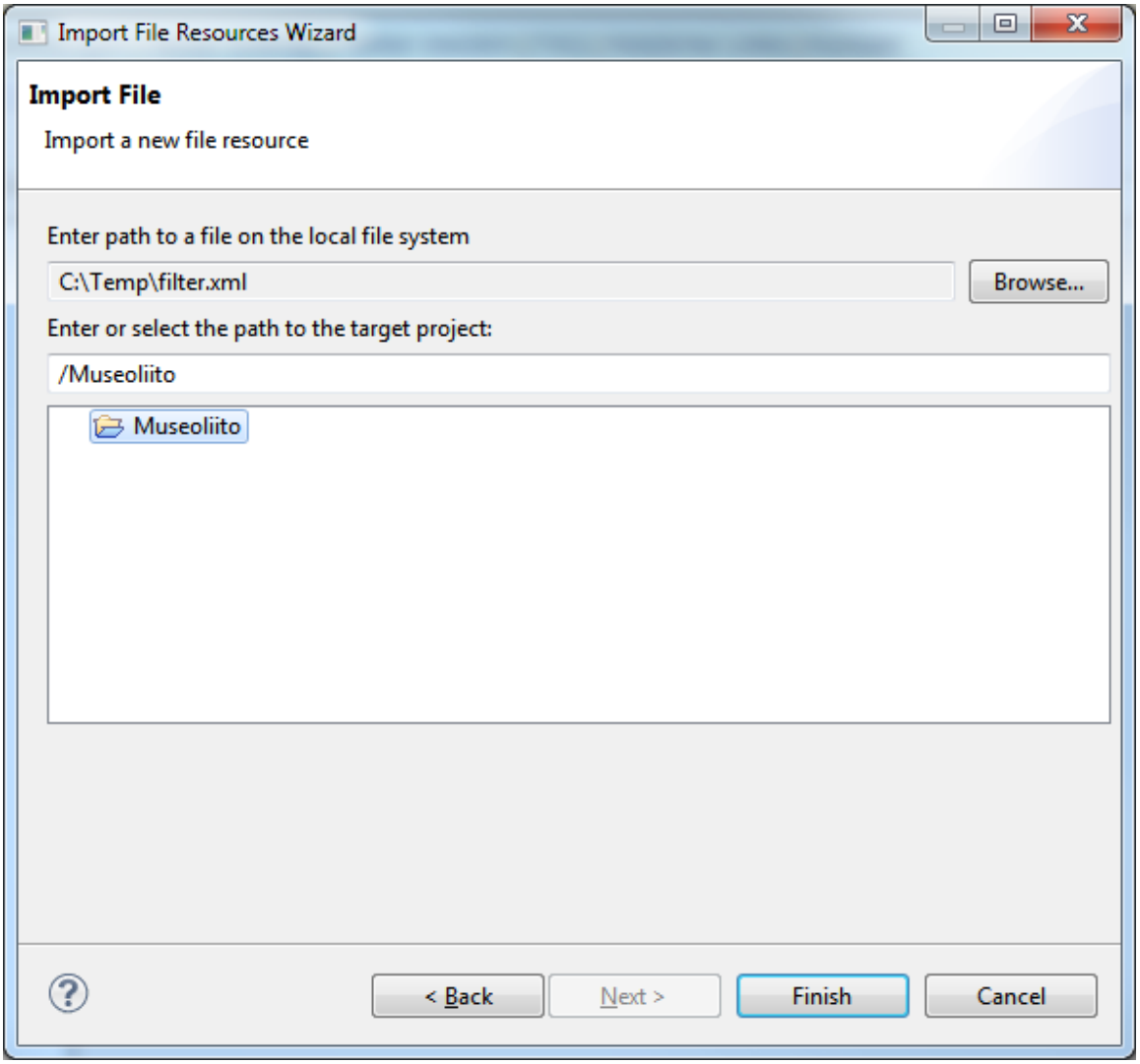


Figure 75: The Import File Resources Wizard

The Import File Resources Wizard has the following options:

Name	Function	Default Value
Enter Path to a File on the Local File System	The file which is the subject of import. The file has to exist on the local file system. Locate the file by using the Browse button.	Blank
Enter or Select the Path to the Target Project	The project the file has to be imported. The project has to exist in the workspace. All available projects are listed in the tree below. Select a project by clicking on it.	Blank

Step-by-step guidance

13. Open the Import File Resources Wizard by clicking **File→Import**
14. Expand **HarmoSearch Wizards**
15. Click on **Import Filter File** and press **Next**
16. Locate the target file by clicking **Browse**
17. Select the **target project** from the current workspace
18. Click **Finish**

6.5.4 Value Mapping Project Wizard

The Value Mapping Project Wizard (cf., Figure 76) is used to create a new value mapping project in the current workspace. The project has to be new and non-existing in the current workspace (otherwise it will not be created). The user has to provide the following information:

- A proper project name. Similar to mapping projects value mapping projects are identified by name. Thus, the project name has to be unique in the current workspace.
- A proper file name for the value mapping file which is created by default

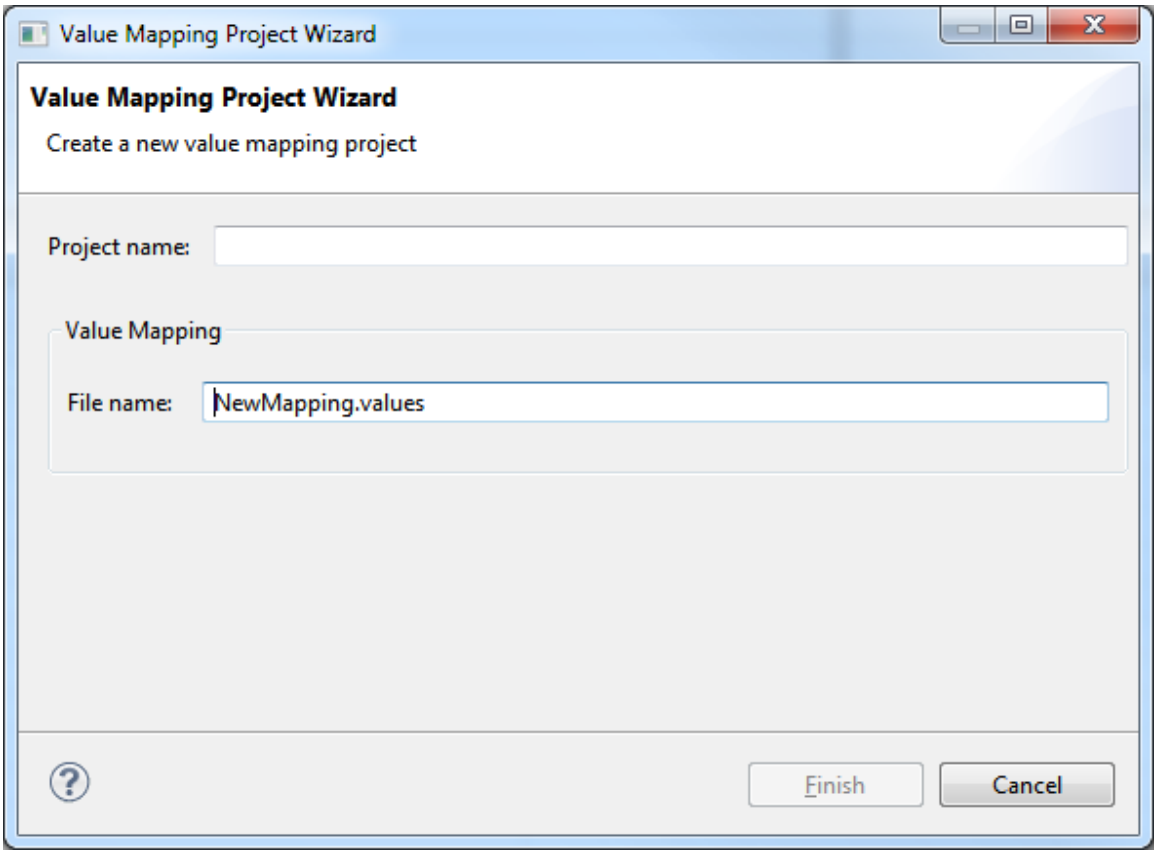


Figure 76: Interface of the Value Mapping Project wizard

The Value Mapping Project Wizard has the following options:

Name	Function	Default Value
Project Name	The name of the project which should be created. The name has to be unique in the workspace. In case it is not an error message is displayed at the top of the Value Mapping Project Wizard.	Blank
File Name	The file name for the value mapping file	NewMapping.values

Step-by-step guidance

19. Open the Value Mapping Project Wizard by clicking **File→New→Value Mapping Project** in the main menu
20. Provide a proper **project name**
21. Optionally change the **mapping file name**
22. Click **Finish**

6.5.5 Value Mapping File Wizard

The Value Mapping File Wizard (cf. Figure 77) is used to create a new value mapping file. The value mapping file is created in a mapping or value mapping project which exists in the current workspace. The current workspace is displayed as a tree in the Mapping Navigator View. The user is allowed to create new mapping files in projects and folders.

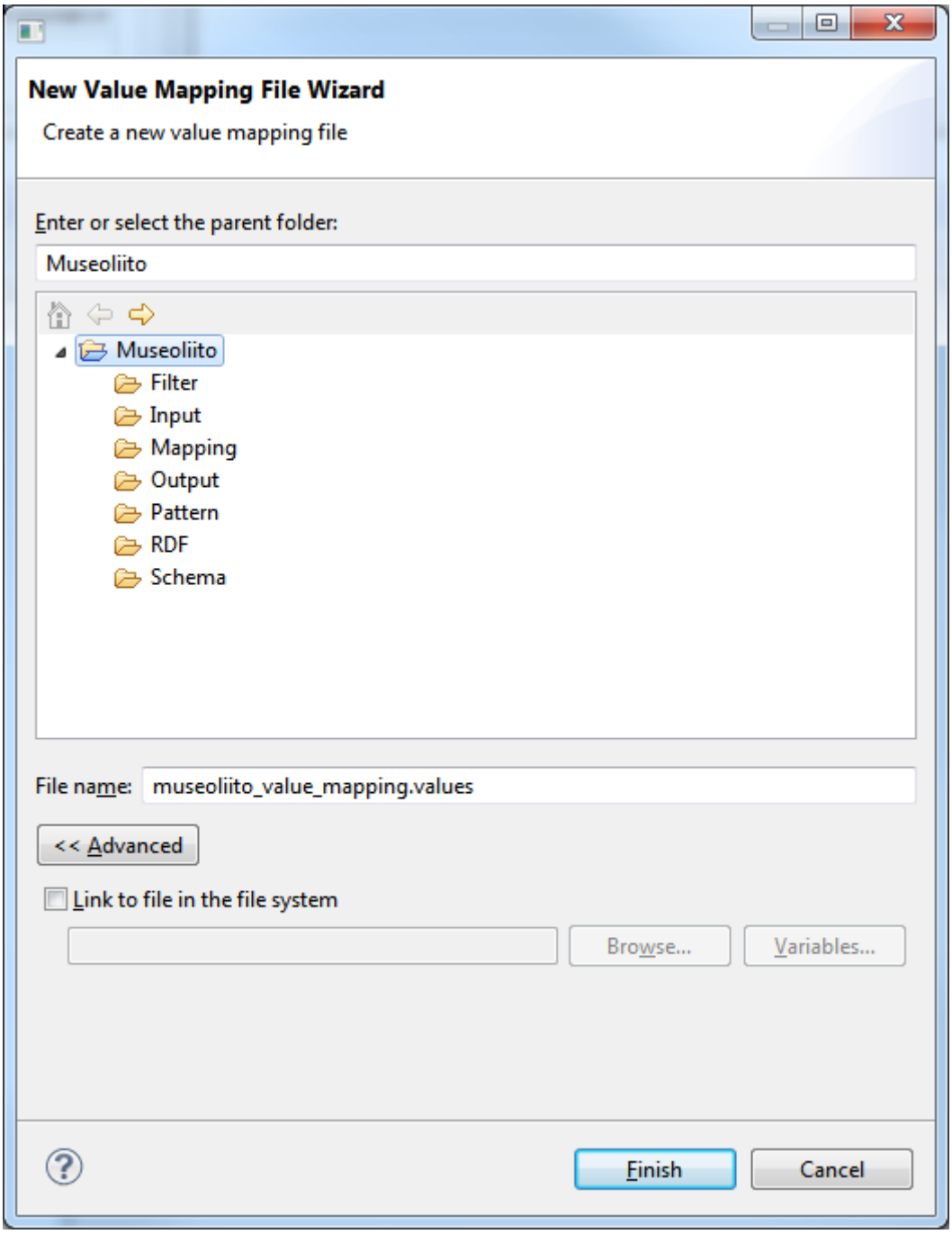


Figure 77: The Value Mapping File Wizard creating a new value mapping file

The Value Mapping File Wizard has the following options:

Name	Function	Default Value
Enter or Select the Parent Folder	Type in or select the location you want the newly created value mapping file be created in the textbox or select the target location from the tree below. Valid locations are projects and folders.	The project of the selected resource or blank
File Name	Specify the name of the value mapping file to be created	Blank
Advanced	Opens the advanced group	
Link to File in the File System	When checked the wizard creates a linked file resource and adds it to the selected project. Use the Browse button to browse for a file in the file system. Alternatively provide the file path or use a path variable.	Unchecked

Step-by-step guidance

1. Open the Value Mapping File Wizard by clicking **File→New→Value Mapping File** in the **main menu**
2. Select the target resource, which can either be a project existing in the current workspace or a folder
3. Provide an adequate **file name** for the value mapping file
4. Optionally click on **Advanced** and check **Link to file in the file system** and locate it by using a path variable or the **Browse** button
5. Click **Finish**

6.5.6 Export Wizard

The Export Wizard (cf., Figure 78) is used to export existing mapping and value mapping projects from the current workspace. The exported projects are stored and compressed in an archive file. The Export Wizard supports the following two archive types:

- Zip format
- Tar format

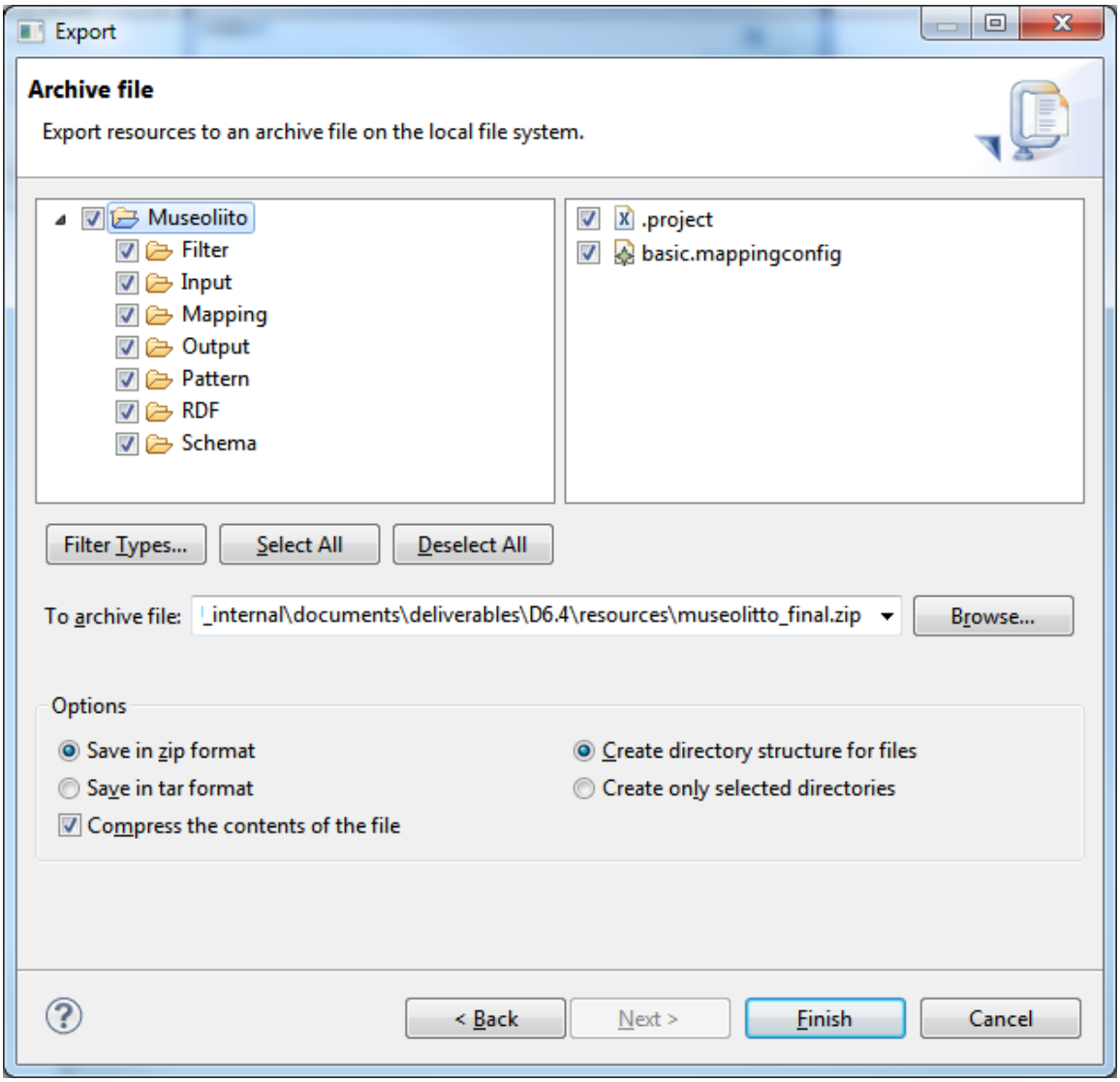


Figure 78: The Export Wizard

The Export Wizard has the following options:

Name	Function	Default Value
Select Resources to Export	Select the mapping or value mapping project and all associated resources (files and folders) to export to an archive.	The project holding the selected resource
Select Types	A dialog that allows restricting the export to only certain file types.	
Select All	Check all resources for a project	
Deselect All	Uncheck all selected resources	
Archive File	Specifies the path and name of the target archive file which is the actual result of the export. All selected resources will be exported into the archive file. Click Browse to locate the archive's location. Previous paths can be selected via the drop-down list.	Blank or the location and archive file name of the previous export
Zip File	Export the selected resources in Zip format	Selected
Tar file	Export the selected resources in Tar format	Not selected
Compress the Contents of the file	Compresses the contents of the selected mapping or value mapping project in the archive that is created	Checked
Create Directory Structure for Files	Copy and create the given hierarchy structure in the file system. The hierarchy reflects the folder structure as it exists in the current workspace.	Selected
Create Only Selected directories	Create folder structure (hierarchy) only for selected folders.	Not selected

Step-by-step guidance

1. Open the Export Wizard by clicking **File→Export** in the **main menu**
2. Select **Archive File** and click Next
3. Check all projects and associated resources you want to export
4. Set the location to which the final archive file will be exported
5. Select the **type of archive** you want to create
6. Select whether Create directory structure for files or **Create only selected directories**
7. Check or uncheck **Compress the contents of the file**
8. Click **Finish**

6.5.7 Import Wizard

The Import Wizard (cf., Figure 79) is used to import existing mapping or value mapping projects into the current workspace.

The Import Wizard supports two sources:

- Import projects from an existing folder on your local hard-disk
- Import projects from an archive file located on your local hard-disk

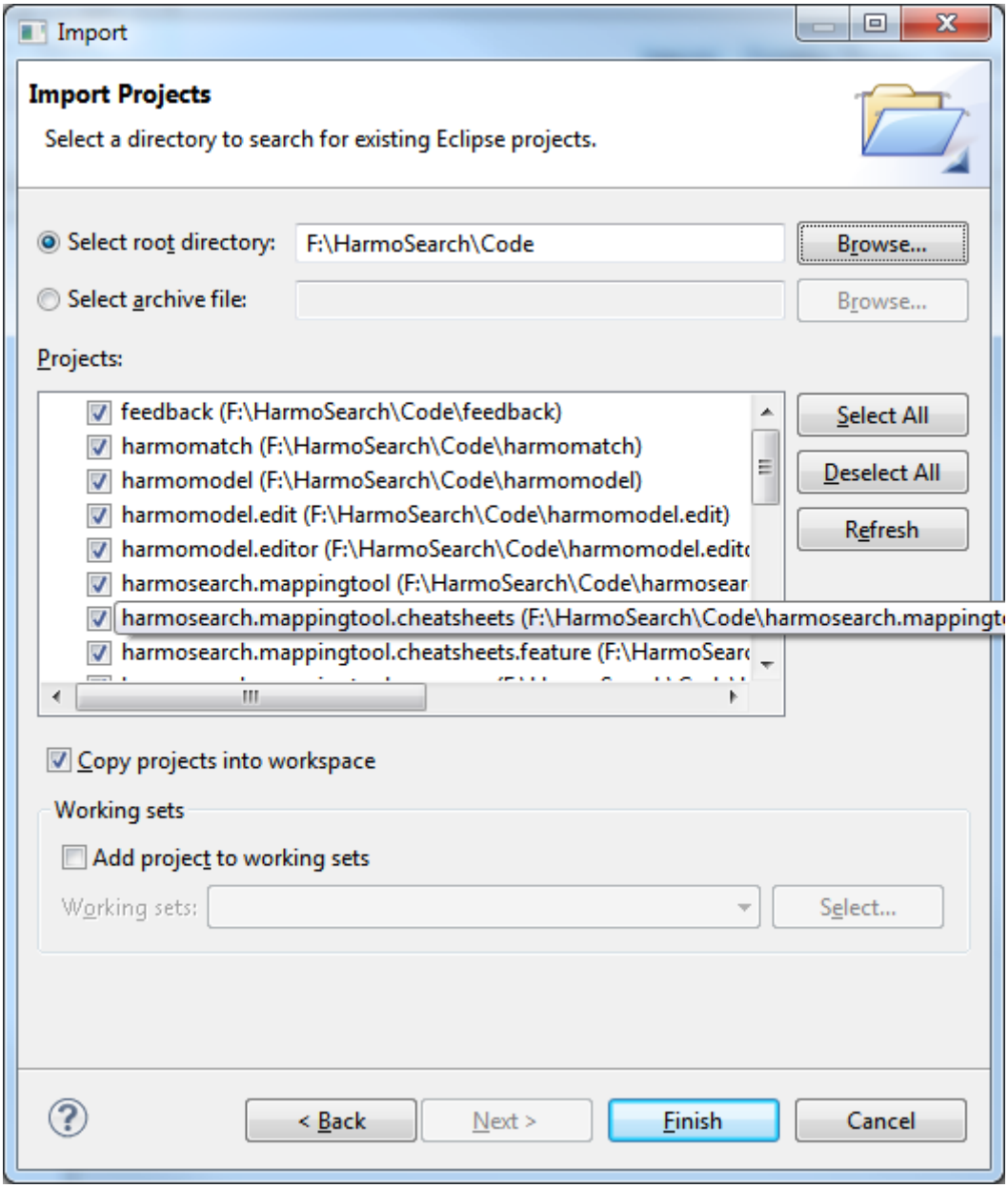


Figure 79: The Import Wizard dialog

The Import Wizard has the following options:

Name	Function	Default Value
Select Root Directory	The root directory in the file system containing projects to import. The root directory is the starting point for scanning potential projects to import. Provide a root directory or use the Browse button to define the path on the system.	Enabled, blank text box
Select Archive File	Selects an archive file to scan for projects to import. Enter the path of the archive file or use the Browse button to locate the archive's location in the file system.	Disabled
Select All	Check all of the projects that were found for import.	
Deselect All	Uncheck all project	
Refresh	Rescan the selected source for projects to import	
Copy Projects into Workspace	When checked the option causes an import of the selected projects into the current workspace	Unchecked
Add Project to Working Sets	When checked the selected projects are added to existing working sets. Locate the working sets the projects should be added to by clicking Select or the drop-down box	Unchecked

Step-by-step guidance

1. Open the Import Wizard by clicking **File→Import** in the **main menu**
2. Expand **General**, select **Existing Projects into Workspace** and click **Next**
3. Choose the source where the projects are located by clicking whether **Select Root directory** or **Select archive file**
4. Select the projects you want to import
5. Check or uncheck **Copy projects into workspace**
6. Check or uncheck **Add project to working sets** and select the target working sets from the drop-down box or click Select
7. Click **Finish**

7 TECHNICAL INFORMATION

The installer for the HarmoSearch mapping tool can be downloaded from the HarmoSearch portal. More information on how to download the HarmoSearch mapping tool can be found in Section 3.2.

In order to install and run the HarmoSearch mapping tool the following system requirements have to be given:

- Supported operating systems:
  - Windows 7 (x86 or x64)
  - Windows Vista (x86 or x64)
  - Windows XP (x86 or x64)
- Java Run-time Kit version 1.6 or higher
  - Click http://www.oracle.com/technetwork/java/javase/downloads/jre-7u2-download-1377135.html for download

8 LIST OF FIGURES