http://iringtools.org

Users Guide

Version 1.02.01

Terms of License

Contents

The total number of pages in this document, including the cover page, is 38.

List of Abbreviations

Acronym	Description
iRING	ISO 15926 Realtime Interoperability Network Grid
ISO	International Organization for Standardization
RDSWIP	Reference Data Service Work in Progress
SP	Service Pack
GUI	Graphical User Interface
IIS	Internet Information Services
MIME	Multipurpose Internet Mail Extensions
OLTP	Online Transaction Processing
API	Application Programming Interface
CRUD	Create, Read, Update and Delete
LAN	Local Area Network
FIPS	Federal Information Processing Standard

1 Overview

iRING is a set of information interoperability and integration protocols and reference data that are compliant with the ISO 15926, Parts 7, 8, and 9 standards, which builds and depends on ISO 15926 Parts 1 through 6.

iRINGTools is a set of free, public domain, open source (BSD 3 license) software applications and utilities that implement ***iRING*** protocols. ***iRINGTools*** provide users with production ready deployable solutions. ***iRINGTools*** also provides technology solution providers with usage patterns for the implementation of ***iRING*** protocols in their respective solutions.

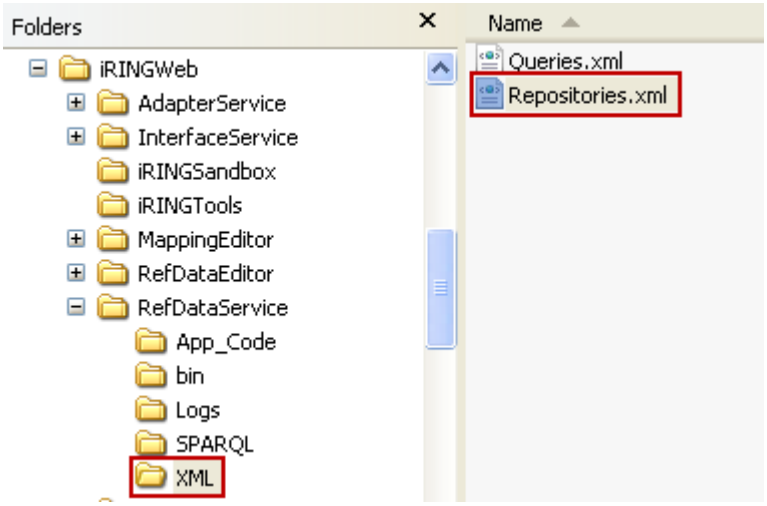
This user guide provides detailed instructions for using ***iRINGTools*** Reference Data Editor and Mapping Editor.

2 Reference Data Repositories

The default installation of Sandbox has three reference data repositories:

1. My Private Sandbox – Development Sandbox (writable)
2. ReferenceData – RDS/WIP Part 4 Reference Data Library (read-only)
3. Templates – RDS/WIP Part 7 Template Library (read-only)

The repositories are defined in the Repositories.xml file located in the RefDataService/XML folder.



To add, change or delete repositiories, open the Repositories.xml in an XML editor and change as necessary. See example below.

Repositories.xml

```
<ArrayOfRepository xmlns="http://schemas.datacontract.org/2004/07/org.i
  <Repository>
    <description>Development Sandbox</description>
    <isReadOnly>false</isReadOnly>
    <name>My Private Sandbox</name>
    <uri>http://localhost/SandboxService/sparql</uri>
  </Repository>
  <Repository>
    <description>RDS/WIP Part 4 Reference Data Library</description>
    <isReadOnly>true</isReadOnly>
    <name>ReferenceData</name>
    <uri>http://rdl.rdlfacade.org/data</uri>
  </Repository>
  <Repository>
    <description>RDS/WIP Part 7 Template Library</description>
    <isReadOnly>true</isReadOnly>
    <name>Templates</name>
    <uri>http://tpl.rdlfacade.org/data</uri>
  </Repository>
</ArrayOfRepository>
```

3 Reference Data Editor

The following sections will provide detailed instructions for using the Reference Data Editor.

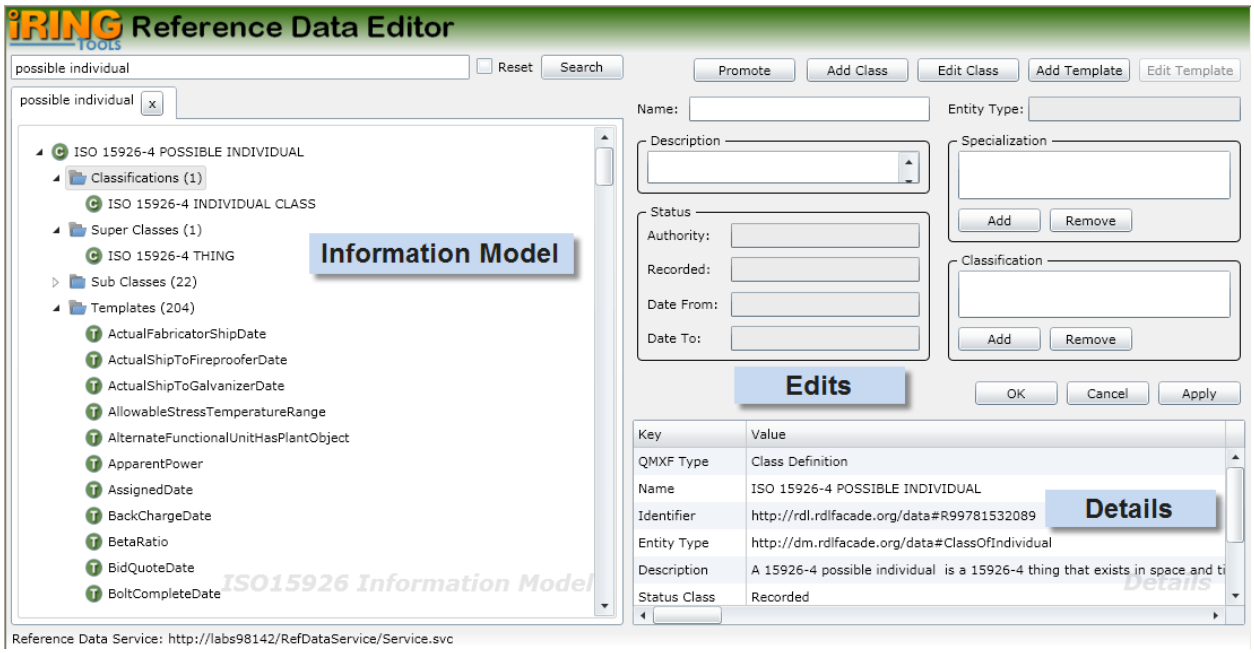
3.1 Purpose

The purpose of the Reference Data Editor is to search and manage reference data in connected repositories.

Note: Known issues prevent add and editing classes and templates with versions 1.2.1 and earlier if using SemWeb (the standard installation). These issues are expected to be resolved in the next release of ***iRINGTools***.

3.2 Reference Data Editor Layout

The mapping editor is arranged into four sections as shown below.



Information Model displays the search results for ISO 15926 classes and templates of connected repositories.

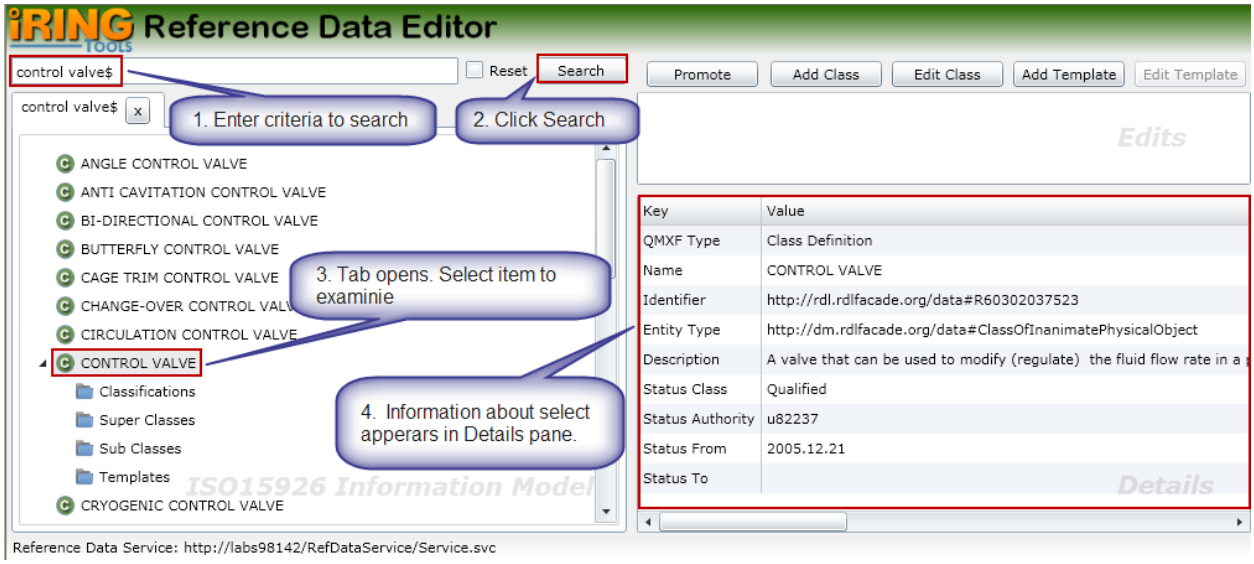
Edits is where you add and edit your classes and templates.

Details displays detail information about what you currently have selected. The content in the Details changes based on the last selected item from the other editor sections.

3.3 Search Reference Data

To search reference data, enter part or all of the class or template to search in the top left text box and then click the Search button.

Note: Only the top 100 search items displayed. If the item you want does not appear in the list, then a more explicit search needs to be done (e.g., entire name using regular expressions such as ^control valve$; the ^ means the expression starts with what follows and the $ means the expression ends with what precedes).



For class results, you can also examine the classifications, super classes, sub classes, and templates. The following is a brief description of each.

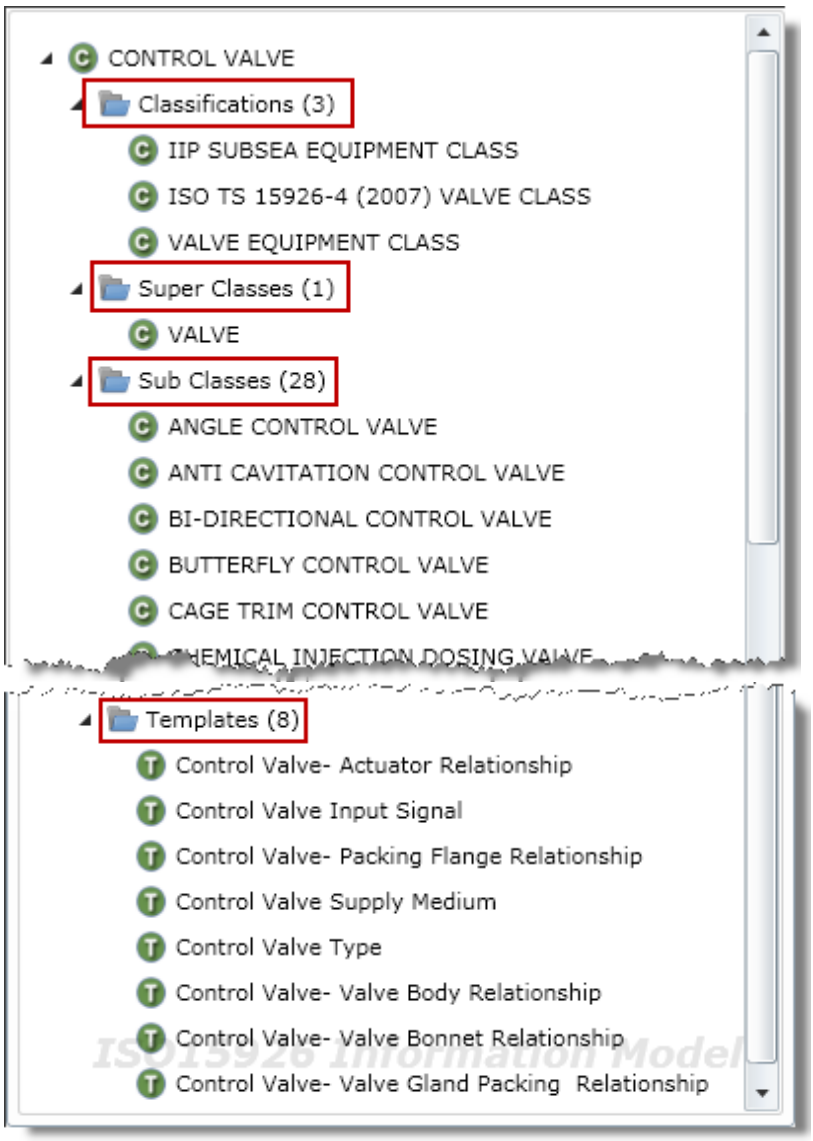
Classifications – The selected class is a member of classes listed in Classifications

Super Classes – The selected class is specialized from the classes listed in Super Classes

Sub Classes – The classes listed in Sub Classes are specialized from the selected class

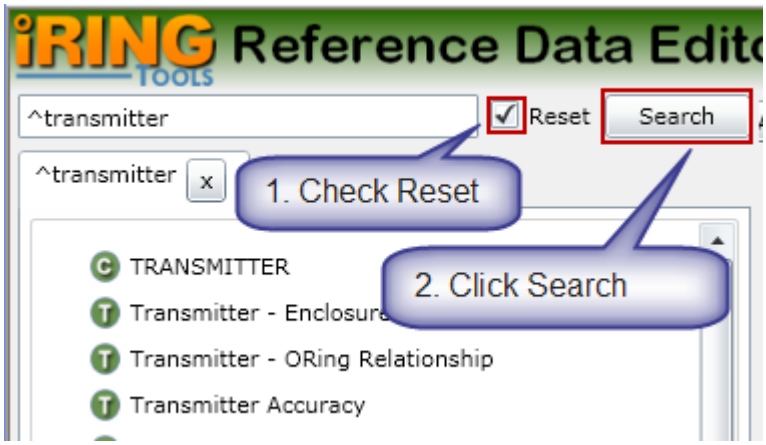
Template – The templates listed have a role type using the selected class

To view, expand the desired class node and select the item to examine. If it contains any items, a count with appear the right and you can expand the item to view the details. In the following example, a CONTROL VALVE has 3 Classifications, 1 Super Class, 28 Sub Classes and 8 Templates.


CONTROL VALVE
Classifications (3)
IIP SUBSEA EQUIPMENT CLASS
ISO TS 15926-4 (2007) VALVE CLASS
VALVE EQUIPMENT CLASS
Super Classes (1)
VALVE
Sub Classes (28)
ANGLE CONTROL VALVE
ANTI CAVITATION CONTROL VALVE
BI-DIRECTIONAL CONTROL VALVE
BUTTERFLY CONTROL VALVE
CAGE TRIM CONTROL VALVE
Templates (8)
Control Valve- Actuator Relationship
Control Valve Input Signal
Control Valve- Packing Flange Relationship
Control Valve Supply Medium
Control Valve Type
Control Valve- Valve Body Relationship
Control Valve- Valve Bonnet Relationship
Control Valve- Valve Gland Packing Relationship

3.4 Reset

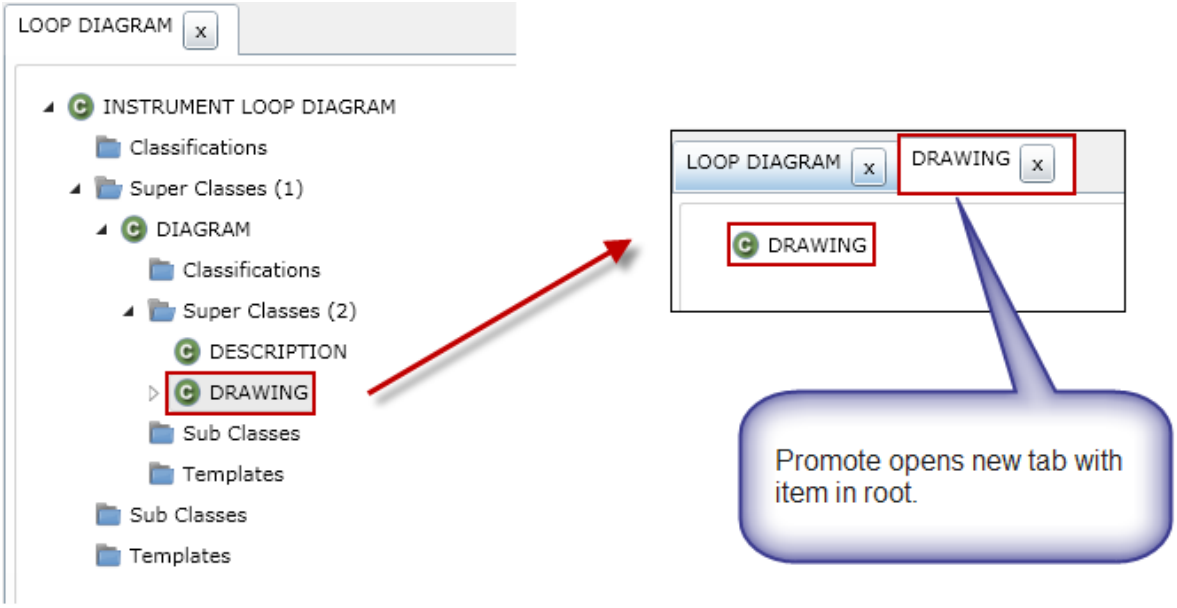
Reference data search items are cached to improve performance. If you know or suspect changes have been made to the reference data repositories, you can update the cache by checking the Reset box and then searching again.



3.5 Promote

The Promote feature lets you open a new tab with the selected itemed as the root. This is useful when you navigate through several nested items and want the selected item to be in the root.

To promote an item, select the item and then click the Promote button. A new tab will open with that item in the root.



3.6 Add Class

To add a new class, perform the following:

1. In the Reference Data Editor, click the Add Class button.
2. Enter a name for the new class.
3. Enter a description for the class.
4. Add classes that the new class is specialized from.
5. Add classifications that the new class is a member of.
6. Click the OK button.
7. The class will be added to the first writable repository in the reference data system.

Note: For version 1.2.1, this feature is only supported for ***iRINGTools*** Sandboxes configured using Joseki and mySQL.

3.7 Edit Class

To edit an existing class, perform the following:

1. In the Reference Data Editor, search for and select the class to be edited.
2. Click the Edit Class button.
3. Change the Name, Description, Specialization and/or Classification as necessary.
4. Click the OK button.
5. The class will be updated if the repository is writable.

Note: For version 1.2.1, this feature is only supported for ***iRINGTools*** Sandboxes configured using Joseki and mySQL.

3.8 Add Template

To add a new temple, perform the following:

1. In the Reference Data Editor, click the Add Template button.
2. Enter a name for the new template.
3. Enter a description for the template.
4. Add roles as necessary in the Role Definition panel.
5. Click the OK button.
6. The template will be added to the first writable repository in the reference data system.

Note: For version 1.2.1, this feature is only supported for ***iRINGTools*** Sandboxes configured using Joseki and mySQL.

3.9 Edit Template

To edit an existing template, perform the following:

1. In the Reference Data Editor, search for and select the template to be edited.
2. Click the Edit Template button.
3. Change the Name, Description, and/or Role Definitions as necessary.
4. Click the OK button.
5. The template will be updated if the repository is writable.

Note: For version 1.2.1, this feature is only supported for ***iRINGTools*** Sandboxes configured using Joseki and mySQL.

4 Application Data Dictionary

The following sections will provide detailed instructions for creating an ***iRINGTools*** application data dictionary using the NHibernate data layer. Other data layers are outside the scope of this guide.

At present, there are two ways to create an application data dictionary:

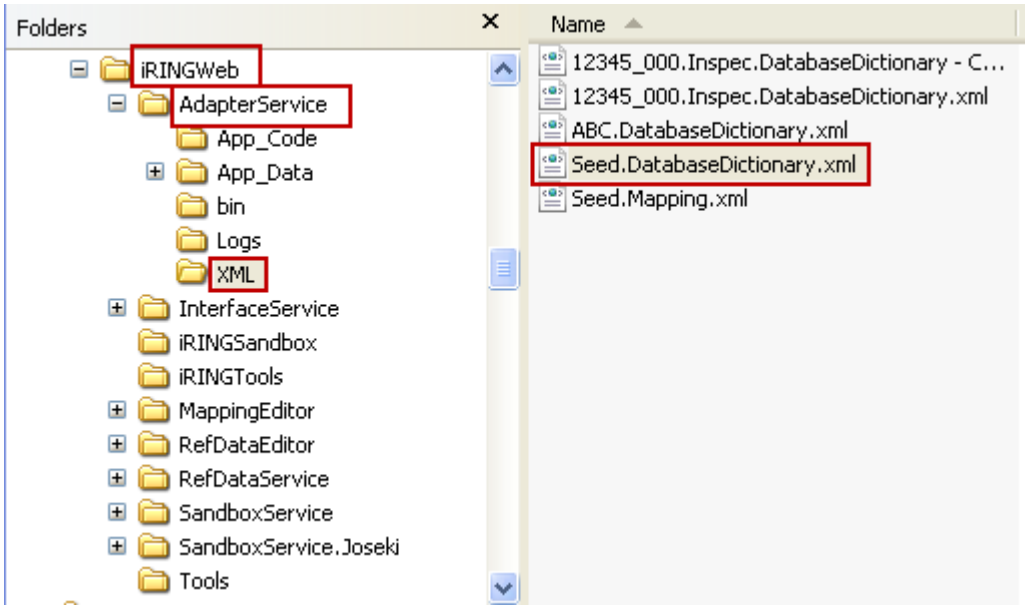
1. Create the file using an XML editor or text editor
2. Using DBDictionaryUtil utility

The second method will save some typing and possibly reduce errors, but will most likely require manual editing as well.

4.1 Manual Creation of Data Dictionary File

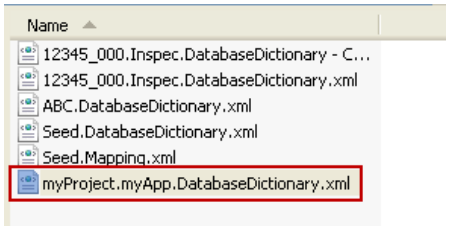
Below are the steps for manual creation of the application data dictionary.

1. In the iRINGWeb\AdapterService\XML folder, locate the file Seed.DatabaseDictionary.xml and copy it to the same folder.

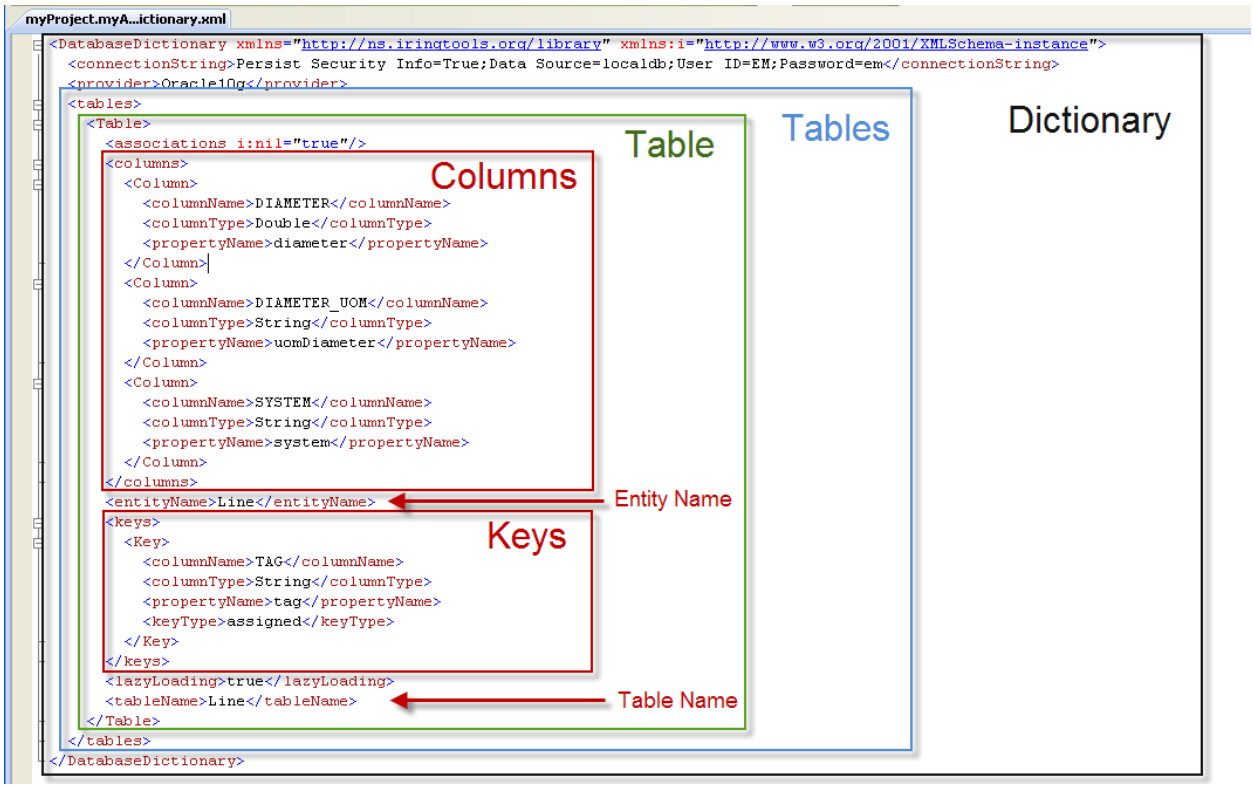


2. Rename the file to <myProject>.<myApp>.DatabaseDictionary.xml

   Where <myProject> is the name of the project (as specified in iRINGTools)
   <myApp> is the name of the application (as specified in iRINGTools)



3. Open the file to edit in an XML or text editor. The XML structure includes Dictionary (for the connection string and tables); tables (for the collection of tables); Table (for each table); columns (for the columns in each Table); Column (for each column in the table that is not a key); entityName (name in iRINGTools); keys (for the keys in each table); Key (for each key in the keys); lazyLoading (should not needed modifying); tableName (for the name of the database table).



Note: When making XML edits, maintain XML syntax as provided in the seed file.

4. Update the connectionString for the Data Source, User ID and Password for your application database.
5. Change the provider to match your application database. The providers must be one of the following:

   - MsSql2005 (for Microsoft SQL Server 2005)
   - MsSql2008 (for Microsoft SQL Server 2008)
   - MsSql2000 (for Microsoft SQL Server 2000)
   - Oracle10g (for Oracle 10)
   - Oracle9i (for Oracle 9)
   - Oracle8i (for Oracle 8)
   - OracleLite (for Oracle Lite)

6. For the first table, change the entityName as you want it to appear in the Mapping Editor (must follow C# variable naming rules; you can prefix names with @ to prevent possible conflicts with compiler keywords)
7. Change the tableName to match the name in your application database.
8. For each column in the table you want to map, update the columnName, columnType and propertyName. The columnName is the name in your application table. The columnType should

correspond to the database column type, but specified using a valid C# data type. The propertyName is the name you want to appear in the Mapping Editor and must follow C# variable naming rules; you can prefix names with @ to prevent possible conflicts with compiler keywords.

9. Update the Keys for the table. These are the columns that define uniqueness and are used for iRINGTools exchanges. Follow the same rules as for columns. The keyType property will always have the value “assigned”.

   Caution: A column defined as a key should not appear in the columns list. Remove any column definitions that are keys.

10. Repeat the above for each table in the dictionary by copying the contents <Table> … </Table> and pasting below an existing table. When done, the tables selection should resemble the following:

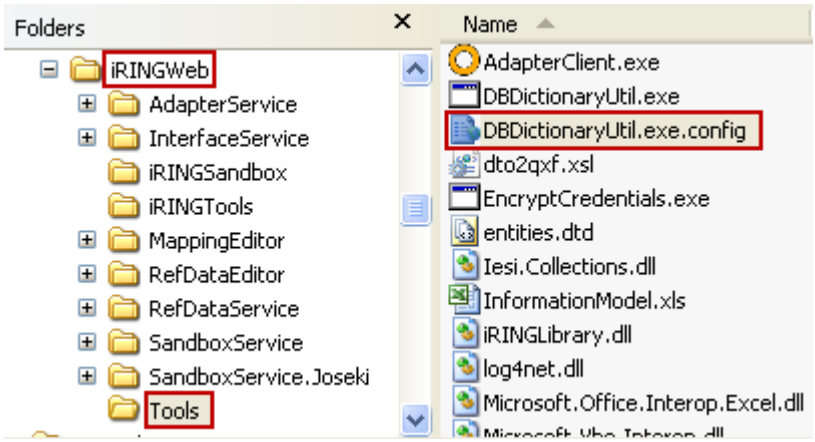
```
<tables>
   <Table>
        Contents…
   </Table>
   <Table>
        Contents…
   </Table>
</tables>
```

11. Save the file and close.
12. Go to the Upload Data Dictionary section to complete the setup of the application data dictionary.

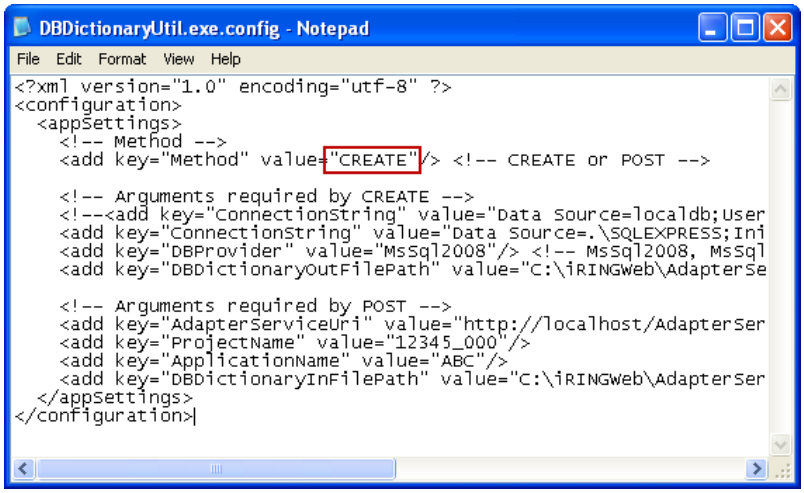
4.2 Automation Creation of Data Dictionary File

Below are the steps for automatic creation of the application data dictionary.

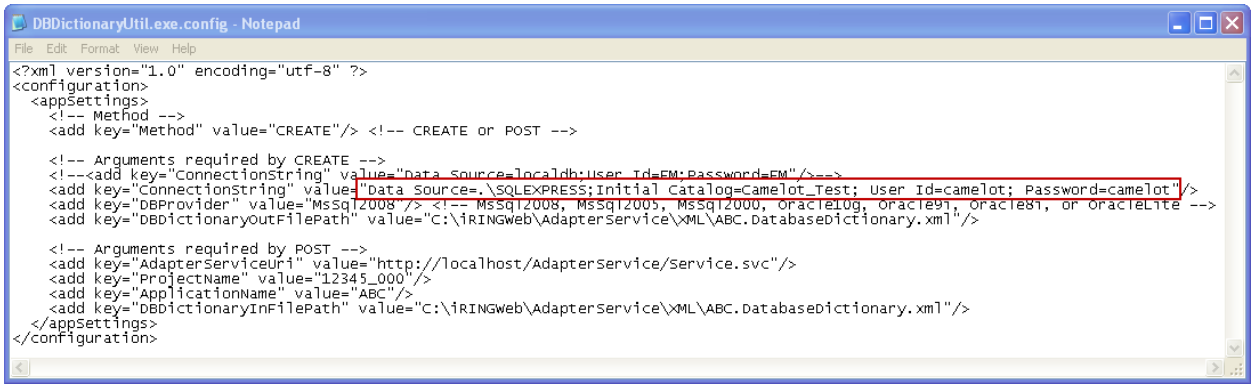
1. In the iRINGWeb\Tools folder, open the file DBDictionaryUtil.exe.config in a text editor such as Notepad.



2. Change the Method value to CREATE.

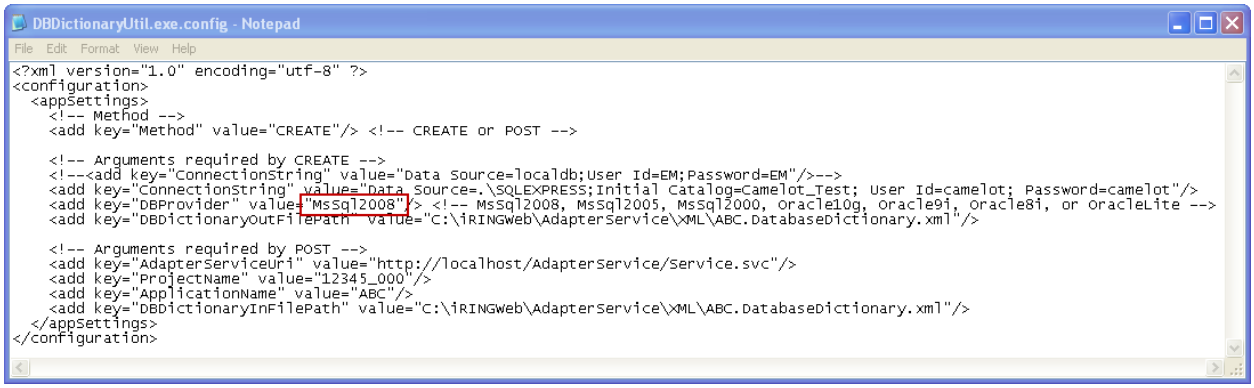


3. Change the ConnectionString value for the Data Source, User ID and Password of your application database.



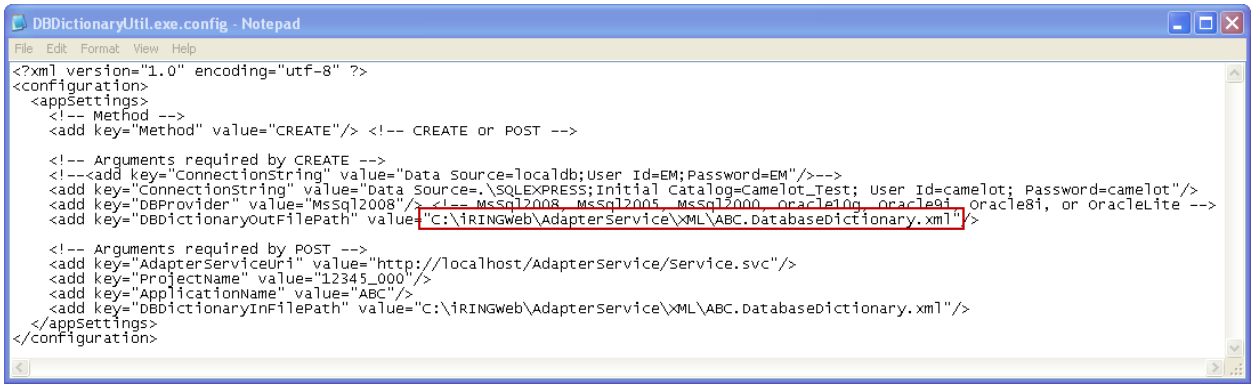
4. Change the DBProvider value to match your application database. The providers must be one of the following:

- MsSql2005 (for Microsoft SQL Server 2005)
- MsSql2008 (for Microsoft SQL Server 2008)
- MsSql2000 (for Microsoft SQL Server 2000)
- Oracle10g (for Oracle 10)
- Oracle9i (for Oracle 9)
- Oracle8i (for Oracle 8)
- OracleLite (for Oracle Lite)



5. Change the DBDictionaryOutFilePath value to the iRINGWeb\AdapterService\XML folder and the file named as <myProject>.<myApp>.DatabaseDictionary.xml

   Where <myProject> is the name of the project (as specified in iRINGTools)
   <myApp> is the name of the application (as specified in iRINGTools)



6. Save the file and close it.

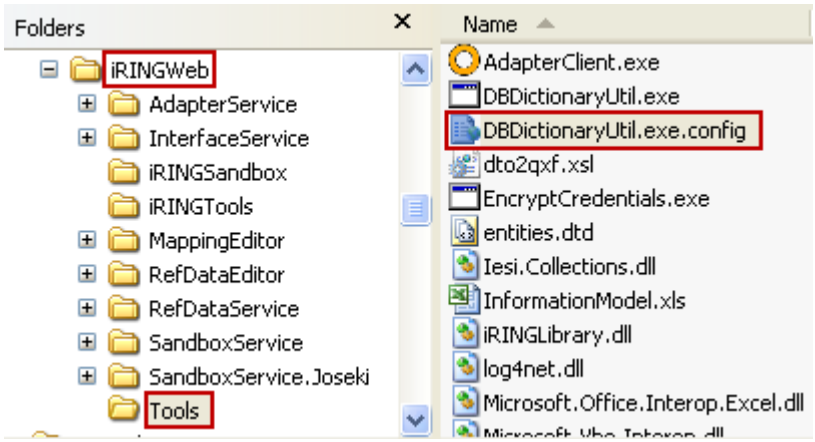
7. Execute the utility DBDictionaryUtil.exe located in the iRINGWeb\Tools folder.

8. The resulting file will contain every table and every column in the specified database. Edit the resulting file (defined DBDictionaryOutFilePath step) to contain on the required tables and columns. Refer to the Manual Creation of Data Dictionary File section on how to manually edit the file.

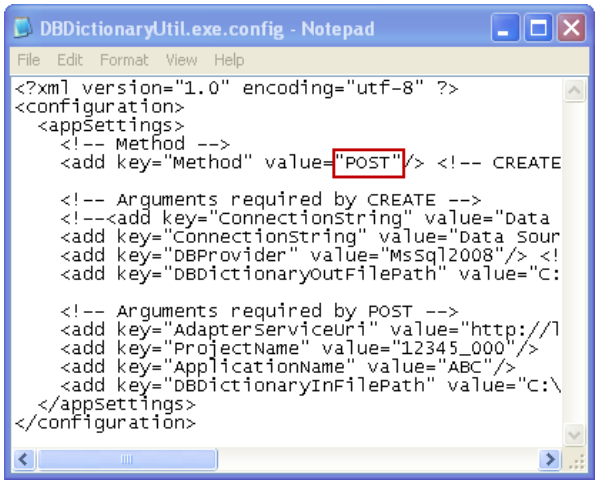
4.3 Post Data Dictionary File

Below are the steps for posting the data dictionary file.

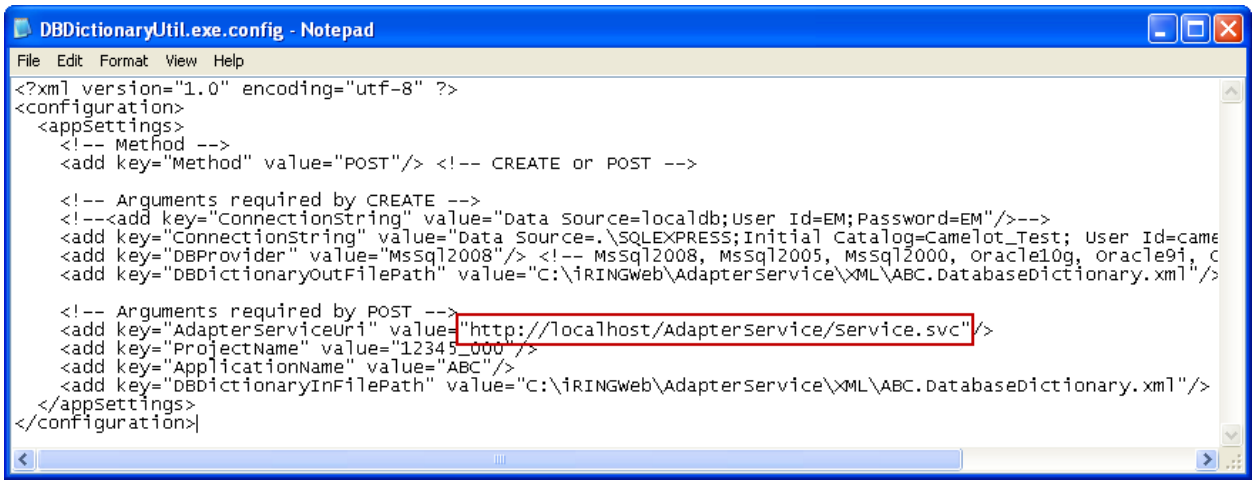
1. In the iRINGWeb\Tools folder, open the file DBDictionaryUtil.exe.config in a text editor such as Notepad.



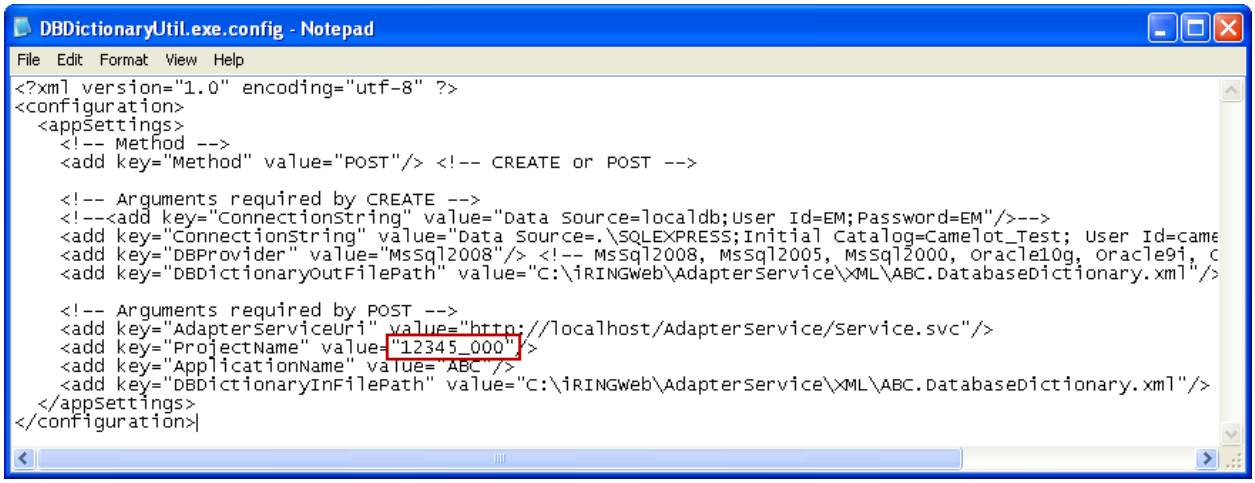
2. Change the Method value to POST.



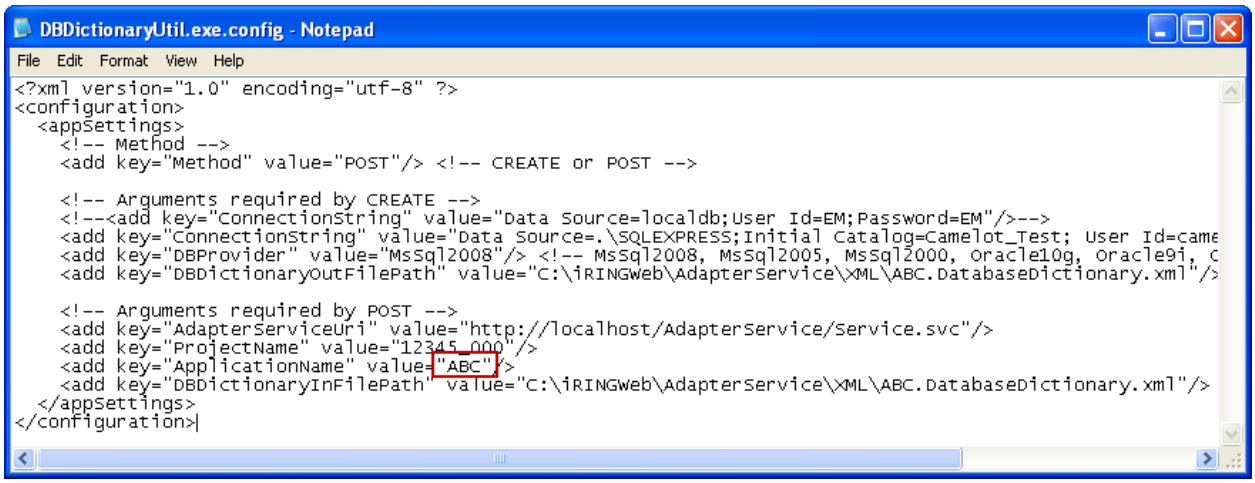
3. Change the AdapterServiceUri value to match your iRINGTools setup.



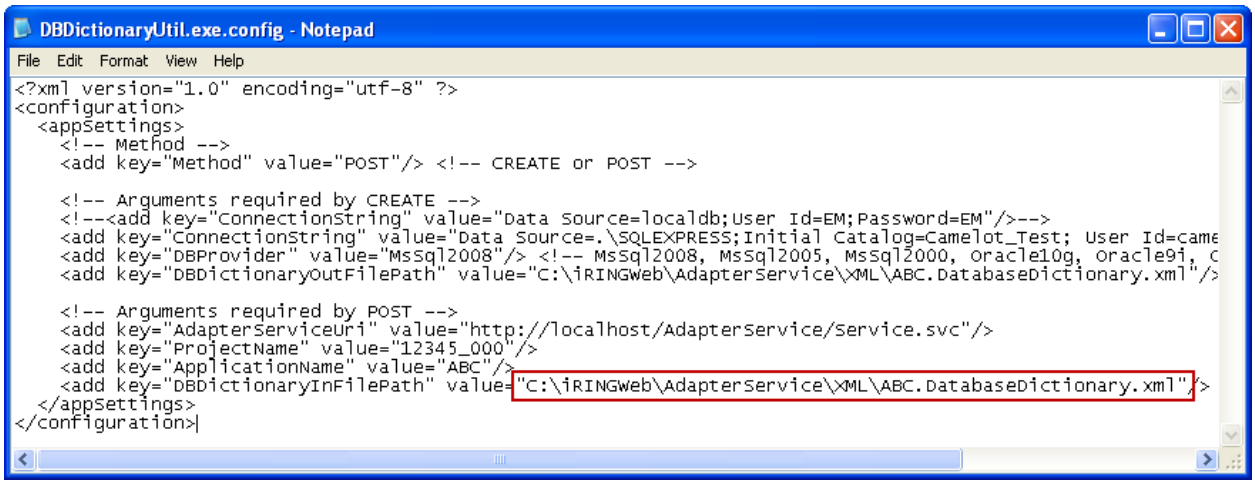
4. Change the ProjectName value to match your iRINGTools setup.



5. Change the ApplicationName value to match your iRINGTools setup.



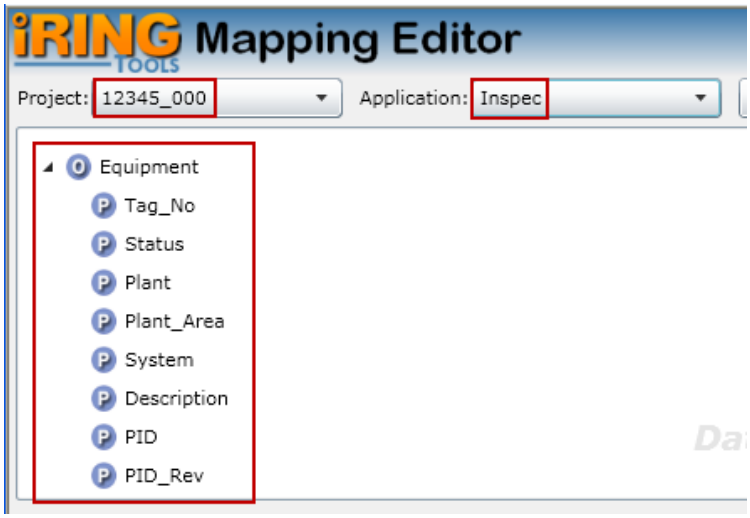
6. Change the DBDictionaryInFilePath value to the file with the application data dictionary definition created earlier (most likely in the iRINGWeb\AdapterService\XML folder).



7. Save the file and close it.

8. Execute the utility DBDictionaryUtil.exe located in the iRINGWeb\Tools folder. You should receive a message that the post was successful.

9. Open the iRINGTools mapping editor and verify:
   - The project can be selected.
   - The application can be selected.
   - The tables and columns appear in the Data Dictionary section of the Mapping Editor as specified.



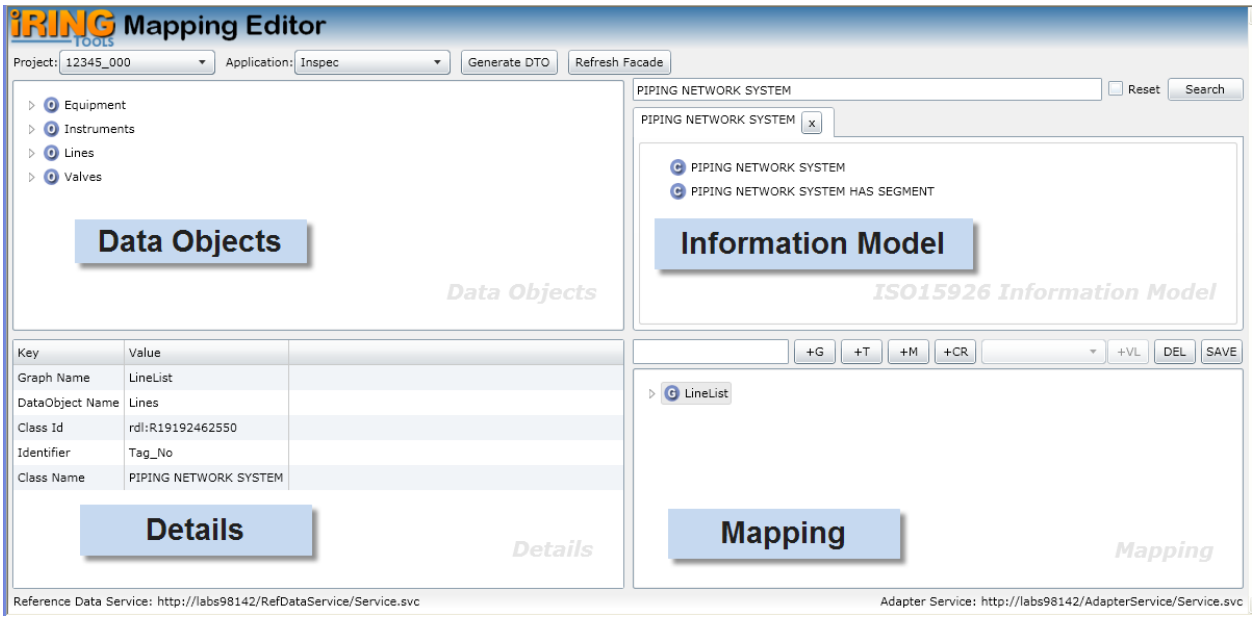
5 Mapping Editor

The following sections will describe how to map application data using the ***iRINGTools*** Mapping Editor.

Note: The techniques and methodology outlined in this section are not the only way to map ISO 15926 reference data to application data, but does represent one way that is presently being used.

5.1 Mapping Editor Layout

The mapping editor is arranged into four sections as shown below.



Data Objects represents the data dictionary of your application database.

Information Model is the ISO 15926 classes and templates available for you to map.

Mapping is where you build and edit you data map.

Details displays detail information about what you currently have selected. The content in the Details changes based on the last selected item from the other editor sections.

5.2 Information Model

The Information Model section lets you search and select ISO 15926 classes and templates to use in mapping. It operates the same as described in the Search Reference Data portion of this manual. Refer to that section for details on how to search and select ISO 15926 classes and templates.

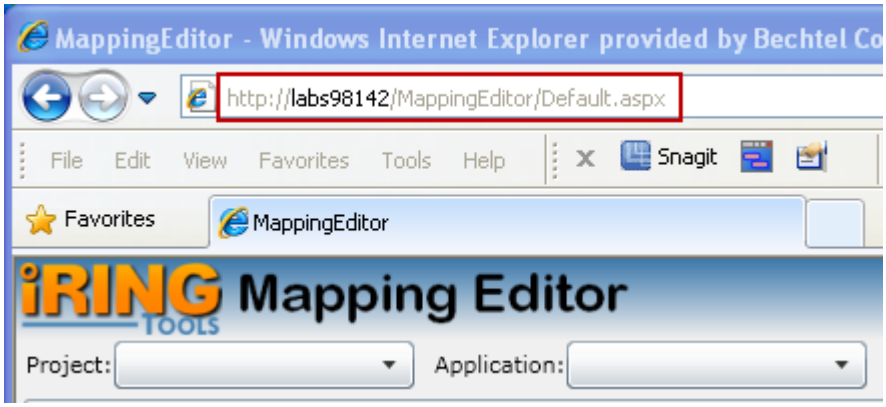
5.3 Open Project and Application

Open the project and application to map by performing the following:

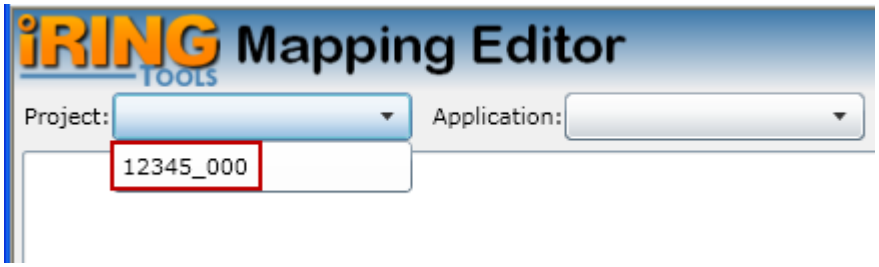
1. Ensure the data dictionary for the application has been generated for the application per Application Data Dictionary section in this document.
2. Using your browser, open the mapping editor by entering that address:

   http://<hostname>/MappingEditor/Default.aspx

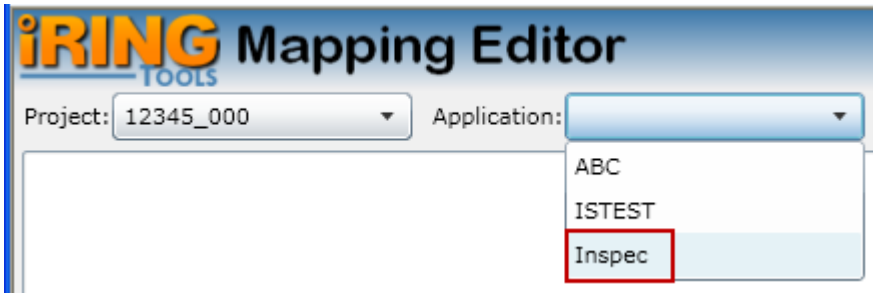
   Where <hostname> is the name of the server hosting the iRINGTools mapping editor.



3. Select the project from the Project dropdown list.



4. Select the application from the Application dropdown list.

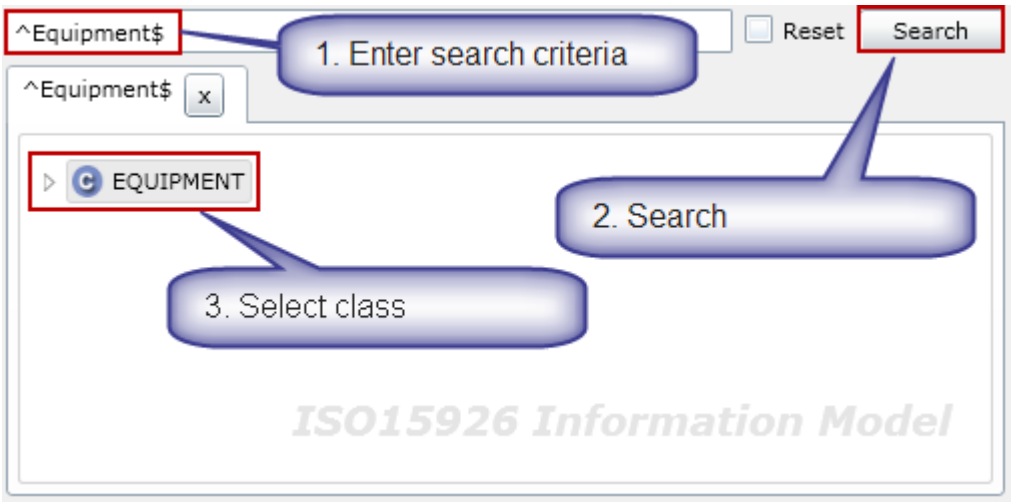


5.4 Create Graph Map

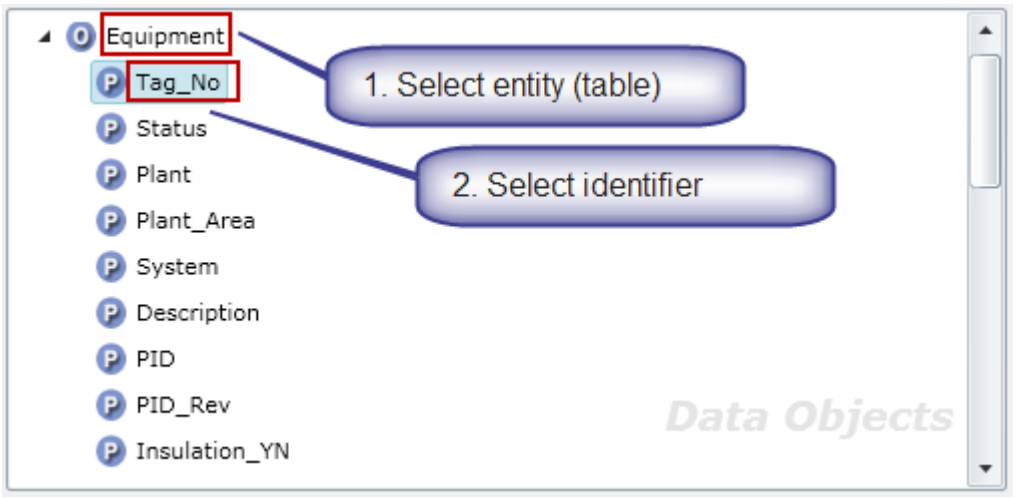
A graph map specifies how ISO 15926 reference data is mapped to your application database and is used in data exchanges. Create a graph map by performing the following:

Note: Prevent possible data loss by frequently saving your graph by clicking the SAVE button on the Mapping toolbar.

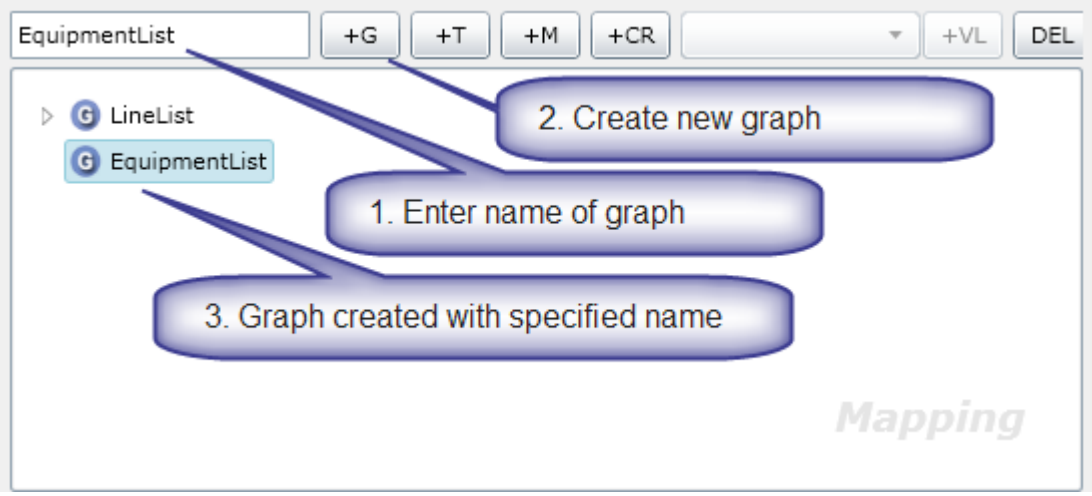
1. Determine the class that aligns with the data map being created, search for it in the Information Model pane, and then select the result.



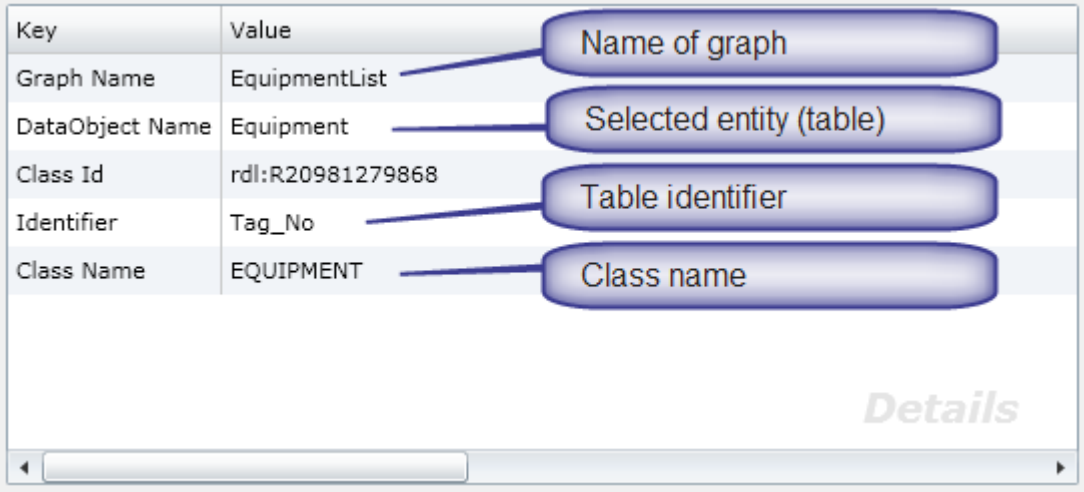
2. In the Data Objects pane, select the entity (table) being mapped and its identifier (i.e., the column that uniquely identifies the table such as tag number).



3. In the Mapping pane, enter the name of the graph and create a new graph map by clicking on the +G button on the Mapping toolbar.



4. In the Mapping pane, click on the graph name. The graph details appear in the Details pane.



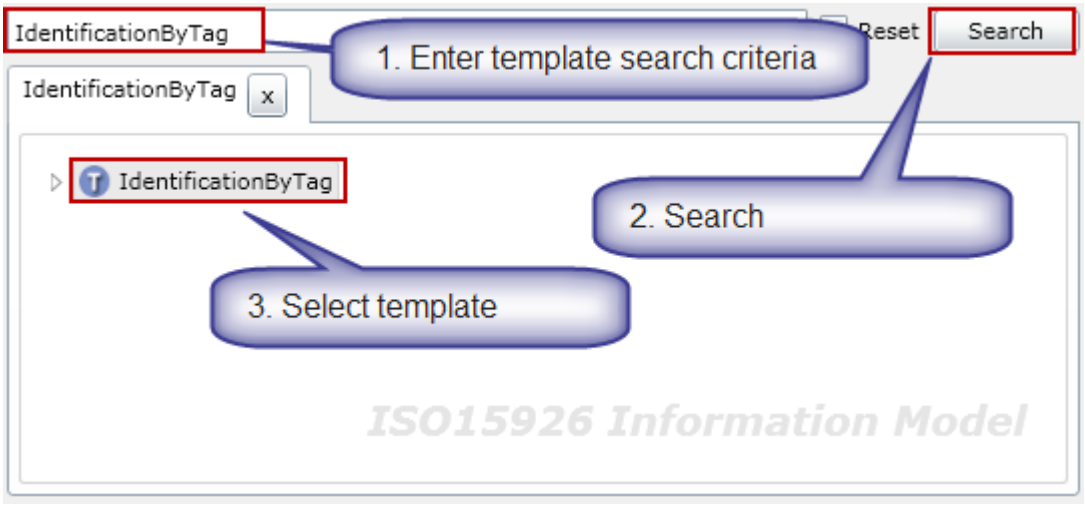
5. Save the graph by clicking on the Save button on the Mapping pane toolbar.

Note: If you do not see the Save button on the toolbar, you may need to resize your browser until it appears.

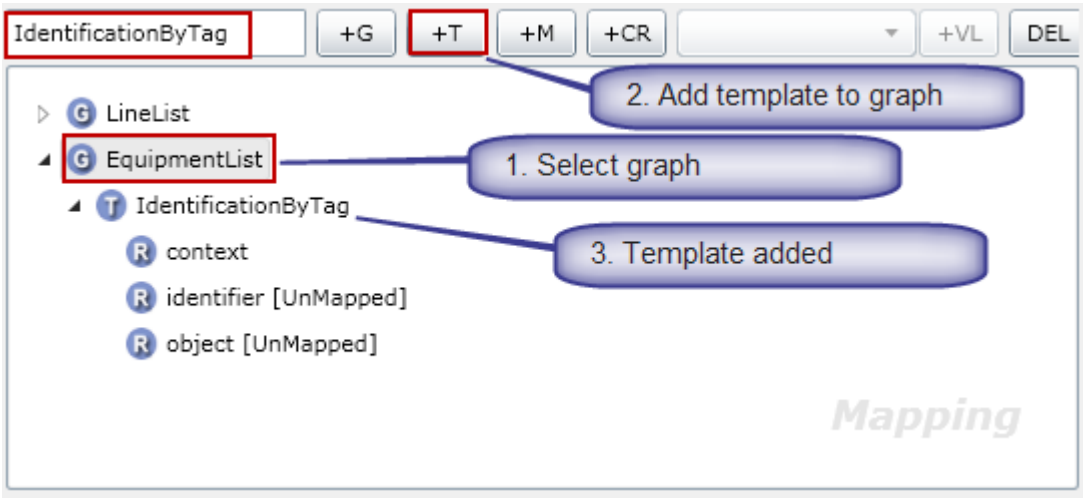
5.5 Mapping Graph Identifier

To successfully exchange data with your application database, the identifier for the entity (table) needs to be mapped. This is the same identifier defined when creating the graph map, but now needs to be explicitly mapped in the graph – typically using a template. Map an identifier by performing the following:

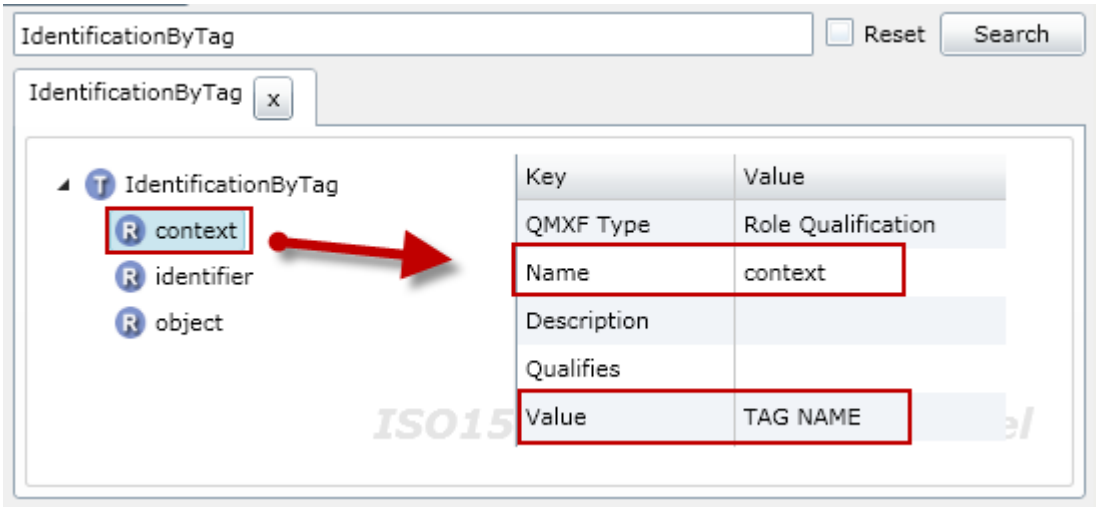
1. Determine the template that will be used to map the identifier, search for it in the Information Model pane, and then select the result.



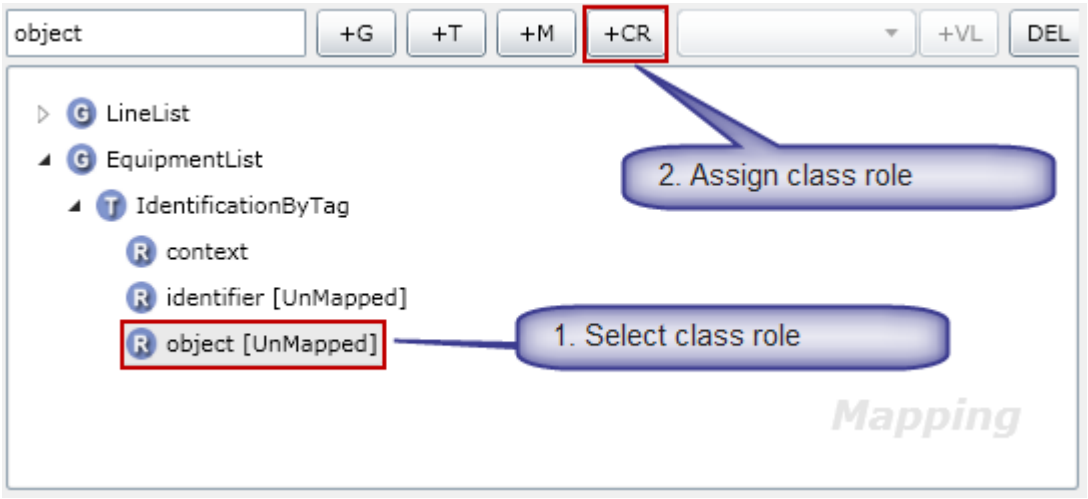
2. The selected template name appears in the Mapping pane. Select the graph to map the identifier and then click the +T button on the toolbar.



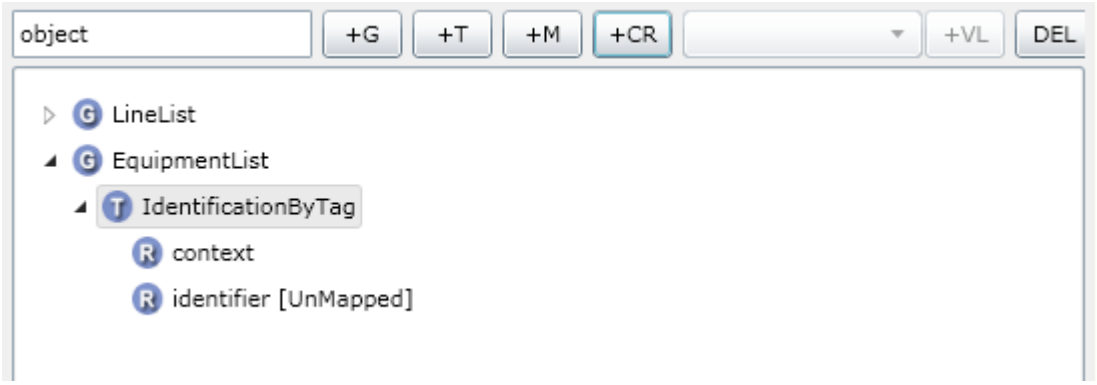
3. Expand the graph to select the added template. Expand the added template to display the roles.
4. Notice the role “context” is already mapped because it is a fixed role. (It is mapped because there is no [UnMapped] status next to it). The role “context”, in this case, has been specialized to the role TAG NAME. (Hence, the purpose for the template.)



5. Every template will have a class role (which is a role pointing to the class). In this template, the role “object” is point to the class “TAGGED ITEM”. Thus, the role “object” is the class role for the IdentificationByTag template.
6. In the Mapping pane, assign the role “object” to the class role by selecting the role “object” and then clicking the +CR button on the toolbar.

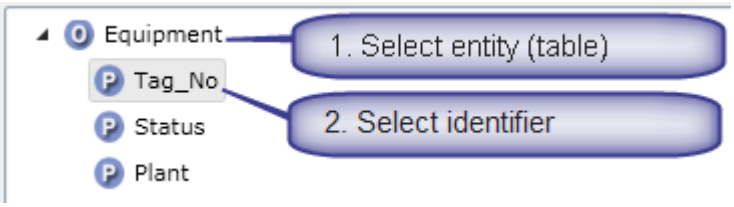


7. Note the role “object” immediately disappears from the graph template role list.

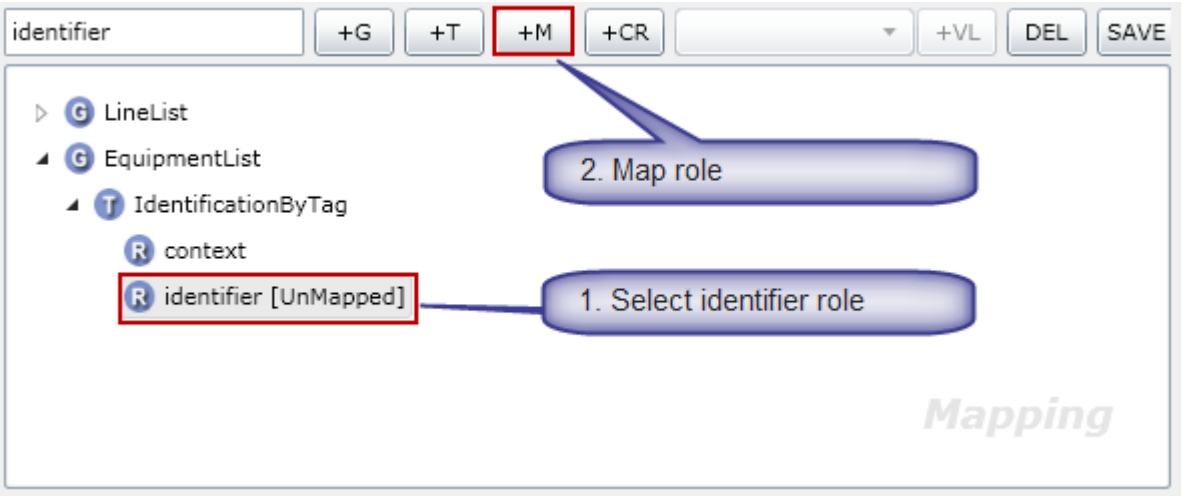


The final role “identifier” now needs to be mapped to the application entity (table) identifier column.

8. In the Data Objects pane, select the entity (table) identifier column.



9. In the Mapping pane, map the role to application by selecting the role “identifier” then clicking the +M button on the toolbar.



10. The role identifier is now mapped because there is no [UnMapped] status next to it.

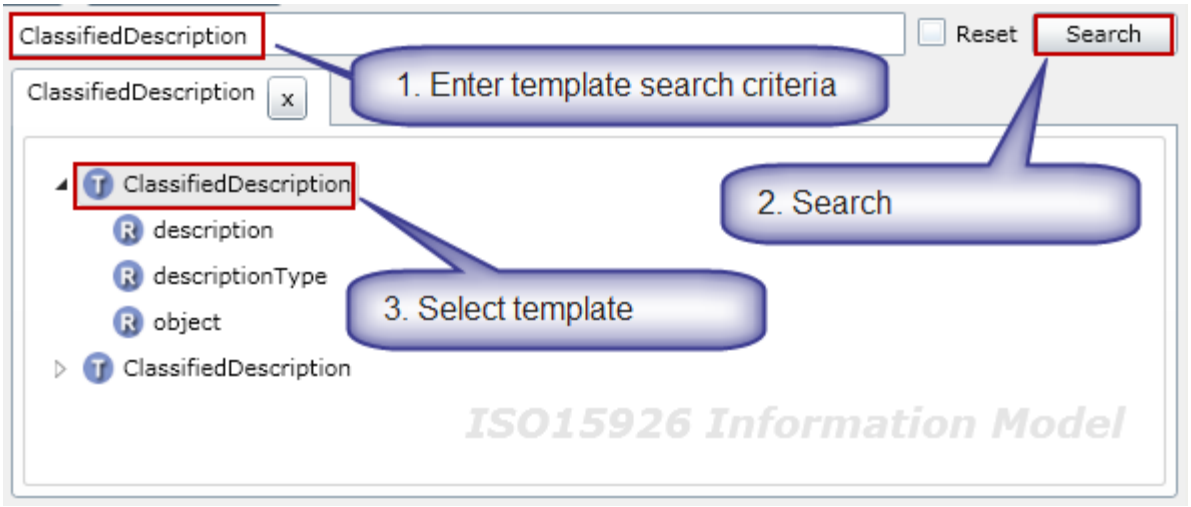
11. This completes the mapping for the IdentificationByTag template. Save the graph by clicking on the Save button on the Mapping pane toolbar.

5.6 Mapping with Property Templates

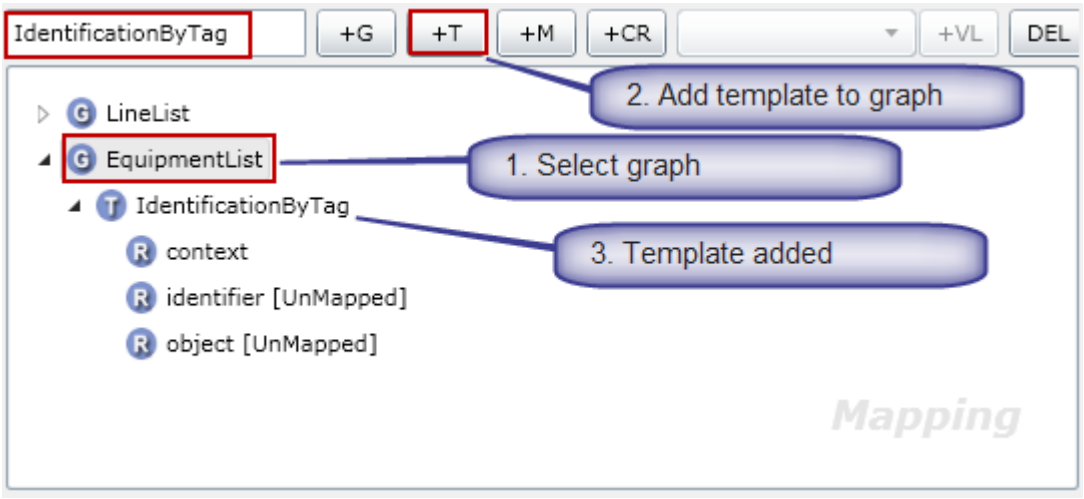
A property template maps an ISO 15926 reference data class to your application database.

Map a property template by performing the following:

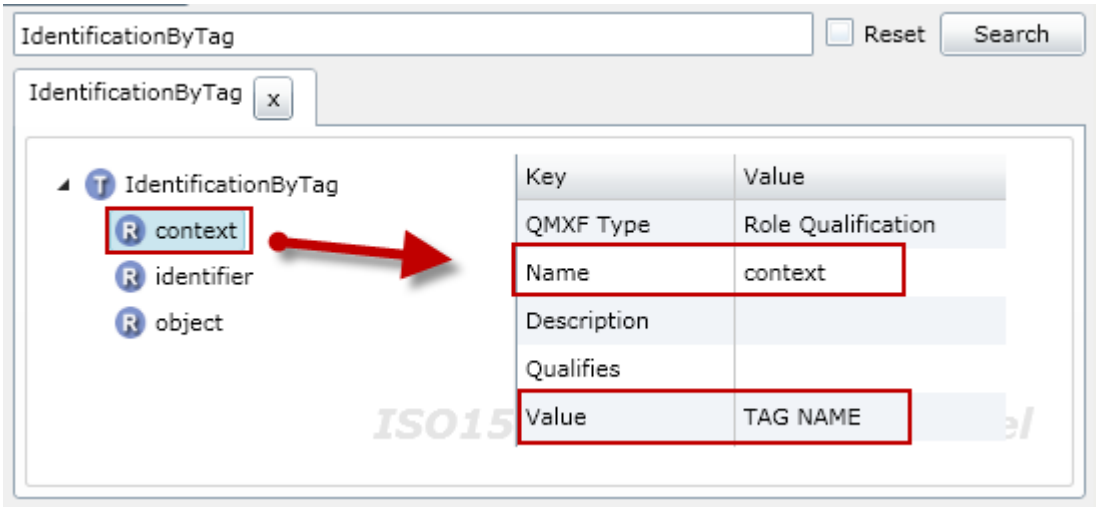
1. Determine the property template that will be used, search for it in the Information Model pane, and then select the result.



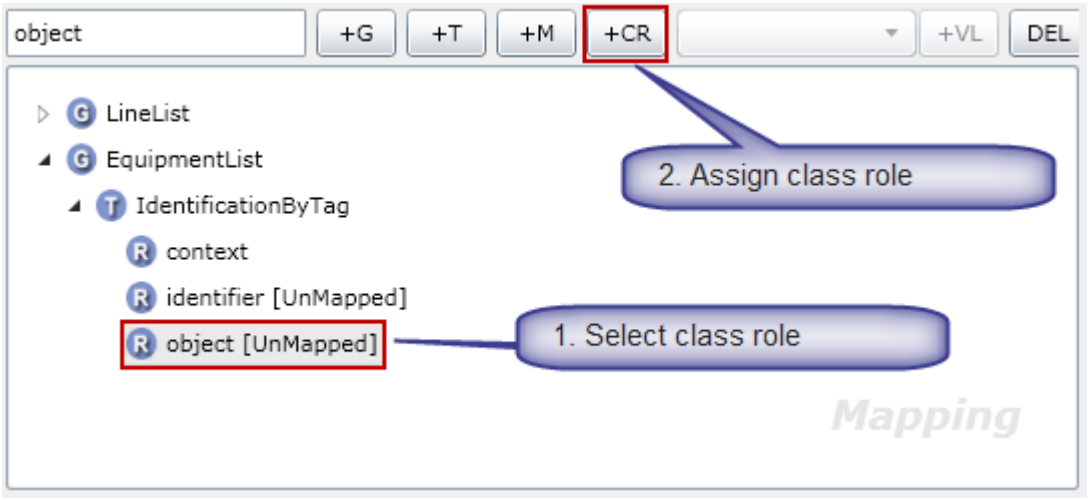
2. The selected template name appears in the Mapping pane. Select the graph to map the identifier and then click the +T button on the toolbar.



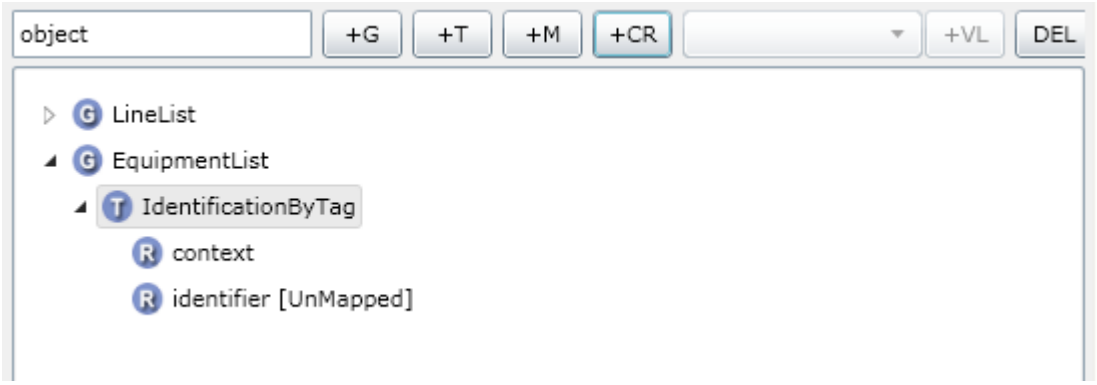
3. Expand the graph to select the added template. Expand the added template to display the roles.
4. Notice the role "context" is already mapped because it is a fixed role. (It is mapped because there is no [UnMapped] status next to it). The role "context", in this case, has been specialized to the role TAG NAME. (Hence, the purpose for the template.)



5. Every template will have a class role (which is a role pointing to the class). In this template, the role "object" is point to the class "TAGGED ITEM". Thus, the role "object" is the class role for the IdentificationByTag template.
6. In the Mapping pane, assign the role "object" to the class role by selecting the role "object" and then clicking the +CR button on the toolbar.

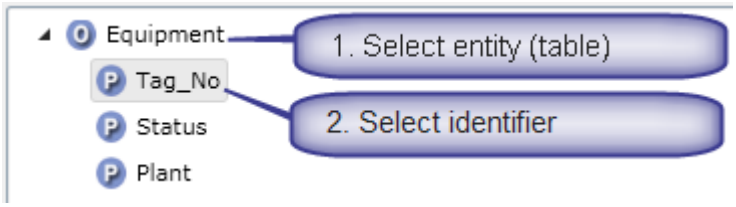


7. Note the role “object” immediately disappears from the graph template role list.

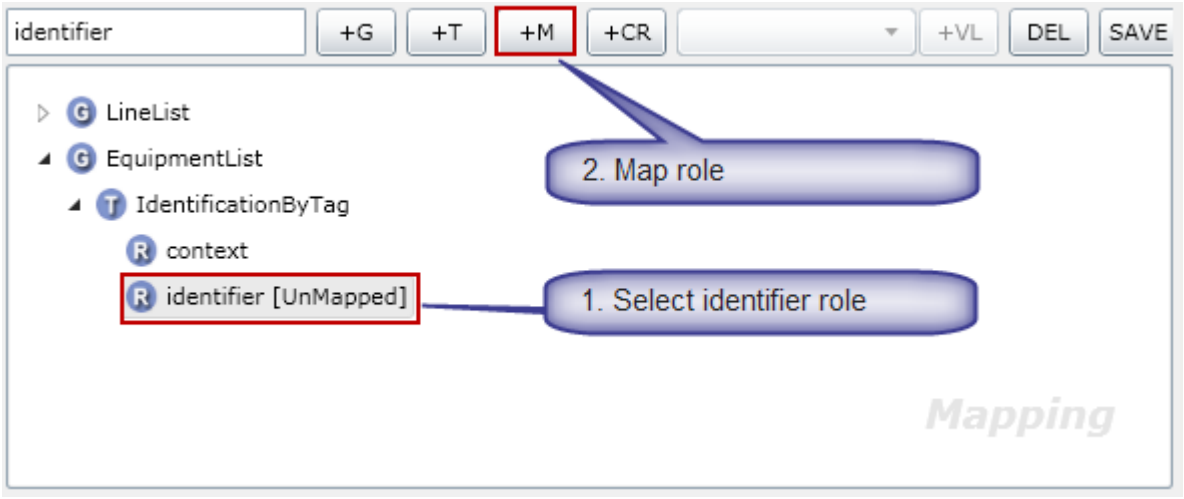


The final role “identifier” now needs to be mapped to the application entity (table) identifier column.

8. In the Data Objects pane, select the entity (table) identifier column.



9. In the Mapping pane, map the role to application by selecting the role “identifier” then clicking the +M button on the toolbar.



10. The role identifier is now mapped because there is no [UnMapped] status next to it.

11. This completes the mapping for the IdentificationByTag template. Save the graph by clicking on the Save button on the Mapping pane toolbar.

5.7 Mapping with Value Lists

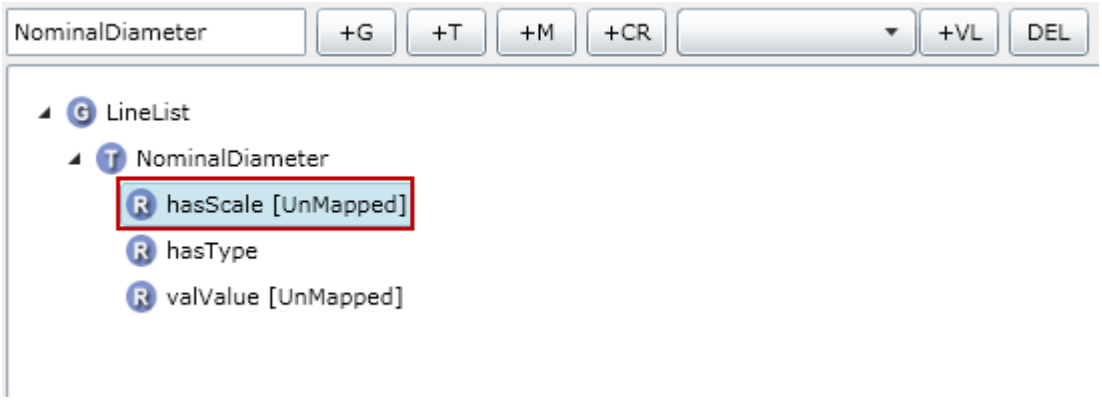
Value Lists provide a way to set values from a predefined list. The classic example is the units of measure for various engineering parameters (e.g., pressure and temperature units). In addition to specifying the list of values, the system can be used to translate values at different endpoints. For example, one endpoint may express pressure as kPa and another may express pressure as kilopascal. Both are valid and data exchanges will properly translate the value at the endpoint. Note that value lists will not convert numerical values from one measurement system to another (e.g., 1 PSI to 6.894 kPa).

Setup Note: At present, Value Lists must be added to application mapping files manually using a text editor. To see how this is done and to get started, examine the file Seed.Mapping.xml located in the ***iRINGTools*** iRINGWeb\AdapterService\XML folder. The following is an excerpt from that file.

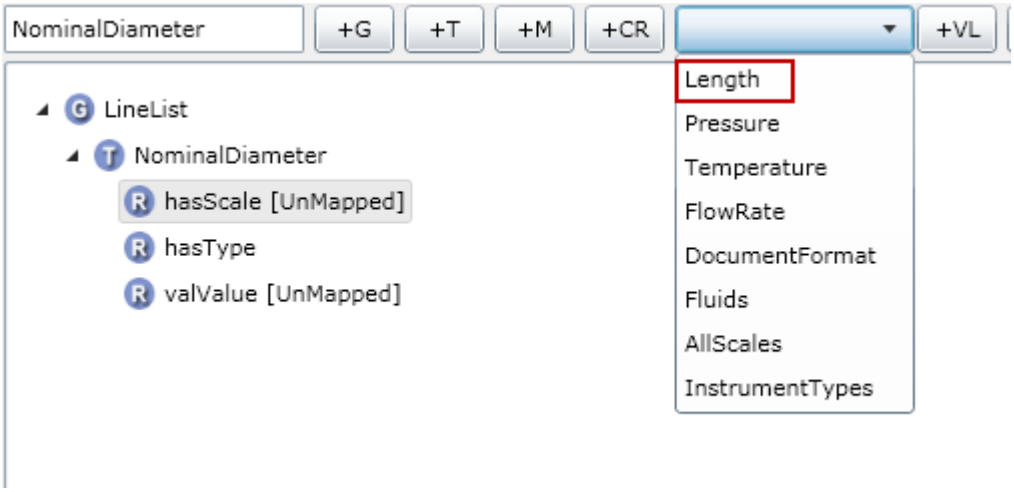

```
<?xml version="1.0" encoding="utf-8"?>
<Mapping xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <GraphMaps>
    <GraphMap name="Lines" classId="rdl:R19192462550" identifier="tag">
      <TemplateMaps>
        <TemplateMap type="Property" templateId="tpl:R31178164691" name="ClassifiedIdentification" classRole="tpl:R47421421239">
          <RoleMaps>
            <RoleMap roleId="tpl:R17963188524" name="identifier" propertyName="tag"/>
            <RoleMap roleId="tpl:R43718740838" name="identificationType" value="rdl:R92093626759" valueList=""/>
          </RoleMaps>
        </TemplateMap>
      </TemplateMaps>
      <DataObjectMaps>
        <DataObjectMap name="Line"/>
      </DataObjectMaps>
    </GraphMap>
  </GraphMaps>
  <ValueMaps>
    <ValueMap valueList="Length" internalValue="MILLIMETRE" modelURI="rdl:R52054275374" />
    <ValueMap valueList="Length" internalValue="METRE" modelURI="rdl:R36970488283" />
    <ValueMap valueList="Length" internalValue="KILOMETRE" modelURI="rdl:R28364552281" />
    <ValueMap valueList="Length" internalValue="INCH" modelURI="rdl:R34755432270" />
    <ValueMap valueList="Length" internalValue="FOOT" modelURI="rdl:R82749412674" />
    <ValueMap valueList="Length" internalValue="MILE" modelURI="rdl:R24581249510" />
    <ValueMap valueList="Pressure" internalValue="PASCAL" modelURI="rdl:R98787754267" />
    <ValueMap valueList="Pressure" internalValue="KILOPASCAL" modelURI="rdl:R63639099552" />
    <ValueMap valueList="Pressure" internalValue="POUND PER SQUARE INCH" modelURI="rdl:R14495957831" />
    <ValueMap valueList="Pressure" internalValue="POUND PER SQUARE INCH ABSOLUTE" modelURI="rdl:R71198757784" />
    <ValueMap valueList="Pressure" internalValue="POUND PER SQUARE INCH GAUGE" modelURI="rdl:R46601209521" />
    <ValueMap valueList="Pressure" internalValue="BAR" modelURI="rdl:R83490506551" />
    <ValueMap valueList="Temperature" internalValue="KELVIN" modelURI="rdl:R53109662405" />
    <ValueMap valueList="Temperature" internalValue="DEGREE CELSIUS" modelURI="rdl:R74877992703" />
    <ValueMap valueList="Temperature" internalValue="DEGREE FAHRENHEIT" modelURI="rdl:R45708437750" />
    <ValueMap valueList="FlowRate" internalValue="TONNE PER HOUR" modelURI="rdl:R97397194940" />
    <ValueMap valueList="FlowRate" internalValue="KILOGRAM PER HOUR" modelURI="rdl:R40880693827" />
    <ValueMap valueList="FlowRate" internalValue="KILOGRAM PER MINUTE" modelURI="rdl:R20832134706" />
    <ValueMap valueList="FlowRate" internalValue="LITRE PER HOUR" modelURI="rdl:R72748932327" />
    <ValueMap valueList="FlowRate" internalValue="LITRE PER MINUTE" modelURI="rdl:R23195462092" />
    <ValueMap valueList="FlowRate" internalValue="CUBIC METRE PER MINUTE" modelURI="rdl:R66068064293" />
    <ValueMap valueList="FlowRate" internalValue="CUBIC METRE PER SECOND" modelURI="rdl:R94408067034" />
    <ValueMap valueList="FlowRate" internalValue="UK TON PER HOUR" modelURI="rdl:R53352916327" />
    <ValueMap valueList="FlowRate" internalValue="US TON PER HOUR" modelURI="rdl:R14730334384" />
    <ValueMap valueList="FlowRate" internalValue="POUND MASS PER HOUR" modelURI="rdl:R51206962402" />
    <ValueMap valueList="FlowRate" internalValue="POUND MASS PER MINUTE" modelURI="rdl:R94787789208" />
    <ValueMap valueList="FlowRate" internalValue="UK GALLON PER HOUR" modelURI="rdl:R45867315335" />
    <ValueMap valueList="FlowRate" internalValue="US GALLON PER HOUR" modelURI="rdl:R57993485640" />
    <ValueMap valueList="FlowRate" internalValue="UK GALLON PER MINUTE" modelURI="rdl:R15856845349" />
    <ValueMap valueList="FlowRate" internalValue="US GALLON PER MINUTE" modelURI="rdl:R90193901690" />
    <ValueMap valueList="FlowRate" internalValue="CUBIC FOOT PER MINUTE" modelURI="rdl:R88625733841" />
    <ValueMap valueList="FlowRate" internalValue="CUBIC FOOT PER SECOND" modelURI="rdl:R68602189675" />
    <ValueMap valueList="DocumentFormat" internalValue="MIME TYPE APPLICATION/PDF" modelURI="rdl:R15309906377" />
  </ValueMaps>
  <Version>1.00.00</Version>
</Mapping>
```

To map a value list in a graph map, perform the following:

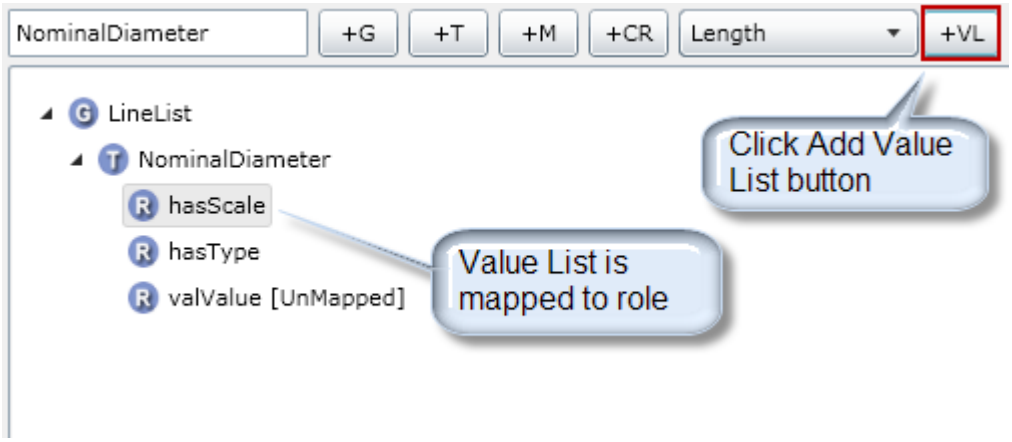
1. In the Mapping pane, select the template role that will be assigned a Value List map.



2. Select the Value List dropdown and select the desired Value List item.

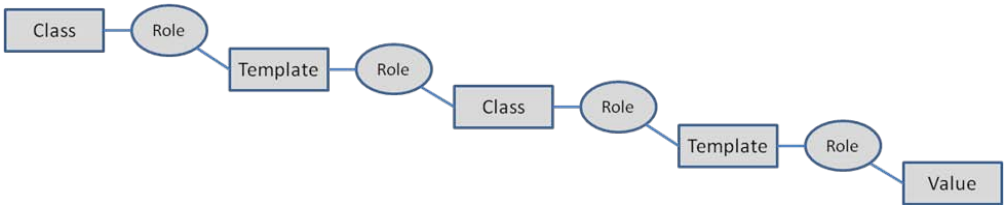


3. Click the +VL button to map the specified Value List item.



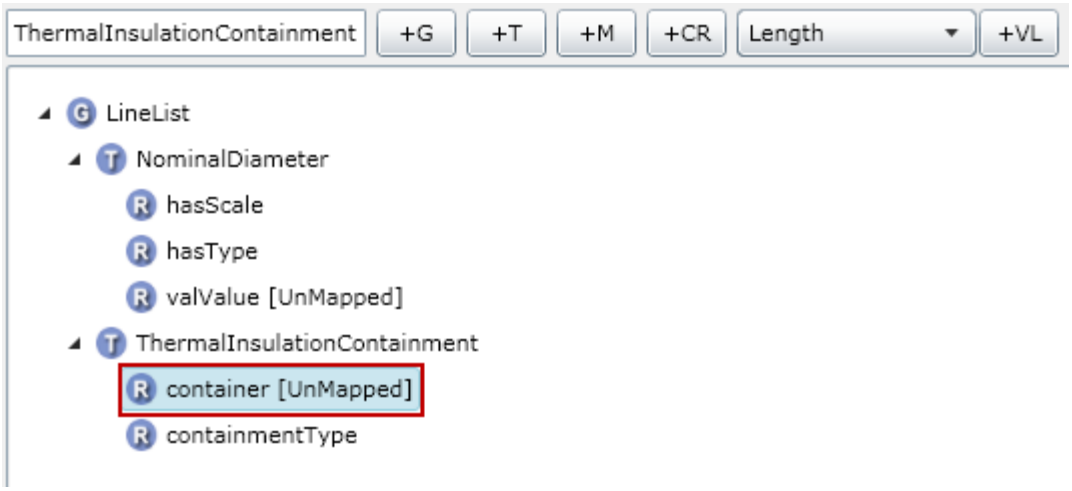
5.8 Mapping with Relationship Templates

Relationship templates are templates whose purpose is to relate one class to another. An example is a process line and line insulation. Each, as a class, has their own characteristics (or properties), but they are also related (i.e., a line may have insulation).

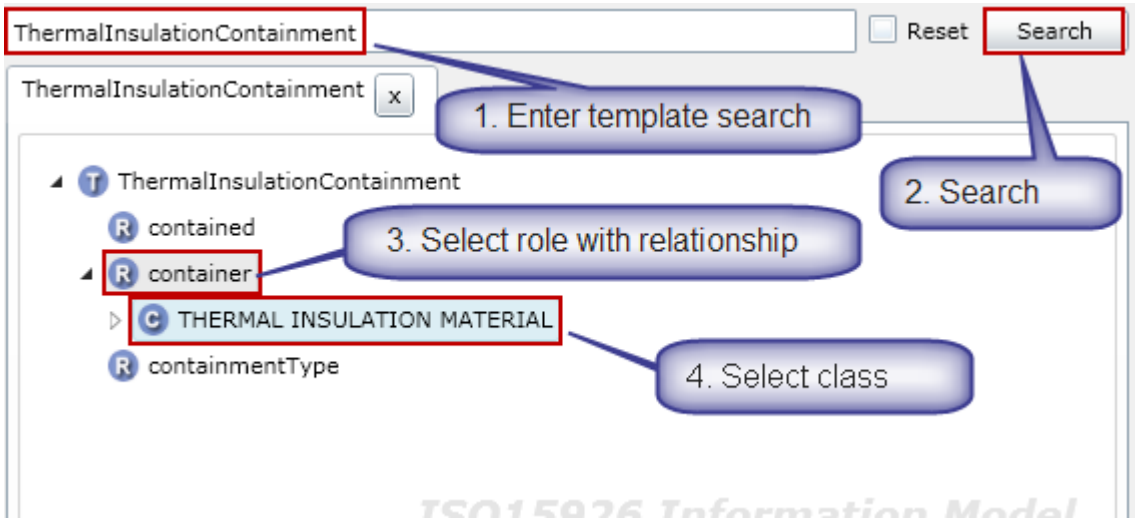


To map a relationship template, perform the following:

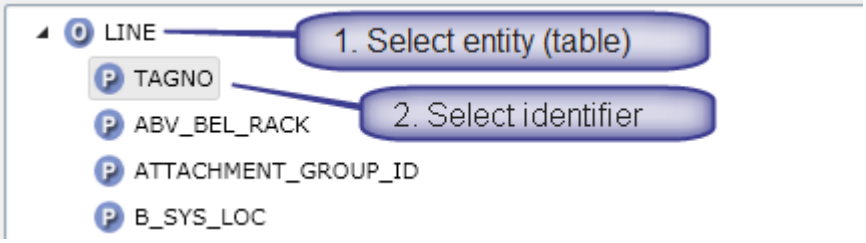
1. In the Mapping pane, select the template role that has a class relationship.



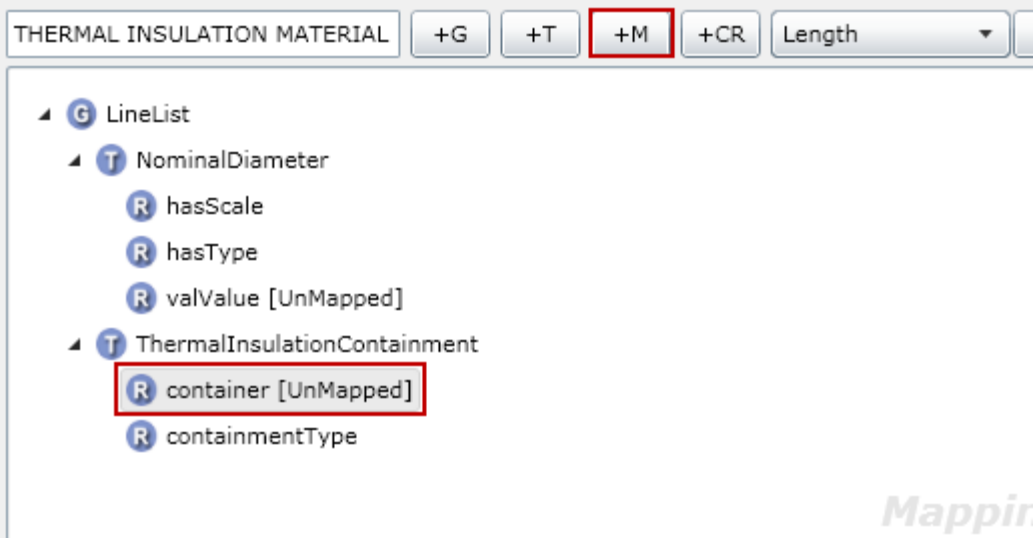
2. Get the class for the role relationship by searching for the template again in the Information Model pane, selecting the role with the relationship and finally selecting the role class.



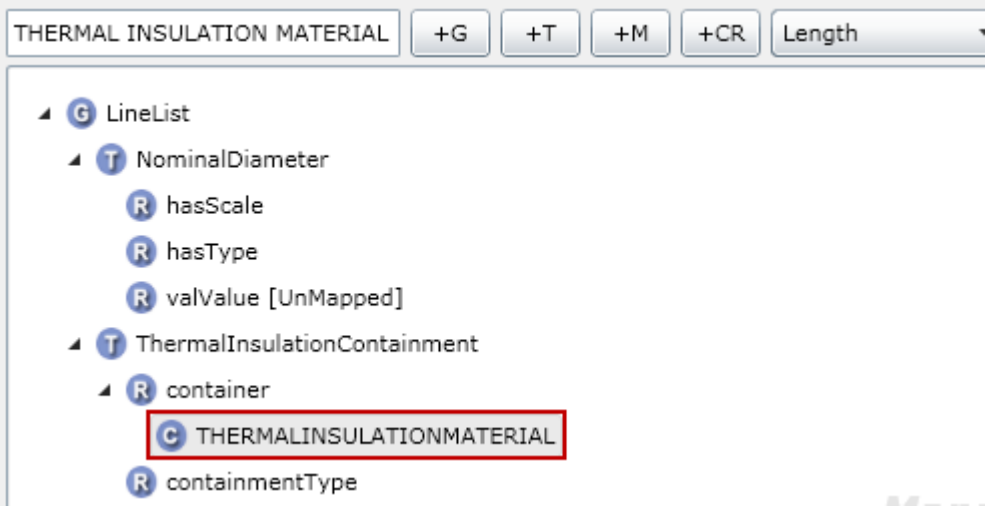
3. In the Data Objects pane, select the entity (table) identifier column.



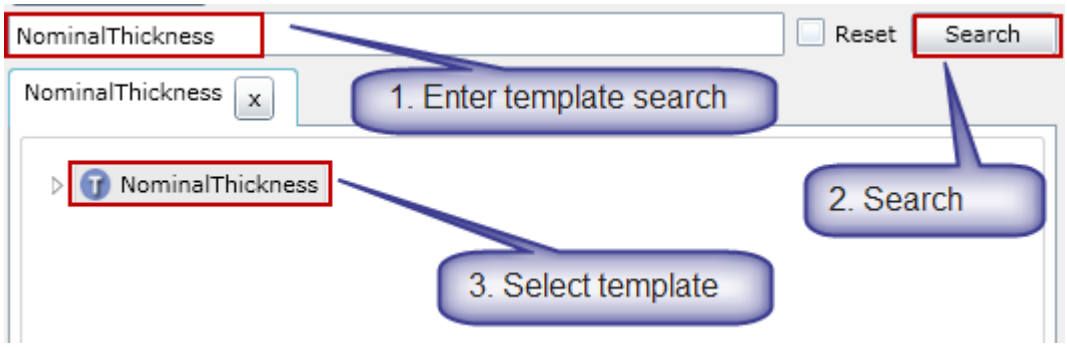
4. With the role, class and entity identifier select, in the Mapping pane map the relationship by clicking the +M button on the toolbar.



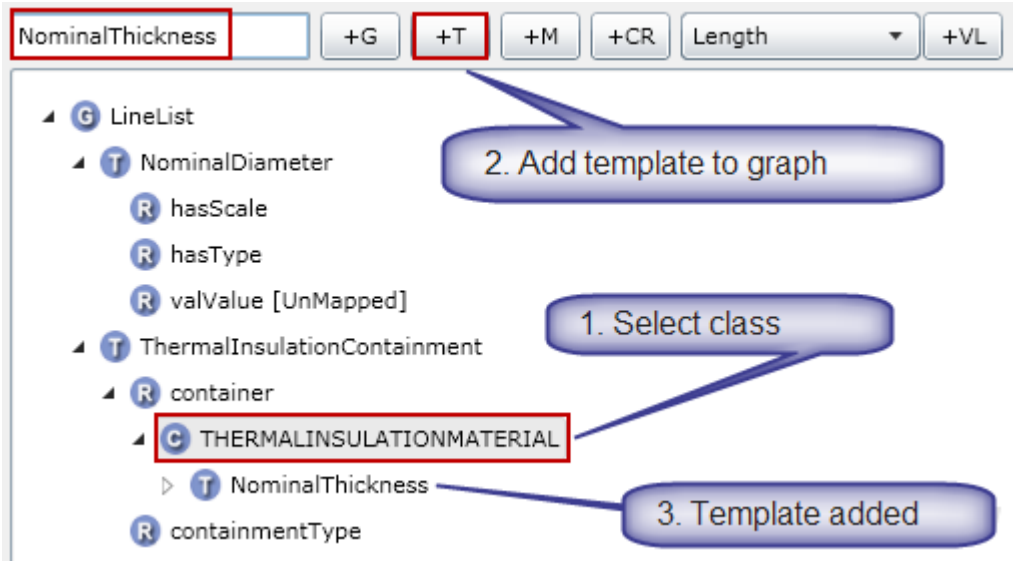
5. Select the role and then select the class that was mapped.



6. Additional templates can now be mapped to the related class.
7. Determine the property template that will be used, search for it in the Information Model pane, and then select the result



8. The selected template name appears in the Mapping pane. With the class selected, click the +T button on the toolbar to map the template to the class.



9. Continue mapping the added template as described earlier as well as adding other templates as necessary.

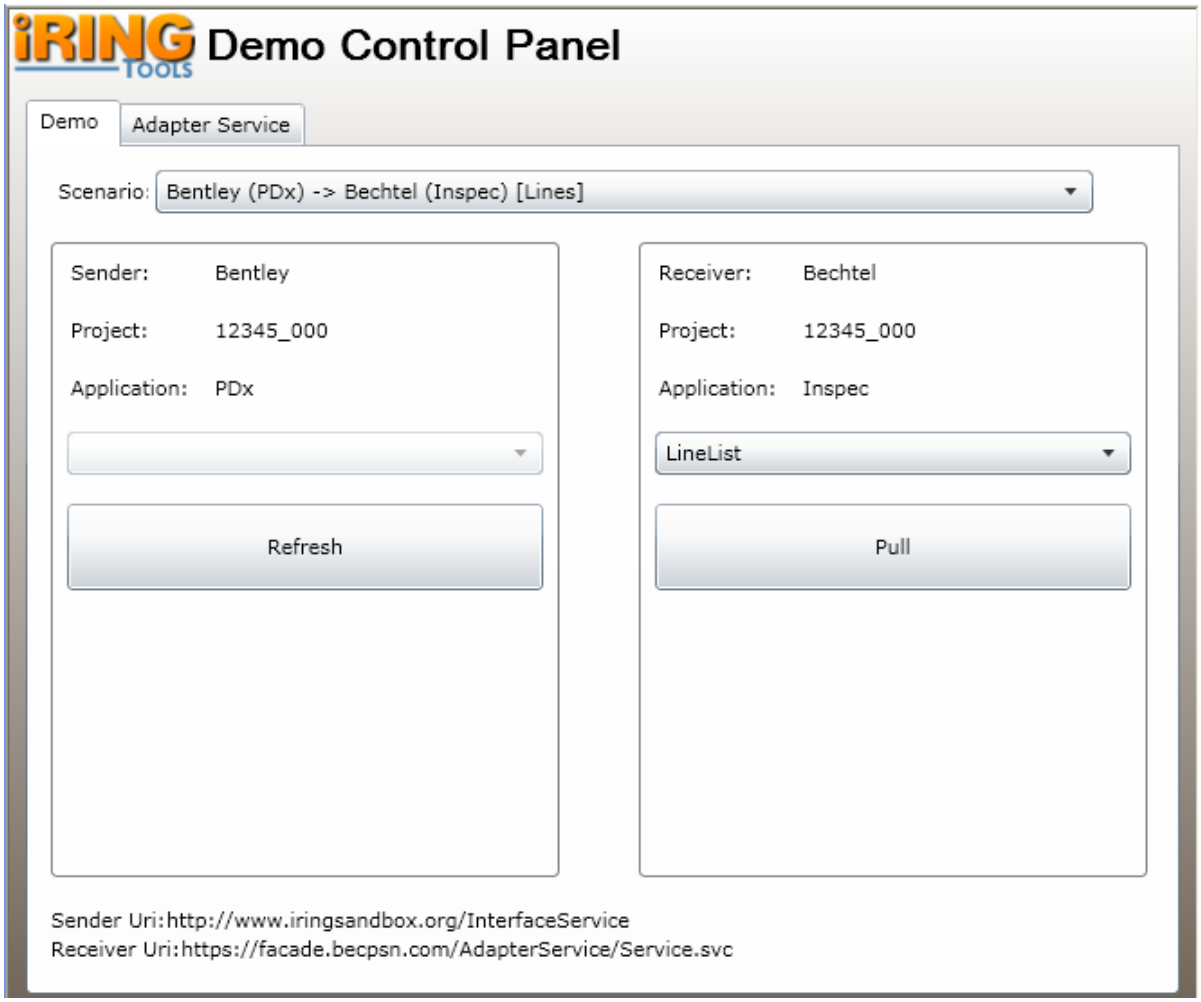
6 Performing Data Exchanges

The following sections will describe how to perform a data exchange using ***iRINGTools***.

6.1 Demo Control Panel

At present, data exchanges with ***iRINGTools*** use the Demo Control Panel. This is a Silverlight application using ***iRINGTools*** services to perform exchanges. For more information on how to setup and use the ***iRINGTools*** Demo Control Panel, contact the ***iRINGTools*** development team.

The following is a screenshot of a sample Demo Control Panel.



To use the Demo Control Panel, perform the following:

1. Select the scenario to execute.
2. If necessary, refresh the sender data
3. Perform the pull on the receiver.