



IMS Schema Profiling Tool

Version 2.0

SchemaProf

User Manual

| | |
|--|-----------|
| INTRODUCTION | 4 |
| 1. IMS SCHEMAPROF USER INTERFACE | 5 |
| 1.1. Menu bar..... | 5 |
| 1.2. Toolbar | 7 |
| 1.3. Selection Tabs | 7 |
| 1.4. Navigation Tree..... | 7 |
| 1.5. Working Area | 9 |
| 1.6. Preferences | 10 |
| 1.6.1. Global Settings | 10 |
| 1.6.2. Internationalization..... | 11 |
| 1.6.3. Plugins..... | 12 |
| 1.7. Command Line Usage | 12 |
| 1.8. Help and Bug Reporting..... | 12 |
| 2. SCHEMA PROFILING | 12 |
| 2.1. File Handling | 12 |
| 2.1.1. Creating a new Profile..... | 12 |
| 2.1.2. Opening an existing Profile..... | 14 |
| 2.1.3. Using the File History | 14 |
| 2.1.4. Saving a Profile..... | 15 |
| 2.1.5. Closing a Profile | 16 |
| 2.1.6. Annotations for Profiles | 16 |
| 2.2. Managers | 17 |
| 2.2.1. Type Manager | 17 |
| 2.2.2. Condition Manager | 18 |
| 2.2.3. Namespace Manager..... | 20 |
| 2.2.4. Profile Manager..... | 21 |
| 2.2.5. Test Configuration Manager..... | 22 |
| 2.3. Modifications..... | 22 |
| 2.3.1. New, Edit, Delete Modifications | 23 |
| 2.3.2. Order of Modifications, Reordering of Modifications | 23 |
| 2.3.3. Cardinality Type Modifications | 23 |
| 2.3.4. Value Type Modifications | 25 |
| 2.3.5. Global and Local Modifications | 32 |
| 2.3.6. Extending a specification | 33 |
| 2.3.7. Profiling Extensions..... | 37 |
| 2.3.8. Assertions | 37 |
| 2.3.9. Additional Constraints..... | 39 |
| 2.4. Annotations..... | 44 |
| 2.4.1. Global Annotations | 44 |
| 2.4.2. Modification Annotations | 45 |
| 2.5. Profile Extension Points | 45 |
| 3. TUTORIAL | 46 |
| 4. USING PROFILES FOR CONFORMANCE TESTING | 50 |
| 4.1. IMS Approved Conformance Test Systems..... | 50 |
| 4.2. Other Test Systems | 52 |
| 5. KNOWN ISSUES AND WORKAROUNDS | 52 |
| REFERENCES | 53 |
| ABOUT THIS DOCUMENT | 53 |
| APPENDIX A – SCHEMAPROF SOFTWARE LICENSE | 54 |

Date: 30. November 2009

Software and Documentation: <http://www.imsglobal.org/profile>
Bug Reports and Suggestions: info@imsglobal.org

Introduction

IMS SchemaProf is an *XML schema profiling tool* that is used to localize XML schemas, typically the XML binding of a specification or standard. This schema is modified to reflect the specific usage of the specification by a community. The result of this process is a community application profile.

A discussion of the principles and technical details of application profiling is beyond the scope of this user manual. A discussion of these subjects is provided in the *IMS Application Profile Guidelines* which are available at <http://www.imsglobal.org/ap/>. IMS SchemaProf has been designed to support the implementation of these guidelines.

An application profile contains only the differences made to the base schema and is intended to be used in combination with the base schema.

During the work with XML schema profiling it was noticed that most specifications define data structures and constraints in text documents. XML bindings of the specification in form of XML schemas cannot express all constraints of a specification. IMS SchemaProf allows capturing such modifications beyond the expressive power of XML schemas. IMS SchemaProf profiles contain – besides modified schemas – also Schematron rules and so-called additional constraints which capture requirements to the structure of packages of multiple files. IMS SchemaProf supports domain profiling, i.e. profiles that consist of a number of application profiles together with descriptions how these profiles are to be used in an interrelated way in conformant data packages.

IMS SchemaProf can be used to create application profiles for upload to the IMS Application Profile Registry. See <http://www.imsglobal.org/sdan/> for the IMS profile definition, registration and maintenance procedures.

While it is possible to use the derived schemas and Schematron rules with any XML schema or Schematron validator, the main intended use is for the production of profile specific test systems. Such test systems can go beyond schema and Schematron validation and test also the additional constraints mentioned above.

IMS SchemaProf originates from the European project TELCERT¹ and has been further developed by the Knowledge Media Institute of the University Koblenz-Landau (IWM Koblenz) with support from the IMS Global Learning Consortium. Version 1 of IMS SchemaProf was released in 2007. The current version 2.0 differs from version 1 mostly in allowing the user to set details of test system configurations.

Section 1 of this document provides a global overview of IMS SchemaProf.

If you wish to jumpstart profiling you may start reading Section 2 and then follow the hyperlinks embedded into the descriptions of the particular modifications you want to make as needed. The last profiling step should be crosschecking the [test system configuration](#) created by *IMS SchemaProf*. In case you intend to develop a domain profile combining one main profile with one or more auxiliary profiles, you should make the auxiliary profiles first in order to re-use them in the main profile.

Section 2 explains the operations that are available for profiling a specification. It is also suitable as a reference during your profiling activities.

Section 3 leads in a tutorial through a profiling exercise to give an impression of the features provided by SchemaProf. It is a good starting point for readers which are new to application profiling.

¹ EU funded research project, Contract No. 507128, see <http://www.opengroup.org/telcert>

The following Section 4 describes the possibilities to use SchemaProf profiles for Conformance Testing. The final Section describes known issues and gives recommendations how to deal with them.

1. IMS SchemaProf User Interface

The *IMS SchemaProf* user interface starts with a splash screen that shows the software logo and current version number while loading the program in background.

The main program window is divided into these main areas:

1. The **menu bar** gives access to the top level functionality.
2. The **toolbar** provides quick access to most often used functions.
3. The **selection tabs** switch between schema modifications and constraint, and group the schema components into top level categories.
4. The **navigation tree** shows the schema structure and components and allows the selection of components for modification and annotation.
5. The **working area** shows information on the currently selected element and allows the user to apply operations on that element.
6. The **status bar** shows the operation mode and compatibility level.

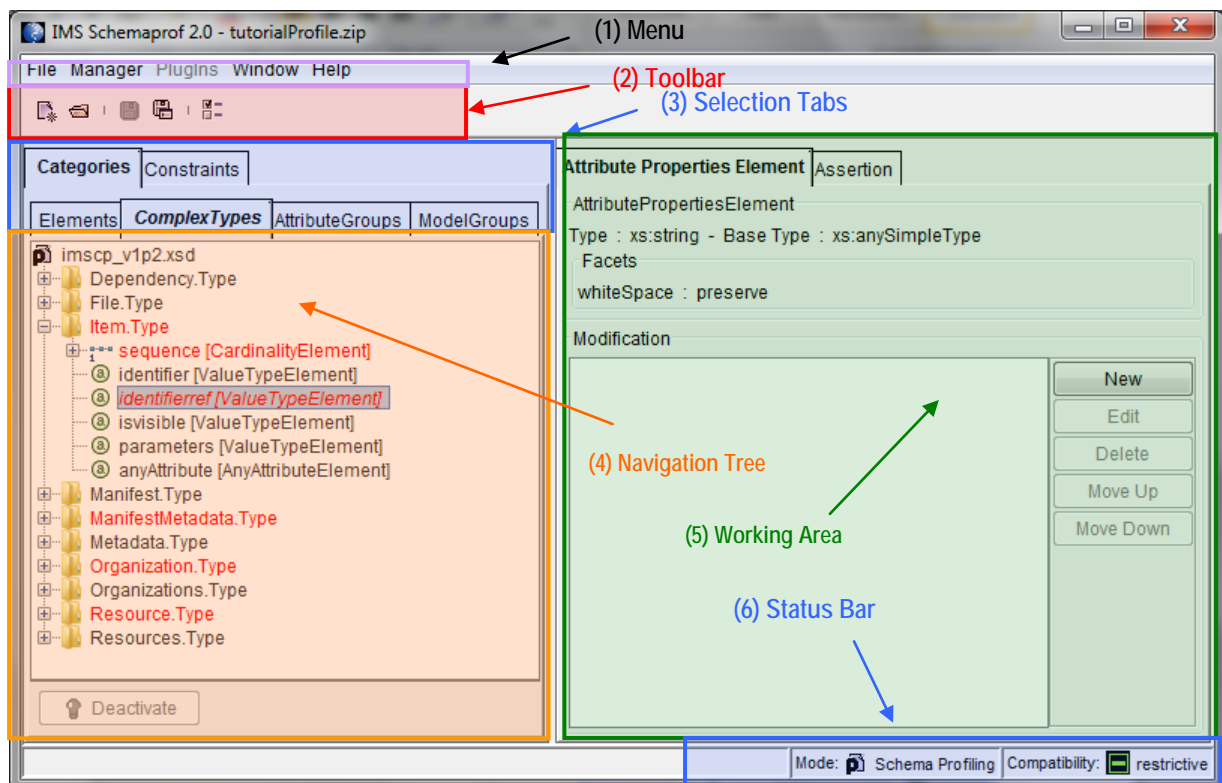


Figure 1 The IMS SchemaProf main window layout

For many operations pop-up dialog boxes will open to request the information needed from the user, some of them guide the user through several dialogs with a wizard interface.

1.1. Menu bar

The *IMS SchemaProf* menu provides access to the following menu items:

File Menu

| | |
|--------------------|---|
| New... | Starts a new profile or constraint file, based on an XML base schema to be specified by the user. Prior to that <i>IMS SchemaProf</i> will ask the user to save any unsaved changes to the current profile. |
| Open... | Open an existing profile or constraint file. |
| Save | Save the current file. If the file name is not specified the user must enter it now. |
| Save as... | Save the current profile or constraint file into another file or directory. |
| Close | Close the currently opened file. If latest changes have not been saved, <i>IMS SchemaProf</i> will ask to save or discard the changes. |
| 1 <filename> [...] | The file history that shows the last five files that have been opened. Select any of the files to open it directly. |
| Quit | Quit <i>IMS SchemaProf</i> . If latest changes have not been saved, <i>IMS SchemaProf</i> will ask to save or discard the changes. |

Manager

| | |
|----------------------------|---|
| Type Manager | The type manager gives an overview over the user defined type of the current profile and allows deletion or editing. A type is defined globally and may be used at several positions of a profile. |
| Condition Manager | The condition manager gives an overview over the user defined conditions of this profile and allows deletion or editing. A condition is defined globally and may be used at several positions of a profile. |
| Namespace Manager | The namespace manager is the place where you can set the target namespace for your profile. It also gives an overview of all namespaces used in the profile. |
| Profile Manager | The profile manager shows all the profiles referenced by the current profile. This is important for domain profiles. |
| Constraint Manager | The constraint manager gives an overview of the constraint file used and allows adding or removing constraint files. |
| Test System Bundle Manager | This item allows extending SchemaProf with support for new specific test systems. Extension of SchemaProf shall be handled in another document. |
| Test Configuration Manager | The Test Configuration Manager allows to determine which tests are to be applied to which files in a package. |

Plugins

IMS SchemaProf has a plug-in interface to allow specific extension to the program. If you intend to develop plugins, please contact schemaprof@uni-koblenz.de.

Windows

| | |
|-------------|--|
| Preferences | Opens the preferences dialog windows that will allow you to change |
|-------------|--|






several setting. See section [2.2 Preferences](#) for more details

Help

About Shows some information about the software, i.e. the software version, license information and contacts.

1.2. Toolbar

The toolbar provides quick access to the most frequently used operations:

| | |
|---|--|
|  | Start new profile or constraint file |
|  | Open existing file |
|  | Save file |
|  | Save file as |
|  | Preferences |

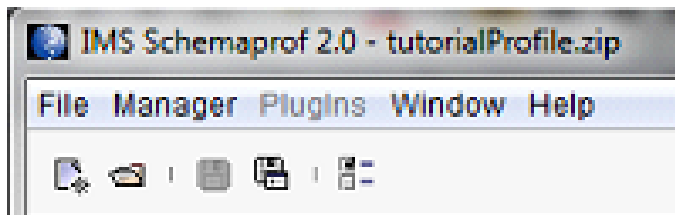


Figure 2: The *IMS SchemaProf* toolbar and menu

1.3. Selection Tabs

The selection tabs give access to the two main areas of work: schema modification and schema constraint editing.



If schema modification is selected, a second row of tabs is displayed grouping the schema components into *Elements*, *ComplexTypes*, *AttributeGroups* and *ModelGroups*. Each of these tabs shows the schema elements of this type and all of its subelements in the navigation tree. To see the schema structure in hierarchical view, which is possibly the most natural way to start, please select the schema root element under elements.

If the *Constraints* tab is selected, no second row of tabs is displayed. The navigation tree now shows a simple tree that holds all the constraints currently defined. See section [Additional Constraints](#) for more information on profile constraints.

1.4. Navigation Tree

IMS SchemaProf shows the structure of the base XML schema currently in use in a navigation tree on the left side of the program window below the rows of tabs.

The **root element** of the tree is labelled with the file name of the base XML schema. The type of file currently edited is shown by the icon of the root element:

-  indicates an additional constraint (AC) file
-  Indicates an application or domain profile

The root element expands to folders showing all the elements of the current element type selected by the second row of tabs. Because of references in the XML schemas, sub elements may appear at more than one location of the navigation tree. Especially when selecting the schema root element under 'Elements', you will find all of the elements used in this schema.

Unfolding the elements of the tree shows the next nesting level subelements. This way it is possible to descent into the schema until you reach a tree leaf, which is typically a simpleContent or attribute (other types are possible).

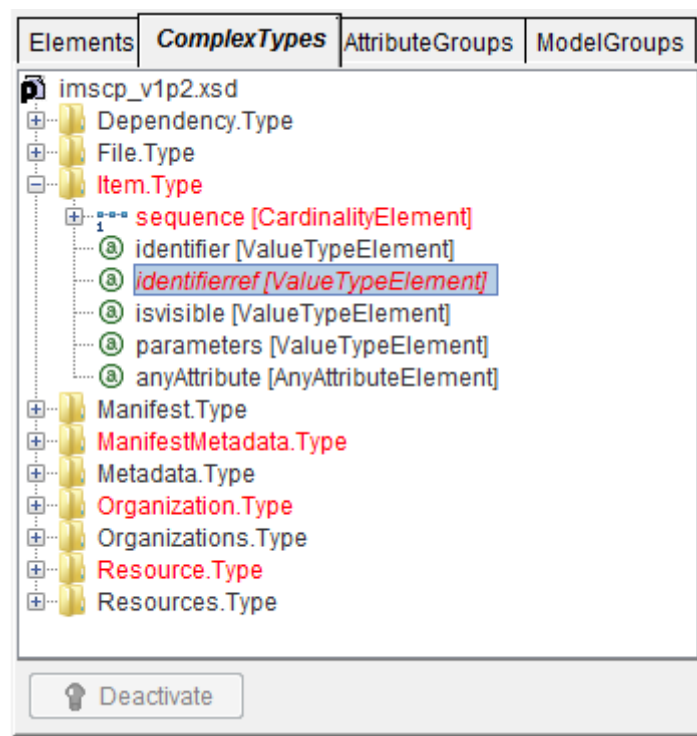








Figure 3: The Schema Navigation Tree

The types of elements in the navigation tree are signalled by different icons:

| | |
|---|--|
|  | A compound element that is made up of one or several sub elements, e.g. complex types. |
|  | Click on those symbols to expand or collapse this element in the tree with the details of the element. |
|  | Sequence element: Contains several subelements and the order of the subelements is relevant. |
|  | Choice element: Only one of the subelements may be chosen in the target document. |
|  | Element: Definition of a schema element. |
|  | Simple Content: A tree leaf that will hold a simple type value in any target document. |

| | |
|---|--|
| ⑧ | Attribute: Definition of an attribute of an element. |
|---|--|

Table 1 Navigation Tree Icons

Additionally the cardinality elements (those elements whose subelements may occur several, i.e. a sequence) show the cardinality of the original definition in the icon, e.g. $1..n$ shows the cardinality one or many elements. See [Table 3](#) for details on cardinality symbols.

1.5. Working Area

The right side of the program windows show the working area.

The content of the working area is adapted according to the element that is currently selected in the navigation tree on the left side. Depending on the type of selected element and whether a constraint file or profile is opened, different actions are available and can be accessed through the buttons on the right side. Others might be disabled or not available.

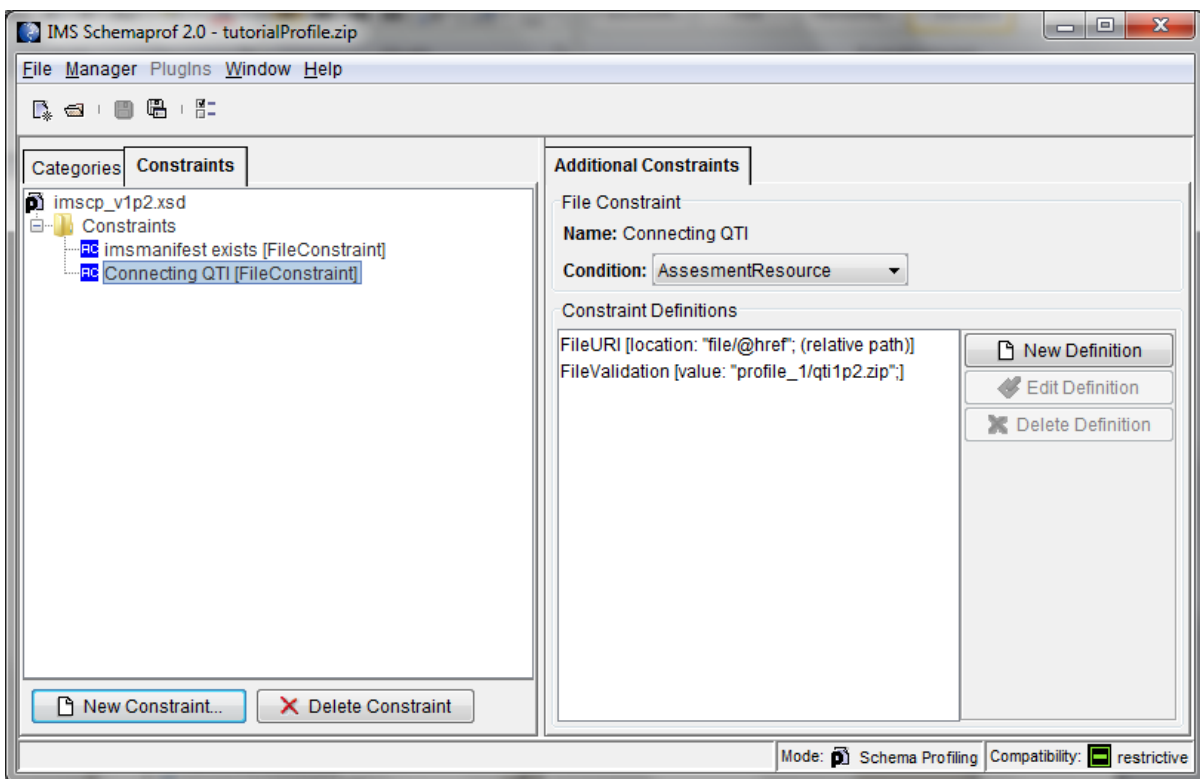



Figure 4 The IMS SchemaProf Working Area

On top one or more **tabs** are available. Normally they show the element type of the selected element, e.g. 'Cardinality Element'. Below, additional information about the element is displayed if available, e.g. element facets or constraint details. The lower part of the working area shows a list of the modifications made to this element when working with the profile, or the constraint definitions when working with constraint files. The buttons on the right are used to add, delete or edit modifications or constraint definitions.

The second tab gives access to the assertions that may be defined through XPath expressions. These assertions are tied to the current element. The assertions are defined in the profile and can only be validated for document instances of the current profile, which can be checked with the TELCERT test system. See section [Assertions](#) for more information.

1.6. Preferences

The *IMS SchemaProf* preferences dialog is available by clicking the toolbar button  or selecting the menu entry **Windows** ► **Preferences**. The preferences dialog window show three main categories that are available through the tabs on top (see Figure 5).

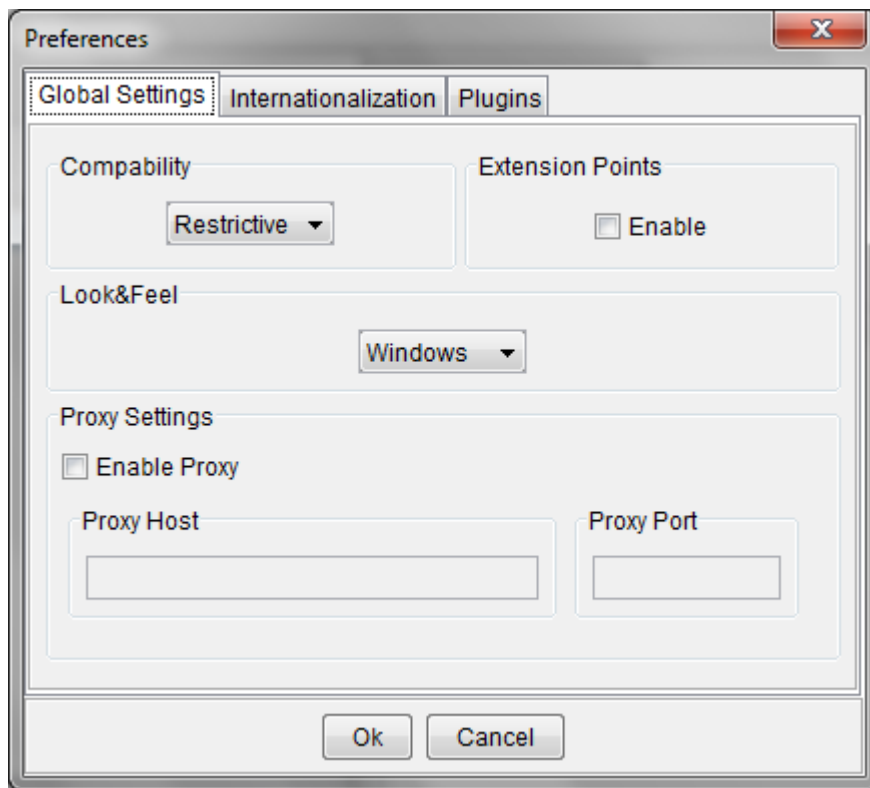


Figure 5 The preferences dialog window

1.6.1. Global Settings

Extension Points

IMS SchemaProf provides extensive support for extension point. In addition to letting you control the extension points predefined in the base schema through *any* elements and *anyAttributes*, it is possible to extend the profile definition format. Enabling this setting gives you access to this functionality. If disable this functionality is hidden for simplicity.

Look and Feel

The ‘*Look&Feel*’ setting allows switching between some visual appearances of *IMS SchemaProf*: ‘Windows’, ‘Metal’ and ‘CDE/Motif’ are available. This setting has no influence on the functionality of the program.

Compatibility level

IMS SchemaProf can switch between three different compatibility levels. Changing the compatibility levels is done in the [preferences window](#). The levels must always be set by the user.

Please note that IMS will only accept restrictive profiles in the profile registry!

The compatibility level chosen by the user influences the possibilities offered to the user to modify a base schema. So it is one of the most important tasks to set the correct compatibility level for your application.

When loading a profile that is not compatible to the current level, *IMS SchemaProf* will refuse to load this profile. *IMS SchemaProf* does not switch the compatibility level automatically when loading a profile.

Restrictive compatibility is the default compatibility level. The domains of predefined types may only be restricted but not extended. The set of possible document instances based on the resulting profile will therefore be a subset of the set of possible instances of the base schema. This ensures read compatibility of the application profile with the base schema. Programs capable of working with instances of the base schema will be able to work with all instances of the application profile.

Extensive compatibility allows only extensions of the domains of predefined types but no restrictions. The set of possible instances of the application profile will be a superset of the set of instances of the base schema.

This ensures write compatibility of the application profile to the base schema. Any instance generated (written) by programs capable of working with instances of the base schema will be compatible with the application profile.

Choosing **Not compatible** allows the user to enter any restrictions or extensions to the predefined types. No checking is done at all on the modifications entered.

A simple example: some profile element is assumed to have 2, 3, 4 or 5 instances in an application. It is therefore specified in the application profile with the cardinalities of minOccurs=2 and maxOccurs=5. The modification that are possible according to the different compatibility levels for minOccurs and maxOccurs are given in the following table.

| Compatibility Level | minOccurs | maxOccurs |
|-------------------------------------|--------------|--------------|
| <i>preset values in base schema</i> | 2 | 5 |
| restrictive | 2..5 | 2..5 |
| extensive | 0..2 | 5..unbounded |
| not compatible | 0..unbounded | 0..unbounded |

Table 2 Compatibility Level Settings

Please note that despite the compatibility checking, the minOccurs value must of course always be less than or equal to the maxOccurs value. Even with the setting ‘Not compatible’, the program will not accept e.g. the values minOccurs=4 and maxOccurs=3.

For more technical details on the concept of application profiling please look at the *IMS Application Profiling Guidelines* available at the IMS website:

<http://www.imsglobal.org/ap/index.html>

Proxy Settings

IMS SchemaProf requires access to all the XML schemas referenced by the current profile and by the XML schemas. Most of them will probably be available through web locations and accessed via HTTP protocol.

To support those network configurations that require the use of a HTTP proxy server, *IMS SchemaProf* will use the proxy server and port specified if the checkbox “Enable Proxy” is checked.

1.6.2. Internationalization

IMS SchemaProf is currently only available in English. Please contact IMSschemaprof@imglobal.org if you are interested to provide support for other languages.

1.6.3. Plugins

IMS SchemaProf functionality can be extended using plugins. The plugin support feature is provided for future extensibility.

1.7. Command Line Usage

IMS SchemaProf can be started from the command line as

```
/path/to/java -jar IMSSchemaProf-2.0.jar
```

1.8. Help and Bug Reporting

The user manual is provided online in PDF format. Please check the IMS website for the latest version of the documentation:

<http://www.imsglobal.org/profile>

There is also a screencast tutorial available on this site.

If you find **bugs** in the program or documentation, please commit them to our Bugzilla system with a detailed description and include the release number of SchemaProf. If available please provide your SchemaProf log files from the directory of your local *SchemaProf* installation and any other files used (XML Schema, Profiles, etc.). You can attach files to the bug report after committing the initial description. Bugzilla is available at:

<http://iwm.uni-koblenz.de/bugzilla/>

Within Bugzilla, select the product *Telcert SchemaProf* with the component *IMS SchemaProf*.

If you need further assistance please contact us directly via email to IMSSchemaprof@imglobal.org Any **feedback and suggestions** concerning SchemaProf are welcome under this email address as well.

The **version number and license** of the program can be found in the about screen that is shown during program start and is available using the menu entry `Help ► About`.

2. Schema Profiling

This chapter gives information on how to create and edit application profiles. Detailed information on the user interface is given in the [previous chapter](#).

2.1. File Handling

2.1.1. Creating a new Profile

To start a new profile, use the menu item `File ► New...` or the corresponding toolbar button. This will open a wizard dialog that allows the user in the first step to choose between creating a new profile or a constraint file. Choose *profile* here.

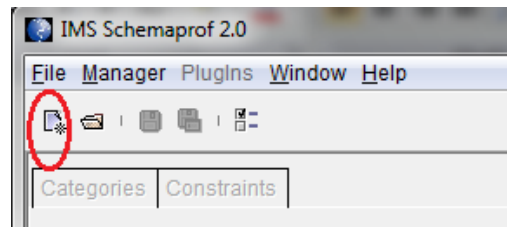


Figure 6 Create new profile using the IMS SchemaProf toolbar

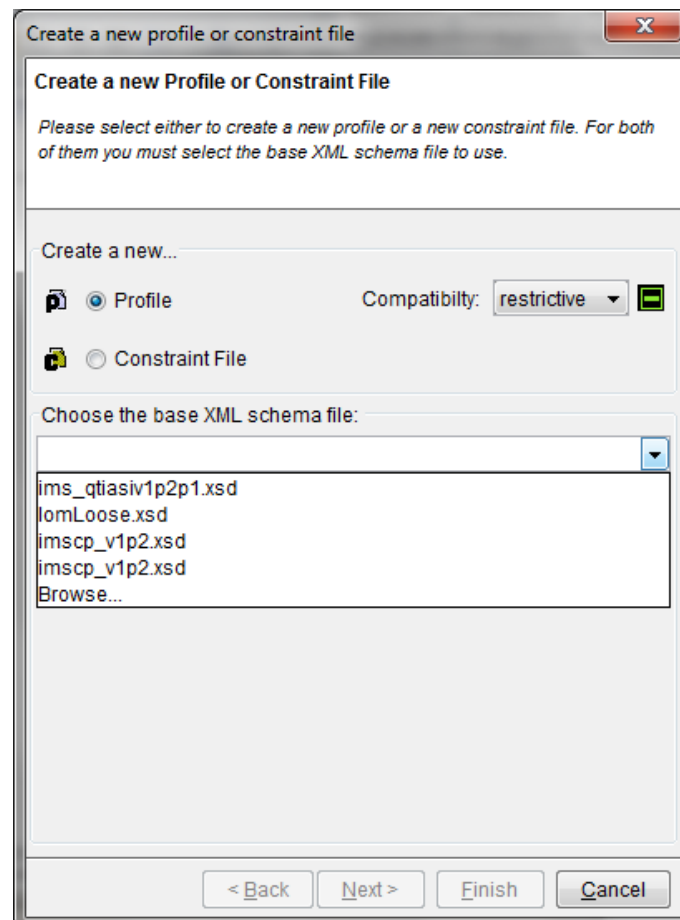


Figure 7 New file wizard

Then select the compatibility level you need. If unsure *restrictive* is recommended. For details on the compatibility level please refer to the section [Compatibility level](#).

Now choose the base XML schema file you want to profile. This is typically the XSD file of a specification you are using. You may choose a local file by choosing *browse...* from the drop-down list or enter an URI directly. The drop-down list also shows the last used schema files. Then click **Finish** to start profiling.

After closing the new file wizard with the finish button *IMS SchemaProf* will parse the selected base XSD file. This can take a few seconds and includes the reading of any files included or imported by the base schema. Warnings that appear during loading the base schema indicate problems with the schema which don't affect the profiling process. If, however, SchemaProf signals Errors, it is recommended to validate the schemas with another tool and to correct any errors.

Alternatively you may load an already existing profile.

When the analysis of the schema file is done, the schema structure is displayed in the navigation tree on the left side of the *IMS SchemaProf* window.

Now you are ready to begin with the actual profiling work!

It is good practice to define as a first step the *namespace* of the profile through the [namespace manager](#). By default the profile shall inherit the namespace of the base schema. Once being in the namespace manager, it is recommended as well to declare prefixes for all namespaces which you intend to use. In particular each namespace to be used in a [condition](#) or [assertion](#) must have a nonempty prefix assigned.

The following modification can be made to the schema:

- [Make global and local annotations to the profile](#)
- [Modifications to existing types](#)
- [Creation of new types](#)
- [Usage of conditions](#)
- [Use further XML namespaces or Profiles](#)
- [Add or modify additional constraints](#)

2.1.2. Opening an existing Profile

To open an existing application profile, use the menu item `File ► Open...` or the corresponding toolbar button. Then choose the file containing your IMS application profiling compliant schema profile from the local file system. This is a ZIP file containing all profile information. This file can be opened directly from the *IMS SchemaProf* file open dialog. No unzipping or uncompressing is needed.

Loading and parsing of the profile will take a few seconds. The profile structure is then shown on the left side of the program windows in the navigation tree.

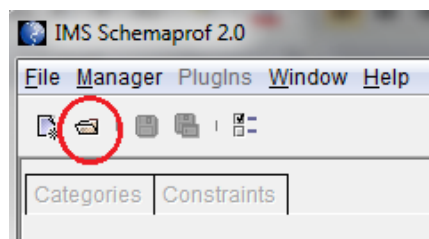


Figure 8: Opening an existing profile using the toolbar

Any XSD schema file (base schema, namespaces) or profile XML file referenced in the profile will be searched for locally in the same directory if no other location is specified. This gives you the possibility to: (a) work with locally modified copies of the XSD files specified, and (b) work offline with a local copy of that file in the directory. The XSD files not found in this directory will be loaded from the URL location. This requires a working Internet connection.

When loading a profile, *IMS SchemaProf* does an integrity check. If the profile has been changed outside *SchemaProf* this shall be reported. This does not imply any restrictions to the further editing of the profile unless it has been corrupted.

2.1.3. Using the File History

To open the most recent files, *IMS SchemaProf* offers a file history in the file menu that will show the file names of the last five files that have been opened. This provides quick access to

those profiles and avoids having to choose the correct directory first. Hovering the mouse pointer over a file name will reveal the complete path to the file.

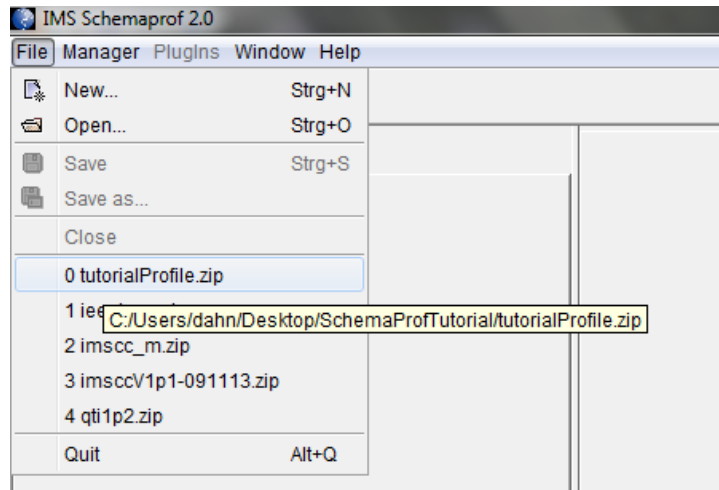


Figure 9 Opening a Profile from the File History

2.1.4. Saving a Profile

To save an existing application profile, use the menu item **File ► Save** to save under the already given filename, or **File ► Save as...** to save under a different filename or directory. Alternatively you can use the corresponding toolbar buttons.

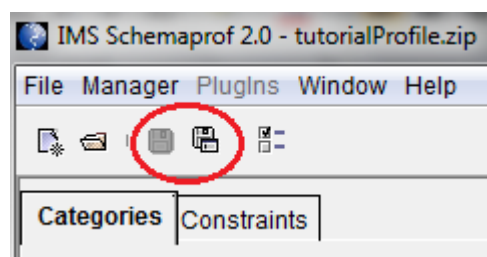


Figure 10: Save Profile / Save Profile as Toolbar Buttons

Your application profile is saved as a ZIP file. The ZIP format includes all local files and directories of the profile in a single archive file which can be opened directly with *IMS SchemaProf* for editing or be extracted with any ZIP tool. Remote files like XML schemas referred to by a URI are not included in the ZIP file. When opening a profile that uses remote files a network connection is expected.

The ZIP file is easy to handle, e.g. to send via email. For inclusion into the profile registry IMS will only accept profiles provided as ZIP files.

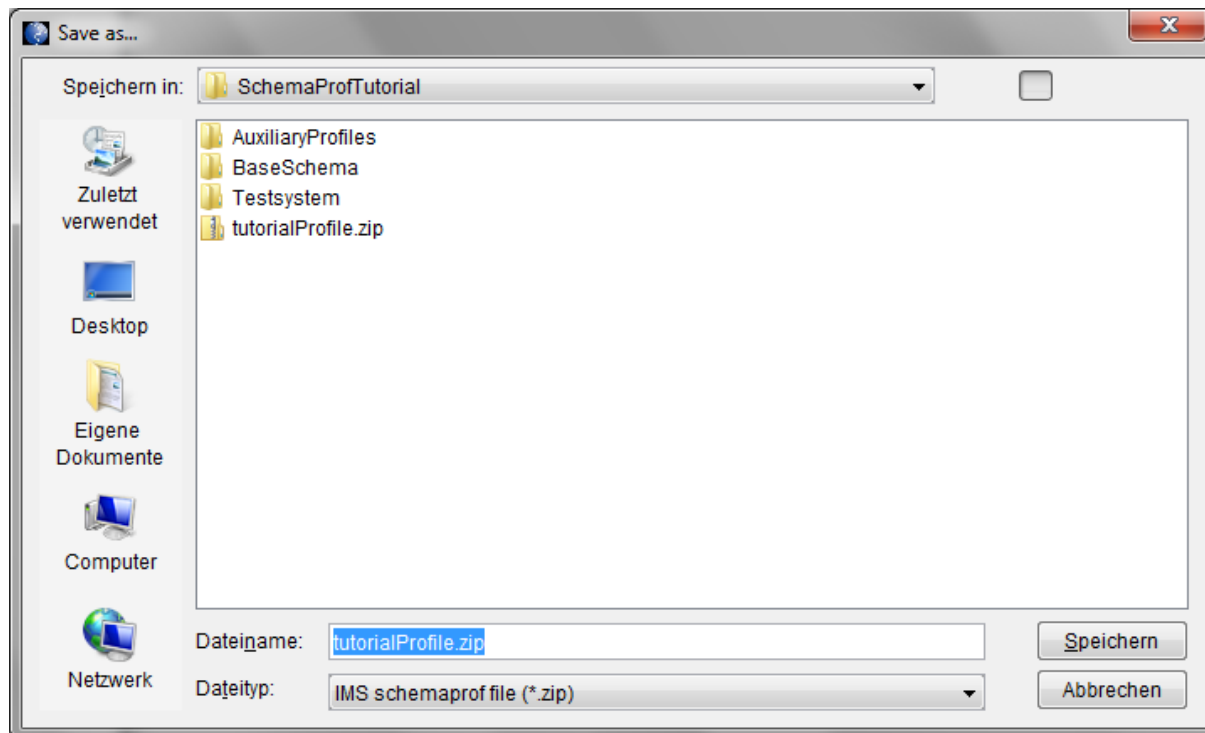


Figure 11 : Saving a profile under a different name

Please note that a copy of the base schema XSD file and of each referenced XSD file is saved together with the application profile file (in the ZIP archive).

2.1.5. Closing a Profile

A profile is closed for editing if the user chooses `close` from the file menu, or if a new profile is created or opened, or if the user quits the program. Prior to closing the profile, *IMS SchemaProf* will check for any changes made since the last saving and ask the user to save the profile or confirm to discard the last changes made.

2.1.6. Annotations for Profiles

IMS SchemaProf supports both global and local annotations.

Global annotations that refer to the application profile as a whole can be entered when the root element on the left side is selected. It is recommended to make annotations and provide descriptions and explanations to the users of the profile whenever possible. For more details please read section [2.4.1 Global Annotations](#).

Local annotations are attached to modifications (see section [Modifications](#)) and provide a possibility to explain and document the intention of this specific modification. This can become very useful when editing a profile (possibly created by someone else) and the idea behind a modification it is not obvious. The following sections on modification show how to make local annotations.

We recommend using annotations extensively since this provides a way of inline documentation that can be extracted later on. In fact, annotations shall be included into the derived schemas and Schematron rules generated by *IMS SchemaProf*. In particular annotations of conditional modifications are important since the Test System shall use them as explanations for the user in case the restrictions set by these modifications are violated in a document under test.

2.2. Managers

Managers from the `Manager` menu allow for the provision of reusable resources beyond the base XML schemas that are needed for a profile. You may return to the appropriate manager whenever you need to add another resource.

2.2.1. Type Manager

The type manager is the user interface to manage user defined simple types for your currently opened application profile. You can enter a new type and edit or delete existing types. The type manager is started by the menu entry `Manager ► Type Manager`.

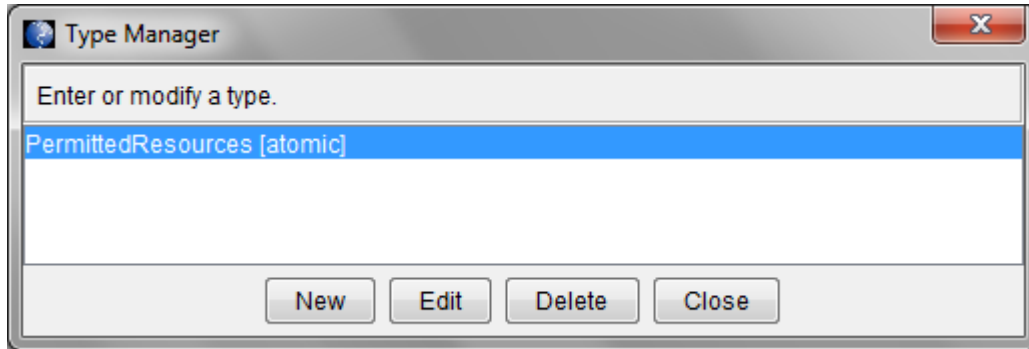


Figure 12 The Type Manager

New Type

To add a new type to your application profile, press the `New` button. This opens a dialog window offering three basic possibilities to derive a new type: make a restriction to an existing type, create a union of several existing types, or create a list of existing types. They are explained in detail in the following.

Create new type from a restriction of an existing type

Step 1: Enter a name for the new type you are creating now.

Step 2: Select *Restriction*.

Step 3: To define a new type based on a restriction of an existing type, you must now choose the base type from the drop-down box below. Then press the `Next` button.

Step 4: Enter the restrictions to be made to the basic type. Leave all other fields untouched to not overwrite the current values. The restrictions available depend on the base type you have chosen in the previous step. The following restrictions are possible:

- The tab *length* allows restricting the number of characters for a value. A fixed length, a minimum and a maximum length can be entered. More details on <http://www.w3.org/TR/xmlschema-2/#rf-length>.
- The tab *pattern* allows entering of a pattern, constraining the lexical value space of a data type to literals which match the given pattern. The pattern is given as a regular expression according to <http://www.w3.org/TR/xmlschema-2/#rf-pattern>.
- The tab *enumeration* makes it possible to enter a set of values. The values of this element are then restricted to this set of values. See <http://www.w3.org/TR/xmlschema-2/#rf-enumeration> for more information.
- The tab *whitespace* can be used to specify the handling of whitespaces in values of this element. Allowed values are *preserve*, *replace* and *collapse*. The meaning of the values is given in <http://www.w3.org/TR/xmlschema-2/#rf-whiteSpace>.

Press the button to continue, when all restrictions are entered.

Step 5: Press the button to finish this type.

Create a new type from a union of existing types

Step 1: Enter a name for the new type you are creating now and press to continue.

Step 2: Select *Union* and press to continue.

Step 3: To add a type to the list of types to unify, select the type from the drop-down list and press the button. The type will now appear on the list below. To remove a type from the list, select the type from the list and press the button. Press when all required types are on the list.

Step 3: Press the button to finish this type.

Create a new type from a list of existing types

Step 1: Enter a name for the new type you are creating now and press to continue.

Step 2: Select *List* and press to continue.

Step 3: Select the type from the drop-down list, the new list element should be based on. Press to continue.

Step 4: Press the button to finish this type.

Edit Type

To edit a previously user defined type select the type from the list of available types. Then press the button. Depending on the nature of the type (atomic, union or list) a different dialog window will open.

The information you can enter in the fields of this dialog are a combination of the fields offered in step 2 and step 3 of the [New Type](#) operation above. Please refer to the description above for further explanations.

Delete Type

To delete a previously user defined type select the type from the list of available types. Then press the button. This will delete the currently selected type.

To close the Type Manager use the button or the symbol.

2.2.2. Condition Manager

The Condition Manager is the user interface to manage the conditions defined by the user. The window shows a list of all defined conditions. A condition has an ID, a name and the condition declaration.

The *ID* is set by the system and must be unique. It is used to identify the condition internally; the user may not change this ID himself.

The *name* is given by the user and should be a meaningful name to identify the condition. The name is presented to the user whenever he has to select a condition.

The *condition declaration* can be any valid XPath expression that is evaluated to the Boolean values. See below for further explanations on declaration of conditions.

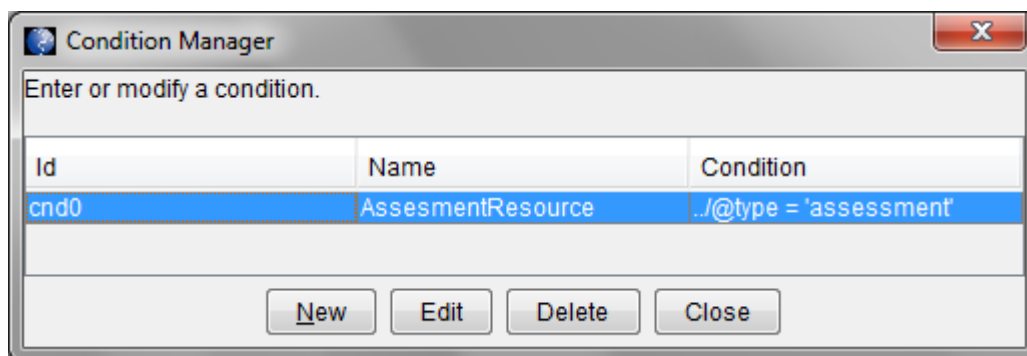


Figure 13 The Condition Manager

Declaring XPath Conditions

A condition for *IMS SchemaProf* is generally any valid XPath expression that is evaluated to the Boolean values of either true or false. See <http://www.w3.org/TR/xpath#booleans> for further information on Boolean objects and their evaluation according to the W3C recommendations. Expressions evaluating to any other value, e.g. to 1 or 0, are not allowed!

You may e.g. use functions, refer to schema elements and their values, or use Boolean and comparison operators. For an authoritative and up-to-date explanation on XPath expressions always refer to the W3C recommendations given above. For a simple and quick overview on the XPath expression elements you can also use some quick reference available on the internet, e.g. <http://gordon.inf.ethz.ch/lectures/isk2003/XPath.pdf>.

Make sure you prefix all element names with prefixes defined in the *Namespace Manager*.

Whitespaces can be used for better readability and are preserved. They should not influence the evaluation of the condition, but this depends on the program working with your profile. According to W3C recommendations, all of the following expressions should be evaluated to the same value:

```
( @myAttribute <= ( count( ../tut:someStructure ) ) )
(@myAttribute <= ( count(../tut:someStructure)) )
( @myAttribute<=(count(../tut:someStructure)) )
(@myAttribute<=(count(../tut:someStructure)))
```

Examples of Conditions

- `true()`
This is a valid condition, making use of the Boolean function `true()` that will always evaluate to the Boolean value *true*. The condition will therefore always be satisfied.
- `true`
This is also a valid condition, but does not refer to the Boolean value *true*, but to an element with name “true” in the current element structure.
- `@myAttribute = "myValue"`
This condition will evaluate to true, if the value of the attribute “myAttribute” of the current element is set to “myValue”. Otherwise it will evaluate to false.
- `((name(..) = "item") or (name(../..) = "item"))`
This condition evaluates to true if the name (function “name()”) parent element (“..”) or the grandparent element (“../..”) of the current element is “item”. Otherwise it evaluates to false.
Assuming that the current element is also called “item”, this condition can be used to

detect if some elements are iterated in the profile structure, that is the element “item” contains other elements called “item”.

- `count(..someListElement) <= "5"`
This condition evaluates to true, if the list element “someListElement” located one level above the current element in the element hierarchy of the profile, consists of no more than 5 list element. Otherwise it evaluates to false.

2.2.3. Namespace Manager

The namespace manager gives you an overview of the namespaces used in your profile and the namespace prefixes they are mapped to. Here you may add other namespaces and define prefixes for them.

Additionally it let’s you control the target namespace of your profile. Profiling a specification does make changes to the original and therefore the original namespace should not be used, although the compatibility to the original specification is preserved especially in the restrictive mode of IMS SchemaProf. By default the IMS SchemaProf will keep the original namespace as target namespace for the profile.

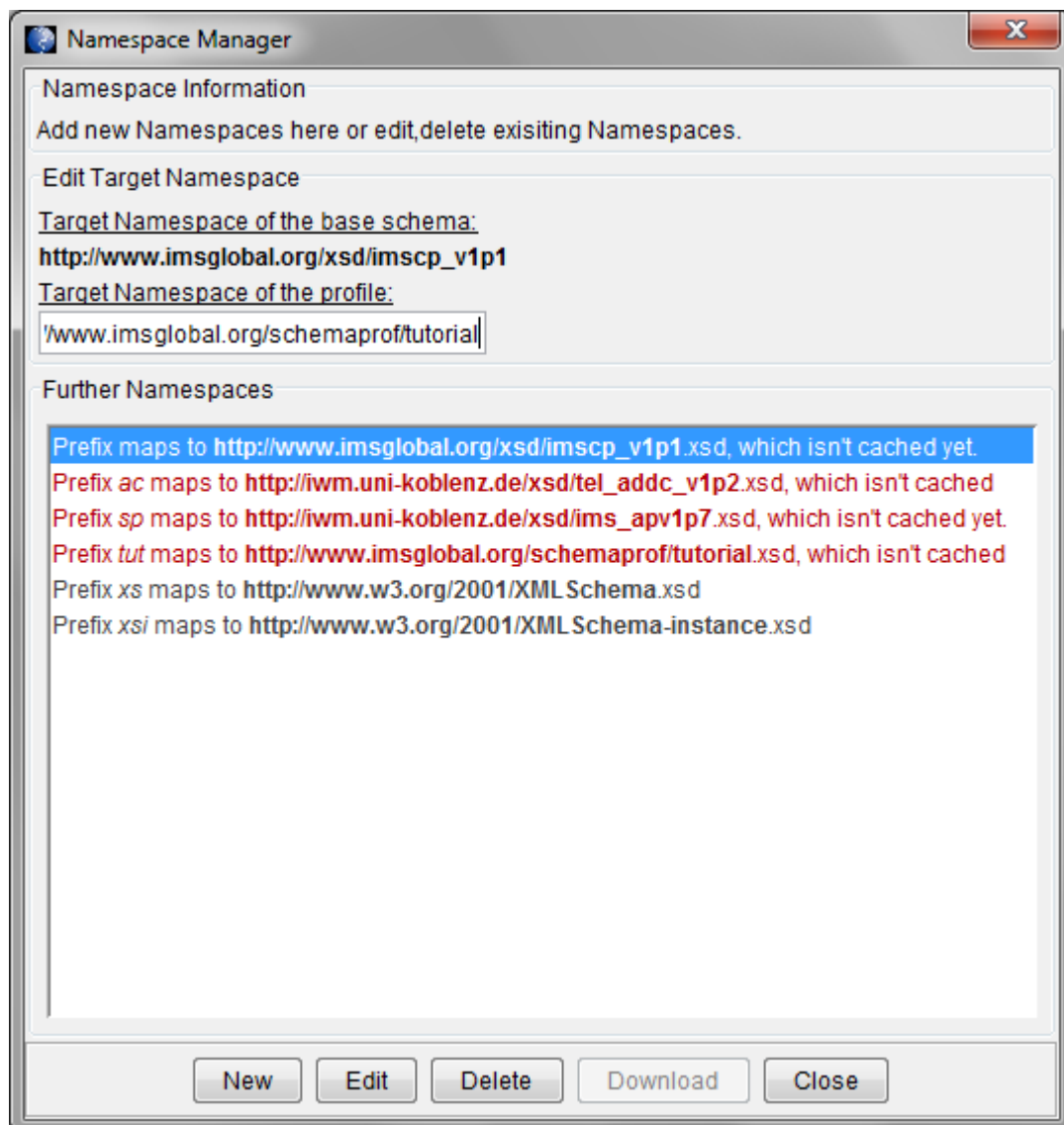


Figure 14 The Namespace Manager

Profile Manager

The profile manager provides the user a complete list of all the profiles that are used at the extension points and additional constraints of the current profile. If the same profile is referenced at more than one extension point of the current profile, it will still appear only once in the profile manager.

Please note that the profile manager will only show those profiles that are referenced directly from the current profile. If a referenced profile again references further profiles (therefore indirectly referenced by the current profile), they are not shown in the list.

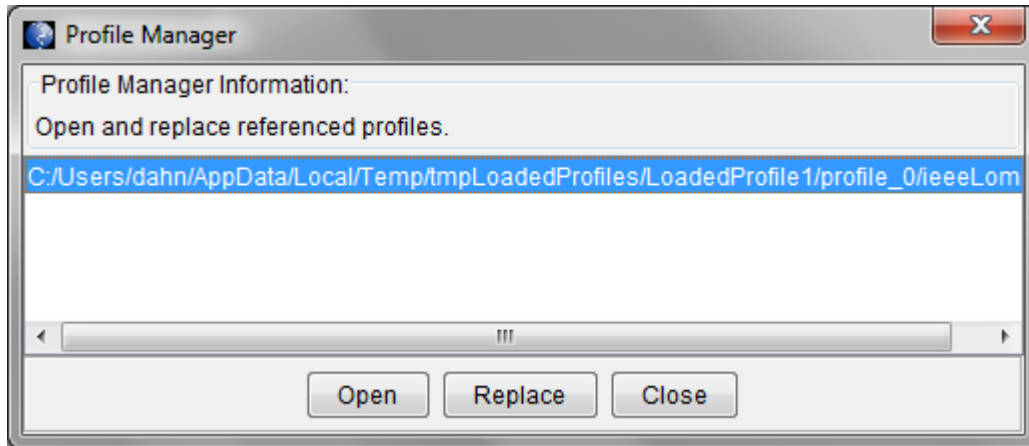


Figure 15 The Profile Manager

Despite giving an overview over the referenced profiles, the profile manager profiles the following functionality:

Open Profile

A profile referenced by the current profile can be opened for editing using the profile manager. Select the profile to open from the profile manager list and press the **Open** button. This supports the concept of domain profiling where several specifications are used in combination, and therefore should be profiled together.

Prior to closing the current profile, IMS SchemaProf will ask the user to save the current profile if any changes have been made that are unsaved.

To come back to the previous profile, please use the functionality of the file history.

Replace Profile

It is possible to modify all references of a profile to reference another profile. This functionality is provided to allow easy replacement of a profile by a new version of the profile. This will hopefully improve consistency of a profile when a profile is referenced more than once. Even if the profile is referenced only once, access to it is much easier through the profile manager instead of having to locate the extension point in the depth of the navigation tree first.

To replace all occurrences of a profile, select the profile from the profile manager list and press the **Replace** button. This will open a file selection dialog, where you choose the new profile XML file to be used. After pressing 'Open' all occurrences are replaced.

2.2.4. Test Configuration Manager

The Test Configuration Manager is accessible from the menu item “Edit Test Configuration”. It is used to create a configuration file for the test system to be generated for the profile.

This configuration specifies for which type of file (given by file name or namespace) schema validation and schematron validation should be done. SchemaProf will guess reasonable settings but it is strongly recommended to inspect these. In particular, if the same namespace is defined in several schemas you shall need to select manually which schema is to be taken as the main schema defining this namespace. SchemaProf will offer you a dropdown list which lists all that are available. If you select schemas from auxiliary profiles you should select those with *_localised* in their names since these are the ones modified for the specific auxiliary profile.

Some specifications, for example IMS Content Packaging, request that all files in a package are referenced at particular locations in a particular file. Using the Test Configuration Manager you can determine this location with an XPath expression or deactivate reference testing.

2.3. Modifications

To make, edit, delete or rearrange modification to your specification’s XML schema, you must first locate and select the element to be modified in the navigation tree on the left side of the program window. The right side of the windows then shows the list of all modification of the currently marked element, if any exist already.

On top of the right side you see the default settings of the element as given in the base schema. On the rightmost side of the program window you find buttons to create new modifications, edit or delete existing modifications, and to rearrange the modifications.

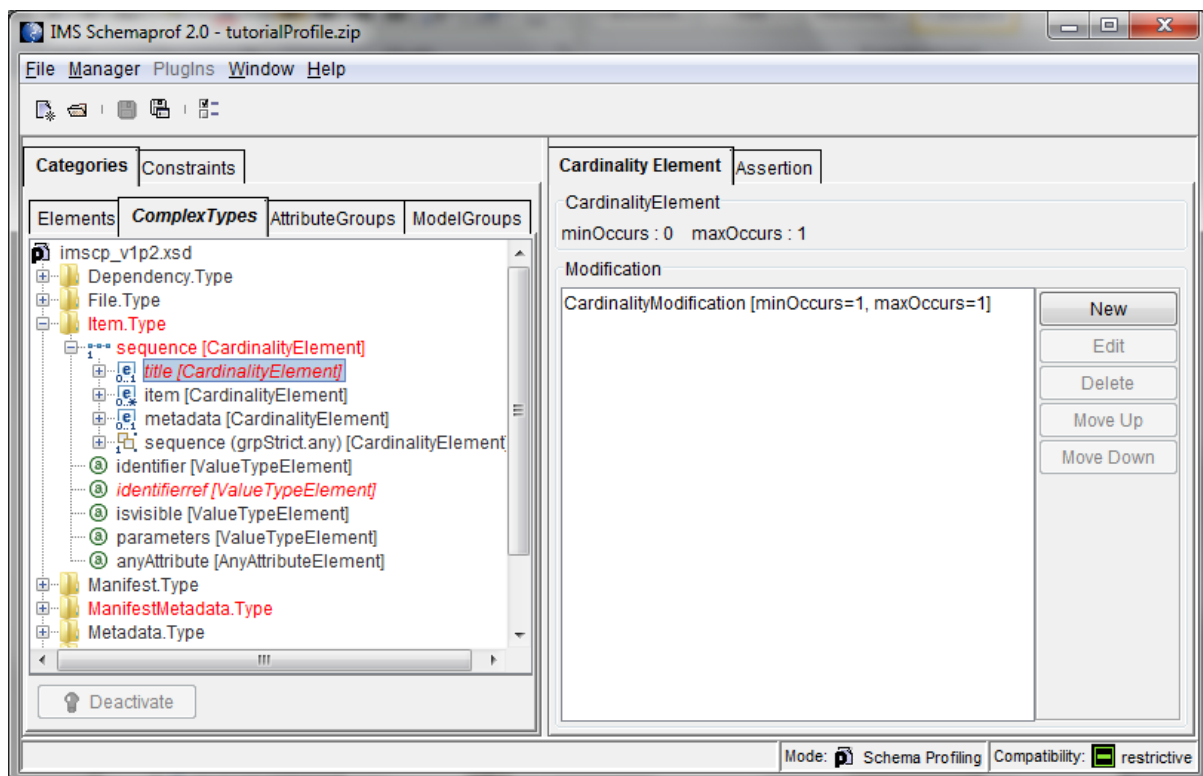



Figure 16 The Modifications Dialog

If modifications are applied to an element this element is highlighted in red colour in the navigation tree. **The order of the modifications is important!** If more than one modification is made, only the first valid modification is applied. See section [Order of Modifications](#), [Reordering Modifications](#) for more information.

2.3.1. New, Edit, Delete Modifications

To enter a **new modification** select some element on the navigation tree on the left side and click the **New** button. This will open a wizard dialog specific to the current element type. For details on the possible values and meaning for specifying the modifications please refer to the following sections [Cardinality Type Modifications](#) and [Value Type Modifications](#).

Elements denoted with a folder symbol () in the navigation tree can not be modified. Still [Assertions](#) and, if extension points are enable in the preferences, [Modification Extensions](#) are possible for those elements.

To **edit a modification** select the element in the navigation tree on the left side first. Modified elements are highlighted in the navigation tree. Then select the modification you want to edit in the working area on the right side. Now press the **Edit** button to edit the selected modification. For details on the possible values and meaning for specifying the modifications please refer to the following sections [Cardinality Type Modifications](#) and [Value Type Modifications](#).

To **delete** a modification select the modified element in the navigation tree on the left side first, then select the modification you want to modify on the right side. Now press the **Delete** button with the modification selected and confirm the deletion.

2.3.2. Order of Modifications, Reordering of Modifications

For a single element, more than one modification can be specified. But only one modification will be applied for each element. Which modification will be applied is determined by the order in which the modifications are given and by the conditions given for the modifications. The first modification whose condition is evaluated to *true* will be applied. All modifications following this modification in the list of modification are ignored.

This means that the order in which the modifications are arranged in the list is important. Modifications that are not bound to a condition (unconditional) are treated as having a true condition. Any modification appearing after an unconditional modification will never be applied. To change the order of the modifications select a modification from the list and use the **Move Up** and **Move Down** buttons to move it up or down in the list.

2.3.3. Cardinality Type Modifications

Cardinality Modifications are possible for the cardinality elements that are denoted different icon symbols and the description [*CardinalityElement*] in the navigation tree on the left side.

The icon symbols also indicate the cardinality of the **unmodified** element:

| | |
|------|---|
| 1 | Mandatory single element of this type |
| 1..n | Mandatory element that must occur at least once, but is limited to a maximum number of occurrences. |
| 1..* | Mandatory element that must occur once and may occur as many times as wanted (<i>unbounded</i>). |
| 0 | Forbidden element that may not be used. |
| 0..1 | Optional element that may or may not be used. If used, only a single occurrence is allowed. |

| | |
|------|--|
| 0..n | Optional element that may or may not be used. If used, only a limited number of occurrences are allowed. |
| n | Element that must have a fixed number of occurrences (>1). |
| n..m | Mandatory element that has a minimum and a maximum number of occurrences set. |
| n..* | Mandatory element that must have a minimum number of occurrences, no maximum number of occurrences set (<i>unbounded</i>). |

Table 3: Cardinality Symbols

Clicking the new button will open a dialog box that allows modifying the *minOccurs* and the *maxOccurs* values. Compatibility is preserved according to the current compatibility level. The drop-down boxes for the minimum and maximum number of occurrences show only the values allowed in the current compatibility level. See section [Compatibility levels](#) for further details on the available compatibility levels and associated restrictions.

Optionally an **annotation** and a condition may be entered. Annotations consist of a category, a language and the annotation text. Use the button to enter your annotation. The condition is chosen with the drop-down list that shows all conditions specified for the profile by the user. If a condition is chosen, the modification will only be applied if the given condition evaluates to *true*.

Use the [condition manager](#) to view, edit or create conditions. To change a conditional modification back to an unconditional modification, select the top element '-none-' from the drop-down list of conditions.

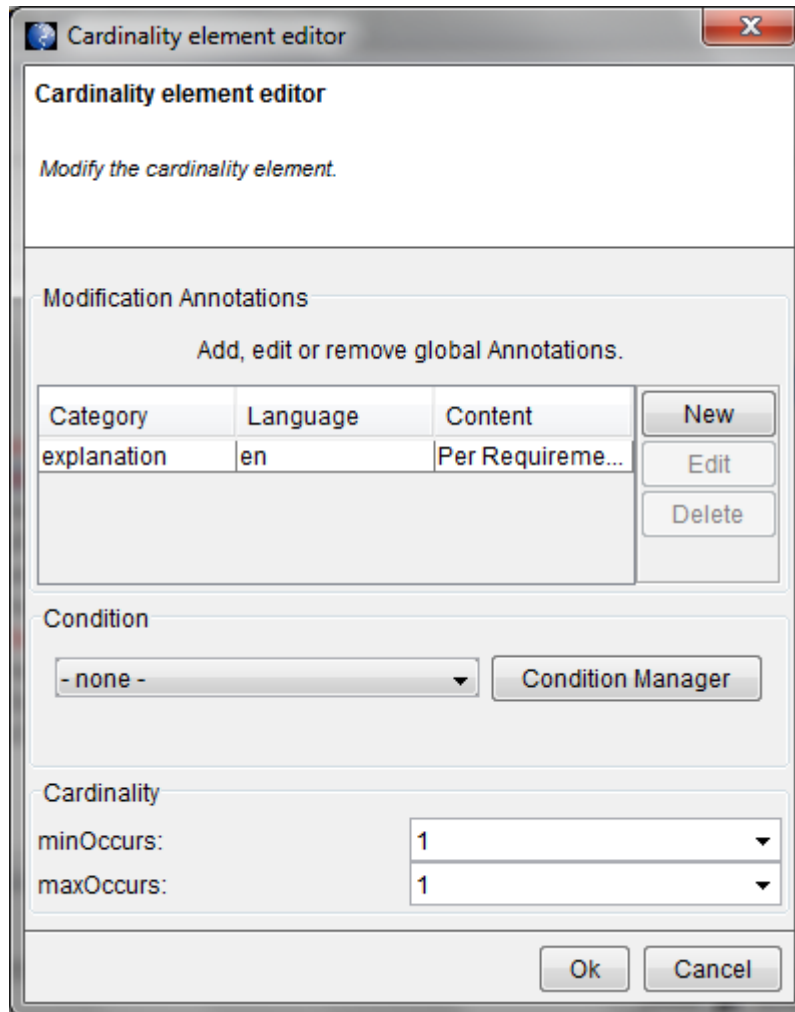




Figure 17 The Cardinality Element Editor

2.3.4. Value Type Modifications

Value type modifications are possible for the value type elements denoted with the respective description *[ValueTypeElement]* in the navigation tree on the left side. Simple content elements (icon: ) and attribute elements (icon: ) are valid value type elements. Modifying a value type opens a wizard that offers four different replacements for the current value type and will take you step by step through all possible modifications.

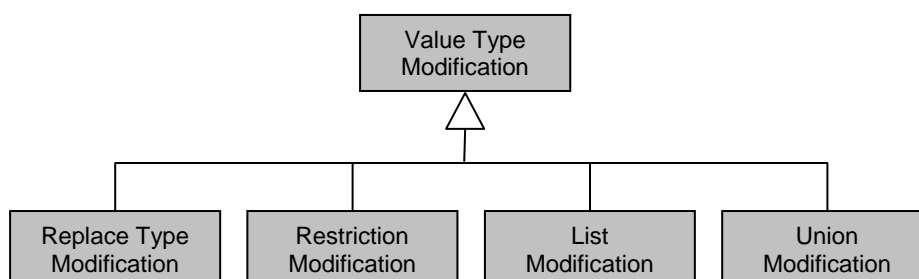


Figure 18: Value Type Modifications

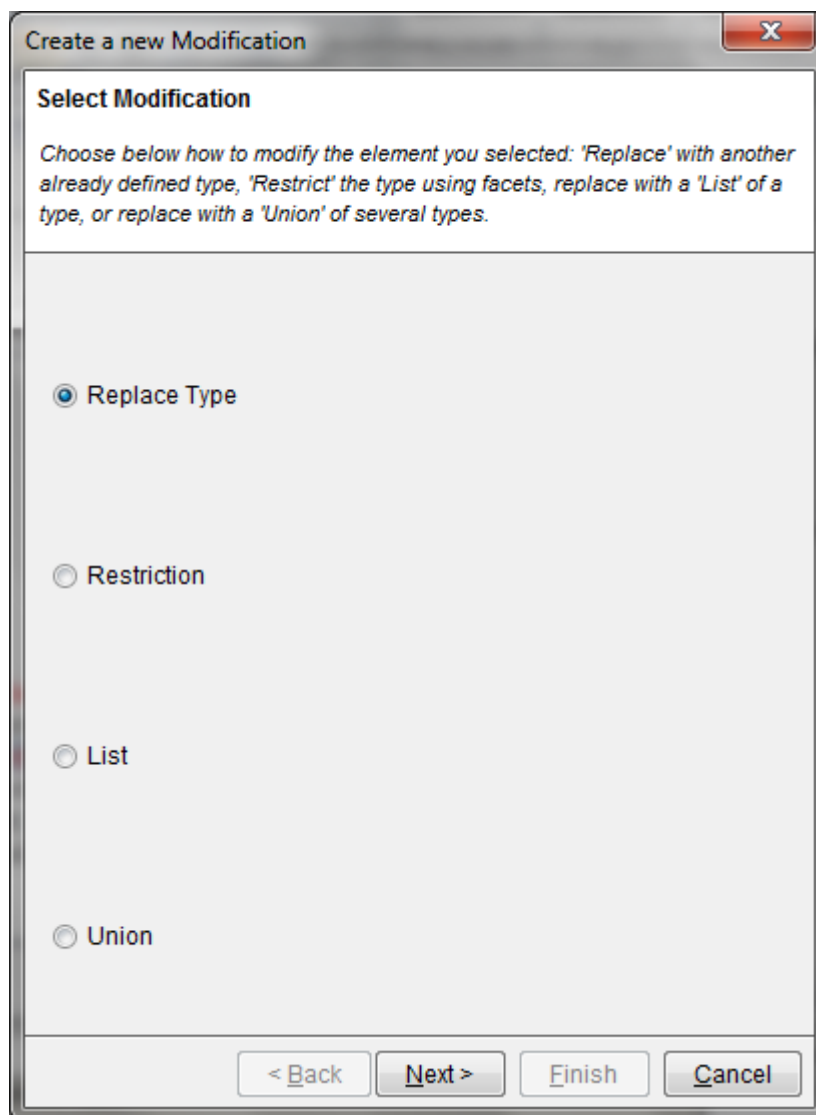


Figure 19 The new modification dialog for value types

Restriction of the selected type

This type of modification does not modify the base type itself, but makes it possible to enter several restrictions (facets) on the value space of current type. Further information on the restrictions, the syntax and validation is specified in the W3C document *XML Schema Part 2: Datatypes* (<http://www.w3.org/TR/xmlschema-2/>).

Step 1: Select the element you want to modify in the navigation tree on the left side. Then press the **New** button on the right side.

Step 2: Select 'Restriction' and press the **Next >** button.

Step 3: Enter the restrictions needed. Depending on the element type different facets (restrictions) are available. Leave all other fields untouched to not change the current values.

- The tab *length* allows restricting the number of characters for a value. A fixed length, a minimum and a maximum length can be entered. More details on <http://www.w3.org/TR/xmlschema-2/#rf-length>.
- The tab *pattern* allows entering of a pattern, constraining the lexical value space of a data type to literals which match the given pattern. The pattern is given as a regular expression according to <http://www.w3.org/TR/xmlschema-2/#rf-pattern>.

- The tab *enumeration* makes it possible to define a set of values. The value space of the current element are then restricted to the values of this set only. See <http://www.w3.org/TR/xmlschema-2/#rf-enumeration> for more information.
- The tab *whitespace* can be used to specify the handling of whitespaces in values of this element. Allowed values are *preserve*, *replace* and *collapse*. The meaning of the values is given in <http://www.w3.org/TR/xmlschema-2/#rf-whiteSpace>.

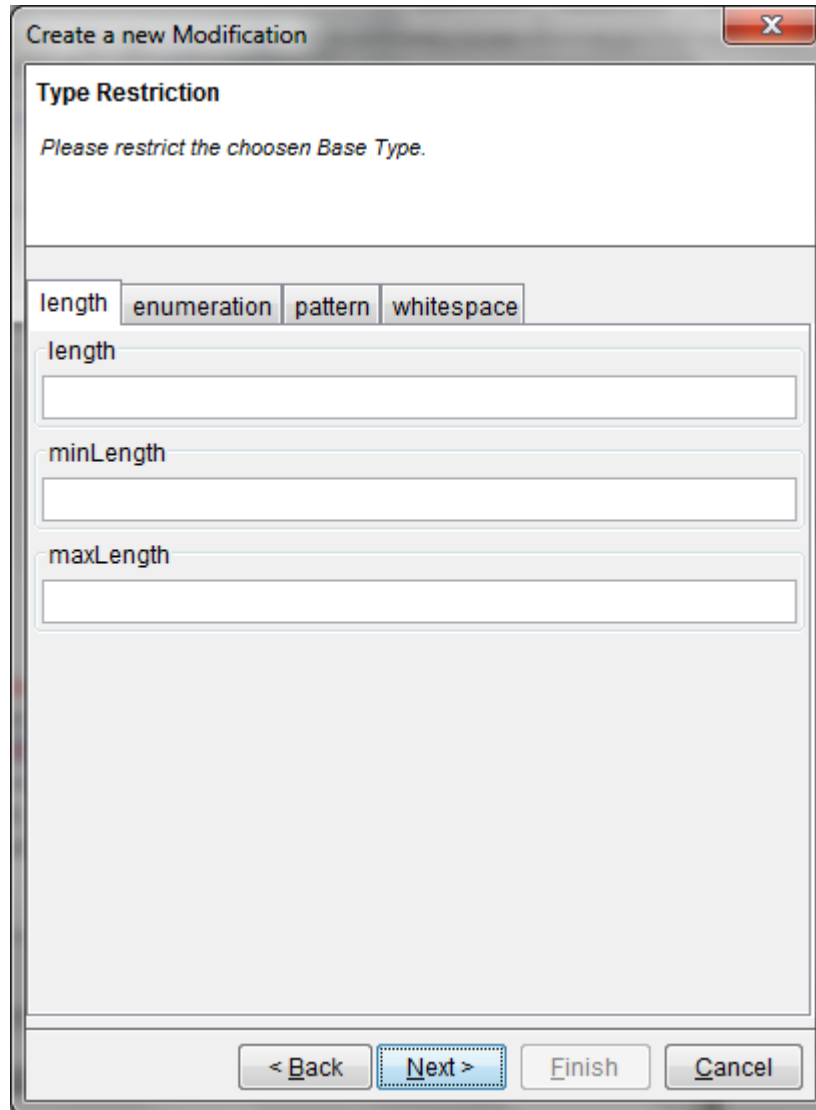


Figure 20 The Facets Modification Dialog

Press the **Next** button to continue, when all needed restrictions are entered.

Step 4: Enter a meaningful name for the new type that will be created based on the setting.

Step 5: In this step of the wizard, the following information can be entered:

- One or more *annotations* may be entered for this modification. To enter a new annotation, use the **New** button on the right side. In the opening dialog the text can be entered. Each annotation is bound to a category and a language, which can be selected below. The category can be selected to be *explanation* or *rationale*. The language can basically be any language string as defined in <http://www.w3.org/TR/xmlschema-2/#language> (based on RFC 1766). The most common language strings are available in the drop-down-box, but it can also be overwritten. Corrections are possible using the **Edit** and **Delete** buttons.

- A *condition* that must be satisfied, to activate this modification. Use the button **Condition Manager** to enter a new condition or modify an existing condition. The conditions are used to determine whether a modification is applied at all, and respectively which modification is applied when several modifications are given (see section '[Order of Modifications, Reordering of Modifications](#)' above for more information and section [Condition Manager](#) for more details).
- The *usage* declaration of attributes allows the values: *optional*, *prohibited* and *required*. See <http://www.w3.org/TR/xmlschema-1/> for further explanations.
- An initial value for this element can be used. If it is given as a *default* value, the user of this profile can overwrite this value entering an own value. If the value is given as *fixed* value, no changes are allowed by the user.

Click **Next** button to continue.

Create a new Modification

Modification Attributes

Optional further details: 'Annotations' explain and document this modification; 'Conditions' make this modification conditional; 'Usage' how to use; 'Value' is used for preset values.

Modification Annotations

Add, edit or remove global Annotations.

| Category | Language | Content | |
|----------|----------|---------|------------|
| | | | New |
| | | | Edit |
| | | | Delete |

Condition

- none - **Condition Manager**

Usage

REQUIRED

Value

DEFAULT

< Back **Next >** Finish Cancel

Figure 21 The modification attributes dialog for value type modification

Step 6: Enter Attribute Extensions, if needed, and click **Next** button.

Step 7: Finalize this entry by clicking the **Finish** button.

Replace by an existing type

Use this option to replace the predefined type with some other type. The value space of this element is then set to the value space allowed for the new type.

Step 1: Select the element you want to modify in the navigation tree on the left side. Then press the **New** button on the right side.

Step 2: Enter a meaningful name for this modification and press the **Next** button.

Step 3: Select 'Replace Type' and press the **Next** button.

Step 4: Select the new base type you want to use instead of the current base type from the drop-down list and press **Next** to continue.

Step 5: Enter or modify (if required) the settings for annotations, conditions, usage and default value. See [Step 5](#) of the previous section for details. Press button **Next** when done.

Step 6: Enter Attribute Extensions, if needed, and click **Next** button. (This is only visible if extension points are enabled in the preferences.)

Step 7: Finalize this entry by clicking the **Finish** button.

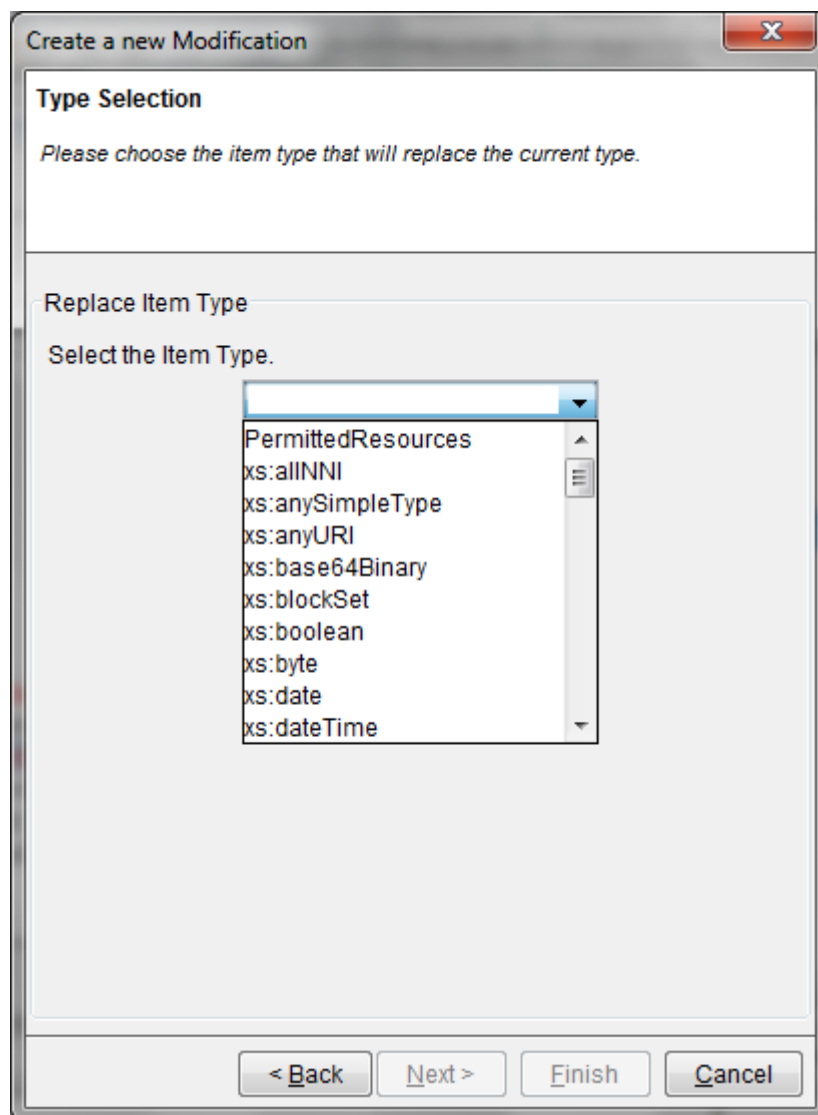


Figure 22 The 'Replace Type' Modification Dialog

Replace by a union of existing types

Use this option to replace the current type with a union of several base types. The value space of this element will then be changed to the union of all value spaces given in the list of types.

Step 1: Select the element you want to modify in the navigation tree on the left side. Then press the **New** button on the right side.

Step 2: Select 'Union' and press the **Next** button.

Step 3: Compile a set of type to be joint in a union. To add a type to the set of types, select the type from the drop-down list and press the **Add** button. The type will now appear on the list below. To remove a type from the list, select the type from the list and press the **Delete** button. Press **Next** when all required type are on the list.

Step 4: Enter further information (if required) as described [above](#) and press the **Next** button.

Step 5: Enter a meaningful name for the new union type and press the **Next** button.

Step 6: Enter annotations, conditions, usage information and a default value as [explained above](#) and press **Next**.

Step 7: Finalize this modification by clicking the **Finish** button.

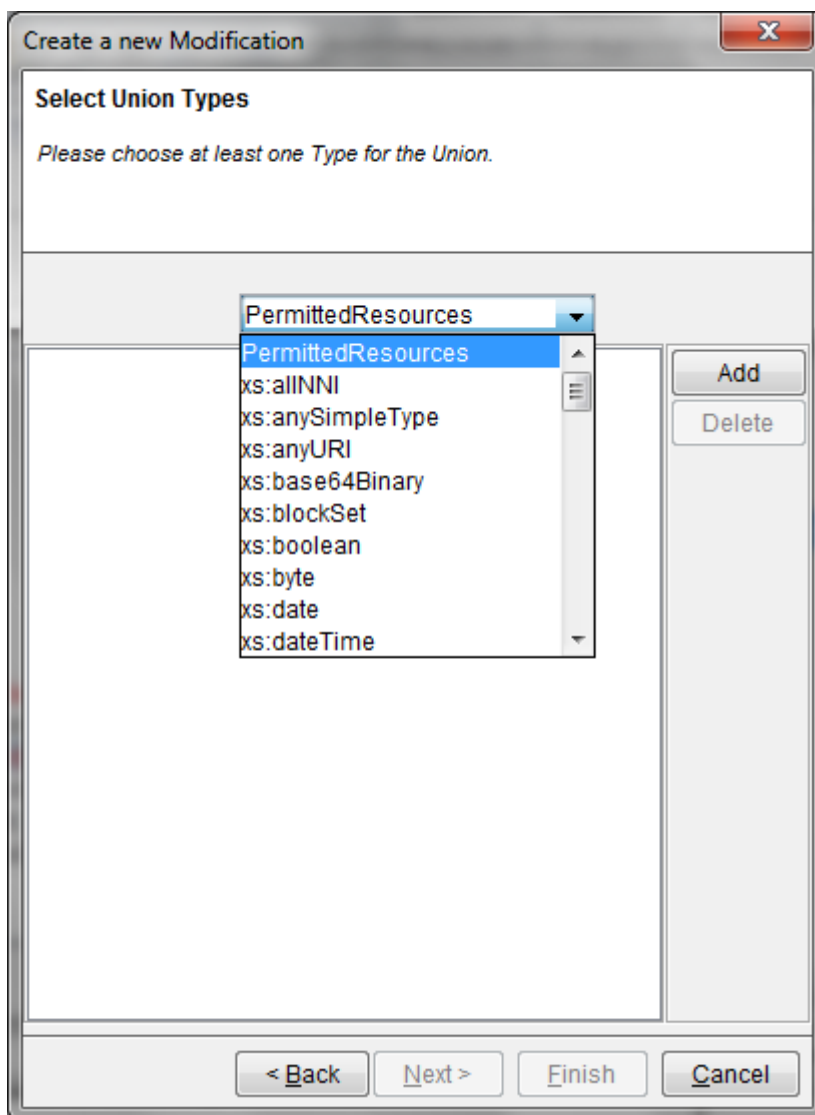


Figure 23 The 'Replace with Union' Modification

Replace by a list of existing types

Use this option to define a data type that will be derived from an atomic or union data type. The value space of this element is then composed of a finite-length sequence of values of the specified type.

Step 1: Select the element you want to modify in the navigation tree on the left side. Then press the **New** button on the right side.

Step 2: Select 'List' and press the **Next** button.

Step 3: Select the base type of the list from the drop-down list. Press **Next** to continue.

Step 4: Enter further facet information as explained in [Step 3](#) above. Then press the **Next** button.

Step 5: Enter a meaningful name for this modification and press the **Next** button.

Step 6: Enter annotations, conditions, usage information and default value as explained in [Step 5](#) above, if required, and click **Next** button.

Step 7: Finalize this entry by clicking the **Finish** button.

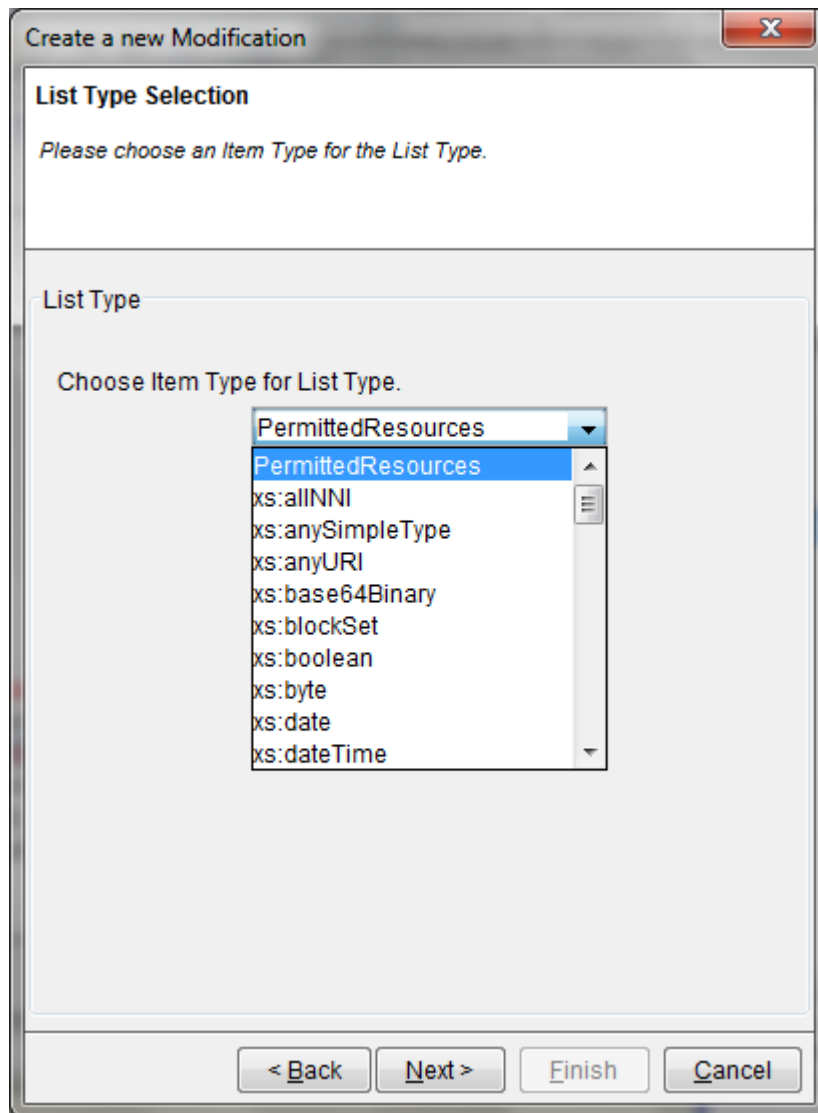


Figure 24: The Replace with List Type Modification

2.3.5. Global and Local Modifications

Sometimes an element of a base XML schema (*referring element*) is defined using a reference to another schema element (*referred element*). More than one *referring element* can exist, pointing to the same *referred element*. If the *referred element* is modified, this of course has impact on **all** *referring elements*.

When a modification must be made to the referring element, a decision must be made whether to do a global and local modification.

Global Modification: When the *referred element* is modified, this is a global modification. This modification will affect any other schema element that has a reference to this element.

Local Modification: This modification is made to a *referring element*. It will modify only this schema element. It leaves the *referred element* untouched, thus having no influence on all other elements referring to the *referred element*.

IMS SchemaProf is able to make local and global modifications for an application profile. If a schema element in the navigation tree is shown in grey coloured text, this means that this is a *referring element*. By default no modification can be made to this element. This forces the user to decide whether to make a global or a local modification at this point.

To make a global modification, the user must switch to the position in the navigation tree where the *referred element* is found. The easiest way to find the global definition is right-clicking the *referring element* and clicking 'Focus Global Element' from the context menu. Now you can modify this global definition. Making a modification to this *referred element* will affect all XSD elements referring to this element.

To make a **local modification**, the user must toggle the activation state of the *referring element* first. This is done by right-clicking on the *referring element* in the navigation tree and selecting 'Activate'. This changes the text colour to black and allows from now on making (local!) modification to this XSD element. These modifications will be realized as local modification in the application profile, influencing only the current element and leaving the *referred element* and any other XSD elements untouched.

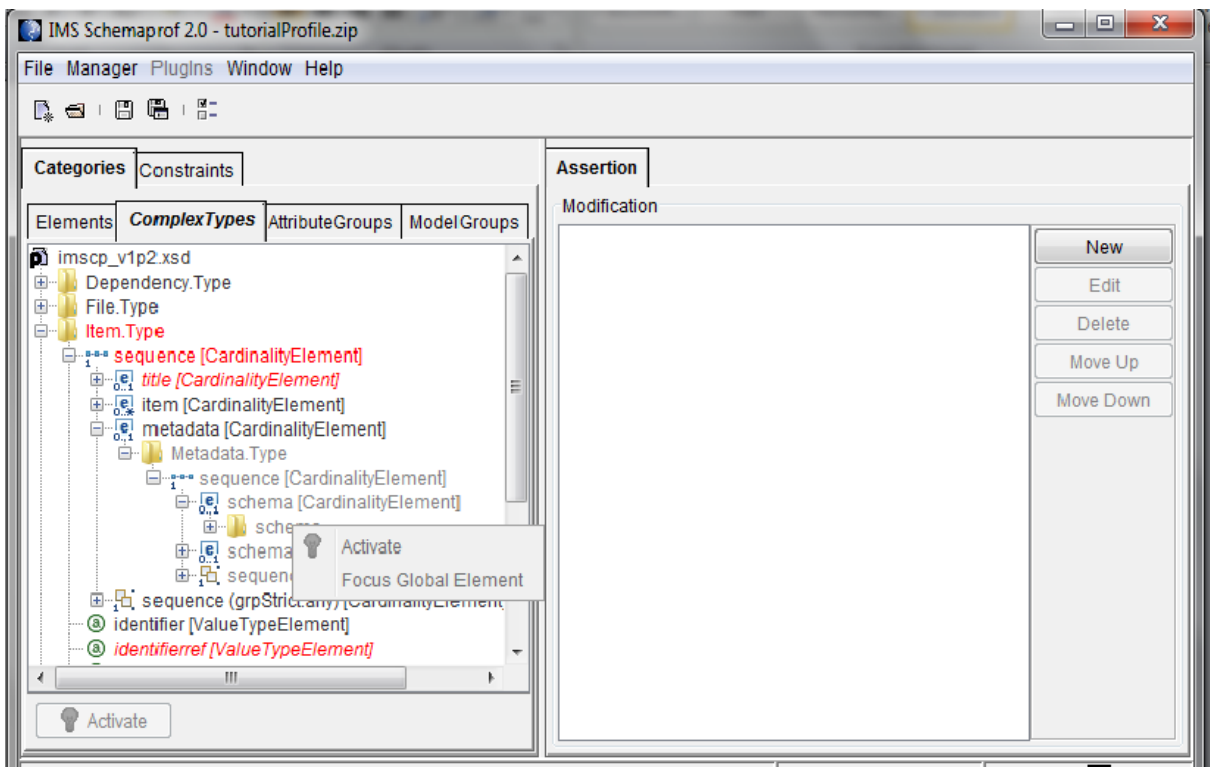


Figure 25: Deciding for a local or global modification of a referring element.

The internal realization of the local modification is done by defining a new XSD element type, making a copy of the *referred element* under a different name, and replace the reference of the with a type attribute pointing to the new XSD element type. All modifications made are applied to this new XSD element type only. This leaves the *referred element* and all elements referring to it untouched.

Note: A local modification is always based on a copy of the original specifications type that is referred (*referred element*). If this type definition of the *referred element* was modified in your profile before, this modification is not considered in any local modification of this type. The local modification is always based on a copy of the unchanged type definition as found in the original specification.

2.3.6. Extending a specification

Mild Extensions and Domain Profiling Extensions

Modifications on wildcard elements are possible where the wildcard elements *any* and *anyAttribute* signal a foreseen extension point in the XML schema. Selecting an *any* or *anyAttribute* element in the navigation tree on the left side shows the configuration of this element in the base specification on top of the right side of the window. Press the **New** button on the right side to modify this extension point.

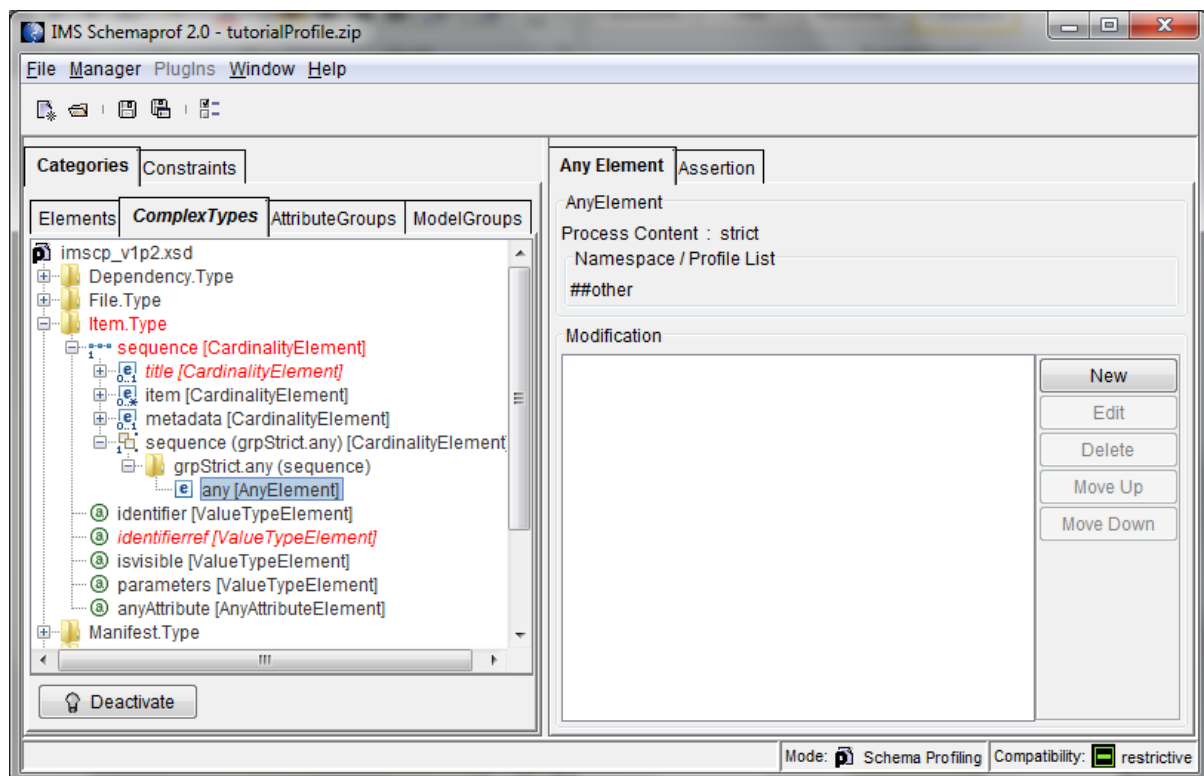


Figure 26 Extension Point Modification

The modification dialog allows redefining the following setting for this extension point:



Process Content

This value specifies how the XML processor should handle validation against the elements specified by this any element. See Table 4 for the possible values and their meaning.

| | |
|---------------|---|
| <i>strict</i> | The document processor must validate the given document element against the allowed namespaces and profiles. If the schemas and profiles are not available and no validation is possible, the validation will fail. |
| <i>lax</i> | The document processor will try to validate the element against the namespace or profile. If the schema or profile is not available, no validation error is reported. |
| <i>skip</i> | No validation of the element will be performed during schema validation. |

Table 4 Processing options for extension points ('any' elements)

Namespace / Application Profile List

This list contains the allowed namespaces and/or application profiles. The document instances will be restricted to contain only those elements from the defined namespace and profiles at the position of the current extension point. More than one namespace or profile may be defined; namespaces and profiles can be mixed. In the list of allowed namespaces and profile, the namespaces are identified by the  icon and application profiles are identified by the  icon.

Using other application profiles, while profiling a specification's extension point, is one form of domain profiling. For more information on domain profiling please refer to the [Telcert website](http://www.telcert.com)² or the [IMS Abstract Framework Glossary](http://www.ims-global.org/af/afv1p0/imsafglossaryv1p0.html#1523383)³.

Use the first dropdown list to select whether you want to add a namespace or an application profile to the list. The second dropdown list offers the options which differ for regular namespaces and application profiles. See Table 4 and Table for the list of available values and their meaning.

Use the **Add** button to add the currently selected chose of a namespace or profile to the list. Use the **Profile Manager** button to start the profile manager for the management of profiles used and possibility to quickly switch editing to the other profile.

| | |
|---------------------------------|---|
| Browse... | Selecting this line will open a file dialog to choose an application profile XML file from the local file system. |
| <i>Choose profile from list</i> | Any application profile used before in this profile will be offered to the user in the dropdown list for easy re-use. |

Table 3: Possible values for profiles

² <http://www.opengroup.org/telcert>

³ <http://www.ims-global.org/af/afv1p0/imsafglossaryv1p0.html#1523383>

| | |
|-----------------------------------|---|
| ##any | “Any well-formed XML from any namespace (default).” ⁴ This namespace makes no restrictions at all, therefore no other namespaces or profiles are allowed in <i>IMS SchemaProf</i> when ##any is on the list. |
| ##local | „Any well-formed XML that is not qualified, i.e. not declared to be in a namespace. “ ⁴ |
| ##other | “Any well-formed XML that is from a namespace other than the target namespace of the type being defined (unqualified elements are not allowed).” ⁴ Like ##any, including this namespace in the list will prevent adding any other namespace or profile. |
| ##targetNamespace | “Any well-formed XML belonging to any namespace in the (whitespace separated) list; ##targetNamespace is shorthand for the target namespace of the type being defined“ ⁴ |
| <i>Choose namespace from list</i> | Any namespace used before in the current profile and thus known will be display in the list for re-use at another location and can be selected. |
| <i>Typing another namespace</i> | To be able to add any further namespace, the user can type any namespace into the dropdown filed and add it to the list. |

Table 4: Possible values for namespaces

Cardinality (only type *any* elements): The cardinality of this element can be modified by entering new values for the *minOccurs* and *maxOccurs* values.

Condition: A *condition* can be chosen from the dropdown list. This condition must be satisfied to activate this modification. Use the button **Condition Manager** to enter a new condition or modify an existing condition. Conditions are XPath expressions which are evaluated in the instance document at test time. All elements in these XPath expressions must have a prefix which is defined in the Namespace Manager.

Only if the XPath expression in the condition evaluates to true in a file to be tested validation for the respective modification will be triggered. The conditions are used to determine whether a modification is applied at all, and respectively which modification is applied when several modifications are given (see section ‘Order of Modifications, Reordering of Modifications’ above for more information).

While unconditional modifications are translated by SchemaProf into modifications of the base XML Schema, this is not possible for conditional modifications. Such conditional modifications are represented as Schematron rules.

⁴ <http://www.w3.org/TR/xmlschema-0/#nsTable>

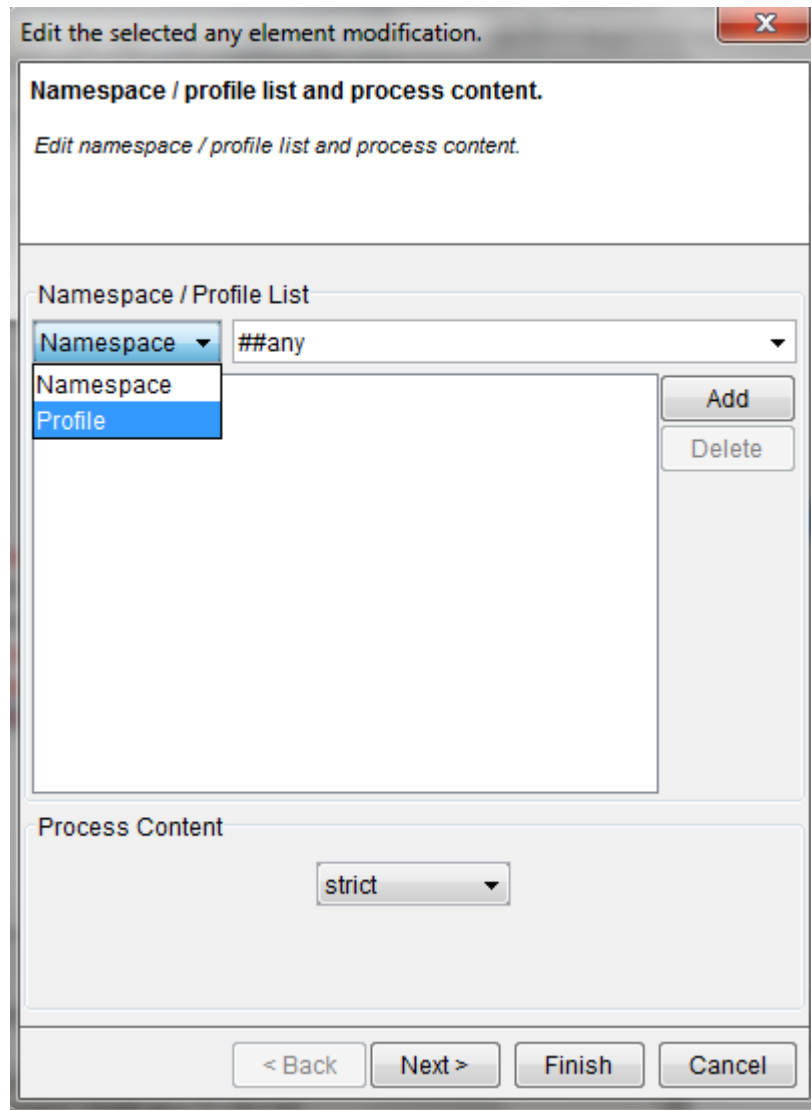


Figure 27 Extension point modification window

Modification Annotations: One or more *annotations* may be entered for this modification. To enter a new annotation, use the **New** button on the right side. In the opening windows the text can be entered. Each annotation is bound to a category and a language, which can be selected below. The category can be selected to be *explanation* or *rationale*. The language can basically be any language string as defined in <http://www.w3.org/TR/xmlschema-2/#language> (based on RFC 1766). The most common language strings are available in the drop-down-box, but can be overwritten. Corrections are possible using the **Edit** and **Delete** buttons.

Click the **OK** button when all information is entered. The modification can be edited later on using the **Edit** button or removed using the **Delete** button and rearranged using the **Move Up** and **Move Down** buttons. Modifications should be always annotated with explanations. This is particularly important for conditional modifications.

Wild Extensions

An XML schema should only be extended at pre-defined extension points that are foreseen by the schema developer and identified in the schema by the names *any* and *anyAttribute*. Those

'legal' extensions are denoted as *mild extensions* and their usage is described in the previous section.

Unfortunately there are rare situations where extensions to a schema are necessary and no extension points have been foreseen by the schema designers. For these cases it is possible to extend the schema at other positions, making a so-called *wild extension*.

We strongly discourage the usage of wild extensions.

Using wild extensions will destroy any compatibility to the base specification and software that is not fault-tolerant might behave unexpectedly or even crash when wild extensions are used.

If the usage of wild extension can not be avoided select the element that is considered the predecessor element of the extension element in the navigation tree. Press the **New** button on the right side to enter an extension at this point. This will pop up a window for text input where you can enter the new element in pure XML code.

Entries made for wild extensions are not checked at all except for being well-formed. Since wild extensions are beyond any specification or standard only very experienced users are expected to use this feature and no further support is provided.

2.3.7. Profiling Extensions

The use of this feature is NOT recommended.

The XML file format of the application profile is also specified by an XML schema. The previous section mentions that in rare situations extensions are necessary but have not been foreseen by the schema designer.

To allow future extension to the schema profiling as used and supported by IMS SchemaProf, the profiling schema has some extension point that are not used by the *IMS SchemaProf* profiling tool itself. They are intended to be used for community specific extensions.

IMS SchemaProf, although not able to semantically 'understand' the meaning of the profiling extensions allows the user to enter them at the pre-defined extension point. This can be done by selecting the root element in the navigation tree on the left side. The working area on the right side now show five tabs apart from the 'Global Schema Annotations' tab. Select a tab and enter your extensions by pressing the **New** button and entering the XML code as described in the previous section.

2.3.8. Assertions

IMS SchemaProf allows the definition of assertions in profiles that can be used as a further instrument to ensure compliance and consistency of document instances with profile and specifications respectively.

Assertions are specified in form of XPath expressions that are evaluated to either *true* or *false*. The possibilities what kind of conditions can be expressed are the same as for the conditions from conditional modifications.

But the logic behind them is different. Assertions are not tied to a particular definition of the base schema but not to a schema modification. The XPath expression of an assertion must be evaluated by a tool (e.g. the TELCERT test system) for a specific document instance. If it evaluates to *true* this consistency check is satisfied what should hopefully happen for all your document instances. But if it evaluates to *false*, then this consistency check failed and the

whole document instance is considered to be invalid! (For conditional modifications the XPath expression triggers a modification when evaluated to *true* and nothing happens if it evaluates to *false*.)

Creating a new Assertion

Assertions can be defined for most of the elements shown in the navigation tree. Make sure that the *Categories* tab is selected on the left side and select the element from the navigation tree where you want to attach the assertion to. Now a second tab, labelled *Assertion*, should be visible on top of the right side. Click it to get the assertions view.

To create a new assertion, now click the **New** button on the right side. A new dialog will open now with a text field where you can enter the XPath expression to be evaluated. Please refer to the section [Declaring XPath Conditions](#) for help on how to specify XPath expression. Click **Next** to continue.

On the next panel it is possible to make annotations and even to use a condition thus making it a conditional assertion. Click **Next** to continue and **Finish** to finalize this assertion.

Editing an Assertion

To edit an existing assertion select it from the list of assertions and click **Edit**, or double-click the assertion in the list.

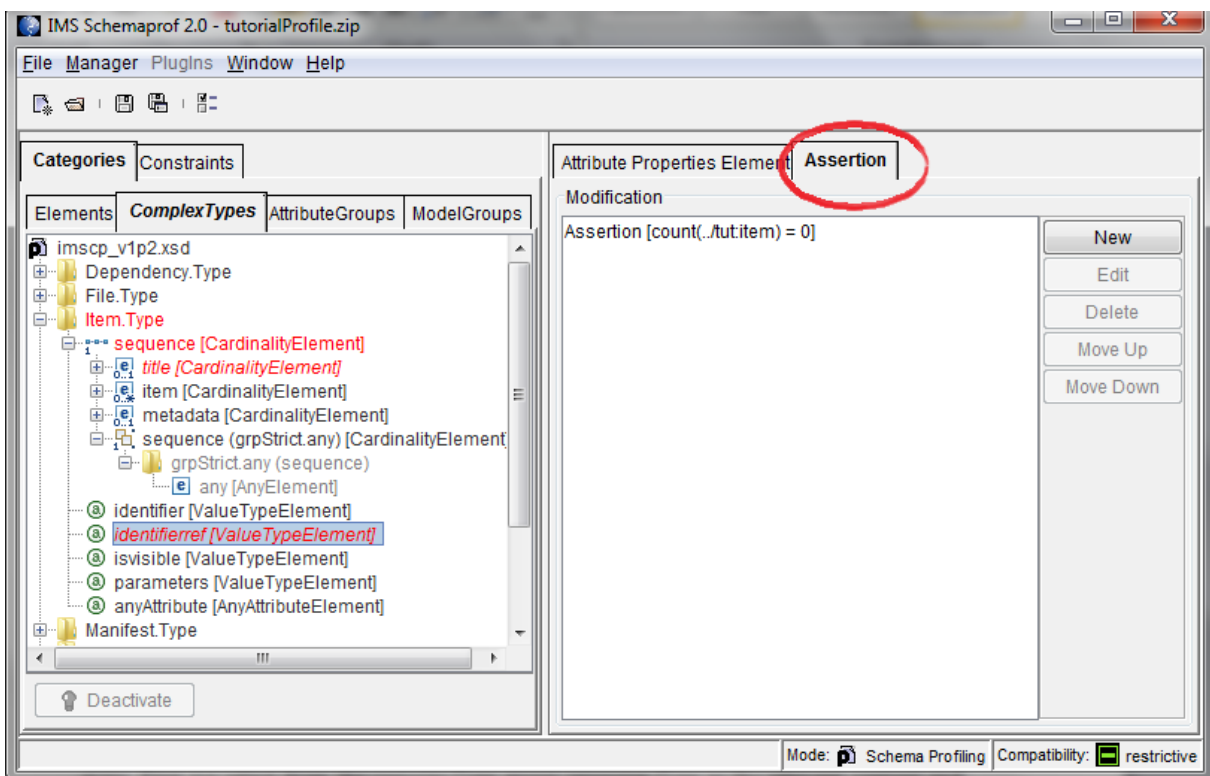


Figure 28 Adding a new Assertion

Deleting an Assertion

Select the assertion from the list, click **Delete** and confirm the deletion.

Moving Assertions Up/Down

If more than one assertion is defined, it is possible to rearrange the assertions. This may be important for conditional assertions. Select an assertion and press the **Move Up** and **Move Down** buttons to rearrange.

2.3.9. Additional Constraints

The modifications introduced so far restrict the structure of XML documents. However, data are frequently delivered in packages. Additional constraints are the method to pose conditions on the structure of these packages. Therefore additional constraints have a global character.

With additional constraints it is possible to define extra information that is used in other tools to check constraints that are often found in the information model of specifications but which are not found (in machine readable form) in the XML binding. If for example an attribute 'fileURI' is supposed to point to an existing file that must be an HTML file and of the size defined in the further attribute 'fileSize', this can be expressed by additional constraints.

The main advantage of defining constraints in a machine readable form is their automated verification of the constraint through appropriate tools like the IMS test systems.

Currently two types of constraints are supported by IMS SchemaProf. **File constraints** define the location of a file and several properties this file must comply to like file extensions, file size or file type. **Vocabulary constraints** make it possible to tie certain element values to a vocabulary (internal or external) where this value must be contained.

Adding a new constraint

To define new constraints please click on the 'Constraints' tab above the navigation tree in the upper tabs row. The seconds row of tabs is only visible when the 'Categories' tab is activated and will disappear when switching from the default view. Now a different navigation tree is displayed and below two buttons are shown to add or delete new constraint.

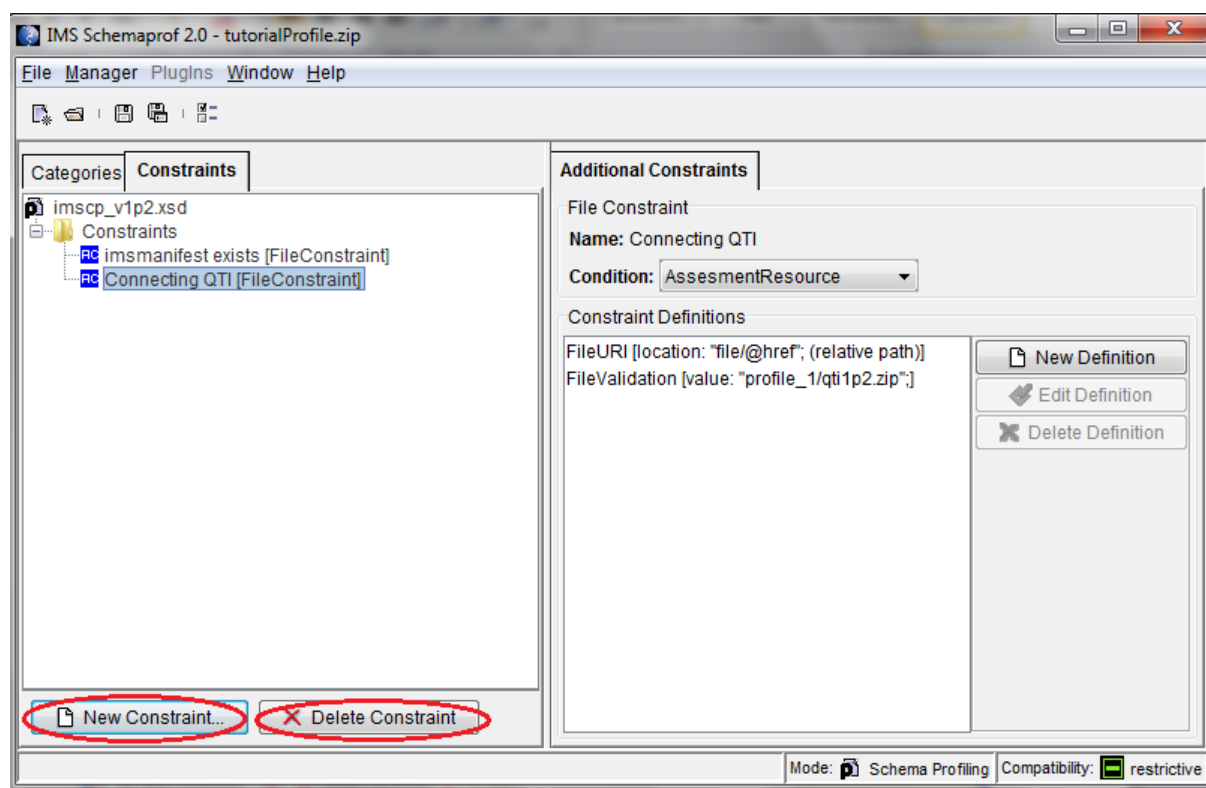


Figure 29 Adding and Deleting Constraints in IMS SchemaProf

Click the button **New constraint...** to start a new constraint. In the dialog opening now, choose whether to add a file constraint or a vocabulary constraint and give the constraint a name.

Then click **Next >** to continue. For file constraints a file URI must be defined immediately since this is a mandatory definition that may be modified but never deleted. Then click **Finish** to get back to the main window where further constraint definitions can be inserted.

Deleting a Constraint

To delete an additional constraint select the constraint in the navigation tree and then click the **Delete Constraint** button below the navigation tree. If the **Categories** tab is currently selected it is necessary to click on the **Constraints** tab before to get the correct view of the navigation tree.

Constraint Definitions

Each constraint defined in a profile consists of one or several *definitions*. Each definition specifies a specific property that any document instance must satisfy, e.g. a certain file type or file size. A document instance can only be considered a valid document if all constraint definitions are satisfied, e.g. the file type must be HTML and the size must be less than 1 MB. Within the definitions alternative values may be possible, e.g. the file type restricted to DOC or PDF. Each definition can occur only once within the same constraint, e.g. it makes no sense to have two declarations of a maximum file size.

Additional constraints can be *static* or *dynamic*. Parameters of static constraints are defined by the author of the application profile. For example a static constraint can require that there is a file named *imsmanifest.xml* at the root of a data package. In contrast, parameters of dynamic constraints are defined as values of specific attributes in instance documents. For example a dynamic constraint can require that the *href* attribute of a specific element points to a file with a specific mime type.

Adding a Constraint Definition

To add a constraint definition, first select the constraint in the navigation tree on the left side. The constraint must exist already; see section [Adding a new constraint](#) on how to create a constraint. On the right side you can see the existing definitions of that constraint if any exist.

Now click the **New definition** button on the right side to add a new definition to the list. Depending on the constraint type (file constraint, vocabulary constraint) different definitions will be available. It is possible that some definitions are disabled if they are in use already or conflict with definitions in use.

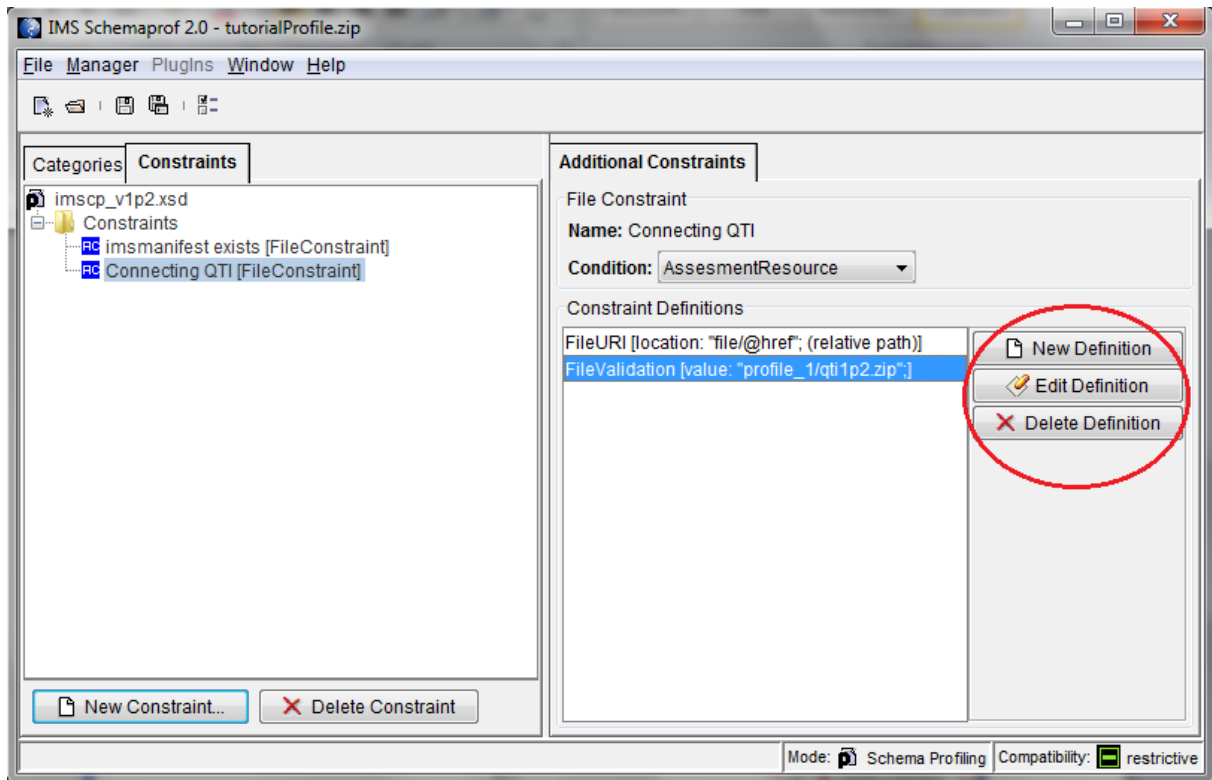


Figure 30 Adding, Editing and Deleting a new constraint definition

Deleting or Editing a Constraint Definition

To delete a constraint definition, select the constraint first in the navigation tree on the left side. Now you see the list of existing definitions in the list on the right side. Select the constraint definition you want to delete from the list and click the **Delete Definition** or **Edit Definition** button.

File Constraints

File constraints define the location of a file and several expected properties of that file. All values can be defined by pointing either to a schema element (value type or attribute) or by setting a user defined value. Each setting made is called a constraint definition. Several constraint definitions can be combined, but not all combinations are allowed. *IMS SchemaProf* will only offer you those constraints that are allowed at the current stage.

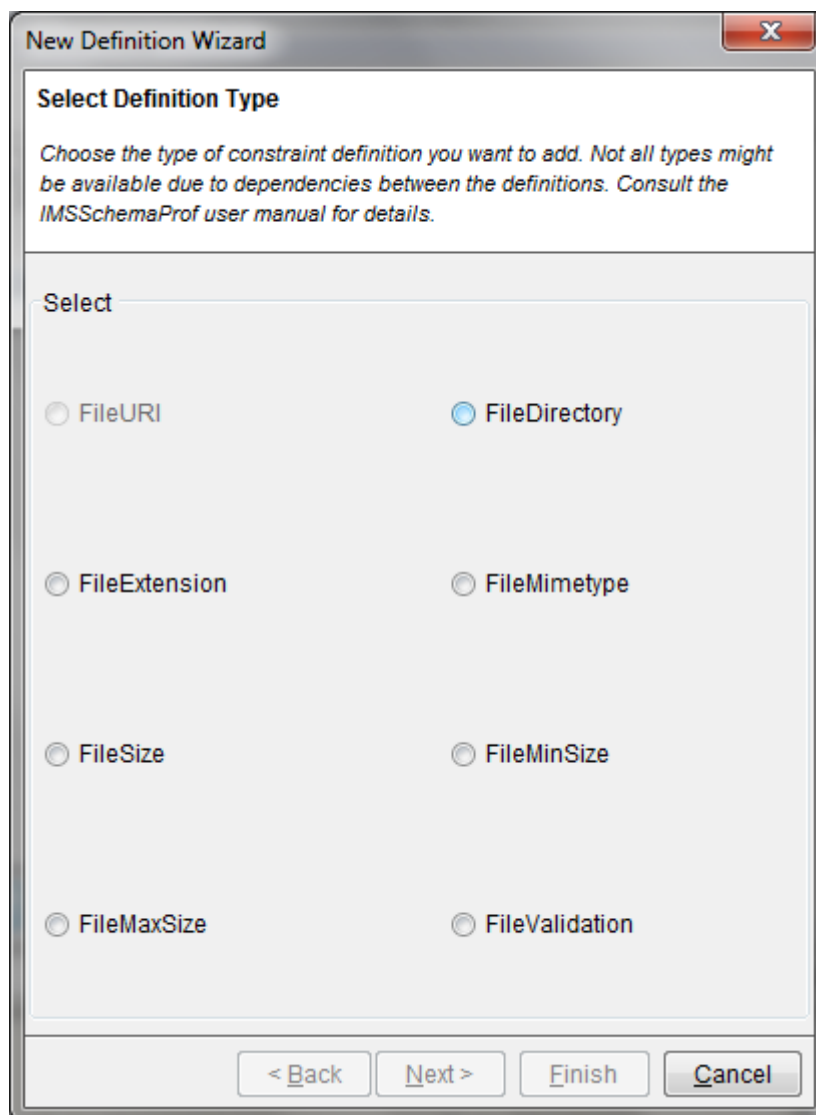


Figure 31 The Definitions of a File Constraint

FileURI Definition

The *FileURI* definition declares the actual file location. As explained above, this can be done statically with a user-defined URI value, or dynamically by choosing a schema element that will later hold the file location. The dynamic definition is probably the more commonly used option.. The static definition on the other hand is more intuitively used for profiling and less abstract to understand. The *FileURI* is mandatory.

FileExtension Definition

With the *FileExtension* definition it is possible to define a list of file extensions, e.g. “html, htm, php”. The file extensions are of course alternatives and each filename in a document instance must end on one of the file extensions. Example: “http://mydomain.org/index.html” is a valid filename, while “myScript.pl” is no allowed.

FileSize Definition

This definition specifies the absolute size of a file. This definition typically points to an attribute where a system is expected to continuously update the exact file size as meta data. Setting a user-defined value in the profile is probably unrealistic but still allowed.

Additionally a divisor must be set, defining whether the given size is expected to be byte (divisor: 1), kilobyte (1024) or megabyte (1024x1024). A user-defined divisor is also possible.

The *FileSize* definition cannot be used in combination with the *FileMinSize* and *FileMaxSize* definitions. If *FileSize* is defined *FileMinSize* and *FileMaxSize* are not available and vice versa.

FileMinSize, FileMaxSize

These definitions allow you to set a minimum or maximum file size. The size can be given, as for *fileSize*, in byte, KB, MB or any user-defined divisor. Setting one or both of these definitions prevents the use of the *FileSize* definition (absolute size).

FileDirectory

This definition defines a directory where the file is expected. This can be a document attribute within the document instance or a user-defined and therefore fixed value. If defined, the file must exist in this directory.

FileMimetype

With this definition a list of MIME types can be specified, e.g. 'application/pdf' or 'text/xml'. If more than one MIME type is given, they are considered to be alternatives. If defined, the file of this constraint is expected to comply with one of the given MIME types.

FileValidation

The *FileValidation* setting allows you to specify a XML schema or a DTD file. The actual file in the document instance is then expected to be compliant to this XML schema or DTD and it's the task of the test system or other tools to verify the compliance.

Vocabulary Constraints

Vocabulary request the use of particular external vocabularies at particular places. NOTE: While vocabulary constraints can be defined in SchemaProf, the IMS Test System will not check them. There is ongoing work to extend the IMS Test System with a vocabulary tester. For now it is recommended to define the permitted values for a vocabulary entry in the profile using a newly defined simple type that enumerates all permitted values.

Vocabulary constraints define a schema element and a vocabulary. The value of the schema element in a document instance must then be contained in the given vocabulary. Only these two constraint definitions are available.

VocabularyElement

This definition specifies the actual schema element that will later hold the actual value. Here a user-defined value makes no sense and is not allowed. The schema element can be an attribute or a value type element.

VocabularySource

The definition specifies the actual vocabulary. The vocabulary must be of the format [IMS VDEX](#) or [XVD](#). The location where the vocabulary can be found is defined through a schema element or a user-defined value. Additionally the vocabulary type must be set.

Editing of additional constraints is handled in greater detail in Section 3.

2.4. Annotations

IMS SchemaProf allows the user to make annotations at different levels and positions of the profile. The extensive usage of the annotation feature is recommended to improve the understanding and correct usage of the profile by users that are not involved in to profiling process. The annotations can always be made in one or more languages supporting multilingual applications.

The following subsections describe the available positions for your annotations.

2.4.1. Global Annotations

Global annotations are available to enter annotations that concern the application profile as a whole. To make a global annotation, select the root element of the base specification in the navigation tree. The working area on the right side will show several tabs on top; please select the tab *Global schema Annotations* and click on the button.

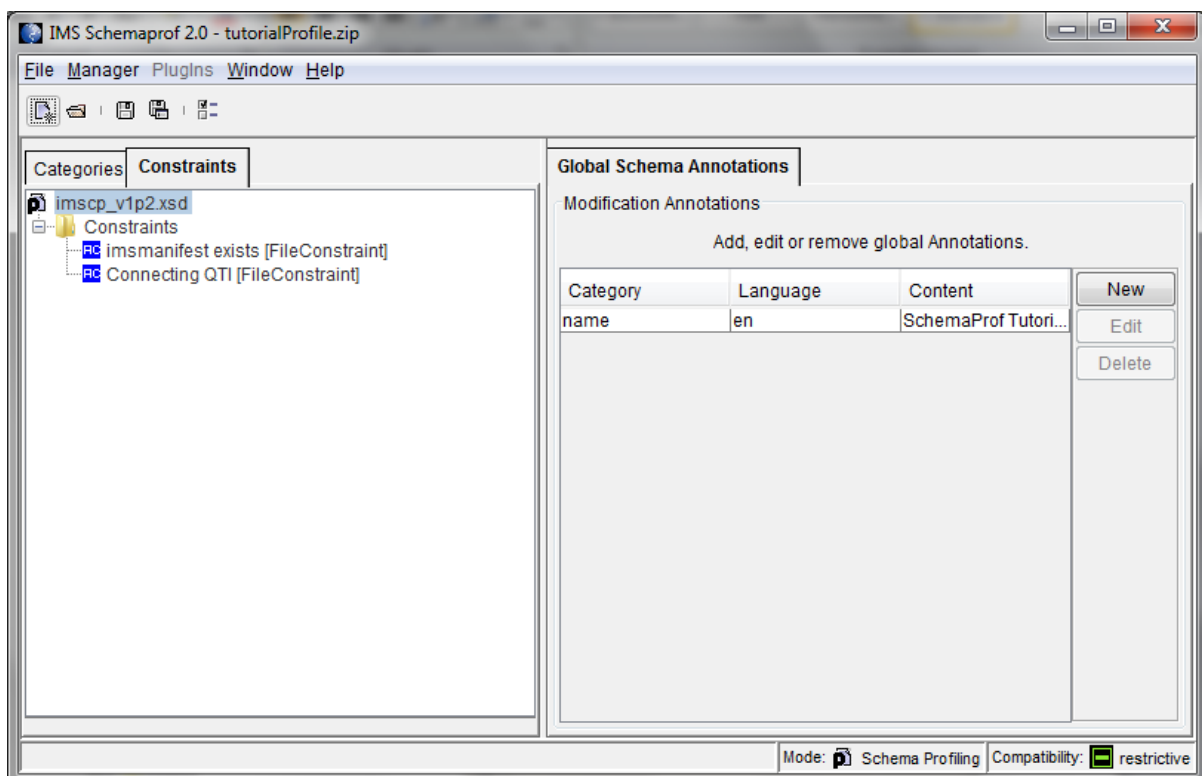


Figure 32: Global Annotations

Now one or several annotations can be entered by clicking on the button and entering the information. An annotation consists of the annotation text bound to a *category* and a *language*.

The *categories* available for the global annotations are *name*, *scope*, *policy*, *conformance* and *general*. It is important to define the *name* of the profile precisely as this will allow later to determine what a specific test system is for.

The *language* can basically be any language string as defined in <http://www.w3.org/TR/xmlschema-2/#language> (based on RFC 1766). The most common language strings are available in the drop-down-box, but can be overwritten.

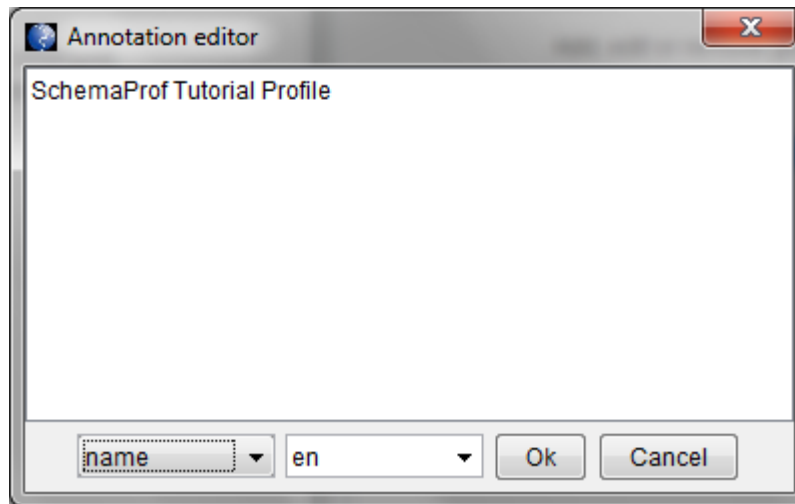


Figure 33: Inserting a New Annotation

Annotations can be edited and deleted by marking them in the list and clicking the **Edit** or **Delete** button.

2.4.2. Modification Annotations

For all types of modifications, it is possible to make annotations. Annotations for modification can be categorised to be either 'explanation' or 'rationale', and the language of the annotation can be specified by selecting one of the languages from the dropdown list or write your own language in this box. For every combination of category and language string only one annotation is allowed.

2.5. Profile Extension Points

Most specifications include extension points at several positions of the XML schema for specific and unforeseen information that must be include in a document apart from the foreseen basic data structure. The application profiling XML format is also defined as an open format with several extension points. This preserves the flexibility to inserting data into profiles that was not foreseen while developing the application profiling specification or is specific to your local requirements and/or may be supported by your own tools.

Since this is a very advanced idea than will probably be used by very few people and may confuse less experienced users, the control extension of the extension points must be enabled in the preferences. Open the preferences and enable the checkbox *Extension Points* under the *Global Setting* tab to make the extension points visible in the GUI.

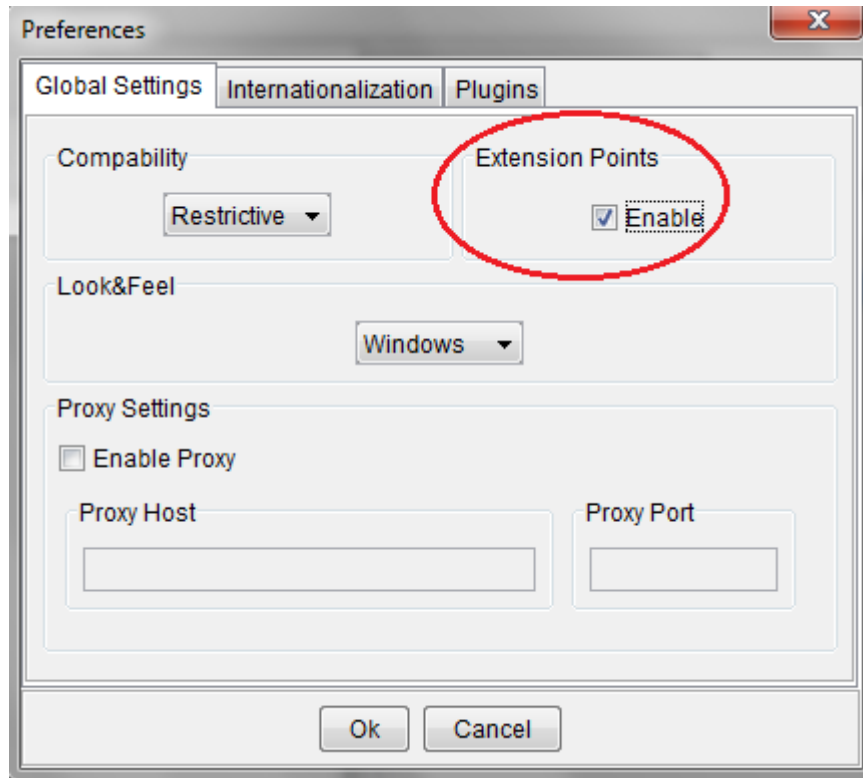


Figure 34 Enabling of Profile Extension Point.

Once enable the extension points are visible for these elements:

- Root element of the profile: the new tabs Annotation Extension, Definition Extension, Mapping Extension, Modification Extension and Section Extension are now available on top of the right side.

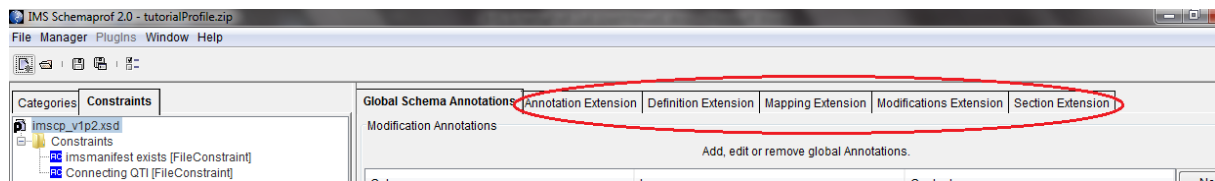


Figure 35 Extension points of the root element.

- For each of the complex types a *Modification Extension* tab is now available.

To enter your own XML data at these positions please enter the appropriate XML code into the text boxes and don't forget to declare any namespaces and prefixes to be used in the [namespace manager](#). NOTE: Currently Test Systems don't support the use of these extensions.

3. Tutorial

In this Section we demonstrate the usage of IMS SchemaProf to develop a domain profile based on the IMS Content Packaging specification and using predefined profiles of the IEEE LOM Metadata specification and the QTI Test and Assessment specification. All required files are located in the *SchemaProfTutorial* directory.

The profile which we are going to define in SchemaProf requires the following modifications of the IMS Content Packaging specification schema:

1. Files conforming to the profile must have the namespace "http://www.imsglobal.org/schemaprof/tutorial".

2. Each item must have exactly one title, except for the root item element which must have no title.
3. Resources must have one of the two types *webcontent* and *assessment*.
4. The manifest must have metadata which conform to the IEEE LOM metadata specification.
5. Assessment Resources must contain a single File object which references the QTI XML file. This file must conform to the IMS QTI 1.2.1 specification.
6. If an item references a resource, then there cannot be subitems.
7. The file *imsmanifest.xml* must exist at the root of the package.
8. Every file, except the file *imsmanifest.xml*, must be referenced in a resource and every file referenced in a resource must exist..

To start capturing this profile we launch SchemaProf with a doubleclick on *IMSSchemaProf-2.0.jar*. We open the IMS Content Packaging specification schema from the *SchemaProfTutorial\ BaseSchema* folder as base schema for a new profile. With a click on the top node of the profile we get the possibility to add global annotations. Clicking the *New* button in the right part of the window opens the dialog for entering a new annotation. We chose “*name*” as type of the annotation and “*en*” for the language and enter a convenient name for our profile, say “*SchemaProf Tutorial Profile*”.

Next we define the profile namespace with the *Namespace Manager*. Selecting this from the Manager menu we see under the heading “*Target Namespace of the profile*” that SchemaProf by default preserves the namespace of the base schema for the profile. According to our Requirement 1 we override this with “*http://www.imsglobal.org/schemaprof/tutorial*”. That completes capturing requirement 1. For the convenience of referring to elements in our profile we also define the prefix *tut* as an abbreviation for this namespace. Then we can *Close* the Namespace manager.

After these preparatory steps we start adding to our profile modifications of the base schema which reflect the changes described in the requirements above. This is done by modifying the type definitions provided by the schema. These definitions can be accessed after clicking the “*Complex Types*” tab in the left part of the SchemaProf window.

The second requirement poses a challenge since there is only one definition of the type of the *item* element which has an optional title, but requirement 2 comes in two variants, depending on the context in which the element is used. Let us for the moment defer handling the exceptional case of the root *item*. We select the *title* element which appears when the *sequence* below the *item.Type* node is opened by left-clicking it. The right panel shows on top the original definition – *minOccurs=0*, i.e. the *title* element is optional. We change this by clicking “*New*” to start a new modification and set *minOccurs* to *1* – we have made *title* unique and mandatory in *item*. You may note that SchemaProf would not allow to increase the *maxOccurs* value. This is because we are working in the default *restrictive* mode of SchemaProf which does not allow for modifications that would break the original specification. Using the *New* button we add a comment that explains this modification.

Note that the path towards the *title* element in the left panel has become **red** and the *title* element name is in *italic*. This indicates a place where our profile contains a modification with respect to the base schema. Please close now the unfolded *item.Type* node in the hierarchy.

In order to handle the exceptional case of the root *item* element we have to access the definition of *item.Type* used in the context of the *organization* element as this element is supposed to keep the hierarchy of *items*. We can unfold *organization.Type* in order to see the *item.Type* definition on which it relies. The *item* element definition is the second node within the *sequence*. (The first *title* element in the sequence is the title of the *organization* but Requirement 2 concerns only titles of *item* elements. Therefore we don’t have to change this

first title.) One further click on the *item* element gives access to its definition in *item.Type*. If there is too little space on your screen for the hierarchy of definitions you may move the border between the left and right part of the SchemaProf window with the mouse.

We unfold *item.Type* further until we reach its *title* element. Note that *item.Type* and everything below it is greyed out. It can be inspected but cannot be modified. This is because it is not defined within this local context but is instead included by reference to the global type definition for the *item* element. When you click on the greyed *title* element whose multiplicity we want to set to 0, a new button “Activate” appears in the lower part of the screen (alternatively you may right-click on the *title*). When you now use this button to activate *title*, the grey color disappears and *title* is now available for modification – internally SchemaProf has created a modifiable copy of the original *item.Type* and has assigned this new type to the root item. Now we can set *maxOccurs* for this copy of the *title* element to 0. This declares that *title* must not be used for *items* which are directly below the *organization* element. Capturing of Requirement 2 is complete.

Requirement 3 allows only two values for the *type* attribute of a *resource* element. We select this attribute in the definition of *resource.Type* and click *New* to start our modification. As we need a new simple type with just two elements we start a new modification of this definition and select a *Restriction* of the existing simple type *string*. From the available modification facettes, we select the “Enumeration” tab. We enter our two permitted values *webcontent* and *assessment*. Our newly defined simple type gets a name; say “*PermittedResources*”. In a next step we check that the use of the *type* attribute is *required*.

This is a good time to go to the Manager menu and open the Type Manager. You will find your newly defined type listed there, ready for further modifications in case you want to add another resource type.

Now let us look at Requirement 4: The manifest must have metadata which conform to the IEEE LOM metadata specification.

Our IMS Content Packaging base specification does not prescribe the use of a particular metadata specification or profile. It just provides an extension point in the type definition of *ManifestMetadata*. This extension point is a reference to *grpStrictAny*. Since we want to modify only the use of this group within *ManifestMetadata*, we have to activate the greyed out *any* element below *ManifestMetadata.Type* first as described above. After this activation we can define a new modification of this local copy of this *any* wildcard element. As a first step we must remove the original definition of allowing all *other* namespaces since we want to be more specific. Since we require the use of a particular specification, and not only of a particular namespace, we select “Profile” from the drop down box. Then, using the dropdown list on the right, we browse for our pre-build IEEE LOM profile which we request to be used in this place. It is in the file *SchemaProfTutorial\AuxiliaryProfiles\ieeeLom.zip*. To make sure that the metadata extension is in fact evaluated we set “*process content*” to “*strict*”.

Requirement 5 is complex and we deal with its more basic part first: Assessment Resources must contain a single File object.

The key point to note is that this requirement applies to *assessment* resources only, i.e. to resources whose *type* attribute has the value “*assessment*”. Unlike the case of the root *item* element, *resource* is not referenced in a context which determines whether Requirement 5 applies. Therefore we cannot use activation to restrict our intended modification to assessment resources only. But we can use conditions instead!

The rest is easy: We open *resource.Type* and select the *file* element in the *sequence*. Then we set its *minOccurs* and *maxOccurs* both to 1. Now we have to capture that this modification is only applied under the condition that the *resource* is of type “*assessment*”. Note that there is a button which takes you right to the condition manager. We use it to define a new condition

containing the XPath expression `../@type = 'assessment'`. We select this as a condition for our modification and close the condition manager. Now we can select this condition from the drop down list, add an explanatory comment and finish. This completes capturing the first part of Requirement 5. The second part shall be handled by additional constraints below.

Requirement 6 says: If an item references a resource, then there cannot be subitems.

An *item* references a *resource* if it has an *identifierRef* attribute. We select the *identifierRef* attribute of *item.Type*. This makes sure that the following requirement shall be applied only if this attribute exists. In this case we can express our requirement “there cannot be subitems” as the XPath assertion `count(..//tut:item) = 0`. Assertions are entered through an additional tab “Assertion” of the right part of the SchemaProf window. Note that *tut* is the prefix that we have defined for our profile namespace earlier on in the namespace manager.

Requirement 7 “The file *imsmanifest.xml* must exist at the root of the package” takes us to additional constraints. Additional constraints require specific test systems. They are defined through the right tab “Constraints” in the left part of the SchemaProf window. After clicking on this tab there is a new button “New Constraint...” available at the bottom of the window. We go for the definition of a *file constraint* with a *user defined* value “imsmanifest.xml” which is entered as the URI.

!!!

The second part of Requirement 5 “Assessment Resources must contain a single File object which references the QTI XML file. This file must conform to the IMS QTI 1.2.1 specification.” requires a dynamic additional constraint, i.e. the constraint, in particular the location of the QTI XML file, depends on values to be found in the instance document’s resource elements: the *href* attribute of an *assessment resource* must point to a file which is valid w.r.t. the QTI specification.

First we select the definition of a file constraint. But this time we don’t use “*user defined value*” but “*value stored in an attribute*”. The respective *href* attribute is determined in two steps. First we select from the *complexType* group *resource.Type* as the relevant context and then the *href* attribute of this type as the attribute which holds the location of the file to be tested. After ticking that the file location is relative leave the file name format at “*any*” and we can finish. But this test should be done only for assessment resources. Therefore we let this test depend on the already defined condition that the *resource* element, which defines the context of the condition, has a *type* attribute with value “*assessment*”. This condition can be selected from the *condition* dropdown list.

So far we have only required that the referenced file exists. We still need to require that it is a QTI file.

To do this we select the constraint that we just defined and click on the right side “*New Definition*”. We want to request a validation against another profile and so we select *FileValidation* as type of the new definition. We, the user, want to determine the profile to be used for validation so we select “*user defined value*”. Then we can browse for the QTI profile to be used for validation. We select the QTI profile *ieeLom.zip* from the *AuxiliaryProfiles* folder.. This completes capturing Requirement 5.

The last step in the creation of a profile is the configuration of the test system using the *Test Configuration Manager*. After saving the profile the *Test Configuration Manager* takes you through a number of steps. The first of these steps allows disabling some of the following test types:

- Reference tests – i.e. testing that all files in a package are referenced as in our Requirement 8

- Constraints Validation – this is only needed if the profile contains additional constraints
- Schema Validation – this shall be required in most cases
- Schematron Validation – this is always required if conditions or assertions are used

In the following steps it can be determined for which files in a package which tests are executed. This is done by giving regular expressions for the file name (for example `.*xml` will select all xml files) and for the namespace (we only want files of our particular namespace <http://www.imsglobal.org/schemaprof/tutorial> tested, so this can be taken as a regular expression for the namespace). In the next steps it can be selected which schema is to be used to define which namespace. In most cases SchemaProf will select the correct schema, schematron or constraint file automatically but in some cases, when a specification defines the same namespace in several schemas, your help may be needed. Make sure that you select one of the offered schema files with *localised* in their name since these will be the ones that have been modified according to your profile.

In this tutorial SchemaProf can determine the correct schemas for all namespaces, except for <http://ltsc.ieee.org/xsd/LOM> for which many defining files exist. We select the localized version of the IEEE LOM root schema, i.e. *domainProfile_0/lomLoose_localised.xsd*.

Then we are taken to configuring the reference tests. We only want references in the file *imsmanifest.xml* be tested. So we enter *imsmanifest.xml* as regular expression for matching files. We can also enter <http://www.imsglobal.org/schemaprof/tutorial> as namespace of files to be tested. Now a little more XPath is needed to describe the location of the references in the manifest file. It is the *href* attribute of the *file* element, so we enter `//tut:file/@href` (the two initial slashes indicate that all *file* elements in the manifest file should be considered).

Don't forget to enter *imsmanifest.xml* under "Exclusion filter" since this file must not be referenced.

A final page summarizes our settings and clicking "Finish" completes the capturing of our requirements. The only thing that remains to be done is saving the profile.

4. Using Profiles for Conformance Testing

Your SchemaProf generated application profile or domain profile can be used to create a conformance test system to test data for compliance with your profile. Test systems are not part of the SchemaProf software. They come with their own user manual.

4.1. IMS Approved Conformance Test Systems

For profiles based on IMS specifications IMS Global Learning offers a service to provide such specific conformance test systems. To use this service you must register your profile at IMS. Please see <http://www.imsglobal.org/sdan/> for the IMS profile registration procedures.

In order to obtain a test system for profiles based on other specifications please contact schemaprof@uni-koblenz.de. The following figure shows the main screen of a test system based on a SchemaProf profile.

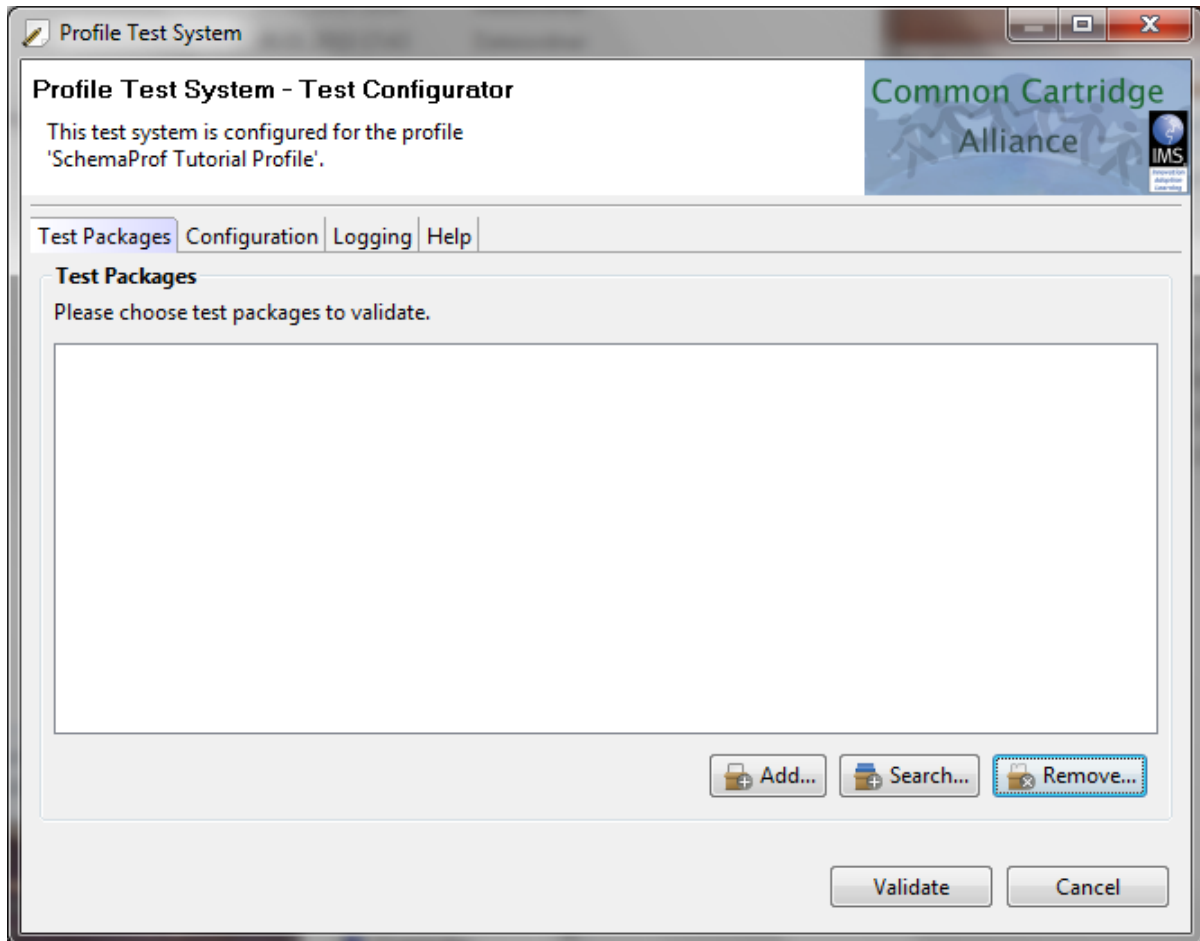



Figure 36 A SchemaProf based Test System

The following figure gives an impression of test reports from such test systems.



Schematron Report

| Document | Constraints file | Number of errors | Link |
|-----------------|------------------------------------|------------------|------------------------------------|
| imsmanifest.xml | imscp_v1p2_constraintsDocument.scm | 3 | To the results ... |

Detailed schematron validation report

Document: [imsmanifest.xml](#)

- ° **Assertions:**
 - **Pattern: pattern_1**

| | |
|-----------|--|
| Context | tut:item/@identifierref tut:item/@identifierref |
| Condition | (count(..tut:item) = 0) |
| Message | Assertion failed for pattern_1. Requirement 6 says: If an item references a resource, then there cannot be subitems. |
| Line | XPath |
| 36 | /manifest/organizations[1]/organization[1]item[1]item[2]/@identifierref |
 - **Pattern: pattern_2**

| | |
|-----------|--|
| Context | tut:resource/tut:file |
| Condition | not (../@type = 'assessment') or (count(..tut:file) >= 1 and count(..tut:file) <= 1) |
| Message | Condition name: AssessmentResource. Error: Cardinality validation failed for pattern_2 |
| Line | XPath |
| 55 | /manifest/resources[1]resource[3]file[1] |
| 57 | /manifest/resources[1]resource[3]file[2] |

Fertig Computer | Geschützter Modus: Inaktiv 100%

Figure 37 A Test System Report

4.2. Other Test Systems

In order to obtain test systems for profiles of specifications from organizations other than the IMS Global Learning Consortium please contact the Knowledge Media Institute of the University in Koblenz at iwm@uni-koblenz.de.

Even without a dedicated test system, the profiles created by IMS SchemaProf can be used for testing, though not with full comfort and power. In particular these will not test for additional constraints and for conditional modifications of simple XML types. They will also not do reference testing and testing of many test items for the constituents of a domain profile in one test run as do dedicated test systems.

To do some conformance tests nevertheless, you will need an XML validator and a Schematron validator. Unzip the profile that has been saved by SchemaProf. Locate the subdirectory named *derived_schema*. In this subdirectory you will find a new XML schema which has the same name as the base schema of the profile with *_localised* appended to the file name root. This schema can be used with an XML validator to test for those parts of the profile that can be expressed in an XML Schema. In the same directory you may find a file with the file name extension *.smt*. This file can be used by a Schematron validator to do further tests.

The root directory of the IMS SchemaProf generated domain profile will also contain a number of directories whose names start with *profile_*. These contain the auxiliary profiles which are used in the domain profile aside of the main profile. Tests for those auxiliary profiles can be set up in the same way.

5. Known Issues and Workarounds

| Issue | Recommendation |
|---|---|
| Vocabulary constraints can be defined but cannot be handled by the test system. | Model vocabularies only as enumeration types. |
| SchemaProf makes only basic plausibility checks on XPath expressions entered by the user. Wrong XPath expressions and missing namespace prefixes are major sources of errors in profiles. | Be careful! |
| All elements in XPath expressions must have a non-empty namespace prefix. | Respect this as good practice, use the SchemaProf namespace manager to define prefixes. |
| Auxiliary Profiles cannot refer to other auxiliary profiles. Only the main profile can refer to auxiliary profiles. | If an auxiliary profile uses another, secondary, auxiliary profile the derived schema of the auxiliary profile should be taken as the base schema of a new profile, thus integrating the auxiliary profile and the secondary profile. Conditional modifications and assertions concerning elements defined in the secondary profile should be defined in this new auxiliary profile. |
| Different application profiles in the same domain profile must utilize | Respect this restriction as good practice. |

different top level namespaces.

If an instance document contains at an extension point of the base schema more than one element conforming to another profile than only the first of these elements will be validated against the profile.

Collect elements to be used at an extension point within one single element

References

- “IMS Application Profile Guidelines”. <http://www.imsglobal.org/ap/index.html>.
- “IMS SchemaProf”. <http://www.imsglobal.org/profile/>
- “IMS Specification Adoption Note 11: Profile Definition, Registration and Maintenance Procedures”. <http://www.imsglobal.org/sdan>
- “IMS TELCERT Information Page ”. <http://www.imsglobal.org/telcert.html>
- “XML Schema Part 1: Structures”. W3C Recommendation, 2 May 2001, <http://www.w3.org/TR/xmlschema-1/>.
- “XML Schema Part 2: Datatypes”. W3C Recommendation, 2 May 2001, <http://www.w3.org/TR/xmlschema-2/>.
- “XML Path Language (XPath) Version 1.0”. W3C Recommendation, 16 November 1999, <http://www.w3.org/TR/xpath>.

About this Document

| | |
|---------------------|---|
| Title | IMS Schema Profiling Tool – User Manual |
| Editor | Hans-Christian Herrmann, Ingo Dahn University Koblenz-Landau |
| Version | 2.0 |
| Version Date | 30 November 2009 |
| Summary | This document describes how to use the IMS Approved Schema Profiling Tool to profile the XML schema bindings of IMS specifications. |
| Location | http://www.imsglobal.org/profile |

Appendix A – SchemaProf Software License

IMS Global Learning Consortium, Inc. Schema Prof Tool END-USER SOFTWARE LICENSE AGREEMENT

The accompanying executable code version of IMS Schema Prof and related documentation (the "Product") is made available to you under the terms of this IMS SCHEMA PROF END-USER SOFTWARE LICENSE AGREEMENT (THE "AGREEMENT"). BY INSTALLING OR USING THE IMS SCHEMA PROF TOOL, YOU ARE CONSENTING TO BE BOUND BY THE AGREEMENT. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT, DO NOT INSTALL OR USE ANY PART OF THE IMS SCHEMA PROF TOOL

INSTALLATION AND USE OF THIRD-PARTY COMPONENTS MAY BE GOVERNED BY ADDITIONAL LICENSE AGREEMENTS.

1. **LICENSE GRANT.** IMS Global Learning Consortium, Inc. grants you a non-exclusive license to use the executable code version of the Product. This Agreement will also govern any software upgrades provided by IMS GLC that replace and/or supplement the original Product, unless such upgrades are accompanied by a separate license, in which case the terms of that license will govern.
2. **TERMINATION.** If you breach this Agreement your right to use the Product will terminate immediately and without notice, but all provisions of this Agreement except the License Grant (Paragraph 1) will survive termination and continue in effect. Upon termination, you must destroy all copies of the Product.
3. **PROPRIETARY RIGHTS.** You may not remove or alter any trademark, logo, copyright or other proprietary notice in or on the Product. This license does not grant you any right to use the trademarks, service marks or logos of IMS Global Learning Consortium, Inc. or its licensors.
4. **DISCLAIMER OF WARRANTY.** THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

5. EXPORT CONTROLS. This license is subject to all applicable export restrictions. You must comply with all export and import laws and restrictions and regulations of any United States or foreign agency or authority relating to the Product and its use.

6. MISCELLANEOUS. (a) This Agreement constitutes the entire agreement between IMS Global Learning Consortium, Inc. and you concerning the subject matter hereof, and it may only be modified by a written amendment signed by an authorized executive of IMS. (b) Except to the extent applicable law, if any, provides otherwise, this Agreement will be governed by the laws of the state of Florida, U.S.A., excluding its conflict of law provisions. (c) This Agreement will not be governed by the United Nations Convention on Contracts for the International Sale of Goods. (d) If any part of this Agreement is held invalid or unenforceable, that part will be construed to reflect the parties' original intent, and the remaining portions will remain in full force and effect. (e) A waiver by either party of any term or condition of this Agreement or any breach thereof, in any one instance, will not waive such term or condition or any subsequent breach thereof. (f) Except as required by law, the controlling language of this Agreement is English. (g) You may assign your rights under this Agreement to any party that consents to, and agrees to be bound by, its terms; IMS Global Learning Consortium, Inc. may assign its rights under this Agreement without condition. (h) This Agreement will be binding upon and will inure to the benefit of the parties, their successors and permitted assigns.

COPYRIGHT

Copyright notice applicable to IMS Global Learning Consortium, Inc.'s IMS Schema Prof Software and Documentation.

©Copyright 2004-2006 Universität Koblenz-Landau, Institut für Wissensmedien (IWM)

- This product includes software developed by the Eclipse Foundation
(<http://www.eclipse.org>).

- This product includes software developed by the Apache Software Foundation
(<http://www.apache.org>).

Copyright (c) 2006 by IMS Global Learning Consortium, Inc.. All Rights Reserved.