



Enterprise Architect

User Guide Series

The Schema Composer

Author: Sparx Systems

Date: 6/06/2016

Version: 1.0

CREATED WITH  ENTERPRISE
ARCHITECT

Table of Contents

The Schema Composer	3
Schema Composer Profiles	6
Create a Schema Profile	8
Schema Compositions	11
Schema Analysis	13
Class Diagrams	14
UPCC Schema Guide	16
CIM Schema Guide	17
Model compositions	19
UML Profile for Core Components	21
Generate Schema	24
Generate Schema File	26
Generate a Model Subset (Transform)	28
Schema Importer	31
Available Frameworks	33
Install a Core Framework	36
Schema Composer Automation Integration	39
Schema Composer Addin Integration	40
Schema Composer Scripting Integration	41
MDG Technologies - UML Profile Extensions	45

The Schema Composer

The **Schema Composer** is a versatile tool for quickly and easily defining a variety of formal schema from a model. Due to the unique nature of the Schema Composer, it is not necessary to use a profile or stereotyped elements when building the definition of an **XSD** (or other) document. This greatly enhances the re-usability of the underlying model and helps alleviate the complexity that arises when dealing directly with XSD or other element types and restrictions.

Many industries have worked hard over the last decade to define shared meta-models specific to their industry, and it is these models which now form the basis for contractual information sharing across organizations and across geographic borders. A typical usage scenario of the Schema Composer is in the creation of message definitions (schema) to exchange information between organizations, ensuring that such messages comply with the underlying meta-model that has been adopted by the involved parties.

When information is shared between organizations, it is frequently the case that only a subset of the full meta-model is required, but it is essential that what is shared conforms precisely to the agreed meta-model. In this case the Schema Composer is the perfect tool for deriving contractual schema based on sub-sets and restricted data sets that take a 'slice' through the meta-model as a whole.

The Schema Composer avoids the common 'pain points' of working with XSD and other schema languages directly:

- There is no need to create a relatively complex XSD model composed of specific XSD elements, in addition to your 'normal' business and data models, to define the required data, its associations and references, and any restrictions or conditions
- You do not need to understand how to use the XSD elements and apply the XSD naming rules and conventions to correctly construct such models; formatting and naming rules as specified by the supported standards are automatically taken care of

The Schema Composer greatly simplifies the process of creating standards compliant schema in a re-usable and accessible manner. In this illustration, you can see how a simple Class diagram is used as the source for the Schema Composer to generate XML Schema.

Schema Composer

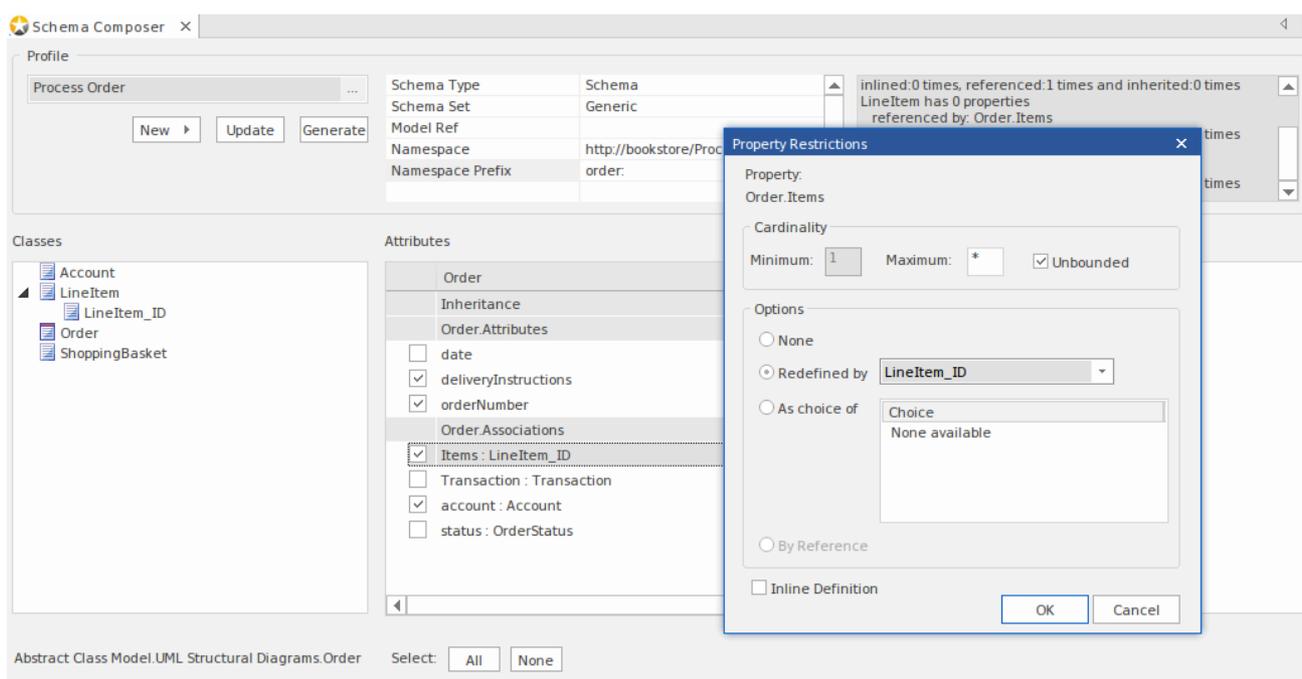


Figure shows a Schema Composition for the Process Order domain in the Example model.

Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer
Menu	Tools Schema Composer

Benefits

The **Schema Composer**:

- Operates on a Class model rather than an XML schema profile
- Relieves you of the XSD-specific design and schema generation decisions, whilst still ensuring consistency across the profile
- Can operate on a generic Class model to provide generic **XSD** documents
- Is most useful when operating on industry standard Class models that have specific domain based meaning
- In most circumstances operates on a full model from which a subset of properties from selected Classes are drawn to build specific messages, to communicate only what is necessary for the information to send or request
- For standards such as NIEM, will generate a new sub-model as part of a broader NIEM compliant schema definition

Standards that the Schema Composer currently supports include:

- The Common Information Model (**CIM**)
- National Information Exchange Modeling (NIEM)
- United Nations Center for Trade Facilitation and Electronic Business (UN/CEFACT) Modeling Methodology (UMM), specifically the Naming and Design Rules (NDR) 2.1 and 3.0
- Universal Business Language (UBL), specifically the Naming and Design Rules (NDR) 3.0

The Schema Composer also helps you to build a definition of the same message using different formats such as:

- XSD
- RDFS
- JSON

In addition the Schema Composer

- Supports formats implemented using a custom **Add-In** that takes advantage of the Schema Composer automation interface
- Has built-in support for various serialization formats and styles used by different industry models

Notes

The **Schema Composer** is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

Schema Composer Profiles

Schema Composer profiles are the configuration files that describe the elements and restrictions that will make up a particular schema or transform. Profiles are generally tied to a particular technology and the interpretation of the material within the Profile and the nature of the output schema or subset is dependent on the technology specific generator used. While Enterprise Architect supports a number of technologies 'out of the box' (and more are planned), it is also possible to customize the process by using the extensive automation interface in the Schema Composer and the elements and restrictions contained in a Profile.

Schema Profiles

A **Schema Composer** profile comes in two forms. Each form fulfills a particular systems requirement: Schema generation (xsd, rdfs, json) and sub model creation. When you create a profile in the Schema Composer you choose which form to use based on your needs. A single profile in the Schema Composer can be used to either compose schema, *in its common forms*, or create a UML sub-model from a core model.

Profile Types

Type	Description
Model Transform	A profile of this type is used to generate a sub model from a core model.
Schema	A profile of this type is used to generate schema; typically XSD schema representing messages, but also other formats such as json object notation and resource descriptor formats.

Schema Composition Methodologies

National Information Exchange Model (NIEM)

Enterprise Architect provides a NIEM framework and **Schema Composer** for generation of sub model and XML schemas.

Common Information Model (CIM)

Enterprise Architect **Schema Composer** supports provides the **CIM** standard out of the box, for composition of CIM compliant schema.

Universal Business Language (UBL)

Enterprise Architect provides a Universal Business Language framework, and the **Schema Composer** which provides the UBL standard for schema generation.

Core Component Technical Specification (CCTS) UN/CEFACT

Enterprise Architect provides a UML Profile for Core Components framework and **Schema Composer** for the generation of business components from **UPCC** core components.

Generic

Where a standard does not meet your requirements, the generic option provides a simpler choice for quick schema composition from your UML model. Typically you will model your own data library using UML Classes with attributes, associations, Aggregation and Inheritance. You can then use this model as the input to the **Schema Composer**.

EA Script engine

Enterprise Architect provides a scripting engine that supports Javascript, VBScript and JScript languages. The scripting engine is also integrated with the **Schema Composer**. When generating schema, either for a particular standard or generic scheme, a script can be employed to perform the operation on its own or as a supplement to the options provided by the standard.

EA Addin

Enterprise Architect provides **Add-In** integration with the **Schema Composer**. An Add-In can participate in the generation of the sub model or schema by registering its interest with Enterprise Architect. The Add-In can provide options and alternatives to be listed in the 'Schema Generation' dialog, and will be invoked should its options be chosen. The Add-In can access the content of the profile using Schema Composer automation interfaces.

Create a Schema Profile

A schema profile identifies the name, technology and content of the schema as a precursor to defining how the schema is generated. You can create and edit as many schema profiles as you need. Schema profiles are typically bound to a single technology and will either map to a generated schema or a sub-setting transform.

Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer
Menu	Tools Schema Composer

Creating a new Profile

If you are creating a schema for a particular technology, start by opening a model that has the requisite meta-model loaded. Sparx Systems make a number of meta-models available when using the **Model Wizard** and/or from the Sparx RAS/Cloud services. If you are using a customized or domain specific model, make sure that model is loaded. Once you have a suitable model and have opened the **Schema Composer**, you can follow these steps to build a new Profile.

Field/Option/Button	Action
Profile (name)	This field is initially blank. When you have created a schema, the name displays in this field.
New	<p>Click on this button to start creating the schema profile.</p> <p>The 'New Message' dialog displays.</p> <ul style="list-style-type: none"> 'Schema Set' - Click on the drop-down arrow and select the standard that you have created your model for; if you have not adopted a standard, select the 'Generic' option 'Namespace' - If required, type in the XML Namespace that this schema is associated with; if you have selected the CIM standard, the 'Namespace' field takes an automatic value 'Save In' - Click on the appropriate radio button for the format in which to save the schema OK button - Click on this to create the profile and close the dialog
Model Ref	Depending on the standard you have selected for the Schema Set, this field takes an automatic value or remains blank. If blank, you can type in a reference name if required.
<other options>	Depending on the standard you have selected for the Schema Set, other fields might display. Either leave blank, type in the appropriate value, or click on the  button and select the appropriate library Package.
Update	Click on this button to save the schema details you have defined.

Define Schema Content

These steps walk you through the basic procedure of adding elements to a Schema profile and show how you might customize the content of the profile to create the required schema output.

Step	Action
1	<p>Drag the required Class elements from the Project Browser into the 'Classes' panel. As you add an element:</p> <ul style="list-style-type: none"> • Its model structure path is shown underneath the 'Classes' panel • Its attributes and Association connectors are listed in alphabetical order in the 'Attributes' panel, with a blank check box against each one; the Associations entries are named according to the role name on the connector. (Aggregations form part of list.) • The schema name is displayed in the 'Schema' panel <p>If necessary, you can list the elements and attributes in reverse alphabetical order (and change the sequence back again) by clicking on the column title.</p>
2	<p>The Schema Composer allows you to leverage the relationships from your model. When you begin composing a class, you will see its ancestry listed in the inheritance section. None of the ancestors are selected to begin with. This is the default composition mode which we might call aggregation. The Composer offers flexibility in dealing with inheritance. For example you can choose to aggregate selected attributes from the class and its parent, while choosing to inherit the grandparent. However when you choose to use inheritance, you choose to inherit the restricted form of that type as well. When an ancestor is selected in this list, the generated XML schema would show an extension element identifying this ancestor. Only one ancestor can be selected.</p>
3	<p>Click on a Class in the 'Classes' list and, in the 'Attributes' list, select the checkbox against each attribute and Association from that Class to include in the schema.</p> <p>The element and its selected attributes and Associations are added to the 'Schema' panel.</p> <p>You can select all checkboxes for an element at once by clicking on the All button, and clear all selected checkboxes by clicking on the None button.</p> <p>If you select the checkbox for an Association that links to an element you have not added to the 'Classes' list, that element is automatically added and reported in the status panel in the top right of the screen.</p>
4	<p>In the 'Classes' list, right-click on a Class and use the context menu to set element properties as required. Use this tool to:</p> <ul style="list-style-type: none"> • Set element as root - root elements typically form the body of the top level element representing the message / profile. • Edit root element cardinality - if you have selected 'Set element as root', the 'Restrictions' dialog displays for the Class; set the minimum and maximum number of instances of the root element that can exist in the schema. • Remove root attribute from element - if you previously selected 'Set element as root', reset the selected element to 'normal' and not to a root • Remove the selected element - delete the selected element from the schema • Find in Project Browser - locate and highlight the element in the Project Browser <p>You can double-click on the element to display its 'Properties' dialog, where you can define a</p>

	wider range of element properties as necessary.
5	<p>In the 'Attributes' list, right-click on a selected property and use the context menu to add, edit or remove a property restriction. use this feature to:</p> <ul style="list-style-type: none">• Modify the property cardinality• Redefine the type of the property• Enable and limit the choices available for this property.• Mark a property to be emitted as an inline element definition• Mark a property to be emitted 'By Reference' <p>You can double-click on the attribute or Association to display its 'Properties' dialog, where you can define a wider range of properties as necessary.</p>
6	<p>Click on the Update button to validate and save your schema profile.</p> <p>If there are any problems with the profile, they are identified in the status panel in the top right of the screen.</p>
7	You can now go on to generate schema based on your profile.

Notes

- The process of creating and generating schema for NIEM has additional notes in the *MDG Technology for NIEM* Help topic
- The **Schema Composer** is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

Schema Compositions

A schema composition refers to the set of elements and associated restrictions and subsets that make up a Profile configuration. Generally schema compositions are used to define material used to build an output schema, for example an **XSD** file. Model compositions in contrast are used to configure the material used as the basis of a subset "transform" - for example when creating a NIEM model subset.

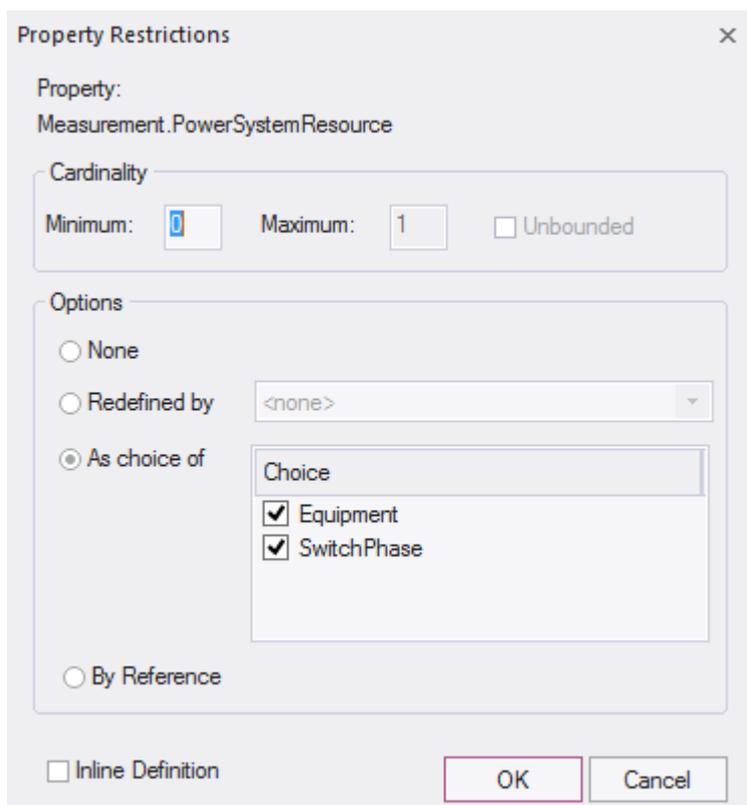
Define schema content

Content is added to the schema by dropping Classes from the model on to the **Schema Composer** Class window and choosing which properties to include. The Schema Composer helps with the composition by adding any referenced types automatically to the schema. The process of composing a schema is like this:

1. Create a Scheme Composer Schema Profile.
2. Create elements from a Class model.
3. Include/exclude properties.
4. Apply any restrictions.
5. Review use of inheritance
6. Generate schema formats.

Schema Restrictions

The **Schema Composer** allows restrictions to be applied to elements and properties for a particular message requirement.



Property Restrictions

Property:
Measurement.PowerSystemResource

Cardinality

Minimum: Maximum: Unbounded

Options

None

Redefined by

As choice of

Choice

- Equipment
- SwitchPhase

By Reference

Inline Definition

OK Cancel

Aggregation

A class listed in a the schema can aggregate the properties of any generalizations that exist in the schema. When a class is selected in the schema that generalizes another class also listed in the schema, the properties of both will be listed in the attributes window. By selecting one or more properties of the generalization these properties will be aggregated on the specialization.

In simple terms, a type can be created that describes an element in simpler terms based on selective properties.

Inheritance

A schema element can be represented according to the entire class model inheritance structure or none at all. When aspects of the inherited model are required and selected in the schema composer, the generalizations will be automatically added to the schema.

Redefinition

When an element is redefined within the Schema Composer it is given a new name. It can then be restricted in different ways to provide the required description of an instance. A 'PaymentMethod' enumeration could be redefined as 'CardPayment' that omits the value 'Cash'. A redefined type is a top level schema element. It has no corollary in the model.

Cardinality

The cardinality of an property can be further restricted from its model counterpart, but it cannot be less restrictive. The cardinality can be changed for any root element class and any class property.

Choice

Where a property type has specializations that are listed in the schema, those specializations can be specified as choice elements in the generated schema. To do this select the property and choose the restriction content menu option.

Any specialization that has been added to the schema will be available as a choice.

Style

This section relates to the style of **XSD** generated by the **Schema Composer**. Each standard has its own rules in relation to the style of xsd produced. UN/CEFACT NDR for example specifies a modular schema with two schemas being emitted for a single message. Enterprise Architect provides flexibility with these schemes and permitting different styles of XML schema to be produced.

Style	Description
Inline elements	Properties can be emitted as inline local definitions, by setting the option in the property restriction page.
Type Redefinition	Allows a type to be defined based on another type. The new type can be re-used within the schema. A restricted payment enumeration for instance, perhaps omitting certain credit cards.
Element re-use	Types are emitted as top level elements by default, allowing re-use of the schema.
Unified schema	A unified schema is emitted that encapsulates and completely defines the message described by the schema.

Schema Analysis

Analysis on the go

The **Schema Composer** performs analysis of each type as it is added to the schema, and whenever the Class is selected. The **System Output** window will show how many, if any, references exist for the type, the number of times it is inherited and other helpful information. This illustration shows a message detailing the elements that are referencing the selected Class.

```
referenced by: Period.DateOfIssue_Start
referenced by: Person.Birth
inlined:0 times, referenced:3 times and inherited:0 times
MyCode has 10 properties
referenced by: Address.Postcode
referenced by: Person.Gender
inlined:0 times, referenced:2 times and inherited:0 times
```

Validation on the go

The **Schema Composer** performs specific validation for a technology should one be assigned. This image shows warnings about missing **Tagged Values** for Classes in a schema built on the UN/CEFACT Core Components standard.

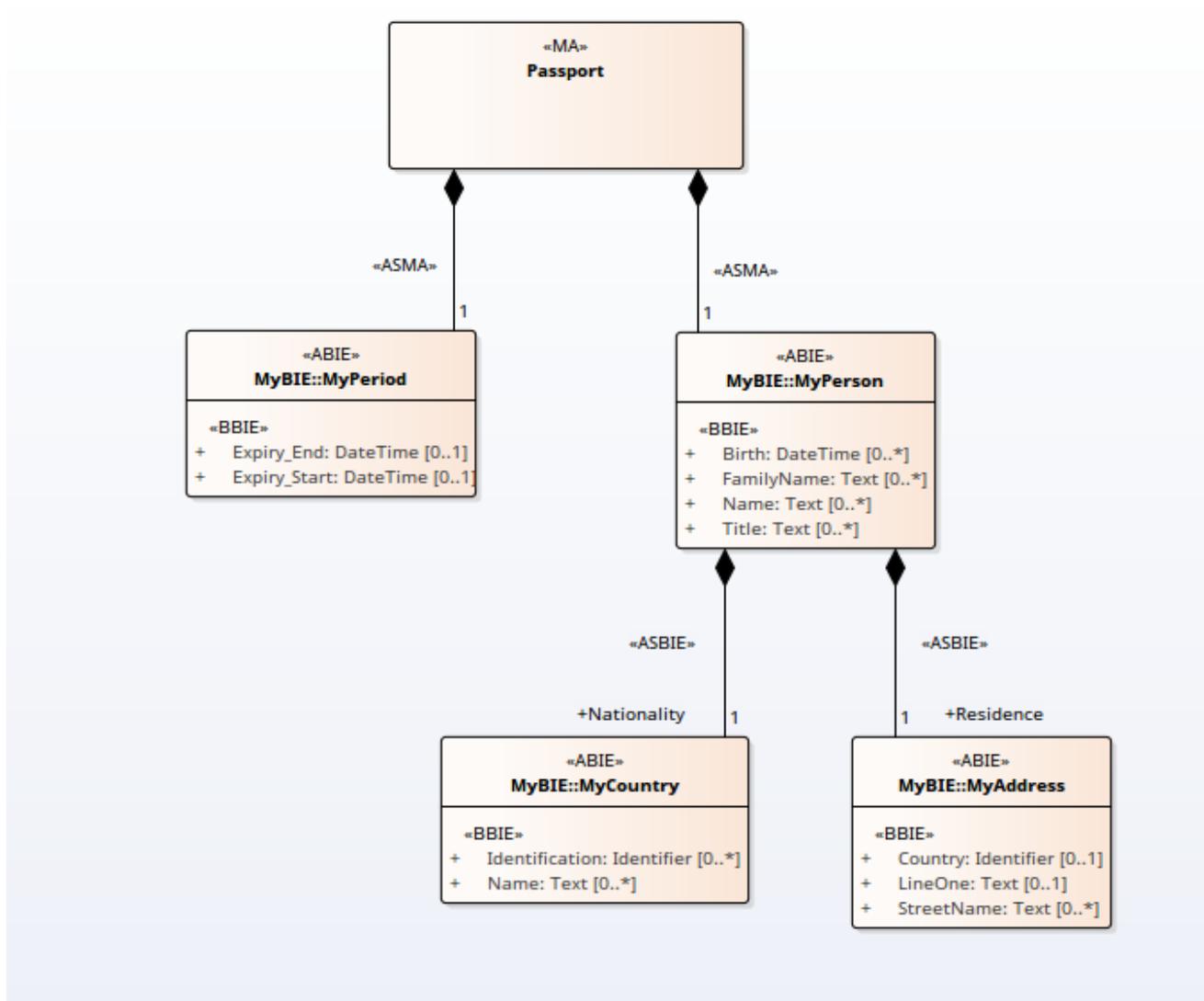
```
Warning: dataTypeQualifierTermName facet missing for class MyIdentifier
Warning: dataTypeQualifierTermName facet missing for class MyMeasure
Warning: dataTypeQualifierTermName facet missing for class MyText
Warning: dataTypeQualifierTermName facet missing for class MyCode
Warning: dataTypeQualifierTermName facet missing for class MyIdentifier
Warning: dataTypeQualifierTermName facet missing for class MyDateTime
Warning: dataTypeQualifierTermName facet missing for class MyMeasure
```

Class Diagrams

The **Schema Composer** also supports the creation of simple **XSD** and other formats from generic UML Classes. This is particularly useful when there is a need to export a class definition in a generic manner for consumption by a script or web based tool, for example.

Generating schema from Class diagram

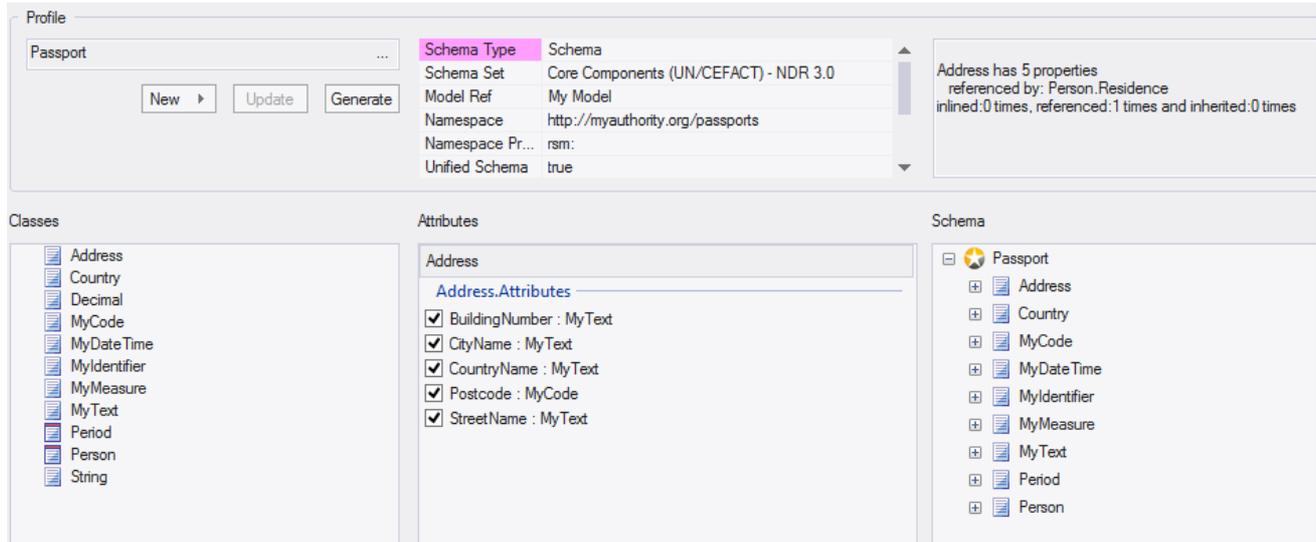
Users who prefer to use a modeling approach in composition can also use the **Schema Composer** for the generation of their chosen format(s). Any Class diagram can be loaded into the Schema Composer. This image illustrates a message composed using the UML Profile for Core Components, but it is not necessary for the message to be modeled according to a particular UML profile.



Loading the message into the Composer

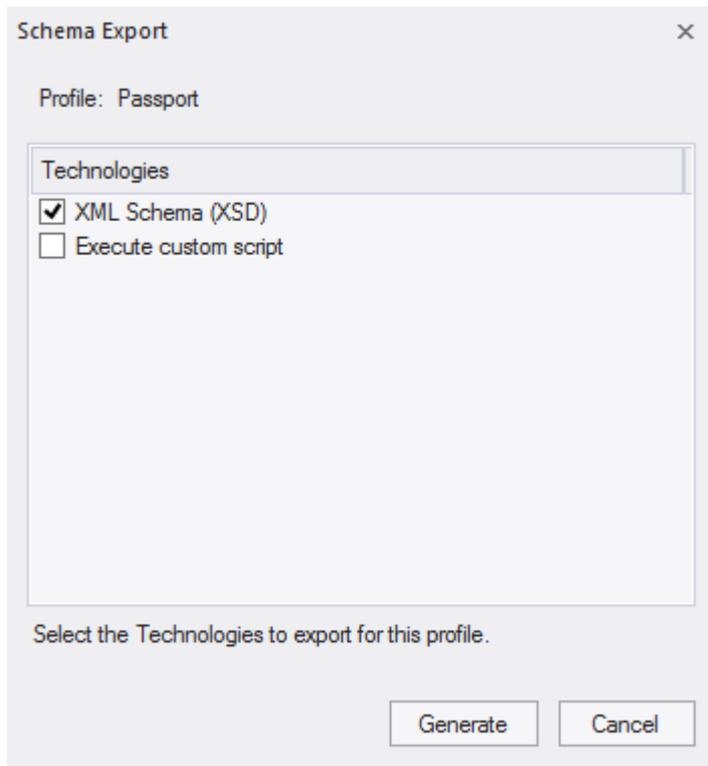
The message is loaded into the Composer by selecting a Class on the diagram that represents the message and using its context menu to present the diagram as a schema in the **Schema Composer**. The selected

Class will become the root element of the message and its relationships will shape the schema that is loaded. This is the Class diagram loaded into the Schema Composer



Generating XML Schema

Once the diagram loaded, the schema can be generated immediately.



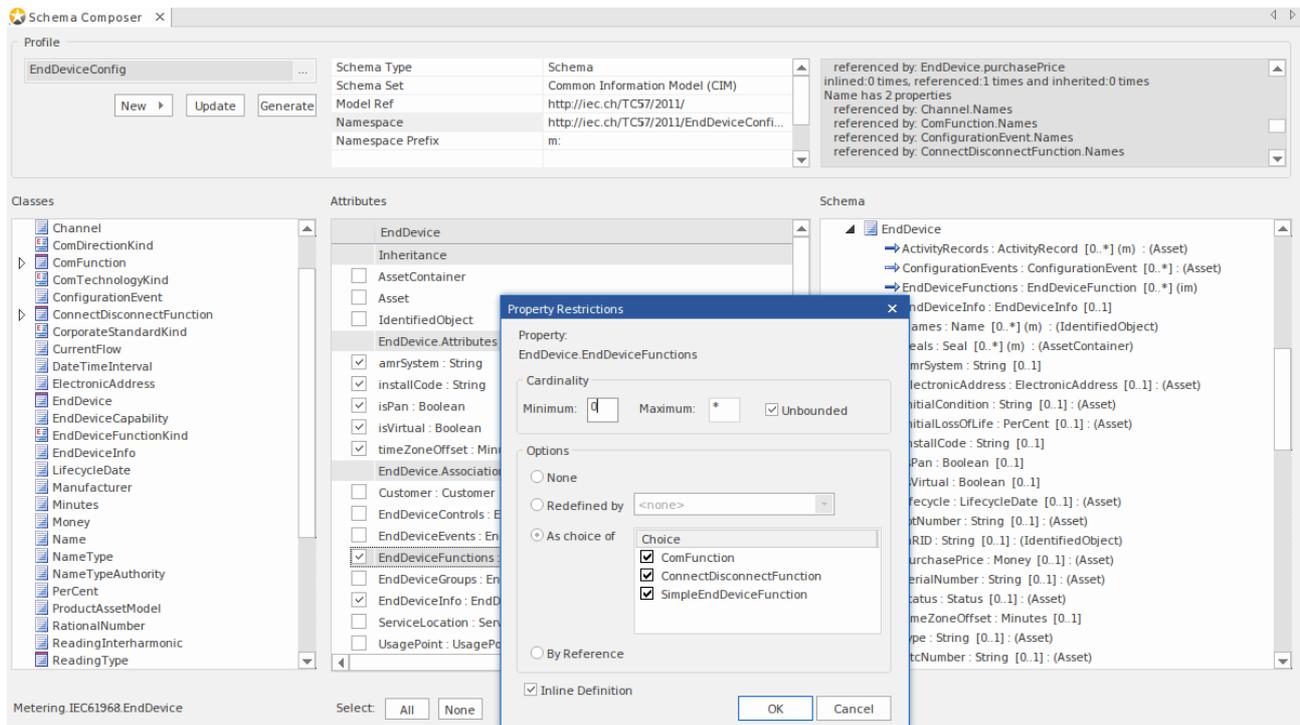
UPCC Schema Guide

This guide describes the composition and generation of a **UPCC** compliant XML schema.

Creation of UPCC Schema

Step	Action
1	Display the Schema Composer .
2	Click 'New Schema'.
3	Enter a unique name for the schema.
4	Select the UPCC Naming and Design rules to use, from the list of standards.
5	Drag one or more < ABIE > components from the <BIE library> into the Class list.
6	Set the Class as a root element using the context menu.
7	Select required attributes (Referenced types are added to the schema).
8	Click on the Update button to save changes.
9	Click on the Generate button and select 'XML Schema'. Click on the OK button .

CIM Schema Guide



This guide describes the creation and generation of a **CIM** compliant XML Schema.

Create a CIM message

Step	Action
1	Display the Schema Composer .
2	Click 'New Schema'.
3	Enter a unique name for this CIM schema (message).
4	Select the Common Information Model.
5	Drag the initial CIM Class(es) into the Class window that best represents the message. Set <i>root</i> elements appropriately using the context menu.
6	If you wish to compose this type using inheritance, select a single ancestor from the inheritance list.
7	Use the checkboxes on the attributes of each class to define the set of properties that will describe this message or schema.
8	Apply restrictions to elements using the context menu on the property.
9	Click update to save the message.

10

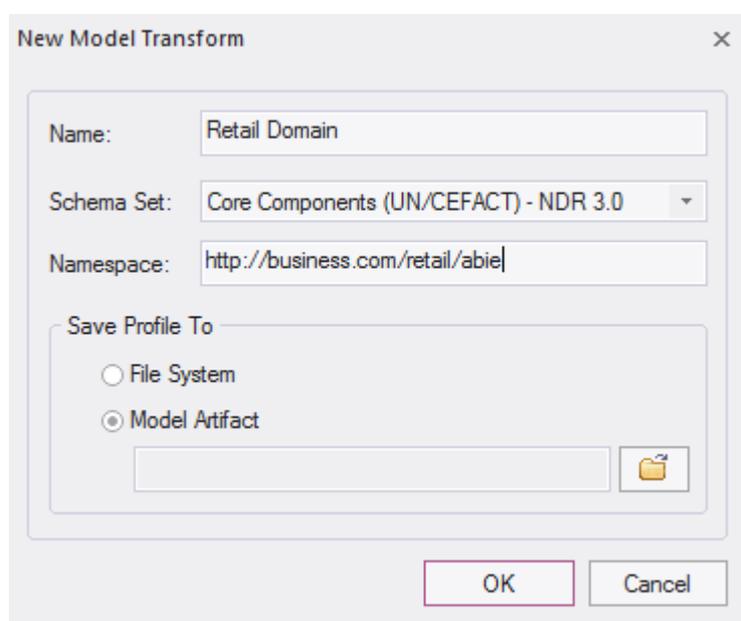
Click the **Generate button** and choose the schema formats you wish to export.

Model compositions

The model composition feature of the **Schema Composer** is useful for creating a sub-model from a core model. This can be as simple as generating a single business Package from a core Package (*CDT library to BDT library transform in UN/CEFACT Core Components standard*) or creating a complete sub-model from a large core model.

The enormity of such a task can be daunting and error prone; *ensuring every type that is referenced by the sub-model is included by the sub-model*, for example. The Schema Composer addresses this problem by automatically working out the dependencies and adding them to the schema for you where necessary.

Create Transform



The screenshot shows a dialog box titled "New Model Transform". It contains the following fields and options:

- Name:** Retail Domain
- Schema Set:** Core Components (UN/CEFACT) - NDR 3.0
- Namespace:** http://business.com/retail/abiel
- Save Profile To:**
 - File System
 - Model Artifact

At the bottom of the dialog are "OK" and "Cancel" buttons.

Define Model Content

Content is added to the model by dropping classes from the model on to the **Schema Composer** class window and choosing which properties to include. The resultant type can mirror the core type or provide a simpler classification. When a property is included, the Schema Composer will check the property type and if the type is missing will add it to the schema automatically.

Reference checking

When a property is excluded that was previously included in the schema and is no longer referenced, the property type is not automatically removed. However the **Schema Composer** will always show the number of references for a type if you select it in the class window. Types that show no references at all can easily be removed.

```
referenced by: Period.DateOfIssue_Start
referenced by: Person.Birth
inlined:0 times, referenced:3 times and inherited:0 times
MyCode has 10 properties
referenced by: Address.Postcode
referenced by: Person.Gender
inlined:0 times, referenced:2 times and inherited:0 times
```

Summary

The process of composing a sub-model is summarized here:

1. Create a Schemer Composer Transform Profile
2. Create elements by dropping Classes from the model into the schema.
3. Include / exclude required properties.
4. Generate the sub model.

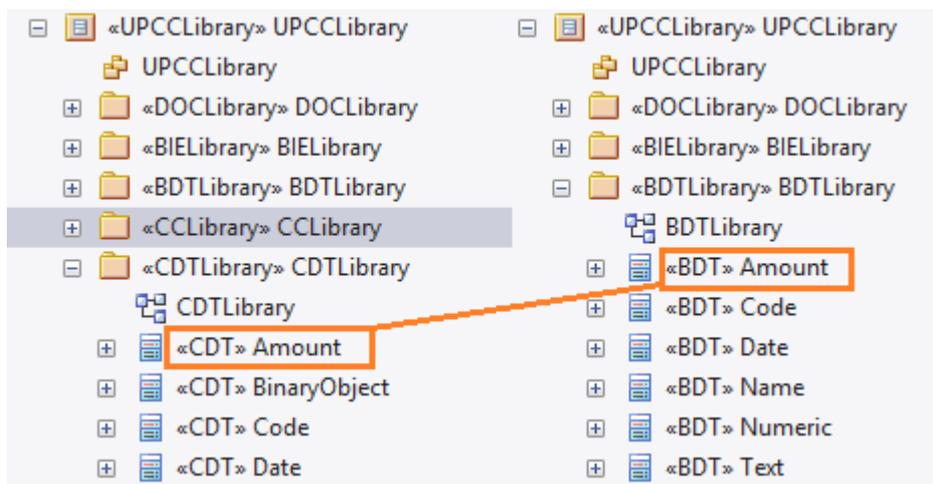
Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer : New > Transform
Menu	Tools Schema Composer : New Transform

UML Profile for Core Components

The **UPCC** framework provides core component and core data type libraries and is available to Enterprise Architect users through the **Model Wizard**. Whether you model according to the UMM specification, or want to leverage the advantages this standard brings, or have a compliance requirement, to model with this technology you will require - as a minimum - a Business datatype library and a Business Information Entity library. The **Schema Composer** can generate these libraries for you.

This image shows a **BDT** library created from a UPCC Core CDT Library



Common Libraries

Libraries shared by both versions of the UML Profile for Core Components.

Library	Description
CCLibrary	The CCTS core component library.
CDTLibrary	The CCTS core datatype library. It contains basic datatypes such as Amount, Code, Text and Graphic.
BIELibrary	A Business library containing ABIE entities based on CCLibrary components. The entities can be composed using the Schema Composer . These can also be modeled using the UML modeling tools available for the technology.
DOCLibrary	A Package typically used for the modeling of Message Assemblies. You can generate the schema for a Message Assembly by loading it into the Schema Composer .-

UPCC Libraries

The UML Profile for Core Components is available in two versions, NDR 3.0 and NDR 2.1. Both profiles describe a common set of libraries, with some differences, as described here:

NDR 3.0

Library	Description
BDTLibrary	A Business library containing BDT types based on CDTLibrary types. The Schema Composer can be used to easily generate the content of a BDTLibrary from selected types in the core library.

NDR 2.1

Library	Description
UDTLibrary	An unqualified datatype library. Basically a mirror of the CDTLibrary for use in a business context. The Schema Composer can be used to easily generate the content of a UDTLibrary from selected types in the core library.
QDTLibrary	A qualified datatype library. The library contains restricted types based on the CDTLibrary with qualified type names. The Schema Composer can easily generate the content of a QDTLibrary from selected types in the core library.

UPCC Diagrams

The UML profile for Core Components uses **UML Class** diagrams for composition of components. There are however specific toolboxes provided by the technology for each of its libraries.

UPCC Toolbox Pages

Common

In this notation, UPCCx represents the **UPCC** profile and x is the version of the NDR

Common	Description
UPCCx - CCLibrary Abstract Syntax	 <p>The screenshot shows a toolbox titled "UPCC2 - CCLibrary Abstract Syntax" with three items: "ACC" (represented by a document icon), "BCC" (represented by a diamond icon), and "ASCC" (represented by a pencil icon).</p>
UPCCx - DOCLibrary	

Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC2 - DOCLibrary Abstract Syntax <ul style="list-style-type: none"> MA ASMA
UPCCx - CDTLibrary Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC2 - CDTLibrary Abstract Syntax <ul style="list-style-type: none"> CDT CON SUP
UPCCx - BIELibrary Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC2 - BIELibrary Abstract Syntax <ul style="list-style-type: none"> ABIE BBIE ASBIE

NDR 3.0

Library Syntax	Description
UPCC - BDTLibrary Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC3 - BDTLibrary Abstract Syntax <ul style="list-style-type: none"> BDT CON SUP

NDR 2.1

Library Syntax	Description
UPCC - UDTLibrary Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC2 - UDTLibrary Abstract Syntax <ul style="list-style-type: none"> UDT CON SUP
UPCC - QDTLibrary Abstract Syntax	<ul style="list-style-type: none"> ▾ UPCC2 - QDTLibrary Abstract Syntax <ul style="list-style-type: none"> QDT CON SUP

Generate Schema

Having defined a schema profile and added the necessary elements and customizations, you can quickly and easily generate the necessary output schema according to the specifications of the relevant technology you are working with. Depending on the technology you are targeting and working with, there may be different output options and formats available to choose from. Note that you may elect to generate multiple output files at once where you are targeting multiple output formats.

If the Schema technology you selected is NIEM, you will generate a Model Subset instead, which you can customize at the model level and output as a schema using the 'Generate NIEM Schema' facility.

One of the options available for all schema sets is Execute custom script; you can use any script that you have written to perform operations on a schema. When you select the Execute custom script option you are prompted to enter the script name, in the format group_name.script_name. The script is then executed.

Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer : Generate
Menu	Tools Schema Composer : Generate

Generate/Export Schema

Field/Option/Button	Action
Technologies	<p>Select the checkbox against each Technology for which you want to generate a schema.</p> <p>If you originally selected, as the Schema Set:</p> <ul style="list-style-type: none"> • * CIM, these options are available: <ul style="list-style-type: none"> - XML Schema (XSD) - Resource Description Framework Schema (RDFS) - Resource Description Framework Schema - Augmented (RDFS) - JavaScript Object Notation (JSON) - Execute Custom Script • * NIEM, these options are available: <ul style="list-style-type: none"> - Model Subset - Execute Custom Script • * UN/CEFACT NDR 3.0, these options are available: <ul style="list-style-type: none"> - Business Data Type (BDT) Transform - Business Information Entity (BIE) Transform - XML Schema (XSD) - Execute Custom Script • * UN/CEFACT NDR 2.1, these options are available: <ul style="list-style-type: none"> - Unqualified Data Type (UDT) Transform - Qualified Data Type (QDT) Transform - Business Information Entity (BIE) Transform - XML Schema (XSD)

	<ul style="list-style-type: none"> - Execute Custom Script • * UBL, these options are available: <ul style="list-style-type: none"> - XML Schema (XSD) - Execute Custom Script • Generic, these options are available: <ul style="list-style-type: none"> - Resource Description Framework Schema (RDFS) - JavaScript Object Notation (JSON) - Generic Model Subset - Execute Custom Script
Generate	<p>Click on this button to generate the schema. If you are generating:</p> <ul style="list-style-type: none"> • Anything other than a NIEM subset, a file Browser window displays; locate and select the directory into which to generate the schema • A NIEM Model Subset, the 'Find Package' dialog displays; locate and select the 'IEPD PIM Niem-coreSubset Package', and click on the OK button <p>The message <i>Export of profile <name> completed</i> displays; click on the OK button to clear this message, then either:</p> <ul style="list-style-type: none"> • Use a file browser to locate and open the schema files in the file directory you specified, or • Expand the Niem coreSubset Package in the Project Browser to see the model subset

Notes

- For NIEM, the Niem-core Subset Package element has the **Tagged Value** defaultPurpose set to subset, to enable the Package to receive the model sub-set
- To generate a schema from the NIEM model subset, click on the <<ModelPackageDescription>> element and select Extensions | NIEM | Generate NIEM Schema
- The **Schema Composer** is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

Generate Schema File

Having defined a schema profile and added the necessary elements and restrictions, you can quickly and easily generate the schema(s). XML schema generation is available in all technologies, but each technology might support additional formats.

Access

NIEM

To generate a schema from the NIEM model subset, right-click on the <<ModelPackageDescription>> element and select 'Extensions | NIEM 3.0 | Generate NIEM 3.0 Schema'.

All other frameworks:

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer : Generate
Menu	Tools Schema Composer : Generate

Schema Formats

Select the checkbox against each schema format to export.

Schema Format	Details
CIM	<ul style="list-style-type: none"> XML Schema (XSD) Resource Description Framework Schema (RDFS) Resource Description Framework Schema - Augmented (RDFS) Javascript Object Notation (JSON) Execute Custom Script
UBL 2.1	<ul style="list-style-type: none"> XML Schema (XSD) Execute Custom Script
Generic	<ul style="list-style-type: none"> XML Schema (XSD) Resource Description Framework Schema (RDFS) Javascript Object Notation (JSON) Execute Custom Script
Execute Custom Script	<p>Although the Schema Composer can generate schema for a number of recognized standards, it also features a scripting solution for those users who want control over the format and medium of the schema. When you specify a script to the generator, it is referring to a language script such as Javascript that has exists in your model. How and what the script produces is pretty much up to you. How the script accesses the schema in the Schema Composer is documented in the Schema Composer Scripting.</p>

	Integration.
UN/CEFACT NDR 3.0	<ul style="list-style-type: none">• XML Schema (XSD)• Execute Custom Script
UN/CEFACT NDR 2.1	<ul style="list-style-type: none">• XML Schema (XSD)• Execute Custom Script

Generate

Click on this button to generate the schema.

Use a file browser to locate and open the schema files.

Notes

- The **Schema Composer** is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect
- You can edit and validate XML documents including **XSD** schema, using Enterprise Architect
- You can set Enterprise Architect as the default document handler for XML documents

Generate a Model Subset (Transform)

Having defined the content of your sub-model or library and applied any restrictions, you can now generate the model. The model transforms that can be performed depend on the technology associated with the profile. Each technology and the transforms it supports are listed here:

Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Open Schema Composer : Generate
Menu	Tools Schema Composer : Generate

Model Transform

Select the model transform(s) to run.

Transform Option	Description
NIEM	<p>NIEM Model Subset</p> <p>This option will generate a NIEM Model Subset containing the schema described by the profile.</p> <p>When you click the OK button, you will be prompted to select the target IEPD PIM Niem-coreSubset Package.</p> <p>The subset will then be created at this location. If the subset already exists it will be replaced.</p> <ul style="list-style-type: none"> The Niem-core Subset Package will have the Tagged Value defaultPurpose set to subset, to enable the Package to receive the model sub-set To generate a schema from the NIEM model subset, click on the <<ModelPackageDescription>> element and select 'Extensions NIEM Generate NIEM Schema' <p>Execute custom script</p> <p>A user defined language script such as Javascript will be executed. The script can obtain access to the profile using the Schema Composer automation interfaces.</p>
Generic	<p>Generic model Subset</p> <p>The Package specified in the profile will be populated with the types it contains, using any qualifier that is entered, and apply any restrictions to the UML elements created. Types that exist in the target Package will be overwritten. New properties will be added. Types or properties that exist in the target but that no longer exist in the profile will not be removed by this process.</p> <p>Execute custom script</p> <p>A user defined language script such as Javascript will be executed. The</p>

	<p>script can obtain access to the profile using the Schema Composer automation interfaces.</p>
UN/CEFACT NDR 3.0	<p>BDT Library</p> <p>A Business Datatype Library will be populated from core datatypes listed in the profile. Stereotypes will be transformed according to the CCTS specification. The types may be more restricted than their core counterparts. Properties of datatypes that exist will be overwritten. New properties and types will be added to the library. Types are matched by name and stereotype.</p> <p>Types or properties that exists in the target which no longer exist in the profile will not be removed by this process.</p> <p>BIE Library</p> <p>A Business Information Entity library will be populated from aggregated core components. listed in the profile. Stereotypes will be transformed according to the CCTS specification. Properties of datatypes that exist will be overwritten. New properties and types will be added to the library. Types are matched by name and stereotype.</p> <p>Types or properties that exists in the target which no longer exist in the profile will not be removed by this process.</p> <p>Execute custom script</p> <p>A user defined language script such as Javascript will be executed. The script can obtain access to the profile using the Schema Composer automation interfaces.</p>
UN/CEFACT NDR 2/1	<p>UDT Library</p> <p>Performs an unqualified copy of selected core datatypes to a UDT library.</p> <p>QDT Library</p> <p>A Qualified Business Datatype Library will be populated from core datatypes listed in the profile. The names of the resultant types will be qualified by the named qualifier in the profile. Stereotypes will be transformed according to the CCTS specification. Properties of datatypes that exist will be overwritten. New properties and types will be added to the library. Types are matched by name and stereotype.</p> <p>BIE Library</p> <p>A Business Information Entity library will be populated from aggregated core components. listed in the profile. Stereotypes will be transformed according to the CCTS specification. Properties of datatypes that exist will be overwritten. New properties and types will be added to the library. Types are matched by name and stereotype.</p> <p>Execute custom script</p> <p>A user defined language script such as Javascript will be executed. The script can obtain access to the profile using the Schema Composer automation interfaces.</p>

Generate

Click on the **OK button** to generate the schema. When the generation is complete, the message *Export of profile <name> completed* displays.

You can then expand the Package in the **Project Browser** to see the generated UML model.

Notes

- The **Schema Composer** is supported in the Corporate, Systems Engineering, Business and Software Engineering and Ultimate editions of Enterprise Architect

Schema Importer

You can import Schemas compatible with the **Schema Composer**, into Enterprise Architect using the Schema Importer. The Schema Importer validates the Schema and creates a *Schema* type Schema Composer Profile upon successful validation, that can be viewed directly in the Schema Composer.

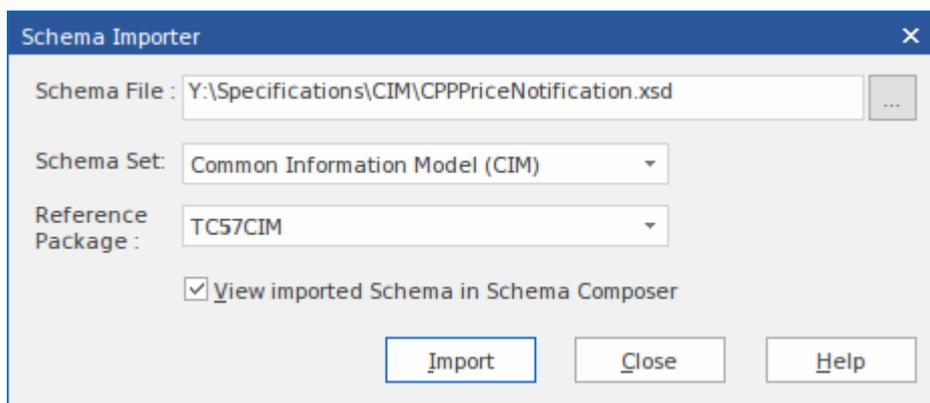
Currently, you can use the Schema Importer to import these Schemas:

- Common Information Model (**CIM**) specific XML Schema
- Common Information Model (CIM) specific RDFS XML

Access

Access	Method
Ribbon	Design > Tools > Schema Composer > Import for Schema Composer
Menu	Tools Schema Importer

Import a Schema using the Schema Importer



Field/Option/Button	Action
Schema File	Type the directory path and filename from which to import the Schema file.
Schema Set	Select the type of Schema being imported. Currently the Schema Importer supports importing CIM specific: <ul style="list-style-type: none"> • XML Schema and • RDFS XML
Reference Package	Select the Package containing the common elements specific to the schema set. The Schema Importer will validate the elements in the Schema being imported against the elements in the reference Package.

View imported Schema in Schema Composer	Select this option to open the imported Profile in the Schema Composer .
Import	Click on this button to start the import process.
Close	Click on this button to close the 'Schema Importer' dialog.
Help	Click on this button to display this Help page.

Notes

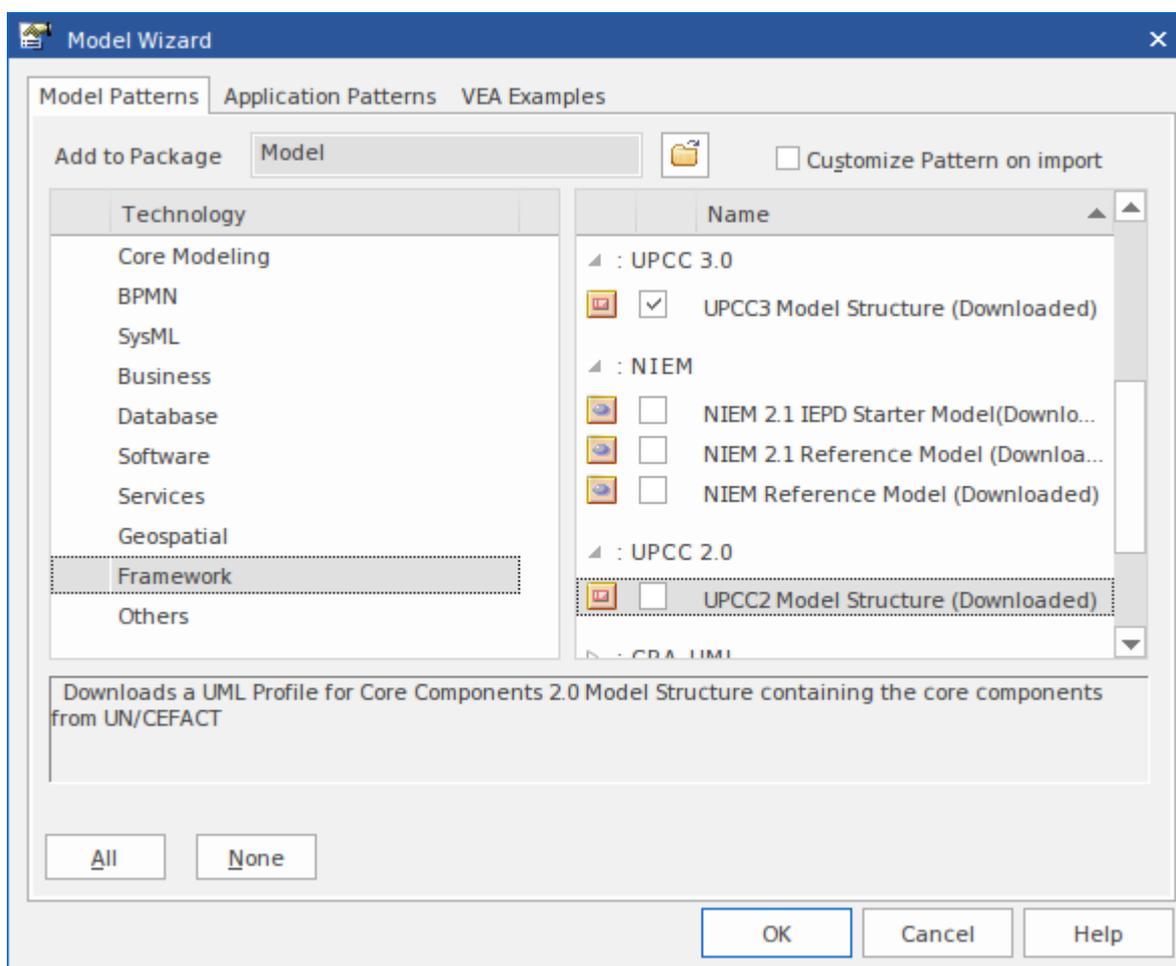
- The progress of import will be displayed in the **System Output Window**
- The **Schema Composer** will validate the Schema against the elements in the Reference Package before importing the Schema; if validation fails, the Schema elements that fail validation will be displayed in the System Output Window and the import process will stop
- Double-click on a validation error entry in the System Output Window to open the Schema in Enterprise Architect's internal file editor and go to the source of the error
- If validation succeeds, the 'New Schema Definition' dialog displays, through which you can save the imported Profile in the file system or as an Artifact in the current model

Available Frameworks

Using the Enterprise Architect **Model Wizard** you can deploy any of the frameworks supported by the **Schema Composer** - such as **NIEM**, **CIM** and **CCTS** - to your model in minutes, providing a powerful UML medium for modeling in those technologies.

The frameworks are also available directly from the Sparx Systems **Reusable Asset Service** (via the Cloud Server 'Cloud Connection' dialog and then 'Frameworks' on the 'Model Patterns' tab of the Model Wizard).

Note: In addition to the custom frameworks such as CIM and NIEM, it is possible to use standard Class models to rapidly build generic Schemas, so if you are not targeting a particular meta-model, it might be simplest to model your data in UML and use the resultant model as input to the Schema Composer.



National Information Exchange Model (NIEM)

This is the [National Information Exchange Model](#) published by the **NIEM Program Management Office (PMO)**.

Enterprise Architect provides these resources for modeling in NIEM:

- Provided Frameworks including NIEM core, NIEM domains, code lists and external schema adapters:
 - NIEM 2.1 modeled using NIEM-UML 1.0
 - NIEM 3.0 modeled using NIEM-UML 1.1
 - NIEM 3.1 modeled using NIEM-UML 1.1
- NIEM subset creation:
 - The **Schema Composer** helps you create a subset of a NIEM conformant namespace

- NIEM schema generation:
 - Generation of complete NIEM IEPDs from a model Package description in either NIEM 2 or NIEM 3 formats

Common Information Model (CIM)

This is the [CIM specification](#) published by International Electrotechnical Commission (IEC) Technical Committee 57.

Enterprise Architect provides these resources for modeling in **CIM**:

- Schema Composition
 - XML schema (**XSD**)
 - Resource Descriptor format (RDFS)
 - Resource Descriptor augmented format
 - Javascript Object notation (JSON)
 - Add-in integration
 - **Scripting** integration

Core Component Technical Specification (CCTS)

This is the [CCTS specification](#) published by UN/CEFACT.

Enterprise Architect provides these resources for modeling in **CCTS**:

- UML Frameworks:
 - **UPCC** 2.1 core component libraries
 - UPCC 3.0 core component libraries
 - UMM 2.0 business requirements, choreography and information views.
- Business Component Library Creation / Management
 - **Schema Composer** for **ABIE** and **BDT** composition
 - Add-in integration
 - **Scripting** integration
- Business Component Schema Composition
 - Schema Composer for **XSD**, JSON
 - Add-in integration
 - Scripting integration

Universal Business Language (UBL)

UBL is a **CCTS** implementation published by [OASIS](#) that is proving popular with European governments for consolidating information exchange between agencies.

Enterprise Architect provides these resources for the composition of business documents using UBL:

- **UML Framework**
 - UBL 2.1 Main Document Libraries
 - UBL 2.1 Common Component Libraries
- Business Document Composition
 - **Schema Composer** for component composition
 - Schema Composer for document composition
 - Schema Composer for schema generation
 - Add-in integration
 - **Scripting** integration

Add-in Framework (Custom)

In addition to the above methodologies the **Schema Composer** integrates with Enterprise Architect **Automation Interface** to support any individual or group in implementing their own. An Add-in that registers its interest to Enterprise Architect in offering Schema generation capabilities will have the opportunity to offer any of its products in the Schema Composer Generation tool.

Scripting Framework (Custom)

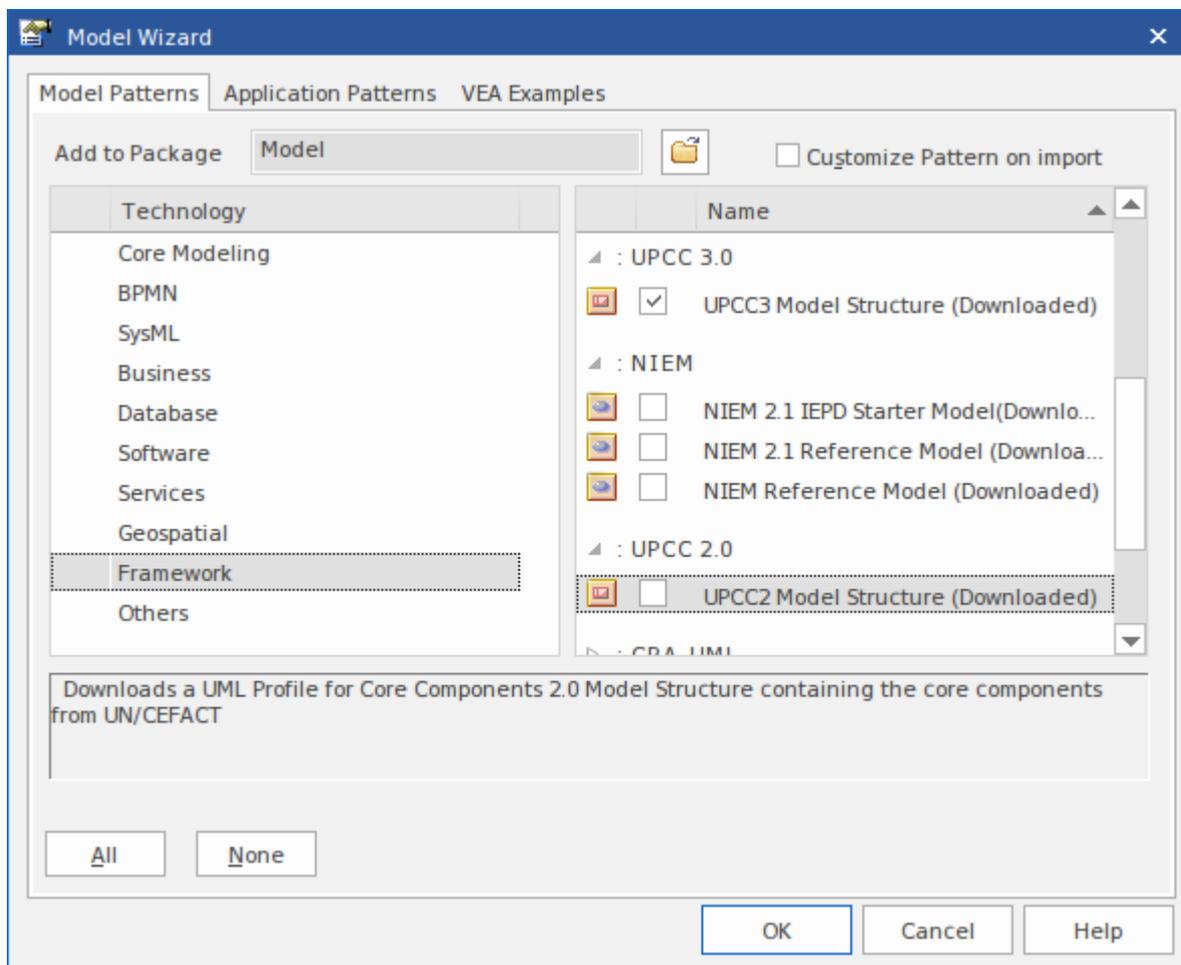
The **Schema Composer** also offers unconditional control over generation of schema for any profiles created with it. By writing their own script an author can access the definition of any schema and ultimately produce whatever documents they wish, in a format of their choosing.

Install a Core Framework

Enterprise Architect provides a rich and diverse range of modeling technologies including every standard listed in the **Schema Composer**. These frameworks are available as UML models and/or MDG Technologies using Enterprise Architect's **Model Wizard**. The models themselves are also directly accessible from Enterprise Architect's **Reusable Asset Service**.

Note: If you are modeling a generic solution and not directly using a core framework such as **CIM** or **UBL**, you do not need to install a core framework/model. In that case you are best served creating a data model using simple UML classes with attributes.

Model Wizard



Access

Access	Method
Ribbon	Design > Package > Model Wizard
Menu	Package New

Context Menu	Right-click on Package Add a Model using Wizard
Keyboard Shortcuts	Ctrl + Shift + M
Other	Project Browser caption bar menu > New Model from Pattern

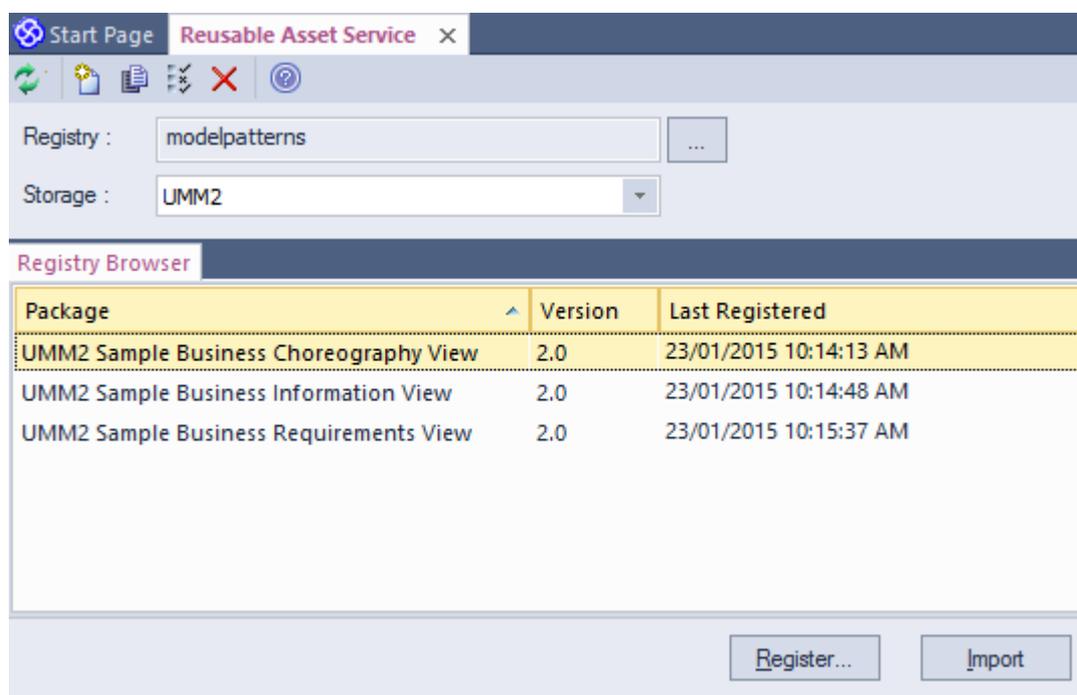
Note

You can limit the MDG Technologies to use by selecting the Ribbon bar option: 'Configure > Technology > Manage'. Here you can see which technologies are currently enabled.

Import Model

Step	Action
1	Display the Model Wizard .
2	Select the 'Model Patterns' tab.
3	Highlight the Technology.
4	Select the Technology standards to import.
5	Click OK to import the framework to your model.

Reusable Asset Service



Access

Main menu: Tools | **Reusable Asset Service**

Import Model

Step	Action
1	Connect to the Reusable Asset Service .
2	Choose from the available list of Repositories
3	Select the UML model Package
4	Click OK to import the selected Package to your model.

Schema Composer Automation Integration

The **Schema Composer** can be accessed from the Enterprise Architect **Automation Interface**. A client (script or **Add-In**) can obtain access to the interface using the 'SchemaComposer' property of the 'Repository' object. This interface is available when a Schema Composer has a profile loaded.

Schema Composer Addin Integration

Enterprise Architect **Add-Ins** can integrate with the **Schema Composer** by providing alternatives to offer users for the generation of schemas and sub models.

Schema Composer Scripting Integration

Although the **Schema Composer** provides out-of-the-box schema composition based on a variety of popular technologies, its scripting integration provides you with some flexibility in how you might go about implementing your own requirements. There are three ways in which you might leverage scripting within the Schema Composer:

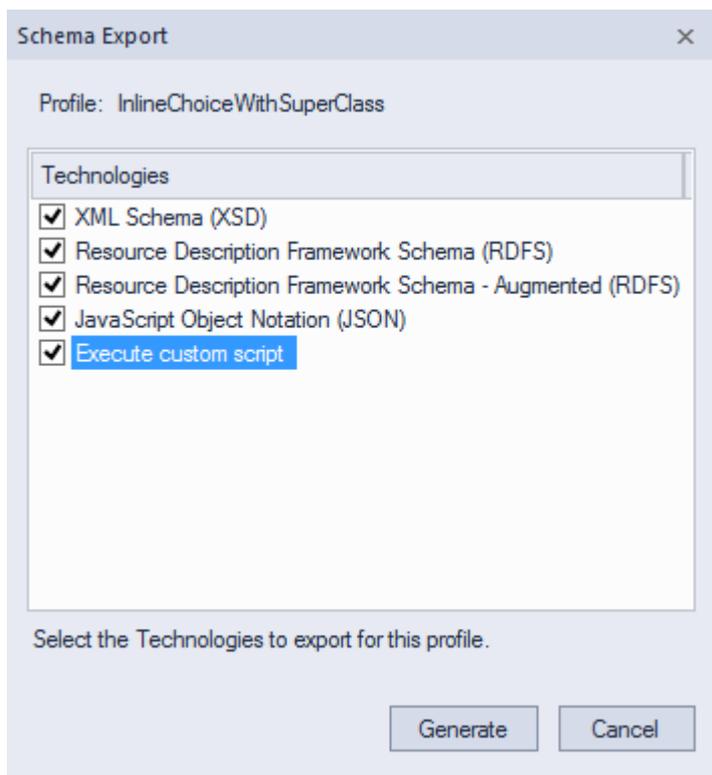
- Provide custom schema generation using a scripting language
- Provide custom model transformation using a scripting language
- Provide custom stereotype mapping to any standard model transform (such as **UPCC**)

Model Transformation by script

While the **Schema Composer** provides in-built transforms for various frameworks, you can always write your own, using the composition tools of the Composer to design the schema, then performing a custom transform with a hand crafted script.

Schema Generation by script

When you select a message in the **Schema Composer** and click generate, you are presented with a number of export formats. One of those choices is 'Execute custom script'



Schema Iteration Scripting Example

This example demonstrates accessing the **Schema Composer** in an Enterprise Architect script written in

Javascript. The script first obtains an interface to the Schema Composer and then traverses the schema, printing out the types and each of its properties.

```
/*
 * Script Name:
 * Author: Sparx Systems
 * Purpose: Demonstrate Schema Composer use in JavaScript
 * Date: 6/2/2015
 */

function main()
{
    var schema as EA.SchemaComposer;

    var xmlType as EA.SchemaType;
    var xmlTypeEnum as EA.SchemaTypeEnum;

    var xmlProp as EA.SchemaProperty;
    var xmlPropEnum as EA.SchemaPropEnum;

    var xmlChoice as EA.SchemaType;
    var xmlChoiceEnum as EA.SchemaTypeEnum;

    // Get SchemaComposer
    schema = Repository.SchemaComposer;

    // Get Schema Types Enumerator
    xmlTypeEnum = schema.SchemaTypes;

    // Get total number of types in schema
    var count = xmlTypeEnum.GetCount();

    // Enumerate Types
    xmlType = xmlTypeEnum.GetFirst();
    while(xmlType)
    {
        Session.Output( "Type: " + xmlType.TypeName );

        xmlPropEnum = xmlType.Properties;
        if(xmlPropEnum)
        {
            xmlProp = xmlPropEnum.GetFirst();
        }
    }
}
```

```
while(xmlProp)
{
    var sPropDesc = xmlProp.Name;
    sPropDesc += ":@"

    if(xmlProp.IsPrimitive())
        sPropDesc += xmlProp.PrimitiveType;
    else
        sPropDesc += xmlProp.TypeName;

    if(xmlProp.IsByReference())
    {
        sPropDesc += " (by reference) ";
    }
    if(xmlProp.IsInline())
    {
        sPropDesc += " (inline) ";
    }

    Session.Output( " " + sPropDesc + " , cardinality: " + xmlProp.Cardinality );
    xmlChoiceEnum = xmlProp.Choices;
    if(xmlChoiceEnum.GetCount(>0)
    {
        Session.Output( " choice of: ");
        xmlChoice = xmlChoiceEnum.GetFirst();
        while(xmlChoice)
        {
            Session.Output( " " + xmlChoice.TypeName );
            xmlChoice = xmlChoiceEnum.GetNext();
        }
        xmlProp = xmlPropEnum.GetNext();
    }
    xmlType = xmlTypeEnum.GetNext();
}
}
main();
```

Intelli-sense help in scripting

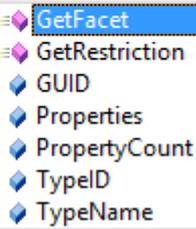
The **Scripting** editor in Enterprise Architect will help you write script that interacts with the **Schema Composer**, by providing intelli-sense on the properties and methods of its automation interface.

```

// Enumerate Types
xmlType = umlModelTypeEnum.GetFirst();
while(xmlType)
{
    print( "Type: " + xmlType.TypeName );

    umlPropEnum = xmlType.
    if(umlPropEnum)
    {
        umlProp = umlPropE
        while(umlProp)
        {
            if(umlProp.IsP

```



Stereotype mapping in Model Transformation

Stereotyping forms a large part of the MDG technology approach. Individual UML profiles for an MDG Technology define stereotypes to offer useful classifications for its elements. It is a common requirement when going from a core framework to a business model or sub-domain to reassign the stereotype. When you work with a **CCTS** framework the business components you generate have their stereotype automatically generated by Enterprise Architect according to a mapping defined by the CCTS specification (ACC to **ABIE**, for example).

When you open or create a model transform profile in the **Schema Composer** you can specify a script to perform this mapping for you. The script can be selected from the **Properties window**. It can be written in either Javascript, JScript or VBScript, and only has to implement this function (described here in Javascript notation):

```

function TranslateStereotype( srcStereo )
{
    var destStereo = srcStereo
    if (srcStereo == "ABIE")
    {
        destStereo = "My_ABIE"
    }
    return destStereo;
}

```

MDG Technologies - UML Profile Extensions

The **Schema Composer** works with MDG technologies. The standards it uses for schema generation, *other than Generic*, are only meaningful for models that adhere to that framework. However it is quite easy to extend an existing MDG Technology. You may wish to ensure that elements authored in your business specific domain or sub-domain provide consistently named metadata or 'Tagged values'.

The Schema Composer supports extensions to UML profiles / frameworks through its scripting integration. When a script is assigned in the Schema Composer, the transform process will invoke this script and ask it to translate keywords. These keywords are usually UML stereotypes. If a particular technology is associated with the profile, the Schema Composer will invoke this function, passing it the name of the MDG Technology.

The script can return the input name, and no mapping will take place, or it can return the name of another MDG Technology. When this occurs, the Schema Composer will again ask for the function to optionally map any UML profiles. Finally it will ask the script to translate the stereotypes from the core technology.

The result of the model transform would then be that any UML elements of the sub model will show the extended **Tagged Values** in addition to any core Tagged Values.

Example script that maps MDG Technology

```
function TranslateStereotype (stereo)
{
  var newStereo = stereo;
  if (stereo == "UPCC3")
  {
    newStereo = "XXX UPCC3"
  }
  return newStereo;
}
```

Example script that maps UML profile

```
function TranslateStereotype (stereo)
{
  var newStereo = stereo;
  if (stereo == "UPCC3 - BIE Library Abstract Syntax")
  {
    newStereo = "UPCC3 - BIE Library XXX Syntax"
  }
  return newStereo;
}
```

Example script that maps UML Stereotype

```
function TranslateStereotype (stereo)
{
```

```
var newStereo = stereo;
if (stereo == "ABIE")
{
    newStereo = "XXX ABIE";
}
return newStereo;
}
```

