



ENTERPRISE ARCHITECT

用户指南系列

软件建模高级

Author: Sparx Systems

Date: 20/06/2023

Version: 16.1

创建于  **ENTERPRISE
ARCHITECT**

目录

信息工程	8
示例图表	9
使用数据模型类型	10
概念数据模型	11
实体关系图 (图表)	12
逻辑数据模型	16
物理数据模型	17
DDL变换	19
更多信息	23
支持数据库管理系统	24
生成数据库定义语言 (DDL)	25
为对象生成DDL	26
编辑 DDL模板	29
DDL 模板语法	31
DDL模板	32
用于 DDL 生成的基本模板	33
用于更改 DDL 生成的基本模板	36
DDL 宏	37
元素字段宏	38
列字段宏	41
约束字段Macros	42
DDL函数宏	44
DDL属性宏	48
模板中的 DDL 选项	53
DDL 限制	56
导入数据库架构	57
创建和管理数据模型	60
数据建模设置	61
数据建模Notations	62
DDL名称模板	66
数据类型	68
DBMS 产品之间的映射数据类型	69
DBMS 产品转换为一个包	70
库表的数据类型转换	71
数据库数据类型	72
MySQL数据类型	74
Oracle数据类型	75
管理 DBMS 选项	76
数据库对象	77
数据库视图	78
数据库连接	80
甲骨文包	83
创建操作容器	84
#数据库#	86
查询	86
数据库序列	88
函数数据库	91

数据库过程	94
数据库表	97
数据库约束库表/索引	98
主键	101
非聚集主键	104
数据库索引	105
独特约束	108
外键	109
选择约束	112
库表触发器	114
数据库库表栏目	116
创建数据库库表列	117
删除数据库库表列	119
重新排序数据库库表列	120
使用库数据库库表属性	121
设置数据库类型	122
设置数据库库表拥有着/架构	123
设置MySQL选项	124
设置 Oracle数据库库表属性	125
创建数据库库表	126
数据库生成器	128
约束/索引	130
数据库约束库表/索引	131
主键	134
数据库索引	137
独特约束	140
外键	141
选择约束	144
库表触发器	146
列	148
创建数据库库表列	149
删除数据库库表列	151
重新排序数据库库表列	152
在数据库生成器中工作	153
打开数据库生成器	157
SQL便签本	159
执行 DDL	161
数据库比较	164
创建一个数据模型图表	170
示例数据模型图表	172
从模型模式数据模型	173
开始	175
XML Schema (XSD)	177
Schema 编辑器	178
Schema 编辑器Profiles	180
创建架构配置文件	182
架构组成	184
类图表	189
架构分析	191
生成架构	192
选择架构配置文件	193

生成架构文件	195
CIM架构指南	197
NIEM架构指南	199
UPCC架构指南	201
模型组成	202
生成模型子集 (变换)	204
核心组件 (UPCC) 的UML配置文件	206
可用的框架	209
安装核心框架	212
架构Importer	215
Schema 编辑器自动化集成	217
Schema 编辑器Addin集成	218
Schema 编辑器脚本集成	219
MDG 技术- UML配置文件Extensions	224
XSD 模型	226
建模XSD	227
XSD图表	229
架构包	230
全局元素	232
当地元素	234
全局属性	236
本地属性	238
属性组	240
复杂类型	242
简单类型	244
团体	246
任何	248
任何属性	250
联盟	252
集团模型	254
枚举	256
来自抽象类模型的 XML	258
默认UML到 XSD 映射	260
生成XSD	261
生成全局元素	263
导入XSD	264
全局元素和复杂类型	266
XSL 转换	267
XSL变换模型	269
执行 XSL变换	271
调试一个 XSL变换	272
XML 验证	273
面向 XML 服务架构	276
WSDL	277
WSDL 1.1模型结构	278
WSDL模型	280
WSDL命名空间	283
WSDL信息	284
WSDL信息部件	286
WSDL端口类型	288
WSDL端口类型操作	290

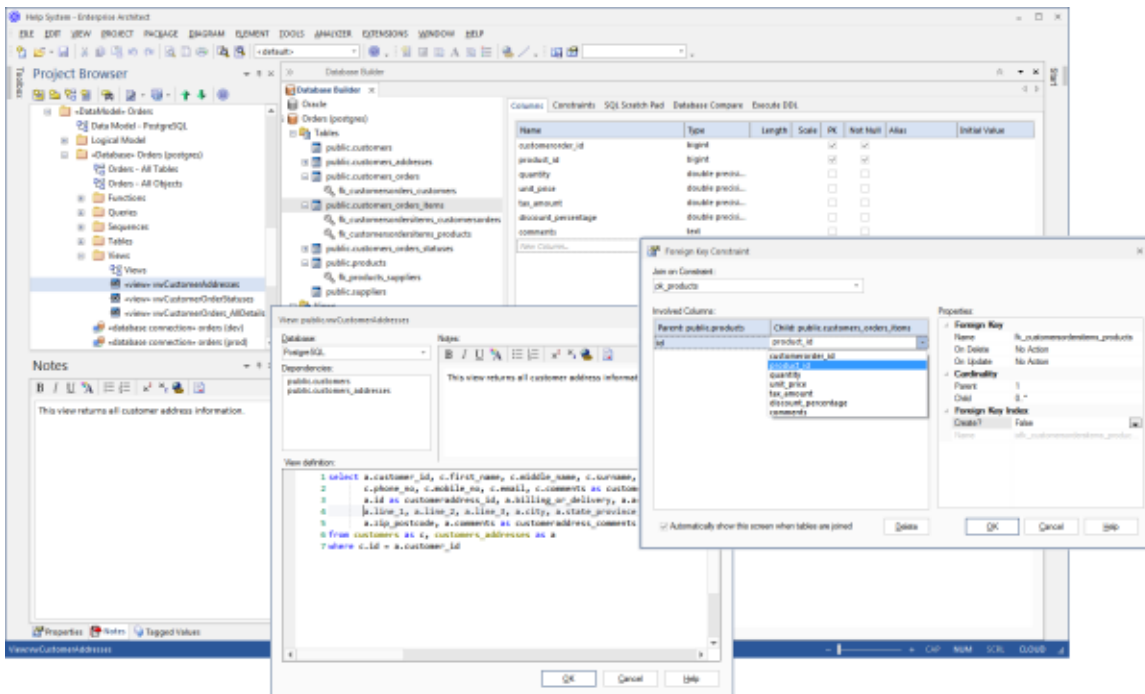
WSDL 捆绑	292
WSDL 捆绑操作	294
WSDL 服务	296
WSDL 文档	298
生成 WSDL	300
导入 WSDL	302
SoaML	303
SoaML 工具箱 Pages	305
SOMF 2.1	307
全国信息交换 NIEM 建模 2.1	308
全国信息交换 NIEM 建模	315
NIEM 的 UML 配置文件	316
下载 NIEM 参考模型	324
创建 NIEM IEPD	325
自定义您的 IEPD 模型	328
NIEM IEPD 生成	332
创建 NIEM 数据模型	333
使用 Schema 编辑器对 NIEM 进行子集化	335
浏览示例	338
示例 NIEM 架构	342
导入 NIEM XML Schema	350
地理空间模型	352
开始	353
ArcGIS 地理数据库	354
示例图表	355
导出 ArcGIS XML 工作空间	356
导入 ArcGIS XML 工作空间	358
Geography Markup Language (GML)	360
示例图表	361
使用 GML 建模	362
更多信息	364
集成来自外部提供商的数据	365
到 WebEA 的项目超链接	368
访问的信息	370
自定义 SBPI 服务	371
ISBPIServicePlugin 接口	373
处理请求	374
设置回调	376
设置配置	377
示例脚本	379
编写自定义集成插件	381
ISBPIntegrationPlugin 接口	383
创建 WebEA 超链接	385
获取授权详情	386
获取默认字段映射	388
获取默认类型映射	390
获取字段	392
获取图标	393
获取项目	394
获取项目讨论	396
获取项目列表	397

GetItemNotes	398
获取项目	399
获取菜单列表	400
获取类型	401
项目链接	402
项目未链接	403
物品已更新	404
后新讨论	405
PostNewItem	406
PostOAuthCode	407
PostUpdateItem	408
PostUpdateItemNotes	409
刷新OAuthToken	410
设置授权	411
设置回调	413
设置配置	414
链接项	416
外部项目详细信息	419
导航外部数据	422
配置	425
安装和配置	428
集成插件	435
添加/编辑数据提供者	439
数据提供者高级认证	445
演练：ServiceNow集成	447
演练：Jira集成	454
演练：DevOps集成	460
演练：Polarion集成	469
演练：Enterprise Architect集成	475
故障排除	480

信息工程

设计、创建和管理概念、逻辑和物理数据模型

基于模型的系统开发的强大之处在于能够可视化、分析和设计系统的各个方面。能够与系统的其他模型一起查看和管理信息和数据提供了极大的清晰度并减少了出错的机会。Enterprise Architect广泛支持数据建模学科，从概念模型中的信息表示一直到数据库对象的生成。无论您是从UML模型生成数据库对象，还是将遗留DBMS逆向工程到模型中进行分析，该工具特征都将节省时间和宝贵的项目资源。



此图显示了数据库生成器接口，包括 DDL 生成和外键对话框。

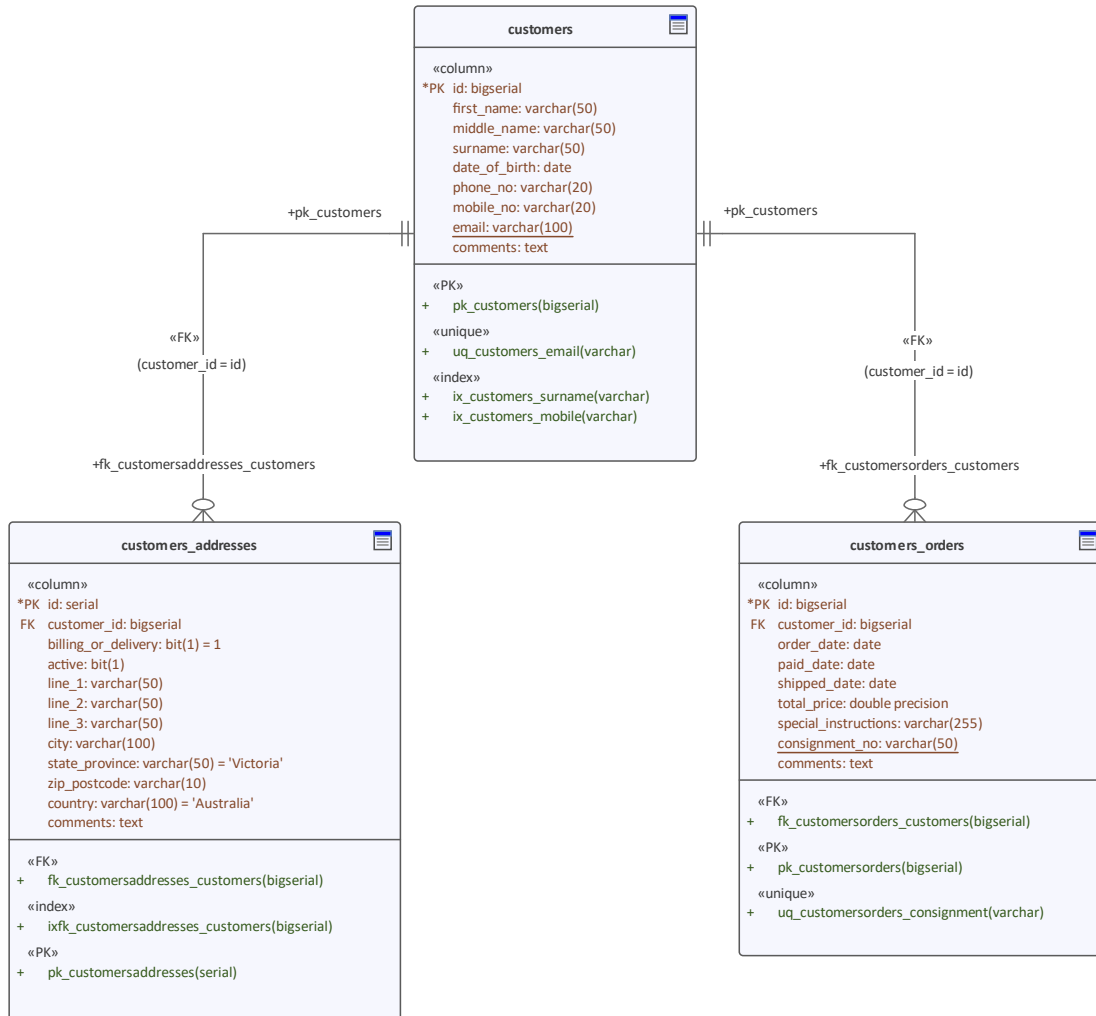
Enterprise Architect支持概念、逻辑和物理层的信息建模。使用许多标准特征，这些模型可以相互连接，提供可追溯性。逻辑和物理模型也可以使用完全可变换引擎自动生成。可以使用方便的逆向工程功能导入、分析和比较遗留系统。

在本主题中，您将学习如何使用包括数据库构建器在内的丰富特征工具集来设计、创建、管理和可视化数据，包括将数据模型反向和正向工程到实时数据库。

数据库工具可用于创建和维护物理数据模型，并可以连接到正在运行的 DBMS，因此您可以导入、生成、比较和更改实时数据库。

示例图表

使用Enterprise Architect的数据库工程特征，您可以创建丰富的对象模型，这些模型构成了从概念到逻辑到物理的任何抽象级别的数据模型。这些模型是通过从工具箱中添加库表和其他数据库对象或通过逆向工程和现有数据库来自一系列 RDBMS 的模型来创建的。数据库图A包含表、视图、过程、序列和函数。库表注释为主键，外键使用专门的关联关系建模。在此示例中，用户创建了客户及其地址和订单的简单物理数据模型。



物理数据模型显示带有列和主键和外键的表。

使用数据模型类型

Enterprise Architect提供了许多特征来协助创建信息模型的过程，包括开发概念、逻辑和物理模型的能力，以及能够跟踪模型之间的基本概念的能力。可以为各种数据库系统开发物理模型，并且正向和逆向工程允许这些模型与实时数据库同步。

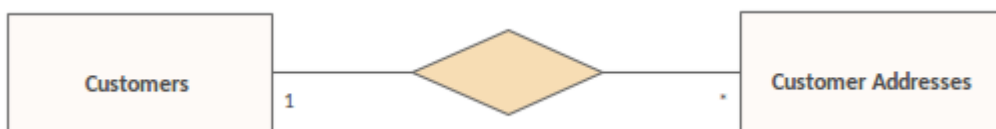
数据模型

类型	描述
概念数据模型	<p>概念数据模型，也称为领域模型，建立给定领域的基本概念和语义，并帮助将这些信息传达给广大利益相关者。</p> <p>概念模型也是项目分析阶段的常用词汇；它们可以在Enterprise Architect中使用实体关系或UML类模型创建。</p>
逻辑数据模型	<p>逻辑数据模型为概念模型元素添加更多细节并细化域的结构；它们可以使用实体关系或UML类模型来定义。</p> <p>逻辑数据模型的一个好处是它提供了物理模型和后续数据库实施的基础。</p> <p>实体-关系建模是一种抽象和概念化的数据库建模方法，用于生成例如关系数据库及其需求的模式或语义数据模型，在实体-关系图表（图表）中可视化。</p> <p>ERD 可帮助您构建概念数据模型，直至为目标 DBMS 生成数据定义语言 (DDL)。</p> <p>可以使用变换变换A逻辑模型转换为物理数据模型。</p>
物理数据模型	<p>Enterprise Architect中的物理数据模型帮助您可视化您的数据库结构并自动导出相应的数据库模式；您专门为此目的使用Enterprise Architect的UML配置文件进行数据建模。</p> <p>该概要文件提供了UML标准的有用扩展，将表和关系的数据库概念映射到类和关联的UML概念上；您还可以模型数据库列、键、约束、索引、触发器、参照完整性和其他关系数据库特征。</p> <p>由于Enterprise Architect帮助您可视化同一存储库中的每种类型的数据模型，您可以轻松管理每个抽象级别之间的依赖关系，以最大限度地提高可追溯性并验证系统实施的完整性。</p>

概念数据模型

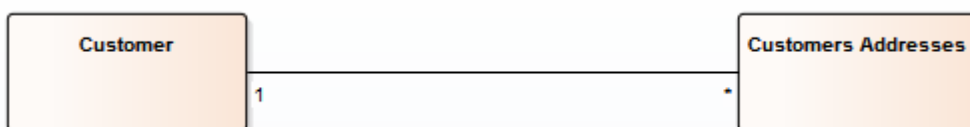
概念数据模型是数据模型的最抽象形式。由于其简单性，它有助于将想法传达给广泛的利益相关者。因此，特定于平台的信息，例如数据类型、索引和键，从概念数据模型中排除。其他实现细节，例如过程和接口定义，也被排除在外。

这是概念数据模型的示例，使用Enterprise Architect支持的两种表示法呈现。



显示一对多关系的实体关系图

使用实体关系 (ER) 表示法，我们将数据概念“客户”和“客户地址”表示为具有一对多关系的实体。我们可以使用UML类和关联来表示完全相同的语义信息。



显示相同的一对多关系的Unified Modeling Language图

是否使用UML或ER表示法来表示项目中的数据概念取决于相关利益相关者的经验和偏好。概念数据模型中说明的数据概念的详细结构由逻辑数据模型定义。

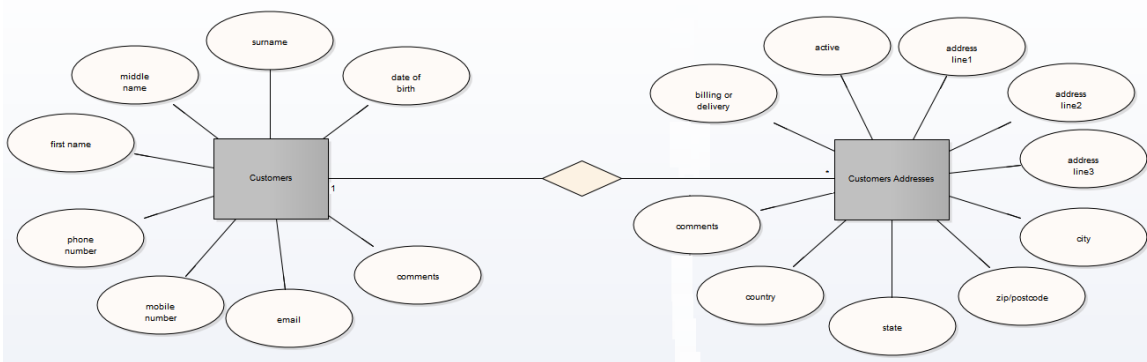
实体关系图 (图表)

根据在线维基百科：

实体关系模型 (*ERM*) 是数据的抽象和概念表示。实体-关系建模是一种数据库建模方法，用于以自上而下的方式生成系统 (通常是关系数据库) 及其需求的一种概念模式或语义数据模型。由该过程创建的图表称为实体-关系图、图表或图表。

Enterprise Architect图表的实体关系图

Enterprise Architect中的实体关系图基于 Chen 的 ERD 构建块：实体 (库表) 表示为矩形，属性 (列) 表示为椭圆 (连接到它们的实体)，关系之间表示为菱形连接器。



Enterprise Architect中的 ERD 技术可在从构建概念数据模型到为目标 DBMS 生成数据定义语言 (DDL) 的每个阶段提供帮助。

ERD 和 ERD 转换

Enterprise Architect使您能够通过使用与Enterprise Architect安装程序集成的MDG 技术快速简单地开发实体关系图。

实体关系图功能以以下形式提供：

- 实体关系图类型，通过“新图表”对话框访问
- 工具箱图表的一个实体关系图表页面
- 工具箱快捷方式“菜单和快速链接器中的实体关系元素和关系条目

Enterprise Architect还提供转换模板，将实体关系图转换为数据建模图，反之亦然。

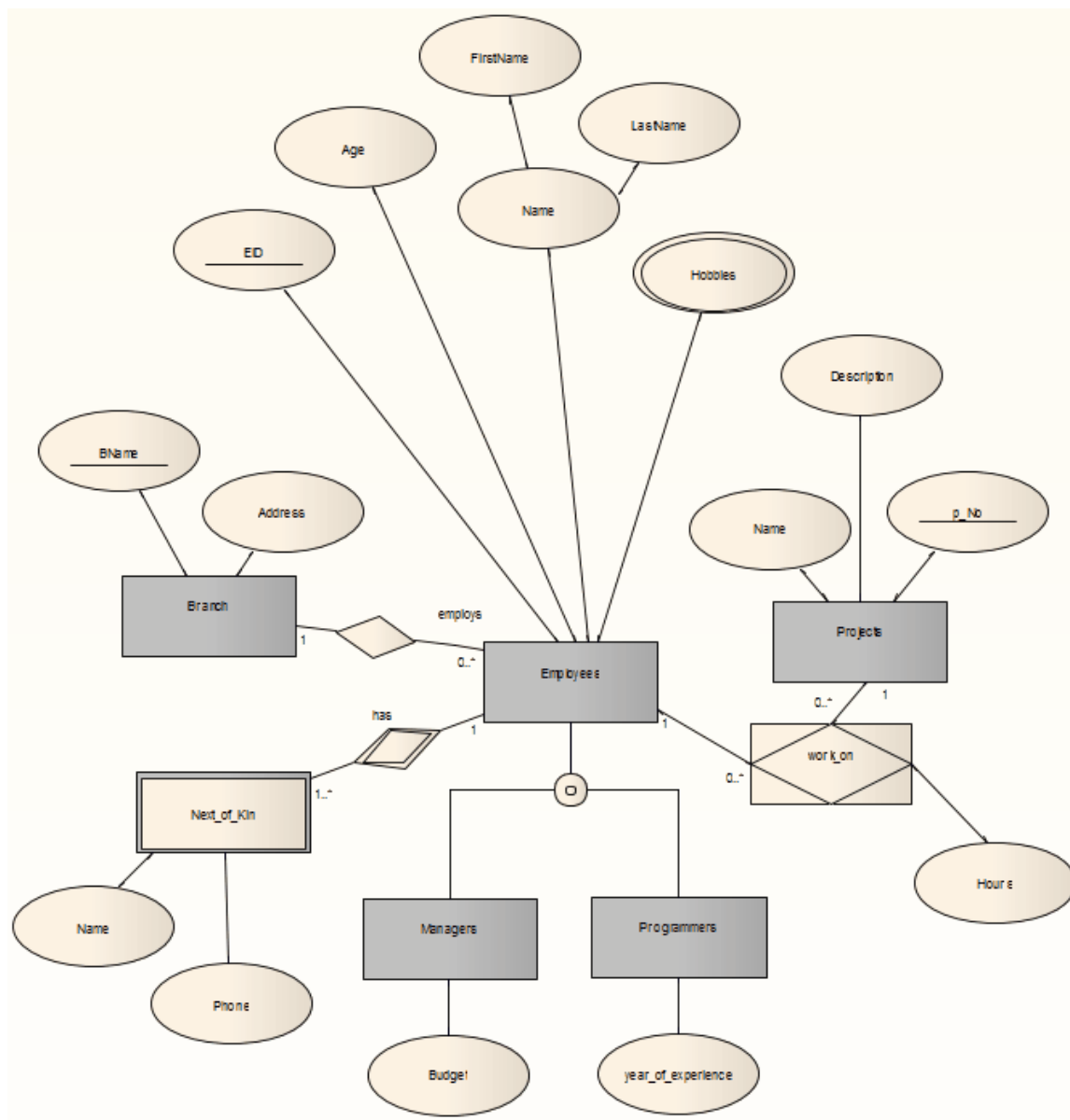
实体关系图表工具箱

您可以通过在工具箱的“查找图表图表”对话框实体指定“工具箱关系图”来访问“实体关系图表”工具箱

- ▾ ERD
 - Entity
 - Attribute
 - N-ary Association
- ▾ ERD - Relationships
 - Connector
 - Relationship
 - Disjoint
 - Overlapping

- 实体是唯一可识别的object或概念；关系连接器的 SourceRole 和属性定义中的 "Multiplicity"属性可用 定义参与此关系的实体的基数
- 属性是实体或关系类型的属性
- N-ary关联表示一元（多对多递归）或三元关系，也可用于表示实体之间具有属性的关系；N-ary关联元素应始终位于连接器的目标端
- 连接器是实体和属性之间以及两个属性之间的连接器
- 关系是菱形连接器，表示实体之间有意义的关联
- Disjoint 和 Overlapping 表示超类实体和子类实体之间的关系

A的实体关系图



标记值

标注标记值可以修改一些实体关系图的组成部分，如下所示：

部件	标记值/笔记
实体	isWeakEntity 注记：如果为真，则该实体为弱实体。
属性	属性类型 注记：有四个有效选项：'normal'、'primary key'、'multi-valued'和'derived'
属性	通用数据类型 注记：定义每个属性的通用数据类型。

属性	<p>dbmsDataType</p> <p>注记：为每个属性定义自定义的 DBMS 数据类型。此选项仅在 <i>commonDataType</i> 标签设置为 <i>na</i> 时可用。</p> <p>您必须首先通过 设置 > 参考 > 设置 > 数据库数据类型”功能区选项定义自定义类型。</p>
N-ary 关联	<p>是递归的</p> <p>注记：如果为真，N-ary 关联表示多对多递归关系。</p> <p>对于一对多和一对一的递归关系，我们建议使用普通关系连接器。</p> <p>有时您可能想要限制菱形关系连接器的伸展；只需选择一个关系连接器，右键单击以显示上下文菜单，然后选择 在光标处弯曲线”选项。</p>
关系	<p>弱</p> <p>注记：若为真，则关系为弱关系。</p>
脱离 重叠	<p>参与</p> <p>注记：有两个有效的选项，'partial' 和 'total'。</p>

注记

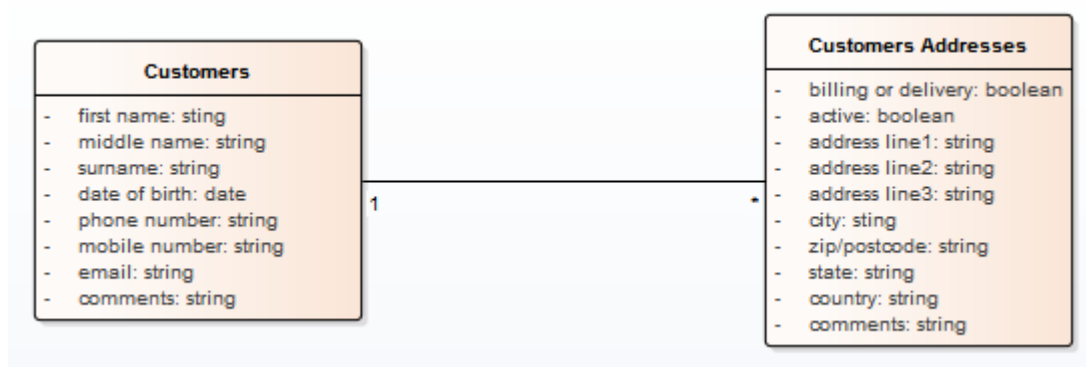
- Enterprise Architect 的企业版、统一版和终极版支持实体关系图

逻辑数据模型

逻辑数据模型有助于定义系统中数据元素的详细结构以及数据元素之间的关系。它们细化了概念数据模型引入的数据元素，并形成了物理数据模型的基础。在Enterprise Architect中，逻辑数据模型通常使用UML类表示法表示。

示例

此图是逻辑数据模型的简单示例。逻辑模型向概念模型添加了细节，但没有达到指定将使用的数据库管理系统的级别。



概念数据模型与库表建模客户及其地址。

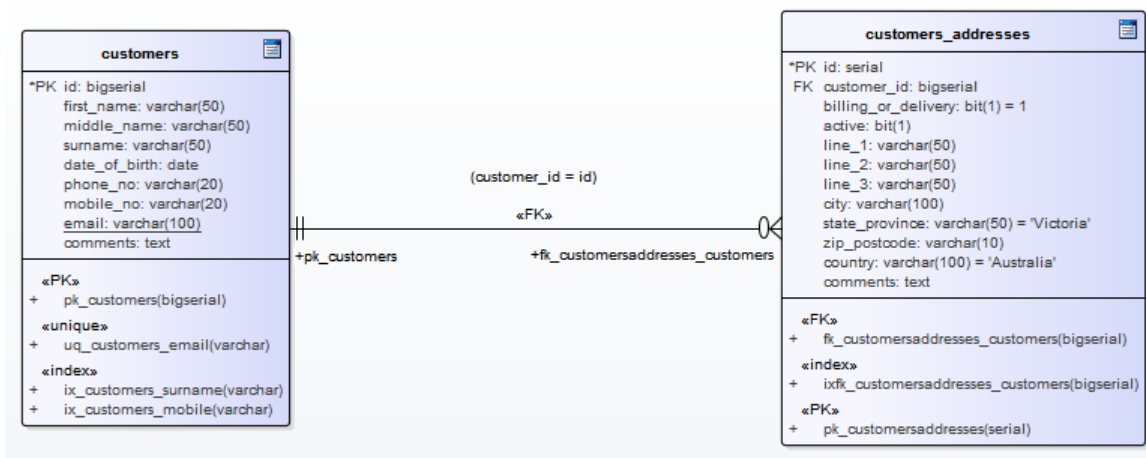
注记数据元素“客户”和“客户地址”包含UML属性；名称和通用数据类型保持平台独立。与特定DBMS实现相关的平台特定数据类型和其他元数据由物理数据模型定义。

物理数据模型

物理数据模型直观地表示由关系数据库模式实现的数据结构。除了提供数据库结构的可视化抽象之外，定义物理数据模型的一个重要好处是您可以从模型中自动导出数据库模式。这是可能的，因为物理数据模型捕获的元数据的丰富性及其与数据库模式的各个方面的紧密映射，例如数据库表、列、主键和外键。

示例数据模型

此示例显示了一个物理数据模型，可用于自动生成数据库模式。每个库表由一个UML类表示；库表列、主键和外键使用UML属性和操作建模。该模型演示了信息工程连接器样式的使用。



符号

示例模型是使用Enterprise Architect的用于数据建模的UML配置文件定义的；表之间的关系使用默认的信息工程符号。

信息工程是Enterprise Architect支持帮助数据建模者识别关系中的基数的三种符号之一。您可以通过选择“设计 > 图表 > 管理 > 属性”功能区选项，单击“连接器”页面并在“连接器符号”下拉列表中选择所需的选项来更改符号。您很可能会将符号更改为 IDEFX1，但 UML2.1 符号也可用。

默认数据库管理系统

在创建物理数据模型之前，建议您为项目设置默认 DBMS。设置默认 DBMS 可确保在图表上创建的所有新数据库元素都自动分配默认 DBMS。

如果未设置默认 DBMS，则会在未分配 DBMS 的情况下创建新表，这会限制Enterprise Architect正确模型物理对象的能力。例如，Enterprise Architect无法确定列的正确数据类型列表。

您可以使用以下方法设置默认 DBMS 类型：

- ‘开始>外观>首选项>首选项>源代码工程>代码编辑器’，或
- ‘设置>参考>设置>数据库数据类型或
- ‘开发>数据建模>数据类型或
- 代码生成工具栏中的第二个数据输入字段

注记：通过数据库生成器建模时，默认的 DBMS 是在模型级别定义的（作为 <<Database>>包的标记值'DBMS'）而不是在项目级别，从而在项目涉及时允许更大的灵活性多个 DBMS。

DDL变换

DDL 转换将逻辑模型转换为结构化的数据模型，以符合支持的 DBMS 之一。目标数据库类型由模型中设置为默认数据库的 DBMS 确定（参见数据库数据类型帮助主题，设置为默认值”选项）。然后可以使用数据模型自动生成运行语句以在系统支持的数据库产品之一中运行。

DDL 转换使用并演示了中间语言对许多特定于数据库的概念的支持。

概念

概念	影响
库表	将一对一映射到类元素上。 转换支持“多对多”关系，创建汇合库表。
柱子	一对一映射到属性。
首要的关键	列出所有涉及的列，以便它们存在于类中，并为它们创建主键方法。
外键	A特殊的连接器，其中源和目标部分列出了所有涉及的列，以便： <ul style="list-style-type: none"> • 列存在 • 目标类中A匹配的主键，并且 • 转换创建适当的外键

MDG 技术自定义默认映射

针对新的、用户定义的 DBMS 的 DDL 转换需要MDG 技术来将 PIM 数据类型映射到新的目标 DBMS。

为此，创建一个名为“UserDBMS Types.xml”的MDG 技术.xml 文件，将 UserDBMS 替换为添加的 DBMS 的名称。将文件放在 EA\MDGTechnologies 文件夹中。MDG 技术文件的内容应具有以下结构：

```
< MDG .技术版本1 “1.0”>
<Documentation id="UserdataTypes" name="Userdata Types" version=" 1 .0" notes="类型for UserDBMS"/>
<代码模块>
<CodeModule 语言="用户数据" 注释="">
<代码选项>
<CodeOption name="DBTypeMapping-bigint">BIGINT</CodeOption>
<CodeOption name="DBTypeMapping-blob">BLOB</CodeOption>
<CodeOption name="DBTypeMapping-boolean">TINYINT</CodeOption>
<CodeOption name="DBTypeMapping-text">CLOB</CodeOption>
...
</代码选项>
</代码模块>
</代码模块>
```

</ MDG .技术>

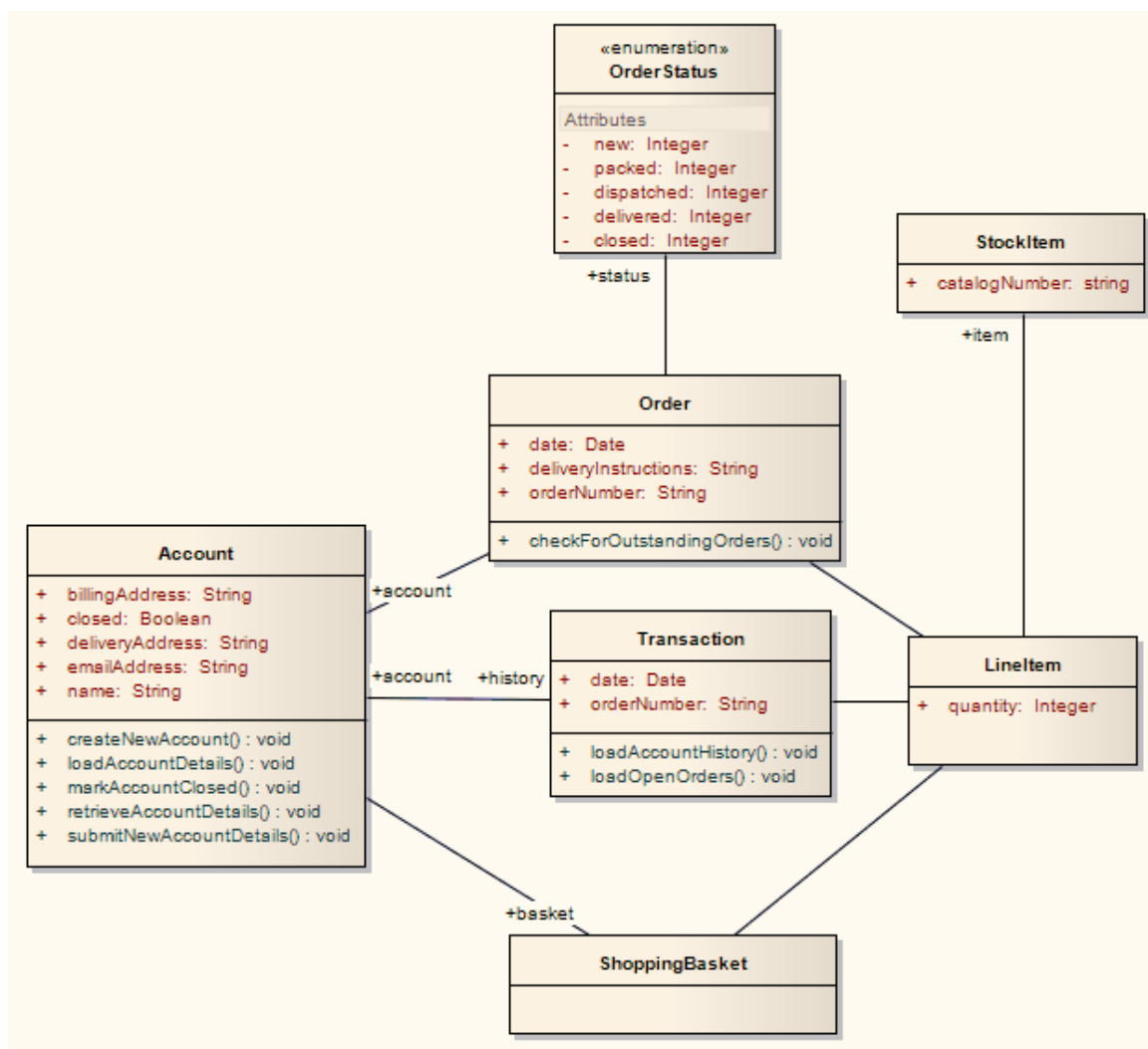
例如，“text”是映射到新 DBMS 的“CLOB”数据类型的公共类型（如“数据库数据类型”对话框中所列）。

注记

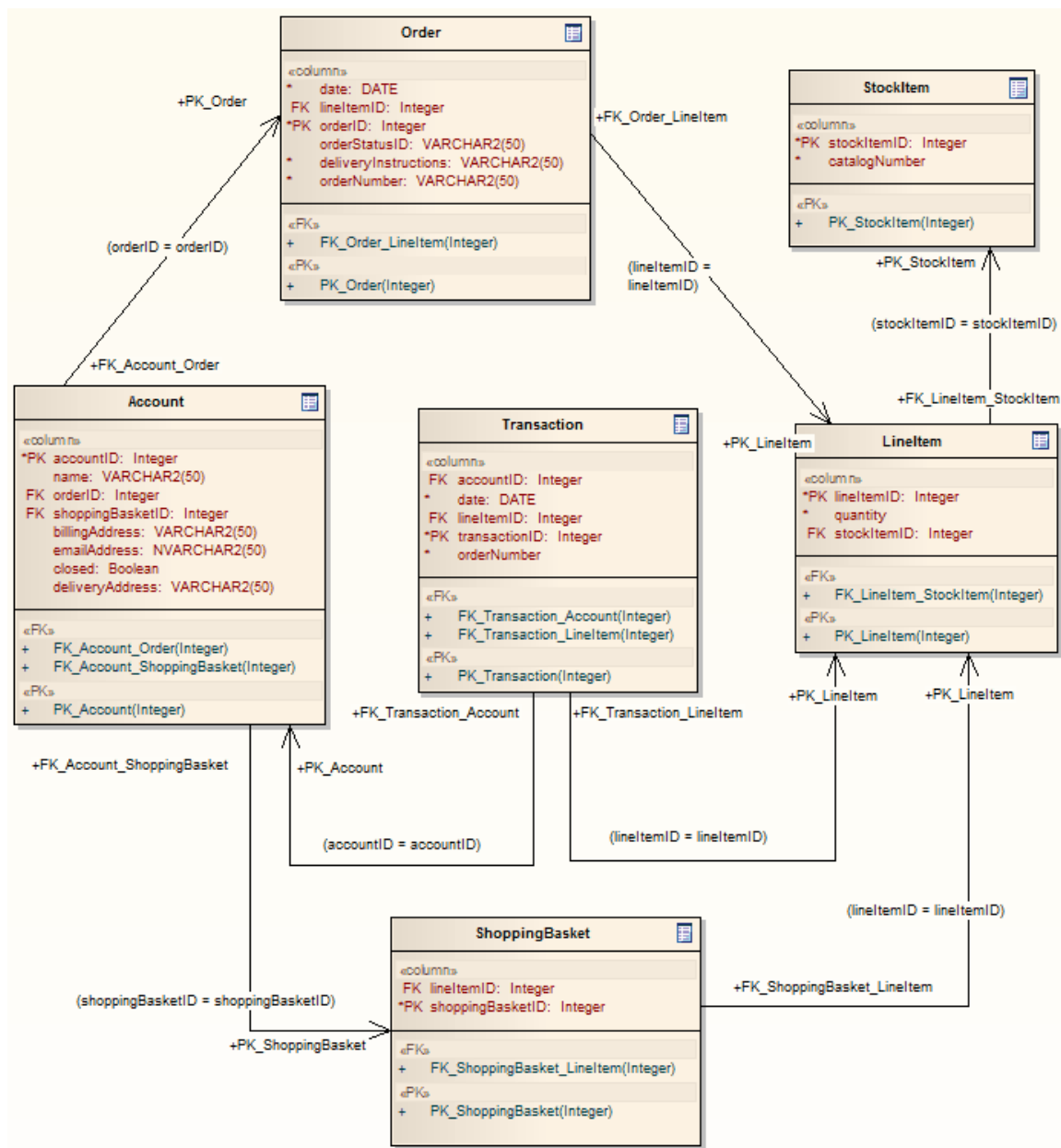
- 您可以在转换后定义逻辑模型中未描述的 DBMS 特定方面，例如存储过程、触发器、视图和选择约束；查看 *Physical Data* 模型帮助

示例

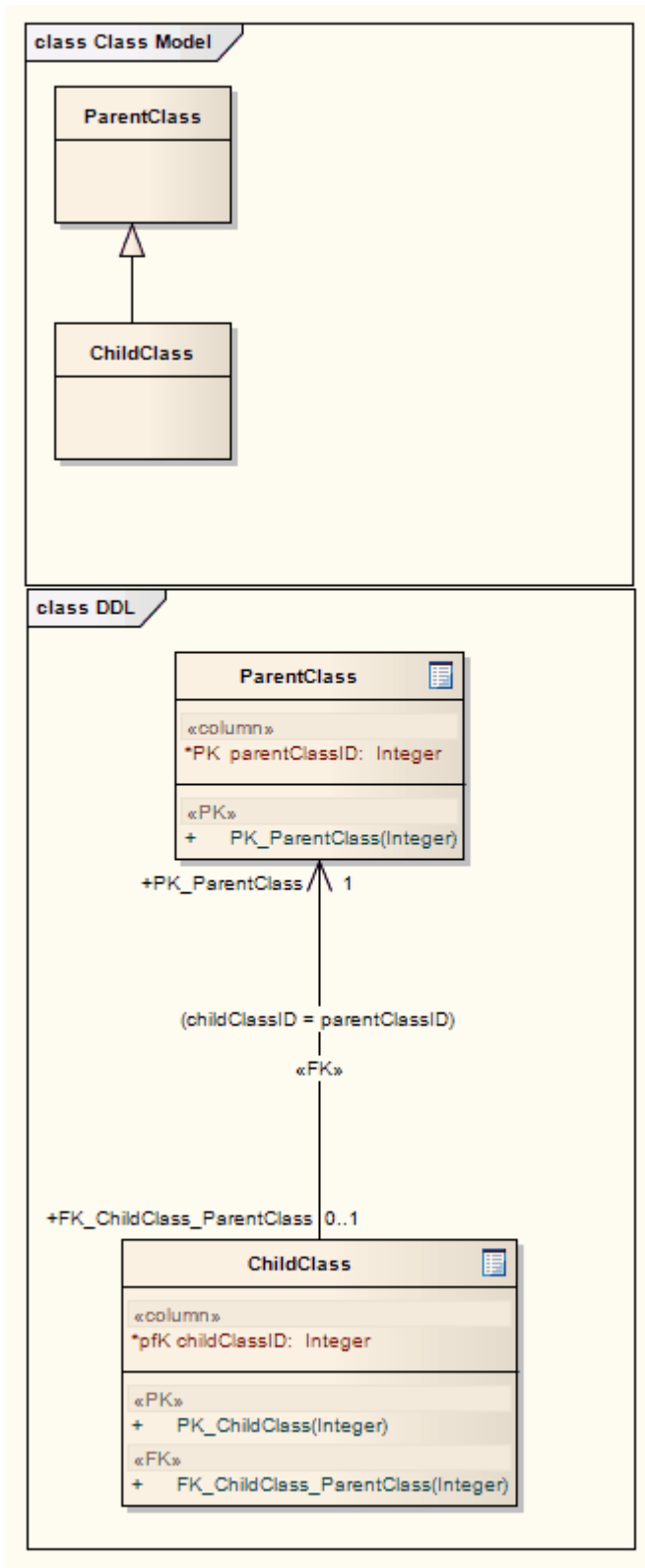
PIM 元素



改造后成为PSM元素



泛化是通过为子元素提供父元素的外键来处理的，如图所示。不支持向下复制继承。



更多信息

版信息

Enterprise Architect的企业统一版和终极版中提供了数据库生成器。

支持数据库管理系统

Enterprise Architect内置了对各种数据库管理系统的支持，但它也提供了扩展产品以支持其他 DBMS 的灵活性。DDL 模板编辑器可用于定义如何为不受支持的 DBMS 生成 DDL，转换模板可用于为不受支持的 DBMS 定义到物理模型的新转换，并且可以为现有或新 DBMS 定义新数据类型。

Enterprise Architect为这些数据库管理系统提供建模结构和正向和反向工程数据库模式的能力：

- DB2 (*)
- 火鸟
- MS Access 97、2000、2003、2007、2013
- 女士#
服务器从 2005 年开始，所有版本，包括 Express 和 Azure #
数据库
- MariaDB
- MySQL v4、v5
- 从 9i 开始的 Oracle (所有版本)
- PostgreSQL (包括版本 12)
- SQLite
- Informix (#)
- 安格尔 (#)
- 基间 (#)
- Sybase Adaptive Server Anywhere (Sybase ASA) (#)
- Sybase Adaptive服务器企业(Sybase ASE) (#)

(*) - 仅在窗口和 Linux 环境中托管时与 DB2 兼容。

(#) - 不会对这些 DBMS 进行进一步开发，因为Enterprise Architect用户群不常用这些产品。这将使Sparx Systems能够将精力集中在广泛使用的其他数据库建模领域。

注记

- 要为特定 DBMS 执行数据建模，您的存储库中必须具有该 DBMS 的适当数据类型；您可以从Sparx Systems网站的“资源”页面下载最新的数据定义

生成数据库定义语言 (DDL)

一旦定义了物理模型并对对象建模，Enterprise Architect可以为各种对象生成数据库定义语言 (DDL)，包括数据库表、视图、函数、序列和过程。这是一种节省时间的机制，并减少了在其他工具中手动执行此操作可能引入的错误。正向工程由一组模板管理，这些模板定义如何将UML构造转换为目标 DBMS 中的对象。为所有受支持的 DBMS 提供了标准模板，并且可以编辑这些模板以自定义 DDL 的生成方式。在不支持开箱即用的 DBMS 的情况下，可以使用现有模板作为起点和参考来创建一组新模板。

在正向工程 DDL 时，可以将输出定向到一个文件（或一系列文件，每个object一个）或 DDL 执行引擎。执行引擎允许您立即执行 DDL，通过活动连接定位实时数据库。如果您将输出定向到文件，您可以稍后在方便时对实时数据库执行 DDL。生成的文件可以使用代码编辑器打开，通过选择 F12、Ctrl+E 或 Alt+7，您可以在 Enterprise Architect 中查看 DDL。

为对象生成DDL

在创建数据库模型时，您可以为单个object、一包对象或完成数据模型生成 DDL。唯一的区别是调用生成 DDL 过程的方式。

访问

打开数据库生成器窗口，然后使用上下文菜单并选择“生成DDL”。

功能区	开发 > 数据建模 > 数据库构建器 > 点击object、包或数据模型节点：生成DDL
-----	--

生成选项卡

字段/按钮	行动
包	单击  按钮并使用导航员窗口（查找包”对话框的一个版本）浏览要为其生成DDL的包。 (注记：此字段可能不会在所有情况下都显示。)
包括所有子包	选中此复选框以将对象包含在“选择要生成的对象”列表中的子包中。
删除目标文件	将对象生成为单个文件时，完整的文件名与object一起存储，并显示在“选择要生成的对象”列表的“目标文件”列中。 单击此按钮可删除所有现有文件名并提示新文件名。
选择要生成的对象	此字段按显示的顺序显示将为其生成 DDL 的对象列表。如果您需要更改此顺序以解决object依赖关系，请单击要移动的object并单击   按钮将该object在序列中向上或向下移动一个位置。 选择要为其生成 DDL 的每个object。点击： <ul style="list-style-type: none"> • 全部按钮选择每个项目 • 无按钮清除所有选择 • 按住 Ctrl 键的同时选择多个对象中的每个对象，以选择多个单独的对象 • 按住 Shift 键时块中的第一个和最后一个对象，以选择块中的每个object
保存生成的订单	如果您更改了列出对象的顺序，请在单击生成按钮时选中复选框以保存新序列。
刷新	重新加载对象列表，将每个object恢复到以前的位置（如果object位置已更改）。
单个文件	如果要保存生成的 DDL 到单个文件，请选择此单选按钮。 单击  按钮以浏览文件路径和文件名。
每个表的单独文件	如果要为每个object生成的 DDL 保存到单独的文件中，请选择此单选按钮。

	<p>钮。</p> <p>当您单击生成按钮时，系统会依次提示您输入每个object的目标文件名（如果尚未指定）。</p>
生成到 DDL 执行引擎	<p>如果要将在 DDL 保存到执行引擎（数据库构建器的“执行 DDL”选项卡），请选择此单选按钮。</p> <p>功能执行引擎提供了执行生成的#脚本并立即响应执行中的错误，而无需创建外部文件并将其加载到另一个工具中。</p> <p>如果数据库构建器打开，“生成到 DDL 执行引擎”是默认选项。</p>
生成	<p>单击此按钮以使用您选择的选项运行生成运行过程。</p>
视图	<p>如果您已将 DDL 生成到单个外部文件，请单击此按钮以查看输出。</p> <p>默认情况下，Enterprise Architect使用默认代码编辑器。但是，您可以在“首选项”对话框中定义一个替代的默认 DDL 编辑器（开始>应用程序>首选项>首选项>源代码工程>代码编辑器>DDL”）。</p>
关	<p>单击此按钮可关闭对话框。如果您没有生成 DDL，此按钮也会放弃为object生成 DDL。</p>

选项选项卡

如果您不想执行它们启动的操作，请将这些标志中的任何一个设置为False。

团体	选项
库表生成选项	<p>表- 表示应生成库表元素的 DDL (*)</p> <p>Primary Keys - 表示应该为 Primary Keys 生成 DDL (\$)</p> <p>Foreign Keys - 表示应生成外键的 DDL (\$)</p> <p>Indexes - 指示应生成索引的 DDL (\$)</p> <p>Unique约束- 表示应生成 Unique约束的 DDL (\$)</p> <p>选择约束- 表示应生成选择约束的 DDL (\$)</p> <p>库表触发器-表示应该生成DDL for库表触发器(\$)</p> <p>库表属性-表示应该生成扩展表的属性 (\$)</p> <p>Length Semantics - 表示应该为 Oracle Length Semantic 生成 DDL (\$)</p>
物件生成选项	<p>-视图- 表示应生成视图元素的 DDL (*)</p> <p>过程- 指示应生成过程元素的 DDL (*)</p> <p>函数- 表示应该生成函数元素的 DDL (*)</p> <p>序列- 指示应生成序列元素的 DDL (*)</p> <p>包- 表示应生成 DDL for Oracle包元素 (*)</p>
格式化	<p>包括前/后查询- 表示生成的 DDL 应该包括# '_PreStatements' 和 '_PostStatements' 中定义的语句# 查询</p> <p>Include Owners - 指示生成的 DDL 应包含所有元素的架构/所有者</p>

	<p>Include Comments - 表示生成的 DDL 应包含任何注释</p> <p>Include Header Comments - 指示生成的 DDL 应包含任何标题注释 (#)</p> <p>物件Comments - 表示生成的 DDL 应包含任何object (如库表或视图) 的注释 (#)</p> <p>Include Column Comments - 指示生成的 DDL 应包含任何列注释 (#)</p> <p>生成DROP statements - 表示生成的 DDL 应该包含对象的 DROP 语句</p> <p>使用数据库- 表示生成的 DDL 应该包含数据库使用语句</p> <p>使用别名- 指示生成的 DDL 使用任何object或列别名</p> <p>约束与库表分离- 表示生成的 DDL 应将约束的创建定义为与库表定义分开的语句</p> <p>在列定义中包含 NULL - 表示生成的 DDL 应将 NULL 关键字应用于定义为可为空的每个列定义; 也就是说, 未选中 "NOT NULL" 标志的列 (此选项仅适用于支持 "NULL" 语法的 DBMS)</p>
--	--

注记

- (*) - 如果您已指定为该类型的单个元素生成 DDL, 则带有此标记的选项将自动设置为 True; 也就是说, 如果您选择了一个库表并且您的 "生成库表" 选项为 False, Enterprise Architect 会将选项更改为 True
- (\$) - 如果 "表" 选项设置为 False, 带有此标记的选项将被禁用
- (#) - 如果 "包含评论" 选项设置为 False, 则带有此标记的选项将被禁用
- 在 Enterprise Architect 的企业版、统一版和终极版中, 如果启用了安全性, 则必须具有 "生成源代码和 DDL" 权限才能生成 DDL
- 对于 PostgreSQL 数据库, 您必须将 'Sequences' 选项设置为 True 以启用要创建的自动增量列
- 如果生成 Oracle 序列, 则必须始终将 "库表触发器" 和 "Sequences" 选项设置为 True, 以便生成预插入触发器以选择下一个序列值来填充列; 此外, 在列属性中, 将 "属性" 属性设置为 True
- 您可以编辑系统用来生成 DDL 的 DDL 模板; 这些存储在存储库级别, 以便同一存储库的所有其他用户将自动使用更新的模板

编辑 DDL 模板

DDL 模板编辑器提供了在从数据模型生成 DDL 时更改系统使用的模板的能力。它应用了公共代码编辑器的功能，包括各种宏的智能感知代码。有关智能感知信息和公共代码编辑器，请参阅编辑源代码帮助。

访问

功能区	开发 > 数据建模 > 模板
-----	----------------

选择和编辑模板

选项	行动
语	单击下拉箭头并选择数据库类型（数据库管理系统）。
新数据库	单击此按钮可为非标准 DBMS 创建一组新模板。 将显示“输入”对话框，您可以在其中键入要为其创建模板的新 DBMS 的名称。 这将更新“语言”字段。
模板	显示所选模板的内容，并提供修改这些内容的编辑器。
模板	列出基本 DDL 模板，单击模板名称可显示和编辑模板内容；所选模板的名称突出显示。 “修改”字段表示您是否修改了系统最初提供的默认模板。
构造型覆盖	列出当前选定的基本模板。 “Modified”字段指示您是否修改了默认的原型模板。
加新自定义模板	单击此按钮以显示“创建新的自定义模板”对话框，您可以在该对话框中从下拉列表中选择模板类型，然后输入模板的名称。 模板类型成为名称的前缀；例如： 命名空间_MyDDLTemplate
加新Stereotyped Override	选择一个基本模板并单击此按钮以显示“新模板覆盖”对话框，以便为所选模板模板从下拉列表中，选择要应用覆盖模板的类和/或特征原型。
获取默认模板	单击此按钮以使用所选模板的默认版本刷新编辑器显示。（这不会删除模板的更改版本。）
节省	单击此按钮以使用模板面板的更新内容覆盖选定的模板。
删除	如果你已经覆盖了选中的模板，点击这个按钮可以删除被覆盖的模板，并用相应的默认 DDL 模板替换它。

注记

- 用户修改和用户定义的参考可以作为模板数据导入和导出 (参见共享参考主题)
- 数据库类型的任何用户定义模板都列在 “参考代码、DDL、变换和 CSV 模板” 表中的 “导出数据” 对话框中，由带有后缀导出的 DBMS 名称标识 - 如果不存在用户定义模板对于 DBMS，对话框中没有 DBMS 条目
- 您还必须为 DBMS 定义任何适当的数据类型，如果将模板导出为参考，您还必须导出 “模型数据类型-代码和 DDL” 表

DDL 模板语法

DDL模板是使用Enterprise Architect的代码模板框架编写的，但它们已被扩展以支持 DDL 生成。

DDL 模板开发

本节将讨论 DDL 模板开发的这些方面。

方面	也见
DDL模板	DDL模板
DDL 宏	DDL 宏
DDL函数宏	DDL函数宏
DDL属性宏	DDL属性宏
模板中的 DDL 选项	模板中的 DDL 选项

DDL模板

DDL 模板编辑器的操作方式与代码模板编辑器相同，只是 DDL 模板编辑器同时显示用于 DDL 生成的模板和用于更改 DDL 生成的模板。Alter DDL Generation 模板显示在列表底部，前缀为“DDL Diff”。

用于 DDL 生成的基本模板

DDL 模板框架由许多用于 DDL 生成的基本模板组成。每个基本模板为UML数据模型的特定方面生成 DDL 语句（或部分语句）。

模板

此表列出并简要描述了用于 DDL 生成的基本模板。

模板	描述
DDL选择约束	由 DDL库表约束模板调用以生成库表object的选择约束声明。
DDL专栏注解	通常由 DDL 创建库表Extras模板为每个库表列生成COMMENT ON 语句（或等效语句）。
DDL 列定义	由众多模板调用以构建语句以创建单个库表列，就像它出现在 CREATE TABLE 语句中一样。
DDL 列附加功能	通常由 DDL 创建库表Extras模板调用以生成每个库表列的任何扩展列属性。
DDL约束栏名称	由每个约束模板调用以检索当前约束中涉及的正确格式的列名。
DDL约束注解	通常由 DDL 创建库表Extras模板调用以为每个库表约束生成 COMMENT ON 语句（或等效语句）。
DDL 创建外键	由 DDL 创建库表约束 为库表object生成外键约束的模板。
DDL 创建函数	由 DDL脚本文件调用 为函数object生成 CREATE FUNCTION 语句的模板。
DDL创建包	由 DDL脚本文件调用 用于为包object生成 CREATE PACKAGE 语句的模板（仅限 Oracle）。
DDL 创建过程	由 DDL脚本文件调用 用于为过程生成 CREATE object语句的模板。
DDL 创建架构	目前未使用。
DDL 创建序列	由 DDL脚本文件调用 用于为序列object生成 CREATE SEQUENCE 语句的模板。
DDL 创建库表	由 DDL脚本文件调用 为库表object生成 CREATE TABLE 语句的模板。
DDL 创建库表约束	由 DDL脚本文件调用 为库表object生成库表约束和索引的模板。
DDL 创建库表Extras	由 DDL脚本文件调用 为库表object生成扩展库表属性的模板。
DDL 创建视图	由 DDL脚本文件调用 为视图object生成 CREATE VIEW 语句的模板。
DDL 数据类型	由 DDL 列定义模板调用，为库表列生成格式正确的数据类型语句。
DDL 落柱附加功能	由 DDL Drop库表Extras模板调用，为列扩展属性生成任何专门的删除语句。

DDL 丢弃外键	由 DDL脚本文件调用 模板生成语句以删除库表object的所有外键。
DDL 丢弃函数	由 DDL脚本文件调用 为函数object生成 DROP FUNCTION 语句的模板。
DDL 下降过程	由 DDL脚本文件调用 用于为过程object生成 DROP PROCEDURE 语句的模板。
DDL 丢弃序列	由 DDL脚本文件调用 用于为序列object生成 DROP SEQUENCE 语句的模板。
DDL Drop库表	由 DDL脚本文件调用 为库表object生成 DROP TABLE 语句的模板。
DDL Drop库表Extras	由 DDL脚本文件模板调用以生成用于 DROP库表object的所有扩展属性的语句。
DDL 拖放视图	由 DDL脚本文件模板调用以生成视图object的视图VIEW 语句。
DDL 外国约束	由 DDL库表约束模板调用以生成库表object的 ADD FOREIGN KEY CONSTRAINT 语句。
DDL 补助金	由 DDL 创建库表Extras模板调用以生成当前object的 GRANT 语句 (仅限 Oracle) 。
DDL 索引	由 DDL库表约束模板调用以生成库表object的 CREATE INDEX 语句。
DDL 左环绕	用于定义用作object (或object组件) 名称左侧分隔符的字符 (或多个字符) 。
DDL名称	<p>大多数模板使用它来提供格式化object (或object特征) 名称的通用方式。这个模板接受四个参数：</p> <ul style="list-style-type: none"> • 物件位置 (值：EA 或 LIVE) • 物件类型 (取值：OWNER、TABLE、VIEW、PROCEDURE、FUNCTION、SEQUENCE、PACKAGE、COLUMN、CONSTRAINT、CONSTRAINT_COLUMN、REFERENCE_TABLE、REFERENCE_COLUMN) • 包括拥有着标志；控制名称是否应以拥有名称为前缀 (值：拥有着或 {blank}) • 包括环绕标志；控制名称是否应由左右环绕字符分隔 (值：INCLUDE_SURROUND 或 {blank})
DDL Primary约束	由 DDL库表约束模板调用以生成库表object的 ADD PRIMARY KEY CONSTRAINT 语句。
DDL参考列名称	通常由 DDL名称模板调用以检索外键中涉及的格式正确的引用列名称。
DDL参考定义	由调用 DDL 外国约束模板为外键约束生成 ON DELETE/ON UPDATE 语句。
DDL 右环绕	用于定义用作object (或object组件) 名称右侧分隔符的字符 (或多个字符) 。
DDL脚本文件	用于生成 DDL A顶级模板；所有其他模板都是从这个模板调用的。
DDL脚本头	由 DDL脚本文件调用 模板在每个 DDL 文件的开头添加标题注释。

DDL脚本Separator	由必须在生成的 DDL 中包含语句分隔符的所有模板使用。
DDL 语句团队	用于定义用作语句终止符的字符 (或多个字符) 。例如 , 对于大多数 DBMS , 分号 (;) 。
DDL 语句团队Alt	用于定义用作替代语句终止符的字符 (或多个字符) 。例如 , 某些 DBMS 必须更改语句终止符 , 以免导致为基于 SQL 的对象 (如视图和过程) 生成的 DDL 语句出现问题。
DDL 同义词	由 DDL 创建库表Extras模板调用以生成 CREATE SYNONYMS 语句 (仅限 Oracle) 。
DDL库表约束表	由 DDL 创建库表约束模板调用 , 为每个库表object生成库表约束和索引 , 同时考虑生成选项。
DDL库表级别注解	由 DDL 创建库表Extras模板调用以生成object的 COMMENT ON 语句 (或等效语句) 。
DDL触发器	由 DDL库表约束 为库表object生成 CREATE TRIGGER 语句的模板。
DDL Unique约束	由 DDL库表约束模板调用以生成库表object的 ADD UNIQUE CONSTRAINT 语句。
DDL使用数据库	由 DDL脚本文件模板调用以在每个 DDL 文件的开头包含一个 USE DATABASE 语句。

用于更改 DDL 生成的基本模板

DDL 模板框架由许多用于 Alter DDL 生成的基本模板组成。每个基本模板根据检测到的动作生成行动语句，这些动作必须用于同步数据模型和实时数据库。

模板

此表列出并简要描述了用于 Alter DDL 生成的基本模板。

模板	描述
DDL 差异列	Enterprise Architect为检测到的每个库表列差异直接调用。
DDL Diff约束	由Enterprise Architect为检测到的每个库表约束差异直接调用。
DDL Diff库表	Enterprise Architect为检测到的每个库表差异直接调用。
DDL 视图	Enterprise Architect针对检测到的每个视图差异直接调用。
DDL 差异过程	由Enterprise Architect为检测到的每个存储过程差异直接调用。
DDL 差异函数	由Enterprise Architect为检测到的每个函数差异直接调用。
DDL 差异序列	由Enterprise Architect为检测到的每个序列差异直接调用。

DDL 宏

字段替换宏提供对模型数据的访问。特别是，它们用于从以下位置访问数据字段：

- 数据库对象（如表和视图）
- 列
- 约束
- 约束专栏

字段替换宏根据 Camel 大小写命名。按照惯例，所有 DDL 宏都以 `ddl` 为前缀。

如果复选框/布尔值为真，则表示复选框或布尔值的宏返回 `string` 值 `"T"`。否则返回一个空 `string`。

内部字段宏 - 字段

`ddlAction` 宏是 `Alter DDL` 模板中可用的内部宏，提供对 Enterprise Architect 内部字段的直接访问；它没有直接映射到任何存储的数据。

`ddlAction` 表示将实时数据库与当前存储库同步所必须执行的操作。例如，`创建库表`、`删除库表`或 `更改拥有着`。

元素字段宏

此列表标识 DDL 模板中可用于访问元素级字段的宏，其中（在Enterprise Architect中）字段是可编辑的，例如“库表名称”和“库表别名”。

ddlFunctionAlias

函数 属性”对话框：主”选项卡：别名”文本域。

ddl函数名

函数 属性”对话框：名称”文本字段。

ddl所有者

{库表元素} 属性”对话框：{元素} 库表详细信息”选项卡：拥有着”文本字段。

ddlPackageAlias

'包'属性'对话框：'主'选项卡：'别名' 文本域。

ddlPackageName

'包'属性'对话框：'名称'文本字段。

ddlProcedureAlias

过程”属性对话框：主”选项卡：别名” 文本域。

ddlProcedureName

过程 属性”对话框：名称”文本字段。

ddlSchemaFunctionName

从实时数据库中读取的函数元素定义的名称。

ddlSchemaOwner

从实时数据库中读入的元素定义的 拥有着"属性"。

ddlSchemaProcedureName

从实时数据库中读取的过程元素定义的名称。

ddlSchemaSequenceName

从实时数据库中读取的序列元素定义的名称。

ddlSchemaTableName

从实时数据库中读入的 库表名称"属性"。

ddlSchemaViewName

从实时数据库中读取的元素视图定义的名称。

ddlSequenceAlias

“ 列"属性对话框： 主"选项卡： 别名"文本字段。

ddlSequenceName

序列 属性"对话框： 名称"文本字段。

ddlTableAlias

库表 属性"对话框： 主"选项卡： 别名" 文本域。

ddlTableDBMS

库表 属性"对话框： 主"选项卡： 数据库"下拉列表字段。

ddlTableLevelComment

库表 属性"对话框： 注记"文本字段。

ddlTableName

库表 属性”对话框：名称”文本字段。

ddlViewAlias

视图 属性”对话框：主”选项卡：别名”文本域。

ddlViewName

视图 属性”对话框：名称”文本字段。

列字段宏

此列表标识 DDL 模板中可用于访问列相关字段的宏，其中（在Enterprise Architect中）字段是可编辑的，例如“列名称”和“列别名”。

ddl列名

列和约束“对话框”：列“选项卡”：名称“单元”。

ddlColumnAlias

列和约束“对话框”：列“选项卡”：别名“单元”。

ddlColumnComment

列和约束“对话框”：列“选项卡”：注释“文本字段”。

ddlSchemaColumnName

从实时数据库中读入的列名称属性。

注记：此字段在Enterprise Architect中不可直接编辑。

约束字段Macros

此表列出了 DDL 模板中可用于访问约束相关字段的宏，其中（在Enterprise Architect中）这些字段是可编辑的，例如“约束类型名称”和“约束”。

ddlConstraintAlias

“Columns约束约束标签：别名”单元。

ddlConstraintColumnAlias

列和约束”对话框：约束”选项卡：涉及的列：已分配”列表。

ddlConstraintColumnName

列和约束”对话框：约束”选项卡：涉及的列：已分配”列表。

ddlConstraintComment

列和约束”对话框：约束”标签：注释”文本字段。

ddlConstraintName

列和约束”对话框：约束”标签：名称”单元。

ddlPKColumnCount

仅当当前约束具有主键类型时才相关，此宏将向主键返回分配列的计数。

列和约束”对话框：约束”选项卡：涉及的列：已分配”列表。

ddlReferenceColumnAlias

仅当当前约束具有外键类型时才相关，此宏将从引用表返回列别名。

专 和约束”对话框：约束”标签：别名”单元。

ddlReferenceColumnName

仅当当前约束具有外键类型时才相关，此宏将返回引用表中的列名。

外键 约束”对话框：涉及的列”列表：父”列。

ddlReferenceTableAlias

仅当当前约束具有外键类型时才相关。此宏将返回引用表的别名。
库表 属性”对话框：主”选项卡：别名”文本域。

ddlReferenceTableName

仅当当前约束具有外键类型时才相关。此宏将返回引用表的名称。
外键 约束”对话框：涉及的列”列表：父”列标题。

ddlReferenceTableOwner

仅当当前约束具有外键类型时才相关。此宏将返回引用表的所有者。
外键 约束”对话框：涉及的列”列表：父”列标题。

ddlSchemaConstraintColumnName

从实时数据库中读取的当前约束中涉及的列名。
注记：此字段在Enterprise Architect中不可直接编辑。

ddlSchemaConstraintName

从实时数据库中读取的约束名称属性。
注记：此字段在Enterprise Architect中不可直接编辑。

ddlSchemaConstraintType

约束类型属性从实时数据库中读取。
注记：此字段在Enterprise Architect中不可直接编辑。

DDL函数宏

DDL函数宏提供了一种方便的方式来操作、检索或格式化与 DDL 生成相关的元素数据。这些宏以及代码函数宏可用于 DDL 模板。每个函数宏返回一个结果string，并以与代码模板函数宏相同的方式使用。

此处描述了可用的函数宏。所有参数都有一个字符串类型，并用方括号表示；即：
FUNCTION_NAME([param])。

DDL_DATATYPE_SIZE ([产品名称] , [数据类型])

以 DDL 语法返回当前列的完全格式化的数据类型。

参数

- productName - 当前库表分配的 DBMS，如#
服务器2012，Oracle 或 PostgreSQL
- datatype - 当前列的数据类型名称，例如 VARCHAR 或 INT

评论

在Enterprise Architect库表列中，数据类型使用影响 DDL 语法的类型（属性1或2）定义；这个函数宏在构建返回值时会考虑长度类型（和其他因素）。

DDL_GET_DEFINITION_PARAS ([定义])

从提供的函数/过程定义中返回参数的string表示形式。

参数

- 定义 -完成#
过程/函数的定义

评论

一些 DBMS（如 PostgreSQL）支持同一过程/函数名的多个定义。这些定义的不同之处仅在于它们的参数列表，因此要操作此类对象，DDL 必须指定名称和参数。这个函数宏使 DDL 模板能够提取参数，以便随后可以使用它们来识别单个对象。

DDL_INCLUDE_SQLQUERY([objectName])

返回#

SQLQuery object中定义的语句。

参数

- objectName - #
的名称#
当前数据模型查询中定义的object

评论

没有任何。

DDL_INDEX_SORT ([产品] , [列])

返回给定索引的排序顺序。

参数

- 产品 - DBMS (目前 · 火鸟)
- columns - 索引中涉及的列名的 CSV

评论

该宏目前仅适用于火鸟索引。

DDL_RESOLVE_NAME ([productName], [name], [leftSurround], [rightSurround])

如果名称是当前 DBMS 的保留字，则返回提供的分隔名称（使用提供的左右字符）。

参数

- productName - 当前库表分配的 DBMS，如# 服务器2012，Oracle 或 PostgreSQL
- name - object /列名
- leftSurround - 用于围绕名称的左侧字符；例如，单引号 {}
- rightSurround - 用于围绕名称的右侧字符；例如，单引号 {}

评论

某些 DBMS 的 DDL 语法要求保留字的名称以不同的方式分隔；这个函数宏可用于安全地格式化 DB2 和火鸟的所有名称。

DDL_TABLE_TAGVALUE ([tagName])

返回当前库表的存储库版本中提供的标记名称的值。

参数

- tagName - 要检索的标记项的名称

评论

没有任何。

EXECUTE_CURRENT ([objectName], [actionName], [priority])

将当前模板的返回string添加到执行引擎的执行队列中。

参数

- 物件- 将在执行队列的“物件”列中显示的值，表示正在更新的object的名称
- 行动- 将在执行队列的“行动”列中显示的值，指示导致生成此语句的操作
- priority - 表示语句优先级的数值；数字越大，语句在队列中的位置越低

评论

这个函数宏可以在整个模板中的任何时候调用，但直到结束才会执行。一旦模板完成，它生成的 DDL 就会被发送到执行队列。

如果用户选择为文件生成 DDL，则此函数宏无效。

EXECUTE_STRING ([objectName], [actionName], [priority], [ddlStatement])

将提供的 DDL 语句添加到执行引擎的执行队列中。

参数

- 物件- 将在执行队列的“物件”列中显示的值，表示正在更新的object的名称
- 行动- 将在执行队列的“行动”列中显示的值，指示导致生成此语句的操作
- priority - 表示语句优先级的数值；数字越大，语句在队列中的位置越低
- ddlStatement - 执行所需操作的单个 DDL 语句

评论

如果用户选择为文件生成 DDL，则此函数宏无效。

EXIST_STRING ([ddlStatement])

在执行引擎的执行队列中搜索提供的 DDL 语句，如果找到该语句，则返回“T”。

参数

- ddlStatement - 单个 DDL 语句

评论

没有任何。

GET_FIRST_SQL_KEYWORD([语句])

返回提供的第一个关键字#

陈述。

参数

- 声明 - #
陈述

评论

没有任何。

ODBC_TABLE_TAGVALUE ([tagName])

返回当前表的实时数据库版本中提供的标记名称的值。

参数

- tagName - 要检索的标记项的名称

评论

没有任何。

PROCESS_DDL_SCRIPT ([type], [parameter2], [parameter3], [parameter4])

为特定目的返回格式化string A通用函数宏。

参数

- type - 指定要采取的特殊行动
- parameter2 - 通用参数 2，每种类型都有不同的用途

- parameter3 - 通用参数 3，每种类型都有不同的用途
- parameter4 - 通用参数 4，每种类型都有不同的用途

评论

对于 Oracle 同义词，请使用以下参数：

- 类型= 同义词”
- parameter2 =表名；例如，TBL_EMPLOYEES
- parameter3 = 一个分隔的string值，用分号分隔，指定同义词所有者和名称之间用全冒号；例如，OE:EMPLOYEES;PUBLIC:PUB_EMPLOYEES;
- parameter4 = 语句终止符

返回结果

格式：

为 TBL_EMPLOYEES 创建同义词 OE.EMPLOYEES ；

为 TBL_EMPLOYEES 创建公共同义词 PUB_EMPLOYEES ；

REMOVE_LAST_SEPARATOR ([ddlStatement], [分隔符])

返回提供的 DDL 语句，删除了最后一个分隔符（如果存在）。

参数

- ddlStatement - 部分 DDL 语句
- separator - 应删除的分隔符

评论

在构建表示 DDL 语句的string时，通常的做法是在每个项目之后附加分隔符；但是，最后一项之后不需要分隔符，因此提供了此函数宏来删除尾随分隔符。

REMOVE_STRING ([ddlStatement])

从执行引擎的执行队列中删除提供的 DDL 语句。

参数

- ddlStatement - 单个 DDL 语句

评论

没有任何。

SUPPRESS_EXECUTE_CURRENT ([boolean])

启用/禁用对 EXECUTE_CURRENT 的后续调用A函数宏。

参数

- 布尔值 - True或False

评论

此标志的默认状态为False；也就是说，对 EXECUTE_CURRENT 的调用不会被忽略。

DDL属性宏

DDL属性宏提供了一种检索元素属性值（即标记值）的便捷方式。在数据建模的范围内，有两组属性：

- 内部属性（Enterprise Architect在其比较中识别和使用的那些）和
- 自定义属性

这些属性宏提供对针对各种属性定义的属性的访问。所有属性宏具有相同的语法，返回一个string并要求指定属性的名称。

语法：`propertyMacroName:"propertyName"`

内部性质

tableBoolProperty:"propertyName"

返回当前库表的存储库版本中的内部属性值的布尔表示（“T”或“”）。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

表属性：属性名称”

返回当前库表的存储库版本中的内部属性值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

列属性：属性名称”

返回当前列的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

columnBoolProperty:"propertyName"

返回当前列的存储库版本中的内部属性值的布尔表示（“T”或“”）。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

约束属性：“属性名称”

返回当前约束的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

约束布尔属性：“属性名称”

返回当前约束的存储库版本中的内部属性值的布尔表示（“T”或””）。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

约束列属性：“属性名称”

返回当前约束列的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

`constraintColumnBoolProperty:"propertyName";`

返回当前约束列的存储库版本中的内部属性值的布尔表示（“T”或””）。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

视图属性：“属性名称”

返回当前视图的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

程序属性：“属性名称”

返回当前过程的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

功能属性：“属性名称”

返回当前函数的存储库版本中的内部属性值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

序列属性：“属性名称”

返回当前序列的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

包属性：“属性名称”

返回当前数据库包的存储库版本中的内部属性值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

`odbcTableProperty:“propertyName”;`

返回当前库表的 ODBC 版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

odbcConstraintProperty:"propertyName";

返回当前约束的 ODBC 版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

用户定义的属性

表用户属性： 属性名称”

返回当前库表的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

列用户属性： 属性名称”

返回当前列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束用户属性： 属性名称”

返回当前约束的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束列用户属性：属性名称”

返回当前约束列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

viewUserProperty:"propertyName";

返回当前视图的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

procedureUserProperty:"propertyName";

返回当前过程的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

函数用户属性：属性名称”

返回当前函数的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

序列用户属性：属性名称”

返回当前序列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

模板中的 DDL 选项

DDL 生成选项宏为 DDL 模板访问生成选项提供了一种方便的方法。

此列表标识并简要描述了每个可用的选项宏。每个选项都有一个值 'T' 代表 true 或一个空string代表false。

ddlGenerateToExecuteEngine

将生成的 DDL 定向到执行引擎。

ddlOptionColumn评论

在生成的 DDL 中包含列注释。

ddlOptionGenerateCheck

在生成的 DDL 中包含选择约束。

ddlOptionGenerateDrop

在生成的 DDL 中包含 DROP 语句。

ddlOptionGenerateForeign

在生成的 DDL 中包含外键。

ddlOptionGenerateFunction

在生成的 DDL 中包含函数。

ddlOptionGenerateIndex

在生成的 DDL 中包含索引。

ddlOptionGenerateLength语义

(仅限 Oracle) 在生成的 DDL 中的文本列上包含长度语义语法。

ddlOptionGenerateNullable

如果在生成的 DDL 中没有将其标记为 NOT NULL 列，则针对每列包含关键字 NULL。

ddlOptionGeneratePackage

(仅限 Oracle) 在生成的 DDL 中包含包。

ddlOptionGeneratePrimary

在生成的 DDL 中包含主键约束。

ddlOptionGenerateProcedure

将过程包含在生成的 DDL 中。

ddlOptionGenerateSeparateConstraint

生成库表约束分别对 CREATE TABLE 语句；也就是说，使用 ALTER TABLE 语句。
注记：一些 DBMS 不支持所有条件下的单独约束。

ddlOptionGenerateSequence

在生成的 DDL 中包含序列。

ddlOptionGenerateTable

在生成的 DDL 中包含表。

ddlOptionGenerateTableProperty

在生成的 DDL 中包含扩展属性表。

ddlOptionGenerateTrigger

在生成的触发器中包含简单的库表。

ddlOptionGenerateUnique

在生成的 DDL 中包含唯一约束。

ddlOptionGenerateView

在生成的 DDL 中包含视图。

ddlOptionHeaderComments

在生成的 DDL 中包含标题注释。

ddlOptionTable评论

在生成的 DDL 中包含库表注释。

ddlOptionUseAlias

使用别名而不是生成DDL 屏幕上指定的所有对象（object组件）的名称。

ddlOptionUseDatabaseName

在每个生成的文件的开头包含 USE DATABASE 语句。

ddlUseAlias

使用别名而不是数据库生成器“数据库比较”选项卡上指定的所有对象（object组件）的名称。

DDL 限制

数据库管理系统 (DBMS) 的基本特征是允许通过结构化语言定义数据库对象；这种语言称为 DDL (数据定义语言, 或数据描述语言)。每个 DBMS 的 DDL 语法都是唯一的。虽然在所有 DBMS 中都有通用的 DDL 语句和关键字, 但存在一些差异, 要求每个 DBMS 在 Enterprise Architect 中有自己的一组 DDL 模板。

本页总结了每个受支持的数据库管理系统的主要限制。

MS Access

- 评论不能应用于 (或更改) 表、库表列、库表约束或视图, 因此 Enterprise Architect 忽略这些差异
- CREATE TABLE 语句不支持列默认值的定义, 因此 Enterprise Architect 将默认定义从所有生成的 DDL 中排除; 但是, 它确实突出了比较逻辑中的默认差异
- 通常 DDL 中的 object 名称可以用方括号 ([]) 括起来, 以便它们可以包含空格和其他非标准字符, 但是 CREATE VIEW DDL 语句不支持方括号表示法; “创建视图” DDL 模板用下划线 ('_') 字符替换所有空格

MySQL

- 只有 MySQL 版本大于 5.5.3 时, 才能对 Indexes 和 Unique 约束应用注释
- 评论只能在创建时应用于索引和唯一约束, 因此更改索引或唯一约束的评论会导致约束被删除并重新创建
- 不支持选择约束; 虽然 MySQL DDL 引擎可以解析此类语句, 但它只是忽略它们
- 评论不能应用于 (或更改) 视图、过程或函数, 因此 Enterprise Architect 忽略这些差异

甲骨文

- 注释不能应用于 (或更改) 过程、序列或函数, 因此 Enterprise Architect 忽略这些差异

PostgreSQL

- 目前 Enterprise Architect 不支持函数参数, 因此任何通过名称引用函数的语句 (COMMENT ON 或 DROP) 都会失败, 因为它们必须使用函数名称和参数的组合

#

精简版

- 约束不能添加到现有的库表中; 必须删除和创建库表 (包括创建语句中的新约束)
- 任何 object 类型都不支持注释, 因此 Enterprise Architect 忽略所有注释差异

导入数据库架构

基于模型的工程的力量在于可视化、分析和设计系统各个方面的能力。能够与系统的其他模型一起查看数据库模式提供了极大的清晰度并减少了出错的机会。Enterprise Architect可以将DBMS模式及其对象逆向工程为多种不同标准下的模型，包括UML、信息工程和IDEF 1X。支持范围广泛A数据库对象，包括表、视图、过程和函数。Enterprise Architect通过询问DBMS的信息模式并将定义导入UML对象来实现这一点。随着对Live数据库的修改，可以将更改同步到模型中。



一旦架构在Enterprise Architect中，数据库对象就可以追溯到其他元素，确保设计和架构的完整性。当系统以多个DBMS为目标时，这些都可以逆向工程到模型中，并且可以在这些模型之间比较元素和数据类型。复杂的报告引擎可以生成高质量的文档，包括数据字典、图表和关系，返回到其他模型，如架构和信息需求，并最终返回到业务目标和驱动程序。

可以通过数据库生成器（推荐）或“开发”功能区导入数据库架构信息。

导入数据库架构

节	行动
1	打开数据库生成器（开发 > 数据建模 > 数据库生成器）
2	加载或创建数据模型。
3	<ul style="list-style-type: none"> • 右键单击数据库构建器中加载的数据模型并选择“导入数据库模式”或 • 从功能区中选择“开发 > 数据建模 > 导入” 将显示“导入数据库架构”对话框，显示当前活动数据库连接的详细信息。

导入数据库架构对话框

选项	描述
数据库	此字段显示当前实时连接的描述，格式为： <code>dbms.database_server.database_name</code> 如有必要，单击  按钮并选择替代连接。
导入到	此字段显示将保存新对象的目标包。 如果要指定不同的包，请单击  按钮并选择替代包。
仅包括来自架构的对象	如果数据库类型支持多种模式（如# 服务器、Oracle、PostgreSQL和DB2 Express），您可以按模式过滤要从数据库中检索的对象。 此面板中会自动列出可用的模式。选中要包含在导入中的每个架构的复选框。 （您可以单击“全部”按钮选择所有模式，或单击“无”按钮清除所有选中的复选框。） 如果您怀疑模式列表在加载后可能已更改，则可以通过单击“重新加载模式”

	”按钮刷新列表。
名称过滤器	<p>名称过滤器：“字段允许使用# 过滤对象# 适用于正在导入的模式的 DBMS 的通配符。 例如，对于 Oracle：</p> <ul style="list-style-type: none"> • LIKE 'A%' - 列出名称以字母'A'开头的对象 • NOT LIKE '%_%' ESCAPE '\' - 列出名称不包含下划线(_)的对象 • IN ('TABLE1','TABLE2') - 列出括号中包含名称的对象 • NOT IN ('TABLE1','TABLE2') - 列出名称未包含在括号中的对象 <p>注记只能输入一个过滤器。您不能使用 AND 子句添加第二个过滤器。 过滤不适用于MS Access</p>
过滤器选项	<p>过滤器选项”面板控制从数据库模式中读取的object类型和属性。在此屏幕上更改的值会保存到注册表中，以便在下一个工作会话中重新应用。此处简要描述了可用的选项；选中选项对应的复选框以激活它。</p> <p>表</p> <ul style="list-style-type: none"> • 表- 选择导入表 • 库表Primary Keys - 选择在表上导入主键定义 • 库表外键 - 选择在表上导入外键定义 • 库表索引 - 选择导入库表索引 • 唯一约束- 选择导入表上的唯一约束定义 • 选择约束- 选择导入表上的选择约束定义 • 库表触发器- 选择导入触发器定义表 • 库表属性-选择导入扩展库表属性 • 约束属性- 选择为表导入约束属性 • 长度语义 - 选择以在 Oracle string列上导入长度语义定义 <p>对象</p> <ul style="list-style-type: none"> • 视图- 选择导入视图 • 过程- 选择导入过程 <ul style="list-style-type: none"> - 作为操作- 选择导入过程作为单个类的操作（方法）；您可以通过数据库object容器“属性”对话框查看和编辑它们（该选项默认为到未选择，其中选定的项目作为单独的类导入） • 函数- 选择导入函数 <ul style="list-style-type: none"> - 作为操作- 选择导入函数作为操作（默认为未选中） • 序列 - 选择以导入序列 <ul style="list-style-type: none"> - 作为操作- 选择将序列导入为操作（默认为未选中） • 包-选择导入Oracle包 <p>高级</p> <ul style="list-style-type: none"> • 系统对象 - 选择导入系统表、视图等系统对象 <p>警告：使用过程函数和序列的“作为操作”选项，如果对象已在一个设置下导入（选中或未选中），然后您更改设置并导入更多对象，则在第一个设置下导入的对象将被删除。</p>
同步	<p>选择适当的单选按钮以指示是否要更新现有的类，或者将数据库对象作为新对象导入。</p> <p>如果您选择“同步现有类”选项，还请选择适当的复选框以确定模型注释、列默认值和/或库表约束是保留还是被导入对象的注释、值和约束覆盖。</p>

导入到	<p>选择适当的单选按钮以指示是更新包和当前打开的数据模型图，还是只更新包。</p> <p>如果没有打开图表，则“仅包”单选按钮默认为选中，并且选项被禁用；如果打开的图表在选定的包中，您可以选择任一选项。</p>
导入	<p>单击此按钮开始导入。</p> <p>将显示“选择要导入的数据库对象”对话框，列出找到的所有符合选择条件的数据库对象。</p> <p>选中每个架构（或object类型）的复选框以自动选择该组中的所有对象或单独导入每个object。</p> <p>单击全部按钮以选择所有类型和对象，或单击无按钮以清除所有选定的复选框。</p> <p>选择所有要导入的对象后，单击确定按钮继续导入。</p>

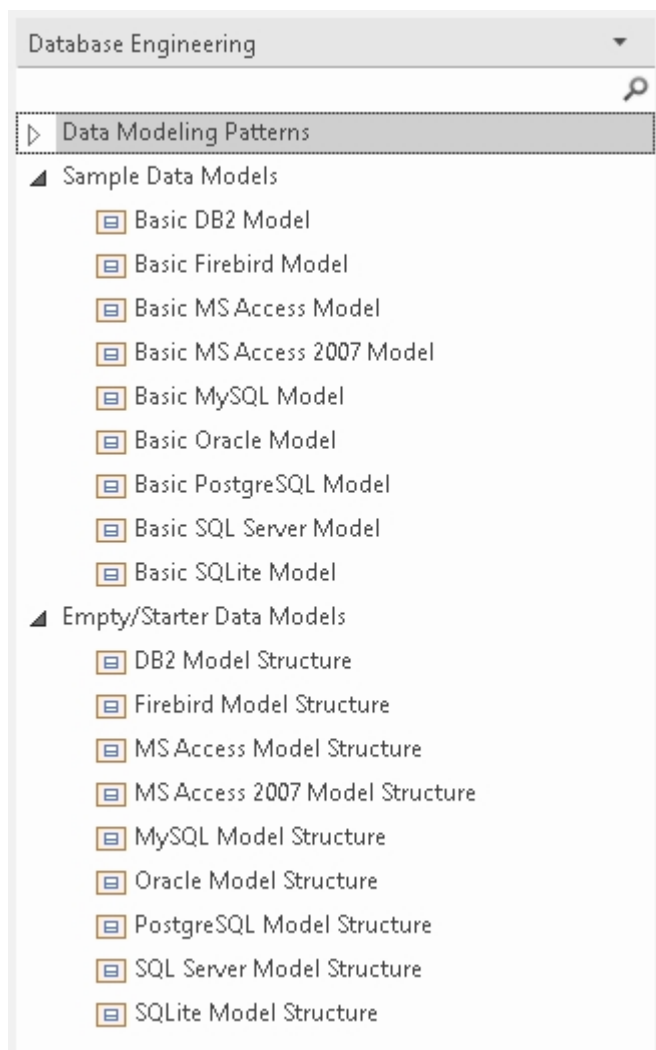
注记

- 在窗口中，可以为 32 位或 64 位应用程序定义 ODBC DSN，因此必须注意确保为Enterprise Architect使用的所有 ODBC DSN 定义共享相同的架构。从Enterprise Architect版本 16 开始，这一点尤其重要，因为它现在可用于 32 位和 64 位版本。另一种解决方案（也是Sparx Systems推荐的）是使用本地连接，因为它们适用于两种架构。
- ODBC 连接应使用 DBMS 供应商提供的 ODBC 驱动程序，例如MySQL的MySQL ODBC 驱动程序和 Oracle 的 Oracle ODBC 驱动程序；不支持第三方供应商提供的驱动程序，包括适用于 Oracle 的 Microsoft ODBC 驱动程序
- 您可以从<http://www.ch-werner.de/sqliteodbc/>为 SQLite 导入合适的 ODBC 驱动程序
- 由于 SQLite 的限制，不能实现 SQLite库表和列注释的round；要在从 ODBC 导入时保留在 SQLite 数据模型中输入的注释，请取消选中“从 ODBC源导入数据库架构”对话框的“同步”部分中的“覆盖物件注释”复选框
- 如果为逆向工程设置 ODBC 连接，默认设置就足够了
- 数据建模数据类型列表被定义为静态数据（在每个存储库中），因此根据存储库的年龄，“资源”的“数据建模数据类型”部分可能会提供其他数据类型Sparx Systems网站上的页面

创建和管理数据模型

Enterprise Architect是一个功能齐全的数据库建模平台，使用户能够在所有阶段使用他们的物理数据模型，从设计到实时数据库的实施，适用于各种数据库管理系统，例如 Microsoft #

服务器、甲骨文、PostgreSQL 和MySQL。



此图显示了用于一系列 RDBMS 的数据库设计的启动模型向导模式。

数据建模设置

Enterprise Architect提供数据建模设置，可用于配置数据库系统在Enterprise Architect中建模的方式。其中包括定义数据建模语言的能力，这决定了连接器的显示方式，以及配置主键、外键和索引命名的设置。这些设置是全局的，将影响任何Enterprise Architect存储库。

访问

功能区	开始>外观>首选项>首选项>源代码工程>代码编辑器>DDL
-----	-------------------------------

DDL 编辑器

在此字段中，您可以浏览Enterprise Architect用于打开由其生成DDL 功能创建的文件的外部程序的完整执行文件路径和名称。如果您将此字段留空，Enterprise Architect将使用默认代码编辑器。

默认数据库

在此字段中，您选择将自动分配给在数据模型工作空间之外创建的数据库对象的帮助（请参阅模式模型模型主题创建数据）。

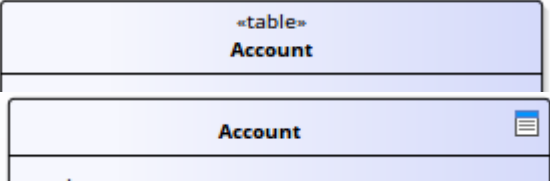
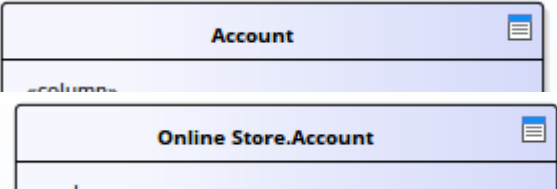
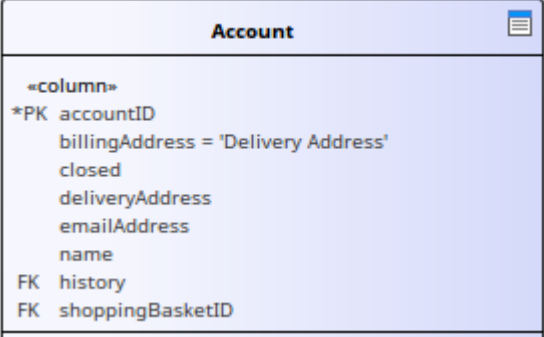
MySQL储存

在此字段中，您选择要分配给MySQL的默认存储引擎表；从MySQL v 5.5 开始，默认值为 InnoDB。

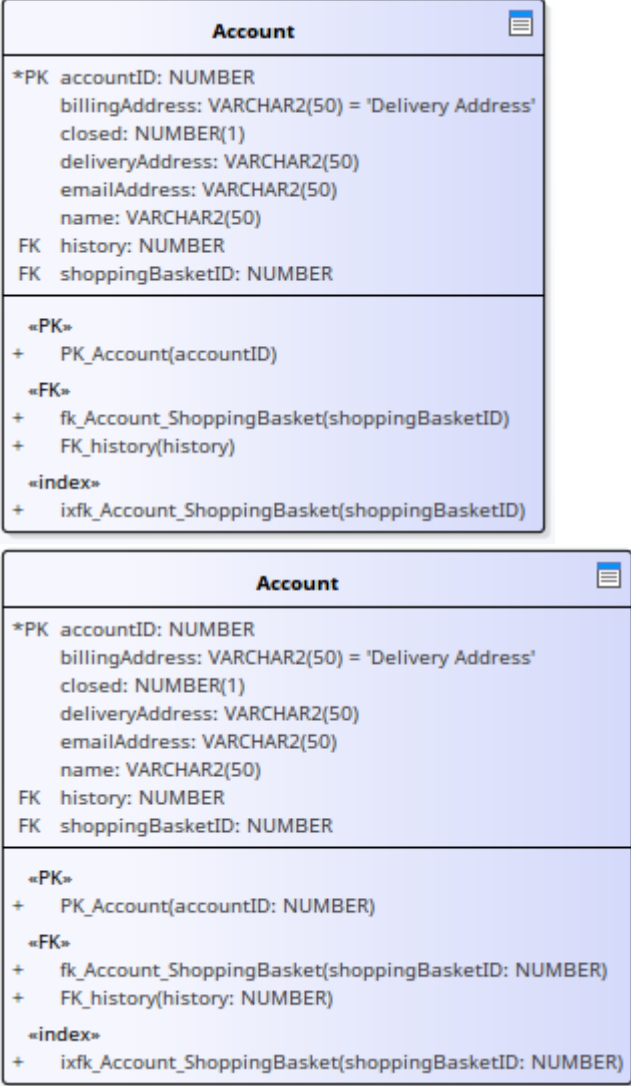
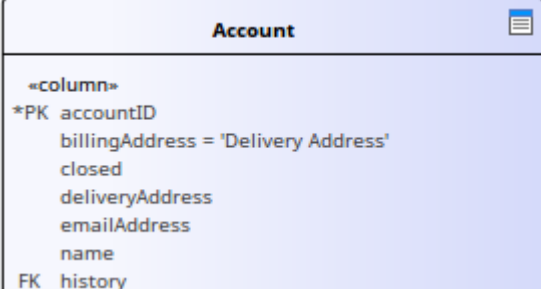
数据建模Notations

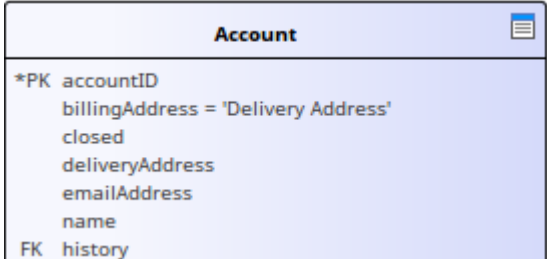
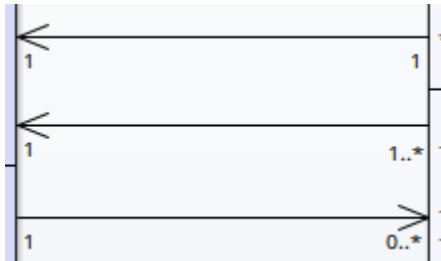
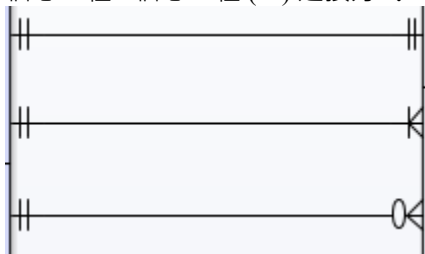
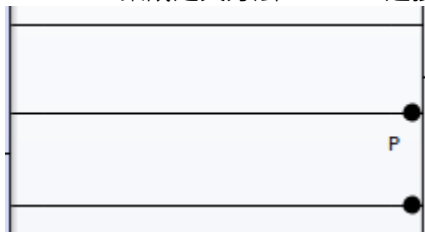
Enterprise Architect支持许多与数据建模相关的设置，这些设置会影响数据库对象在图表上的表示方式。此处描述了这些设置以及它们如何影响数据库对象的表示。

设置

环境	细节
构造型图标	<p>访问：'设计>图表>管理>属性>元素：使用构造型图标'</p> <p>默认值：True</p> <p>Enterprise Architect为原型对象的显示提供了图表级别的设置。当复选框被选中时，图表上的数据库对象将显示一个图标，代表它们的原型而不是原型名称。</p> 
显示数据模型拥有着	<p>访问：'设计>图表>管理>属性>元素：显示数据模型拥有着'</p> <p>默认值：True</p> <p>系统为所有者的显示提供了图表级别的设置。当复选框被选中时，当前图表上的数据库对象将以其全名 '{拥有着.}ObjectName' 显示。</p> 
显示列详细信息	<p>访问：'设计>图表>管理>属性>特征：显示属性详细信息'</p> <p>默认值：名称仅</p> <p>系统为库表列名和数据类型的显示提供了图表级设置。可用选项有：'仅名称' 或 '名称和类型'。</p> 

	<div data-bbox="520 192 1067 528"> <p style="text-align: center;">Account</p> <hr/> <p>«column» *PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> </div>
<p>显示涉及的列详细信息</p>	<p>访问：'设计>图表>管理>属性>特征 显示参数详情' 默认值：仅类型 系统为库表约束中涉及的列的显示提供了图表级别的设置。可用选项有：“无”、“仅类型”、“仅名称”和“完整详细信息”。 在这些示例中，主键 (PK) 约束 PK_account“涉及列 accountID”。</p> <div data-bbox="520 770 1153 1339"> <p style="text-align: center;">Account</p> <hr/> <p>*PK accountID billingAddress = 'Delivery Address' closed deliveryAddress emailAddress name FK history FK shoppingBasketID</p> <hr/> <p>«PK» + PK_Account() «FK» + fk_Account_ShoppingBasket() + FK_history() «index» + ixfk_Account_ShoppingBasket()</p> </div> <div data-bbox="520 1352 1067 1888"> <p style="text-align: center;">Account</p> <hr/> <p>*PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> <hr/> <p>«PK» + PK_Account(NUMBER) «FK» + fk_Account_ShoppingBasket(NUMBER) + FK_history(NUMBER) «index» + ixfk_Account_ShoppingBasket(NUMBER)</p> </div> <p style="text-align: right;">列2.png</p>


	 <p>列3.png</p>
<p>显示列构造型</p>	<p>访问：开始> Application > Preferences > Preferences > Objects: Show <<column>> stereotype</p> <p>默认值：True</p> <p>Enterprise Architect提供了一个全局级别的设置来控制<<column>> 原型是否显示在每个库表的列上方。因此，如果您愿意，可以隐藏构造型，考虑到构造型为 <<column>> 的属性是表的唯一有效选项。</p> 

	
<p>连接器表示法</p>	<p>访问：'设计>图表>管理>属性>连接器：连接器符号'</p> <p>默认值：UML 2.1</p> <p>Enterprise Architect支持数据建模的三种图表符号：</p> <ul style="list-style-type: none"> UML 2.1 - 用于连接器的标准UML 2.1符号  <ul style="list-style-type: none"> 信息工程 - 信息工程 (IE) 连接方式  <ul style="list-style-type: none"> IDEF1X - 集成定义方法 IDEF1X 连接样式  <p>(这些是使用不同符号的相同三个连接器。) 数据建模模型图的默认符号是信息工程”，而从工程创建的模型的默认模式是“数据库”。</p>

DDL名称模板

在数据建模过程中的不同时期，Enterprise Architect都需要自动生成库表约束。这些生成的约束的命名标准在DDL名称模板中定义和应用，您可以随时自由更改。这些名称模板是在存储库级别定义的，因此无论何时更改它们，存储库的所有用户都将使用新模板。

访问

功能区	设置>模型>选项>源代码工程：DDL名称模板 
-----	--

DDL名称模板

选项	行动
首要的关键	定义创建主键约束时使用的名称模板。
独特约束	定义创建唯一约束时使用的名称模板。
外键	定义创建外键约束时使用的名称模板。
外键索引	定义创建外键索引时使用的名称模板。
节省	单击此按钮以保存您定义的名称模板。

模板宏

在创建约束名称期间，这些已识别的宏将被名称文本替换。

宏观	适用于
%表名%	首要的关键 独特约束 描述：由库表名称替换的string。
%列名%	由约束的列名替换的string。
%主表名%	外键 描述：被主（父）库表名称替换的string。
%foreigntablename%	由外部（子）库表名称替换的string。
%外键名%	外键索引

	描述：被外键名称替换的string。
--	--------------------

数据类型

您在数据模型中定义的每个库表列都分配了一个数据类型，该数据类型指定了该列可以存储的信息类型。列的可用数据类型取决于为库表选择的 DBMS，因为每个 DBMS 都支持自己的数据类型列表。虽然每个 DBMS 都支持相同的基本类型，例如 string、整数或十进制数，但每个 DBMS 以不同的名称调用它们并具有不同的属性。

每个 Enterprise Architect 存储库都包含许多标准 DBMS 产品的核心数据类型的定义。然而，由于数据类型从一个 DBMS 产品到另一个，从一个产品版本到另一个版本，Enterprise Architect 为您提供了以下工具：

- 为新版本的 DBMS 产品定义新数据类型
- 为新的非标准数据库产品定义数据类型
- 自动将数据类型从一种已定义的 DBMS 产品转换为另一种
- 在存储库之间导入和导出数据类型

DBMS 产品之间的映射数据类型

虽然建模物理数据模型提供了关于所有表及其列的大量细节，但这一级别的细节确实使更改目标技术或平台变得更加困难。例如，在将数据库逆向工程为物理数据模型之后，您必须在为新的 DBMS 产品生成模式之前重新映射数据类型。

Enterprise Architect 为标准的、受支持的 DBMS 产品提供了一组默认映射，以帮助您自动化转换过程。

但是，您可能希望自定义默认映射以满足您的特定项目要求，或者当前未定义一种数据类型到另一种数据类型的映射。例如，在从一个 DBMS 平台迁移到另一个平台时，其中一个平台可能是非标准的，或者 Enterprise Architect 不支持。

访问

功能区	设置>参考>设置>数据库数据类型：Datatype 映射
-----	------------------------------

数据库数据类型映射

对每个要映射的数据类型重复此过程。

一旦您对数据类型映射感到满意，您就可以将单个表或整个表包转换为新的目标 DBMS 产品。

字段/按钮	行动
从产品名称	单击下拉箭头并选择要从中映射数据类型的 DBMS 产品。
为数据库定义的数据类型	显示产品的所有已定义数据类型，并在适当的情况下显示它们的大小和值。 单击要映射的数据类型 - 这必须具有定义的大小单位和值。 来自产品名称"字段下的 "Datatype" 和 "公共类型" 字段显示此数据类型。
至产品名称	单击下拉箭头并选择要将数据类型映射到的 DBMS 产品。 此字段下的 "Datatype" 和 "公共类型" 字段显示 "发件人" 产品的字段中的值相对应的值。
尺寸	单击相应尺寸单位的单选按钮并在相应的数据字段中键入默认值。
节省	单击此按钮以保存映射。

DBMS 产品转换为一个包

使用 DBMS 包映射器，您可以自动将数据库表包从一种支持的 DBMS 类型转换为另一种支持的 DBMS 类型。您还可以更改单个表的 DBMS 类型。

如果其中一种 DBMS 类型是非标准的，或者 Enterprise Architect 不支持，您应该检查数据类型从一种 DBMS 类型到另一种的映射是否已经定义。

访问

功能区	设计>包>管理> DBMS 选项 开发>数据建模>数据库生成器>在需要的数据库上右击 加载 右键单击根 管理 DBMS 选项
-----	--

将一个包的 DBMS 数据类型映射到另一个 DBMS 的数据类型

字段/按钮	行动
在子包中包含对象	如果子包中的对象也需要更改，请选中该复选框。
更改数据库管理系统	选中复选框。
当前的数据库管理系统	单击下拉箭头并选择当前 DBMS。
新的数据库管理系统	单击下拉箭头并选择目标 DBMS。
确定	单击此按钮将所选包中的所有表映射到新的 DBMS。

库表的数据类型转换

一旦在Enterprise Architect图表上设置了数据库模式（通过 ODBC 导入或手动设置表），可以将 DBMS 更改为另一种类型，并为每个库表相应地映射列数据类型。

如果您从其他地方将少量表复制到项目中，则可以使用此过程，但如果您有很多表，您可以在其父包中一次转换所有表。

如果其中一种 DBMS 类型是非标准的，或者Enterprise Architect不支持，您应该检查数据类型从一种 DBMS 类型到另一种的映射是否已经定义。

将一个库表的 DBMS 类型映射到另一个 DBMS 类型

节	行动
1	双击图表中的库表元素。 库表 属性”对话框显示，“数据库”字段显示此库表的当前 DBMS。
2	要将数据类型映射到另一个 DBMS，请单击 数据库”下拉箭头并选择目标 DBMS。
3	单击应用按钮。 转换数据类型以匹配新 DBMS 的数据类型，这些数据类型反映在从该库表生成的任何 DDL 中。

数据库数据类型

使用Enterprise Architect的“数据库数据类型”对话框，您可以添加可用于特定 DBMS 的数据类型集。您可以：

- 识别正在使用的 DBMS，如果需要，将其设置为模型默认值
- 包括由更高版本的 DBMS 支持但尚未包含在Enterprise Architect中的任何新数据类型
- 删除任何先前添加的不再相关的数据类型
- 添加一个新的 DBMS 产品及其内置数据类型，例如，如果您想为Enterprise Architect本身不支持的 DBMS 产品创建物理数据模型

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型或 开发 > 数据建模 > 数据类型
-----	---

管理数据类型

您可以使用“导出参考”和“导入参考”选项在Enterprise Architect模型之间传输这些数据库数据类型。

字段/按钮	行动
产品名称	单击下拉箭头并选择现有的 DBMS。 选择产品后，所有已定义的数据类型将显示在“已定义的数据库数据类型”列表中。
添加产品	如果您的 DBMS 未列出，请单击此按钮添加它。 将显示“输入”提示，您可以在其中键入 DBMS 名称；单击确定按钮将名称添加到下拉列表中。
设为默认	选中复选框以将选定的 DBMS 设置为数据库工程和建模的默认值。 设置默认数据库后，当您创建任何新的库表元素时，数据库类型会自动预设为此默认值。 您还可以在代码生成工具栏的第二个数据输入字段中设置默认数据库类型。
新的	单击此按钮可清除对话框中的数据类型字段，以便您可以定义其他数据类型。
Datatype	类型数据类型的名称。
尺寸	为所需大小选择适当的单选按钮，并在适当的情况下指定默认值和最大值： <ul style="list-style-type: none"> • 无 - 对于没有大小组件的数据类型，例如 INT • 长度 - 对于需要定义长度的单一大小的数据类型，例如 VARCHAR(10) • Precision & Scale - 适用于需要两个数值的数据类型，例如 DECIMAL(18,2)
公共类型	单击下拉箭头并选择每种数据类型的通用名称。这在库表的 DBMS 发生更改

	时使用。
节省	单击按钮立即将您的数据类型保存到存储库（并将其添加到“为数据库定义的数据类型”列表中）。
为数据库定义的数据类型	此面板列出当前为选定 DBMS 定义的数据类型，系统提供的或用户定义的。
删除	在“Defined Datatypes for Databases”列表中选择一种数据类型，然后单击此按钮以删除该数据类型。
Datatype映射	如果您已将已定义数据类型的 DBMS 或技术更改为不受支持的 DBMS 类型或更改为不受支持的 DBMS 类型，请单击此按钮以定义如何将数据类型自动重新映射到新的 DBMS 或技术。

MySQL数据类型

MySQL支持 ENUM 和 SET 数据类型，必须将它们添加到Enterprise Architect模型中，然后才能将它们用作列的类型。

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型
-----	------------------------

为MySQL添加 ENUM 和 SET 数据类型

稍后在列的“初始”字段中使用这些数据类型时，将值键入为逗号分隔的列表，格式如下：

('一二三')

如果一个值是默认值，请使用以下格式：

('一;'二;'三') 默认'三'

节	行动
1	将显示“数据库数据类型”对话框。
2	在“产品名称”字段中选择“MySQL”。
3	添加数据类型 ENUM 和 SET。

Oracle数据类型

Oracle 数据类型 NUMBER 和 VARCHAR 具有您可以模型的附加属性。

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型
-----	------------------------

数据类型

数据类型	细节
数字	<p>NUMBER 数据类型需要精度和比例属性。</p> <p>当数据类型设置为NUMBER时，“Precision”和“Scale”字段显示在特征窗口的“Attributes”页面上；如果您在这些字段中输入信息，它会显示在您的图表上。</p> <p>例如：</p> <p>通过设置 'Precision' = 0 和 'Scale' = 0 创建 NUMBER</p> <p>通过设置 'Precision' = 8 和 'Scale' = 0 创建 NUMBER(8)</p> <p>通过设置 'Precision' = 8 和 'Scale' = 2 创建 NUMBER(8,2)</p>
VARCHAR	<p>Oracle VARCHAR2(15 CHAR) 和 VARCHAR2(50 BYTE) 数据类型可以通过添加标记值LengthType 值CHAR 或BYTE 来创建。</p>

管理 DBMS 选项

使用“管理 DBMS 选项”对话框，您可以快速更改单个数据库object或单个包或包层次结构中的多个对象的 DBMS类型和/或拥有者。您还可以在所有还没有索引的外键上创建批量外键索引。

访问

功能区	设计>包>管理> DBMS选项 开发>数据建模>数据库生成器>在需要的数据库上右击 加载 右键单击根 管理 DBMS 选项
-----	---

选项

选项	行动
包	在您当前正在处理的浏览器窗口中显示包的名称。如有必要，单击  按钮并使用导航员窗口（“查找包”对话框的一个版本）选择不同的包。
在子包中包含对象	选中此复选框以在所有子包中包含所有数据库对象。选择或取消选择此控件将立即刷新对象列表。
对象列表	此列表控件将显示当前包（或包层次结构）中所有对象的名称及其分配的 DBMS 和所有者。默认情况下，每当加载或刷新列表时，每个object都会选中其复选框。
全部	单击此按钮可选择“对象列表”中所有取消选中的复选框。
没有任何	单击此按钮可取消选择“对象列表”中所有选中的复选框。
更改数据库管理系统	如果要更改包中对象的分配 DBMS，请选中此复选框。为“当前 DBMS”和“新 DBMS”字段提供值以继续。“当前 DBMS”下拉列表包括选项“<All>”，它将几个不同的 DBMS 值全部更改为新值。 注记：执行此函数时，所有库表列的数据类型会自动转换为与所选 DBMS 最接近的匹配；因此，您应该在运行该过程后对数据类型进行手动审阅。
更改拥有者	如果要更改“对象列表”中选定对象的拥有者，请选中此复选框。在“当前拥有者拥有者”栏内指定当前拥有者以继续。将“新拥有者”字段留空将删除所有选定对象的拥有者属性。
在外键上创建索引	选中此复选框可在包中的所有外键上创建一个索引，其中一个尚不存在。
确定	单击此按钮开始更新过程。除非选择了列表中的至少一个object和其中一个更新选项，否则该按钮将被禁用。

数据库对象

虽然表是关系数据库的基本组件并允许定义列、数据类型、键和索引，但在 RDBM 系统中还有许多其他重要的对象，包括：

- 视图-视图表示预定义查询的结果集；它们是从存储在一个或多个表（或其他视图）中的数据动态得出的
- 过程-一些 DBMS 产品实施以提供可以包含一个或多个例程的特征#
执行特定任务的语句，例如数据验证、访问控制或减少客户端和 DBMS 服务器之间的网络流量
- 函数——一些特征产品实现的特性，用于提供扩展数据库服务器功能的机制；each 是一个例程，可以接受参数、执行一个动作（例如复杂的计算）并将该动作的结果作为值返回
- 序列-一些 DBMS 产品实现的特征，以提供生成唯一值的机制 - 确保对它的每次调用都返回唯一值

UML 本身并没有指定如何执行数据建模，但 Enterprise Architect 具有用于数据建模的完全集成的 UML 配置文件以及一系列内置于核心产品中的特征，这些特征将使数据建模变得容易。

该配置文件使用构造型和标记值将标准 UML 元素扩展为数据建模结构。这是通过将数据库 object 型添加到 UML 类来实现的；这样您就可以模型：

- 数据建模 diagrams as extended UML 类 diagrams
- 表作为具有 <<table>> 构造型的 UML 类对象
- 视图作为具有 <<view>> 构造型的 UML 类对象
- 过程作为具有 <<procedure>> 构造型的 UML 类对象
- 函数作为具有 <<function>> 构造型的 UML 类对象
- 序列作为具有 <<dbsequence>> 构造型的 UML 类对象

您可以使用 Enterprise Architect 在您的数据库模型中快速创建和配置所有这些对象。

数据库视图

数据库视图表示预定义查询A结果。与库表不同，视图是从一个或多个表（或其他视图）中的数据动态得出的。Enterprise Architect支持视图的定义，视图有没有这个语句：

创建视图{viewName} As"声

每当执行 DDL 生成时，系统将自动动态添加（如果缺少）。不定义此语句的优点是，当视图object被重命名时，视图定义"属性不必手动更新。


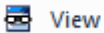
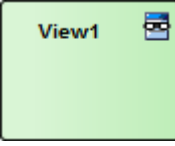
您可以创建数据库视图：

- 在数据库生成器中或
- 通过将数据建模工具箱中的 视图"图标拖到图表上

使用数据库生成器添加数据库视图

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击 视图"包并选择 加新视图"。
4	用视图的适当名称改写默认名称，然后按 Enter 键。
5	双击新视图，或右键单击它并选择' # 物件属性"。 ' # 显示 物件编辑器"对话框。

将数据库视图添加到图表

节	行动
1	打开您的数据建模图，如果需要，显示  工具箱数据建模图表工具箱项"对话框并指定 数据建模") 。
2	将 视图"图标拖到图表上。  View 这会生成视图元素： 

3	右键单击新的视图元素并选择'# 物件属性'。 '# 显示 物件编辑器'对话框。
---	--

物件编辑

'#

多个基于 SQL 的数据库对象 (视图、过程、函数和序列) 共享 物件编辑器"对话框；它帮助数据建模者管理基于 SQL 的object的各种属性。

选项	行动
数据库	如果已设置，则显示默认数据库类型。 如果未设置默认值，或者您想更改此视图的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。
依赖项	当前object所依赖A对象列表。 依赖项"列表显示： <ul style="list-style-type: none"> • 此视图与另一个库表或视图之间的每个视图连接器 • 父母"标记值中的任何object名称 (指定为 CSV 列表)
注记	如有必要，请在当前视图上输入评论。
定义	类型全# 视图定义。对于高达 12.1 (编译1227) 的Enterprise Architect版本，这必须包括适用于目标 DBMS 的 CREATE_VIEW 语法 (对于更高版本，这不需)。例如： 创建视图 'MyViewName' 为 [查看定义] 代码编辑智能感知为基本的# 当前数据模型中所有对象的关键字、功能和名称。

数据库连接

数据库object提供了A方便的方式来存储实时数据库的连接详细信息。Enterprise Architect支持定义许多不同的连接类型：

- MS Access
- 火鸟
- SQLite (在Enterprise Architect v16 中引入)
- 本机连接 (在Enterprise Architect v16 中引入) , 以及
- ODBC

对于基于文件的连接 (MS Access、火鸟和 SQLite) , 您只需指定数据库文件的完整路径。对于本机连接, 系统将提示您输入数据库服务器的连接详细信息。对于 ODBC 类型的连接, 系统会提示您从计算机上的预定义 ODBC DSN 列表中进行选择。

创建数据库连接元素

A数据库元素在UML数据建模配置文件中表示为 < 一个工件元素与立体连接>>。您可以创建这些：

- 在数据库生成器中或
- 通过将“数据库连接”图标从“数据建模”工具箱页面拖到图表上

使用数据库添加数据库连接

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击 Connections包并选择“加新DB Connection”。
4	用适当的 Connection 名称改写默认名称, 然后按 Enter 键。
5	双击新连接, 或右键单击它并选择“数据库连接属性”。 将显示“数据库连接属性”对话框。

将数据库连接添加到图表

节	行动
1	打开您的数据建模图, 如有必要, 显示  “工具箱数据建模图表查找工具箱项”对话框并指定“数据建模”。
2	将“数据库连接”图标拖到图表上。

	 <p>Database Connection 这会生成数据库连接元素。</p>
3	<p>双击新元素。 将显示“数据库连接属性”对话框。</p>

连接数据库属性

选项	行动
DBMS类型	<p>单击相应类型的单选按钮：</p> <ul style="list-style-type: none"> • MS Access基于文件的数据库 • 基于火鸟文件的数据库 • 基于 SQLite 文件的数据库 • 直接本机连接，或 • 基于 ODBC 的数据库 <p>保存密码？“复选框仅对 ODBC 连接类型启用，并指示Enterprise Architect是否应存储所选 ODBC DSN 的密码。复选框默认选中；即，保存密码。虽然所有连接密码在保存之前都已加密，但有时数据建模者希望将访问权限限制为仅具有所需权限的用户。</p>
文件名/DSN	<p>如果您选择了MS Access或火鸟的“DBMS类型”，请输入或浏览物理文件的位置和名称。如果该文件尚不存在，它将被创建。</p> <p>如果您选择了 ODBC 的“DBMS类型”，请输入或选择定义的 ODBC DSN。根据 DBMS，可能会提示您输入其他详细信息，例如服务器、连接用户 ID 和密码。</p>
其它模式	<p>此字段充当模式过滤器，以限制对 ODBC 连接进行的查询返回的对象数。由于大量的系统对象，在此字段中输入一个值对于 Oracle 数据库来说尤其重要，因为它可以减少连接到数据库所需的时间。</p> <p>如果您需要输入多个要过滤的模式，请用逗号分隔它们。</p>
确定	<p>单击此按钮以保存您所做的更改。</p>

删除连接

如果不再需要连接，您可以将其删除，就像从数据库构建器、浏览器窗口或图表中删除任何其他元素一样。右键单击元素并选择相应的“删除<元素名称>”选项。

注记

- 建议在团队环境中工作（即，多个用户共享一个Enterprise Architect存储库）时，所有基于 ODBC 的数据库连接对象都定义为“无 DSN”，以便数据库连接object包含所有必要的详细信息，因此可以在所有用户之间共享，虽然本机连接这样做并且更容易设置
- 保存初始选择后，无法更改数据库连接object的 DBMS 类型

甲骨文包

Oracle包是 Oracle DBMS 独有的数据库对象。它们是将逻辑相关的对象组合成一个定义的容器。包有两个部分——规范和主体。这：

- 规范部分声明了各种组件
- 体部分提供了组件的完整定义

包组件可以由类型、变量、常量、异常、游标和子程序组成。

在Enterprise Architect中，Oracle包被建模为具有 <<package>> 构造型的UML类。它有两个操作：

- 规格
- 体

对于这些操作中的每一个，完成#

语法包含在“初始代码”字段中。

创建 Oracle包

节	行动
1	将类元素添加到您的数据模型中。
2	打开元素的属性窗口，并在“构造型”字段中键入值“包”。
3	单击元素并按 F10，在“操作”页面显示特征窗口。 对于包规范，按 Ctrl+N 并创建一个名为“规范”且没有返回类型的操作。
4	属性窗口显示操作的属性；单击“代码”选项卡，然后在文本面板中输入整个包规范。
5	返回“操作”页面的特征窗口，对于包体，按 Ctrl+N 并创建一个名称为“体”且没有返回类型的操作。
6	在属性窗口上，单击“代码”选项卡，然后在文本面板中输入整个包体代码。

创建操作容器

虽然建模数据库函数、过程和序列的默认方法是将它们创建为单独的元素，但您也可以将每种类型的结构的数量表示为容器类的操作。您向类添加一个构造型，它指定：

- 类将包含的数据结构类型
- 将自动分配给类中创建的每个操作的构造型（对于给定的数据结构，操作只能是一个构造型）

访问

工具箱	将“类”图标拖到图表上
-----	-------------

创建容器类

节	行动
1	右键单击图表上的类元素并选择 设计>元素>编辑对话框>属性”选项。显示元素 属性”对话框，显示 常规”选项卡。
2	在 名称”字段中，输入容器的适当名称。
3	在 构造型”字段（在对话框右边缘的表中）输入： <ul style="list-style-type: none"> • 数据库函数容器的 函数” • 存储过程容器的 程序” • 序列容器的 序列” 原型名称末尾的 \$”很重要。
4	单击确定按钮保存设置并关闭对话框。

创建数据库结构作为类的操作

节	行动
1	单击图表上的类元素，然后按 F10。将显示 数据库<Structure> 容器：<Classname>”对话框。
2	右键单击 函数”（ 过程”或 “ 列”）列表并选择 加新<结构>”。
3	在 名称”字段中，为操作键入适当的名称，例如： <ul style="list-style-type: none"> • fn_WorkDays • sp_AddOrder 或

	<ul style="list-style-type: none">• seq_AddressID
4	在“注记”字段中输入任何支持性评论或操作说明。 在“函数定义”字段（或“过程定义”或“序列定义”）中键入适当的文本。
5	重复步骤 2 到 4，直到您定义了所需的所有操作。
6	单击列表，然后单击关闭按钮关闭对话框并在图表和浏览器窗口中显示类中的操作。

#数据库#

查询

一个#

查询object提供了一种方便的机制来存储#

存储库中的语句，用于在实时数据库上重复执行。

一个#

在UML元素数据建模配置文件中以工件数据建模配置文件为代表的<查询元素查询>>。您可以创建这些元素：

- 在数据库生成器中或
- 通过拖动'#
数据查询数据建模工具'工具箱页面中的 图标"到图表中


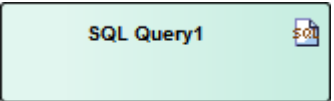
添加数据库#

使用数据库生成器查询

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击查询包并选择'加新# '查询'。
4	用查询的适当名称改写默认名称，然后按 Enter 键。
5	右键单击新元素并选择 编辑”。 将显示 'SQL便签本'选项卡，您可以在其上创建# 查询语句。
6	当你完成# 声明，点击保存到# 工具栏中的按钮以保存对查询元素的更改。

将数据库函数添加到图表

节	行动
1	打开您的数据建模图，如果需要，显示  '工具箱数据建模图表工具箱项'对话框并指定 '数据建模')。

2	<p>拖动' # '查询'图标到图表上。</p>  <p>这会生成# 查询工件元素</p> 
3	<p>双击新元素并根据需要更新元素名称和其他属性。</p> <p>编辑元素的# 语句，访问数据库构建器，单击查询包中的元素并在“SQL便签本”选项卡上编辑查询。</p>

数据库序列

序列是一些 DBMS 产品实现的特征，旨在为用户提供一种生成唯一值的机制 - 确保对它的每次调用都返回一个唯一值。当序列的结果用作主键时，这一点尤其重要。这些可以使用模式生成以加载到 DBMS 服务器上。

提供了序列，以便数据库用户不会被迫实现他们自己的唯一值生成器。并非所有 DBMS 产品都支持序列；那些不提供使用递增值初始化列的功能。

在Enterprise Architect中，可以通过以下两种方式之一对序列进行建模：

- 作为单个对象（默认方法）或
- 在容器中操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象

建模为单个对象的序列是具有构造型 «dbsequence» 的UML类；您可以创建这些：

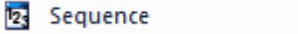
- 在数据库生成器中或
- 通过将“数据建模”工具箱页面中的“序列”图标拖到图表上

使用数据库构建器添加数据库序列

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击序列包并选择“加新序列”。
4	用适当的序列名称改写默认名称，然后按 Enter 键。
5	双击新序列，或右键单击它并选择“# 物件属性”。 '# 显示“物件编辑器”对话框。

将数据库序列添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱项”对话框并指定“数据建模”。

2	<p>将 序列”图标拖到图表上。</p>  <p>这会生成序列元素：</p> 
3	<p>右键单击新的序列元素并选择' # 物件属性”。</p> <p>' #</p> <p>显示 物件编辑器”对话框。</p>

物件编辑

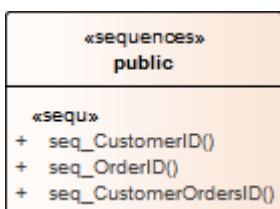
' #

多个基于 SQL 的数据库对象（视图、过程和函数）共享 物件编辑器”对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	<p>如果已设置，则显示默认数据库类型。</p> <p>如果未设置默认值，或者您想更改此序列的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。</p>
注记	<p>如有必要，请在当前序列上输入注释。</p>
定义	<p>类型全#</p> <p>包括 CREATE SEQUENCE 语法的序列定义。</p> <p>代码编辑智能感知为基本的#</p> <p>当前数据模型中所有对象的关键字、功能和名称。</p>

在容器中操作

数据库Sequences 建模为操作有一个容器object，这是一个具有构造型«sequences»（末尾带有's'）的UML类。每个序列都是一个具有刻板印象«sequ»的操作。该系统提供了一个专用的维护窗口，建模者可以通过该窗口轻松管理定义为操作的序列。



函数数据库

数据库函数为您提供了一种扩展数据库服务器功能的机制。数据库函数是接受参数、执行操作（例如复杂计算）并将该操作的结果作为值返回的例程。根据函数，返回值可以是单个值或结果集。

创建后，数据库函数可以用作#
中的表达式#
陈述。

在Enterprise Architect中，可以通过以下两种方式之一对数据库函数进行建模：

- 作为单个对象（默认方法）或
- 在容器中操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象


建模为单个对象的数据库函数是具有构造型 «function» 的UML类；您可以创建这些：

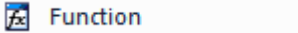
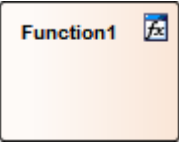
- 在数据库生成器中或
- 通过将函数图标从数据建模工具箱拖到图表上

使用数据库添加数据库函数

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击函数包并选择“加新函数”。
4	用函数的适当名称改写默认名称，然后按 Enter 键。
5	双击新函数，或右键单击它并选择“# 物件属性”。 # 显示物件编辑器屏幕。

将数据库函数添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱”对话框并指定“数据建模”。

2	<p>将 函数"图标拖到图表上。</p>  <p>这会生成函数元素：</p> 
3	<p>右键单击新的函数元素并选择' # 物件属性"。</p> <p>#</p> <p>显示物件编辑器屏幕。</p>

物件编辑

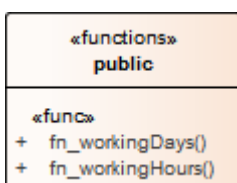
' #

多个基于 SQL 的数据库对象（视图、过程和函数）共享“物件编辑器”对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	<p>如果已设置，则显示默认数据库类型。</p> <p>如果未设置默认值，或者您想更改此函数的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。</p>
注记	<p>如有必要，请输入对当前函数的评论。</p>
定义	<p>类型全#</p> <p>函数定义包括 CREATE FUNCTION 语法。</p> <p>代码编辑智能感知为基本的#</p> <p>当前数据模型中所有对象的关键字、功能和名称。</p>

在容器中操作

被建模为操作的数据库函数有一个容器 object，这是一个具有构造型«functions»（末尾带有's'）的UML类。每个函数都是一个具有构造型«func»的操作。系统提供专门的维护窗口，您可以方便地管理存储为操作的数据库函数。



数据库过程

数据库过程（有时称为存储过程或过程）是可以包含一个或多个的子例程#

执行特定任务的语句。它们可用于数据验证、访问控制或减少客户端和 DBMS 服务器之间的网络流量。广泛而复杂的业务逻辑可以嵌入到子程序中，从而提供更好的性能。

数据库过程类似于数据库函数。主要区别在于调用它们的方式 - 数据库函数的使用方式与#

中的任何其他表达式相同#

语句，而数据库过程必须使用 CALL 或 EXEC 语句调用，具体取决于 DBMS。

在Enterprise Architect中，数据库过程可以通过以下两种方式之一进行建模：

- 作为单个对象（默认方法）或

作为容器中的操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象

建模为单个对象的数据库过程是具有构造型 «procedure» 的UML类；您可以创建这些：


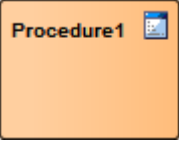
- 在数据库生成器中或
- 通过将数据建模工具箱中的“过程”图标拖到图表上

使用数据库添加数据库过程

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击过程包并选择“加新过程”。
4	用适当的过程名称改写默认名称，然后按 Enter 键。
5	双击新过程，或右键单击它并选择“# 物件属性”。 # 显示物件编辑器屏幕。

将数据库过程添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱项”对话框并指定 数

	据建模”)。
2	<p>将 过程”图标拖到图表上。</p>  <p>这会生成过程元素：</p> 
3	<p>右键单击新的过程元素并选择' # 物件属性”。</p> <p>#</p> <p>显示物件编辑器屏幕。</p>

物件编辑

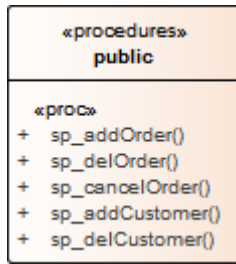
' #

多个基于 SQL 的数据库对象 (视图、过程和函数) 共享 物件编辑器”对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	<p>如果已设置，则显示默认数据库类型。</p> <p>如果未设置默认值，或者您想更改此过程的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。</p>
笔记	<p>如有必要，请输入对当前过程的评论。</p>
定义	<p>类型全#</p> <p>过程定义，包括 CREATE PROCEDURE 语法。</p> <p>代码编辑智能感知为基本的#</p> <p>当前数据模型中所有对象的关键字、功能和名称。</p>

在容器中操作

被建模为操作的数据库过程有一个容器 object，这是一个具有原型«procedures» (末尾带有's') 的UML类。每个数据库过程都是一个具有构造型«proc»的操作。系统提供专门的过程维护窗口，您可以通过该窗口过程管理定义为操作的数据库。



数据库表

表是关系数据库的基本组件，表示多行结构化数据元素（称为列）。输入到关系数据库中的每一项数据都由列中的值表示。

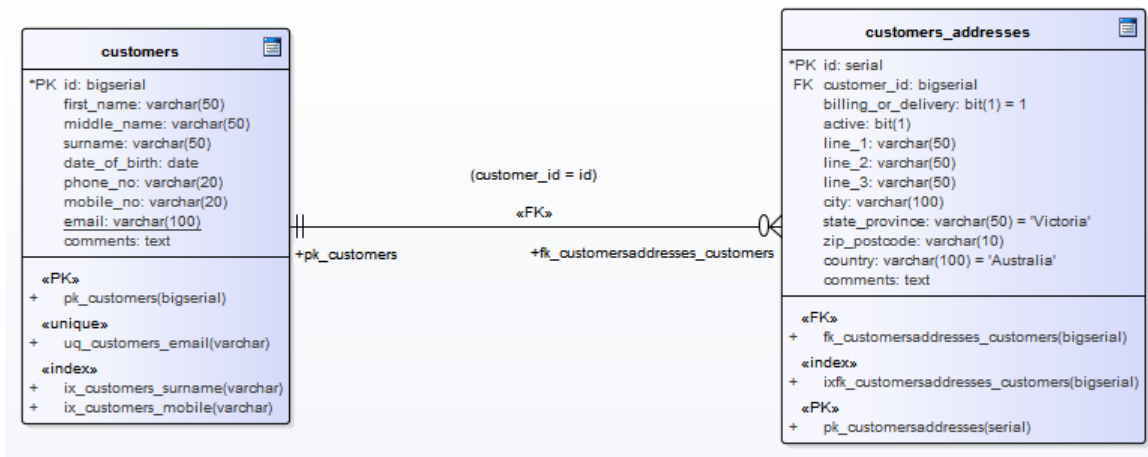
Enterprise Architect的用于数据建模的UML配置文件表示：

- 数据库表作为具有 <<table>> 构造型的UML类对象
- 库表列作为库表的UML属性，库表型为 <<column>>
- 主键作为库表的UML操作/方法，具有 <<PK>> 的构造型
- 外键作为库表的UML操作/方法，具有 <<FK>> 的构造型
- 索引作为库表的UML操作/方法，具有 <<index>> 的构造型
- Unique约束作为库表的UML操作/方法，具有 <<unique>> 的刻板印象
- 选择约束作为库表的UML操作/方法，具有 <<check>> 的刻板印象
- 库表作为库表表的UML操作/方法，具有 <<trigger>> 的触发器型

Enterprise Architect将库表的所有UML操作统称为约束，因此您用于维护库表的UML属性和操作的屏幕称为列和约束屏幕。

示例

Enterprise Architect中的物理数据模型图的这个简单示例由两个由UML类表示的数据库表组成，名为 *customers* 和 *customer_addresses*。



每个库表定义数据库列，使用为目标 DBMS（在本例中为 PostgreSQL）适当键入的UML属性。

笔记

- 库表stereotype由每个类右上角的图标表示（参见数据建模 *Notation* 主题）
- 用于管理库表列的Enterprise Architect维护屏幕不允许您更改属性构造型，因为 <<column>> 是唯一有效的选项
- 可以隐藏示例表中显示的 <<column>> 构造型标签（参见数据建模注释主题）

数据库约束库表/索引

在Enterprise Architect中，库表约束和索引在同一屏幕上建模；它们统称为约束。数据库约束定义了强加于数据库行为的条件库表。他们包括：

- Primary Key - 唯一标识库表中的一条记录，由一列或多列组成
- 索引 - 提高对库表数据的检索和排序操作的性能
- 唯一约束- 唯一标识库表中一行的值的组合
- 外键 - 强制两个表之间的关系（或列集合）
- 选择约束- 通过限制列接受的值来强制域完整性
- 库表触发器- #
或由于库表中的数据被修改而自动执行的代码

在Enterprise Architect中，您可以使用数据库构建器中专门设计的“约束/索引”页面或列和约束屏幕来定义和维护库表约束。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

创建任何这些约束类型的过程是相同的，并且可以通过此处描述的一种方式来实现。

创建一个约束-使用上下文菜单或键盘

节	行动
1	A新的约束被自动创建并分配了默认名称约束 <i>n</i> （其中 <i>n</i> 是一个计数器）和一个“索引”的类型”。用您自己的约束名称改写默认名称。
2	如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。 “列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

创建一个约束-改写模板文本

节	行动
1	在所选库表的“约束/索引”选项卡上，约束列表以模板文本新约束结束。用适当的约束名称改写此文本，然后按 Enter 键。
2	新约束会自动创建并分配默认类型索引。如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。“列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

将列分配给约束

Primary Key、Foreign Key、Index、Unique 这些约束类型都必须至少分配一列；这定义了约束中涉及的列。

节	行动
1	在所选库表的“约束/索引”选项卡上，单击要为其分配列的约束。
2	“可用列”面板列出了为库表定义的所有列。对于要分配给约束的每一列，右键单击列名称并选择“分配列 <名称>”。列名被传送到“分配的列”列表中。

从约束中取消分配列

节	行动
1	在所选库表的“约束/索引”选项卡上，单击要取消分配列的约束。
2	在“分配的列”列表中，右键单击要从约束中取消分配的列的名称，然后选择“取消分配列 <名称>”。列名被转移到“可用列”列表中。

重新排序约束中的指定列

如果约束中有许多列，则可以通过将选定的列名一次在列表中向上或向下移动一个位置来重新排列序列。去做这个：

- 右键单击要移动的列名称并选择：
 - 向上移动列“<名称>”（Ctrl+向上箭头）或
 - 向下移动列“<name>”（Ctrl+向下箭头）

删除约束

要删除不再需要的约束，请右键单击“约束/索引”选项卡上列表中的约束名称，然后选择“删除约束 <名称>”选项。如果满足给定约束类型的所有验证规则，则会立即从存储库中删除该约束以及所有相关关系（如果有的话）。

主键

主键是唯一标识库表中每条记录A列（或列集）。A库表只能有一个主键。一些 DBMS 支持主键的附加属性，例如 Clustered 或 Fill Factor。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称
上下文菜单	图中 右键单击库表 特征 约束/索引

创建主键

在Enterprise Architect中，您可以从“列”选项卡或“约束/索引”选项卡创建主键。无论哪种情况，当您将列添加到主键约束时，该列都会自动设置为“非空”。此外，任何包含库表元素的图表（假设设置了“显示限定词和可见性指示器”选项）都将针对列名显示“PK”前缀。在此图像中，请参见第一列“id: bigserial”。

customers

***PK** id: bigserial
first_name: varchar(50)
middle_name: varchar(50)
surname: varchar(50)
date_of_birth: date
phone_no: varchar(20)
mobile_no: varchar(20)
email: varchar(100)
comments: text

«PK»
+ pk_customers(id: bigserial)

«unique»
+ uq_customers_email(email: varchar)

«index»
+ ix_customers_surname(surname: varchar)
+ ix_customers_mobile(mobile_no: varchar)

创建主键 - 从列选项卡

节	行动
---	----

1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击具有一个或多个已定义列的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	<p>对于要包含在主键中的每一列，选择 PK”复选框。</p> <p>如果之前没有为当前库表定义主键约束，系统将使用主键名称模板创建一个新约束。</p>

创建主键 - 从约束选项卡

节	行动
1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击包含一个或多个已定义列的库表，然后单击 约束/索引”选项卡，或 在图表上，单击库表并按 F10 以显示 约束/索引”选项卡
2	<p>用主键名称改写新约束文本，按 Enter 键并单击 类型”字段下拉箭头，然后选择 PK”。</p>
3	<p>将所需的列分配给 PK 约束。</p>
4	<p>使用属性面板设置主键的扩展属性。</p> <ul style="list-style-type: none"> 填充因子是 0 到 100 之间的数值 Is Clustered 是一个布尔值，用于确定数据存储方式的物理顺序；对于大多数 DBMS，主键的 Is Clustered 属性默认为 True

从主键中删除列

您可以使用 列”选项卡或 约束/索引”选项卡从主键中删除列。

从主键中删除列 - 使用 列”选项卡

节	行动
1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击带有主键的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	<p>针对要从主键中删除的每一列，取消选中 PK”复选框。</p> <p>如果您已从主键约束中删除所有列并且不再需要主键，则必须手动删除它。</p>

从主键中删除列 - 使用约束/索引选项卡

节	行动
1	任何一个： <ul style="list-style-type: none">在数据库构建器中，单击带有主键的库表，然后单击“约束/索引”选项卡，或在图表上，单击库表并按 F10 以显示“约束/索引”选项卡
2	根据需要取消分配 PK 约束上的列。

注记

- 警告：Enterprise Architect假定主键约束至少分配了一列；但是，Enterprise Architect在建模期间不强制执行此规则
如果为主键未分配列的库表生成 DDL，则该 DDL 无效

非聚集主键

当您在某些 DBMS 中创建主键时 (例如#

服务器或ASA)，它会在 'Is Clustered'属性设置为True的情况下自动创建。因此，当您在Enterprise Architect数据模型中建立主键模型，会发生相同的行为。

聚集索引通过按列物理组织数据，为访问所涉及的列提供了改进的性能。每个库表只能有一个聚集索引。

在某些情况下，您可能对分配给主键的列之外的列的性能更感兴趣，因此您需要更改默认分配，以便主键不聚集。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	在图表或浏览器窗口 右键单击库表 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引

将主键定义为非集群

随后，您可以将模型库表的索引建模为集群。

节	行动
1	突出显示现有的主键约束。 主键属性显示在“属性”面板中。
2	对于 <i>Is Clustered</i> 属性，在“值”字段中单击下拉箭头并将值更改为 False。

数据库索引

将数据库索引应用于表以提高数据检索和排序操作的性能。一个库表可以定义多个索引；但是，每个索引都会在数据库服务器上增加开销（以处理时间和存储的形式）来维护它们，因为在库表中添加和删除信息

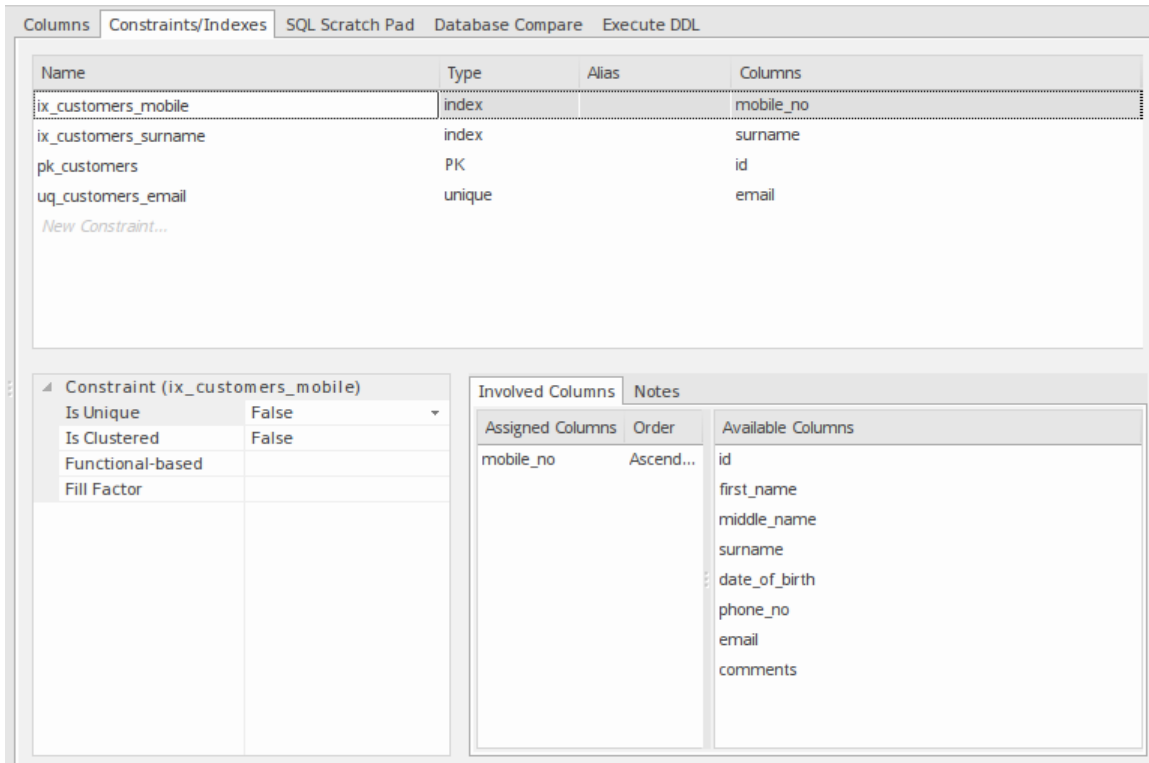
在Enterprise Architect中，索引被建模为原型操作。

一些 DBMS 支持特殊类型的索引； Enterprise Architect使用其他属性来定义这些，例如基于函数的、集群的和填充因子。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	图中 右键单击库表 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引

处理索引



节	行动
1	在库表的“约束/索引”选项卡上，右键单击并选择“添加新约束”。 使用默认名称“constraint1”和“index”类型添加新约束。 用您喜欢的索引名称改写名称。

2	<p>将适当的列分配给索引。</p> <p>'Assigned Columns' 列表有一个附加的 'Order' 字段，用于指定每个已分配列存储在索引中的顺序（升序或降序）。您可以根据需要切换每列的顺序。</p> <p>此外，对于MySQL索引，“Len”字段将可见，您可以在其中定义部分索引；即，使用基于文本的字段的前导 <i>n</i> 个字符的索引。“Len”字段仅采用介于 0 和列的定义长度之间的整数数值。A 0（这是默认值）表示要索引整个列。</p>
3	<p>在“属性”面板中，审阅为当前 DBMS 定义的扩展属性的设置。</p>

附加属性

属性	描述
是独特的	<p>(True / False) 指示当前索引是否为 唯一索引”。唯一索引确保索引A（或列）不包含重复值，从而确保每一行具有唯一值（或索引由多列组成时的值组合）。</p>
是集群的	<p>(True / False) 指示当前索引是否为 聚集索引”。使用聚集索引，表的行在物理上以与索引中相同的顺序存储，因此每个表只能有一个聚集索引。默认情况下，一个表的主键是集群的。</p> <p>并非所有 DBMS 都支持聚集索引，因此 Is Clustered”索引属性仅对支持它的 DBMS 可见。</p>
是位图	<p>(True / False) 指示当前索引是否为 位图”索引。位图索引旨在用于具有相对较少唯一值的列（称为 低基数”列）并且物理上由每个唯一值的位数组（通常称为位图）组成。对于表中的每一行，每个数组都有一个位。</p> <p>考虑这个示例：在名为 性别”的列上创建位图索引，该列具有 男性”或 女性”选项。物理上，索引将由两个位数组组成，一个用于 男性”，一个用于 女性”。女性位数组在匹配行的值为 女性”的每个位中都有一个1。</p> <p>Is Bitmap 和 Is Unique属性是互斥的，所以当 Is Bitmap属性为True时，DDL 生成将忽略 Is属性。</p> <p>位图索引仅受 Oracle 支持；因此，此属性仅在对 Oracle 索引建模时可见。</p>
填充系数	<p>A介于 0 和 100 之间的数值，用于定义应用于数据的可用空间百分比。</p> <p>并非所有 DBMS 都支持填充因子，因此 填充因子”索引属性仅对支持它的 DBMS 可见。</p>
基于功能	<p>A #</p> <p>定义将被评估的函数/语句和索引结果的语句；例如： 降低（“字段”）</p> <p>并非所有 DBMS 都支持基于功能的索引，因此 基于功能的”索引属性仅对支持它们的 DBMS 可见，例如 PostgreSQL 和 Oracle。</p>
包括	<p>从当前表中标识非键列的逗号分隔列表 (CSV)。</p> <p>并非所有 DBMS 都支持索引的 包含”属性，因此该属性仅对支持它的 DBMS 可见。</p>

注记

- 警告：Enterprise Architect假定索引至少有一列分配给它们；但是，Enterprise Architect在建模期间不强制执行此规则
如果为定义了未分配列的索引的库表生成 DDL，则该 DDL 将无效，除非该索引是基于功能的
- 分配给基于功能的索引的任何列都将被忽略

独特约束

Unique约束在库表的所有行中强制一组字段的“唯一性”，这意味着库表库表中的任何两行不能在唯一约束的字段中具有相同的值。唯一约束与主键相似（因为它们也强制执行“唯一性”），但主要区别在于库表可以定义多个唯一约束但只有一个主键。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	在图表或浏览器窗口 右键单击库表元素 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将其标识为唯一约束的名称改写约束名称。
2	在“类型”字段中，将值从“索引”更改为“唯一”。

注记

- 警告：Enterprise Architect假定 Unique约束至少有一个列分配给他们；但是，Enterprise Architect在建模期间不强制执行此规则。如果为具有定义的唯一约束但未分配列的库表生成 DDL，则该 DDL 将无效。

外键

外键定义了A列（或列的集合），它强制两个表之间的关系。执行此关系以确保数据完整性是数据库服务器的责任。外键的模型定义由包含一组唯一数据的父（主）库表组成，然后在子（外）库表中引用该数据集。

在Enterprise Architect中，外键使用两个不同（但相关）的UML组件建模：

- 存储在子库表中A外键约束（具有 <<FK>> 原型的UML操作）
- 定义两者之间关系的关联连接器（<<FK>> 的刻板印象）表

创建外键

虽然外键的定义可能很复杂，但外键约束屏幕简化了外键的建模。此屏幕旨在帮助您选择要使用父库表中的哪个约束，并且会自动将子库表列与父库表中属于约束的列进行匹配。为了说明，这里分别描述了开发外键过程的不同方面，但整个过程应该是一个平稳的过渡。

A保存外键定义之前，必须满足许多条件：

- 两个表都必须定义匹配的 DBMS
- 父库表必须至少有一栏
- 父库表必须定义一个主键、唯一约束或唯一索引

创建外键 - 使用数据库

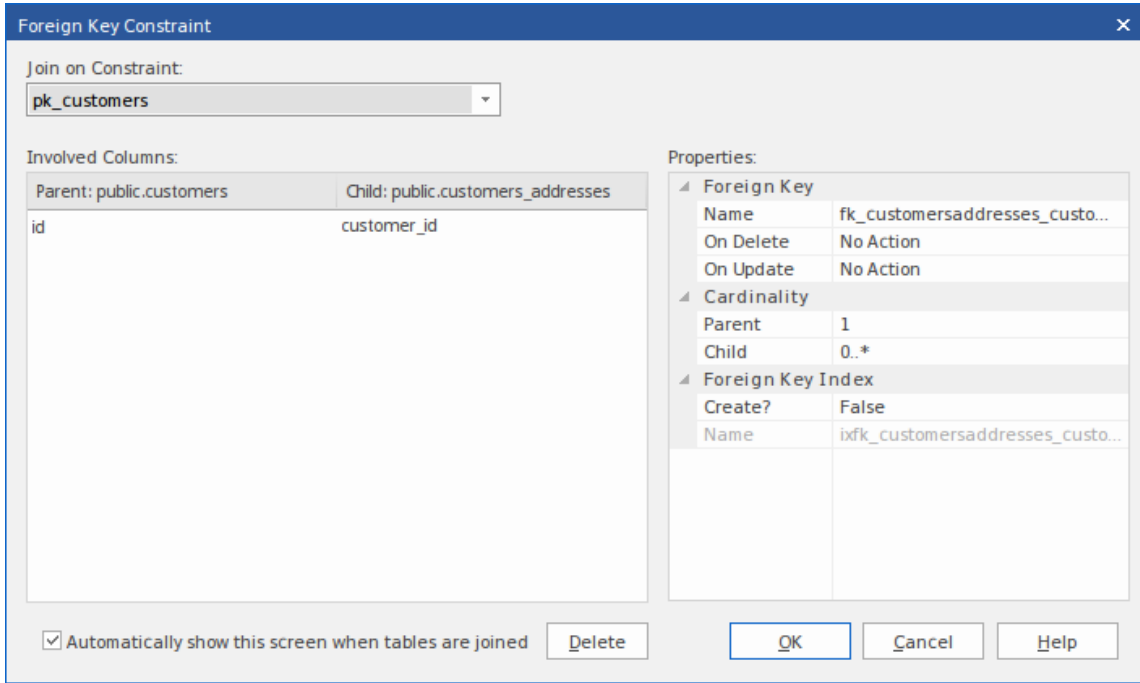
节	行动
1	在数据库构建器树中，右键单击子库表名称并单击 在 <表名称> 上添加新外键”。显示A对话框，列出所有可能的父表。
2	双击列表中所需的父库表名称或选择它并单击确定按钮。显示 外键约束”屏幕。

创建外键 - 在图表上使用关系

节	行动
1	在数据建模图中，找到需要的child（Foreign Key）库表和parent（Primary Key）库表。
2	在图表工具的“数据建模工具箱”页面中图表一个关联连接器。
3	单击子库表并将连接器绘制到父库表。
4	如果外键约束屏幕已设置为两个表连接时自动显示，则显示现在。否则，要么： <ul style="list-style-type: none"> • 双击连接器或 • 右键单击连接器并选择 外键”选项 外键约束屏幕显示。

外键约束屏幕

作为示例，此图像显示加载了“示例”详细信息的外键约束屏幕（如示例模型中所定义）。

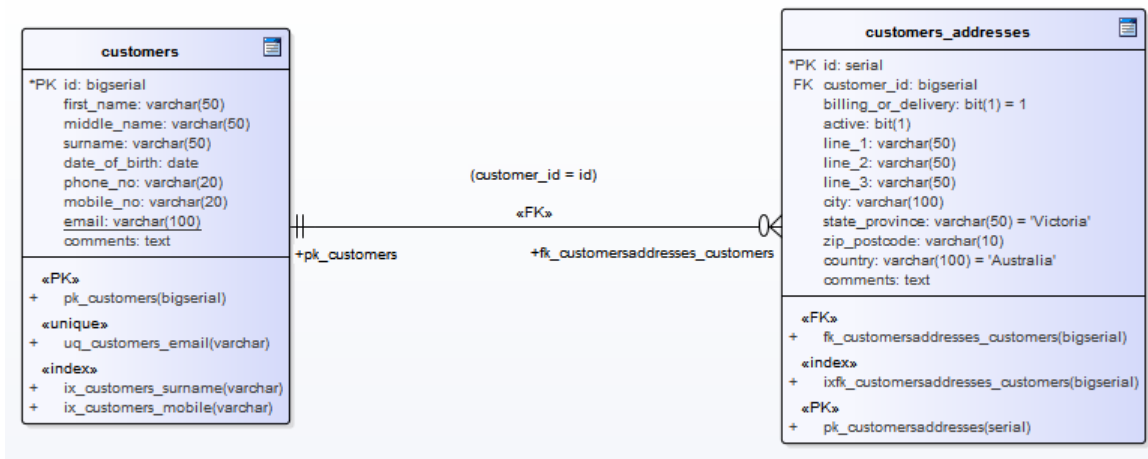


选项	行动
汇合关于约束	<p>这个组合框列出了父库表中所有定义的约束，这些约束可以用作外键的基础。（这些约束可以是主键、唯一约束或唯一索引。）</p> <p>默认选择列表中的第一个约束；如果这不是您想要的约束，请从组合框中选择正确的约束。</p> <p>当您选择约束时，它的列会自动列在“涉及的列”面板中的“父级：<tablename>”列下。</p>
涉及的列	<p>该列表分为两部分：左侧列出了所选约束中涉及的列，右侧列出了要与父列配对的子列。</p> <p>When a constraint is selected (in the '汇合 on constraint' field) the parent side is refreshed to display all columns assigned to the selected constraint. 在子方面，系统将自动尝试将每个父列与子库表中的同名列匹配。如果子库表没有同名列，则该名称的新列将被添加到列表中，用 (*) 标记表示将在库表中创建一个新列。</p> <p>但是，如果您想强制配对到现有的子库表列或具有不同名称的新列，请单击列名称字段，然后：</p> <ul style="list-style-type: none"> • 替换名称中的类型，或 • 选择现有列（单击下拉箭头并从列表中选择名称）
名称	<p>该字段定义外键约束的名称，默认为外键名称模板构造的名称。</p> <p>要将名称更改为默认值以外的名称，只需改写该值即可。</p>
关于删除	<p>选择删除父级数据时对子库表中的数据应采取的动作，以保持参照完整性。</p>
更新时	<p>选择当父项中的数据更新时对子库表中的数据应采取的动作，以保持参照完</p>

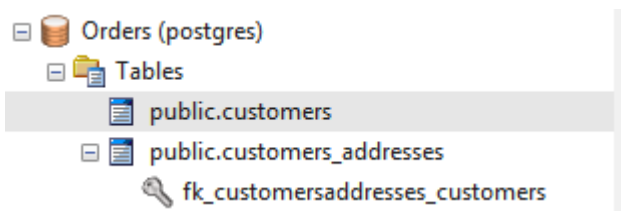
	整性。
家长	单击下拉箭头并在外键中选择父库表的基数。
子	单击下拉箭头并在外键中选择子库表的基数。
创造？	如果要在创建外键的同时创建外键索引，请将此属性设置为True。 外键索引的名称由外键索引模板控制，生成的名称显示在“创建？”下方的“名称”字段中场地。
加入库表时自动显示此画面	(对于图表建模)选中此复选框可在两个表之间创建关联时自动显示此屏幕。
删除	单击此按钮可删除当前选择的现有(已保存)外键。 将显示A提示以确认删除(以及外键索引的删除，如果存在)-单击是按钮。 删除外键会留下一个关联连接器，您可以对其进行编辑或删除(右键单击并选择“删除关联：到<库表名称>”)。
确定	单击此按钮以保存外键。

例子

此示例在图表中显示了简单的外键：



相同的外键将作为库表“customers.addresses”下的子节点显示在数据库构建器的树中。




选择约束

A选择约束通过限制列接受的值来强制域完整性。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在列和约束屏幕的“约束/索引”选项卡上，会自动创建一个新约束并分配默认约束名称和索引的“类型”。 使用将约束标识为检查约束的名称改写约束名称，例如“CHK_ColumnName”（CHK_前缀是可选的）。
2	在“类型”字段中，将值从“索引”更改为“检查”。
3	在条件属性的“属性”面板中，键入# 将用作选择条件的语句；例如，第 1 列 < 1000。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。

删除一个约束选择

如果您不想保留检查约束，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

约束被立即删除。

注记

- 分配给检查约束的任何列都将被忽略

库表触发器

库表触发器A

或者由于数据库中的数据被修改而自动执行的代码库表。触发器可高度定制，并且可以以多种不同方式使用；例如，它们可用于停止在工作时间执行某些数据库活动，或在删除主库表中的记录时提供验证或在辅助表中执行删除。

在Enterprise Architect中，库表触发器被建模为一个定型操作并使用库表的“约束”屏幕进行管理。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建库表触发器

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将约束标识为触发器的名称改写约束名称，例如 TRG_OnCustomerUpdate。（TRG_前缀是可选的。）
2	在“类型”字段中，将值从“索引”更改为“触发器”。
3	在“Statement”属性的“属性”面板中，输入完成# 将定义简单的语句（包括CREATE触发器）。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。
4	触发器时间和触发器事件属性当前是仅供参考的值，不用于DDL生成。

删除库表触发器

如果您不想保留触发器，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

触发器立即被删除。

注记

- 分配给表触发器的任何列都将被忽略

数据库库表栏目

在关系数据库中，库表列（有时称为字段）在库表的每一行中存储特定类型的单个数据值。库表列可以具有各种单独的属性，例如默认值或是否字段接受 Null 值。

A数据库库表列在UML数据建模配置文件中表示为一个原型属性；即，具有 <<column>> 构造型的属性。在 Enterprise Architect中，您可以使用专门设计的数据库构建器的“列”页面或“列和约束”对话框来定义和维护库表列。

创建数据库库表列

数据库库表列A UML数据建模配置文件中表示为具有 <<column>> 构造型的属性。对于选定的库表，您可以在数据库构建器的“列”页面或“列和约束”屏幕上审阅现有列并创建新列。

您可以直接在“列”选项卡上的列列表中定义列详细信息。当您完成每个字段时，更改会自动保存。某些字段对您可以输入的数据有一定的限制，如此处所述。该选项卡还包含一个“属性”面板和一个“注记”字段，其中填充了所选列上的现有信息。您创建的每个新列都会自动分配一组默认值并添加到列表底部。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表 > 列 > 右键 > 添加新列
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右击 添加新列
键盘快捷键	选择表 F9 Tab 键 (在 “列”选项卡上设置输入聚焦) Ctrl+N

在库表中创建列

选项	行动
名称	用适当的列名文本改写默认名称。
类型	单击下拉箭头并为列选择适当的数据类型。 可用的数据类型取决于分配给父库表的 DBMS。
长度	(可选) 某些数据类型具有长度组件 - 例如，VARCHAR 具有定义可以存储的字符数的长度。如果数据类型没有长度组件，则禁用此字段。 如果该字段可用并且您需要定义多个字符，请在此处键入值。
规模	(可选) 某些数据类型具有比例组件 - 例如，DECIMAL 具有定义可以保留的小数位数的比例。如果数据类型没有比例组件，则禁用此字段。 如果该字段可用并且您需要定义比例，请在此处键入值。
PK	如果该列是此库表的主键的一部分，请选中该复选框。
不为空	如果该列禁止使用空值，请选中该复选框。 如果选中“PK”复选框，则该复选框被禁用。
别名	如果出于显示和文档目的需要，请输入该字段的替代名称。
初始值	如果需要，输入一个可用作此列默认值的值。
注记	类型在记录该列所需的任何附加信息中。 您可以使用字段顶部的注记工具栏来格式化文本。

列属性

库表的数据库管理系统的相应属性会自动显示在“属性”面板中（如果不可见，请展开“列（<名称>）”分支）。

属性	数据库管理系统
自动编号（起始编号增量）	甲骨文 MySQL # 服务器 DB2 PostgreSQL 注记：如果您需要自动编号序列，请将此属性设置为True，并在必要时定义起始编号和增量。
生成	DB2 注记：将此附加属性设置为DB2中的自动编号，设置为“默认”或“始终”。
不代表	SQL服务器 注记：如果要阻止复制，请将此属性设置为True。
零填充	MySQL 注记：将此属性设置为True或False以指示字段是否填充为零。
未签名	MySQL 注记：将此属性设置为True或False以指示字段是否接受无符号数字。
长度类型	甲骨文 注记：设置该属性以定义字符语义为“None”、“Byte”或“Char”。

删除数据库库表列

对于选定的数据库库表，您可以在“列和约束”屏幕的“列”选项卡上审阅现有列并删除任何单个列。

访问

使用此处概述的方法之一来显示表格的列表，然后选择一列并将其删除。

当您选择“删除列<名称>”选项时，如果满足所有验证规则，则立即删除该列。

功能区	开发>数据建模>数据库生成器>点击库表>栏目>右键栏名>删除栏目<名称>
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右键单击列名 删除列<名称>
键盘快捷键	F9 使用'向上箭头'或'向下箭头'选择一列 Ctrl+D

注记

- 如果删除的数据库库表列涉及任何约束，它将自动从它们中删除

重新排序数据库库表列

如果您在数据库库表中定义了多个列，则可以更改它们的列出顺序。列表中的顺序是列在生成的 DDL 中出现的顺序。

访问

使用此处概述的方法之一来显示库表的列列表，然后选择一列并在列表中重新定位它。

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表
上下文菜单	在图表中，右键单击所需的库表 特征 列
键盘快捷键	F9

更改列顺序

节	行动
1	在“列”选项卡中，单击列表中所需的列名。
2	右键单击并选择： <ul style="list-style-type: none">“上移列 <名称>”选项（或按 Ctrl+向上箭头）将列上移一位“将列 <名称> 向下移动”选项（或按 Ctrl+向下箭头）将列向下移动一个位置 这些选项在“列”选项卡和图表中都会立即生效。

使用库数据库库表属性

创建数据库库表后，您可以审阅其属性并检查 DBMS 和拥有者值是否正确。要显示库表的“属性”对话框，请双击“数据库生成器表”包中的库表名称或图表上的库表元素。

重要的

先将 DBMS 分配给库表，然后才能在其中添加列。如果您使用数据库生成器，则数据模型的 DBMS 将自动应用于所有新表；但是，如果您通过其他方式（例如处理图表）添加了库表，则这是一个手动步骤。

任务

一旦定义了数据库库表属性，您就可以添加列了。

任务
为库表设置数据库类型——除了库表名称之外，为数据库库表设置的最重要的属性是数据库类型。
设置数据库库表 Owner - 对于某些 DBMS，所有表必须分配一个拥有者/架构；在 Enterprise Architect 中，此属性定义为名称为拥有者的标记值。
设置扩展选项- 一些 DBMS 具有仅与该 DBMS 相关的扩展选项。这些扩展属性被存储为标记值。

默认数据库管理系统

在创建物理数据模型之前，建议您设置默认 DBMS，它将自动应用于您在数据库构建器之外创建的新数据库对象。您可以通过以下方式之一设置默认 DBMS 类型：

- 选择 开始>外观>首选项>首选项>源代码工程>代码编辑器”，然后设置字段“默认数据库”
- 选择 设置 >参考>设置 >数据库数据类型”，然后选择产品名称并选中“设置为默认值”复选框
- 在代码生成工具栏的第二个字段中设置 DBMS

设置数据库类型

为数据库库表设置的最重要的属性（在其名称之后）是数据库类型或 DBMS。选择的 DBMS 值将控制 Enterprise Architect 将如何确定：

- 如何显示库表名称（有或没有拥有着）
- 数据库建模时将应用哪组验证规则
- 创建列时可用的数据类型。
- DDL 生成中将使用哪组 DDL 模板

访问

在浏览器窗口或图表中选择一个库表，然后使用此处概述的任何方法，打开库表的“属性”对话框，显示“常规”选项卡，然后显示“主”子选项卡。

功能区	设计>元素>编辑器>属性对话框>常规>主
上下文菜单	右键单击库表元素 属性 特别行动 一般 主要的
键盘快捷键	Shift+Enter 一般 主要的
其它	双击库表元素 General 主要的

选项

字段/按钮	行动
数据库	单击下拉按钮并从列表中选择所需的数据库类型。
申请	单击“应用”按钮以保存任何待处理的更改。
确定	单击确定按钮以保存任何未决更改并关闭屏幕。

设置数据库库表拥有着/架构

对于某些 DBMS，必须为所有表分配一个拥有着/架构。在Enterprise Architect中，此属性在物理上定义为名称为拥有着的标记值。但是，提供了一个特殊的属性页面，可以帮助您轻松管理拥有着的属性。

访问

在浏览器窗口或图表中选择一个库表，然后使用此处列出的任何方法，打开库表的“属性”对话框，显示“常规”选项卡并显示“库表详细信息”子选项卡。

功能区	设计>元素>编辑>属性><<表>>
上下文菜单	右键单击库表元素 属性 特殊行动>常规>库表详细信息
键盘快捷键	Shift+Enter 综合 库表详情
其它	双击库表元素 'General' '库表详情'

设置数据库库表所有者

节	行动
1	在“拥有着者”字段中，输入库表的所有者或模式的名称。

设置MySQL选项

要在MySQL中使用外键，您必须将数据库库表类型声明为 InnoDB。

将库表类型声明为 InnoDB

节	行动
1	在库表中添加一个名为类型的标记值。
2	将 值"字段设置为 InnoDB"。

生成DDL

为这个库表生成 DDL 时，库表类型包含在#脚本。

为了允许更高版本的MySQL，可以以相同方式添加的其他库表选项包括：

标签	价值 (示例)
引擎	InnoDB
字符集	拉丁语1
字符集	拉丁语1
整理	latin1_german2_ci

设置 Oracle 数据库库表属性

要设置额外的 Oracle 数据库库表属性，请使用库表的标记值。

设置属性

通过突出显示索引或约束操作并添加适当的标记值，可以将相同的属性添加到索引和约束中。

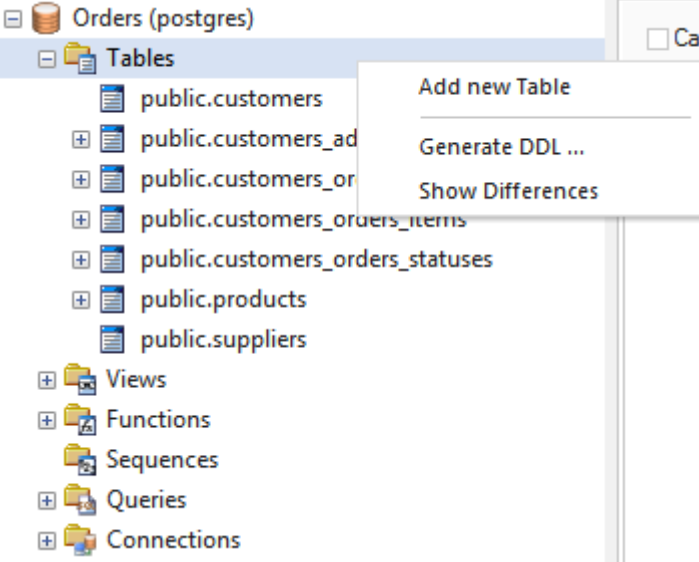
节	行动
1	使用“属性”库表的“属性/标签”列中提供的名称，将一个或多个标记值添加到库表库表中。
2	<p>为每个标签指定适当的值。 此属性库表的“值”列中提供了示例。</p> <ul style="list-style-type: none"> • 缓存-NOCACHE • 数据库版本 - 9.0.111 • 自由清单 - 1 • 授予所有者 1 - SELECT • GRANT OWNER2 - 删除、插入、SELECT、更新 • 首字母 - 65536 • 初始化 - 1 • 记录 - 记录 • MAXEXTENTS - 2147483645 • MAXTRANS - 255 • MINEXTENTS - 1 • 监控 - 监控 • 所有者 - 所有者 1 • 平行 - 不平行 • 无PCT - 10 • PCT 增加 - 0 • PCT 已使用 - 0 • 同义词 - PUBLIC:TABLE_PUB;OWNER2:TABLE_OWNER2 • 表空间 - MY_TABLESPACE • 临时的 - 是的

创建数据库库表


数据建模的基础是在模型中创建数据库表。创建库表的三种方式：

- 在数据库生成器中
- 在打开的数据模型图上
- 使用浏览器新元素选项

使用数据库添加数据库库表

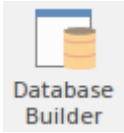
节	行动
1	打开数据库构建器（ 开发 >数据建模>数据库构建器” ）。
2	加载或创建数据模型。
3	右击表包，选择 “加新库表”。 
4	用适当的库表名称改写默认名称，然后按 Enter 键。
5	双击库表元素，定义库表属性。

将数据库库表添加到图表

节	行动
1	创建和/或打开数据建模图。
2	将 “库表” 工具箱图标拖放到图表上。 

	这会生成一个新的库表元素： 
3	双击库表元素，定义库表属性。

数据库生成器

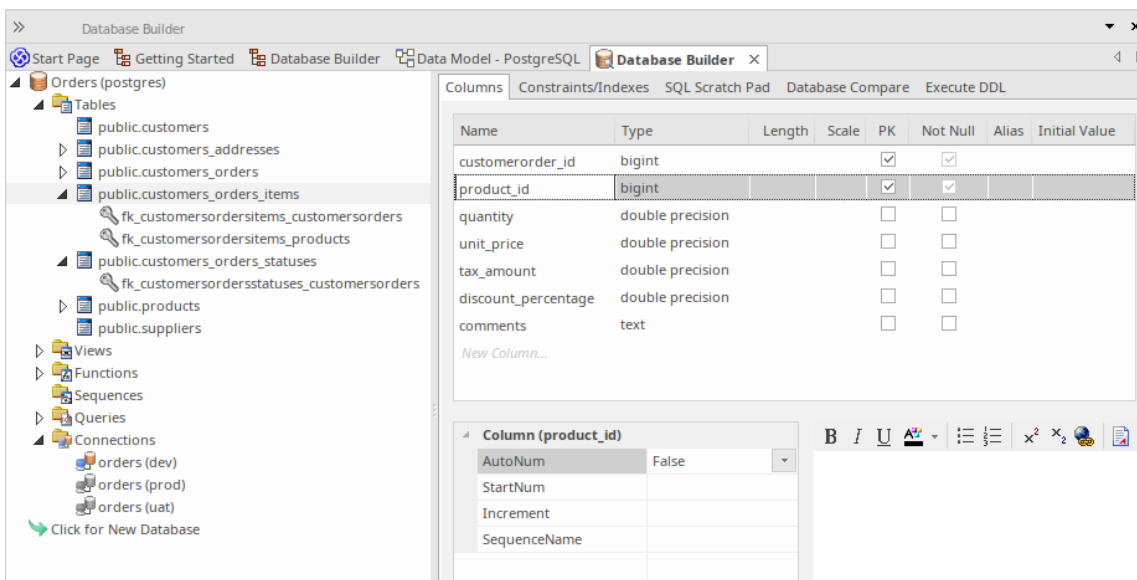


数据库是为数据建模器量身定制的界面；所有与数据库相关的建模任务都可以在一个位置执行。界面及其相关屏幕仅包括与数据建模相关的信息，从而简化和简化建模过程。

访问

功能区	开发 > 数据建模 > 数据库构建器
-----	--------------------

数据库生成器



此图显示了加载了“Orders (postgres)”数据模型的数据库构建器，它出现在示例模型中。

概述

数据库生成器的接口包括：

- 数据A树，列出当前存储库中所有已定义的数据模型
- A 列”您可以通过它直接管理库表列的选项卡
- A '约束/索引'用于直接管理主键、外键和索引等库表约束的选项卡
- 可用于运行# SQL便签本# 针对实时数据库的查询
- A数据库比较”选项卡，显示数据模型和实时数据库之间的比较结果

- 一个“执行 DDL”选项卡，您可以在其中立即针对实时数据库执行生成的 DDL
- 您可以使用数据库来：
- 创建、编辑和删除数据库对象（表、视图、过程、序列和函数）
 - 创建、编辑和删除库表约束（主键、索引、唯一约束、选择约束和触发器）
 - 创建、编辑和删除库表外键
 - 逆向工程数据库架构信息
 - 从建模数据库生成 DDL
 - 将实时数据库模式与建模数据库进行比较
 - 针对实时数据库执行生成的 DDL
 - 执行即席#
针对实时数据库的语句

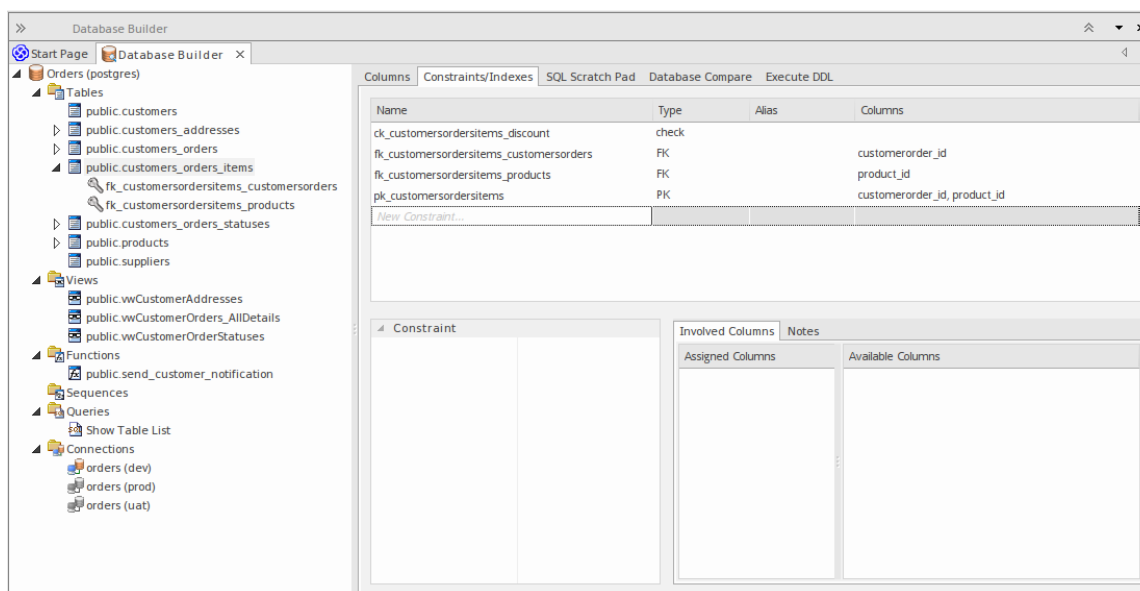
注记

- Enterprise Architect 的企业统一版和终极版中提供了数据库构建器

约束/索引

约束表是基本的数据库object，约束表和索引（及其属性）是数据建模人员更新和更改的第二频繁修改的库表特征，因此“约束表/索引”页面方便地位于数据库的第二页生成器的界面。

一旦在数据库构建器的树中选择了库表，“约束/索引”页面将填充当前定义的所选库表的约束和索引列表。然后，数据建模者可以直接在列表中对主要属性进行更改。当数据建模者在列表中选择单个约束或索引时，约束的扩展属性（和注释）立即显示在列表下方，允许修改这些扩展属性。



此图显示了显示对象树的数据库生成器界面和显示所选表的列的列选项卡。

笔记

- “约束/索引”页面仅在选择数据库构建器树中的库表项时才会填充

数据库约束库表/索引

在Enterprise Architect中，库表约束和索引在同一屏幕上建模；它们统称为约束。数据库约束定义了强加于数据库行为的条件库表。他们包括：

- Primary Key - 唯一标识库表中的一条记录，由一列或多列组成
- 索引 - 提高对库表数据的检索和排序操作的性能
- 唯一约束- 唯一标识库表中一行的值的组合
- 外键 - 强制两个表之间的关系（或列集合）
- 选择约束- 通过限制列接受的值来强制域完整性
- 库表触发器- #
或由于库表中的数据被修改而自动执行的代码

在Enterprise Architect中，您可以使用数据库构建器中专门设计的“约束/索引”页面或列和约束屏幕来定义和维护库表约束。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

创建任何这些约束类型的过程是相同的，并且可以通过此处描述的一种方式来实现。

创建一个约束-使用上下文菜单或键盘

节	行动
1	A新的约束被自动创建并分配了默认名称约束 <i>n</i> （其中 <i>n</i> 是一个计数器）和一个“索引”的类型”。用您自己的约束名称改写默认名称。
2	如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。 “列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

创建一个约束-改写模板文本

节	行动
1	在所选库表的“约束/索引”选项卡上，约束列表以模板文本新约束结束。用适当的约束名称改写此文本，然后按 Enter 键。
2	新约束会自动创建并分配默认类型索引。如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。“列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

将列分配给约束

Primary Key、Foreign Key、Index、Unique 这些约束类型都必须至少分配一列；这定义了约束中涉及的列。

节	行动
1	在所选库表的“约束/索引”选项卡上，单击要为其分配列的约束。
2	“可用列”面板列出了为库表定义的所有列。对于要分配给约束的每一列，右键单击列名称并选择“分配列 <名称>”。列名被传送到“分配的列”列表中。

从约束中取消分配列

节	行动
1	在所选库表的“约束/索引”选项卡上，单击要取消分配列的约束。
2	在“分配的列”列表中，右键单击要从约束中取消分配的列的名称，然后选择“取消分配列 <名称>”。列名被转移到“可用列”列表中。

重新排序约束中的指定列

如果约束中有许多列，则可以通过将选定的列名一次在列表中向上或向下移动一个位置来重新排列序列。去做这个：

- 右键单击要移动的列名称并选择：
 - 向上移动列“<名称>”（Ctrl+向上箭头）或
 - 向下移动列“<name>”（Ctrl+向下箭头）

删除约束

要删除不再需要的约束，请右键单击“约束/索引”选项卡上列表中的约束名称，然后选择“删除约束 <名称>”选项。如果满足给定约束类型的所有验证规则，则会立即从存储库中删除该约束以及所有相关关系（如果有的话）。

主键

主键是唯一标识库表中每条记录A列（或列集）。A库表只能有一个主键。一些 DBMS 支持主键的附加属性，例如 Clustered 或 Fill Factor。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称
上下文菜单	图中 右键单击库表 特征 约束/索引

创建主键

在Enterprise Architect中，您可以从“列”选项卡或“约束/索引”选项卡创建主键。无论哪种情况，当您将列添加到主键约束时，该列都会自动设置为“非空”。此外，任何包含库表元素的图表（假设设置了“显示限定词和可见性指示器”选项）都将针对列名显示“PK”前缀。在此图像中，请参见第一列“id: bigserial”。

customers

***PK** id: bigserial
first_name: varchar(50)
middle_name: varchar(50)
surname: varchar(50)
date_of_birth: date
phone_no: varchar(20)
mobile_no: varchar(20)
email: varchar(100)
comments: text

«PK»
+ pk_customers(id: bigserial)

«unique»
+ uq_customers_email(email: varchar)

«index»
+ ix_customers_surname(surname: varchar)
+ ix_customers_mobile(mobile_no: varchar)

创建主键 - 从列选项卡

节	行动
---	----

1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击具有一个或多个已定义列的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	<p>对于要包含在主键中的每一列，选择 PK”复选框。</p> <p>如果之前没有为当前库表定义主键约束，系统将使用主键名称模板创建一个新约束。</p>

创建主键 - 从约束选项卡

节	行动
1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击包含一个或多个已定义列的库表，然后单击 约束/索引”选项卡，或 在图表上，单击库表并按 F10 以显示 约束/索引”选项卡
2	<p>用主键名称改写新约束文本，按 Enter 键并单击 类型”字段下拉箭头，然后选择 PK”。</p>
3	<p>将所需的列分配给 PK 约束。</p>
4	<p>使用属性面板设置主键的扩展属性。</p> <ul style="list-style-type: none"> 填充因子是 0 到 100 之间的数值 Is Clustered 是一个布尔值，用于确定数据存储方式的物理顺序；对于大多数 DBMS，主键的 Is Clustered 属性默认为 True

从主键中删除列

您可以使用 列”选项卡或 约束/索引”选项卡从主键中删除列。

从主键中删除列 - 使用 列”选项卡

节	行动
1	<p>任何一个：</p> <ul style="list-style-type: none"> 在数据库构建器中，单击带有主键的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	<p>针对要从主键中删除的每一列，取消选中 PK”复选框。</p> <p>如果您已从主键约束中删除所有列并且不再需要主键，则必须手动删除它。</p>

从主键中删除列 - 使用约束/索引选项卡

节	行动
1	任何一个： <ul style="list-style-type: none">在数据库构建器中，单击带有主键的库表，然后单击“约束/索引”选项卡，或在图表上，单击库表并按 F10 以显示“约束/索引”选项卡
2	根据需要取消分配 PK 约束上的列。

注记

- 警告：Enterprise Architect假定主键约束至少分配了一列；但是，Enterprise Architect在建模期间不强制执行此规则
如果为主键未分配列的库表生成 DDL，则该 DDL 无效

数据库索引

将数据库索引应用于表以提高数据检索和排序操作的性能。一个库表可以定义多个索引；但是，每个索引都会在数据库服务器上增加开销（以处理时间和存储的形式）来维护它们，因为在库表中添加和删除信息

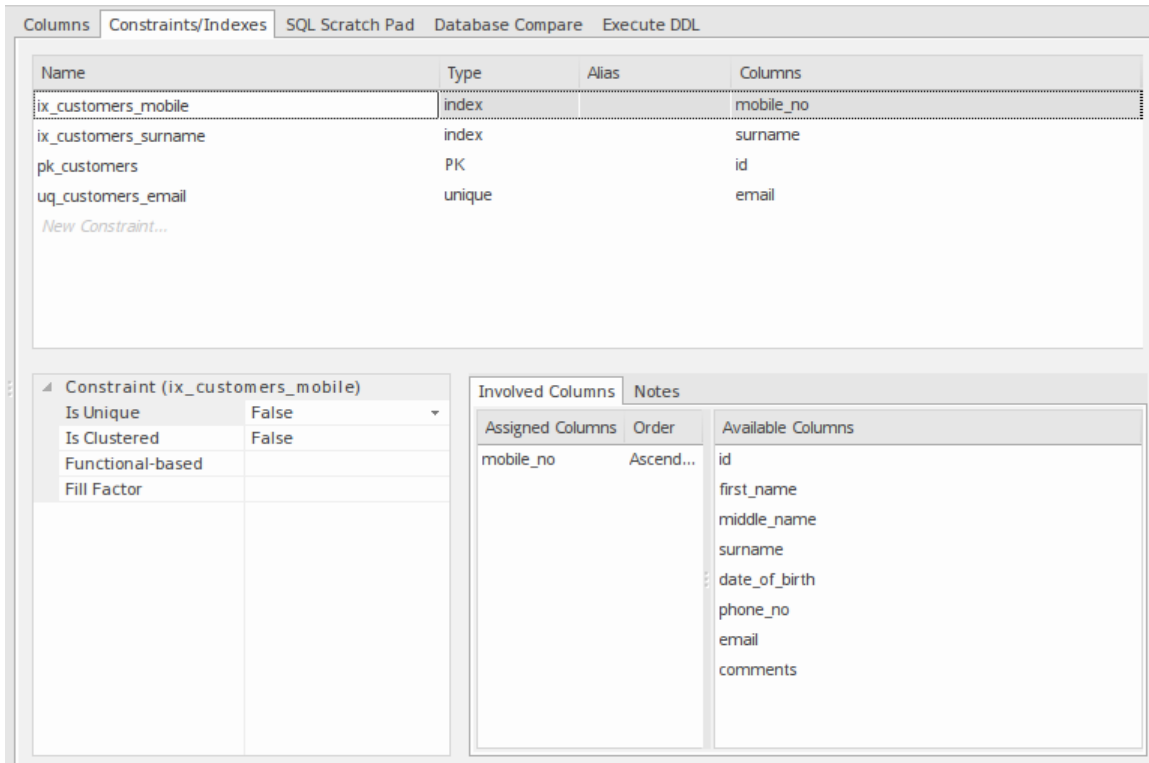
在Enterprise Architect中，索引被建模为原型操作。

一些 DBMS 支持特殊类型的索引； Enterprise Architect使用其他属性来定义这些，例如基于函数的、集群的和填充因子。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	图中 右键单击库表 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引

处理索引



节	行动
1	在库表的“约束/索引”选项卡上，右键单击并选择“添加新约束”。 使用默认名称“constraint1”和“index”类型添加新约束。 用您喜欢的索引名称改写名称。

2	<p>将适当的列分配给索引。</p> <p>'Assigned Columns' 列表有一个附加的 'Order' 字段，用于指定每个已分配列存储在索引中的顺序（升序或降序）。您可以根据需要切换每列的顺序。</p> <p>此外，对于MySQL索引，“Len”字段将可见，您可以在其中定义部分索引；即，使用基于文本的字段的前导 <i>n</i> 个字符的索引。“Len”字段仅采用介于 0 和列的定义长度之间的整数数值。A 0（这是默认值）表示要索引整个列。</p>
3	<p>在“属性”面板中，审阅为当前 DBMS 定义的扩展属性的设置。</p>

附加属性

属性	描述
是独特的	<p>(True / False) 指示当前索引是否为 唯一索引”。唯一索引确保索引A（或列）不包含重复值，从而确保每一行具有唯一值（或索引由多列组成时的值组合）。</p>
是集群的	<p>(True / False) 指示当前索引是否为 聚集索引”。使用聚集索引，表的行在物理上以与索引中相同的顺序存储，因此每个表只能有一个聚集索引。默认情况下，一个表的主键是集群的。</p> <p>并非所有 DBMS 都支持聚集索引，因此 Is Clustered”索引属性仅对支持它的 DBMS 可见。</p>
是位图	<p>(True / False) 指示当前索引是否为 位图”索引。位图索引旨在用于具有相对较少唯一值的列（称为 低基数”列）并且物理上由每个唯一值的位数组（通常称为位图）组成。对于表中的每一行，每个数组都有一个位。</p> <p>考虑这个示例：在名为 性别”的列上创建位图索引，该列具有 男性”或 女性”选项。物理上，索引将由两个位数组组成，一个用于 男性”，一个用于 女性”。女性位数组在匹配行的值为 女性”的每个位中都有一个1。</p> <p>Is Bitmap 和 Is Unique属性是互斥的，所以当 Is Bitmap属性为True时，DDL 生成将忽略 Is属性。</p> <p>位图索引仅受 Oracle 支持；因此，此属性仅在对 Oracle 索引建模时可见。</p>
填充系数	<p>A介于 0 和 100 之间的数值，用于定义应用于数据的可用空间百分比。</p> <p>并非所有 DBMS 都支持填充因子，因此 填充因子”索引属性仅对支持它的 DBMS 可见。</p>
基于功能	<p>A #</p> <p>定义将被评估的函数/语句和索引结果的语句；例如： 降低（“字段”）</p> <p>并非所有 DBMS 都支持基于功能的索引，因此 基于功能的”索引属性仅对支持它们的 DBMS 可见，例如 PostgreSQL 和 Oracle。</p>
包括	<p>从当前表中标识非键列的逗号分隔列表 (CSV)。</p> <p>并非所有 DBMS 都支持索引的 包含”属性，因此该属性仅对支持它的 DBMS 可见。</p>

注记

- 警告：Enterprise Architect假定索引至少有一列分配给它们；但是，Enterprise Architect在建模期间不强制执行此规则
如果为定义了未分配列的索引的库表生成 DDL，则该 DDL 将无效，除非该索引是基于功能的
- 分配给基于功能的索引的任何列都将被忽略

独特约束

Unique约束在库表的所有行中强制一组字段的“唯一性”，这意味着库表库表中的任何两行不能在唯一约束的字段中具有相同的值。唯一约束与主键相似（因为它们也强制执行“唯一性”），但主要区别在于库表可以定义多个唯一约束但只有一个主键。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	在图表或浏览器窗口 右键单击库表元素 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将其标识为唯一约束的名称改写约束名称。
2	在“类型”字段中，将值从“索引”更改为“唯一”。

注记

- 警告：Enterprise Architect假定 Unique约束至少有一个列分配给他们；但是，Enterprise Architect在建模期间不强制执行此规则。如果为具有定义的唯一约束但未分配列的库表生成 DDL，则该 DDL 将无效。

外键

外键定义了A列（或列的集合），它强制两个表之间的关系。执行此关系以确保数据完整性是数据库服务器的责任。外键的模型定义由包含一组唯一数据的父（主）库表组成，然后在子（外）库表中引用该数据集。

在Enterprise Architect中，外键使用两个不同（但相关）的UML组件建模：

- 存储在子库表中A外键约束（具有 <<FK>> 原型的UML操作）
- 定义两者之间关系的关联连接器（<<FK>> 的刻板印象）表

创建外键

虽然外键的定义可能很复杂，但外键约束屏幕简化了外键的建模。此屏幕旨在帮助您选择要使用父库表中的哪个约束，并且会自动将子库表列与父库表中属于约束的列进行匹配。为了说明，这里分别描述了开发外键过程的不同方面，但整个过程应该是一个平稳的过渡。

A保存外键定义之前，必须满足许多条件：

- 两个表都必须定义匹配的 DBMS
- 父库表必须至少有一栏
- 父库表必须定义一个主键、唯一约束或唯一索引

创建外键 - 使用数据库

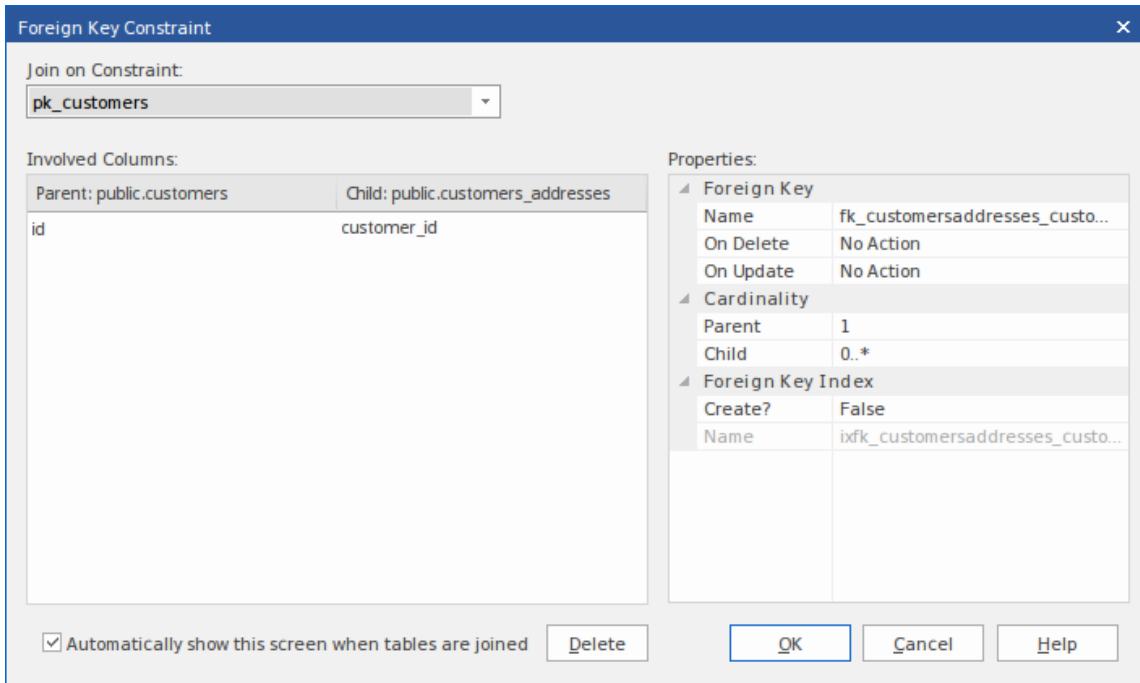
节	行动
1	在数据库构建器树中，右键单击子库表名称并单击 在 <表名称> 上添加新外键”。显示A对话框，列出所有可能的父表。
2	双击列表中所需的父库表名称或选择它并单击确定按钮。显示 外键约束”屏幕。

创建外键 - 在图表上使用关系

节	行动
1	在数据建模图中，找到需要的child（Foreign Key）库表和parent（Primary Key）库表。
2	在图表工具的“数据建模工具箱”页面中图表一个关联连接器。
3	单击子库表并将连接器绘制到父库表。
4	如果外键约束屏幕已设置为两个表连接时自动显示，则显示现在。否则，要么： <ul style="list-style-type: none"> • 双击连接器或 • 右键单击连接器并选择 外键”选项 外键约束屏幕显示。

外键约束屏幕

作为示例，此图像显示加载了“示例”详细信息的外键约束屏幕（如示例模型中所定义）。

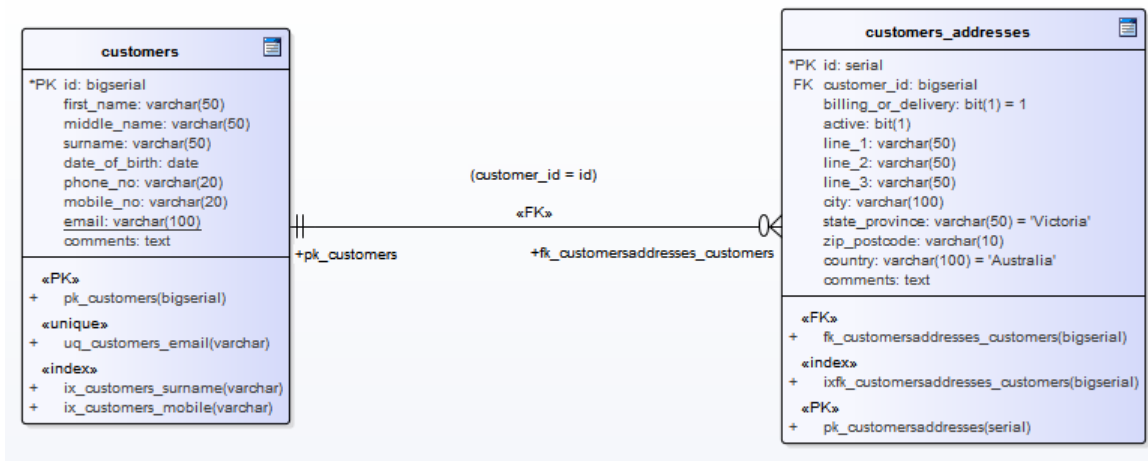


选项	行动
汇合关于约束	<p>这个组合框列出了父库表中所有定义的约束，这些约束可以用作外键的基础。（这些约束可以是主键、唯一约束或唯一索引。）</p> <p>默认选择列表中的第一个约束；如果这不是您想要的约束，请从组合框中选择正确的约束。</p> <p>当您选择约束时，它的列会自动列在“涉及的列”面板中的“父级：<tablename>”列下。</p>
涉及的列	<p>该列表分为两部分：左侧列出了所选约束中涉及的列，右侧列出了要与父列配对的子列。</p> <p>When a constraint is selected (in the '汇合 on constraint' field) the parent side is refreshed to display all columns assigned to the selected constraint. 在子方面，系统将自动尝试将每个父列与子库表中的同名列匹配。如果子库表没有同名列，则该名称的新列将被添加到列表中，用 (*) 标记表示将在库表中创建一个新列。</p> <p>但是，如果您想强制配对到现有的子库表列或具有不同名称的新列，请单击列名称字段，然后：</p> <ul style="list-style-type: none"> • 替换名称中的类型，或 • 选择现有列（单击下拉箭头并从列表中选择名称）
名称	<p>该字段定义外键约束的名称，默认为外键名称模板构造的名称。</p> <p>要将名称更改为默认值以外的名称，只需改写该值即可。</p>
关于删除	<p>选择删除父级数据时对子库表中的数据应采取的动作，以保持参照完整性。</p>
更新时	<p>选择当父项中的数据更新时对子库表中的数据应采取的动作，以保持参照完</p>

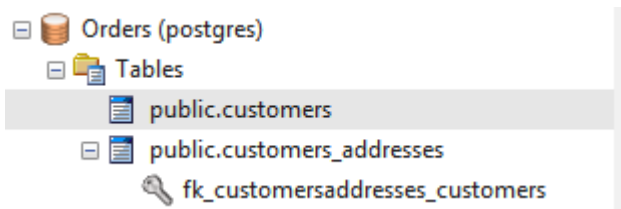
	整性。
家长	单击下拉箭头并在外键中选择父库表的基数。
子	单击下拉箭头并在外键中选择子库表的基数。
创造？	如果要在创建外键的同时创建外键索引，请将此属性设置为True。 外键索引的名称由外键索引模板控制，生成的名称显示在“创建？”下方的“名称”字段中场地。
加入库表时自动显示此画面	(对于图表建模)选中此复选框可在两个表之间创建关联时自动显示此屏幕。
删除	单击此按钮可删除当前选择的现有(已保存)外键。 将显示A提示以确认删除(以及外键索引的删除，如果存在)-单击是按钮。 删除外键会留下一个关联连接器，您可以对其进行编辑或删除(右键单击并选择“删除关联：到<库表名称>”)。
确定	单击此按钮以保存外键。

例子

此示例在图表中显示了简单的外键：



相同的外键将作为库表“customers.addresses”下的子节点显示在数据库构建器的树中。




选择约束

A选择约束通过限制列接受的值来强制域完整性。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在列和约束屏幕的“约束/索引”选项卡上，会自动创建一个新约束并分配默认约束名称和索引的“类型”。 使用将约束标识为检查约束的名称改写约束名称，例如“CHK_ColumnName”（CHK_前缀是可选的）。
2	在“类型”字段中，将值从“索引”更改为“检查”。
3	在条件属性的“属性”面板中，键入# 将用作选择条件的语句；例如，第 1 列 < 1000。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。

删除一个约束选择

如果您不想保留检查约束，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

约束被立即删除。

注记

- 分配给检查约束的任何列都将被忽略

库表触发器

库表触发器A

或者由于数据库中的数据被修改而自动执行的代码库表。触发器可高度定制，并且可以以多种不同方式使用；例如，它们可用于停止在工作时间执行某些数据库活动，或在删除主库表中的记录时提供验证或在辅助表中执行删除。

在Enterprise Architect中，库表触发器被建模为一个定型操作并使用库表的“约束”屏幕进行管理。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建库表触发器

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将约束标识为触发器的名称改写约束名称，例如 TRG_OnCustomerUpdate。（TRG_前缀是可选的。）
2	在“类型”字段中，将值从“索引”更改为“触发器”。
3	在 Statement 属性的“属性”面板中，输入完成# 将定义简单的语句（包括 CREATE 触发器）。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。
4	触发器时间和触发器事件属性当前是仅供参考的值，不用于 DDL 生成。

删除库表触发器

如果您不想保留触发器，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

触发器立即被删除。

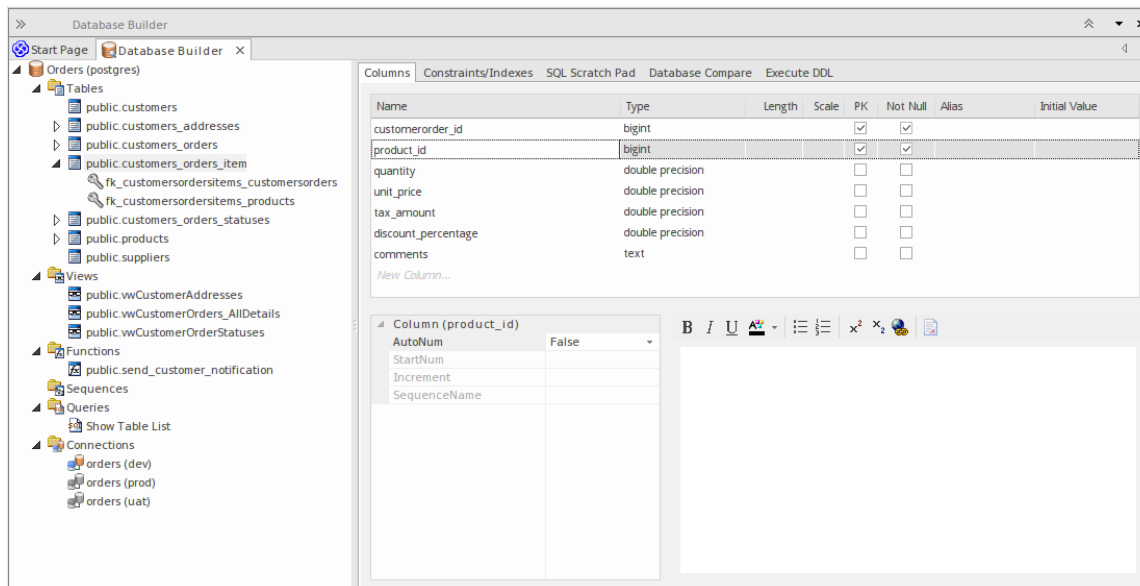
注记

- 分配给表触发器的任何列都将被忽略

列

表是基本的数据库object，而列（及其属性）是最常修改的库表特征，由数据建模者更新和更改，因此“列”页面位于数据库构建器界面的第一页，非常方便。

一旦在数据库生成器的树中选择一个库表，“列”页面就会填充当前为该库表定义的列列表。然后，数据建模者可以直接在列表或网格中对主列属性进行更改。当数据建模者选择列表中的各个列时，列的扩展属性（和注释）立即显示在列表下方，允许修改这些扩展属性。



此图显示了显示对象树的数据库生成器界面和显示所选表的列的列选项卡。

注记

- 只有在数据库构建器的树中选择了库表项时，才会填充“列”页面

创建数据库库表列

数据库库表列A UML数据建模配置文件中表示为具有 <<column>> 构造型的属性。对于选定的库表，您可以在数据库构建器的“列”页面或“列和约束”屏幕上审阅现有列并创建新列。

您可以直接在“列”选项卡上的列列表中定义列详细信息。当您完成每个字段时，更改会自动保存。某些字段对您可以输入的数据有一定的限制，如此处所述。该选项卡还包含一个“属性”面板和一个“注记”字段，其中填充了所选列上的现有信息。您创建的每个新列都会自动分配一组默认值并添加到列表底部。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表 > 列 > 右键 > 添加新列
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右击 添加新列
键盘快捷键	选择表 F9 Tab 键 (在 “列”选项卡上设置输入聚焦) Ctrl+N

在库表中创建列

选项	行动
名称	用适当的列名文本改写默认名称。
类型	单击下拉箭头并为列选择适当的数据类型。 可用的数据类型取决于分配给父库表的 DBMS。
长度	(可选) 某些数据类型具有长度组件 - 例如，VARCHAR 具有定义可以存储的字符数的长度。如果数据类型没有长度组件，则禁用此字段。 如果该字段可用并且您需要定义多个字符，请在此处键入值。
规模	(可选) 某些数据类型具有比例组件 - 例如，DECIMAL 具有定义可以保留的小数位数的比例。如果数据类型没有比例组件，则禁用此字段。 如果该字段可用并且您需要定义比例，请在此处键入值。
PK	如果该列是此库表的主键的一部分，请选中该复选框。
不为空	如果该列禁止使用空值，请选中该复选框。 如果选中“PK”复选框，则该复选框被禁用。
别名	如果出于显示和文档目的需要，请输入该字段的替代名称。
初始值	如果需要，输入一个可用作此列默认值的值。
注记	类型在记录该列所需的任何附加信息中。 您可以使用字段顶部的注记工具栏来格式化文本。

列属性

库表的数据库管理系统的相应属性会自动显示在“属性”面板中（如果不可见，请展开“列（<名称>）”分支）。

属性	数据库管理系统
自动编号（起始编号增量）	甲骨文 MySQL # 服务器 DB2 PostgreSQL 注记：如果您需要自动编号序列，请将此属性设置为True，并在必要时定义起始编号和增量。
生成	DB2 注记：将此附加属性设置为DB2中的自动编号，设置为“默认”或“始终”。
不代表	SQL服务器 注记：如果要阻止复制，请将此属性设置为True。
零填充	MySQL 注记：将此属性设置为True或False以指示字段是否填充为零。
未签名	MySQL 注记：将此属性设置为True或False以指示字段是否接受无符号数字。
长度类型	甲骨文 注记：设置该属性以定义字符语义为“None”、“Byte”或“Char”。

删除数据库库表列

对于选定的数据库库表，您可以在“列和约束”屏幕的“列”选项卡上审阅现有列并删除任何单个列。

访问

使用此处概述的方法之一来显示表格的列表，然后选择一列并将其删除。

当您选择“删除列<名称>”选项时，如果满足所有验证规则，则立即删除该列。

功能区	开发>数据建模>数据库生成器>点击库表>栏目>右键栏名>删除栏目<名称>
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右键单击列名 删除列<名称>
键盘快捷键	F9 使用'向上箭头'或'向下箭头'选择一列 Ctrl+D

注记

- 如果删除的数据库库表列涉及任何约束，它将自动从它们中删除

重新排序数据库库表列

如果您在数据库库表中定义了多个列，则可以更改它们的列出顺序。列表中的顺序是列在生成的 DDL 中出现的顺序。

访问

使用此处概述的方法之一来显示库表的列列表，然后选择一列并在列表中重新定位它。

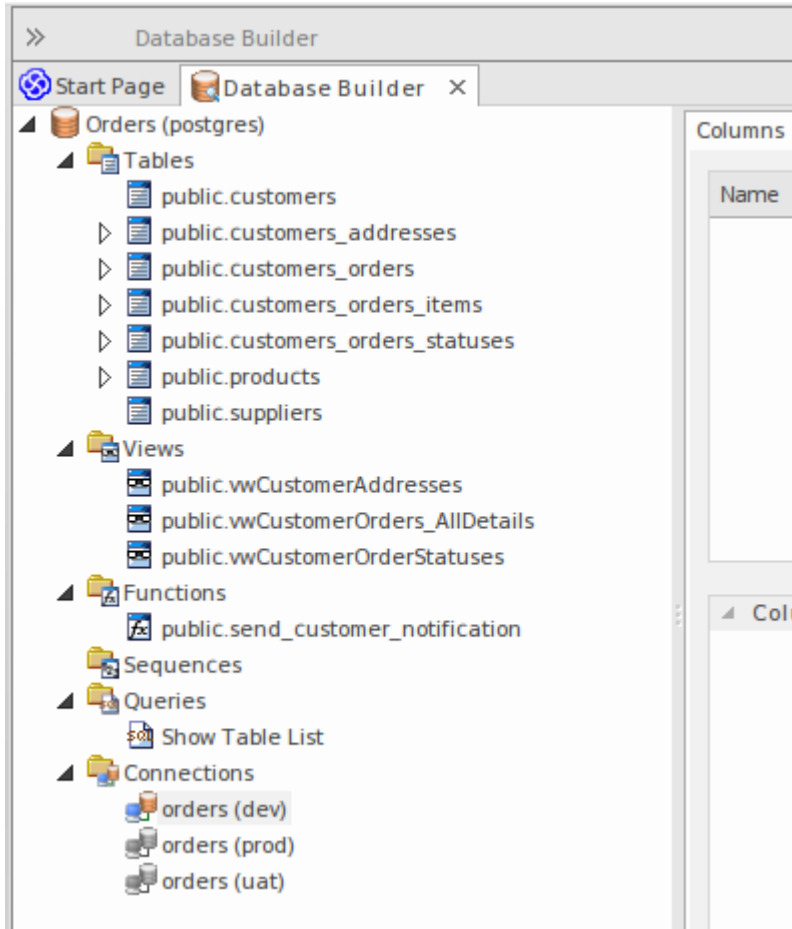
功能区	开发 > 数据建模 > 数据库生成器 > 点击库表
上下文菜单	在图表中，右键单击所需的库表 特征 列
键盘快捷键	F9

更改列顺序

节	行动
1	在“列”选项卡中，单击列表中所需的列名。
2	右键单击并选择： <ul style="list-style-type: none">“上移列 <名称>”选项（或按 Ctrl+向上箭头）将列上移一位“将列 <名称> 向下移动”选项（或按 Ctrl+向下箭头）将列向下移动一个位置 这些选项在“列”选项卡和图表中都会立即生效。

在数据库生成器中工作

加载数据模型后，数据库构建器会创建一组逻辑文件夹，每个文件夹对应当前 DBMS 支持的每种object类型。每个逻辑文件夹都填充有在数据模型的包层次结构中的所有该类型的对象（如浏览器窗口所示）。



在此图像中，数据模型'Orders (postgres)' 显示了用于表、视图、函数、序列、查询和连接的逻辑文件夹。值得注意的是，没有“过程”文件夹，因为 PostgreSQL 不支持数据库过程。

在数据库生成器树行动可用

大多数数据库生成器功能都可以通过上下文菜单访问。树中的每个object都有自己的组基于其类型和状态的独特菜单项。此表描述了可用的上下文菜单项并确定它们适用于哪些对象。

菜单选项	适用于/描述
新数据模型	适用于：空白空间 打开“从模式创建首页”选项卡（模型向导描述）。
全部刷新	适用于：空白空间 描述：重新加载数据模型的完成列表。
加载	适用于：根 描述：加载数据模型的全部细节。

卸下	适用于：根 描述：卸载数据模型的全部细节。
导入数据库架构	适用于：Loaded根 描述：使用当前活动连接作为实时数据库源打开“导入数据库模式”对话框。
生成DDL	适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包 描述：打开“生成DDL”对话框并选择当前object。
显示差异	适用于：加载根、文件夹、库表、视图、过程、函数、序列 描述：将选定对象与当前活动连接进行比较。
显示与选项的差异	适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包 描述：将选定对象与当前活动连接进行比较，并根据指定的比较选项选择性地忽略一些差异。
管理 DBMS 选项	适用于：Loaded根 描述：打开“管理 DBMS 选项”对话框，可用于更改分配的 DBMS 和拥有着多个对象。
视图记录数	适用于：库表、视图 描述：构建并运行一个SELECT查询（格式化以适合元素的 DBMS）以显示所选库表或视图中的记录数。 如果没有活动连接，系统会提示您选择一个。
视图前 100 行	适用于：库表、视图 描述：构建并运行一个SELECT查询（格式适合元素的 DBMS）以显示所选库表或视图的前 100 行。 如果没有活动连接，系统会提示您选择一个。
视图前 1000 行	适用于：库表、视图 描述：构建并运行一个SELECT查询（格式适合元素的 DBMS）以显示所选库表或视图的前 1000 行。 如果没有活动连接，系统会提示您选择一个。
视图所有行	适用于：库表、视图 描述：构建并运行一个SELECT查询（格式化以适合元素的 DBMS）以显示所选库表或视图的所有行。 如果没有活动连接，系统会提示您选择一个。
属性	适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包、连接 描述：打开选定object的标准“属性”对话框。
在项目中查找浏览器	适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包、# 查询、连接 描述：在浏览器窗口中查找选定object。
刷新	适用于：Loaded根 描述：重新加载当前加载的数据模型的详细信息。当其他用户添加、更改或

	删除对象或在数据库构建器之外执行更改时，这是必需的。
添加新的<type>	适用于：文件夹、库表、视图、过程、函数、序列、包、# 查询、连接 描述：创建指定类型的新object。
克隆<名称>	适用于：文件夹、库表、视图、过程、函数、序列、包、# 查询、连接 描述：制作所选object的新副本。当您选择此选项时，会显示一个提示，您可以在其中设置新object的名称和所有者。对于库表对象，您可以选择应该复制哪些现有约束（并为每个约束设置一个名称）以及应该复制哪些外键。对于基于SQL的对象，您可以对# 为新元素。
删除<名称>	适用于：库表、视图、过程、函数、序列、包、# 查询、连接 描述：从存储库中永久删除选定object。
在 <name> 上添加新的外键	适用于：库表 描述：在选定的库表和另一个库表之间创建一个新关系，然后显示新关系的外键约束"屏幕。
# 物件属性	适用于：视图、过程、函数、序列 描述：打开'# 物件编辑器的屏幕。
编辑	适用于：# 查询 描述：加载# (在选定的元素中定义)放入SQL便签本。
运行	适用于：# 查询 描述：加载# 在SQL便签本并运行它。 如果没有活动连接，系统会提示您选择一个。
设置为活动数据库连接	适用于：连接 描述：将选定的数据库连接标记为当前会话的活动连接。
设置为默认数据库连接	适用于：连接 描述：每次加载数据模型时，将选定的数据库连接标记为活动连接。
数据库连接属性	适用于：连接 描述：打开"数据库属性"屏幕，管理连接设置。

创建/编辑/删除数据库对象

本节列出的页面详细描述了细节使用数据库生成器的界面来创建和操作数据库表；但是，创建和操作基于 SQL 的数据库对象的过程记录在其他领域。有关详细信息，请参阅这些主题：

- [Database Views](#)
- [Database Procedures](#)
- [Database Functions](#)
- [Database Sequences](#)
- [Database Connections](#)

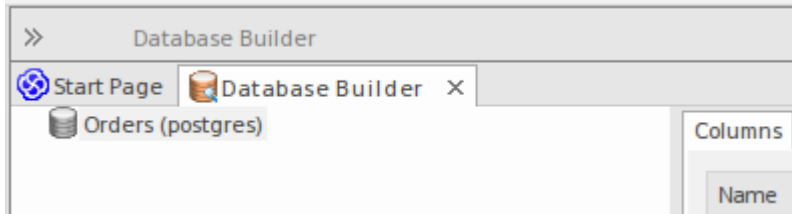
数据库构建器中的数据库连接

执行某些任务（例如“比较”或“执行 DDL”）时，数据库构建器需要一个活动的数据库连接。在给定时间，只能激活一个数据库连接（由彩色“数据库连接”图标指示，而其他数据库连接为灰色）。如果数据库连接当前未处于活动状态，并且您尝试执行需要连接的任务，则数据库构建器会根据定义的连接数执行以下操作之一：

- 0 连接 - 提示您创建连接，如果成功，则继续
- 1 连接 - 将其设置为活动并继续
- 2 个（或更多）连接——提示您选择一个，如果成功，继续

打开数据库生成器

当您第一次打开数据库构建器时，它会在整个项目中搜索所有具有构造型 <<Data Model>> 的包，并将相应的数据模型作为根加载到树中。灰色图标表示未加载数据模型A详细信息。



此图显示了具有单个未加载数据模型的数据库构建器，称为“订单 (postgres)”。

使用数据库生成器

您可以通过以下两种方式之一开始在数据库生成器中工作：

任务	行动
创建新数据模型	打开数据库构建器视图后，在树的空白处单击鼠标右键，选择“新建数据模型”以调用模型向导（“首页”从“模式创建”选项卡）。
加载现有数据模型	打开数据库构建器视图后，通过以下任一方式加载任何定义的数据模型： <ul style="list-style-type: none"> • 右键单击名称并选择“加载”，或 • 双击名称

数据属性模型

在早期版本的Enterprise Architect中（在引入数据库生成器之前），数据建模器有必要在允许某些任务之前手动设置属性对象。例如，Enterprise Architect不允许在未首先为库表分配DBMS的情况下定义库表列。这是因为DBMS控制可用数据类型的列表。

为了提高效率和用户体验，数据库构建器在数据模型级别定义了许多属性的默认值，然后在创建新对象时自动应用这些默认值。

属性

选项	描述
数据库管理系统	定义对象：数据模型的<<Database>>包 定义为：标记值 详细信息：定义当前数据模型的DBMS 额外的信息： <ul style="list-style-type: none"> • 控制在数据库构建器树中为当前数据模型显示哪些逻辑文件夹 • 控制在数据库比较期间应用哪些DBMS规则

	<ul style="list-style-type: none"> • 自动分配给当前数据模型中创建的每个新数据库object
默认所有者	<p>定义对象：数据模型的<<Database>>包</p> <p>定义为：标记值</p> <p>详细信息：定义当前数据模型的默认拥有者</p> <p>额外的信息：</p> <ul style="list-style-type: none"> • 如果 DBMS 支持所有者/模式，则自动分配给当前数据模型中创建的每个新数据库object
默认连接	<p>定义对象：数据模型的<<Database>>包</p> <p>定义为：标记值</p> <p>详细信息：（可选）默认连接的名称</p> <p>额外的信息：</p> <ul style="list-style-type: none"> • 每当加载数据模型时，都会检查“默认连接”属性；如果存在，该名称的连接将自动激活 • 数据库工程模型模式没有为此属性定义值，只要用户将连接设置为默认值，就会创建或更新它

注记

- 如果打开数据库构建器时在浏览器窗口中选择一个数据模型，将自动加载该模型的详细信息

SQL便签本

SQL便签本提供了一种机制来开发和运行ad-hoc #

针对实时数据库的查询。在开发数据模型时，您可能想要执行和测试 ad-hoc #

查询 DDL 脚本，或在实时数据库上运行查询；所有这些都可以在Enterprise Architect数据库生成器界面中实现。

SQL便签本要求数据库构建器具有与实时数据库的有效连接。此数据库连接在数据库构建器的“SQL便签本”、“数据库比较”和“执行 DDL”选项卡之间共享。

Scratch Pad 包括：

- 提供导入、保存、执行和清除#
功能A工具栏#
查询
- 一个编辑器面板，您可以在其中创建或导入#
查询 - 此面板为当前数据模型提供基于 SQL 的语法突出显示
- 由两页组成A选项卡式面板，一页显示执行查询的结果，一页显示执行期间生成的任何消息

访问

打开数据库生成器窗口，然后显示“SQL便签本”选项卡。

功能区	开发 > 数据建模 > 数据库 > SQL便签本
-----	--------------------------

便签本工具栏



Scratch Pad Toolbar 上每个按钮的功能在此表中进行了描述，从左到右。

按钮	行动
运行#	执行# 当前显示在查询Pad 中。 为执行查询的输出选择“结果”和“消息”选项卡。
新的	清除# 查询编辑器字段，以便您可以输入新查询。
打开	加载一个# 从文件中查询。 A源文件浏览器显示，默认显示# 文件。单击文件名和打开按钮以在便笺簿中显示文件内容。
保存到# 查询	保存这个# 对# 的声明# 它来自的查询object 。

保存到新# 查询	创建一个新的# 查询object并将此语句保存到该object。
保存到文件	将当前显示的查询保存到它来自的文件。 如果您从头开始创建查询，则会显示一个源文件浏览器，您可以在其中键入新文件名并单击“保存”按钮以保存查询。
保存到新文件	将当前显示的查询保存到新的 .sql 文件。 A源文件浏览器显示您在其中键入新文件名并单击“保存”按钮以保存查询。
清除	清除便笺本的内容。 Scratch Pad 中显示的任何内容都会保留在那里，直到您将其替换为另一个来自文件的查询或关闭模型查询。
切换注解	应用# 注释字符 '--' 到每个选定行的开头，或者，如果选定的行已被注释，则删除注释字符。或者，按 Ctrl+Shift+C。
语句分隔符	用于标记每个语句结束的字符中的类型。
帮助	在# 上显示帮助# 查询便笺。
查询描述	显示一个标签，提供当前# 的描述# , 是否有挂起的更改 (由前导 * 表示)，以及已加载的名称# 查询object或文件名。

注记

- SQL便笺本不会操纵您的#
无论如何，因此您必须对当前 DBMS 使用正确的语法
- 而SQL便笺本可以执行多个#
语句，并且每个语句的状态和消息都显示在“消息”列表中，一次只能在“结果”列表中显示一个SELECT语句的结果；所有后续的SELECT语句都将被忽略

执行 DDL

执行 DDL”选项卡提供了一种机制，可以轻松地针对实时数据库执行生成的 DDL 语句，并提供有关其成功的即时反馈，所有这些都 Enterprise Architect 界面内，无需其他产品。

Enterprise Architect 可以生成两种不同类型的 DDL 语句并将其发送到 执行 DDL”选项卡：

- 创建由生成 DDL 屏幕创建的 DDL 语句，以及
- 由数据库比较窗口创建的更改 DDL 语句

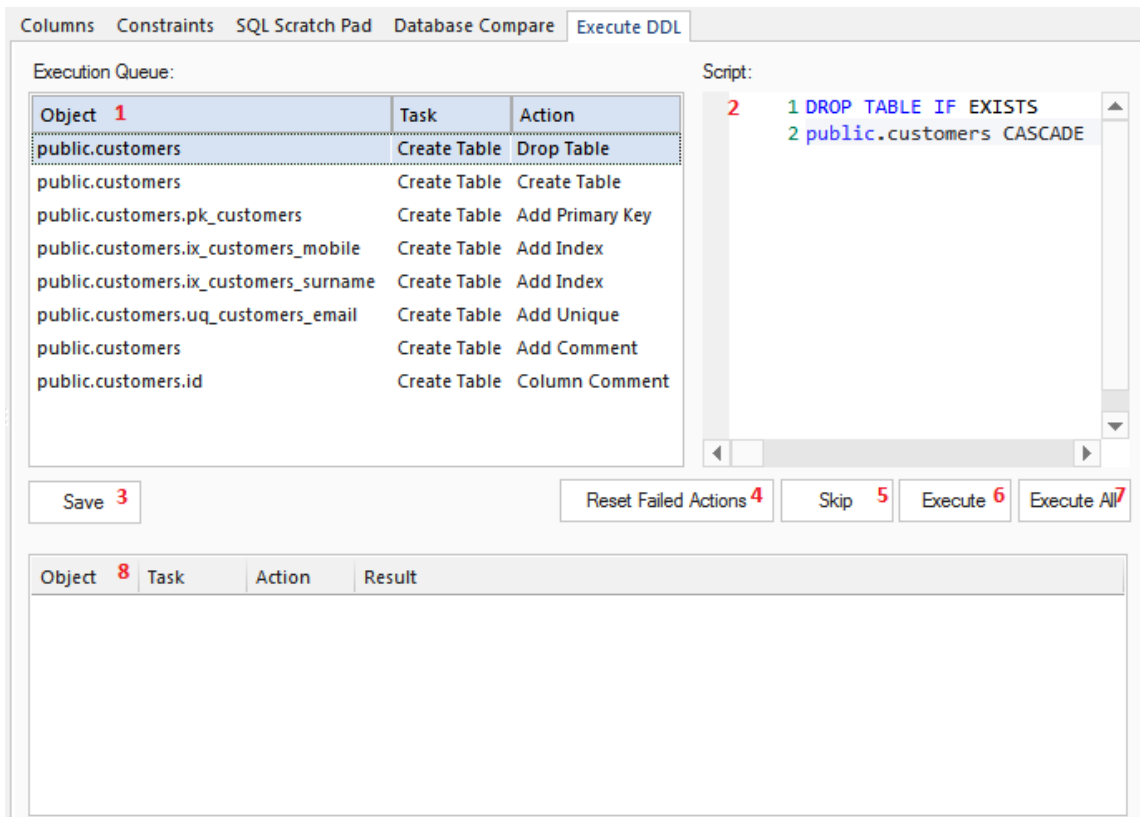
执行 DDL 功能需要数据库生成器与实时数据库建立有效连接。此数据库连接在数据库构建器的 SQL 便签本、数据库比较和 执行 DDL”选项卡之间共享。

访问

打开数据库生成器窗口，然后显示 执行 DDL”选项卡。

功能区	开发 > 数据建模 > 数据库 > 执行 DDL
-----	--------------------------

执行 DDL



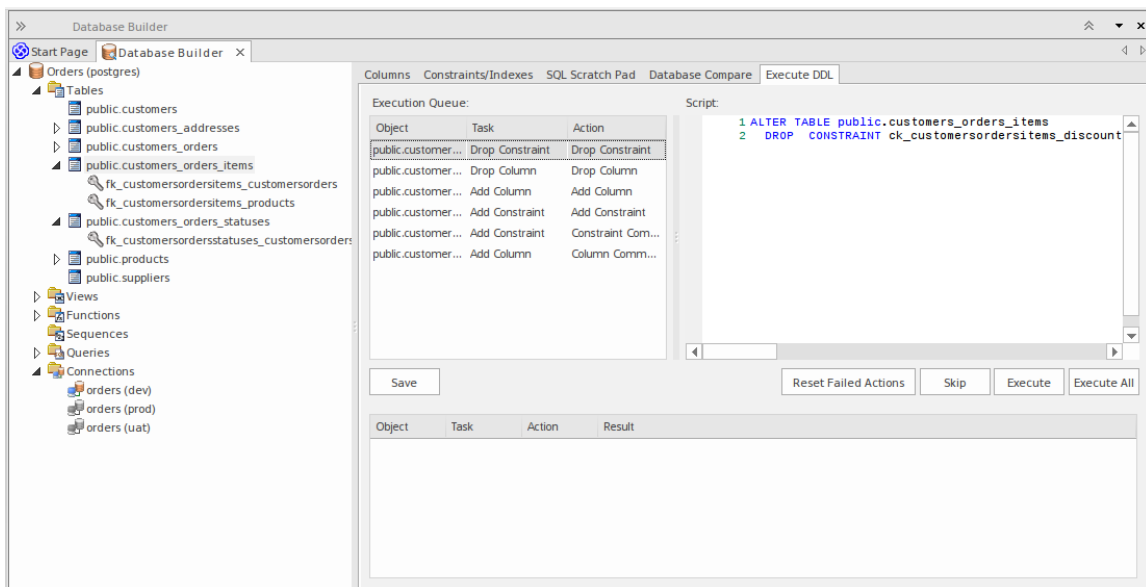
执行 DDL”选项卡具有以下字段和按钮：

字段/按钮	行动
1 执行队列	列出尚未执行的任务（每个都有关联的 DDL 语句）。该列表包含三列，指定所涉及 object 的名称、任务和正在执行的操作。

	在列表中选择个项目将显示给定任务的关联 DDL 语句 (在 “脚本” 字段中) 。
2 脚本	带有 # A 文本框 # 语法高亮显示，显示所选任务的 DDL 语句。
3 保存	单击此按钮可将 “执行队列” 和 “结果列表” 中的所有单个 DDL 语句保存到单个文件中。
4 重置失败行动	单击此按钮可将任何失败或跳过的任务从 “结果列表” 重新排入 “执行队列” 底部。
5 跳过	单击此按钮可跳过 “执行队列” 中的下一个任务而不执行它。该任务将被移动到 “结果列表” 中，并且不会给出结果。 当您单击 “重置失败的行动” 按钮时，跳过的任务将与任何失败的任务一起返回到执行队列。
6 执行	单击此按钮以执行 “执行队列” 中的下一个任务。该任务从 “执行队列” 的顶部删除，并与执行结果一起添加到 “结果列表” 的末尾。
7 全部执行	单击此按钮以执行 “执行队列” 中的所有任务。当执行完成时，“结果列表” 将显示每个单独任务的结果。
8 个结果列表	列出已执行的任务以及每个任务的执行结果。在此列表中选择个项目将在 “脚本” 字段中显示已执行的 DDL 语句。

示例

在前面关于数据库比较的部分中使用的示例中 (当重命名列和约束时)，如果使用默认值将数据模型更改 推送 “到实时数据库中，则执行 DDL 屏幕将填充此处显示的详细信息。



总之，生成 DDL 以删除旧列和旧约束（任务“删除列”和“删除约束”），然后使用新名称创建列和约束（任务“添加列”和“添加约束”），最后每个人都应用了他们的评论/备注（任务“添加约束-约束注解”和“添加列-注解”）。

数据库比较

数据库比较”选项卡提供了一种将当前数据模型 实时数据库进行比较的机制，并可选择同步任一方向的任何差异。使用 “Alter DDL” 语句执行 “推送”到实时数据库的差异，而从实时数据库导入的更改可以直接 “拉”到模型中。

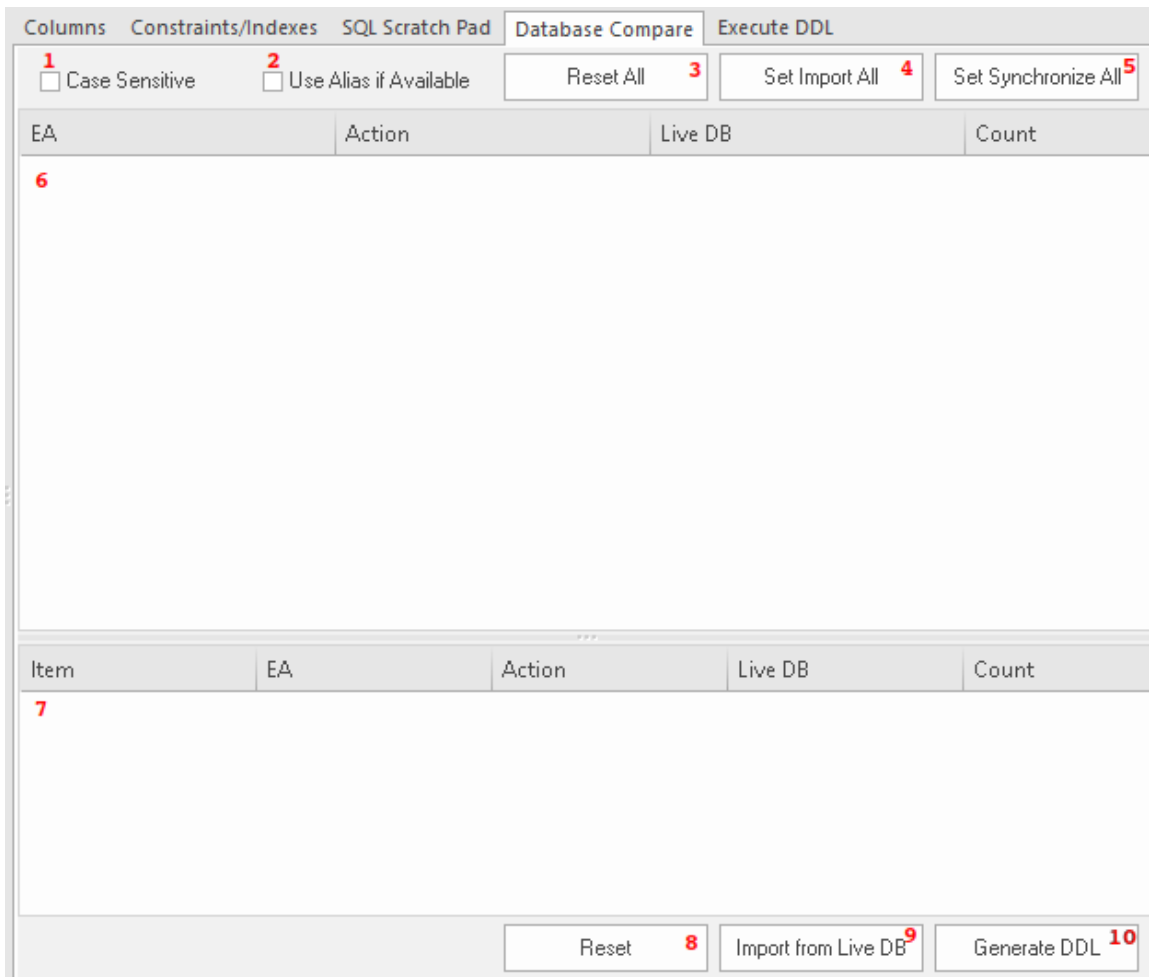
数据库比较功能要求数据库生成器与实时数据库建立有效连接。此数据库连接由数据库构建器的 “SQL便签本”、“数据库比较”和 “执行 DDL”选项卡共享。

访问

打开数据库生成器窗口，然后显示 “数据库比较”选项卡。

功能区	开发 > 数据建模 > 数据库构建器 > 数据库比较
-----	----------------------------

DDL 比较选项卡



“数据库比较”选项卡有许多控件，如此处所述。

编号和名称	描述

1区分大小写	单击此复选框可使所有属性比较识别属性文本中字母大小写的差异。
2使用别名 (如果可用)	单击此复选框以指示应使用任何已定义的别名而不是object名称 (在object和列级别) 。
3 全部重置	单击此按钮可将所有对象的 “行动” 标志设置回默认值。
4 设置全部导入	单击此按钮可将所有检测到的差异的 “行动” 标志设置为<====; 也就是说, 使用实时数据库中的值更新模型。
5 设置全部同步	单击此按钮可将所有检测到的差异的 “行动” 标志设置为====>; 也就是说, 使用模型中的值更新实时数据库。
6 差异	审阅发现模型与实时数据库不匹配的对象列表。在此列表中选择个项目将填充 “组件” 列表。 (有关每列的详细说明, 请参阅差异列表表。)
7 个组件	审阅此模型和实时数据库之间不同的所选object的属性列表。 (有关每列的详细说明, 请参阅部件列表表。)
8 重置	单击此按钮可将当前object的所有属性的 “行动” 标志设置回默认值。
9从 Live DB导入	单击此按钮可将实时数据库中的所有属性值 (带有 <==== 的 “行动”) 导入模型。
10生成DDL	单击此按钮可为所有具有====> '行动'的对象生成'Alter DDL'语句, 并将语句发送到'Execute DDL'选项卡。

差异列表

柱子	描述
EA	显示模型中检测到一个或多个差异的每个object的名称。空白值表示模型中缺少object, 但在实时数据库中存在。
行动	<p>考虑到此object的差异, 默认为 “无行动” 作为要采取的行动。单击下拉箭头并选择特定操作。列表中可用操作的列表将取决于给定object是否在模型和实时数据库中配对。</p> <p>配对对象</p> <ul style="list-style-type: none"> • 无行动- 不更新数据库或模型与此更改 • ====> - 从模型更新数据库中的object • <==== - 从数据库更新模型中的object • 自定义 - 在下方面板中对每个项目设置不同的操作之前, 将项目设置为无行动 • Unpair - 将配对的对象分开, 这样它们就不会相互比较或相互更新 <p>未配对的对象</p> <ul style="list-style-type: none"> • Create < object name> - 在数据库或模型中创建缺失的数据库object, 视情

	<p>况而定</p> <ul style="list-style-type: none"> • 删除< object名称> - 从模型中删除object • Drop < object名称> - 从数据库中删除object • Pair with < object name> - 将数据库中的object与模型中的命名 (未配对) object配对，以便比较它们之间的差异 <p>组件列表” (下部面板) 中的 “行动” 字段将根据该字段的选择进行更新。</p> <p>例如，如果实时数据库有一个库表列 “地址1” 而模型没有，则将object “行动” 设置为 “==>” (从模型更新数据库中的object) 设置列 “项移动” 到 “Drop行动”，这将从实时数据库中删除该列。</p>
实时数据库	显示实时数据库中检测到一个或多个差异的每个object的名称。空白值表示该object存在于模型中，但在实时数据库中缺失。
数数	显示在模型和实时数据库之间检测到的object (及其所有组件) 差异的总数。

部件列表

柱子	描述
物品	显示每个检测到的差异的组件名称或描述。差异分为三类：属性、列和约束，采用树形结构。
EA	显示模型中检测到的给定组件的值。空白值表示模型中缺少该值，但在实时数据库中不存在。
行动	<p>默认为与 “差异” 列表中object “行动” 字段的设置相对应的操作，以指示针对为组件检测到的差异要采取的操作。单击下拉箭头以选择替代操作；列表中的可用选项取决于组件的类型和检测到的差异。</p> <ul style="list-style-type: none"> • 无行动——不更新数据库或模型 • ==> - 从模型更新实时数据库中的object • <=== - 从实时数据库更新模型中的object • 添加 <项目名称> - 在数据库或模型中创建缺少的项目，视情况而定 • 删除<项目名称> - 从模型中删除项目 • Drop <item name> - 从实时数据库中删除该项目
实时数据库	显示活动数据库中选定组件的值。空白值表示该值存在于模型中，但在实时数据库中缺失。
数数	显示在所选组件中检测到的模型和实时数据库之间的差异数量。

使用数据库比较

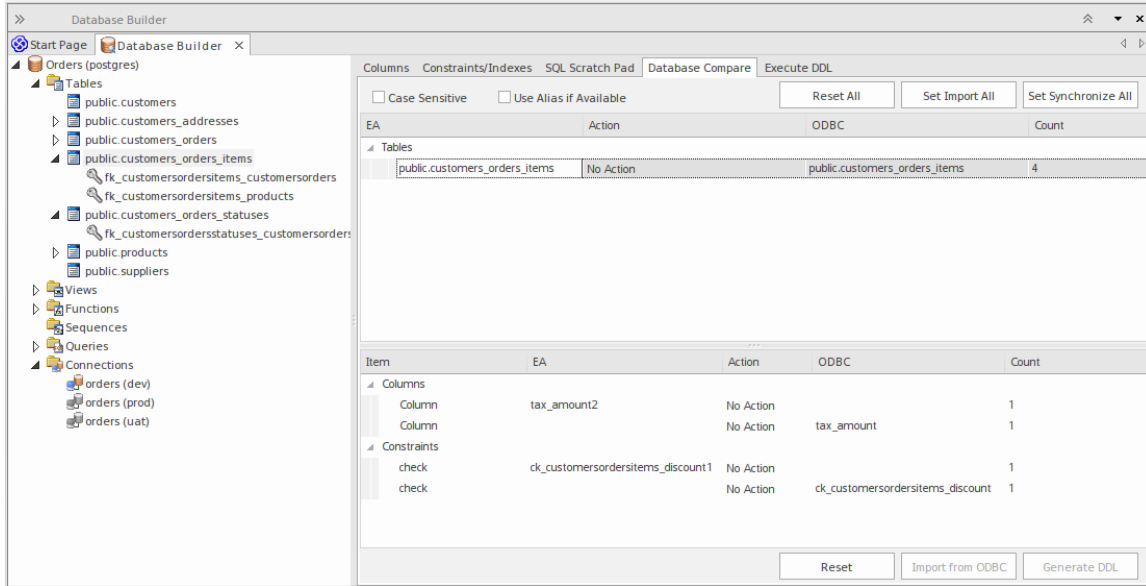
每当您执行比较时，Enterprise Architect都会从实时数据库和模型中读取定义，然后尝试使用其名称 (和模式，如果与当前 DBMS 相关) 将来自一个源的每个object与另一个源 “配对”。

如果找到匹配项，则object名称将显示在 “EA” 和 “Live DB” 列中，默认操作为 “无行动”。 “计数” 列表示为object及其组件或属性找到的差异总数。

如果在系统之间未找到匹配项，则object名称显示在源列 (“EA”或 “Live DB”) 中，而另一列为空白。在这种状态下，可以将object与不同名称的object配对； “行动”下拉列表将显示可用对象。如果进行了新配对，则会比较两个对象的定义是否存在差异，结果将显示在 “组件”列表中，并选择默认操作 “====>”。

如果您在object级别选择一个动作，这将为object的所有组件和属性设置匹配动作。但是，如果您在object级别选择 “自定义”操作，您可以为每个组件确定不同的操作。

例如，列 (tax_amount) 和约束 (ck_customersordersitems_discount) 在库表 “public.customers_order_items” (在示例模型中) 中都被重命名，并执行了数据库比较；这张图片显示了发现的差异：



在图像中只有一个库表检测到差异 - 'public.customers_order_items'; 选择此项会填充 “组件”列表。从检测结果可以确定数据模型包含实时数据库不包含的列 (tax_amount2) 和检查约束 (ck_customerordersitems_discount1)，而实时数据库又包含列 (tax_amount) 和检查约束 (ck_customerordersitems_discount) 数据模型没有。

与选项比较

“选项比较”功能的工作方式与直接比较相同，只是提示您选择应执行的object /属性。这使您可以忽略当前不相关的特定差异。

这些库表描述了可以启用或禁用的不同比较。

所有对象，拥有着

比较	行动
拥有着	选择表示所有数据库对象的 “拥有着”属性应在对象 “配对”后进行比较。

库表选项

选项	行动
----	----

表	选择此父选项以启用所有库表比较选项。取消选择以禁用所有其他选项。然后，您将取消选择或选择列表中的特定选项。
库表-扩展属性	选择以指示应该比较表的扩展属性（例如 DB版本和表空间）。
库表-备注	选择表示应比较应用于表的备注。
列	选择此父选项以启用所有“列比较”选项。取消选择以禁用所有其他“列”选项。然后，您将取消选择或选择列表中的特定选项。
列 - 类型	选择以指示应比较库表列的数据类型名称。
列 - 大小	选择以指示应比较库表列的数据类型大小。
列 - 默认值	选择表示应比较库表列的默认值。
列 - 位置	选择以指示应比较库表列的位置。
列 - 非空	选择表示应该比较库表列的非空属性。
列 - 自动编号	选择以指示应比较库表列的自动编号属性（例如 AutoNum、StartNum 和 Increment）。
列 - 不匹配的列	选择以指示应比较模型和实时数据库之间不匹配的库表列。通常，这些列存在于一个系统中，但不存在于另一个系统中。
列 - 扩展属性	选择以指示应比较库表列的扩展属性（例如 Unsigned 和 Zerofill）。
专栏 - 备注	选择以指示应比较应用于库表列的备注。
约束	选择此父选项以启用所有“库表约束比较”选项。取消选择以禁用所有“库表约束”选项。然后，您将取消选择或选择列表中的特定选项。
约束-主键	选择以指示应比较与主键相关的属性。
约束-外键	选择以指示应比较与外键相关的属性。
约束-索引	选择以指示应比较与索引相关的属性。
约束-约束独特	选择以指示应比较与唯一约束条件相关的属性。
约束-约束选择	选择以指示应比较与选择约束相关的属性。
约束-库表触发器	选择以触发器应比较与库表相关的属性。
约束-无与伦比约束	选择以指示应比较模型和实时数据库之间不匹配的库表约束。通常这些是存在于一个系统中但不存在于另一个系统中的约束。
约束-扩展属性	选择以指示应比较库表约束的扩展属性（如填充因子和集群）。
约束-备注	选择以指示应比较应用于库表约束的备注。

注记

- 数据库比较功能目前可以对库表、视图、过程、函数和序列object类型进行比较

创建一个数据模型图表

要对关系模型的结构进行建模，您可以使用数据建模图，它是扩展类图。当您打开数据建模图时，工具箱图表其中包含图表元素：


- 库表
- 视图
- 过程
- 序列
- 函数
- 关联和
- 连接数据库

访问

使用此处列出的任何方法显示“新图表”对话框。

功能区	设计>图表> 添加图表
上下文菜单	右键单击包 添加图表 右键单击元素 添加 添加图表
键盘快捷键	Ctrl+插入
其它	浏览器窗口标题栏菜单 新图表

创建数据建模图

字段/按钮	行动
包	默认为在浏览器窗口中选择的包的名称，或者，如果父项是元素，则为包含该元素的包的名称。 如果您将图表直接添加到包中并注意到它不是正确的包，请单击  按钮并浏览正确的包。
家长	如果要图表添加到元素，则此字段会显示元素名称。
图表	该字段默认为父包或元素的名称。 如果需要，请使用您的首选名称改写默认名称。
选择自	单击此标题并选择最适合您工作区域的蓝图组和蓝图或工作空间（在本例中，信息工程 >数据库模型”）。 从面板中列出的选项中，单击“扩展”。
图表类型	单击“数据建模”。

确定	<p>单击此按钮以创建图表。</p> <p>图表视图显示空白图表，“数据建模模型”页面显示在图表工具箱中。</p> <p>将元素和连接器从工具箱拖到您的图表上，以创建您的数据模型。</p>
----	--

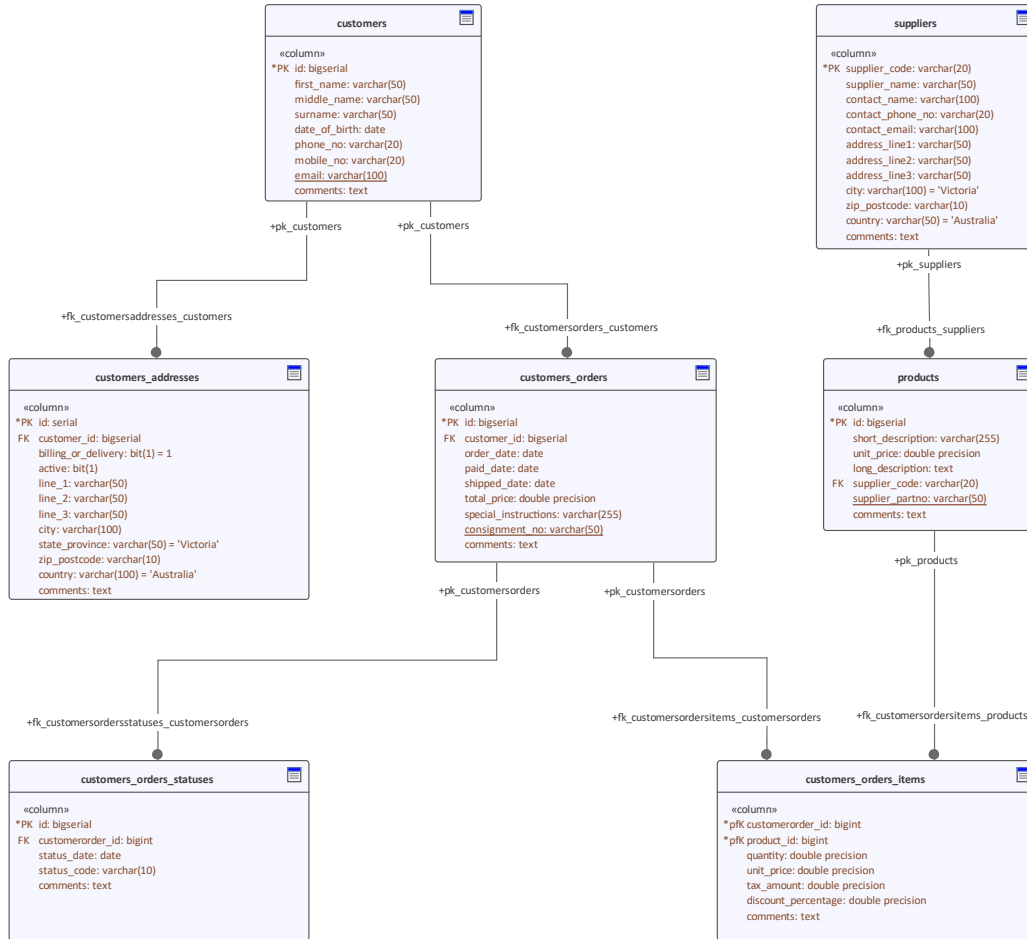
注记

- 所有新图表的默认图表连接器符号是信息工程，尽管许多数据建模者更喜欢符号 IDEF1X；要进行此更改，请选择 设计>图表>管理>属性>连接器”并单击 连接器符号”下拉列表中所需的选项

示例数据模型图表

此数据模型图示例显示了书店仓储系统的数据模型。库表是使用具有用于显示列名称和类型的列隔间的构造型类建模的。主键和外键由列上的构造型表示。您可以在示例模型中更细节地检查此模型，该模型与Enterprise Architect一起安装，可从此功能区位置获得。

开始>帮助>帮助> 打开示例模型



数据建模图，带有隐藏的操作隔间，显示连接以指示外键关系的库表。

从模型模式数据模型

创建数据建模工作的最简单方法是使用预定模式的模型数据库模型模型向导空间（首页'Create from模式）。Enterprise Architect为系统支持的每个 DBMS 提供了一个模式。

访问

使用此处列出的任何方法显示模型向导（首页 从模式创建”选项卡）。

在工程模型向导中，选择“数据库工程”蓝图。

功能区	设计>包>模型向导
上下文菜单	右键单击包 添加模型使用向导
键盘快捷键	Ctrl+Shift+M
其它	浏览器窗口标题栏菜单 模型模式

创建数据模型

字段/按钮	行动
添加到包	显示所选根包的名称。
技术	点击“数据库”。
名称	如有必要，展开模式的数据库工程组。 单击您在模型中支持的每个数据库管理系统的复选框。
全部	单击此按钮可选择所有数据库工程模型类型和实体关系图的复选框，以将它们全部包含在模型中。
没有任何	单击此按钮可清除所有选中的复选框，以便您可以单独重新选择某些复选框。
确定	单击此按钮可将您正在建模的每个数据库管理系统的包和图表添加到浏览器窗口。

每个数据建模模式提供什么

- A模型图
- 报表工件元素）（A汇总图上快速记录数据模型）
- 每个逻辑模型和物理模型的A包

- 在物理模型包中，预定义的子包层次结构，一个用于被建模的 DBMS 支持的每种object类型（例如表、视图、过程和函数）；这些会在添加数据库对象时自动组织它们
- 工作区的 DBMS 类型
- A所有者
- A包中的数据建模图，连接符设置为 IDEF1X

注记


- 创建数据建模工作区后，您可以通过以下两种方式之一开始开发您的模型：
 - 通过数据库构建器，这是一个支持数据库建模器的专用视图
 - 通过浏览器窗口和图表，这是可能适合用户的传统方法经验丰富的UML建模人员


开始

信息建模师、数据建模师和架构师负责创建跨越多个抽象级别的组织信息模型，从概念到逻辑和物理。概念模型与技术无关，可用于与业务人员和领域专家的讨论，允许表示、讨论和同意领域中的基本概念。逻辑模型对概念模型进行了详细说明，增加了更多细节和精度，但通常仍然是技术中立的，允许信息分析师讨论并就逻辑结构达成一致。物理模型将技术特定数据应用到模型中，并允许工程师讨论并商定技术决策，为生成目标环境（例如数据库管理系统）做准备。

选择蓝图

Enterprise Architect将工具的广泛特征划分为蓝图，确保您可以聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用数据建模特征，您首先需要选择以下蓝图：

 <透视名称> >数据库工程 >数据库工程

 <透视名称> >数据库工程 >实体关系

设置蓝图可确保默认情况下可以使用案例管理模型和符号图、它们的工具箱和蓝图的其他特征。

示例图表

示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述数据模型定义方式时创建的一些重要元素和连接器，包括：表、视图、过程、序列、函数。

数据模型类型

可以从通常由业务人员创建或为业务人员创建的概念模型、业务和系统分析师使用的逻辑模型以及技术人员关心的物理模型（例如数据库工程师。在本主题中，您将学习如何管理所有三个级别的信息模型。

创建和管理数据模型

在本主题中，您将学习如何使用Enterprise Architect细节您的物理数据库模式。这包括使用数据库工具，它允许您通过 ODBC 连接与任意数量的实时数据库进行交互。

导入数据库架构

本主题将连接到实时数据库，包括生产，并将数据库逆向工程为模型创建表、视图、校正参考和过程测试，并展示您如何进行测试。自动创建数据库图，库表等元素可以与模型中其他元素相关联，包括概念和逻辑模型、编程类测试等。

生成数据库定义语言 (DDL)

在本主题中，您将学习如何通过直接从模型生成数据库定义语言代码来利用数据模型的强大功能。Enterprise Architect可以将代码生成到范围广泛的数据库管理系统中。

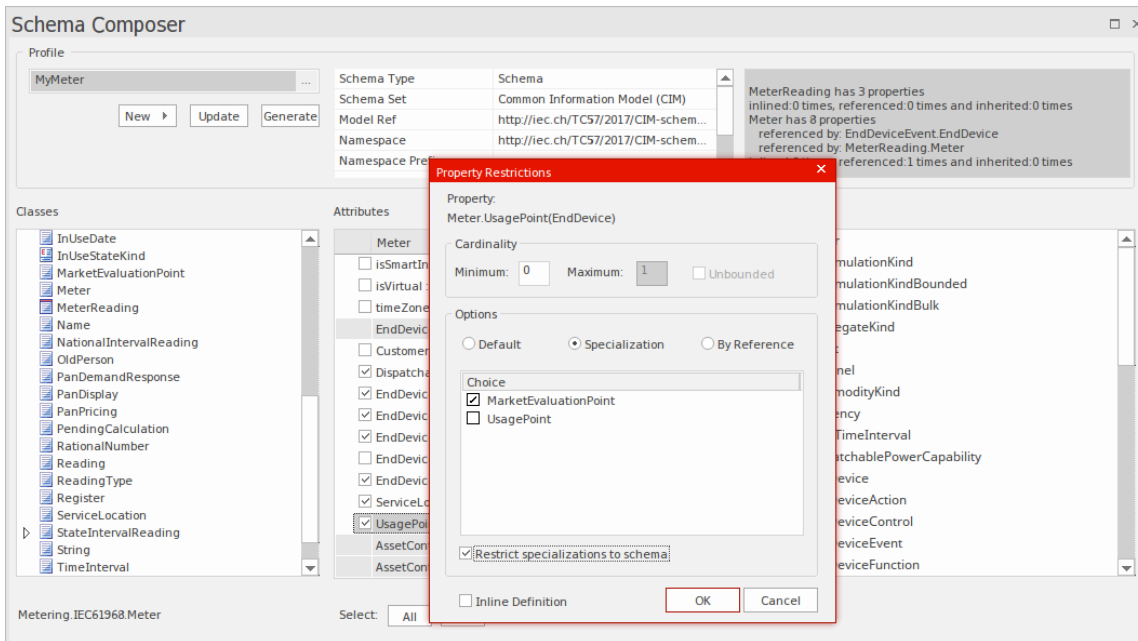
支持数据库管理系统

Enterprise Architect对大多数主流数据库管理系统 (DBMS) 都有丰富的支持。此特征允许使用导入特征来比较来自不同系统的模型以进行代码生成或分析。本主题列出了支持的 DBMS 和

更多信息

本部分提供了指向其他主题和资源的有用链接，您在使用数据建模工具特征时可能会发现它们很有用。

XML Schema (XSD)



Enterprise Architect中的结构模型，尤其是类模型，经常用于定义某些感兴趣领域的元模型。例如，可以使用类模型来定义元模型，以严格定义构成地理空间信息领域的对象、数据、关系和类型。同样，可以（并且正在）构建模型来描述诸如水管理、健康、零售、保险、汽车注册、娱乐等领域。

这些模型非常有价值，并且经常代表商业或基于标准的组织在时间和金钱上的重大投资。实现这些模型的好处的一个重要部分，特别是在必须在多方之间交换信息的情况下，是在模式的定义中（通常基于 XSD），它规定了如何形成消息以符合底层元模型。传统上，这种消息模式是基于元模型手工编写的。这通常是一项费力且容易出错的练习。

Enterprise Architect长期以来一直与基于商业和标准的元模型的开发相关联，并且在Enterprise Architect模型文件中定义了许多模型示例，用于指定感兴趣的信息域的确切构造。

Enterprise Architect中的Schema 编辑器旨在最大限度地利用存储在Enterprise Architect模型文件或存储库（或基于云的服务器）中的模型，将模型信息转换为符合命名标准和格式的模式。各种流行的行业元模型。这种方法大大减少了形成有效模式所需的时间，并消除了将模型信息转录为模式文本的人为错误。

当前版本的Schema 编辑器支持多种技术的 XSD 生成，此外还支持通过与自动化接口和插件

紧密集成来自定义输出插件

框架。以这种方式，可以使用“原样”提供的模式生成器之一或使用JavaScript编写自定义生成器，或者通过编写合适的插件

来进一步完全自定义过程插件

在选择的语言。

除了新的Schema 编辑器，Enterprise Architect还支持使用支持相关类型显式建模的UML Profiles对 XSD 和 WSDL 定义进行建模。当从头开始构建复杂的 XSD 或 WSDL 并且需要有一个完整的最终模式的可视化模型时，这有时是必要的。注记由于Enterprise Architect还支持 XSD 文档的导入，因此可以使用生成模式Schema 编辑器，然后出于文档和可视化目的（甚至进一步自定义），将该模式导入回当前或不同的模型中。

架构工程部分中包含的其他主题专门用于元物件功能（MOF）、本体定义元模型（ODM）和国家信息交换模型（NIEM）。关于NIEM的部分非常广泛，因为Enterprise Architect包括许多对模型和使用NIEM域和模式所必需的特征。与其他一些技术一样，还有一个可下载的NIEM核心版本作为Enterprise Architect模型。

Schema 编辑器

无缝模型架构兼容信息在一个简单而高效的工具中定义

Schema 编辑器是一个多功能工具，用于从模型中快速轻松地定义各种正式模式。由于 Schema 编辑器的独特性，在构建 XSD（或其他）文档的定义时，没有必要使用配置文件或原型元素。这极大地增强了底层模型的可重用性，并有助于减轻直接处理 XSD 或其他元素类型和限制时出现的复杂性。

在过去十年中，许多行业一直在努力定义特定于其行业的共享元模型，而这些模型现在构成了跨组织和跨地理边界共享合同信息的基础。Schema 编辑器的典型使用场景是创建消息定义（schema）以在组织之间交换信息，确保此类消息符合相关各方已采用的底层元模型。

当组织之间共享信息时，通常只需要完整元模型的一个子集，但共享的内容必须精确地符合商定的元模型。在这种情况下，Schema 编辑器是基于子集和受限数据集推导契约模式的完美工具，这些数据集在整个元模型中进行“切片”。

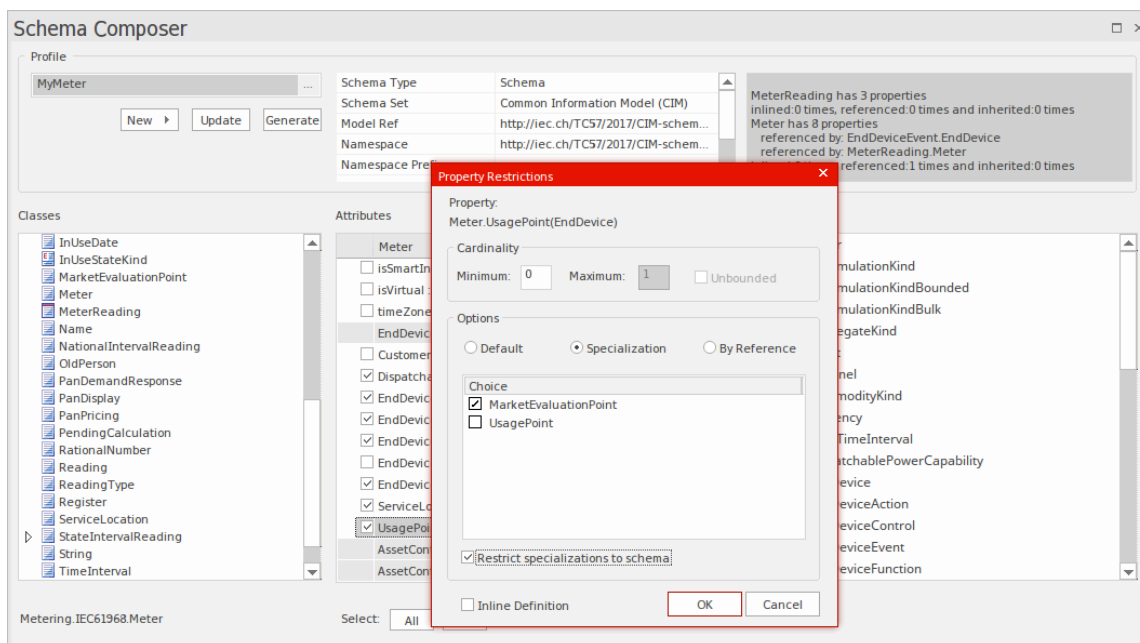
Schema 编辑器避免了直接使用 XSD 和其他模式语言的常见“痛点”：

- 除了您的“正常”业务和数据模型之外，无需创建由特定 XSD 元素组成的相对复杂的 XSD 模型来定义所需的数据、其关联和引用以及任何限制或条件
- 您无需了解如何使用 XSD 元素并应用 XSD 命名规则和约定来正确构建此类模型；支持的标准指定的格式和命名规则会自动得到处理

Schema 编辑器以可重用和可访问的方式极大地简化了创建符合标准的模式的过程。在此图中，您可以看到如何使用简单的类图作为 Schema 编辑器的源来生成 XML Schema。

Schema 编辑器在 Enterprise Architect 的企业版、统一版和终极版中

Schema 编辑器



此图显示了示例模型中进程顺序域的架构组合。

访问

--	--

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器
-----	---------------------------------------

好处

Schema 编辑器：

- 在类模型而不是 XML 模式配置文件上运行
- 让您摆脱特定于 XSD 的设计和模式生成决策，同时仍确保整个配置文件的一致性
- 可以对通用类模型进行操作以提供通用的 XSD 文档
- 在具有特定领域意义的行业标准类模型上操作时最有用
- 在大多数情况下，在一个完整模型上运行，从中抽取选定类的属性子集以构建特定消息，仅传达信息发送或请求所必需的内容
- 对于 NIEM 等标准，将生成一个新的子模型，作为更广泛的 NIEM 兼容模式定义的一部分

Schema 编辑器目前支持的标准包括：

- 公共信息模型(CIM)
- 全国信息交换NIEM建模
- 联合国贸易便利化和电子业务中心 (UN/CEFACT)建模方法 (UMM)，特别是命名和设计规则 (NDR) 2.1 和 3.0
- 通用业务语言 (UBL)，特别是命名和设计规则 (NDR) 3.0

Schema 编辑器还可以帮助您使用不同格式构建同一消息的定义，例如：

- XSD
- RDFS
- JSON

另外Schema 编辑器：

- 支持使用自定义插件实现的格式插件
它利用了Schema 编辑器自动化接口
- 内置支持不同行业模型使用的各种序列化格式和样式

Schema 编辑器Profiles

Schema 编辑器配置文件是描述构成特定模式或子模型的元素和限制的配置文件。Profiles通常与特定技术相关，例如公共信息模型(CIM) 或核心组件的UML配置文件(UPCC)，以及对配置文件中材料的解释以及发布的模式或子模型的性质将取决于所使用的特定技术生成器。虽然Enterprise Architect支持许多“开箱即用”的技术（并且正在计划更多），但也可以通过利用Enterprise Architect中广泛的自动化接口来利用Schema 编辑器Profiles的丰富内容来自定义流程根据您的条件，在插件

中插件

或脚本。

架构Profiles

Schema 编辑器配置文件A两种形式。每种形式都满足特定的系统要求——架构生成（xsd、rdfs、json）和子模型创建。当您在Schema 编辑器中创建配置文件时，您可以根据需要选择要使用的表单。Schema 编辑器中A单个配置文件可用于以常用形式组成模式，或从核心模型创建UML子模型。

配置文件类型

类型	描述
模型变换	这种类型A配置文件用于从核心模型。
架构	这种类型A配置文件用于生成模式；通常是表示消息的 XSD 模式，还有其他格式，例如 JSON object表示法和资源描述符格式。

架构组合方法论

国家信息交换模型 (NIEM)

Enterprise Architect提供了一个NIEM框架和Schema 编辑器，用于生成子模型和 XML 模式。

公共信息模型(CIM)

Enterprise Architect Schema 编辑器支持提供开箱即用的 CIM 标准，用于组合符合 CIM 的模式。

通用业务语言 (UBL)

Enterprise Architect提供了一个通用业务语言框架，以及为模式生成提供 UBL 标准的Schema 编辑器。

核心部件技术规范 (CCTS) UN/CEFACT

Enterprise Architect为 Core Components 框架和Schema 编辑器提供了UML配置文件。Composer 可以从核心组件库生成业务组件库，并简化从消息组件/业务信息实体的模式的组合/发布。

通用的

在标准不满足您的要求的情况下，通用选项为您的UML模型中的快速模式组合提供了更简单的选择。通常，您将使用具有属性、关联、聚合和继承的UML类来模型您自己的数据库。然后，您可以将此模型用作Schema 编辑器的输入。

EA脚本引擎

Enterprise Architect提供了一个脚本引擎，支持JavaScript、VBScript 和 JScript 语言。脚本引擎也与Schema 编辑器集成。在为特定标准或通用方案生成模式时，可以使用脚本自行执行操作或作为标准提供的选项的补充。

EA插件

Enterprise Architect提供插件

与Schema 编辑器集成。一个插件

可以通过向Enterprise Architect注册其兴趣来参与子模型或模式的生成。插件

可以提供要在“架构生成”对话框中列出的选项和备选方案，并且在选择其选项时将被调用。插件

可以使用Schema 编辑器自动化接口访问配置文件的内容。

创建架构配置文件

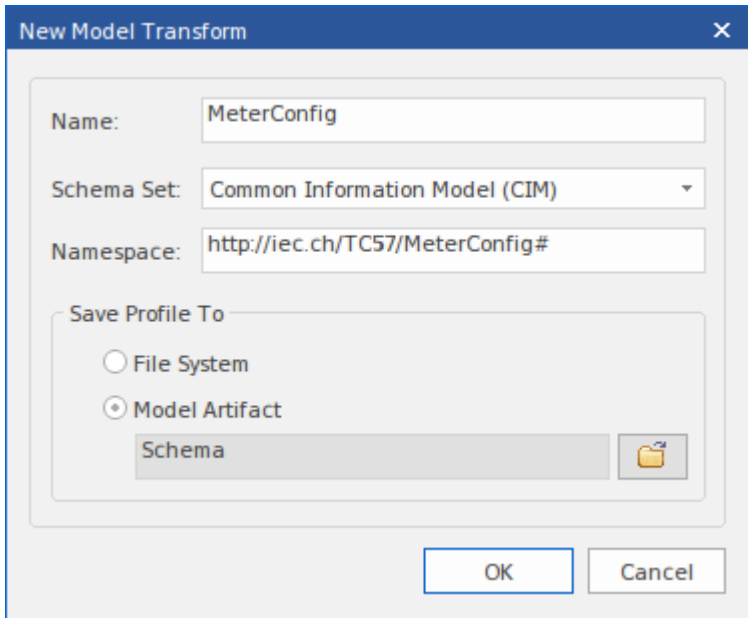
模式配置文件A模式的名称、技术和内容标识为定义如何生成模式的前兆。您可以根据需要创建和编辑任意数量的架构配置文件。架构配置文件绑定到单一技术，并将映射到生成的架构或子设置转换。

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器
-----	---------------------------------------

新建配置文件

如果您正在为特定技术创建模式，请首先打开已加载所需元模型的模型。使用Sparx Systems模型向导和/或Sparx RAS/云服务模型时，可以使用许多元模型。按照以下步骤构建新的配置文件。显示Schema 编辑器后，单击“新建”按钮并选择配置文件类型，架构或转换。



新配置文件屏幕

选项	行动
架构集	选择要使用的标准，或选择“通用”选项。
命名空间	根据您选择的标准，此字段可能采用自动值或保持空白。如果为空，请提供相关的命名空间。请参阅下一节，了解如何在Schema 编辑器中管理命名空间。
将个人资料保存到：	Profiles可以存储在文件系统或模型中。存储在模型中的Profiles可以与他人共享，而文件系统配置文件是私有的。

确定	单击此按钮可在 Composer 中编辑新模式。
----	--------------------------

命名空间

创建新配置文件时，您指定目标命名空间和命名空间前缀。模式通常涉及多个名称空间，并且 Schema 编辑器在单个模型中提供了对此的支持。标识名称空间的方案是在一个包上存在两个特定属性。属性是指定命名空间的“URI”和提供命名空间前缀的“别名”。该属性可以出现在直接包或父包上。当存在时，类的元素将采用该命名空间。在不存在命名空间的情况下，类将采用创建配置文件时指定的目标命名空间。

保存配置文件

单击更新按钮以保存您刚刚创建的配置文件。

注记

- 为 NIEM 创建和生成模式的过程在 NIEM 帮助主题中有额外的注记
- Schema 编辑器在 Enterprise Architect 的企业版、统一版和终极版中

架构组成

模式组合是指从模型中提取的一组受限制的元素，它们共同描述了模型中没有等效的唯一实体。通常，模式组合用于生成模式文件，例如 XSD 文件。相反，模型组合用于将材料配置为子集“转换”的基础 - 例如在创建 NIEM 模型子集时。

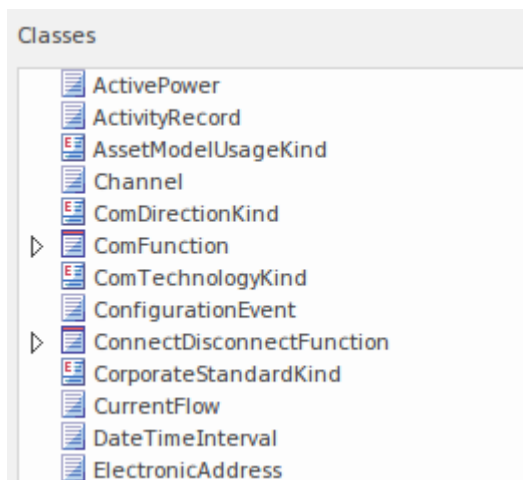
定义架构内容

这些步骤将引导您完成在架构配置文件中编写类型的基本过程，并展示如何限制元素的内容以满足消息要求。

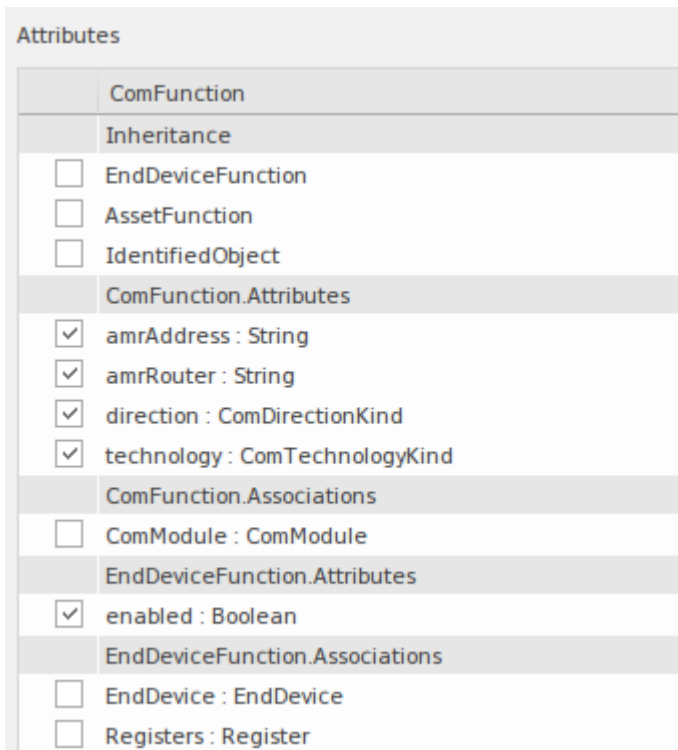
添加类

将所需的类元素从浏览器窗口拖到“类”面板中。当你添加一个类时：

- 它的祖先列在中间面板的“继承”部分
- 其属性列在“继承”部分下，每个属性对应一个空白复选框；关联和聚合条目根据连接器上的角色名称命名
- 它的模型结构路径显示在“类”面板下方



选择属性



每当您在“类”列表中选择一类时，其属性和模型祖先都会列在“属性”列表中。选中每个属性对应的复选框以定义此类型的元素。选择后，属性的类型会自动添加到架构中，出现在“类”列表和右侧的“架构”面板中。

取消选中某个属性时，不会自动删除该类型。可以使用类上下文菜单删除类型。值得注意的是，每次选择一个类时，所有对该类的引用都会显示在状态面板中，让您可以快速审阅任何类的使用情况。

```
referenced by: ConfigurationEvent.Names
inlined by: ConnectDisconnectFunction.Names
referenced by: Manufacturer.Names
referenced by: Meter.Names
referenced by: MeterMultiplier.Names
referenced by: ReadingType.Names
referenced by: Register.Names
```

继承

如果您喜欢或预见到在您准备的模式中需要继承，那么首先从祖先开始组合是有意义的，然后在添加子类时重新使用它们。该方法不是一成不变的。您可以随时从继承模型切换到聚合组合，反之亦然。这里简单介绍一下 Schema 编辑器中提供的继承。

Schema 编辑器在处理继承方面提供了灵活性。例如，您可以选择从类及其父级中聚合选定的属性，同时选择继承祖父级。但是，当您选择使用继承时，您也选择继承该类型的受限形式。在此列表中选择祖先时，生成的 XML 模式将显示标识此祖先的扩展元素。只能选择一个祖先。

单击更新按钮以验证并保存您的架构配置文件。

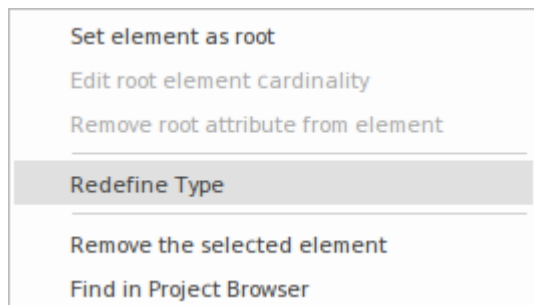
如果配置文件有任何问题，它们会在屏幕右上角的状态面板中进行标识。

重新定义的类型

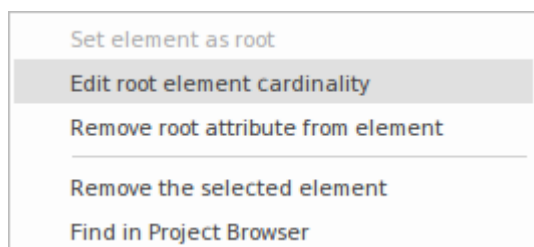
模式组合的常见问题之一是需要能够改变类型描述以满足模式描述的实例的各种需求。例如，车辆可以按其品牌、模型和价格通过卡车类型的元素来描述，但通过年份、模型和颜色可以通过轿车类型的元素来描述。问题是我们可能只有一个实际的 *Vehicle* 类可供我们使用。为了解决这个问题，Schema 编辑器允许您克隆 *Vehicle* 类并为其命名。然后，您可以将此版本的 *Vehicle* 分配属性 *Vehicle* 作为其类型的任何属性。创建的类型仅在模式域内可用，模型未受影响。

要创建类型的新定义，首先在“类”列表中选择类，然后右键单击它并选择“重新定义类型”选项。输入此类型的

唯一名称，然后按 Enter 键。然后，您可以像对任何类一样独立定义或限制此类型。



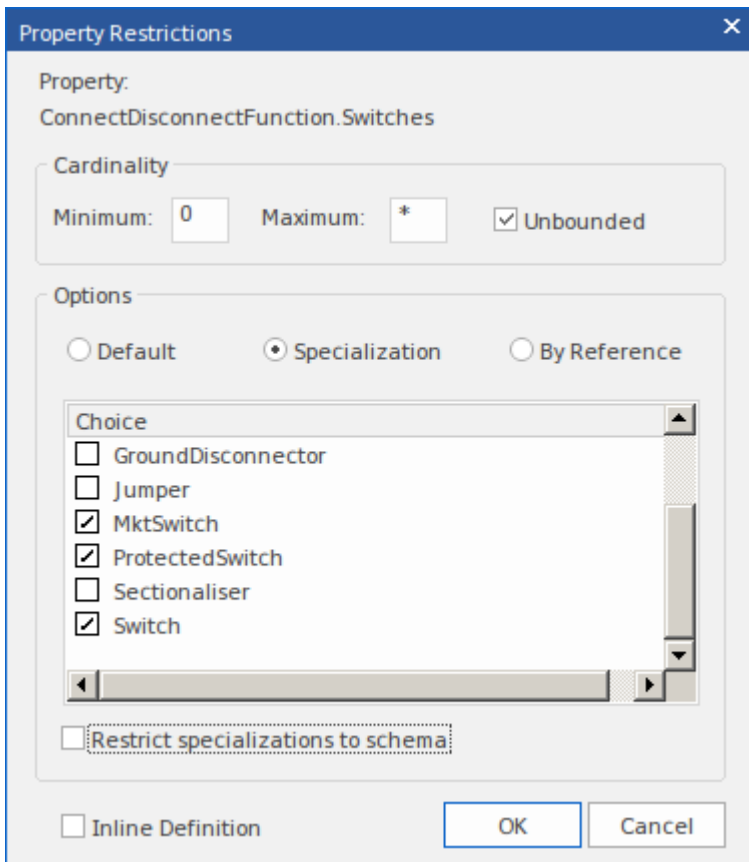
根元素



生成模式时，会生成表示消息的单个顶级元素。这个顶层元素的主体或元素是标记为根元素的类。这些根元素的基数可以调整。要将类标记为根元素或限制其基数，请右键单击列表中的类并使用以下上下文菜单选项：

- 将元素设置为根-根元素构成表示消息/配置文件的顶级元素的主体
- 编辑元素根- 设置最小和最大实例数
- 从元素中删除根属性-从类中删除根标记
- 删除选定的元素- 从架构中删除选定的元素
- 在项目中查找浏览器- 在浏览器窗口中定位并突出显示元素

属性限制



在“属性”列表中，右键单击选定的属性并使用上下文菜单添加、编辑或属性限制。使用特征用于：

- 修改属性基数
- 重新定义属性的类型
- 启用和限制可用于此属性的选项
- 标记要作为内联元素定义发出的属性
- 将要发出的属性标记为“通过参考”

基数

属性的基数可以从其对应的模型进一步限制，但不能减少限制。任何根元素类和任何类属性都可以更改基数。

类型重新定义

当类在Schema编辑器中重新定义时，它会创建一个新类型。新类型是原始类型的克隆，但具有模式唯一的名称。A · Payment 枚举类型可能被重新定义为 CardPayment 以更好地适应模式目的。新类型是对原始类型的限制，因为不能向其添加新属性。其它属性共享此类型并通过在其限制对话框中指定新类型来类似地受到限制。在其他属性的限制下，可以提供重新定义的类型（例如子类型）作为附加的选择元素。

专长

如果存在属性类型的特化，则这些子类型将在“限制”对话框中可用。When more than one specialization is selected, these will appear as choice elements in the schema. 当仅选择一个时，该属性将在模式中显示此子类型。

内联元素

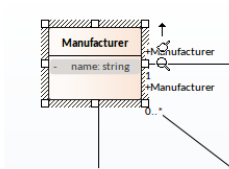
选中A框时，属性类型将作为内联定义发出。

通过参考

属性在模式中A时将采用 按引用"形式。'By reference' 表单发出一个内联 complexType，它定义了一个名为 'ref' 的类型为 'string' 的单个属性。

属性约束- Facets

Schema 编辑器通用配置文件支持构面。分面的来源是属性上的标记值。如果从XML Schema规范中命名了一个约束分面，则该标记值被识别为分面；JSON 验证关键字也被识别。



Tagged Values	
Attribute (name)	
minLength	3
whitespace	preserve
maxLength	64

```

<!--
-->
<xs:complexType name="Manufacturer">
  <xs:sequence>
    <xs:element name="name" minOccurs="1" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="64"/>
          <xs:minLength value="3"/>
          <xs:whiteSpace value="preserve"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

从XML Schema约束方面：

- 长度
- 最小长度
- 最长长度
- 图案
- 枚举
- 空白
- 最大包容性
- 最大独家
- 分钟独家
- 最少包含
- 总数
- 分数位数

JSON中的验证关键字：

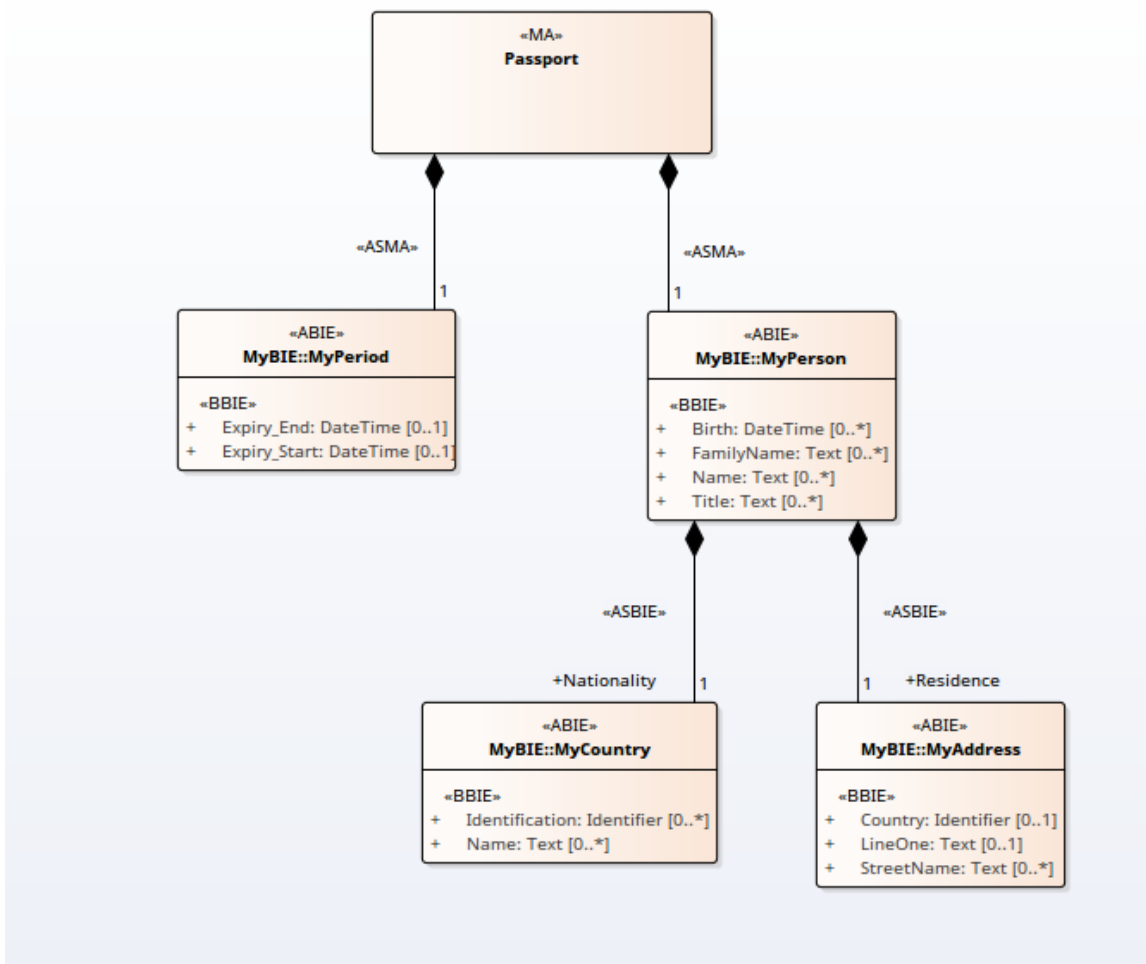
- 数字和整数
 - 倍数
 - 最低限度
 - 最大
 - 独家最低
 - 独家最大
- 字符串
 - 最小长度
 - 最长长度
 - 图案
- 数组
 - minItems
 - 最大物品
 - 独特的物品

类图表

Schema 编辑器还支持从通用UML类创建简单的 XSD 和其他格式。例如，当需要以通用方式导出类定义以供脚本或基于 Web 的工具使用时，这尤其有用。

从类图生成模式

喜欢在合成中使用建模方法的用户也可以使用Schema 编辑器来生成他们选择的格式。任何类图都可以加载到 Schema 编辑器中。此图像说明了使用核心组件的UML配置文件组成的消息，但没有必要根据特定的UML配置文件对消息进行建模。



将消息加载到 Composer

通过在表示消息的图表上选择一个类并使用其上下文菜单在Schema 编辑器中将图表呈现为模式，将消息加载到 Composer 中。选择的类将成为消息的根元素，其关系将塑造加载的模式。

这是加载到Schema 编辑器中的类图

Profile

Passport ...

New Update Generate

Schema Type	Schema
Schema Set	Core Components (UN/CEFACT) - NDR 3.0
Model Ref	My Model
Namespace	http://myauthority.org/passports
Namespace Pr...	rsm:
Unified Schema	true

Address has 5 properties
referenced by: Person.Residence
inlined:0 times, referenced:1 times and inherited:0 times

Classes

- Address
- Country
- Decimal
- MyCode
- MyDateTime
- MyIdentifier
- MyMeasure
- MyText
- Period
- Person
- String

Attributes

Address

Address.Attributes

- BuildingNumber : MyText
- CityName : MyText
- CountryName : MyText
- Postcode : MyCode
- StreetName : MyText

Schema

- Passport
 - Address
 - Country
 - MyCode
 - MyDateTime
 - MyIdentifier
 - MyMeasure
 - MyText
 - Period
 - Person

架构分析

随时随地分析

Schema 编辑器在将每种类型添加到模式时以及在选择类时执行分析。系统输出窗口将显示该类型存在多少（如果有）引用、它被继承的次数和其他有用的信息。此图显示了一条消息，详细说明了引用选定类的元素。

```
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
```

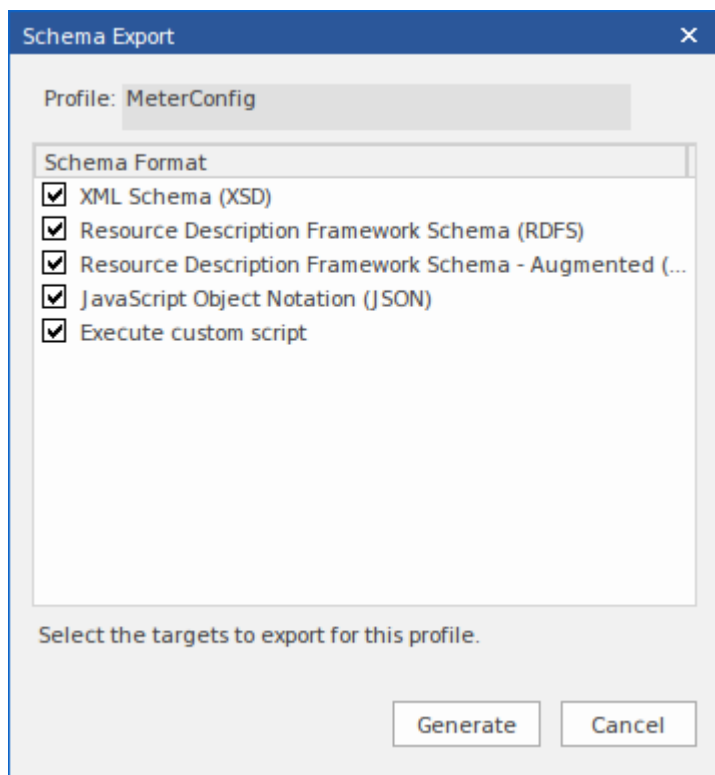
随时随地验证

Schema 编辑器对一项技术执行特定的验证。此图像显示有关基于 UN/CEFACT 核心组件标准构建的架构中的类缺少标记值的警告。

```
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
```

生成架构

在任何阶段都设计了配置文件，并且具有最少的定义和自定义，您可以快速轻松地生成模式或子模型。根据选择的技术和配置文件类型（模式或转换），呈现给您的格式会有所不同。注记通常可以一次生成多种格式。而且，当然，您可以轻松地重复该过程，随着组合的发展或模型的设计更改。



访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

注记

- Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

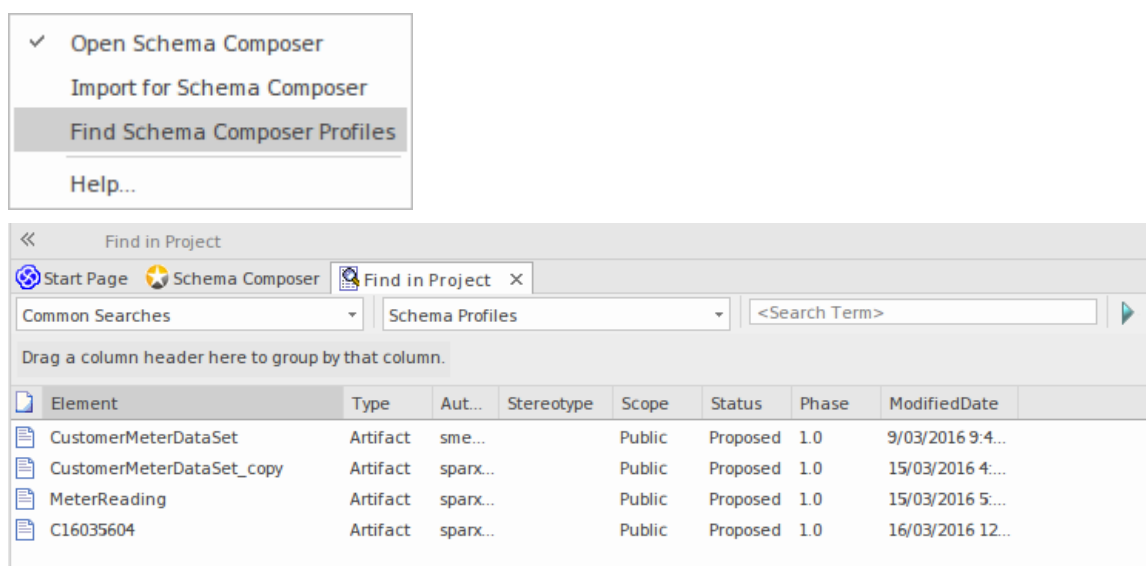
选择架构配置文件

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 查找Schema 编辑器Profiles 开发 > 架构建模 > Schema 编辑器(icon)
-----	---

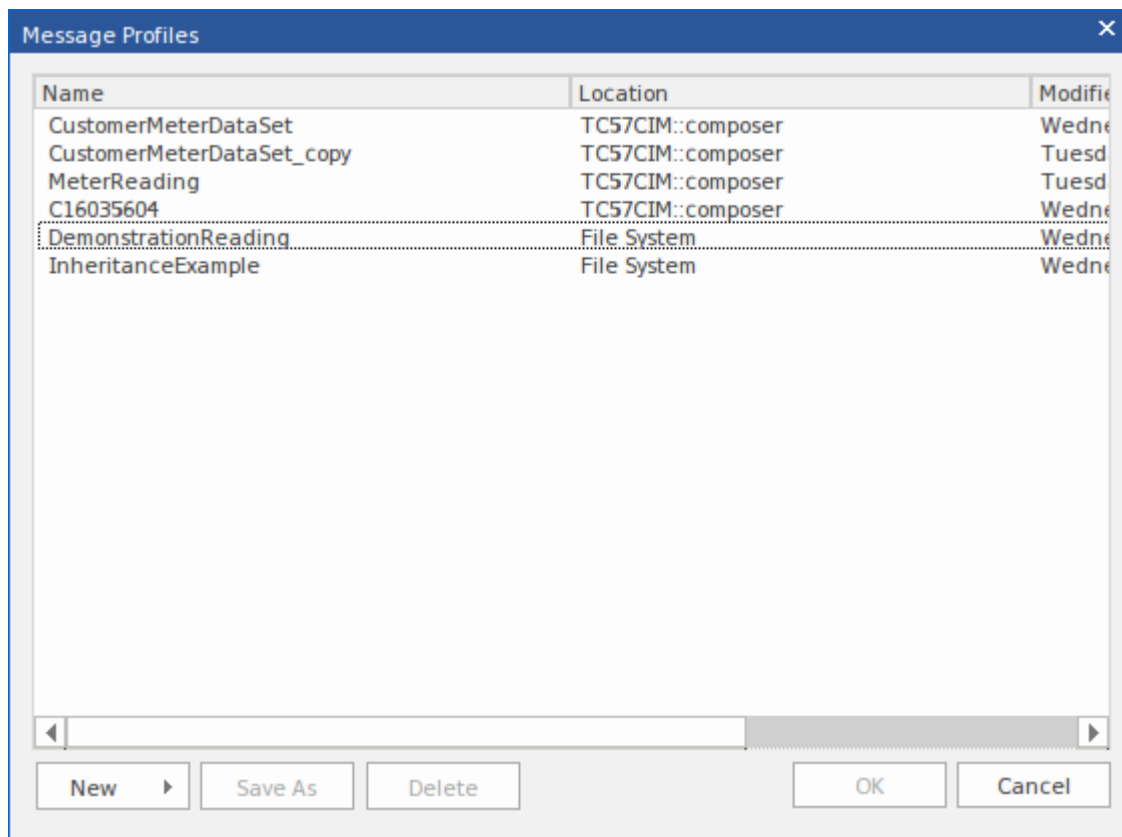
在模型中查找架构Profiles

架构配置文件可以从功能区的Schema 编辑器下拉菜单中快速定位。该菜单提供对模型中现有配置文件的快速访问。



在Schema 编辑器中定位架构Profiles

架构配置文件可以存储在模型和文件系统中。您可以通过打开Schema 编辑器并单击选择配置文件按钮（带有省略号“...”的按钮）在您的模型中创建的每个配置文件轻松找到该模型的所有配置文件列表并指示位置他们住;模型或文件系统。您可能有许多您使用的Enterprise Architect模型，但只会列出与开放模型相关的那些文件系统配置文件。



生成架构文件

定义了模式配置文件并添加了必要的元素和限制后，您可以快速轻松地生成模式。XML 模式生成适用于所有技术，但每种技术都可能支持其他格式。

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

架构格式

选中要导出的每种模式格式的复选框。

架构格式	细节
CIM	<ul style="list-style-type: none"> XML Schema (XSD) 资源描述框架架构(RDFS) 资源描述框架架构- 增强型 (RDFS) JavaScript物件表示法 (JSON) 执行自定义脚本
UN/CEFACT NDR 3.0	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
UN/CEFACT NDR 2.1	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
通用的	<ul style="list-style-type: none"> XML Schema (XSD) 资源描述框架架构(RDFS) JavaScript物件表示法 (JSON) 执行自定义脚本
UBL 2.1	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
执行自定义脚本	<p>虽然Schema 编辑器可以为许多公认的标准生成模式，但它也为那些想要控制模式的格式和特征的用户提供了一个脚本解决方案。当您为生成器指定脚本时，它指的是您的模型中存在的语言脚本，例如JavaScript。脚本产生的方式和内容几乎取决于您。脚本如何访问Schema 编辑器中的 Schema 记录在脚本 Schema Composer Scripting Integration 中。</p>

生成

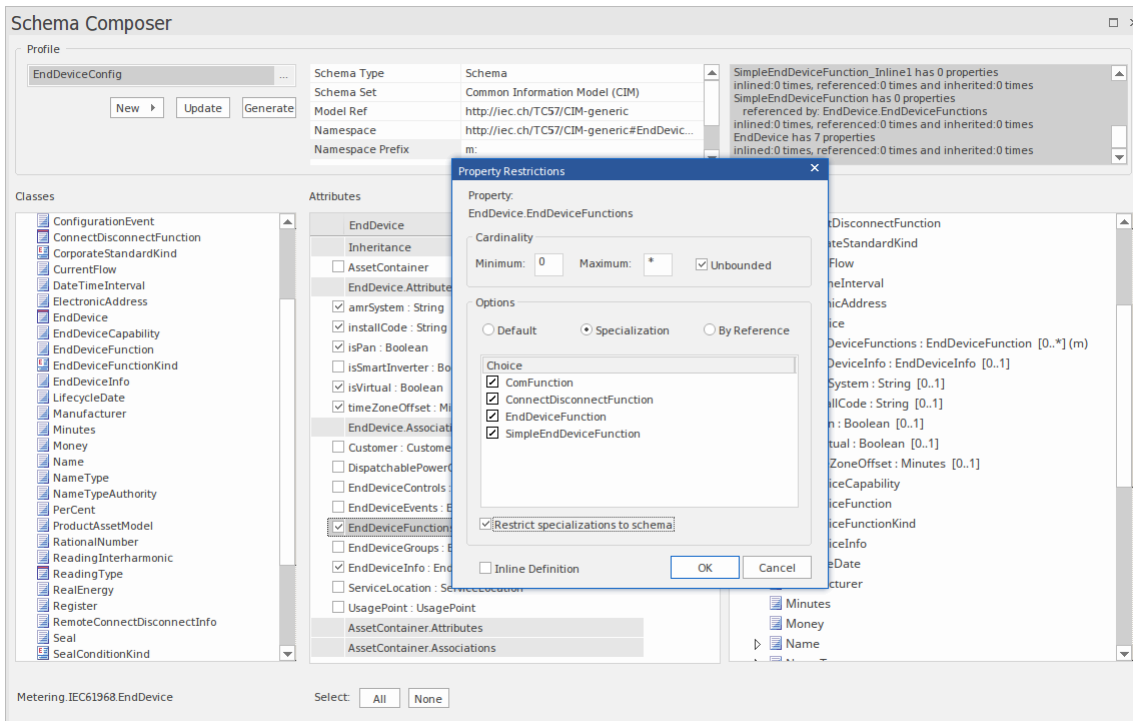
单击此按钮以生成模式。

使用文件浏览器来定位和打开模式文件。

注记

- 您可以使用Enterprise Architect编辑和验证 XML 文档，包括 XSD 架构
- 您可以将Enterprise Architect设置为 XML 文档的默认文档处理程序

CIM架构指南



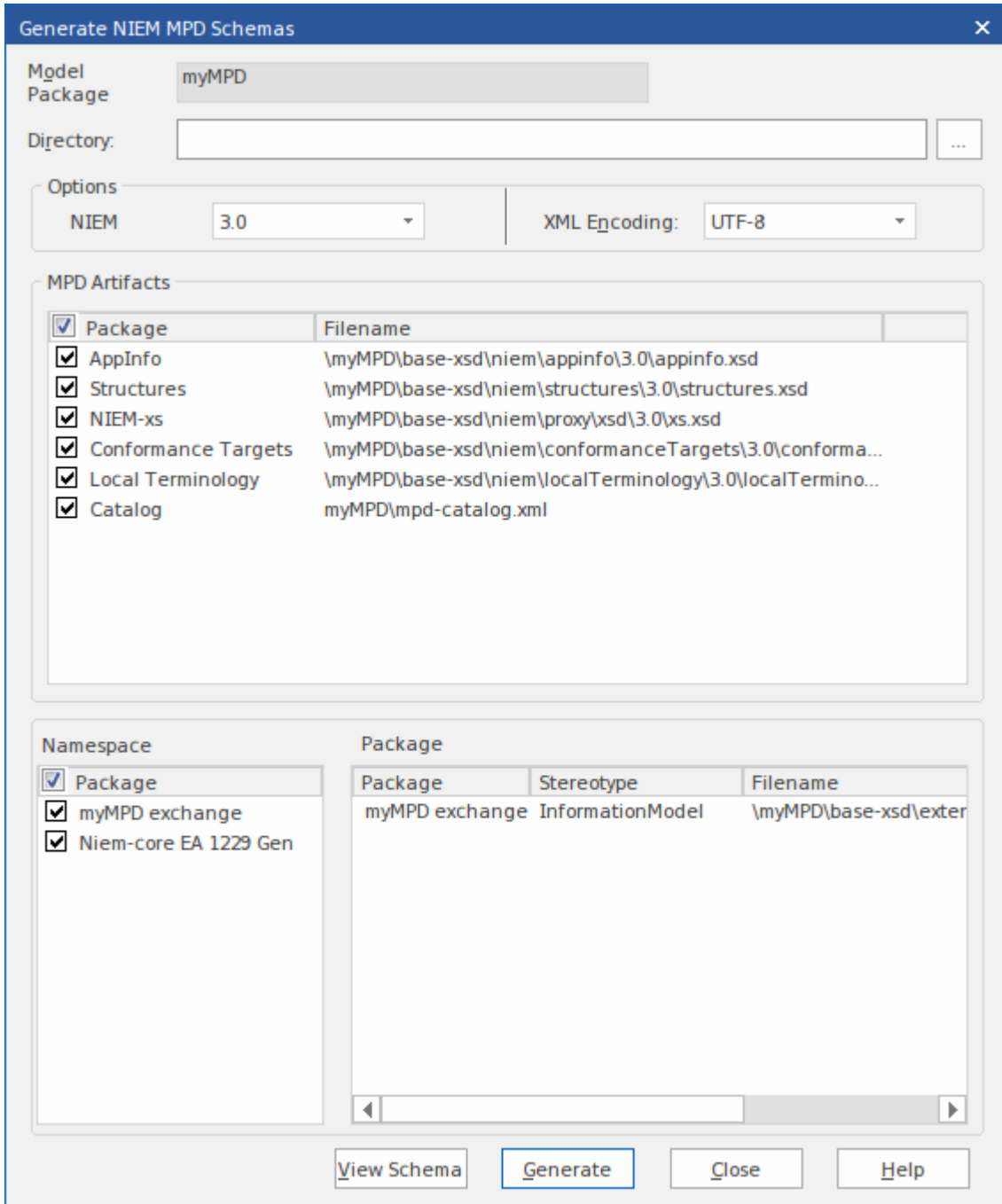
本指南描述了 CIM 兼容XML Schema的创建和生成。

创建 CIM 消息

节	行动
1	显示Schema 编辑器。
2	单击“新建 架构”。
3	输入此 CIM 模式（消息）的唯一名称。
4	选择公共信息模型。
5	将初始 CIM类拖到最能代表消息的类窗口中。使用上下文菜单适当地设置根元素。
6	如果您想使用继承来组合此类型，请从继承列表中选择一个祖先。
7	使用每个类的属性上的复选框来定义将描述此消息或模式的属性集。
8	使用属性上的上下文菜单对元素应用限制。
9	单击更新以保存消息。
10	单击生成按钮并选择要导出的模式格式。

NIEM架构指南

NIEM模式的生成是在 ModelPackageDescription类 (NIEM 3.0 及更高版本) 或 «ModelPackageDescription» 原型组件 (NIEM 2.1) 的实例上完成的。无论哪种情况，都会出现一个对话框，允许您配置生成的模式。



生成NIEM Schemas (NIEM 2.1)

单击具有 «ModelPackageDescription» 构造型的组件并选择以下选项之一：

功能区	特定>技术> NIEM 2.1 >生成NIEM 2.1架构或
-----	--------------------------------

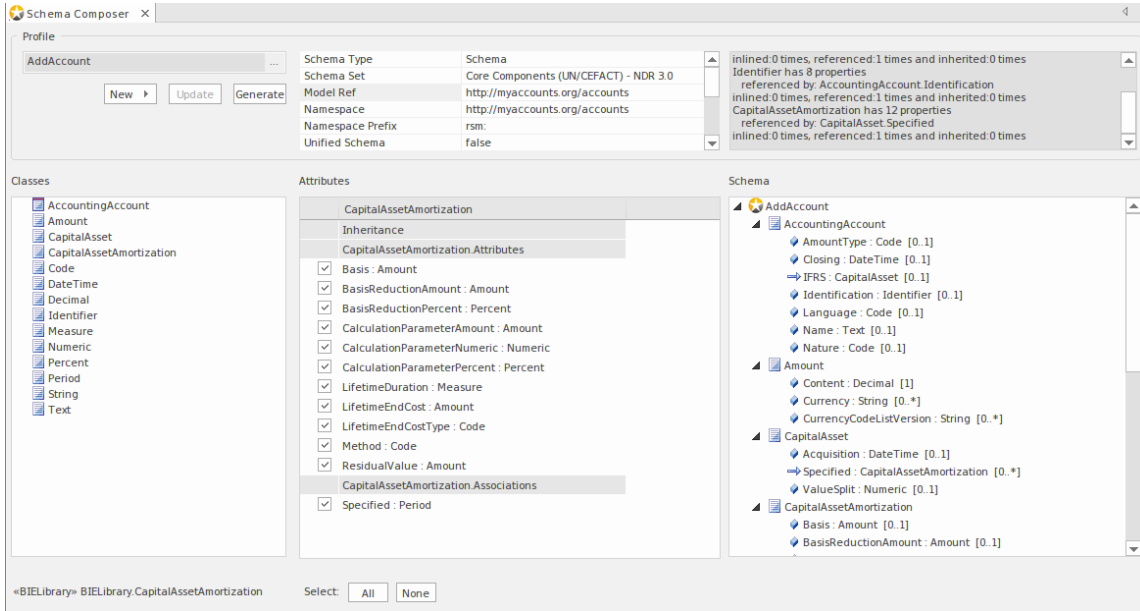
上下文菜单	右键单击元素 特定 NIEM 1 生成NIEM 1架构
-------	-------------------------------

生成NIEM Schemas (NIEM 3.0 及以上)

单击 ModelPackageDescription类的任何object实例并选择以下选项之一：

功能区	特定>技术> NIEM >生成NIEM架构
上下文菜单	右键单击元素 特定 NIEM 生成NIEM架构

UPCC架构指南



本指南描述了符合 UPCC 的 XML 模式的组成和生成。

创建UPCC架构

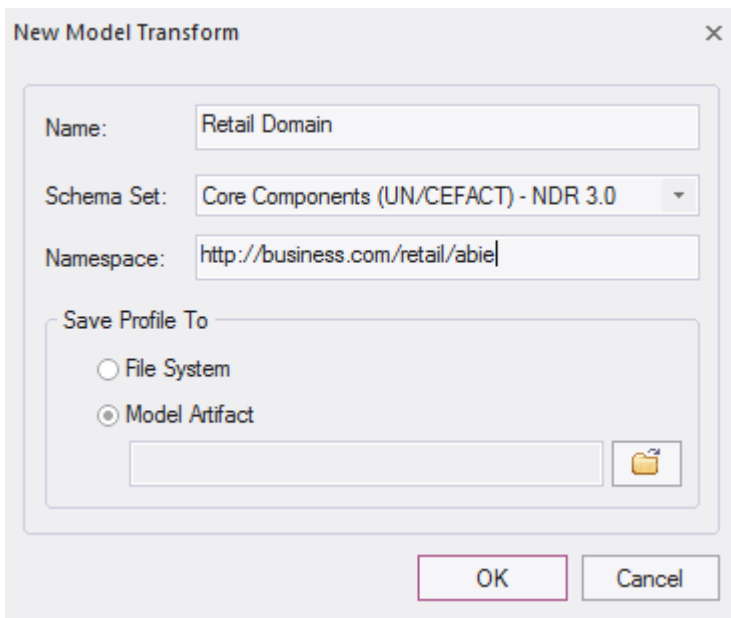
节	行动
1	显示Schema 编辑器。
2	单击 新建 架构”。
3	输入架构的唯一名称。
4	从标准列表中选择要使用的 UPCC 命名和设计规则。
5	将一个或多个 <ABIE> 组件从 <BIE library> 拖到类列表中。
6	使用上下文菜单将类设置为根元素。
7	选择所需的属性（将引用类型添加到架构中）。
8	单击更新按钮以保存更改。
9	单击生成按钮并选择“XML Schema”。点击确定按钮。

模型组成

Schema 编辑器的组成特征对于从核心模型创建子模型很有用。这可以像从核心包生成单个业务包（*UN/CEFACT* 核心组件标准中的 *CDT* 库到 *BDT* 库转换）或从大型核心模型创建完成子模型一样简单。

这种任务的艰巨性可能令人生畏并且容易出错。例如，确保子模型引用的每个类型都包含在子模型中。Schema 编辑器通过自动计算依赖关系并在必要时将它们添加到模式中来解决这个问题。

创建变换

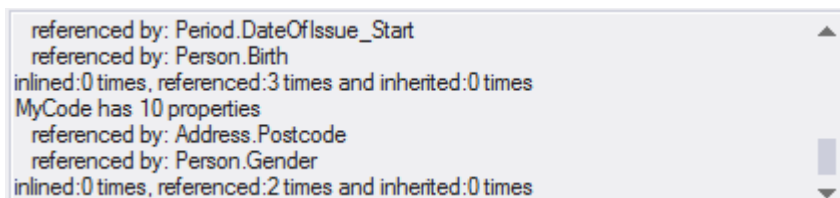


定义内容模型

通过将类从模型中拖放到 Schema 编辑器类窗口并选择要包含的属性，可以将内容添加到模型中。结果类型可以反映核心类型或提供更简单的分类。当包含一个属性时，Schema 编辑器将检查属性类型，如果缺少类型，将自动将其添加到模式中。

参考检查

当属性先前包含在架构中且不再引用的属性时，不会自动删除该属性类型。但是，如果您在类窗口中选择该类，Schema 编辑器将始终显示该类型的引用数。完全没有引用的类型可以很容易地删除。



概括

这里总结了组成子模型的过程：

1. 创建一个 Schemer Composer 转换配置文件
2. 通过将模型中的类拖放到模式中来创建元素。
3. 包括/排除所需的属性。
4. 生成子模型。

访问

功能区	开发>架构建模> Schema 编辑器>打开Schema 编辑器：新建>转换
-----	--

生成模型子集 (变换)

定义了子模型或库的内容并应用了任何限制后，您现在可以生成模型。可以执行的模型转换取决于与配置文件关联的技术。此处列出了每种技术及其支持的转换：

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

模型变换

选择要运行的模型。

转换选项	描述
NIEM	<p>NIEM模型子集</p> <p>此选项将生成一个NIEM模型子集，其中包含配置文件描述的模式。当您单击确定按钮时，系统将提示您选择目标包以创建子集。构成子集的<<Namespace>>包随后将在此位置创建。如果任何子集包已经存在于该位置，它们的内容将被添加到。所有的subset包都会有'defaultPurpose'标记值设置为'subset'。</p> <p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
通用的	<p>通用模型子集</p> <p>系统将提示您输入目标包。这将使用模式中的类型填充。如果输入限定符，这将应用于生成的类。架构中的任何限制也将被应用。目标包中存在的类型将被覆盖。将添加新属性。此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p> <p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
UN/CEFACT NDR 3.0	<p>BDT 图书馆</p> <p>业务Datatype库将从配置文件中列出A核心数据类型填充。构造型将根据CCTS 规范进行转换。这些类型可能比它们的核心对应物更受限制。存在的数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p> <p>国际展览局图书馆</p> <p>业务信息实体库将从聚合A核心组件中填充。配置文件中列出的构造型将根据CCTS 规范进行转换。存在的数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p>

	<p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
UN/CEFACT NDR 2. 1	<p>UDT 库</p> <p>将选定核心数据类型的非限定副本执行到 UDT 库。</p> <p>QDT 库</p> <p>合格A业务Datatype库将从配置文件中列出的核心数据类型填充。结果类型的名称将由配置文件中的命名限定符限定。构造型将根据属性规范进行转换。存在的数据类型将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>国际展览局图书馆</p> <p>业务信息实体库将从配置文件中列出A聚合核心组件填充。构造型将根据 CCTS 规范进行转换。现有数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>

生成

单击确定按钮以生成模式。当生成完成时，将显示消息导出配置文件 <名称>已完成。然后您可以在浏览器窗口中展开该包以查看生成的UML模型。

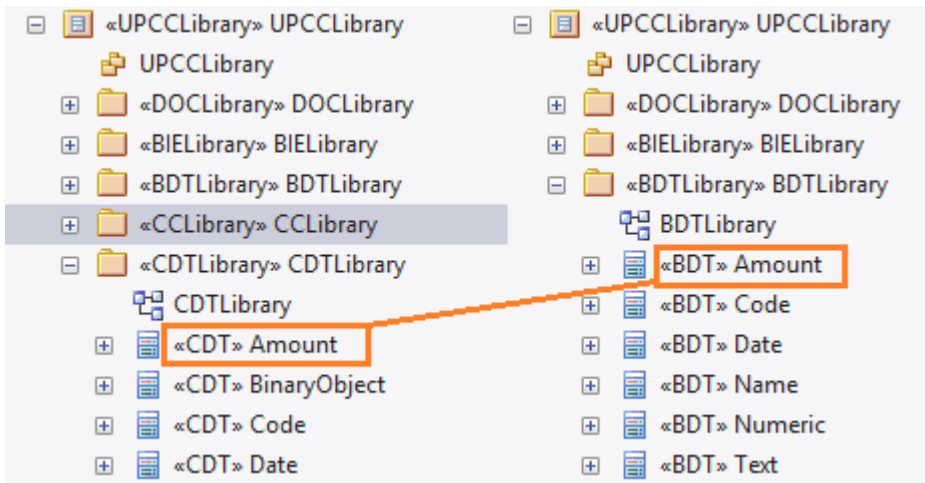
注记

- Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

核心组件 (UPCC) 的UML配置文件

UPCC 提供核心组件和核心数据类型库，Enterprise Architect用户可以通过模型向导测试框架（首页的“从模式创建”选项卡）使用。无论您是根据模型规范，还是想要利用此标准带来的优势，或有合规性要求，使用此技术的模型，您都需要 - 作为最低要求 - 模型业务数据类型库和业务信息实体库。Schema 编辑器可以为您生成这些库。

此图显示了从 UPCC 核心 CDT 库创建的 BDT 库



公共库

核心组件的UML配置文件的两个版本共享的库。

图书馆	描述
CC图书馆	CCTS 核心组件库。
CDT图书馆	CCTS 核心数据类型库。它包含基本的数据类型，如金额、代码、文本和图形。
BIE图书馆	包含基于 CCLibrary 组件的 ABIE 实体A业务库。可以使用Schema 编辑器来组合实体。这些也可以使用该技术可用的UML建模工具进行建模。
文档库	A包通常用于信息程序集的建模。您可以通过将信息组件加载到Schema 编辑器中来生成信息组件的架构组装

UPCC 库

核心组件的UML配置文件有两个版本，NDR 3.0 和 NDR 2.1。两个配置文件都描述了一组通用的库，但有一些区别，如下所述：

NDR 3.0

图书馆	描述
BDT图书馆	包含基于 CDTLibrary 类型的 BDT 类型A业务库。Schema 编辑器可用于从核心库中的选定类型轻松生成 BDTLibrary 的内容。

NDR 2.1

图书馆	描述
UDT库	不合格的数据类型库。基本上是用业务时间的上下文的镜像。Schema 编辑器可用于从核心库中的选定类型轻松生成 UDTLibrary 的内容。
QDT库	A合格的数据类型库。该库包含基于具有限定类型名称的 CDTLibrary 的受限类型。Schema 编辑器可以轻松地从核心库中的选定类型生成 QDTLibrary 的内容。

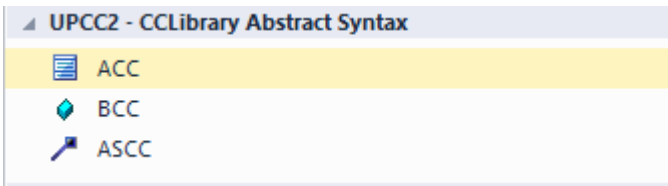

图表







核心组件的UML配置文件使用UML类图来组合组件。然而，该技术为其每个库提供了特定的工具箱页面。

UPCC工具箱Pages




公共

在此表示法中，UPCCx 表示 UPCC 配置文件，x 是 NDR 的版本







公共	描述
UPCCx - CCLibrary 抽象语法	
UPCCx - DOCLibrary 抽象语法	
UPCCx - CDTLibrary 抽象语法	

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

NDR 3.0

库语法	描述
UPCC - BDTLibrary 抽象语法	<ul style="list-style-type: none"> ▾ UPCC3 - BDTLibrary Abstract Syntax <ul style="list-style-type: none">  BDT  CON  SUP

NDR 2.1

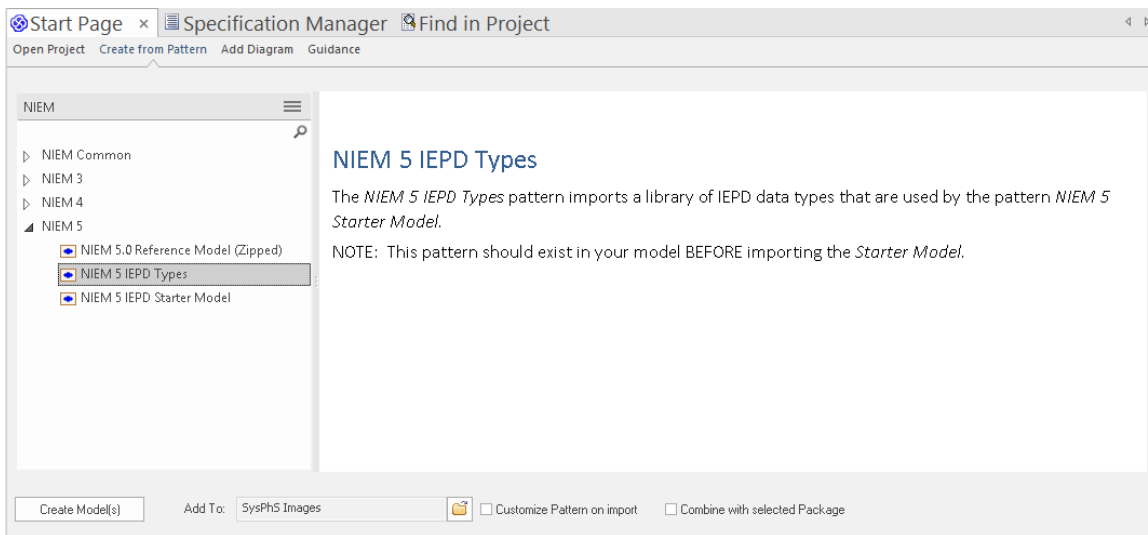
库语法	描述
UPCC - UDTLibrary 抽象语法	<ul style="list-style-type: none"> ▾ UPCC2 - UDTLibrary Abstract Syntax <ul style="list-style-type: none">  UDT  CON  SUP
UPCC - QDTLibrary 抽象语法	<ul style="list-style-type: none"> ▾ UPCC2 - QDTLibrary Abstract Syntax <ul style="list-style-type: none">  QDT  CON  SUP

可用的框架

使用Enterprise Architect模型向导（首页'Create from'）选项卡，您可以在几分钟内将任何受Schema编辑器支持的框架如NIEM、IM、模式部署到您的模型中，提供了一个多功能的UML介质用于在这些技术中建模。

这些框架也可以直接从Sparx Systems Reusable Asset Service 获得（通过云服务器的“连接”对话框，然后在模型向导上的“框架”）。

注记：除了CIM和NIEM等自定义框架外，还可以使用标准类模型来快速构建通用Schema，因此如果您不是针对特定的元模型，可能最简单的方法是在您的数据中建模UML并使用结果模型作为Schema编辑器的输入。



国家信息交换模型 (NIEM)

这是NIEM项目管理办公室 (PMO) 发布的[国家信息交换模型](#)。

Enterprise Architect为在NIEM中建模提供了以下资源：

- 提供的框架包括NIEM核心、NIEM域、代码列表和外部架构适配器：
 - NIEM 2.1 使用 NIEM-UML 1.0 建模
 - NIEM 3.0 使用 NIEM-UML 1 建模。1
 - NIEM 3.1 使用 NIEM-UML 1 建模。1
 - NIEM 4.0 使用 NIEM-UML 1 建模。1
- NIEM子集创建：
 - Schema 编辑器帮助您创建符合NIEM的命名空间的子集
- NIEM模式生成：
 - 从NIEM 2 中的模型包描述生成完成NIEM IEPD，NIEM 3 或NIEM 4 格式

公共信息模型(CIM)

这是国际电工委员会 (IEC) 技术委员会 57 发布的[CIM 规范](#)。

Enterprise Architect为在 CIM 中建模提供了以下资源：

- 架构组合
 - XML 模式 (XSD)
 - 资源描述符格式 (RDFS)
 - 资源描述符增强格式
 - JavaScript物件表示法 (JSON)

- 插件
- 一体化
- 脚本集成

通用业务语言 (UBL)

UBL 是由[OASIS](#)发布的 CCTS 实施，事实证明它在欧洲政府中很受欢迎，用于整合机构之间的信息交换。

Enterprise Architect为使用 UBL 组合业务文档提供了以下资源：

- UML框架
 - UBL 2.1主要文档库
 - UBL 2.1公共部件库
- 业务文档组合
 - 用于组件组合的Schema 编辑器
 - 用于文档合成的Schema 编辑器
 - 用于模式生成的Schema 编辑器
 - 插件
 - 一体化
 - 脚本集成

核心部件技术规范 (CCTS)

这是 UN/CEFACT 发布的[CCTS 规范](#)。

Enterprise Architect为在 CCTS 中建模提供了以下资源：

- UML框架：
 - 1 2.1 核心组件库
 - UPCC 3.0 核心组件库
 - UMM 2.0 业务需求、编排和信息视图。
- 业务部件库创建/管理
 - Schema 编辑器用于 ABIE 和 BDT 组合
 - 插件
 - 一体化
 - 脚本集成
- 业务部件架构组合
 - XSD、JSON 的Schema 编辑器
 - 插件
 - 一体化
 - 脚本集成

插件

框架 (自定义)

除了这些方法之外，Schema 编辑器还与Enterprise Architect自动化接口集成，以支持任何个人或团体实施他们自己的方法。一个插件

向Enterprise Architect注册提供架构生成功能的兴趣的公司将有机会在Schema 编辑器工具中提供其任何产品。

脚本Framework (自定义)

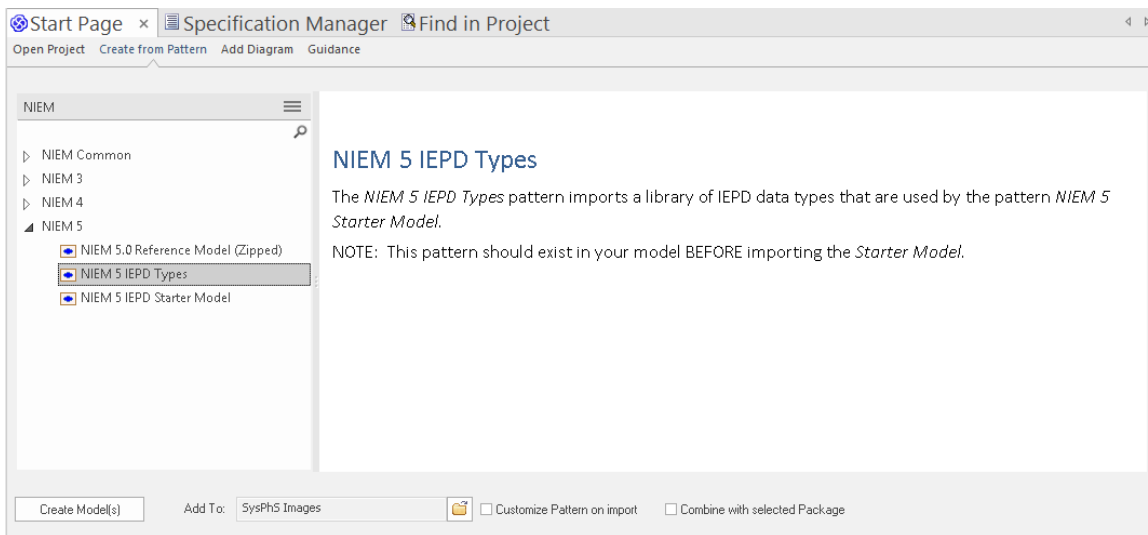
Schema 编辑器还为使用它创建的任何配置文件提供对模式生成的无条件控制。通过编写自己的脚本，作者可以访问任何模式的定义，并最终以他们选择的格式生成他们想要的任何文档。

安装核心框架

Enterprise Architect提供了丰富多样的建模技术，包括Schema 编辑器中列出的每个标准。这些框架可用作UML模型和/或MDG 技术，使用Enterprise Architect的模型向导选项卡（首页 从模式创建”选项卡）。模型本身也可以从Enterprise Architect的 Reusable Asset Service 直接访问。

注记：如果您正在对通用解决方案进行建模，而不是直接使用 CIM 或 UBL 等核心框架，则无需安装核心框架/模型。在这种情况下，您最好使用带有属性的简单UML类来创建数据模型。

模型向导



访问

功能区	设计>包>模型向导
上下文菜单	右键单击包 使用向导添加模型
键盘快捷键	Ctrl+Shift+M
其它	浏览器window '汉堡包模式' icon > 新模型来自

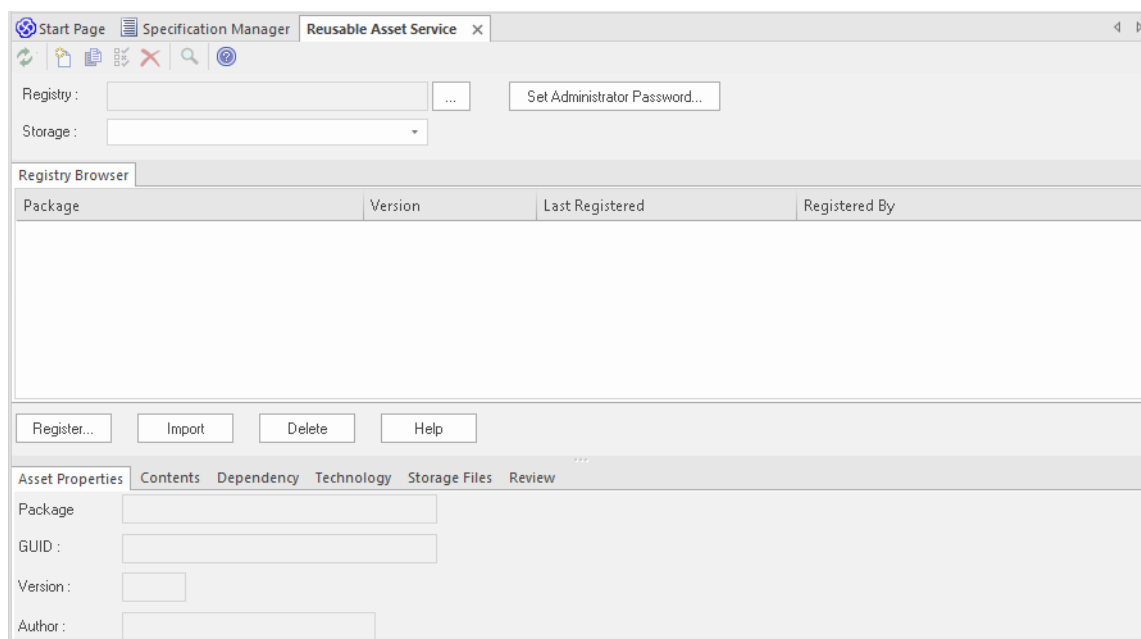
注记

您可以通过选择功能区选项来限制要使用的MDG 技术：特定>技术> 管理技术”。在这里您可以看到当前启用了哪些技术。

导入模型

节	行动
1	显示首页'从模式创建'选项卡 (模型向导) 。
2	单击 <perspective name> 按钮并选择所需的技术 。
3	突出技术 。
4	选择要导入的技术标准 。
5	单击模型(s) 按钮将框架导入您的模型 。

可重用资产服务



访问

功能区	发布 >模型交换 > Reusable Assets
-----	----------------------------

导入模型

节	行动
1	连接到可重用资产服务 。

2	从可用的存储库列表中选择
3	选择UML模型包
4	单击确定将选定的包导入您的模型。

架构Importer

您可以使用架构导入器将与Schema 编辑器兼容的架构导入Enterprise Architect。架构导入器对架构进行验证，验证成功后创建一个架构类型的Schema 编辑器配置文件，可以直接在Schema 编辑器中查看。

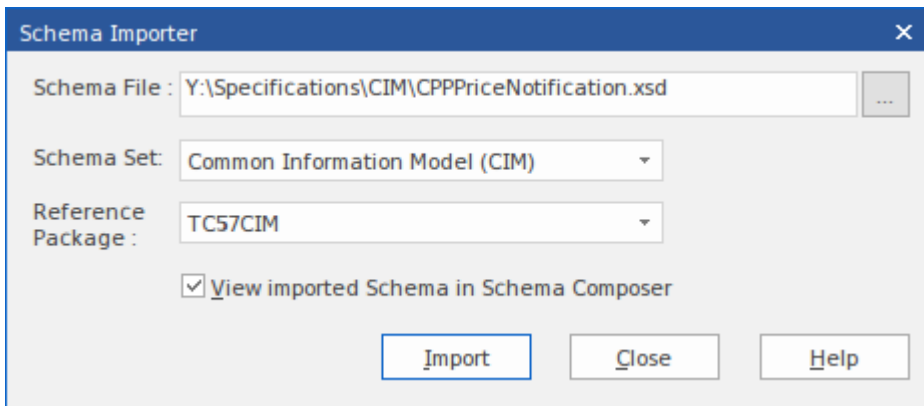
目前，您可以使用架构导入器来导入这些架构：

- 公共信息模型 (CIM) 特定的XML Schema
- 公共信息模型(CIM) 特定 RDFS XML

访问

功能区	开发>架构建模> Schema 编辑器>导入Schema 编辑器
-----	----------------------------------

使用架构导入器导入架构



选项	行动
架构文件	类型从中导入架构文件的目录路径和文件名。
架构集	选择要导入的架构类型。 目前架构Importer 支持导入特定的CIM： <ul style="list-style-type: none"> • XML Schema和 • RDFS XML
参考包	选择包含特定于模式集的公共元素的包。 架构导入器将根据参考包中的元素验证正在导入的架构中的元素。
在Schema 编辑器中视图导入的架构	选择此选项可在Schema 编辑器中打开导入的配置文件。
导入	单击此按钮开始导入过程。
关	单击此按钮可关闭 架构导入器”对话框。

注记

- 系统输出窗口会显示导入进度
- Schema 编辑器会在导入架构之前根据参考包中的元素验证架构；如果验证失败，系统输出窗口将显示验证失败的架构元素，并停止导入过程
- 双击系统输出窗口中的验证错误条目以在Enterprise Architect的内部文件编辑器中打开架构并源错误的来源
- 如果验证成功，弹出“架构定义”对话框，您可以将导入的配置文件保存在文件系统中，也可以作为工件文件保存在当前模型中

Schema 编辑器自动化集成

Schema 编辑器可以从Enterprise Architect Automation接口访问。客户端A脚本或插件

) 可以使用 存储库“ object的 属性”属性获得对接口的访问。当Schema 编辑器加载了配置文件时，此接口可用。

Schema 编辑器Addin集成

Enterprise Architect插件可以与Schema 编辑器集成，通过提供替代方案为用户提供模式和子模型的生成。

Schema 编辑器脚本集成

尽管Schema 编辑器提供了基于各种流行技术的开箱即用的模式组合，但它的脚本集成为您提供了一些灵活性，让您可以实现自己的需求。您可以通过三种方式在Schema 编辑器中利用脚本：

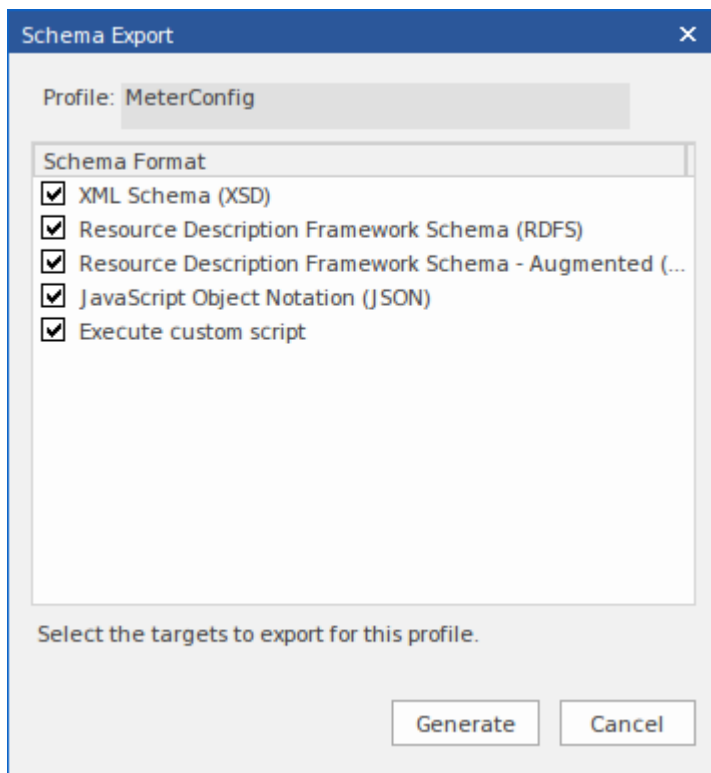
- 使用脚本语言提供自定义模式生成
- 使用脚本语言提供自定义模型转换
- 提供自定义原型映射到任何标准模型转换（例如 UPCC）

脚本模型变换

虽然Schema 编辑器为各种框架提供内置转换，但您始终可以编写自己的转换，使用 Composer 的组合工具设计模式，然后使用手工制作的脚本执行自定义转换。

架构脚本生成

当您在Schema 编辑器中选择一条消息并单击生成时，您会看到多种导出格式。这些选择之一是“执行自定义脚本”



架构迭代脚本示例

这个例子演示了在一个用JavaScript编写的Enterprise Architect脚本中访问Schema 编辑器。该脚本首先获取到Schema 编辑器的接口，然后遍历模式，打印出类型及其每个属性。

```
/*
*示例名称:脚本Schema 编辑器脚本
*作者： Sparx Systems
* 目的：演示使用自动化和JavaScript访问Schema 编辑器
* 语言： JavaScript
* 日期：2020
*/
函数 打印类型( xml 类型 , xmlns , 乌里 )
{
    变量 xmlProp 作为 艺电 . 架构属性 ;
    变量 xmlPropEnum 作为 艺电 . SchemaPropEnum ;
    变量 xmlChoiceEnum1 作为 艺电 . 模式类型枚举 ;
    变量 xmlChoiceEnum2 作为 艺电 . 模式类型枚举 ;

    会话.输出( "类型: " + xml 类型 . 类型名称 + " 在命名空间中 : " + xmlns + " : " + uri );
    xmlPropEnum = xmlType . 属性 ;
    如果( xmlPropEnum )
    {
        xmlProp = xmlPropEnum . 获取第一 ( ) ;
        而( xmlProp )
        {
            if ( xmlType . IsEnumeration () )
            {
                会话.输出( " " + xmlProp . 名称 ) ;
            }
            else
            {
                变量 sPropDesc = xmlProp . 名称 ;
                sPropDesc += " : " ;
                if ( xmlProp . IsPrimitive () )
                    sPropDesc += xml 属性 . 原始类型 ;
                else
                    sPropDesc += xml 属性 . 类型名称 ;

                if ( xmlProp . IsByReference () )
                {
                    sPropDesc += " ( 通过引用 ) " ;
                }
                if ( xmlProp . IsInline () )
                {
                    sPropDesc += " ( 内联 ) " ;
                }
            }
        }
    }
}
```

```
        会话.输出( " " + sPropDesc + " · 基数 : " + xml 属性 · 基数 ) ;

        xmlChoiceEnum1 = xml 属性 · 选择 ;
        xmlChoiceEnum2 = xml 属性 · 模式选择 ;
        变量 基数 = xmlChoiceEnum1 · 获取计数() + xmlChoiceEnum2 · 获取计数 (
    ) ;

    如果(计数>1)
    {
        会话.输出( " 选择 : " ) ;
        xml选择 = xmlChoiceEnum1 · 获取第一 ( ) ;
        而( xmlChoice )
        {
            会话.输出( " " + xml 选择 · 类型名称 ) ;
            xml选择 = xmlChoiceEnum1 · 获取下一个 ( ) ;
        }
        xml选择 = xmlChoiceEnum2 · 获取第一 ( ) ;
        而( xmlChoice )
        {
            会话.输出( " " + xml 选择 · 类型名称 ) ;
            xml选择 = xmlChoiceEnum2 · 获取下一个 ( ) ;
        }
    }
}

xmlProp = xmlPropEnum · 获取下一个 ( ) ;
}
}
}

函数 主要 ( )
{
    变量 图式 作为 艺电 · 架构作曲家 ;
    变量 xml类型 作为 艺电 · 模式类型 ;
    变量 xmlTypeEnum 作为 艺电 · 模式类型枚举 ;
    变量 xml命名空间枚举 作为 艺电 · 模式名称空间枚举 ;
    变量 xmlNS 作为 艺电 · 模式命名空间 ;

    // 获取 SchemaComposer
    图式 = 存储库 · 架构作曲家 ;

    // 打印命名空间引用
    xml命名空间枚举 = 架构 · 命名空间 ;
    如果( xmlNamespaceEnum )
```

```

    {
        xmlNS = xmlNamespaceEnum . 获取第一 ( ) ;
        而( xmlNS )
        {
            会话.输出( "xmlns : " + xmlNS .名称 + "URI=" + xmlNS . URI );
            xmlNS = xmlNamespaceEnum . 获取下一个 ( ) ;
        }
    }
    // 获取架构类型枚举器
    xmlTypeEnum = 架构 . 模式类型 ;
    xml类型 = xmlTypeEnum . 获取第一 ( ) ;
    而( xmlType )
    {
        变量 xmlns = 架构 . GetNamespacePrefixForType ( xml 类型 . TypeID );
        乌里 = 架构 . GetNamespaceForPrefix ( xmlns );
        打印类型( xmlType , xmlns , uri );
        xml类型 = xmlTypeEnum . 获取下一个 ( ) ;
    }
}

```

主要 () ;

智能感知帮助脚本

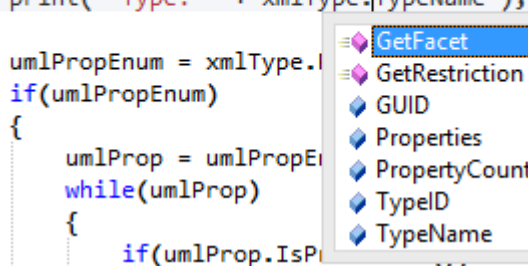
编辑Enterprise Architect脚本帮助您使用Schema 编辑器属性编写脚本。方法是在智能感知脚本中提供编辑脚本或其自动化接口的交互和方法。

```

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

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

```



模型变换中的构造型映射

构造型构成了MDG 技术方法的很大一部分。MDG 技术的单独UML配置文件定义了构造型以为其元素提供有用的分类。从核心框架转到业务模型或子域时，重新分配构造型是一个常见的要求。当您使用 CCTS 框架时，您

生成的业务组件的原型由Enterprise Architect根据 CCTS 规范定义的映射（例如，ACC 到 ABIE）自动生成。

当您在Schema 编辑器中打开或创建模型转换配置文件时，您可以指定一个脚本来为您执行此映射。可以从属性窗口中选择脚本。

Schema Type	Transform
Schema Set	Core Components (UN/CEFACT) - NDR 3.0
Transform Rule Script	Schema Composer.BDTTransformRule ...
Qualifier	
BDTLibrary	BDTLibrary
BIELibrary	BIELibrary

该脚本可以用JavaScript、JScript 或 VBScript 编写，并且只需要实现这个函数（这里用JavaScript表示法描述）：

函数TranslateStereotype(srcStereo)

```
{  
var destStereo = srcStereo  
if (srcStereo == "BDT")  
{  
destStereo = "My_BDT"  
}  
返回目标立体声；  
}
```

MDG 技术- UML配置文件Extensions

Schema 编辑器使用MDG技术。它用于模式生成的标准（除 *Generic* 外）仅对遵循该框架的模型有意义。然而，扩展现有的MDG 技术非常容易。确保在您的业务特定域或子域中创作的元素提供一致命名的元数据或 标记值”。

Schema 编辑器通过其脚本集成支持对UML配置文件/框架的扩展。当在Schema 编辑器中分配一个脚本时，转换过程将调用该脚本并要求它翻译关键字。这些关键字通常是UML原型。如果特定技术与配置文件相关联，Schema 编辑器将调用此函数，将MDG 技术的名称传递给它。

该脚本可以返回输入名称，并且不会发生映射，或者它可以返回另一个MDG 技术的名称。发生这种情况时，Schema 编辑器将再次要求该函数选择性地映射任何UML配置文件。最后，它会要求脚本翻译来自核心技术的刻板印象。

模型转换的结果将是子模型的任何UML元素将显示除了任何核心标记值之外的扩展标记值。

映射MDG 技术的示例脚本

函数TranslateStereotype (stereo)

```
{
var newStereo = 立体声 ;
如果 ( 立体声=="UPCC3" )
{
newStereo = "XXX UPCC3"
}
返回新立体声 ;
}
```

映射UML配置文件的示例脚本

函数TranslateStereotype (stereo)

```
{
var newStereo = 立体声 ;
if (stereo == "UPCC3 - BIE 库抽象语法")
{
newStereo = "UPCC3 - BIE 库 XXX 语法"
}
返回新立体声 ;
}
```

映射UML构造型的示例脚本

函数TranslateStereotype (stereo)

```
{
var newStereo = 立体声 ;
如果 ( 立体声=="ABIE" )
```



```
{  
newStereo = "XXX ABIE";  
}  
返回新立体声 ;  
}
```

XSD 模型

XML Schema Definition (XSD)，也称为XML Schema，是一种 World Wide网络Consortium (W3C) XML 技术，用于指定 XML 文档必须遵守的规则。XSD 支持对于开发完整的完成服务的架构(SOA) 至关重要，UML 2.5 和 XML 的耦合提供了在组织内指定、构建和部署基于 XML 的 SOA 工件的自然机制。

XSD 的UML配置文件指定了一组可以应用于UML模型的构造型、标记值和约束，以更改结果模式的特定方面。Enterprise Architect通过工具箱的XML Schema页面为图表配置文件提供原生支持。Enterprise Architect支持的XSD配置文件是对出版物建模Applications with UML中定义的配置文件的改编。

通过Enterprise Architect使用 XSD配置文件，您可以快速模型、正向工程和反向工程XML Schema。

您还可以使用Enterprise Architect Schema 编辑器快速定义和生成 XSD 和其他模式。

建模XSD

您可以使用以下UML类图在两个级别上模型XML 模式：

- 没有特定于 XML 模式的实现细节，由Enterprise Architect的架构生成器直接生成；生成器应用一组默认映射将抽象模型包转换为 W3C XML Schema (XSD) 文件
- 使用图表工具箱的“XML Schema”页面，使用 XML 模式特定的定义进行图表，该工具箱提供 XSD 的 UML配置文件的结构

对XML Schema进行模型

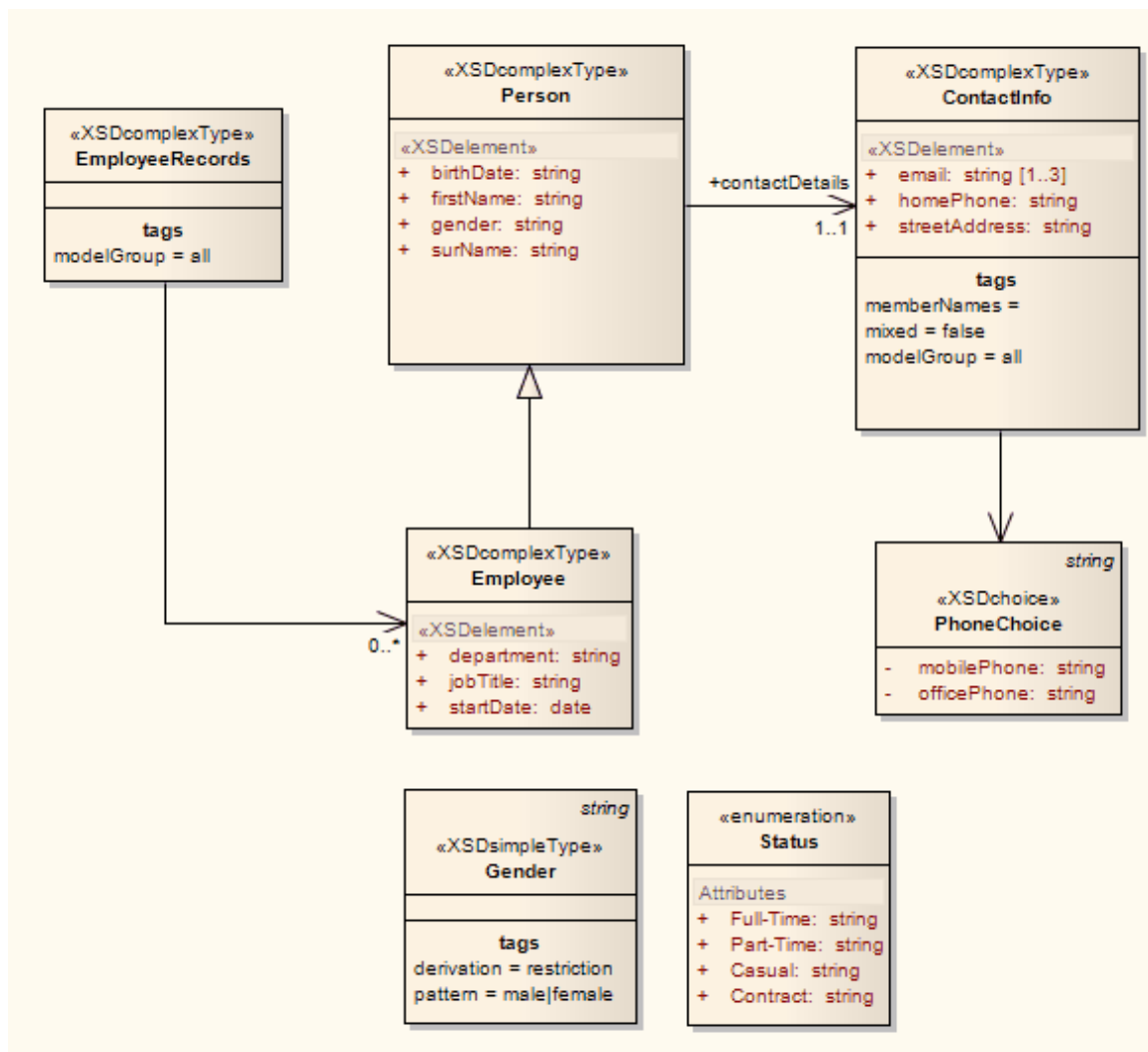
节	行动
1	在浏览器窗口中，创建您需要的顶层项目结构（模型和视图），然后单击相应的视图。
2	单击浏览器窗口标题下拉菜单中的“新包”选项。将显示“新模型包”对话框。
3	在“名称”字段中输入新包的名称，然后选择“创建图表”单选按钮。 单击确定按钮。将显示“新图表”对话框。
4	在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。
5	单击确定按钮。在浏览器窗口中，双击新图表名称旁边的图标；图表在“图表视图”中打开，类“页面显示在“图表工具箱”中。 此时，您可以： <ul style="list-style-type: none"> • 使用类工具箱图标创建类图，或 • 使用工具箱的“XML Schema”页面创建定制的图表XML Schema图（继续第6步）
6	单击  以显示“查找工具箱项”对话框并指定“XML Schema”以显示“XML Schema”工具箱页面。
7	单击工具箱中的“架构”图标并将其拖到类图中。 将显示“XSD 模式属性”对话框。 完全此对话框，然后单击确定按钮。 将显示“新图表”对话框。
8	同样，在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。 单击确定按钮。
9	在浏览器窗口和图表上创建了一个XSDschema原型包，带有一个子类图表。 双击图表上的包打开子类图表，并使用XML Schema工具箱中的构造来模型XML Schema。

注记

- 类的UML属性直接映射到 XML 元素或属性
- 如果您已将 XSD架构建模为直类图，则可以使用Schema 编辑器定义和生成架构
- XML Schema模型中的类没有方法，因为类方法和 XSD 构造之间没有有意义的对应关系
- 建模限制 - 这些XML Schema构造不能在Enterprise Architect中建模：
 - 应用信息
 - 场地
 - 钥匙
 - keyref
 - 符号
 - 重新定义
 - 选择器
 - 替代组
 - 独特的

XSD图表

此示例图显示了一个类图，其中包含使用工具箱的“XSD架构箱”页面创建的图表特定元素。该图模拟了一个员工记录系统。



架构包


«XSDschema» 原型包充当 XSD 构造的容器，可以从中生成XML Schema。包中的所有类都在一个模式中定义；架构元素提供默认的架构范围设置。您可以通过从XML Schema工具箱中拖动架构图标并将其直接放到图表上来创建一个«工具箱»包。

工具箱图标




访问

要显示选定 «XSDschema» 原型包的 XSD 模式 属性”对话框，请使用此处概述的方法之一：

功能区	设计>包>管理>属性
上下文菜单	右键单击 «XSDschema» 定型包 属性
其它	在浏览器窗口中，双击现有的 «XSDschema» 原型包，或将  Schema 图标从工具箱拖到图表上（这将创建一个新的 «XSDschema» 原型包）

定义属性

字段/按钮	行动
架构名称	如果您不想使用模式包的默认名称，请用另一个名称改写它。
目标命名空间	（可选）此架构包的目标命名空间中的类型。
前缀	（可选）类型在缩写文本中代表目标命名空间。
默认命名空间	（可选）所有无前缀 XSD 元素和 XSD 属性的默认命名空间中的类型。
架构文件	类型在或浏览（单击  ）要生成此包的XML Schema文件的文件路径。
XMLNS	识别此架构包中使用的其他命名空间或命名空间前缀对。 要添加命名空间或命名空间前缀对，请单击 新建”按钮；要编辑现有条目，请双击它。在任何一种情况下，都会显示 命名空间详细信息”对话框。 <ul style="list-style-type: none"> 前缀类型在缩略文本中代表命名空间 命名空间-命名命名空间名称中的类型 确定- 单击此按钮保存新信息并关闭 命名空间详细信息”对话框 取消 - 单击此按钮可放弃新信息并关闭 命名空间详细信息”对话框

	<ul style="list-style-type: none">帮助- 单击此按钮显示此帮助主题 要从列表中删除条目，请单击它并单击删除按钮。
确定	单击此按钮保存输入的模式数据并关闭 XSD 模式 属性”对话框。
取消	单击此按钮可放弃输入的模式数据并关闭 XSD 模式 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有架构包信息时会显示此按钮。 单击 按钮打开架构元素的UML元素 属性”对话框。

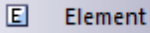
注记

- 默认的模式范围设置由标记值定义，您可以在模式元素审阅的 标签”属性选项卡或元素的属性窗口中查看查看；如果需要，您可以编辑模式范围的设置，或者在各个 XSD 构造元素的属性和标记值中提供元素特定的覆盖

全球元素

«XSDtopLevelElement» 原型类充当 XSD 全局元素。您可以通过从XML Schema工具箱中拖动“元素”图标并将其直接放到图表上来创建它。

工具箱图标



访问

要为选定的 «XSDtopLevelElement» 原型类显示“XSD元素属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDtopLevelElement» 原型类 属性
键盘快捷键	Alt+Enter
其它	在浏览器窗口中，双击 «XSDtopLevelElement» 原型类，或将  图标从工具箱拖到图表上（这将创建一个新的 «XSDtopLevelElement» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用全局元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSD 分类器元素，或 • 选择这两个复选框之一
嵌套的 complexType	选中此复选框以创建一个 XSDcomplexType 作为此全局元素的子元素。
嵌套的 simpleType	选中此复选框可创建一个 XSDsimpleType 作为此全局元素的子元素。
价值	(可选) 如果您在“类型”字段中输入了 XSD 内置数据类型，请输入一个值。
默认	选择此单选按钮可将值设置为默认值。

固定的	选择此单选按钮可将值设置为固定值。
注解	(可选) 类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的元素数据并关闭 XSD元素的 属性”对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD元素的 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD元素信息时，会显示此按钮。 单击按钮打开全局元素的UML元素 属性”对话框。

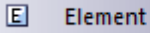
注记

- '类型'、'Nested complexType' 和 'Nested simpleType' 字段是互斥的；选择一个禁用其他
- 只有在创建新的全局元素时（而不是在编辑全局元素时），对话框中的 嵌套复杂类型”和 嵌套简单类型”字段才可用
- A全局元素：
 - 不能包含任何UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型类或组元素类的关联连接器的目标
 - 不能成为概括器的目标
 - 可以是复杂类型类或简单类型类的一个概括器的源

当地元素

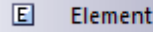
本地元素是A «XSDelement» 原型UML属性，它充当本地 XSD元素。您可以通过将 无素”图标从XML Schema工具箱中拖放到 «XSDcomplexType» 或 «XSDgroup» 原型类上来创建它。

工具箱图标




访问

要为选定的 «XSDelement» 原型UML属性显示 “XSD元素属性”对话框，请使用此处概述的方法之一。

功能区	在图表中选择特定的 «XSDelement» 原型属性： 设计>元素>特征>属性
上下文菜单	在图表中选择特定的 «XSDelement» 原型属性： 右键单击属性 视图属性
键盘快捷键	在图表中选择特定的 «XSDelement» 原型属性： F9
其它	双击 «XSDelement» 原型属性，或 将  图标拖到 «XSDcomplexType» 或 «XSDgroup» 原型类上，（这会在类中创建一个新的 «类»）

定义属性

字段/按钮	行动
名称	如果您不想使用本地元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSDcomplexType 或 XSDsimpleType元素作为分类器
参考	（可选）在生成的 XSD 中指定是否使用 ref 属性（而不是 type 属性）来引用您在 “类型”字段中选择的 XSDcomplexType 或 XSDsimpleType元素。
价值	（可选）如果您在 “类型”字段中输入了 XSD 内置数据类型，请输入一个值。

默认	选择此单选按钮可将值设置为默认值。
固定的	选择此单选按钮可将值设置为固定值。
MinOccurs	(可选) 类型该元素必须在类中出现的最小次数。 类型'0' 表示该元素是可选的。 默认值为 " 1 " 。
最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1 。
形式	(可选) 单击下拉箭头并选择是否限定元素 : <ul style="list-style-type: none"> • 合格 -使用架构包中定义的前缀来限定此元素 • unqualified - 不限定此元素
注解	(可选) 在您需要的任何注记类型输入此本地元素。
确定	单击此按钮保存输入的元素数据并关闭 XSD元素的 属性"对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD元素的 属性"对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD元素信息时，会显示此按钮。 点击按钮打开本地元素的属性属性。

注记


- 只有 «Complex Type»、«Group» 和 «Model Group» 原型元素可以具有此UML属性

全局属性

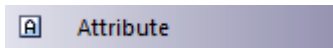
全局属性是A «XSDtopLevelAttribute» 原型类。您可以通过从XML Schema工具箱中拖动 属性”图标并将其直接放到图表上来创建它。

访问


要为选定的 «属性» 原型元素显示 XSD 属性属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDtopLevelAttribute» 原型元素 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDtopLevelAttribute» 原型类，或 从工具箱中拖动  Attribute 图标并直接拖放到图表上（这将创建一个新的 «XSDtopLevelAttribute» 原型类）

工具箱图标



定义属性

字段/按钮	行动
名称	如果您不想使用全局属性的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSDsimpleType元素作为分类器或者，选中 嵌套的 simpleType”复选框。
嵌套的 simpleType	选中此复选框可创建一个 XSDsimpleType元素作为此全局属性元素的子元素。
价值	(可选) 如果您在 类型”字段中选择了 XSD 内置数据类型，请输入一个值。
默认	选择此单选按钮可将 值”字段设置为默认值。
固定的	选择此单选按钮可将 值”字段设置为固定值。

形式	(可选) 单击下拉箭头并选择 : <ul style="list-style-type: none"> • 有资格使用提供的架构前缀包限定此属性 • unqualified 在属性名称上不显示限定前缀
注解	(可选) 类型在任何注记中您需要此属性。
确定	单击此按钮保存输入的属性数据并关闭 XSD 属性 属性”对话框。
取消	单击此按钮可放弃输入的属性数据并关闭 XSD 属性 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD元素信息时，会显示此按钮。 单击按钮打开全局属性类的UML元素 属性”对话框。

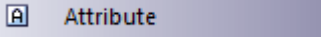
注记

- 只有在创建新的全局属性时 (而不是在编辑全局属性时) ，对话框中的 嵌套 simpleType”字段才可用
- 类型”和 嵌套的简单类型”字段是互斥的；选择一个禁用另一个
- 全局属性A
 - 不能包含任何UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型类的关联连接器的目标
 - 不能成为概括器的目标
 - 可以是一个概括类型类的连接器的源

本地属性

本地属性是A «XSDattribute» 原型UML属性。您可以通过从XML Schema工具箱中拖动 属性”图标并将其放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类上来创建它。

工具箱图标




访问

要为选定的 «属性» 原型UML属性显示 “XSD 属性属性”对话框，请使用此处概述的方法之一。

功能区	在图表上选择特定的 «XSDattribute» 原型UML属性： 设计>元素>特征>属性
上下文菜单	在图表上选择特定的 «XSDattribute» 原型UML属性： 右键单击属性 视图属性
键盘快捷键	在图表上选择特定的 «XSDattribute» 原型UML属性： F9
其它	双击 «XSDattribute» 原型UML属性，或 将  图标从工具箱拖放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类（这将创建一个新的 «XSDattribute» 原型UML属性）

定义属性

字段/按钮	行动
名称	如果您不想使用本地属性的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSDsimpleType元素作为分类器
参考	（可选）指定是否在生成的 XSD 中使用 ref 属性（而不是 type 属性）来引用您在 “元素类型”字段中选择的 XSDsimpleType。
价值	（可选）如果您在 “类型”字段中输入了 XSD 内置数据类型，请输入一个值。

默认	选择此单选按钮可将值设置为默认值。
固定的	选择此单选按钮可将值设置为固定值。
形式	(可选) 单击下拉箭头并选择是否限定属性： <ul style="list-style-type: none">• 限定 -使用架构前缀中包的限定此属性• unqualified - 不限定此属性
注解	(可选) 在您需要的任何注记类型输入此本地属性。
确定	单击此按钮保存输入的属性数据并关闭 XSD 属性 属性”对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD 属性 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有 XSD 属性信息时会显示此按钮。 点击按钮打开本地属性的属性属性。

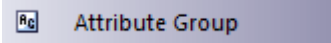
注记

- 只有复杂类型和属性组可以有这个UML属性

属性组

属性组类用于对一组 «XSDattribute» 原型UML属性和可以从 «类型» 原型类引用的 SimpleTypeClasses 进行分组。您可以通过从XML Schema工具箱中拖动 属性组”图标并将其直接放到图表上来创建它。

工具箱图标



访问

要为选定的 «属性» 原型类显示 XSD 属性组属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«XSDattributeGroup»原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDattributeGroup» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDattributeGroup» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用属性组的默认名称，请用另一个名称改写它。
注解	（可选）此属性组所需的任何注记中的类型。
确定	单击此按钮保存输入的属性组数据并关闭 XSD 属性组的 属性”对话框。
取消	单击此按钮可丢弃输入的属性组数据并关闭 XSD 属性组的 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有 XSD 属性组信息时会显示此按钮。 单击按钮打开属性组的UML元素 属性”对话框。

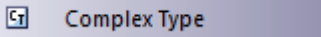
注记

- 一个属性组元素：
 - 不能是任何其他 XSD类的孩子
 - 只能包含 «类型» 原型UML属性和 SimpleTypeClasses
 - 可以是另一个属性组的关联连接器的源
 - 可以是复杂类型类的关联连接器的目标
 - 不能成为概括或连接的源或目标

复杂类型

«XSDcomplexType» 构造型应用于通用UML类，以定制架构中 complexType 定义的生成。您可以通过将复杂类型图标从XML Schema工具箱中拖放到图表上来创建一个 «类» 原型。

工具箱图标



访问

要为选定的 «XSDcomplexType» 原型类显示 “XSD complexType属性” 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDcomplexType» 原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDcomplexType» 原型类，或 将  Complex Type 图标从工具箱拖到图表上（这将创建一个新的 «XSDcomplexType» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 complexType类的默认名称，请使用其他名称改写它。
集团模型	单击向下箭头并选择定义此 complexType 的子元素应如何出现在架构中的选项。 <ul style="list-style-type: none"> • '序列' - 子元素必须以指定的顺序出现 • 'choice' - 只能出现一个子元素 • 'all' - 子元素可以以任何顺序出现
MinOccurs	(可选) 类型该元素必须在类中出现的最小次数。 类型0 表示该元素是可选的。 默认值为1。
最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1。

注解	(可选) 类型任何注记你需要的这个元素。
抽象的	(可选) 选中此复选框以在实例 XML 文件中使用此 complexType。
混合	(可选) 选中此复选框以允许字符数据显示在子元素中。
确定	单击此按钮保存输入的 complexType 数据并关闭 XSD complexType 的 属性“对话框”。
取消	单击此按钮可丢弃输入的 complexType 数据并关闭 XSD complexType 的 属性“对话框”。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD complexType 信息时，会显示此按钮。 单击按钮打开 complexType类的UML元素 属性“对话框”。

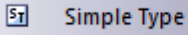
注记

- A complexType 可以：
 - 包含 XSDelement 和 XSDattribute 原型UML属性
 - 包含其他 complexTypes 作为子元素
 - 成为全球元素的孩子
 - 成为其他复杂类型、简单类型、属性组、组和模型组的关联连接器源
 - 最多有一个概括源或另一个复杂类型或简单类型类

简单类型

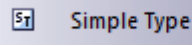
«XSDsimpleType» 构造型应用于通用UML类，以定制架构中 simpleType 定义的生成。您可以通过从XML Schema工具箱中拖动 SimpleType 图标并将其直接放到图表上来创建一个 «类型»类。

工具箱图标




访问

要为选定的 «XSDsimpleType» 原型类显示 “XSD simpleType属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDsimpleType» 原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDsimpleType» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDsimpleType» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 simpleType元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的«XSDsimpleType»元素作为分类器
限制	选择此单选按钮可将此 simpleType 的值限制为所选类型的值。 类的各种限制（方面）可作为此类的标记值使用。
列表	选择此单选按钮可将此 simpleType 指定为所选类型的值列表。
注解	（可选）类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的 simpleType 数据并关闭 XSD simpleType 的 “属性”对话框。

取消	单击此按钮可丢弃输入的 simpleType 数据并关闭 XSD simpleType 的 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD simpleType 信息时，会显示此按钮。 单击按钮打开simpleType类的UML元素 属性”对话框。

注记

- 简单类型A
- 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
- 不能包含任何子类
- 不能是源连接器的关联
- 可以成为一个概括器的目标
- 最多可以有一个概括器到另一个简单类型类

团体

Group类用于对一组 «类型» 原型UML属性、ComplexTypeClasses 和 SimpleTypeClasses 进行分组，这些属性可以从 «类型»类中引用。您可以通过从XML Schema工具箱中拖动 Group 图标并将其直接放到图表上来创建这种类型的元素。

工具箱图标



访问

要为选定的 «XSDgroup» 原型类显示 “XSD 组属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«XSDgroup»原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDgroup» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDgroup» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用组元素的默认名称，请用其他名称改写它。
集团模型	单击下拉箭头并选择定义该组的子元素应如何出现在复杂类型类中的值： <ul style="list-style-type: none"> • 序列指定子元素必须以指定的顺序出现 • 选择 - 指定只能出现一个子元素 • all - 指定子元素可以以任何顺序出现
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的组数据并关闭 XSD 组的 “属性”对话框。
取消	单击此按钮可丢弃输入的组数据并关闭 XSD 组 “属性”对话框。
帮助	单击此按钮可显示此帮助主题。

UML	在编辑现有 XSD 组信息时会显示此按钮。 单击按钮打开 Group类的UML元素 属性”对话框。
-----	--

注记

- A Group元素可以：
- 仅包含 «XSDelement» 原型UML属性
- 包含复杂类型和简单类型作为子元素
- 成为其他复杂类型、简单类型和组的关联连接器的源
- 成为复杂类型元素的关联连接器的目标
- 概括连接或连接的源或目标

任何

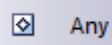
«XSDany» 构造型类允许复杂类型类包含架构包中未指定的元素。您可以通过从XML Schema工具箱中拖动 Any 图标并将其直接放到图表上来创建它。

工具箱图标



访问

要为选定的 «XSDany» 原型类显示 “XSD 任何属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDany» 刻板印象类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDany» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDany» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 Any元素的默认名称，请用另一个名称改写它。
命名空间	(可选) 类型命名空间包含可以在复杂类型中使用的元素。
过程内容	(可选) 单击下拉箭头并选择定义 XML解析器应如何验证这些元素的值： <ul style="list-style-type: none"> • 松懈 - 尝试根据其架构验证元素；获取不到架构时不提示错误 • skip - 跳过验证元素 • 严格 - 根据其架构验证元素并在无法获得架构时标记错误
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮以保存输入的任何“元素数据并关闭 XSD 任何 属性”对话框。
取消	单击此按钮可丢弃输入的任何元素数据并关闭 XSD 组 属性”对话框。
帮助	单击此按钮可显示此帮助主题。

UML	当您编辑现有的 «XSDany»元素信息时，会显示此按钮。 单击按钮打开 Any类的UML元素 属性”对话框。
-----	--

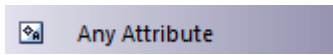
注记

- 任何类：
- 不能包含任何UML属性或子 XSD 类
- 不能是任何 XSD类的子级
- 不能成为概括器的目标
- 不能是源或概括或连接的关联
- 可以是复杂类型、组和模型组的关联连接器的目标
- 必须是至少一个传入关联连接器的目标

任何属性


«XSDany» 原型UML属性允许复杂元素类型或属性组元素包含未在架构包中指定的属性。您可以通过从XML Schema工具箱中拖动 “Any Attribute” 图标并将其放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类上来创建它。

工具箱图标



访问

要为选定的 «XSDany» 原型UML属性显示 “XSD anyAttribute属性” 对话框，请使用此处概述的方法之一。

功能区	在图表上选择特定的 «XSDany» 原型UML属性： 设计>元素>特征>属性
上下文菜单	在图表上选择特定的 «XSDany» 原型UML属性： 右键单击属性 视图属性
键盘快捷键	在图表上选择特定的 «XSDany» 原型UML属性： F9
其它	双击 «XSDany» 原型UML属性，或 从工具箱中拖动  “Any Attribute” 图标并将其拖放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类（这将创建一个新的 «XSDany» 原型UML属性）

定义属性

字段/按钮	行动
名称	如果您不想使用属性的默认名称，请用另一个名称改写它。
命名空间	（可选）类型命名空间包含可以在复杂类型或属性组元素中使用的属性。
过程内容	（可选）单击下拉箭头并选择定义 XML 解析器应如何验证这些属性的值： <ul style="list-style-type: none"> • 松懈 - 尝试根据其架构验证属性；获取不到架构时不提示错误 • skip - 跳过验证属性 • 严格 - 根据其架构验证属性并在无法获得架构时标记错误
注解	（可选）类型any注记你需要这个属性。

确定	单击此按钮保存输入的属性数据并关闭 XSD anyAttribute '属性' 对话框。
取消	单击此按钮可丢弃输入的属性数据并关闭 XSD anyAttribute '属性' 对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 «XSDany» 属性信息时，会显示此按钮。 单击按钮打开 «属性» 属性的属性属性。

注记

- 只有复杂类型和属性组元素可以有这个UML属性

联盟


联合类是一个简单类型元素，它定义了一个简单类型的集合。您可以通过从XML Schema工具箱中拖动 Union 图标并将其直接放在图表上来创建它。

工具箱图标




访问

要为选定的 «XSDUnion» 原型类显示 “XSD 联合属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDUnion» 刻板印象类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDUnion» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDUnion» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用联合的默认名称，请用另一个名称改写它。
会员类型	<p>单击  按钮以显示 “XSD 联合成员”对话框，并选择内置 XSD 数据类型和简单类型元素作为集合的成员。</p> <ul style="list-style-type: none"> 选择 - 不是在 “类名称”字段中键入或选择值，而是单击此按钮以显示 “选择分类器”浏览器并找到并选择一个简单类型元素；单击确定按钮关闭浏览器并立即将所选元素添加到 “类型详细信息”列表 该选项一般用于指定与 Union元素在同一个包中的对象，但您也可以选择任何其他包中的对象 添加 - 单击此按钮可将 “类名称”字段中指定的数据类型或元素添加到 “类型详细信息”列表 接受分类器即使不在模型中 - 选中此复选框以包含已命名但不存在于与联合元素相同的模型包中的元素或数据类型 类型详细信息 - 审阅选定元素或数据类型的列表；如果您打算从列表中删除一个object，请突出显示它并单击删除选定按钮 删除已选择 - 单击此按钮可从 “类型详细信息”列表中删除当前选择的分类器

	<ul style="list-style-type: none"> • 关闭 - 单击此按钮可关闭 “XSD 联合成员”对话框并在 “成员类型”字段中列出所选元素和数据类型
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的属性数据并关闭 “XSD 联合 属性”对话框。
取消	单击此按钮可丢弃输入的属性数据并关闭 “XSD 联合 属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	<p>当您正在编辑现有的 «XSDunion»元素信息时，会显示此按钮。</p> <p>单击按钮打开 «XSDunion»元素的UML元素 “属性”对话框。</p>

注记

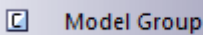
- 当您单击 “属性联合元素”对话框上的关闭按钮时，将添加一个从概括联合到同一图表上的每个成员元素的图表；任何不在同一张图上的元素都列在 “XSD 联合元素”的右上角
- 如果未列出与元素不同的成员类型，请选择 “开始>外观>首选项>首选项>图表>行为”并选中 “显示隐藏的父项”复选框
- A联合元素：
 - 不能包含任何子类
 - 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型元素的关联连接器的目标
 - 可以是简单类型元素中的概括器的目标

集团模型

您可以通过从XML Schema工具箱中拖动模型组图标并将其直接放到图表上来创建 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类。

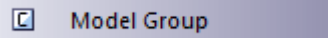
«XSDsequence»模型组（默认模型组类型）是类拥有的属性和关联的容器。模型组依次添加到类各自所有者的模型组中。类所有者指定的标记值一直持续到模型组的子元素；如果 memberNames 对于 complexType 不合格，则此模型组的子代在添加到该 complexType 时也是如此。

工具箱图标



访问

要为选定的 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类显示 “XSD模型组属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDsequence»、«XSDchoice» 或 «XSDall»属性类
键盘快捷键	Alt+Enter
其它	<ul style="list-style-type: none"> • 双击 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类，或 • 将  图标从工具箱拖到图表上（这将创建一个新的模型组元素；您可以从 «XSDsequence»、«XSDchoice» 或 «XSDall» 构造型中进行选择，其中 «XSDsequence» 是默认值）

定义属性

字段/按钮	行动
名称	如果您不想使用模型组的默认名称，请用其他名称改写它。
集团模型	单击下拉箭头并选择定义该组的子元素应如何出现在复杂类型类中的值： <ul style="list-style-type: none"> • 序列指定子元素必须以指定的顺序出现；创建一个 «XSDsequence» 原型类 • 选择 - 指定只能出现一个子元素；创建一个 «XSDchoice» 原型类 • all - 指定子元素可以以任何顺序出现；创建一个 «XSDall» 原型类
MinOccurs	（可选）类型该元素必须在类中出现的最小次数。 类型0 表示该元素是可选的。 默认值为1。

最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1 。
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的模型组数据并关闭 XSD元素的 属性"对话框。
取消	单击此按钮可丢弃输入的模型组数据并关闭 XSD元素的 属性"对话框。
帮助	单击此按钮可显示此帮助主题。
UML	编辑现有模型组元素信息时会显示此按钮。 单击 按钮打开模型组类的UML元素 属性"对话框。

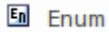
注记

- A模型组：
- 只能包含 «XSDelement» 构造型UML属性
- 可以包含复杂类型和简单类型作为子元素
- 可以是复杂类型、简单类型、组和模型组源的关联连接器的来源
- 必须是来自复杂类型的至少一个传入关联连接器的目标
- 不能是概括或连接的源或目标

枚举

枚举定义了类的可接受值的列表。您可以通过将 Enum 图标从XML Schema工具箱中拖放到图表上来创建枚举元素。

工具箱图标




访问

要为选定的 枚举“原型元素显示 XSD 枚举属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«枚举»定型元素 属性
键盘快捷键	Alt+Enter
其它	双击 枚举“定型元素，或 将  Enum 图标从工具箱中拖放到图表上（这将创建一个新的 枚举“原型元素）

定义属性

字段/按钮	行动
名称	如果您不想使用枚举的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSDsimpleType元素
价值观	类型所选类型的每个值，以逗号分隔。 这些值列在作为属性的元素上。
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的枚举元素数据并关闭 XSD 枚举的 属性”对话框。
取消	单击此按钮可丢弃输入的枚举元素数据并关闭 XSD 枚举 属性”对话框。

帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的枚举元素信息时，会显示此按钮。 单击按钮打开枚举类的UML元素“属性”对话框。

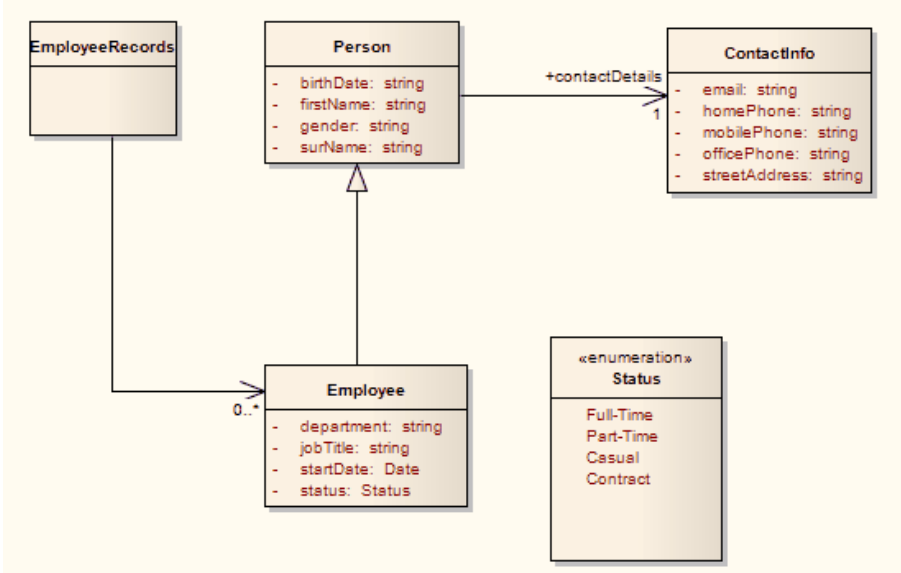
注记

- 一个枚举：
- 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
- 不能包含任何子类
- 不能是源连接器的关联
- 可以成为一个概括器的目标
- 最多可以有一个概括类型的连接类

来自抽象类模型的 XML

您可以仅使用简单的抽象类模型来建模 XML 模式。例如，这使得架构师可以开始在更高的抽象级别上工作，而无需关心架构的实现细节。虽然这种抽象模型随后可以使用工具箱的“XML Schema”页面进行工具箱，但也可以直接由Enterprise Architect的架构生成器生成，在这种情况下，架构生成器应用一组默认映射来转换将抽象模型转换为 XSD 文件。

示例

结构	细节
<p>图表</p>	<p>这是早期员工详细信息示例模型的简单类元素版本。它不使用 XSD 特定的刻板印象或标记值。</p>  <pre> classDiagram class EmployeeRecords class Person { - birthDate: string - firstName: string - gender: string - surName: string } class Employee { - department: string - jobTitle: string - startDate: Date - status: Status } class ContactInfo { - email: string - homePhone: string - mobilePhone: string - officePhone: string - streetAddress: string } class Status { Full-Time Part-Time Casual Contract } EmployeeRecords "0..*" --> Employee Employee < -- Person Person "1" --> ContactInfo : +contactDetails </pre>
<p>架构</p>	<p>该模式片段可以从示例模型中生成：</p> <pre> <?xml 版本=" 1 .0"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:simpleType name="状态"> <xs:restriction base="xs: string "> <xs:enumeration value="全职"/> <xs:enumeration value="兼职"/> <xs:enumeration value="休闲"/> <xs:enumeration value="合约"/> </xs:限制> </xs:simpleType> <xs:元素名称=" Person " type=" Person "/> <xs:complexType name=" Person "> <xs:序列> </pre>

```
<xs:元素名="firstName" type="xs: string" />
<xs:元素名="surName" type="xs: string" />
<xs:元素名="birthDate" type="xs: string" />
<xs:元素名="gender" type="xs: string" />
<xs:元素="contactDetails" type="ContactInfo"/>
</xs:sequence>
</xs:complexType>
<xs:元素名称="Employee" type="Employee"/>
<xs:complexType name="员工">
<xs:复杂内容>
<xs:extension base=" Person ">
<xs:序列>
<xs:元素名称="状态" type="状态"/>
<xs:元素名="jobTitle" type="xs: string" />
<xs:元素名称="startDate" type="xs:date"/>
<xs:元素名="department" type="xs: string" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:元素名称="EmployeeRecords" type="EmployeeRecords"/>
<xs:complexType name="EmployeeRecords">
<xs:序列>
<xs:元素="Employee" type="Employee" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:元素名称="ContactInfo" type="ContactInfo"/>
<xs:complexType name="ContactInfo">
<xs:序列>
<xs:元素名="homePhone" type="xs: string" />
<xs:元素名="mobilePhone" type="xs: string" />
<xs:元素名="officePhone" type="xs: string" />
<xs:元素名="email" type="xs: string" />
<xs:元素名="streetAddress" type="xs: string" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

默认UML到 XSD 映射

当您使用抽象类模型定义简单模式时，Enterprise Architect架构生成器使用UML到 XSD 构造的默认映射将UML信息转换为 XSD。架构生成器也使用这些默认值在抽象模型中生成未定型的元素。

当您使用工具箱的“XML Schema”页面的模型XML Schema图表，工具箱元素的构造型和标记值将覆盖默认映射。

结构体

UML构建	默认 XSD 生产规则
包	<p>A目标包生成架构元素。如果目标包中包含另一个包中的Classes，并且设置了标记值targetNamespace 和targetNamespacePrefix，则这些都作为架构元素的属性包含在内。</p> <p>此外，为每个引用的包创建一个导入或包含元素：</p> <ul style="list-style-type: none"> 如果外部包与目标包共享相同的targetNamespace标记值，则使用包含元素值 在 targetNamespaces 不同的地方使用导入元素
类	<p>生成根级元素声明和 complexType A。元素名和类型与类名相同。还会生成一个 XSD序列模型组，以包含作为元素生成的UML属性。</p>
属性	<p>为每个类属性声明一个元素。元素名称设置为UML属性名称。这以类名为前缀，以使元素唯一。minOccurs 和 maxOccurs 属性设置为反映属性基数。</p> <p>如果属性引用另一个类，则元素声明后跟一个 complexType 定义，其中包含对适当 complexType 的引用。</p>
关联	<p>为类所拥有的每个关联声明一个元素。元素名称设置为关联角色的名称。minOccurs 和 maxOccurs 属性反映了关联的基数。</p>
(概括继承	<p>对于单一继承，会生成一个扩展元素，并将基本属性设置为基本类名。然后将子类的UML属性附加到扩展元素内的 XSDall模型组。</p>
枚举	<p>为枚举声明A simpleType元素，并将 name 属性设置为枚举名称。A Restriction元素是在 base 设置为string的情况下生成的。每个枚举属性都附加到限制元素作为 XSD枚举元素，其值设置为UML属性名称。模式生成器会忽略UML属性的任何类型规范。</p>

注记

- 如果未指定，则 minOccurs 和 maxOccurs 属性默认为1
- 如果未指定关联的方向，则假定所有者为源

生成XSD

当您开发XML Schema模型时，无论是作为抽象类模型还是定制的 XSD类模型，您都可以使用生成XML Schema特征将其转发工程为 W3C XML Schema (XSD) 文件。由于XML Schema对应于Enterprise Architect中的UML包，因此XML Schema生成是包级别的操作。

您可以在模型的架构包元素中定义要生成XML Schema的文件的位置。

访问

功能区	开发>架构建模>导出XSD
-----	---------------

生成架构文件

选项	行动
编码	任何一个： <ul style="list-style-type: none"> 单击下拉箭头并选择要使用的 XML 编码方案，或者 单击默认按钮以应用默认编码方案 (UTF-8)
生成所有全局 ComplexTypes 的全局元素 (伊甸园风格)	默认选择生成伊甸园风格的架构，包含全局元素。 如果要省略全局元素，请清除复选框。
为引用包生成XSD	选中该复选框可为在此对话框中选择的任何包引用的包生成架构。
缺少文件名时提示	如果缺少生成架构文件的路径，请选中复选框以在生成架构期间提示引用包的文件名。 如果未选择“为引用包生成XSD”选项，则此选项不可用。
使用相对路径来引用 XSD (如果 'schemaLocation' 标记为空)	选中复选框以在引用外部包时在 XSD 导入 (或 XSD 包含) 语句中使用相对路径，前提是被引用包上的 schemaLocation 标记为空。 您在 XSD架构属性”对话框 (架构元素的元素 属性”对话框) 上为引用和引用的属性原型包设置了“架构文件”字段，XSDschema正确确定相对路径。
子包生成XSD	选中复选框为所选包的子包生成架构，然后选择： <ul style="list-style-type: none"> 包括所有包 - 在列表框中列出父包下的所有子包，或 包括 <XSDschema> 包 - 仅列出那些具有构造型 «XSDschema» 的包 对于每个包，列表框显示包名和可以生成模式文件的文件路径 (如果已设置)。 要更改包的文件路径，请双击列表框中的条目，然后在提示字段中输入或浏览新文件路径。 如果包已经设置了文件路径，则默认选中其复选框，以生成 XSD 模式；如果您不想从该包生成 XSD 架构，您可以取消选中该复选框。 如果您对没有设置文件路径的包选择复选框，则会自动显示文件路径的提示。

生成	单击此按钮可为列表框中选定的每个包生成架构。
关	单击此按钮关闭对话框，而不保存您的选项选择。
视图架构	单击此按钮可查看为列表框中突出显示的包生成的架构。
进步	选择架构生成的进度。

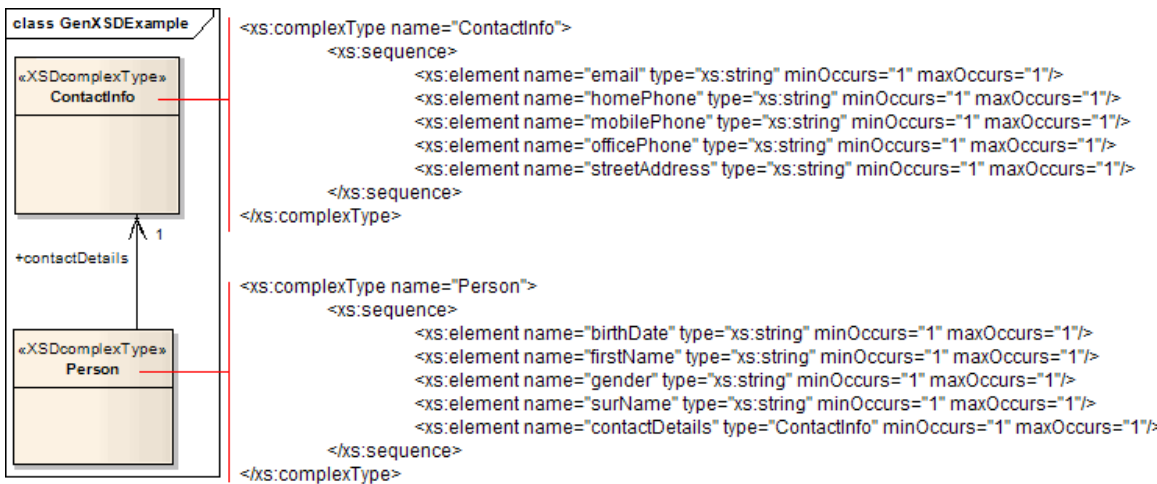
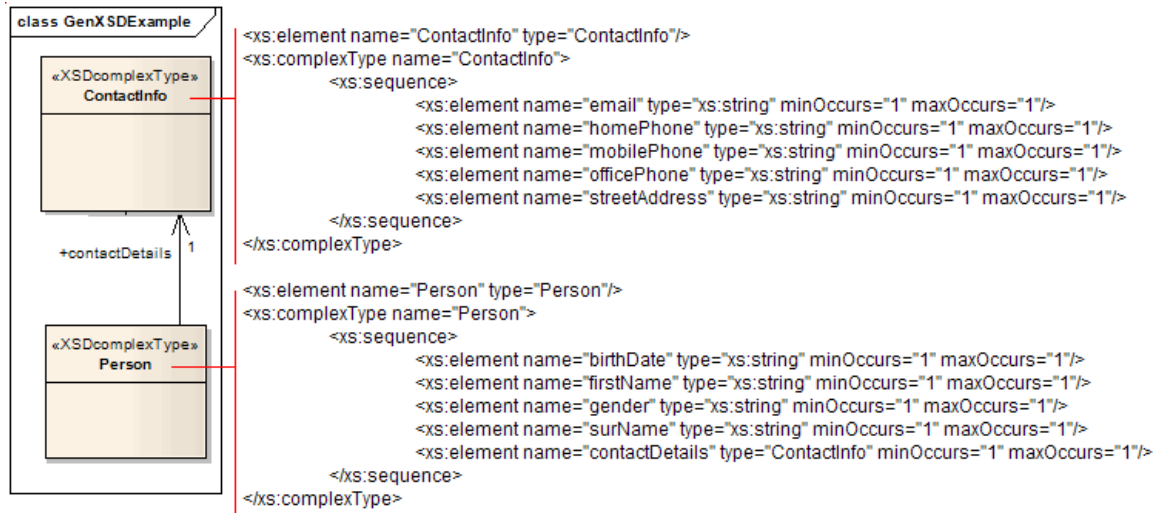
生成全局元素

默认情况下，Enterprise Architect以伊甸园风格生成XML Schema。对于每个全局 XSDcomplexType 原型类，系统都会生成一个全局元素。

示例

您可以通过取消选中 生成XML Schema “对话框上的 生成全局元素for all global ComplexTypes”复选框来更改指定的默认行为。然后，生成的 XSD 不再包含全局元素；也就是说，它不再有以下行：

- <xs:元素="ContactInfo" type="ContactInfo"/> 和
- <xs:元素名称=" Person " type=" Person "/>



导入XSD

要对 W3C XML Schema (XSD) 文件进行逆向工程以创建或覆盖UML类模型的包，您可以使用XML Schema导入功能。

访问

功能区	开发>架构建模>导入XSD
-----	---------------

导入架构文件

选项	行动
包	显示选定目标包的名称。
目录	类型在或浏览 (单击 ) 包含源XSD 文件的目录。
选定的文件	<p>列出当前可用于导入的XML Schema。</p> <ul style="list-style-type: none"> • 要选择单个文件，请单击它 • 选择多个单独的文件 Ctrl+单击每个文件 • 要选择文件范围，请按 Shift 并选择范围中的第一个和最后一个文件
以“类型”缀导入全局元素	选中此复选框可将全局元素和它所指的 ComplexType 视为两个单独的实体。
导入引用的XML Schema (s)	选中此复选框以导入在“选定文件”字段中选择的任何文件所引用的任何XML Schema。
图表XML Schema (s)	选中此复选框以在每个导入XSDschema包下创建一个类图。
导入XSD元素/属性为	<p>选择适当的单选按钮以指示如何将内联 XSD 元素和 XSD 属性导入到类中，可以是：</p> <ul style="list-style-type: none"> • UML关联 • UML属性
导入	单击此按钮开始 XSD 导入。
关	单击此按钮关闭对话框，而不保存您的选项选择。
进步	<p>显示指示架构导入进度的系统消息。</p> <p>在包含大量外部引用的导入中，捕获进度消息以准确检查已导入的内容可能很有用。为此，请右键单击消息并：</p> <ul style="list-style-type: none"> • 将所选消息复制到剪贴板 (选择“将所选消息复制到剪贴板”菜单选项) • 将所有消息复制到剪贴板 (选择“全部复制到剪贴板”菜单选项)，或

- | | |
|--|--|
| | <ul style="list-style-type: none">• 将所有消息保存到文件 (选择 “保存到文件” 菜单选项) |
|--|--|

注记

- 如果模型中已经存在正在导入的XML Schema文件，Enterprise Architect将跳过导入文件
- 对 XSD原始类型的引用总是作为UML属性导入
- 外部架构文件中对 XSD 结构的引用始终作为UML属性导入
- Enterprise Architect使用XML Schema的 XSD导入和 XSD 包含元素中的 schemaLocation 属性来确定文件之间的依赖关系；此属性必须设置为有效的文件路径（而不是 URL），才能正确导入依赖的XML Schema

全局元素和复杂类型

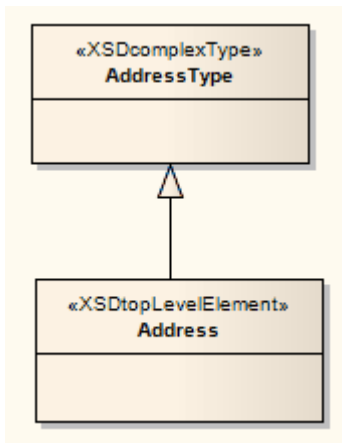
一些 XML Schema 具有与引用全局元素同名的 ComplexType 元素，但带有后缀“类型”，如下所示：

```
<xs:元素="Address" type="AddressType"/>
<xs:complexType name="AddressType">
<xs:序列/>
</xs:complexType>
```

默认情况下，在 XSD 导入时，Enterprise Architect 将此全局元素及其边界 ComplexType 视为单个实体，并创建一个与全局元素同名的单个 XSDcomplexType 原型类，如下所示：



您可以通过选择“导入XML Schema”对话框上的“使用类型”后缀的“导入全局元素”复选框来更改此默认行为。当您选择此选项时，系统会将全局元素和它所指的 ComplexType 视为两个单独的实体。例如，系统为全局元素创建一个«XSDtopLevelElement»原型类，为 ComplexType 创建一个«XSDcomplexType»原型类，连接如下所示：



注记

- Enterprise Architect 将这两个定义视为单独的实体，无论是否选中“导入带有类型”后缀的全局元素”复选框：

```
<xs:元素="HomeAddress" type="AddressType"/>
<xs:complexType name="AddressType">
<xs:序列/>
</xs:complexType>
```

XSL 转换

模型、作者和使用 XML 文档执行 XSLT 转换和样式表

Enterprise Architect提供建模和执行 XSL 转换的功能。XSLT 是一种可用于将 XML 输入文档转换为其他类型文档的技术。样式表是用于转换内容的 XSL 组件。功能包括：

- 用于对 XSLT 转换进行建模的专用图表和工具箱
- 用于样式表创作、调试和执行的专业编辑器。
- XML 文档验证
- XML Schema验证

模型使用 XML 转换图进行转换。在此图中，您可以创建 xml 文档和样式表，将它们链接到转换 (活动)，然后执行或调试转换。XSL Transform模型和 XML 文件的输入是 XSLT 和 XML文件，可以从工件工具箱中选择。这些工件通常通过将输出和 xsl 文件拖动到图表来创建。输出格式通常使用适当的输出变换来描述。转换的进度和成功/失败显示在系统输出窗口的“XSLT”选项卡上。

创建 XML 转换图表

节	行动
1	在浏览器窗口中，右键单击相应的包并选择“添加图表”选项。

工件元素在 XML 变换工具箱

工件	描述
XML 转换	转换的模型参考，提供输入和可选输出。用于运行或调试转换。 输入：XML文件、XSLT 输出：输出 (工件)
2	在“新图表”对话框中，在“名称”字段中输入适当的图表名称 (如果需要)，然后单击“选择来源”列表中的“扩展”和“图表类型”列表中的“XML 转换”。点击确定按钮。 新图表打开，显示“图表变换”页面的图形工具箱。
XSLT	标识要执行的样式表。 输入A N 输出A N
XML文件	标识要转换的输入文档。 输入A N 输出A N
XSD	标识可用于 (可选) 自动对输出文档执行 XML 验证的模式。 输入：输出工件、XML文件、(或可选)

	输出A N
输出工件	使用此工件操作来定义输出。当工件输出由转换创建时使用的文件路径。要选择名称输出文件，工件文件或文件以显示其属性并在“文件”选项卡下输入路径。使用工件，从元素中画出一个跟踪连接器。

手动验证文档

使用Enterprise Architect，您可以对要转换的文档和 XSLT 样式表执行 XML 验证。

要运行验证，请右键单击 XSL 调试器中的 XML 文档或样式表，然后选择“XML 验证”。将显示A提示以确认您是根据文档类型定义还是XML Schema进行验证。

- 对于文档类型定义，只需单击确定按钮；验证进行
- 对于 XML 模式，选择适当的单选按钮来确定验证语法是在文档中还是在其他地方定义；如果在其他地方，请输入语法的名称空间和 URL 或文件路径

如果在调试运行过程中发现错误，将输出到调试窗口（按Alt+8显示此窗口）。

如果在正常验证运行过程中发现错误，将输出到系统输出窗口的“XSL”选项卡（如果该窗口没有自动显示，请按Alt+1并选择“系统输出”）。要查找文档中的错误，请双击错误消息。

XSLT 处理器和版本

这些特征中使用的 XSL 处理器是从[Apache Xalan 项目](#)（C++ 版本1.11）构建的

XSL变换模型

当您使用 XSL 变换模型时，您可以绘制文件系统中已经存在的文件，也可以在模型元素中创建样式表和源的内容。

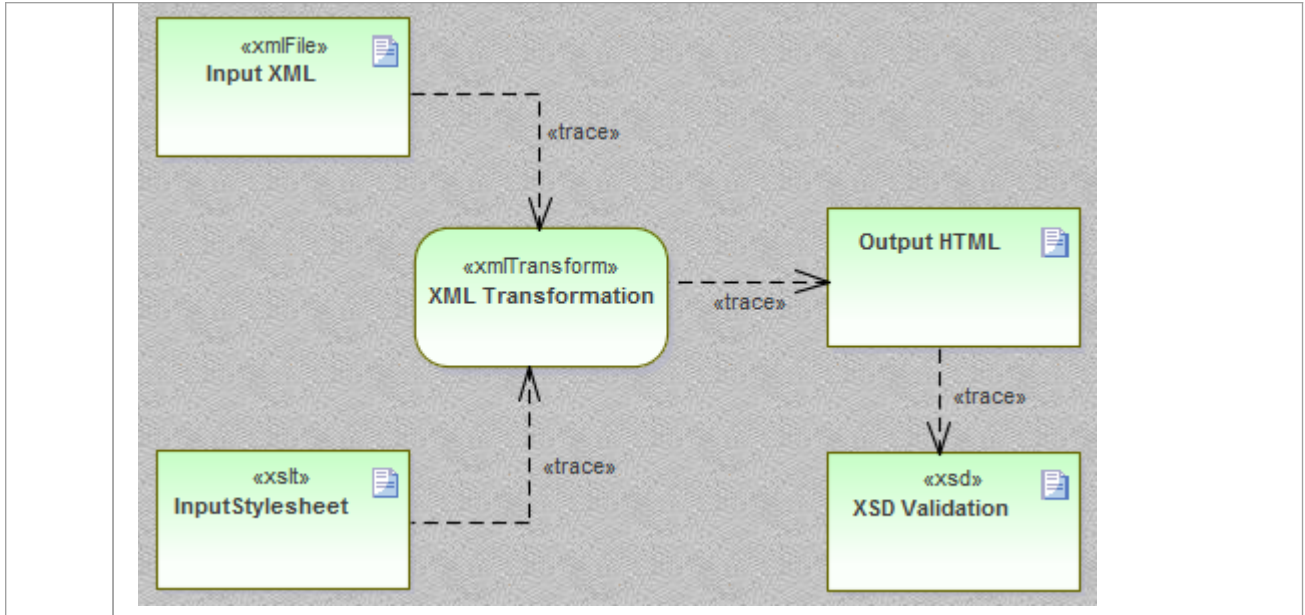
现有文件中的模型元素

这是建模转换的最简单和最常用的方法。当您把文件拖到 XML 转换图上时，元素工件然后，您可以使用跟踪连接器使用快速链接器将文件工件作为输入链接到元素 XML 转换。

或者，您可以：

- 指定一个输出位置（文件），方法是从 XML 文件或输出链接工件到一个替代位置（从 XML 工件为跟踪连接器）
- 通过拖动 XSD 模式文件并将工件元素连接到 XML 转换元素上的任何输出结果来验证文档

节	行动
1	打开文件浏览器和 XML 转换图。
2	在文件浏览器中，单击输入文件并将其拖到 XML 转换图上。 将显示 A 提示以将文件另存为： <ul style="list-style-type: none"> • 工件，将 XML 文件工件作为文件系统中的文件的快捷方式提供服务 • 工件，将文件内容读取到 XML 文件工件中并存储在模型文件中的位置；您将选择此选项以使源文件内容对模型的其他用户可用
3	选择“外部工件”选项。 为输入文件生成一个 XML 文件工件元素。
4	在文件浏览器中，单击 XSL 样式表文件并将其拖到 XML 转换图上。 作为对外部提示的回应，选择“外部工件”选项。 为样式表文件生成一个 XSLT 元素。
5	将 XML 转换图标从工具箱拖到图表上，以创建 XML 转换活动元素。如果你愿意，可以给这个元素一个新名字。
6	从“公共”工具箱页面拖动跟踪图标，在以下关系之间创建： <ul style="list-style-type: none"> • 输入 XML 文件元素及转换活动元素 • XSLT 样式表文件元素和转换活动元素
7	（可选的。） <ul style="list-style-type: none"> • 如果您想在文件中捕获输出，在文件浏览器中找到相应的文件并将其拖到图表上以生成另一个文件工件到该图表上；使用跟踪连接器将其链接到 XML 转换元素 • 如果要验证输出文档，请在文件浏览器中找到 XSD 模式文件并将其拖到图表上以生成 XSD 元素；将此工件到变换元素的输出文件工件（或任何输出）
8	按 Ctrl+S 保存图表内容。 如果输出是 HTML，您的图表可能类似于以下内容：



从零开始建模元素

当您使用“XML变换”工具箱创建 XSLT 文件元素时，系统会将这些文件存储为模型文档。双击 XML 转换图上的元素，在 Enterprise Architect 的 XSLT 调试器中打开模型文档，您可以在其中编写和编辑文件内容。保存文档后，内容将保存回模型中。

否则，建模转换的过程与从现有文件建模元素中描述的过程相同。

使用 XML 编辑器编辑文档

Enterprise Architect 提供了一个强大且有用的 XML 编辑器，具有许多特征，包括：

- 智能感知
- 提供快速替代导航的上下文结构树（提示：按 Ctrl+1 切换文档树视图）
- XSL 和 XSD 文档元素的自定义图标
- 代码完成和
- 验证文档和引用的模式

当在 Enterprise Architect 中打开任何带有 XML 声明的文档时，XML 编辑器将打开。（或者，按 Ctrl+Shift+O。） XSLT 调试器并排使用两个 XML 编辑器来显示样式表和正在转换的文档。

执行 XSL 变换

在对 XSL 变换建模后，您可以直接从模型图中执行它。您还可以直接从 XSL 样式表和输入文件执行转换。

执行图表the变换

节	行动
1	<p>在 XML 变换图上，右键单击 XML 变换活动元素并选择“运行XSL变换”选项。</p> <p>将显示 XSLT 调试器视图，其中显示了转换中使用的样式表 (.xsl) 文件和 XML 文档。</p> <p>系统输出窗口也会显示，在“XSL”选项卡中显示错误或成功消息。（如果系统输出窗口不显示，请按 Ctrl+Shift+8。）</p> <p>如果您设置了输出验证，系统输出窗口也会显示验证注释。</p>
2	<p>如果您已通过输出或文件工件将输出定向到文件，请按 F12 以查看输出。</p>

调试一个 XSL 变换

当您使用 XSLT 调试器运行转换时，您可以使用 Enterprise Architect 的调试器结合断点来控制过程并检查转换的状态。XSLT 调试器提供运行按钮和各种节按钮。您可以通过单击样式表的左边距来设置断点。

当一个步骤完成或遇到上下文时，可以在本地窗口窗口（执行>窗口>局部变量”）中查看转换的时间 - 包括模板调用的任何参数。也可以通过显示窗口（调用堆栈>>调用堆栈）查看转换的当前状态是如何到达的。

调试变换

节	行动
1	<p>在 XML 变换图上，右键单击 XML 变换活动元素并选择“调试 XSL 变换”选项。</p> <p>XSLT 调试器视图显示，其中显示了自动启动的转换中使用的样式表 (.xsl) 文件和 XML 文档。样式表中当前正在执行的语句被突出显示。</p> <p>视图顶部是一个调试器工具栏，提供开始、暂停、节结束、节输入、节输出和停止调试过程的常规调试选项。工具栏中的最后一个图标提供了在视图中隐藏或显示“.xml 源”选项卡的选项。您可以使用这些按钮重复和控制调试过程。</p> <p>系统输出窗口还会显示“XSLT 变换”选项卡上的调试器进度消息。（如果系统输出窗口没有显示，请按 Alt+1 并选择“系统输出”。）</p> <p>错误消息被定向到调试窗口（按 Alt+8）。您还可以使用调试窗口工具栏按钮和选项来控制 XSL 的变换。</p>
2	<p>如有必要，选择显示本地窗口和调用堆栈窗口。</p> <p>单击 XSLT 调试器样式表面板的左边距并设置您要用于检查处理的任何断点。</p>
3	<p>再次运行调试器检查系统输出窗口、调用堆栈调试器和本地窗口以及您要应用的任何其他执行分析工具所指示的执行情况。</p>

XML 验证

Enterprise Architect提供 XML 文档的验证。可以根据 XML 模式或数据类型定义 (DTD) 验证文档。验证是在 Enterprise Architect编辑器中使用其上下文菜单执行的。通常，XML 文档将包含与其所遵循的模式相关的信息。但是，您可以选择覆盖它，根据任何模式验证文档，无论是在本地计算机上的路径上还是在 URL 上。此示例演示了如何将特征用于包含不正确属性的文档。

访问

上下文菜单	可从显示上下文内容的任何编辑器窗口的时间菜单访问。 在编辑器窗口中右键单击并选择“XML 验证”
-------	---

XML 文档验证

节	行动
1	打开要验证的 XML 文档。
2	使用编辑上下文菜单并选择“XML Validation”选项。
3	从可用选项中选择选择的语法： <ul style="list-style-type: none"> XML Schema (默认) 数据类型定义
4	选择架构位置。默认选择“在文档中定义”。XML 文档通常会指定管理其内容的模式。要选择与文档中定义的架构不同的架构，请选择“外部”并提供 URL 或文件路径。例子： <ul style="list-style-type: none"> http://mydomain/myschema.xsd c:\mydomain\myschema.xsd
5	点击确定。验证的输出将显示在系统输出窗口的“XML 验证”选项卡中。

XML 文档验证示例

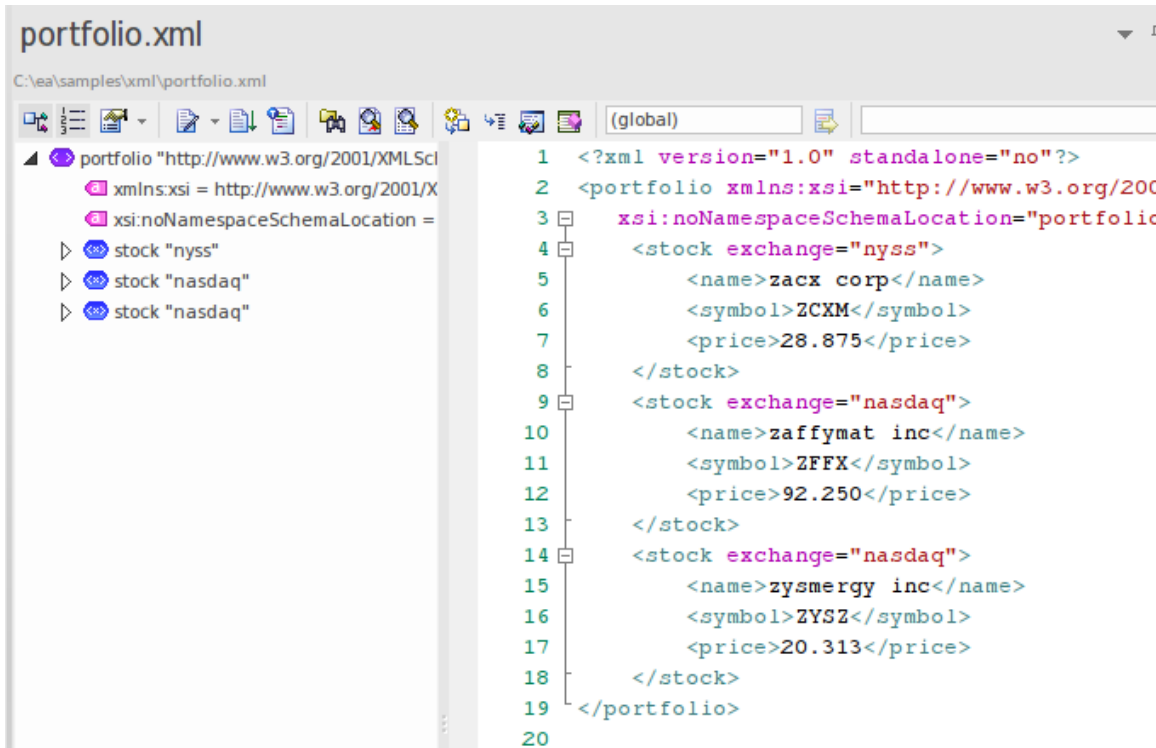


图1：具有无效属性值“nyss”的 XML 文档

在此示例中，文档描述了具有无效交换代码“nyss”的库存项目。从这个模 可以看出，“exchange”属性的唯一有效值是“nyse”、“nasdaq”或“ftsi”。

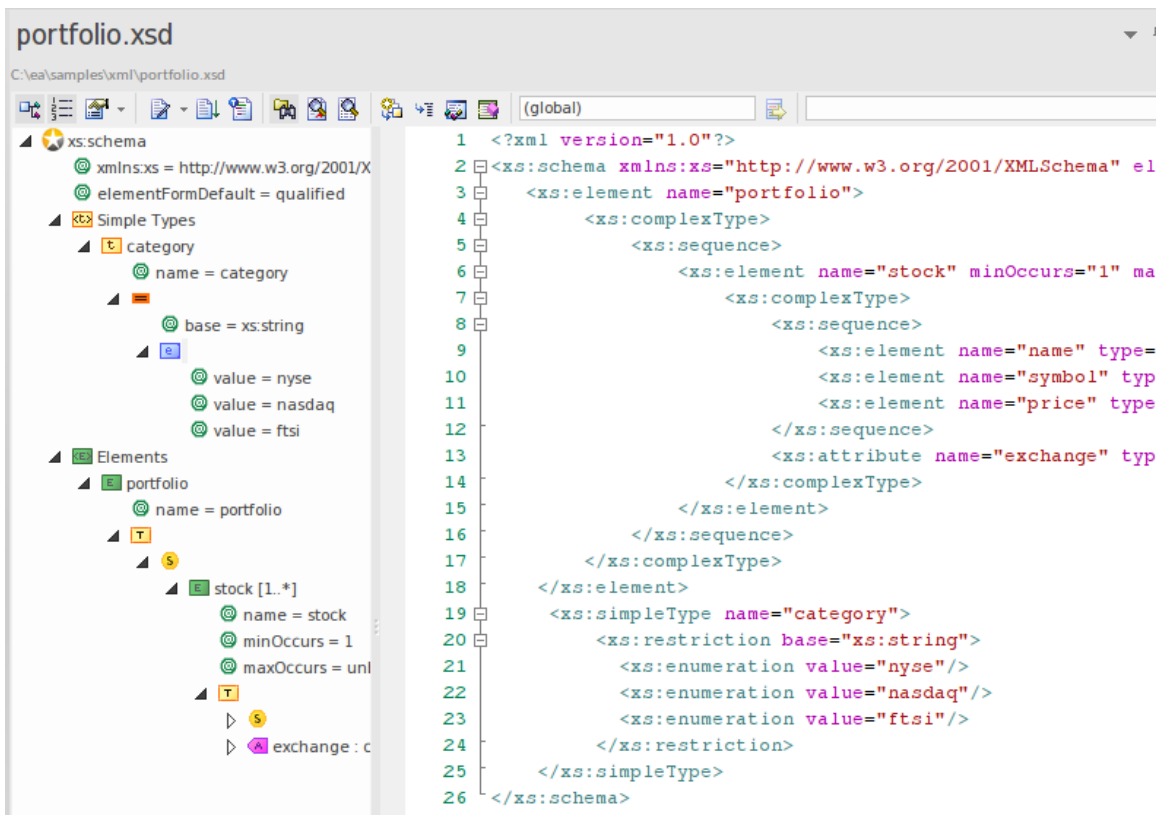


图2：描述允许的证券交易所代码的XML Schema

此图像显示了验证中使用的架构。在这里可以看到“投资组合”元素的声明由一个或多个“股票”元素组成。反过来，每个股票元素都需要一个“交易所”属性来命名相关证券交易所的代码。

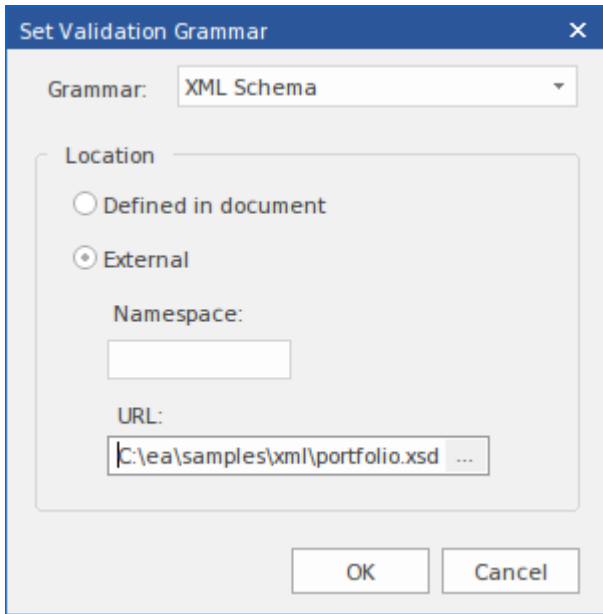


图 3：命名本地模式文件的“XML 验证”对话框

这是“XML 验证”对话框。它可以从 Enterprise Architect 中包含 XML 内容的任何编辑器的上下文菜单访问。您可以在此处选择要在验证中使用的架构。在示例中，处理器将使用本地模式文件验证文档。这恰好是文档命名的相同模式，但它可以是任何模式（例如模式的开发或更高版本）。

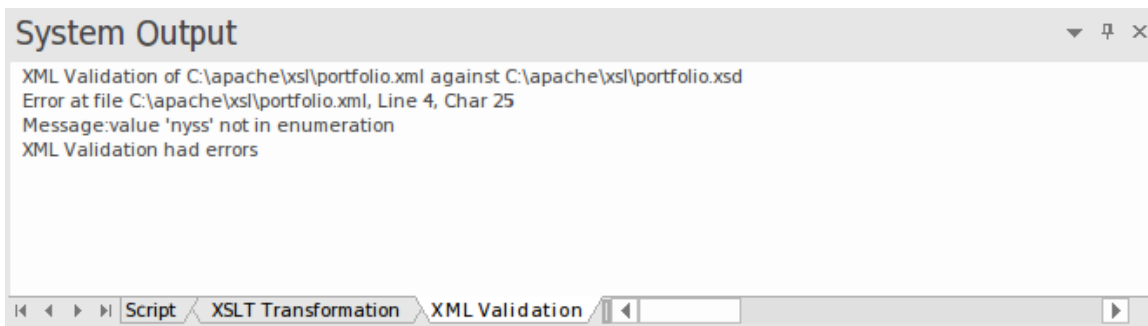


图 4：显示验证错误的系统输出窗口

此图像显示了验证的结果。根据模式描述的枚举，属性值“nyss”已被识别为不正确。双击错误将在编辑器中显示可以轻松更正的代码行。

面向 XML 服务架构

最大化敏捷性、规模和互操作性模型性的组织、社区和系统

面向服务架构(SOA)是一种架构范式，用于定义人员、组织和系统如何提供和使用服务以实现结果。

服务是A定义明确的界面向另一个人提供价值，该界面可供社区（可能是一般公众）使用。A服务导致工作由另一个人提供。

面向服务的架构(SOA)是一种组织和理解（表示）组织、社区和系统的方式，以最大限度地提高敏捷性、规模和互操作性。SOA方法很简单——人员、组织和系统相互提供服务。这些服务使我们能够在不自己动手甚至不知道如何做情的情况下完成某事——使我们更加高效和敏捷。服务还使我们能够向他人提供我们的能力以换取一些价值——从而建立一个社区、流程或市场。SOA范式同样适用于集成现有功能以及创建和集成新功能。

（源自面向服务的架构建模语言 (SoaML) - UML配置文件和服务元模型 (UPMS) 规范 (OMG 文档 ad/2008-11-01)；第 25-26 页。）

在Enterprise Architect中建模和开发完成面向服务的架构时，您可以使用以下任何或全部：

- XML Schema Definition (XSD)，也称为XML Schema一种XML技术，用于指定XML文档必须遵守的规则；Enterprise Architect提供了Schema编辑器界面，帮助您模型和生成XML模式
- XSL 转换使用 XSL 样式表将输入文档转换为 XML 或其他类型的文档，为此您可以使用 XSLT 编辑器和调试器来建模和执行转换
- 网络服务描述语言 1.1 (WSDL) - 一种基于 XML 的关键语言，用于描述 Web 服务
- 面向服务的架构建模语言 (SoaML) - 使用 Unified Modeling Language (UML) 设计和建模 SOA 解决方案的标准方法
- 面向服务的建模框架 (SOMF) - 一种面向服务的开发生命周期方法论，提供了许多有助于成功的面向服务的生命周期管理和建模的建模实践和学科
- 国家信息交换NIEM建模- 一个通用框架，用于定义如何在系统、政府机构和部门之间共享信息
- 元对象功能(MOF) - 一种物件管理组 (OMG) 标准，作为一种元建模架构来定义UML，因此提供了一种定义语言或数据的结构或抽象语法的方法

WSDL

网络服务描述语言 1.1 (WSDL) 是一种关键的基于 XML 的万维网络联盟 (W3C) 语言，用于描述 Web 服务。WSDL 支持对于开发完整的完成服务的架构(SOA) 至关重要，UML 2.5 和 XML 的耦合为在组织内指定、构造和部署基于 XML 的 SOA 工件提供了自然机制。

使用 Enterprise Architect，您可以快速模型、正向工程和反向工程 WSDL 文件。

WSDL 1 。 1模型结构

规范1下A网络服务描述语言 (WSDL) 。 1 ， 定义在一个«WSDLnamespace»原型包中 ， 它表示WSDL 元素的顶级容器 。 从概念上讲 ， 它映射到 WSDL 定义元素中的 targetNamespace 。

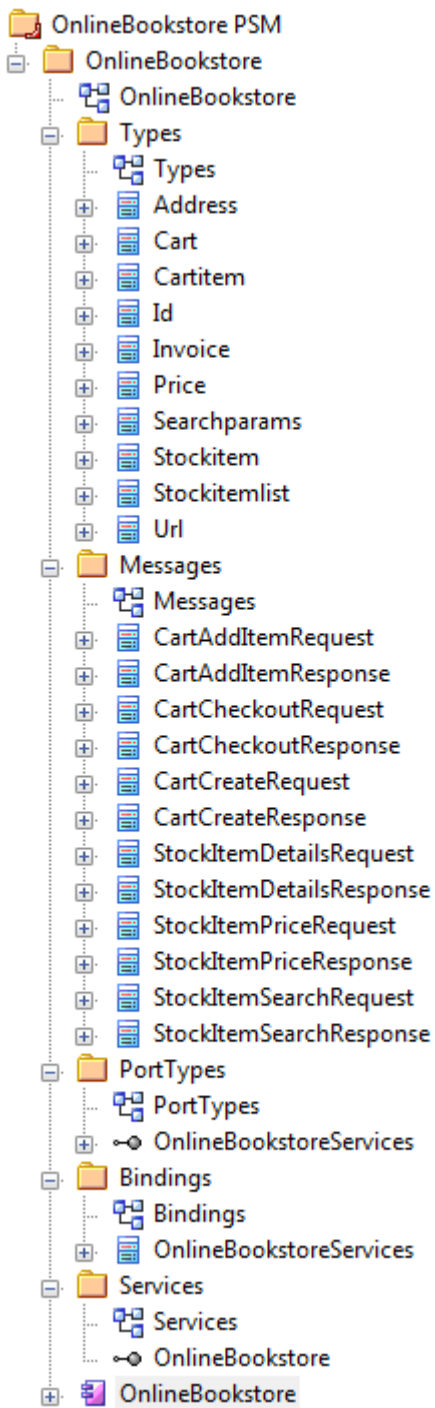
当您创建 WSDL模型时 ， Enterprise Architect会创建命名空间并提供一组子包 ， 每个子包都包含一个用于定义模型的组成元素的图表 ， 以及用于在子包之间导航的概览图 。 您通过序列中的子包来定义后面的对象使用的对象 ， 这些对象本身被调用到更后面的对象中 。

WSDL 结构开发

WSDL元素类型	描述
类型	在 XSD架构中定义 ， 这些是 Web 服务使用并由 WSDL 消息通信的 XSD 数据类型 ； 您将 «XSDelement» 、 «XSDsimpleType» 和 «图表» 原型元素从图形工具箱的 “XML Schema ” 页面拖到类型图上 。
留言	WSDL 消息标识 Web 服务正在传送的数据 。 每个信息元素都包含一个或多个信息部分 ， 这些信息部分是属性 ， 每个都标识正在通信的 XSD 数据类型 。
端口类型	WSDL端口类型是 Web 服务的基本核心 ， 定义了服务的接口 。 每个端口类型操作由一组端口类型组成 ， 每个端口类型都将消息的交换 (数据输入到接口的输出和从接口输出的数据标识为该操作) 。 端口类型操作还可以识别作为故障指示器的消息 。
绑定	捆绑指定为特定端口类型定义的操作和消息A协议和数据格式 。 每个«binding »类实现 (实现) «接口»接口指定的操作——端口类型元素中的端口类型操作会自动复制到捆绑元素中作为捆绑操作 。
服务	WSDL 服务定义了 Web 服务A正式接口 。 它描述了暴露特定捆绑的端口类型的集合 ， 每个暴露的捆绑都有一个关联 。 因此 ， 它封装了模型中定义的一组其他数据结构——如果不是所有数据结构的话 。
文件	WSDL 文档由具有构造型 «WSDL» 的组件表示 。 这是生成 WSDL 文件的元素 。

示例

此图显示了一个示例 WSDL 名称空间 OnlineBookstore PSM ， 它包括一个 WSDL 文档 OnlineBookstore (位于层次结构的底部) 。



注记

- 您还可以使用 WSDL模型变换从UML接口生成包结构

WSDL模型

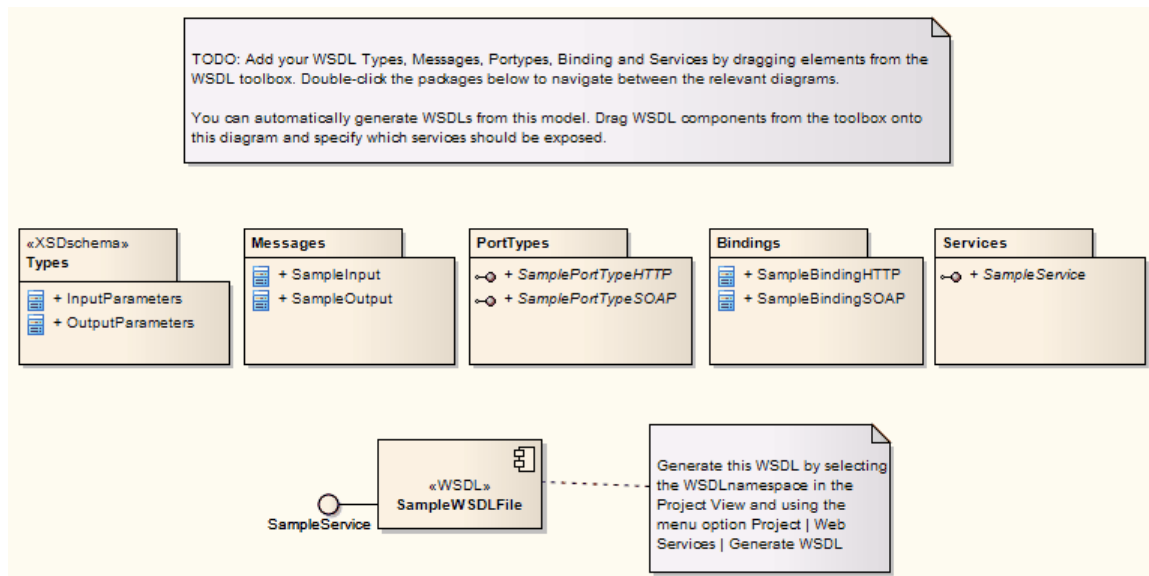
您可以使用图形工具箱的 WSDL 页面快速模型地为网络服务定义中的图表工具箱。作为第一步，您可以使用 WSDL 页面中的命名空间图标在浏览器窗口中创建一个示例 WSDL包结构。您可以将此示例包结构用作开发 WSDL 的模板。

创建一个新的 WSDL包结构

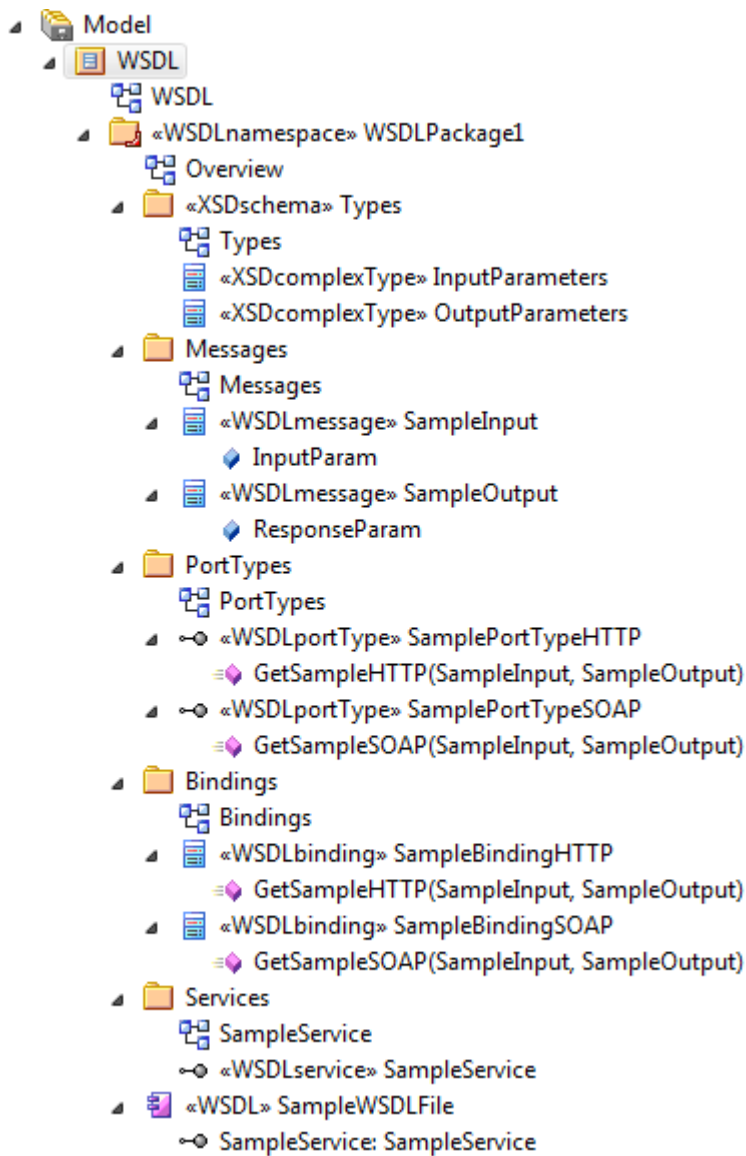
节	行动
1	在浏览器窗口中，创建您需要的顶层项目结构（模型和视图），然后单击相应的视图。
2	单击浏览器窗口标题下拉列表中的“新包”选项。 将显示“新模型包”对话框。
3	在“名称”字段中输入新包的名称，然后选择“创建图表”单选按钮。
4	单击确定按钮。 将显示“新图表”对话框。
5	在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。
6	单击确定按钮。 在浏览器窗口中，双击新图表名称旁边的图标；图表在“图表视图”中打开，类页面显示在“图表工具箱”中。
7	在  工具箱显示“查找工具箱项”对话框并指定“WSDL”，然后从结果中选择工具箱页面。 显示“WSDL”工具箱页面。
8	单击工具箱中的“命名空间”图标并将其拖到类图中。 将显示“WSDL命名空间属性”对话框。 WSDL包名和目标命名空间的 URL 中的类型。您可以稍后编辑这些值。
9	单击确定按钮。 在图表上创建样本«WSDLnamespace»原型包结构，并在浏览器窗口中显示、展开完整模型结构。 模型结构由这些子包组成，并带有概览图以在它们之间导航： <ul style="list-style-type: none"> • 类型：包含由 Web 服务通信的数据的 XSD 类型，在类型图上 • 消息：包含 WSDL 消息，建模为 UML 类，用构造型 «WSDLmessage» 标记 • PortTypes：包含 WSDL 端口类型，建模为 UML 接口，标有构造型 «WSDLportType» • 绑定：包含 WSDL 绑定，建模为实现 PortTypes 的 UML 类 • 服务：包含 WSDL 服务，建模为 UML 接口，并关联到每个暴露捆绑
10	在其对应的包中对每个模型构造进行建模。

模板 WSDL模型-图表

WSDL 命名空间充当 WSDL 结构的容器。



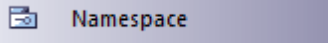
Template WSDL模型-浏览器Window Hierarchy



WSDL命名空间

«A» 原型包代表Enterprise Architect中 WSDL 构造的顶级容器。您可以通过将 WSDL工具箱页面中的命名空间图标拖放到图表上来创建命名空间包。

工具箱图标



访问

要为选定的 «属性» 原型包显示 “WSDL 命名空间属性”对话框，请使用此处概述的方法之一。

功能区	设计>包>管理>属性
上下文菜单	右键单击 «WSDLnamespace» 原型包 属性
其它	在浏览器窗口中，双击 «WSDLnamespace» 原型包，或将  图标从工具箱拖到图表上（这将创建一个新的 «WSDLnamespace» 原型包）

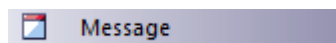
定义属性

选项	行动
WSDL包名称	类型在 WSDL命名空间包元素的名称中。
目标命名空间	(可选) 类型在 WSDL命名空间包的 URL 中。
确定	单击此按钮保存输入的值并关闭 WSDL命名空间的 “属性”对话框。如果您刚刚创建了命名空间，则会打开一个包含示例模板WSDL模型的新包图。
取消	单击此按钮放弃输入的数据并关闭 “WSDL命名空间属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 WSDL命名空间元素信息时，会显示此按钮。单击 按钮打开命名空间包元素的UML元素 “属性”对话框。

WSDL信息

«WSDLmessage» 原型类表示A WSDL信息并充当一个或多个 WSDL信息部件的容器。您可以通过将信息图标从 WSDL工具箱拖放到消息图上（在 WSDL包结构中的 Messages包下）来创建 WSDL 消息。

工具箱图标



访问

要为选定的 «WSDLmessage» 原型类显示 “WSDL信息”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «WSDLmessage» 原型类 属性
键盘快捷键	Alt+Enter
其它	<p>双击 «WSDLmessage» 原型类，或</p> <p>将  Message 图标从工具箱中拖放到 Messages 图表上，在 WSDL包结构中的 Messages包下。</p> <p>（这会创建一个新的 «WSDLmessage» 原型类。）</p>

定义属性

选项	行动
名称	WSDL信息名称中的类型。
文档	（可选）类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的数据并关闭 WSDL信息对话框。
取消	单击此按钮放弃输入的数据并关闭 “WSDL信息”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL信息元素信息时，会显示此按钮。单击按钮打开元素的UML类的 “属性”对话框。

注记

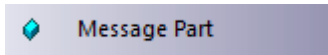
- WSDL 消息只能在 WSDL包结构中的 Messages包下创建
- WSDL信息的名称在 WSDL 内的所有 WSDL 消息中应该是唯一的

WSDL信息部件

WSDL信息部件是 WSDL信息A一部分，它标识由信息通信的数据的 XSD 数据类型。如果信息传递的数据不止一种数据类型，则每种数据类型都在单独的信息部件中标识。

在Enterprise Architect中，WSDL信息部件由 WSDL信息类的UML属性表示。您可以通过从 WSDL工具箱中拖动“信息部件”图标并将其放到«部件»原型类上来创建信息元件属性。

工具箱图标




访问

要显示选定信息元件的“部件信息部件”对话框，请使用此处列出的方法之一。

功能区	在图表上的 WSDL信息中选择特定的信息部件（属性）： 设计>元素>特征>属性
上下文菜单	在图表上的 WSDL信息中选择特定的信息部件（属性）： 右键单击属性 视图属性
键盘快捷键	在图表上的 WSDL信息中选择特定的信息部件（属性）： F9
其它	双击 «部件» 原型类中的信息元件（属性），或 将  图标从工具箱拖放到 «WSDLmessage» 原型类上（这会在 «WSDLmessage» 原型类中创建一个新的信息部件（作为属性）。

定义属性

选项	行动
名称	WSDL信息部件属性名称中的类型。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 «XSDelement»、«XSDcomplexType» 或 «XSDsimpleType»元素作为分类器 XSD 类型可以定义在： <ul style="list-style-type: none"> • WSDL包结构下的Types包或

	<ul style="list-style-type: none">模型中的任何其他包
确定	单击此按钮保存输入的数据并关闭“WSDL信息部件”对话框。
取消	单击此按钮可放弃输入的数据并关闭“WSDL信息部件”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL 信息部件属性信息时，会显示此按钮。 单击按钮打开信息部件的属性属性。

注记

- WSDLmessage 构造型类只能有效地包含信息部件属性；如果在类元素中添加其他属性，它们将被重新转换为信息部件

WSDL端口类型

«A» 原型接口表示 WSDL PortType。它描述了 WSDL 公开的操作，充当一个或多个 WSDL 端口操作类型的容器。您可以通过将 WSDL 端口类型图标从 WSDL 工具箱直接拖放到 PortTypes 图（在 WSDL 包结构中的 PortTypes 包下）来创建 WSDL PortType 元素。

工具箱图标



访问

要为选定的 «WSDLportType» 原型接口显示 “WSDL PortType” 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDLportType»原型接口 属性
键盘快捷键	Alt+Enter
其它	<ul style="list-style-type: none"> • 双击 «WSDLportType» 原型接口，或 • 将  图标从工具箱中拖放到 PortTypes 图上，在 WSDL 包结构中的 PortTypes 包下（这会创建一个新的 «WSDLportType» 原型接口。）

定义属性

选项	行动
名称	WSDL PortType 名称中的类型。
文档	(可选) 类型在您需要此元素的任何注释中。
确定	单击此按钮以保存输入的数据并关闭 WSDL PortType 对话框。
取消	单击此按钮放弃输入的数据并关闭 “WSDL PortType” 对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL PortType 元素信息时，会显示此按钮。单击按钮打开该元素的 UML 元素 “属性” 对话框。

注记

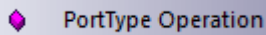
- WSDL PortTypes 只能在 WSDL包结构中的 PortTypes包下创建
- WSDL PortType 的名称在 WSDL 内的所有 WSDL PortType 中应该是唯一的

WSDL端口类型操作

A类型端口标识消息的交换（作为操作的接口的数据输入和输出）。端口类型操作还可以识别作为故障指示器的消息。


在Enterprise Architect中，WSDL PortType 操作由 WSDL PortType接口的UML操作表示。您可以通过将 PortType Operation 图标从 WSDL 工具箱拖放到 «工具箱» 原型接口上来创建 PortType Operation。

工具箱图标



访问

要为选定的 PortType 操作显示 “WSDL PortType 操作”对话框，请使用此处概述的方法之一。

功能区	在图表上的 接口“原型接口中选择特定的端口类型操作： 设计>元素>特征>操作
上下文菜单	在图表上的 接口“原型接口中选择特定的端口类型操作： 右键单击属性 视图属性
键盘快捷键	在图表上的 接口“原型接口中选择特定的端口类型操作： F10
其它	双击 «WSDLportType» 原型接口中的 PortType 操作，或 将  PortType Operation 图标从工具箱中拖放到 «WSDLportType» 原型接口上。（这会在 «WSDLportType» 原型接口中创建一个新的 PortType 操作（作为UML操作）。）

定义属性

选项	行动
名称	WSDL PortType 操作的名称中的类型。
文档	（可选）此操作所需的任何注记中的类型。
操作类型	单击下拉箭头并选择一种受支持的 PortType 操作类型： <ul style="list-style-type: none"> • 单程 • 请求-响应 • 请求响应 • 通知

输入	<p>如果您选择了通知作为操作类型，此部分将显示为灰色。</p> <ul style="list-style-type: none"> 名称- 默认为与操作类型平行的名称。如果您不想使用默认值，请键入输入信息的替代名称。 信息- 单击下拉箭头并选择先前在消息包中创建的 WSDL 消息之一。 文档 - (可选) 输入信息所需的任何注记类型。
输出	<p>如果您已选择 OneWay 作为操作类型，此部分将显示为灰色。</p> <ul style="list-style-type: none"> 名称- 默认为与操作类型平行的名称。如果您不想使用默认值，请键入输出信息的替代名称。 信息- 单击下拉箭头并选择先前在消息包中创建的 WSDL 消息之一。 文档 - (可选) 类型在您需要此输出信息的任何注记中。
故障	<p>审阅可以充当故障的 WSDL 消息的详细信息。</p> <p>故障显示在此列表中，最近创建的在顶部，最旧的在最后。如果定义了四个以上的故障消息，请使用垂直滚动条显示列表的其余部分。</p> <p>要添加信息，请单击新建按钮。将显示 WSDL PortType Operation Fault”对话框。</p> <ul style="list-style-type: none"> 名称- 默认为 'Fault<n>';如果您不想使用默认值，请键入故障信息的替代名称 信息- 单击下拉箭头并选择先前在信息包中创建的 WSDL 消息之一 文档 - (可选) 类型in any注记you need for this fault信息 确定- 单击此按钮保存输入的数据并关闭 WSDL PortType Operation Fault”对话框 取消 - 单击此按钮放弃输入的数据并关闭 WSDL PortType Operation Fault”对话框 帮助- 单击此按钮显示此帮助主题 <p>要从列表中删除信息，请单击它并单击删除按钮。</p>
确定	单击此按钮保存输入的数据并关闭 WSDL PortType Operation ”对话框。
取消	单击此按钮放弃输入的数据并关闭 WSDL PortType Operation ”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	<p>当您正在编辑现有的 WSDL 端口类型操作信息时，会显示此按钮。</p> <p>单击按钮打开元素的UML操作 属性”对话框。</p>

注记

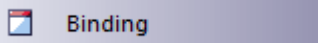
- WSDL操作只能由 WSDL PortTypes 包含
- 为 PortType 操作中的输入、输出或故障信息提供的名称在 WSDL PortType 中的输入、输出和故障消息中必须分别是唯一的
- 在UML操作 属性”对话框中，标识为 Input、输出和 Fault 的 Messages 可以作为操作的参数进行检查

WSDL 捆绑

A WSDL 捆绑元素实现由特定的 «接口» 原型接口指定的操作，并描述此 WSDL PortType 定义的操作和消息的消息格式和协议细节。A WSDL 捆绑元素由一个 «WSDLbinding» 原型类表示。

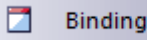
您可以通过将 WSDL 工具中的捆绑图标直接拖到 WSDL 包结构中工具箱包下的 Bindings 图上来创建一个 WSDL 捆绑元素。

工具箱图标



访问

要为选定的 «WSDLbinding» 原型类显示 “WSDL 捆绑”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «WSDLbinding» 原型类 属性
键盘快捷键	Alt+Enter
其它	<p>双击一个 «WSDLbinding» 原型类，或者</p> <p>将  图标从工具箱中拖放到 Bindings 图上，在 WSDL 包结构中的 Bindings 包下。</p> <p>(这会创建一个新的 «WSDLbinding» 原型类。)</p>

定义属性

选项	行动
名称	类型在 WSDL 捆绑元素的名称中。
端口类型	点击下拉箭头，选择此 WSDL 捆绑要实现 WSDL 捆绑。
协议	<p>单击下拉箭头并选择传输所选 WSDL PortType 操作的协议。支持的协议有：</p> <ul style="list-style-type: none"> • 肥皂 • HTTP
运输	<p>如果您选择了 HTTP 协议，此字段将被禁用。</p> <p>默认为 <code>http://schemas.xmlsoap.org/soap/http</code>。</p> <p>如有必要，输入 SOAP 协议的替代 URL。</p>
风格	如果您选择了 HTTP 协议，此字段将被禁用。

	单击下拉箭头并选择 SOAP 协议的样式。
动词	如果您选择了 SOAP 协议，此字段将被禁用。 单击下拉箭头并选择适当的 HTTP 动词。支持的动词有： <ul style="list-style-type: none">• 得到• 邮政
文档	(可选) 类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的数据并关闭 “WSDL 捆绑”对话框。
取消	单击此按钮可丢弃输入的数据并关闭 “WSDL 捆绑”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 WSDL 捆绑元素信息时，会显示此按钮。 单击按钮打开该元素的UML元素 “属性”对话框。

注记

- A WSDL 捆绑必须实现一个 WSDL PortType；因此，应该在创建 WSDL 绑定之前定义 WSDL PortTypes
- WSDL Bindings 只能在 WSDL 包结构中的 Bindings 包下创建
- WSDL 绑定的名称在 WSDL 内的所有 WSDL 捆绑中应该是唯一的

WSDL 捆绑操作

当你保存一个新创建的 «WSDLbinding» 原型类时，系统：

1. 添加到捆绑图中，WSDL 捆绑实现的 WSDL 端口类型元素。
2. 从捆绑到 PortType 绘制一个实现连接器。
3. 使用捆绑中的所有 UML 操作自动填充捆绑，作为 WSDL 捆绑操作。

访问

要显示所选捆绑操作的 “WSDL 捆绑操作详细信息” 对话框，请使用此处概述的方法之一。

功能区	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： 设计>元素>特征>操作
上下文菜单	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： 右键单击属性 视图属性
键盘快捷键	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： F10
其它	双击«捆绑»类中的捆绑操作

定义属性

选项	行动
操作名称	显示从 WSDL PortType 元素复制的操作的名称。 此字段中的值无法编辑。
行动	如果父级 WSDL 捆绑元素的协议定义为 HTTP，则该字段灰显。 此操作的 SOAP 行动标头 (URL) 中的类型。
风格	如果父级 WSDL 捆绑元素的协议定义为 HTTP，则该字段灰显。 单击下拉箭头并选择操作的 SOAP 样式。
地点	如果父 WSDL 捆绑元素的协议定义为 SOAP，则该字段灰显。 此操作的相对 URL 中的类型。
文档	(可选) 此操作所需的任何注记中的类型。
参数	单击此按钮可定义此操作的参数。 将显示 “WSDL 捆绑操作参数” 对话框，显示操作输入、输出和故障的名称。您不能更改这些名称。 单击详细信息按钮以指定输入、输出和故障操作 (信息) 参数的详细信息。请

	<p>注记·详细信息按钮在：</p> <ul style="list-style-type: none"> • 通知操作类型的输入部分被禁用 • 单向操作类型禁用输出部分 • 如果没有故障消息·则故障部分被禁用 • 使用-如果父WSDL捆绑元素的协议定义为HTTP·则该字段灰显；单击下拉箭头并选择要使用的编码 • Encoding Style - 如果父 WSDL捆绑元素的协议定义为 HTTP·则该字段灰显；如果 使用“设置为 编码”·请输入要应用的样式 (URL) • 命名空间-如果父WSDL捆绑元素的协议定义为HTTP·则该字段灰显； (可选) 在命名空间中输入 • Parts - 如果父 WSDL捆绑元素的协议定义为 HTTP·则该字段灰显；此字段也不适用于故障 - (可选) 出现在 SOAP体部分中的信息部件属性中的类型 • Header - 此字段不适用于故障； (可选) 输入 SOAP/HTTP 标头的文本 • 文档 - (可选) 类型在您需要此消息的任何注记中 • 确定- 单击此按钮保存输入的数据并关闭 “WSDL捆绑参数详细信息”对话框 • 取消 - 单击此按钮放弃输入的数据并关闭 “WSDL捆绑参数详细信息”对话框 • 帮助- 单击此按钮显示此帮助主题
<p>确定</p>	<p>单击此按钮保存输入的数据并关闭 “WSDL捆绑操作详细信息”对话框。</p>
<p>取消</p>	<p>单击此按钮可丢弃输入的数据并关闭 “WSDL捆绑操作详细信息”对话框。</p>
<p>帮助</p>	<p>单击此按钮可显示此帮助主题。</p>
<p>UML</p>	<p>当您正在编辑现有的 WSDL捆绑操作信息时·会显示此按钮。 单击按钮打开元素的UML操作 属性”对话框。</p>

注记

- 如果后续更改了 WSDL端口类型的操作·可以通过删除实现连接器并重新建立来刷新捆绑操作；将显示 覆盖和实现”对话框·您可以在其中选择要建立的更新操作
- 您可以通过在图表中突出显示操作或浏览器窗口并展开属性窗口中的条目来捆绑审阅的参数

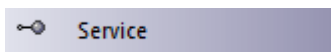
WSDL 服务

WSDL 服务由A «接口» 原型接口表示；它描述了暴露特定捆绑的端口集合。您可以通过将服务图标从 WSDL 工具箱拖放到您的 WSDL 模型的 Services 包中的图表上来创建 WSDL 服务元素。

当您保存新创建的 «接口» 原型接口时，系统：

1. 将 WSDL 服务公开的 WSDL 捆绑元素添加到服务图中。
2. 从服务元素到每个捆绑元素绘制一个关联连接器。
3. 用相应的端口名称标记每个连接器。

工具箱图标



访问

要为选定的 «接口» 原型接口显示 “WSDL 服务” 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDLservice»原型接口 属性
键盘快捷键	Alt+Enter
其它	<p>双击 «WSDLservice» 原型接口，或</p> <p>将  图标从工具箱中拖放到 SampleService 图上，在 WSDL 包结构中的 Services 包下。（这会创建一个新的 «WSDLservice» 原型接口。）</p>

定义属性

选项	行动
名称	WSDL 服务名称中的类型。
文档	(可选) 类型在您需要此元素的任何注释中。
端口	<p>标识此 WSDL 服务的端口 (或端点) 。</p> <p>要将端口添加到列表中，请单击 “新建” 按钮。将显示 “WSDL 端口” 对话框。</p> <ul style="list-style-type: none"> • 端口名称-类型中的端口名称 • 捆绑-点击下拉箭头，从 BindingsPackage 中创建的所有 WSDL Bindings 列表选择一个捆绑元素 • Location -端口的 URL 类型 • 文档 - (可选) 在您需要此端口类型的任何注释中

	<ul style="list-style-type: none"> • 确定- 单击此按钮保存输入的值并关闭 “WSDL端口”对话框 • 取消 - 单击此按钮放弃输入的值并关闭 “WSDL端口”对话框 • 帮助- 单击此按钮显示此帮助主题 <p>端口排列在列表中，最新的在顶部，最旧的在最后。</p> <p>要从列表中删除条目，请单击它并单击删除按钮。</p>
确定	单击此按钮以保存输入的数据并关闭 WSDL 服务对话框。
取消	单击此按钮放弃输入的数据并关闭 “WSDL 服务”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL 服务元素信息时，会显示此按钮。 单击按钮打开该元素的UML元素 “属性”对话框。

注记

- WSDL 服务只能在 WSDL包结构中的 Service包下创建
- WSDL 服务的名称在 WSDL 内的所有 WSDL 服务中应该是唯一的

WSDL 文档

WSDL 文档封装了在«WSDLnamespace» 原型包中定义A网络服务，并且是生成WSDL 文件的源。它由«WSDL» 原型组件元素表示为«部件» 原型包的直接子元素。您可以在单个 WSDL命名空间下拥有多个 WSDL 文档，以便跨多个 WSDL 重用和公开该名称空间的 WSDL 服务。


当您创建命名空间包结构时，会自动创建一个 «WSDL» 定型部件元素。您可以通过将 WSDL 图标从 WSDL工具箱拖放到名称空间概览图上来添加更多 WSDL 元素。

工具箱图标



访问

要为选定的 «WSDL» 原型部件显示 “WSDL 文档属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDL»构造型部件 属性
键盘快捷键	Alt+Enter
其它	双击一个«WSDL»原型部件，或者 将  WSDL 图标从工具箱中拖放到概述图上，在 WSDL包结构中的 «WSDLnamespace» 原型包下。（这将创建一个新的 WSDL 文档，由 “WSDL”原型部件表示。）

定义属性

选项	行动
名称	WSDL 文档名称中的类型。
文件名称	类型WSDL 1所在的文件路径。将生成1文件。
文档	(可选) 类型在您需要此元素的任何注记中。
XMLNS	识别此 WSDL 文档中使用的附加名称空间或名称空间前缀对。 要添加命名空间或命名空间前缀对，请单击 “新建”按钮；要编辑现有条目，请双击它。在任何一种情况下，都会显示 “命名空间详细信息”对话框。 <ul style="list-style-type: none"> 前缀类型在缩略文本中代表命名空间 命名空间-命名命名空间名称中的类型 确定- 单击此按钮保存新信息并关闭 “命名空间详细信息”对话框 取消 - 单击此按钮可放弃新信息并关闭 “命名空间详细信息”对话框

	<ul style="list-style-type: none">帮助- 点击此按钮显示此帮助主题 要从列表中删除条目，请单击它并单击删除按钮。
服务	审阅服务包中存在的 WSDL 服务。 选中要包含在当前 WSDL 文件中的服务对应的复选框。
确定	单击此按钮保存输入的数据并关闭 WSDL 文档的“属性”对话框。
取消	单击此按钮放弃输入的数据并关闭 WSDL 文档“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL 文档元素信息时，会显示此按钮。 单击按钮打开该元素的UML元素“属性”对话框。

生成WSDL

如果您已经在UML中开发了一个 WSDL 模型，您可以将它转发工程到 WSDL 1 中。1 文件使用生成的 WSDL 特征。该特征作用于 WSDLnamespace“原型包或 WSDL”原型部件（文档），并生成目标 WSDLnamespace“结构拥有的任何或所有WSDL组件。

访问

功能区	开发>架构建模>导出WSDL
-----	----------------

生成WSDL 1.1文件

选项	行动
WSDL包	显示包含要从中生成 WSDL 文件的源部件的 WSDL命名空间的名称。
编码	任何一个： <ul style="list-style-type: none"> 单击下拉箭头并选择您需要的 XML 编码方案，或 单击默认按钮以应用默认编码方案 (UTF-8)
选择要生成组件	单击要为其生成 WSDL 文件的列表中的 «WSDL» 原型部件。至： <ul style="list-style-type: none"> 使用 Ctrl+单击选择多个单独的组件 使用 Shift+单击选择范围 选择列表中的所有条目单击“全选”按钮 清除列表中的所有条目单击“选择无”按钮 提供为组件生成 WSDL 文件的文件路径和名称，双击组件名称；将显示“部件文件名称”对话框，请表中的说明
生成	单击此按钮为选定的 «WSDL» 原型组件生成 WSDL 文件。生成完成时会显示A消息；单击消息上的确定按钮将其关闭。
视图WSDL	单击此按钮可显示最近生成的 WSDL。
关	单击此按钮可关闭此对话框。
帮助	单击此按钮可显示此帮助主题。
进步	监视 WSDL 文件生成的进度。

部件文件名称对话框

字段/按钮	描述
名称	显示选定的 «WSDL» 原型部件的名称。
前缀	如果已指定多个前缀，请单击下拉箭头并为 WSDL 命名空间选择适当的前缀。
文件名称	类型输入或浏览 (单击 ) 要生成 WSDL 文件的文件路径和名称。
确定	单击此按钮保存输入的数据并关闭“部件文件名称”对话框。
取消	单击此按钮放弃输入的数据并关闭“部件文件名称”对话框。
帮助	单击此按钮可显示此帮助主题。

注记

- 您还可以通过自动化接口生成 WSDL 文件

导入WSDL


如果您有 WSDL 1.1 Enterprise Architect外部的文件，如果您想逆向工程为UML类模型，您可以使用 WSDL导入功能将它们导入系统。

访问

浏览器窗口 | 单击根包包含导入的文件，然后：

功能区	开发>架构建模>导入WSDL
-----	----------------

导入一个 WSDL文件

选项	行动
包	显示要导入WSDL文件的根包的名称。
文件名	类型输入或浏览 (单击 ) 要导入的 WSDL 文件的名称和路径。
目标包	默认为正在导入的 WSDL 文件的名称，作为包的名称来表示导入的文件。如果您不想使用默认名称，请键入不同的名称。
导入	单击此按钮以启动 WSDL导入。 导入完成时会显示A消息；单击消息上的确定按钮将其关闭。
关	单击此按钮可关闭此对话框。
进步	监视 WSDL导入的进度。

注记

- Enterprise Architect不能导入引用存在于该文件之外的 WSDL 构造的 WSDL 文件；如果其他文件中有引用的结构，请将所有文件合并到一个文件中并导入该合并文件
- 可导入文件的示例：http://www.w3.org/TR/wsdl.html#_wsdl
- 不可导入文件的示例：http://www.w3.org/TR/wsdl.html#_style；尝试导入此文件会导致错误消息无法导入 Split Files

SoaML

面向服务的架构建模语言 (SoaML) 是一种使用 Unified Modeling Language (UML) 设计和建模 SOA 解决方案的标准方法。

本文来源于面向服务的架构建模语言 (SoaML) —— UML 配置文件和服务元模型 (UPMS) 规范 (OMG 文档 ad/2008-11-01) ; 第 25-26 页:

A 服务是通过定义明确的界面向他人提供价值, 并可供社区 (可能是一般公众) A。服务导致工作由他人提供。”

面向服务的架构(SOA) 是一种组织和理解 (表示) 组织、社区和系统的方式, 以最大限度地提高敏捷性、规模和互操作性。SOA 方法很简单——人员、组织和系统相互提供服务。这些服务让我们不用自己做, 甚至不知道怎么做就能完成某事——使我们更高效、更敏捷。服务还使我们能够将我们的能力提供给他人以换取一些价值——从而建立一个社区、流程或市场。SOA 范式在集成现有功能和创建和集成新功能方面同样适用。

“SOA ... 是一种架构范式, 用于定义人员、组织和系统如何提供和使用服务来实现结果模型... 提供了一种使用 Unified Modeling Language (UML) 来构建和建模 SOA 解决方案的标准方法。配置文件使用 UML 的内置扩展机制根据现有 UML 概念定义 SOA 概念。”

“..... 使用 SOA 的最大优势来自于将社区、流程或企业理解为一组相互关联的服务, 并..... 用支持服务的系统支持面向服务的企业。SoaML 使面向业务和面向系统的服务架构能够相互协作地支持企业使命。... SoaML 依靠模型驱动架构® (MDA®) 来帮助将业务和系统架构、企业设计映射到支持 SOA 的技术, 例如 Web 服务和 CORBA®。”

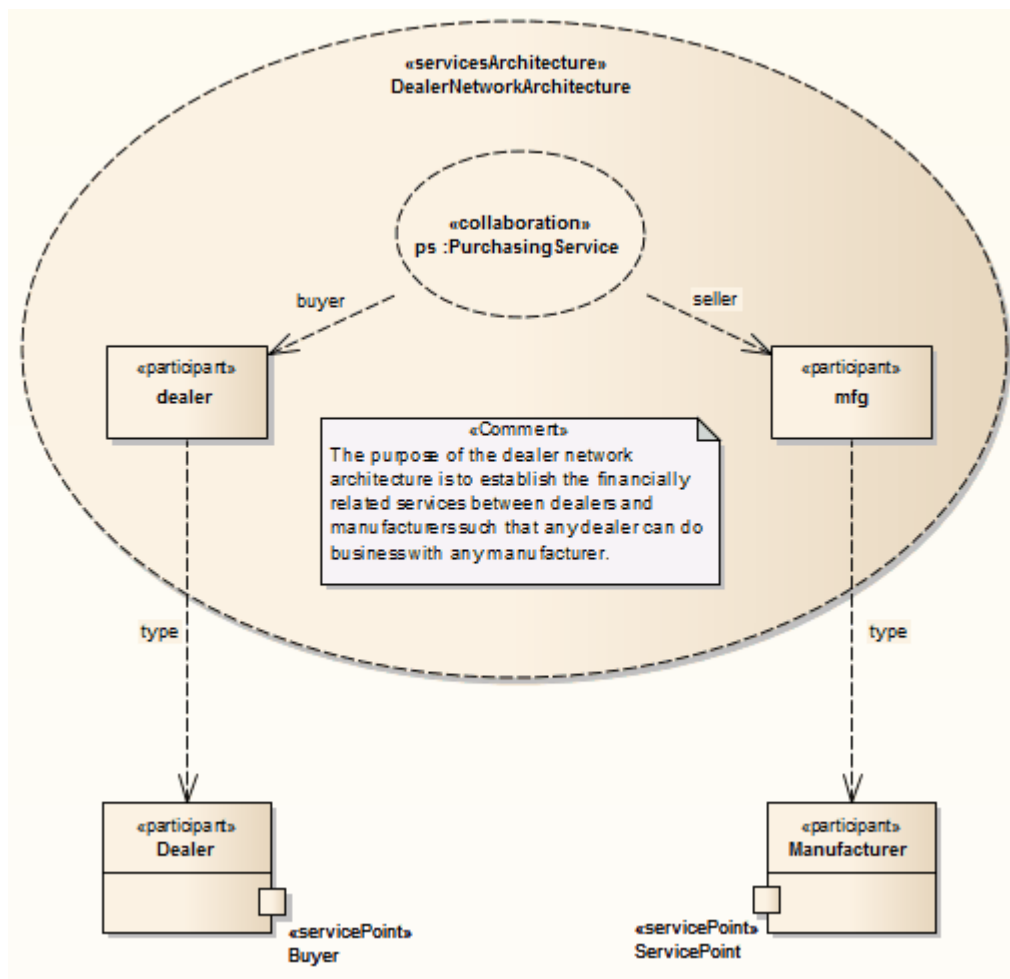
有关 SoaML 概念的更多信息, 请参阅 OMG 网站 SoaML 文档页面上的规范文档。”

Enterprise Architect 中的 SoaML

在 Enterprise Architect 中, 您可以通过使用与 Enterprise Architect 安装程序集成的模型 MDG 技术快速简单地对服务架构进行建模。SoaML 功能以以下形式提供:

- 两种 SoaML 图类型 - SoaML 部件图和 SoaML 序列图 - 通过 “新图表” 对话框访问
- 工具箱中的图表页面
- 工具箱快捷方式 “菜单和快速链接器中的 SoaML 元素和关系条目

示例 SoaML 图表



注记

- Enterprise Architect的企业版、统一版和终极版支持面向服务的架构建模语言（SoaML）

SoaML工具箱Pages

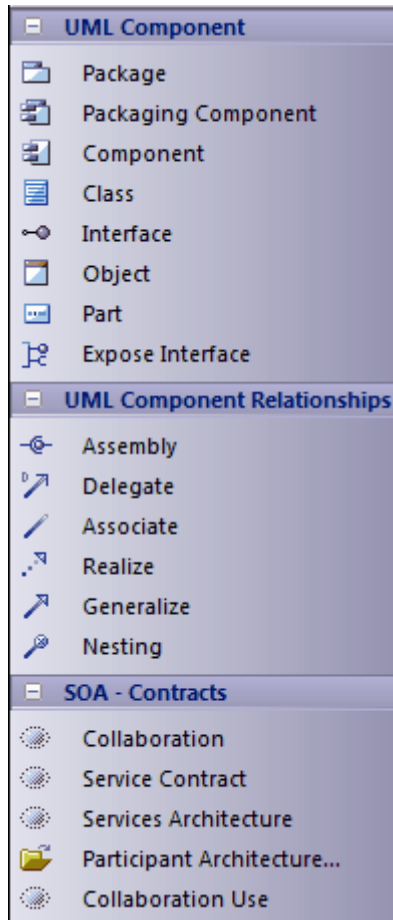
您可以使用工具箱的“SoaML”页面创建图表模型的元素和关系。这两种 SoaML 图类型中的每一种都有一组单独的页面，尽管这两组中的最后五个（SOA 特定）页面是相同的。

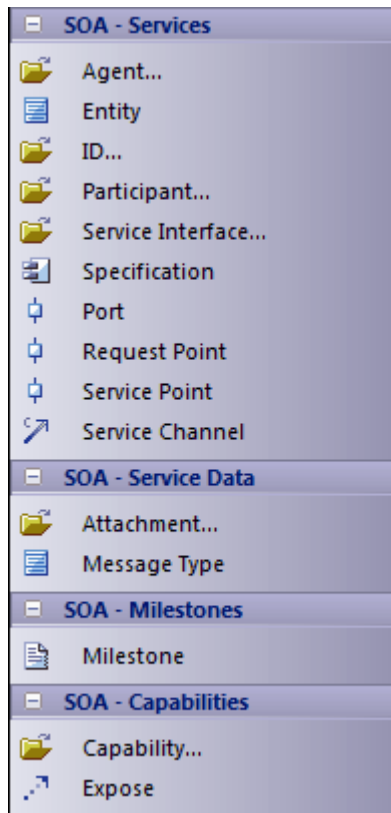
访问

功能区	设计 >  工具箱图表查找工具箱项”对话框并指定 部件工具工具”或“SoaML序列”
键盘快捷键	Ctrl+Shift+3 :  > 在 查找工具箱项”对话框中指定 SoaML部件”或“SoaML序列”
其它	图表标题栏 单击  图标以显示  工具箱图表 查找工具箱项”对话框中指定 SoaML部件”或 SoaML序列”

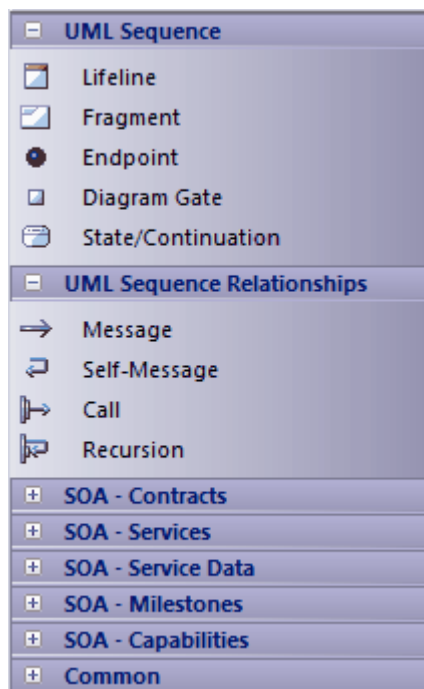
工具箱页面

图表部件工具箱





图表序列工具箱



SOMF 2.1

面向服务的建模框架 (SOMF) 是一种面向服务的开发生命周期方法论，提供了许多有助于成功的面向服务的生命周期管理和建模的建模实践和规则。这段文字来自于广泛的 Wikipedia 上关于面向服务的建模的条目：

'面向服务的建模框架 (作者) 已由 Michael Bell 提出，作为一种用于软件开发的整体和拟人化建模语言，它采用学科和通用语言为企业问题提供战术和战略解决方案。术语“整体语言”属于一种建模语言，可用于设计任何应用程序、业务和技术环境，无论是本地的还是分布式的。这种普遍性可能包括应用程序级和企业级解决方案的设计，包括 SOA 环境或云计算环境。另一方面，术语“拟人化”将 SOMF 语言与实现的直观性和使用的简单性联系起来。

'SOMF ...说明了确定服务开发计划“做什么”方面的主要元素。这些建模支柱将使从业者能够制定有效的项目计划并确定面向服务的计划的里程碑——无论是小型企业还是大型企业或技术企业。

Enterprise Architect中的 SOMF

在Enterprise Architect中，SOMF 2.1被实现为与Enterprise Architect安装程序集成的MDG 技术中的配置文件。SOMF 2.1功能以以下形式提供：

- 11 种 SOMF 图类型，可通过“新图表”对话框访问：
 - 概念
 - 分析
 - 云计算
 - 逻辑设计关系
 - 逻辑设计组合
 - 业务集成
 - 概念架构
 - 资产利用
 - 交易
 - 交易目录
 - 参考架构
- 工具箱中的 SOMF 页面工具箱Enterprise Architect为每个 SOMF 2 包括几个建模结构的工具箱页面。1 图表类型，通过工具箱搜索功能定位；这些提供了广泛的 SOMF 建模功能
- 工具箱快捷菜单和快速链接器中的工具箱元素和关系条目

全国信息交换NIEM建模2.1

国家信息交换NIEM建模提供了一个通用框架，用于定义如何在系统、政府机构和部门之间共享信息。NIEM UML配置文件帮助您：

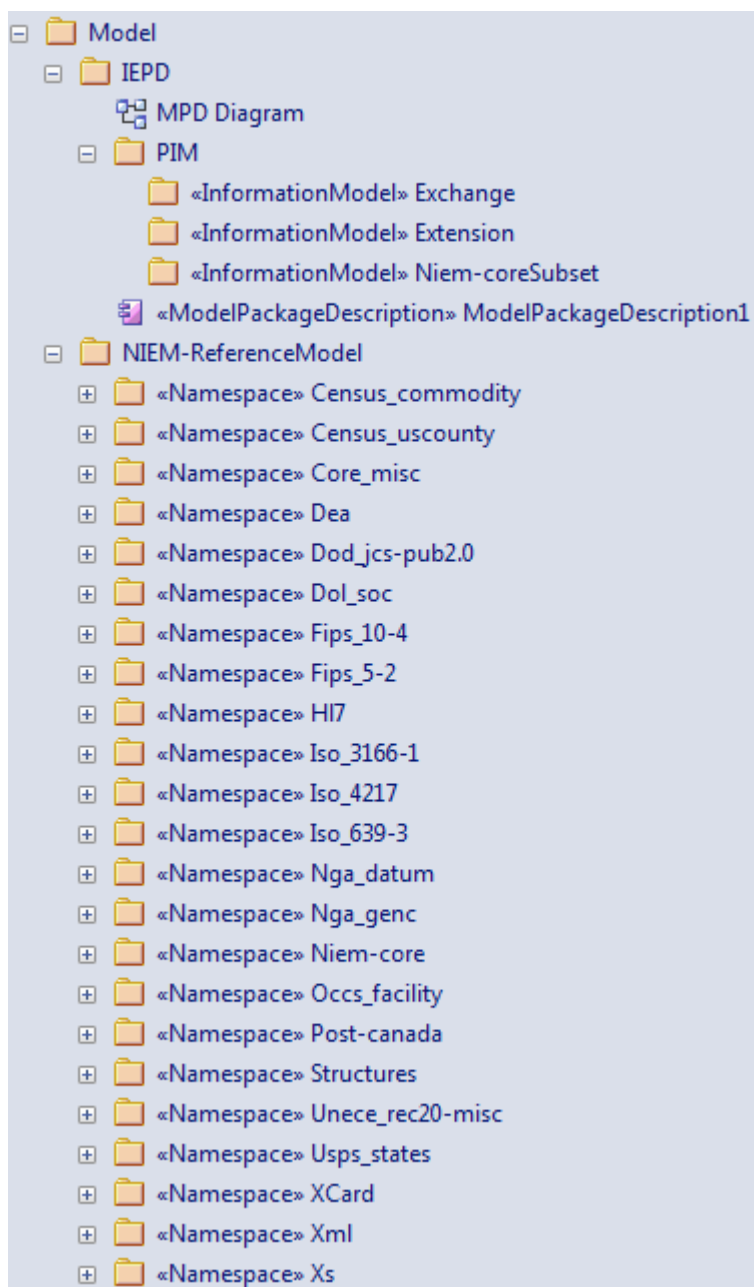
- 创建和开发基于 UML 的信息交换包文档 (IEPD) 模型，或者：
 - 从Enterprise Architect模式生成 IEPD 以生成所有必要的交换文件，静态工件、元数据和目录文件，或
 - 使用Schema 编辑器自动生成您自己的NIEM子集命名空间检测相互依赖关系，并使用生成的子集模式来构建您自己的 IEPD
- 使用NIEM工具箱页面创建 PIM、PSM 和模型包描述(MPD) 图
- NIEM导入参考架构您的模型
- 从您的模型生成NIEM架构

从模式创建NIEM IEPD模型

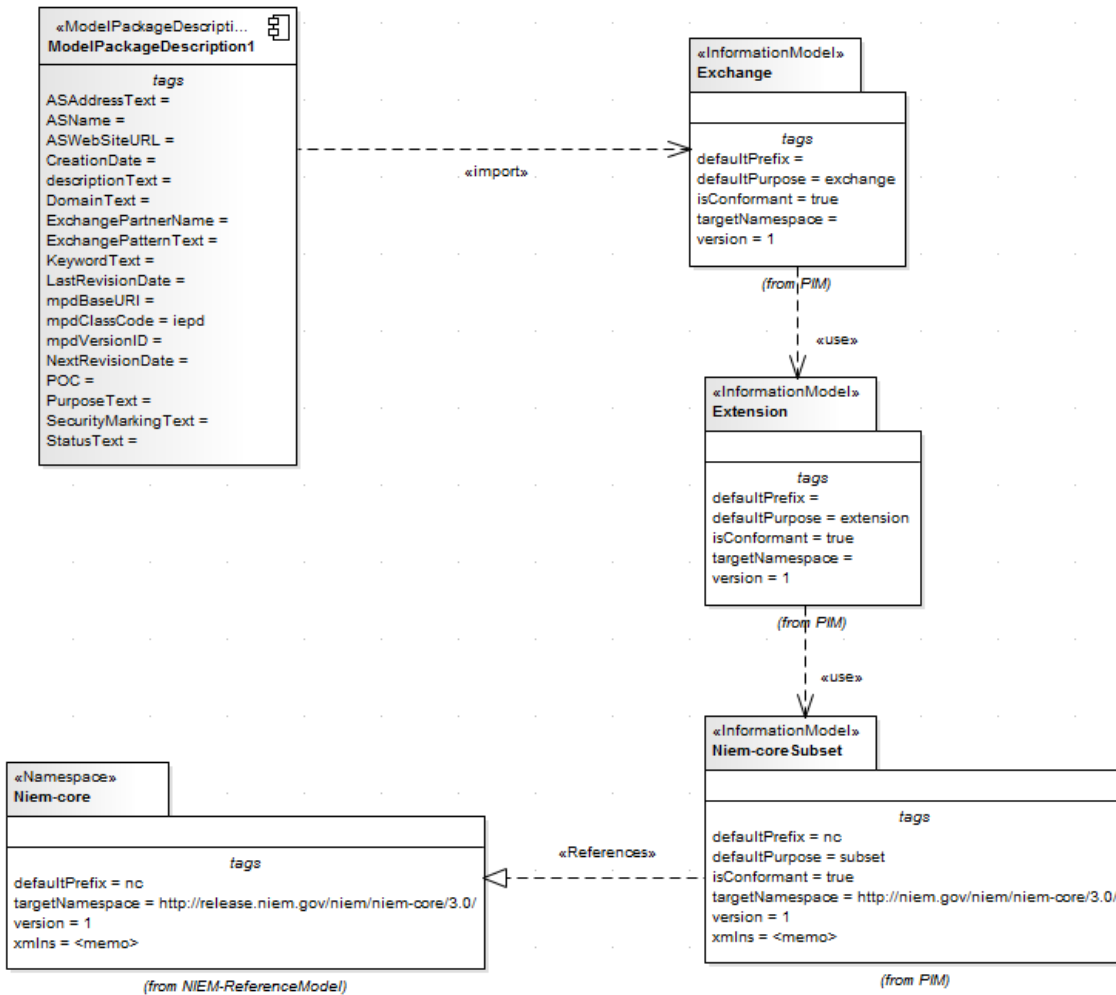
NIEM UML 配置模式配置文件提供了用于构建 IEPD模型的模型。您可以在您的NIEM项目中应用此模型，使用模型向导模式。

1. 在首页 从模式创建”选项卡 (模型向导) 中，在 “<<perspective>>”字段中选择 信息交换> NIEM ”。
2. 在 “NIEM 3 和 4”列表中，滚动浏览技术并单击所需的 “NIEM *nn*参考模型”，然后 Ctrl+单击相应的 “NIEM *n* IEPD Starter模型”。
3. 单击创建模型按钮。

系统生成包含IEPD包 (本身包含PIM包) 和NIEM ReferenceModel包的新模型。参考模型可能需要一些时间才能下载。



IEPD包包含一个顶层模型包描述(MPD)图(如图所示),其中包含MPD部件以及与其相关的所有命名空间和文件。



PIM包由 IEPD 的所有名称空间和子名称空间组成。MPD 图中显示了命名空间和 MPD 部件之间的关系。MPD 部件必须导入至少一个命名空间才能成功生成 NIEM 模式。

NIEM ReferenceModel 包包括所选 NIEM 版本的所有 NIEM 参考模式模型。

NIEM 图表

您还可以从 NIEM 图表集和相应的 NIEM 图表页面创建所有适当的工具箱。这些图表分为三种类型：

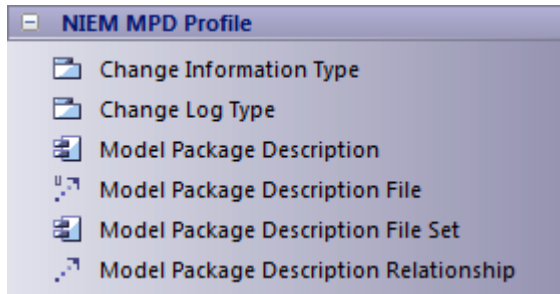
- NIEM 平台独立模型 (PIM) 图
- NIEM 平台特定模型 (PSM) 图
- NIEM 模型包描述 (MPD) 图

用于开发这些图表的模板可通过 “新建图表” 对话框获得。

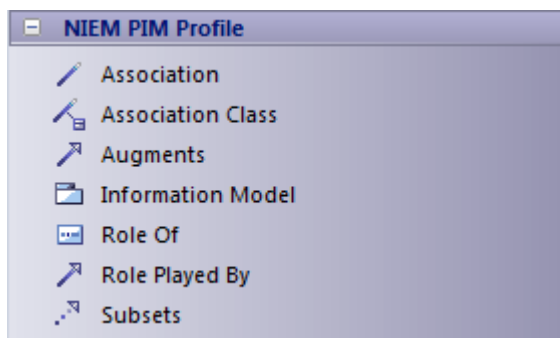
NIEM 工具箱 Pages

在图表工具箱中，每种图表类型都有自己的元素和连接图表工具箱。NIEM UML 配置文件还提供了所有三种图表类型共有的元素和连接器页面。

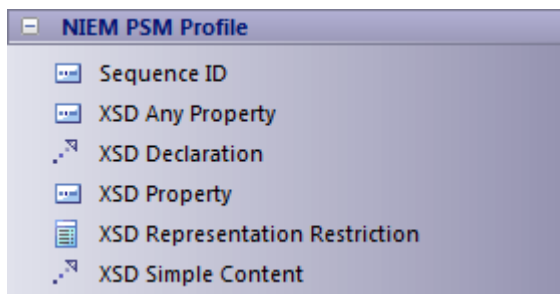
NIEM 模型包描述 (MPD) 配置文件工具箱



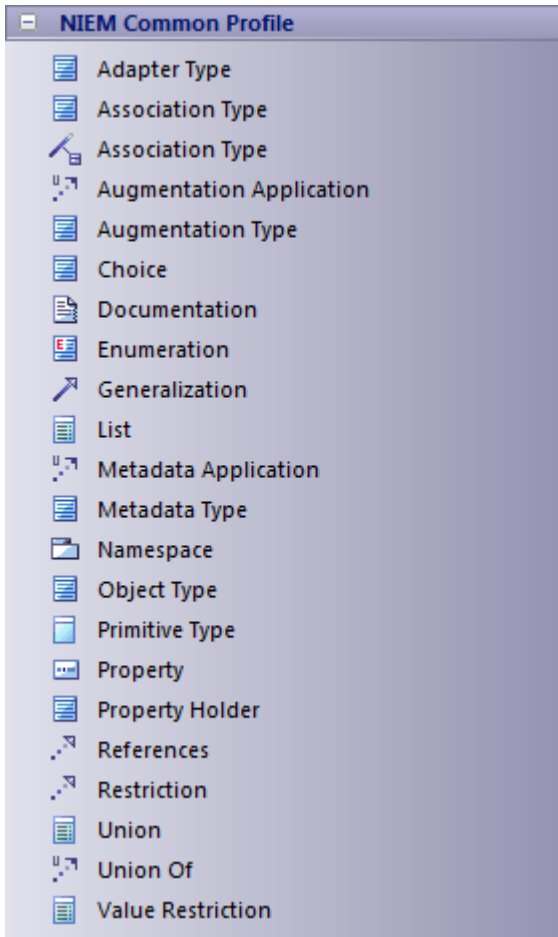
NIEM平台独立模型工具箱



NIEM平台特定工具箱



NIEM公共配置文件工具箱



导入NIEM参考架构

节	行动
1	在浏览器窗口中，右键单击目标包并选择 特定 NIEM 2.1 导入NIEM 2.1架构的菜单选项。
2	在 导入XML Schema “对话框的 目录”字段中，输入或浏览包含要导入的模式的目录，然后选择要导入的 .xsd 模式文件。
3	在 导入XSD元素/属性为：“下选择 “UML属性”单选按钮。
4	点击导入按钮。架构的NIEM模型将被导入到选定的包中。

生成NIEM子集命名空间

您可以使用Enterprise Architect Schema 编辑器生成子集命名空间。这要求NIEM参考模型在模型中可用，因为它是 IEPD模型模式的一部分。

节	行动

1	选择 开发>架构建模> Schema 编辑器>打开Schema 编辑器”功能区选项。
2	单击 配置文件”字段右侧的新建按钮。 将显示 新信息”对话框。
3	在 名称”字段中，键入子集的名称，在 命名空间”字段中，键入名称空间的 http 地址。
4	在 架构集”字段中单击下拉箭头并选择 国家信息交换模型 (NIEM) ”选项。
5	在 保存在”面板中，选择 模型”工件按钮。
6	点击  图标，使用导航员在IEPD中选择namespace/information模型包 PIM，在其下生成子集。
7	在浏览器窗口中，打开NIEM ReferenceModel包NIEM-core。 将此包中的活动拖到Schema 编辑器的左侧列中。 这个元素的属性列在Schema 编辑器的中间一列。
8	单击您需要的每个属性的复选框 - 例如， ActivityName 和 ActivityDateRepresentation。 相应的类/ NIEM object类型被添加到左侧列，而右侧列将它们显示为子项。
9	单击更新按钮以保存子集配置文件。 状态的状态显示在左侧列中的项目名称和列底部的面板中。
10	单击生成按钮。 将显示 架构导出”对话框。
11	在 技术”面板中选中要生成的项目的复选框。必须选择 ‘NIEM模型子集’。
12	单击生成按钮，然后在 查找包”对话框中，选择要在其中生成子集的名称空间/信息模型。
13	单击确定按钮，然后单击第二个确定按钮。子集模型已生成。

NIEM架构一代

一旦您的NIEM IEPD模型及其扩展信息模型、交换信息模型和子集信息模型完成，您就可以从中生成模式。

节	行动
1	右键单击导入 Exchange模型的 MPD部件，然后选择 特定 NIEM 2. 1 生成NIEM 2. 1架构的选项。 将显示 生成NIEM MPD 模式”对话框。
2	在 目录”字段中，键入或浏览要生成架构的目录路径。
3	在 ‘NIEM版本”字段中，单击下拉箭头并选择要为其生成架构的NIEM版本。 将生成的常见工件 (C、Metadata) 是在 MPD 相关工作”面板中列出的静态 MPD 工件和路径，每个都有其对应的路径。 命名空间架构”面板显示将为信息模型生成的架构文件。

4	点击生成按钮。 生成成功完成后，单击视图架构按钮以访问目录文件。
---	-------------------------------------

全国信息交换NIEM建模

国家信息交换模型 (NIEM) 提供了一个通用框架，用于定义如何在系统、政府机构和组织之间共享信息。Enterprise Architect的NIEM UML配置文件帮助您：

- 创建和开发基于 UML 的信息交换包文档 (IEPD) 模型，通过提供启动模型、模型和许多工具箱页面来创建 IEPD 模型和模式模型
- 从您的 IEPD模型生成完成IEPD
- 从您的信息模型中生成符合NIEM的模式
- NIEM导入参考架构您的模型
- 创建NIEM子集命名空间，由NIEM参考架构的元素组成
- 使用NIEM工具箱页面创建 PIM、PSM 和模型包描述(MPD) 图

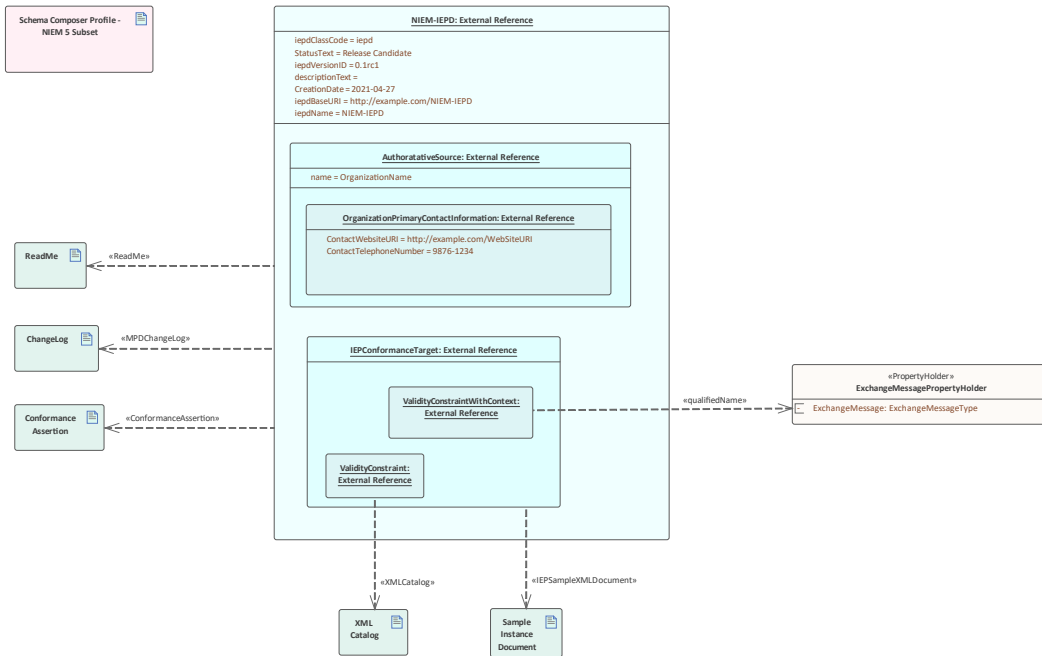
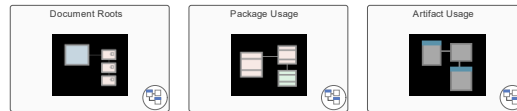
此图显示了NIEM 5.0 starter模型，这是Enterprise Architect中作为NIEM的一部分提供的模式。（请参阅创建NIEM IEPD帮助主题。）

IEPD Overview

This IEPD Overview diagram provides a quick overview of what is contained in the NIEM Starter Model.

In the top-left corner is a Schema Composer profile artifact. You can double-click on this artifact to open the Schema Composer, pre-configured to create a NIEM subset schema.

The other diagrams in the Starter Model, can be opened using the buttons to the right. Each of these diagrams focus in on just one aspect of the IEPD model, giving you room to add additional items without causing the diagram to become cluttered.




NIEM的UML配置文件


Enterprise Architect与NIEM的UML配置文件（支持NIEM NIEM和3）集成，以及许多模型模式，以帮助您开始为您的NIEM项目建模。


该配置文件定义了一组用于构建NIEM模型的构造型。它还定义了三种不同的图表类型：模型包描述（MPD）图表、平台独立模型（PIM）图表和平台特定模型（PSM）图表。这些图表类型中的每一个都有相应的图表工具箱页面，您可以从中选择要添加到模型中的项目，方法是将它们拖放到图表上。

访问

使用此处工具箱的任何方法来显示图表，然后单击使用以显示  工具箱“对话框并指定“NIEM 3.0 MPD”（或“PIM”或“PSM”）。

当您打开该类型的图表时，对应于特定图表类型的图表工具箱就会变为活动状态。但是，您也可以随时使用此方法访问任何图表工具箱：

- 在  工具箱图表 查找工具箱项”对话框并指定“<profile> <toolbox>”
要将工具箱重置为当前图表的默认类型，只需关闭然后重新打开图表即可。

功能区	设计>图表>工具箱
键盘快捷键	Ctrl+Shift+3
其它	单击图形图表的  图标以显示图表工具箱

图表工具箱

NIEM图表提供对特定类型图表中常用元素和连接器的快速访问。

图表工具箱分为多个单独的页面：模型模式、关系、文件用途类型和架构文档用途。PIM和PSM图共享一个共同的工具箱页面，并且每个都有自己特定的工具箱页面。

公共工具箱项

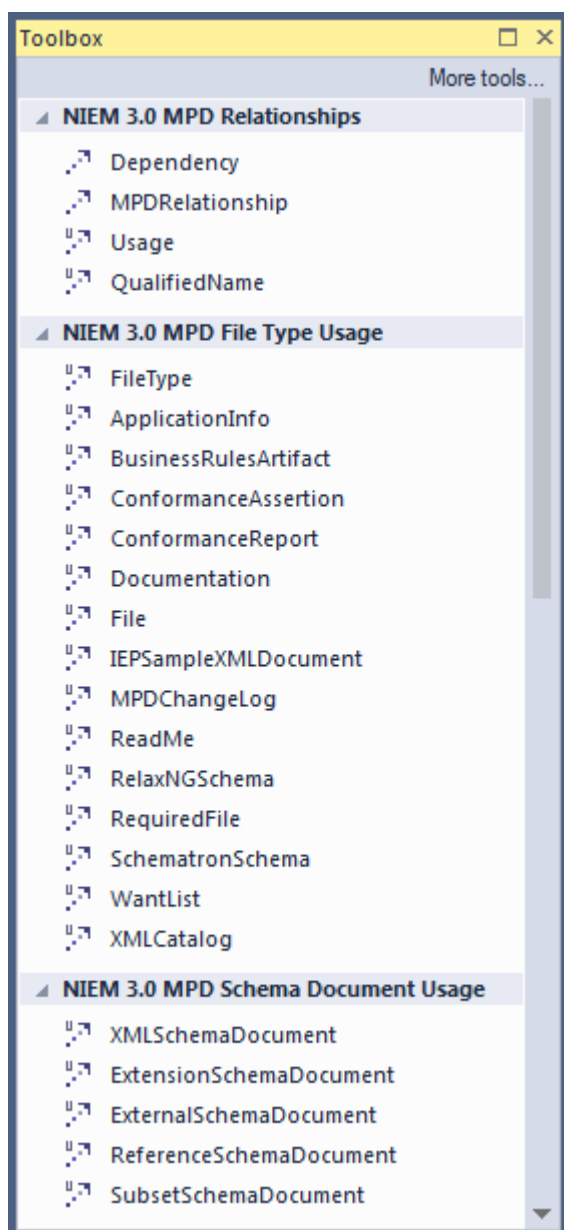
NIEM公共配置文件由在NIEM PIM配置文件和NIEM PSM配置文件中使用的构造型组成。

图标	描述
适配器类型	NIEM适配器类型是A NIEM object类型，它调整外部组件以在NIEM中使用。
关联类型	NIEM关联类型A对象之间建立关系，以及该关系的属性。
关联类型	NIEM关联类型A对象之间建立关系，以及该关系的属性。
增强类型	NIEM扩充类型是A复杂类型，它提供可重用的数据块，可以添加到object类型或关联类型。

选择	A选择类将一组其值互斥的属性分组。
文档	文档注释是拥有它的元素A数据定义。
概括	A UML概括
列表	List 是A DataType，其值由另一个 DataType 的值的有限长度（可能为空）序列组成，该 DataType 是 List 的项类型。
本地词汇	本地词汇定义了一组特定领域的术语或缩写，然后可以在NIEM名称和定义中使用。
本地术语	LocalTerm 原型定义了在本地的 LocalVocabulary 中使用的特定领域的单词、短语、首字母缩写词或其他string。
元数据应用	«MetadataApplication» 原型适用于 «MetadataType»类与另一个 «MetadataType»类或属性之间的用途。它表示对NIEM «MetadataType» 的约束，将NIEM «MetadataType» 的应用程序限制为特定的模式类型或模式元素。
元数据类型	NIEM元数据类型描述有关数据A数据，即不描述对象及其关系但描述数据本身的信息。
命名空间	命名空间包表示由目标命名空间URI标识A NIEM命名空间。
NIEM类型	A是表示NIEM复杂类型（即可以具有属性结构的类型）的特定语义类型之一的类。NIEMType 是抽象的。
ObjectType	NIEM object类型代表某种object：具有自己生命周期且存在A事物。
原始类型	NIEM原始类型库定义了一组预定义的UML原始类型，用于 NIEM-UML 模型。为了确保在 PIM 级别使用的类型系统与NIEM兼容模式生成的完整性和一致性，该库中的原始类型基于 XML 模式原始类型。
属性	
财产持有人	PropertyHolder 是A持有全局属性的类，它不是任何特定NIEM类型的主题。这种类型的属性声明定义了属性的object类型，属性限制它的使用特定类型的主题。
参考	引用构造型适用于属性、类或包之间的实现。它允许通过属性另一个类中的属性来定义一个类中的属性。
表示	NIEM表示模式，允许一个类型包含一个表示元素，并且该元素类型的各种表示都在该表示元素的替换组中。
限制	Restriction Realization 表示两种类型定义之间的关系：第一种是通过第二种定义A限制导出的。
联盟	Union 是A DataType，其值空间是一个或多个其他 DataTypes 的并集，这些 DataTypes 是 Union 的成员类型。
联盟	UnionOf 构造型应用于用途依赖关系，其客户端必须是 Union DataType，而

	提供者必须是表示合法联合成员类型的 DataType 。A 依赖项指定供应商 DataType 是客户端 Union 的成员类型。
价值限制	

NIEM 3.0 MPD 工具箱



MPD 工具箱项

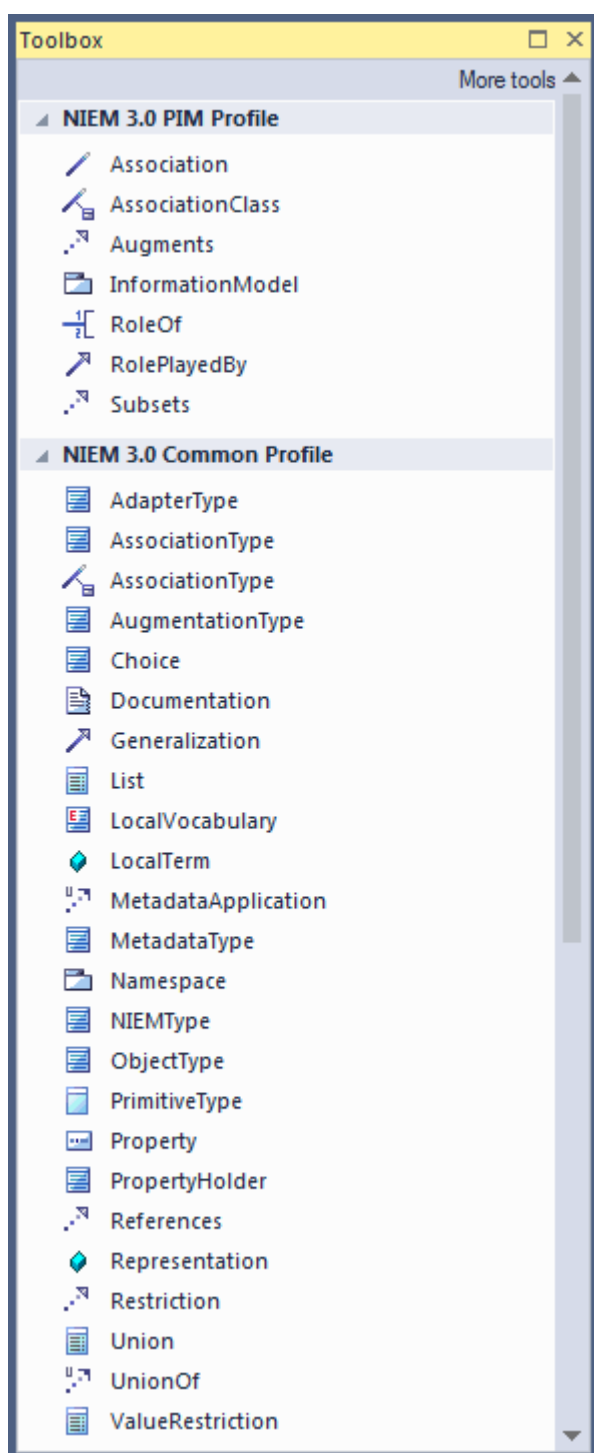
模型包描述配置文件包含用于模型 NIEM MPD 的构造型和工件。

图标	描述

关系	
依赖	UML依赖关系A
MPD关系	关系原型适用于表示 MPD 之间或 MPD 与另一个资源（例如NIEM规范；如符合的情况）之间的关系的依赖项。
用途	A UML使用关系
合格名称	<p><<qualifiedName>>用途连接器用于指定 IEP 的文档元素。</p> <p>在UML中识别 IEP 的文档元素：</p> <ul style="list-style-type: none"> • 将实例 IEPConformanceTargetType 添加到 ModelPackageDescription 的 ModelPackageDeConformanceTarget 槽的工件实例 • 将 QualifiedNamesType 实例添加到新 IEPConformanceTargetType 实例的 ValidityConstraintWithContext 槽 • 添加具有应用构造型<<qualifiedName>>的用途，其中客户端是新的用途属性实例，供应商是表示 XSD元素的
文件类型用途	
文件类型	<<用于描述用途文件类型的连接器是一种数据工件。它也是许多其他 <<FileType用途>>连接器派生的基本类型。
应用信息	<<ApplicationInfo>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件工具，用于软件工具使用（例如，输入和输出）。
商业规则神器	<<BusinessRulesArtifact>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件，用于包含业务规则和约束的交换内容。
一致性断言	<<ConformanceAssertion>> 连接器扩展了 <<FileType>> 使用连接器。它指定一个工件用于表示一个 NI IEPD 符合NIEM的声明。
一致性报告	<<ConformanceReport>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个 MPD 软件和其他由工件感知工具自动生成或手动准备的软件，用于检查 NIEM符合性质量并呈现详细的结果报告。
文档	<<Documentation>> 连接器扩展了 <<FileType>> 使用连接器。它指定 MPD 用于工件文件的一种形式。
文件	<<File>> 连接器扩展了 <<FileType>> 使用连接器。它是用于指定工件的通用电子文件；存储在计算机系统上的文件。
IEPSampleXML 文档	<<IEPSampleXMLDocument>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定示例 MPD 文档或工件实例。
MPDChangeLog	<<MPDChangeLog>> 连接器扩展了 <<FileType>> 使用连接器。它指定用于包含 MP 更改记录的工件。
自述文件	<<ReadMe>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定工件读取。
松弛NGSchema	<<RelaxNG>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定一个 RelaxNG 模式。

所需文件	<<RequiredFile>> 连接器扩展了 <<FileType>> 使用连接器。它是用于与 MPD 文件工件的，而不应指定另一个工件于
SchematronSchema	<<SchematronSchema>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定 Schematron 模式文档。
想要列表	<<WantList>> 连接器扩展了 <<FileType>> 使用连接器。它指定一个 MPD 架构子集，用于表示一个 NIEM 架构子集，并用作 NIEM 工件的导入或导出
XML 目录	<<XMLCatalog>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件用于 OASIS XML 目录。
架构文档用途	
XMLSchema 文档	<<XMLSchemaDocument>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件，该 MPD 不是 XML 模式文档，XSD 不一定是 NIEM 子集，扩展或引用模式)。
扩展架构文档	<<ExtensionSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它用于指定一个工件扩展架构文档，该文档是一个 NIEM 扩展架构文档。
外部架构文档	<<ExternalSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定一个工件用于外部的一个架构文档，用于 NIEM
参考架构文档	<<ReferenceSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定了一个工件，用于从参考架构文档 (参考架构文档、发布、域更新或核心更新)。
子集架构文档	<<SubsetSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定了一个工件，它是一个子集模式文档。

NIEM 3.0 PIM 工具箱



PIM工具箱项

NIEM PIM配置文件包含用于NIEM PIM 但不用于NIEM PSM 的构造型。

图标	描述
关联	UML关联A
关联类	A UML关联类。

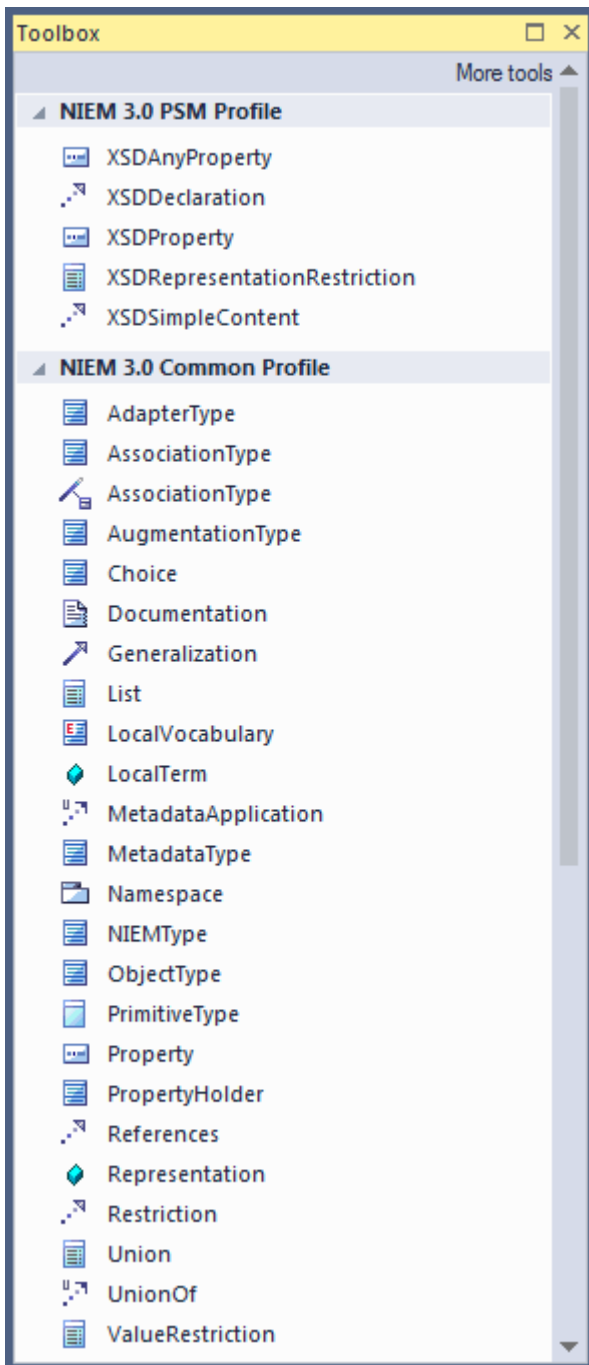
增强	A原型的实现连接器，用于指定一个类（供应商）增强另一个类（客户端）。
信息模型	InformationModel 是一个原型包，它为在NIEM消息中交换的信息结构提供了与平台无关的视角。它表示NIEM命名空间，但也可以指定默认用途，例如子集、交换或扩展。
的角色	RoleOf 构造型应用于 AssociationEnd 以指定关联属性的角色类型。
角色扮演者	A原型概括实例扮演的角色连接器指定一般类的角色是特殊类的类型。
子集	Subsets 连接器是一个原型实现，它指定子集客户端（派生元素）和它的参考供应商（基本元素）之间的子集关系。

PSM工具箱项

NIEM PSM配置文件包含在NIEM PSM 中使用的构造型。这些构造型不需要与NIEM PIM 一起使用，但它们可以用于提供额外的特定于平台的标记。

图标	描述
XSDAnyProperty	属性构造型表示其类型不受限制的属性，该类型在XML Schema中实现为xs:any 粒子。
XSD声明	XSDDeclaration 原型是公共引用原型的特化。
XSD 属性	属性表示NIEM属性，它在XML Schema中实现为属性声明和使用或元素声明和粒子
XSDRepresentationRestriction	XSDRepresentationRestriction 指定对基本 DataType 的值在 XML 模式中的表示形式的限制。
XSDS简单内容	«XSDSimpleContent» 构造型表示两个类型定义之间的关系：第一个是具有简单内容的复杂类型定义，第二个是简单类型。

NIEM 3.0 PSM工具箱



下载NIEM参考模型

NIEM 5参考模型是NIEM 5 Release包XSD文件内容的UML表示。

它包含代表 NIEM 核心的包，以及NIEM 5 版本中包含的各种域模式、它们的相关代码列表和其他相关包。NIEM 5 参考模型可从Sparx Systems可重用资产服务器下载到您的Enterprise Architect项目中。

早期版本的NIEM参考模型也可从Sparx Systems可重用资产服务器下载。

访问

使用此处列出的任何方法显示模型向导（首页 从模式创建”选项卡）。

在模型向导中，选择 Information Exchange > NIEM蓝图，然后选择 “NIEM 3, 4 and NI5”。

根据需要选择参考模型、MPD Types 和 Starter模型。

功能区	设计>包>模型向导
上下文菜单	右键单击包 添加模型使用向导
键盘快捷键	Ctrl+Shift+M
其它	浏览器窗口标题栏菜单 模型模式

创建NIEM IEPD

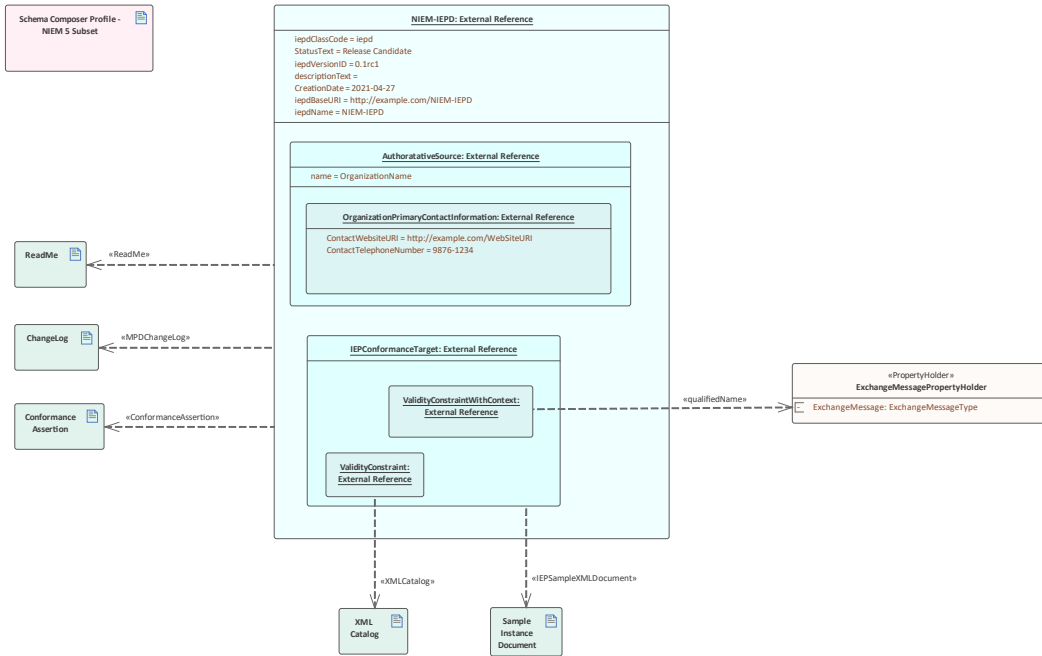
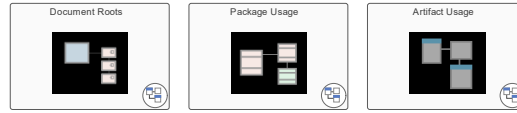
Enterprise Architect的NIEM配置文件提供了一个基本的 IEPD模型，作为您可以构建自己的 IEPD模型的起点。您可以使用“模型向导”选项卡将模式启动您的首页。

IEPD Overview

This IEPD Overview diagram provides a quick overview of what is contained in the NIEM Starter Model.

In the top-left corner is a Schema Composer profile artifact. You can double-click on this artifact to open the Schema Composer, pre-configured to create a NIEM subset schema.

The other diagrams in the Starter Model, can be opened using the buttons to the right. Each of these diagrams focus in on just one aspect of the IEPD model, giving you room to add additional items without causing the diagram to become cluttered.



模式Starter模型，可从模型向导中获得。

本主题概述了在Enterprise Architect中创建新的NIEM IEPD模型以及从该模型生成 IEPD 所需的步骤。

创建NIEM IEPD模型并生成NIEM IEPD

脚步	描述
创建一个新的Enterprise Architect项目	<p>启动Enterprise Architect并创建一个新项目。</p> <p>单击首页'从模式创建'选项卡（模型向导）。</p> <p>点击<perspective name>按钮并选择'Information Exchange NIEM ' 并扩展了 ' NIEM 3、4 和 5'蓝图。</p> <p>您的NIEM项目必须包含NIEM IEPD 类型和至少一个NIEM参考模型。</p> <p>选择 'NIEM 5.0参考模型'以及 'NIEM 5 IEPD Types'。</p> <p>单击模型(s) 按钮下载所选模型并将其导入到您的项目中。</p> <p>模型向导中还提供基本NIEM IEPD 的模型模式。这旨在作为您的NIEM项目的起点。</p> <p>或者，选择 'NIEM 5 IEPD模型'并单击模型(s) 按钮。</p>

<p>创建 IEPD模型</p>	<p>如果您在上一步中选择不包含 IEPD模型，您现在可以创建自己的模型。在浏览器窗口中，创建一个新包或（视图节点）来保存您的 IEPD模型。在新包中，创建一个NIEM IEPD 图。</p> <p>您可以将NIEM IEPD Types包中可用类型的实例添加到您的图表（以及您的 IEPD模型）中，方法是将它们拖到您的 IEPD 图表上。</p> <p>使用浏览器窗口找到您需要的类类型，然后在将元素拖到图表上的位置时按 Ctrl。系统提示您选择一个动作；任何一个：</p> <ul style="list-style-type: none"> • 在图表上放置类的链接，或 • 创建并添加类的新实例规范 <p>对于 IEPD模型，您通常会使用物件实例。</p> <p>首先，创建 IEPD类的object实例。（你很快就会看到，你需要一个 IEPD 类的实例来驱动你的 IEPD 的生成。）</p>
<p>自定义您的模型</p>	<p>IEPD 类的实例携带识别 IEPD 的信息。它与各种其他模型工件之间的链接决定了在生成模式文件和目录文件时生成什么（以及生成到哪里）。</p> <p>无论您选择下载 IEPD Starter模型，还是通过从浏览器窗口拖动实例来创建您自己的 IEPD模型，您都必须设置适合您正在创建的模型的物件实例的属性值。这是通过设置 IEPD模型中使用的各种object实例的运行状态属性来实现的。</p> <p>有关为 IEPD 实例对象设置值的详细信息，请参阅帮助主题自定义您的帮助模型</p>
<p>创建您的数据模型</p>	<p>这是您模型将在您的信息交换消息中发送的数据。</p> <p>在NIEM中，这通常在具有 <<InformationModel>> 构造型的包内建模，表示模型中使用的不同名称空间。这些包通常包括一个 NIEM-core包，它是 NIEM-core参考模型包的子集，以及两个扩展包，它们扩展了 NIEM-core 中可用的内容，其中一个代表交换消息。</p> <p>您的项目可能还需要其他NIEM模式的子集，例如来自 Biometrics 或 EmergencyManagement 域的那些。</p> <p>有关创建数据模型的更多信息，请参阅帮助主题创建NIEM数据模型帮助使用 Schema 编辑器对NIEM进行子集化。</p>
<p>生成IEPD</p>	<p>在您从中生成 IEPD 之前，您的NIEM模型不必完成。</p> <p>生成 IEPD 可以被认为是一个迭代过程。在完成 IEPD 和定义一致性目标之前，您可能只生成命名空间模式。在定义信息模型之前，您可能会使用完整描述的 IEPD 实例和一致性目标来生成。您可以继续更新您的模型并生成您认为合适的 IEPD。</p> <p>要生成您的 IEPD，请在图表或浏览器窗口中选择 IEPD 实例规范。转到 特定“功能区并选择选项 技术> NIEM >生成NIEM架构”。</p> <p>生成NIEM IEPD Schemas 窗口打开。</p> <p>此窗口列出了模型中使用的命名空间模式，您可以选择生成其中的哪些。您还可以选择在生成中包含哪些NIEM基础架构模式。</p> <p>在此窗口中，您还可以设置生成输出文件的根目录。</p> <p>选择并指定输出文件夹后，单击 生成“按钮开始生成 IEPD。</p> <p>有关生成NIEM IEPD Schemas 窗口的详细信息，请参阅帮助主题 NIEM帮助 NIEM。</p>

注记

- 所有包含NIEM模型的项目都必须包含通过首页“从模式创建”选项卡（模型向导）下载的NIEM IEPD 类型包；IEPD 实例是NIEM模型的核心
<<InformationModel>>包和其他工件的实例和关系用于驱动IEPD生成；如果您的模型中没有 IEPD 实例，您将无法生成 IEPD
- 通常，您必须将至少一个NIEM参考模型导入到您的项目中；参考模型包含 NIEM 核心参考模式的UML表示，以及许多特定领域的参考模式，如果您打算使用Enterprise Architect的Schema 编辑器创建子模式，则必须在您的项目中可用

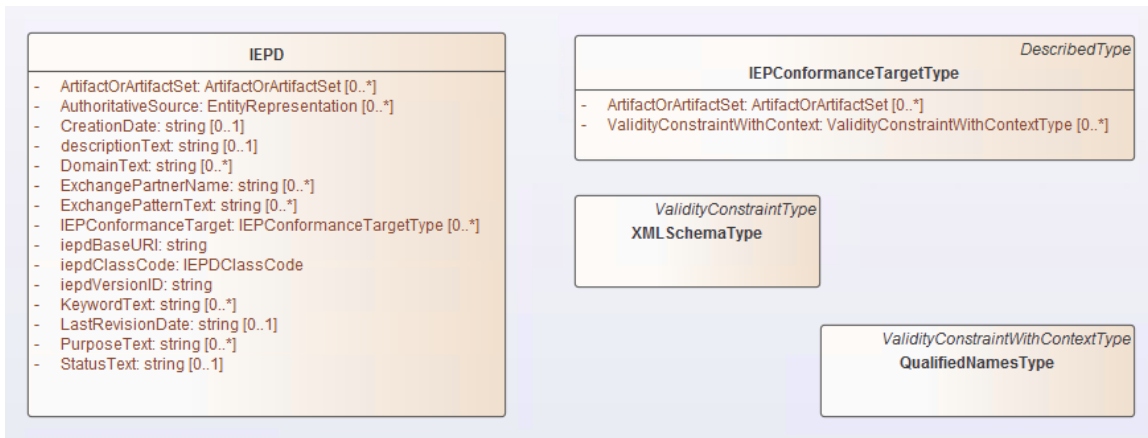
自定义您的 IEPD 模型

无论您选择下载 IEPD Starter 模型，还是通过从浏览器窗口拖动实例来创建自己的 IEPD 模型，您都必须设置适合您正在创建的模型的物件实例的属性值。这是通过设置 IEPD 模型中使用的各种 object 实例的运行状态属性来实现的。

NIEM IEPD 类型

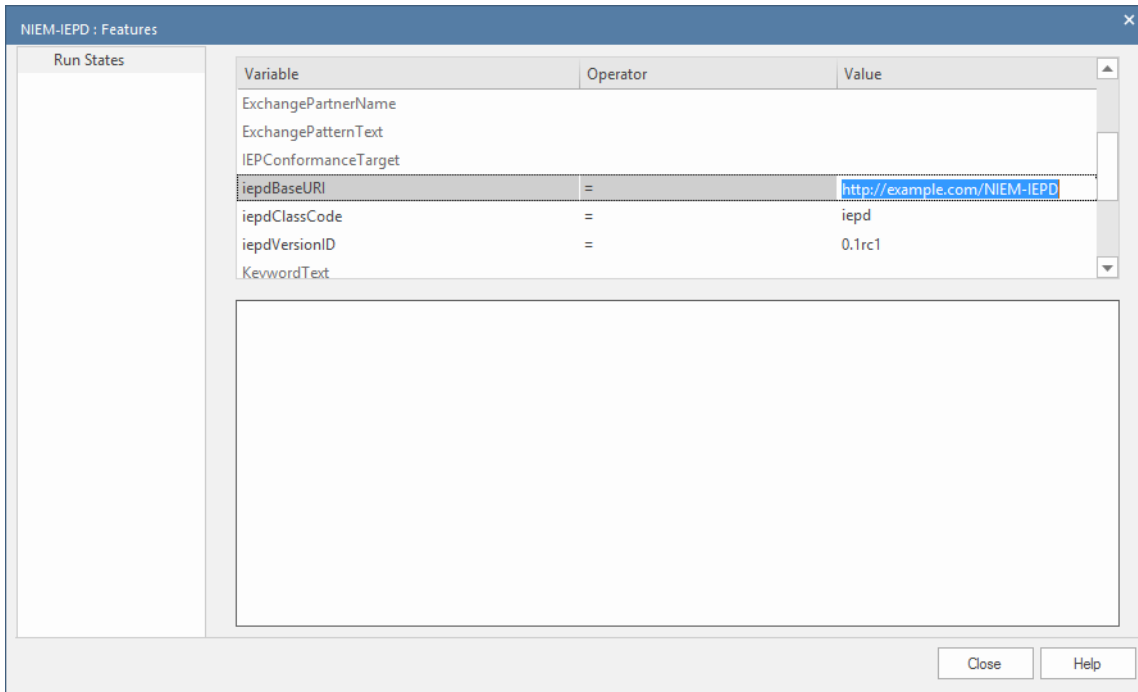
包 'NIEM IEPD Types' 包含类 'IEPD' 的定义，以及许多其他类。这些其他类被引用为 'IEPD' 类属性的分类器。在这个包中定义的各种类之间的关系可以在图表 'NIEM-UML IEPD Types' 上查看。

'IEPD' 类有许多简单 string 类型的属性，以及一些按在 'NIEM IEPD Types' 包中定义的类型分类的属性。



设置 IEPD 对象的运行状态

在运行 object 中为简单类型的属性设置运行状态值时，可以使用“设置运行状态”命令。这可以通过右键单击物件上的物件，然后选择“特征”来访问。设置运行状态...! (或按 Ctrl+Shift+R)。

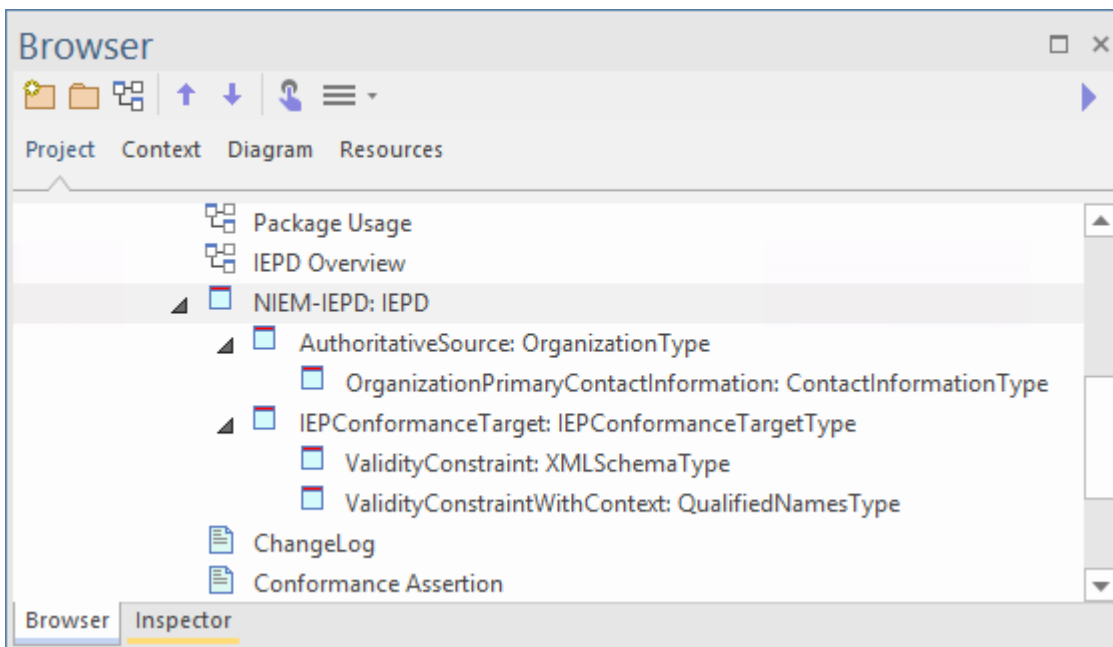


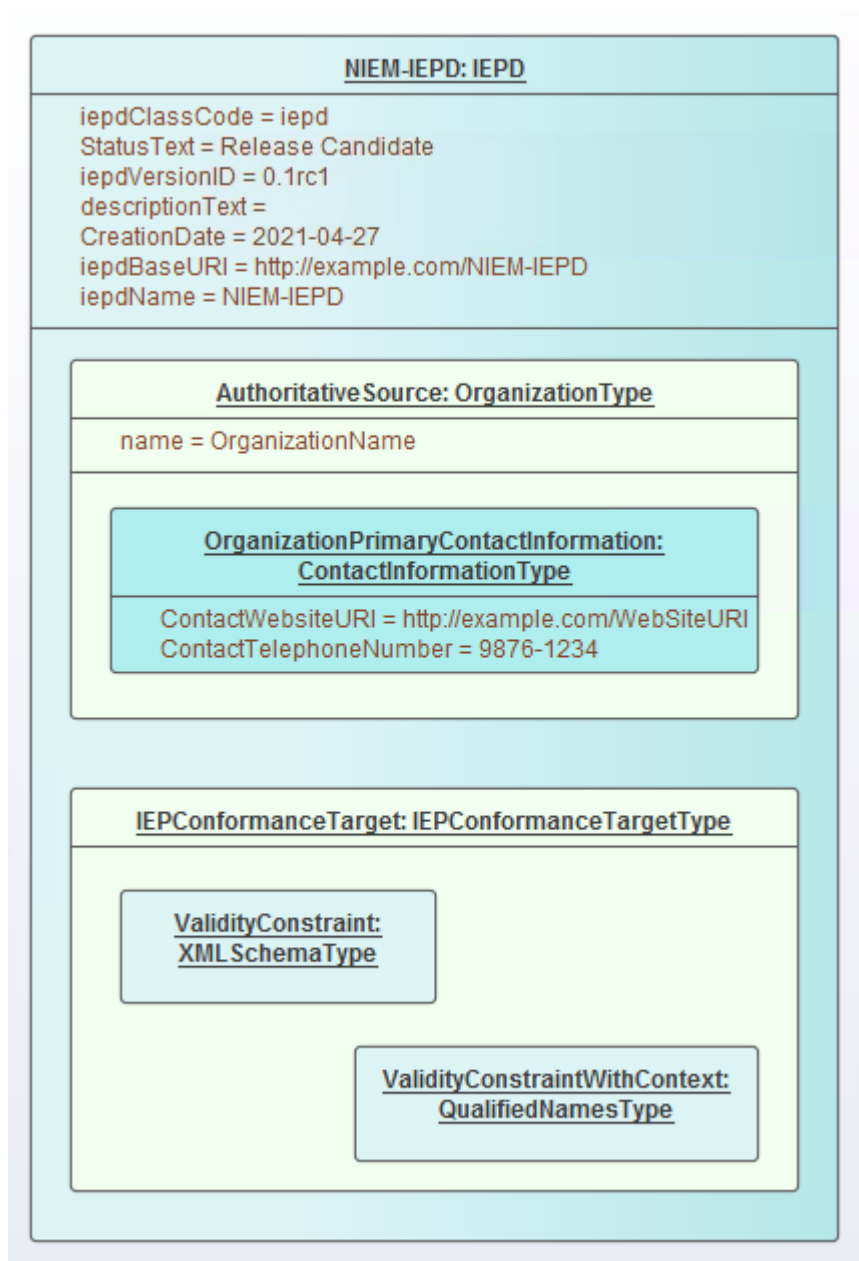
在属性引用其他类作为其类型的情况下，不能简单地输入运行状态值。

Enterprise Architect支持两种为这些属性指定值的方法，每种方法都需要创建被引用类的物件实例。

您属性为属性设置的属性对象关联两个嵌套的物件物件作为子物件并使用正在设置的属性的名称命名子物件。关联物件时，物件的名称并不重要，但角色名称必须与设置的属性名称相匹配。

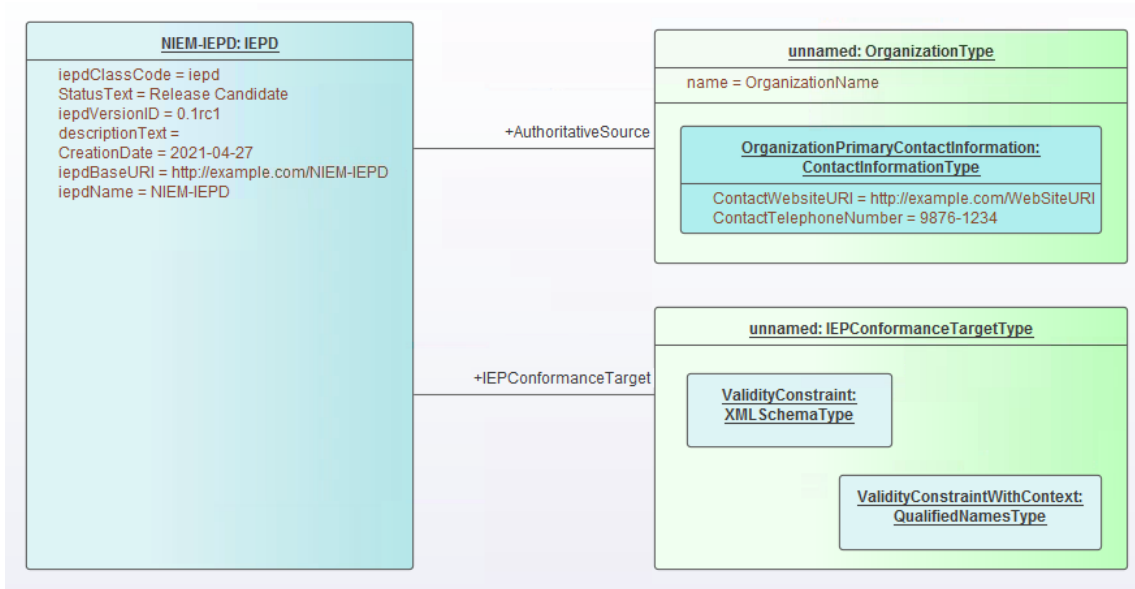
例如，您可以在 IEPD 中创建类型为物件的物件和嵌套的物件。在这种情况下，子物件必须命名为“IEPConformanceTarget”，以与类定义中该名称的属性相对应。通过检查浏览器窗口中显示的层次结构，确保子物件确实嵌套在父级中。





如果在关联上使用角色名称，则将“属性”物件创建为单独的（非嵌套）物件实例，然后创建从“所有者”到“object”的关联，最后为目标物件指定角色名称。例如，创建从模型包描述物件到关联物件实例的关联。打开关联的“属性”对话框并将目标的角色命名为“IEPConformanceTarget”，以对应于“IEPD”类中的属性名称。同样，在这种情况下，物件本身的名称并不重要，它甚至可以是匿名的，但角色名称必须与您正在设置其值的属性的名称匹配。

注册一个 IEPD 物件可能指定许多 IEPConformanceTargets。您必须为每个实例创建一个物件实例，并且每个实例都必须命名为“IEPConformanceTarget”。

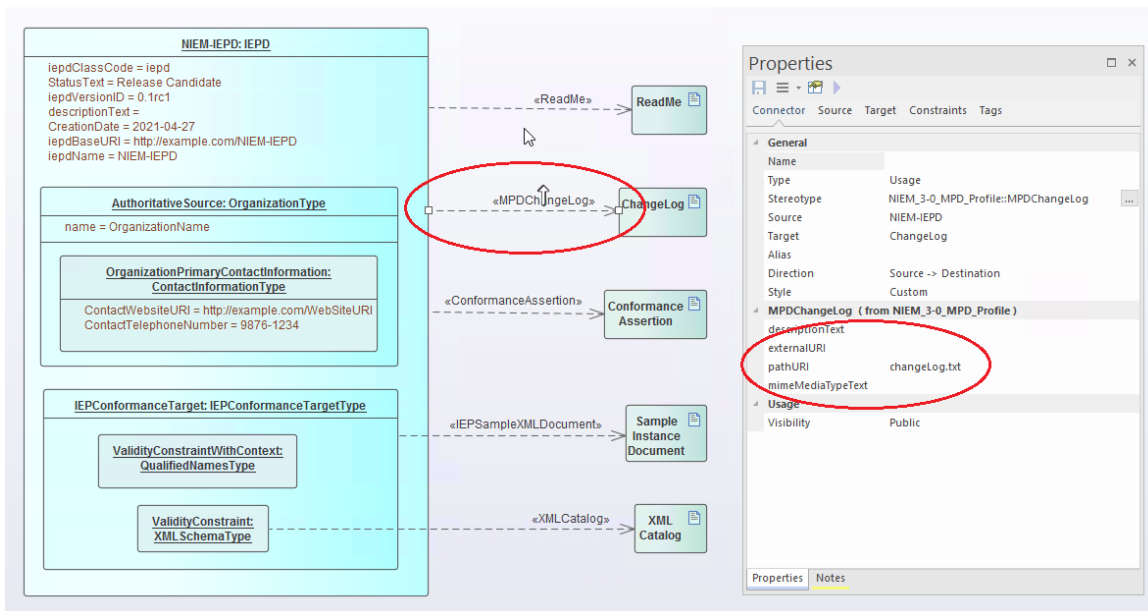


然后可以使用这些技术中的任何一种来设置属性物件中的属性。例如，要设置物件实例的类（可能是 `QualifiedType` 类型的实例）以及它的嵌套名称，为派生的 `ValidityNamestraintType` 属性和名称和角色关联。

建模文件用途

使用文件类型可以通过将使用文件类型与用途中所需的连接器添加到工件中来建模。

使用相关用途连接器的属性中指定的值，将各种文件名生成到 IEPD 的 XML 目录文件中。



NIEM IEPD 生成

生成 IEPD 可以被认为是一个迭代过程。您不必等到您的NIEM模型完成后才能从中生成 IEPD。

您的NIEM 'IEPD Overview' 图应包含 IEPD 的实例规范。IEPD 实例及其与一致性目标实例以及其他工件的关系是 IEPD 目录的表示。当您从您的模型生成 IEPD 时，Enterprise Architect将根据您的 IEPD模型中的项目生成目录文件和其他工件。它还将为您的模型引用的 <<InformationModel>> 包生成NIEM模式。结果将是一组文件输出到您为生成过程指定的目录中。

生成 IEPD 的步骤

节	行动
1	您的NIEM IEPD 图应包含 IEPD 的实例规范。 在图表或浏览器窗口中选择 IEPD 实例。
2	从 特定“功能区中，选择 技术> NIEM >生成NIEM架构”选项。 将显示 生成NIEM IEPD 模式”对话框。
3	在 目录”字段中，键入或浏览生成 IEPD 的目录路径。
4	“NIEM版本”字段默认为 5.0”。如果生成NIEM 3 或 4 IEPD，请将此字段设置为适当的值。 '工件' 面板列出了此模型工件静态目录中使用的工件和静态目录，每个都有其相对路径。 选择或取消选择这些项目旁边的复选框以生成或跳过这些项目。 命名空间架构”面板显示将为信息模型生成的架构文件。 选择或取消选择命名空间架构旁边的复选框以生成或跳过该架构。 选择一个架构命名空间架构以显示该架构的包详细信息。
5	点击生成按钮。 生成成功完成后，单击查看架构按钮打开视图，显示用于生成的输出目录的内容窗口如果当前的架构工件将被取消选择，则单击视图按钮打开一个编辑器以查看与所选架构命名空间包关联的架构。

注记

- 为包生成的模式文件的输出位置由将包与 IEPD 实例规范相关联的使用连接器上的 用途”标记值指定；默认值由Schema 编辑器在创建子集包时设置，但用户可以覆盖这些值

创建NIEM数据模型

NIEM背后的基本原则之一是重用通用参考词汇表——一组预定义的数据元素和定义，用于定义信息交换。为此，构建NIEM数据模型的核心任务之一是创建NIEM参考模式的子集。目标是通过重用已在NIEM参考模型中定义的类型和元素来模型尽可能多的数据交换。

A NIEM数据模型通常由多个应用了 <<InformationModel>> 原型的包组成。

通常，一个模型将有一个包表示 NIEM 核心子集模式，一些其他包表示特定域模式的子集，以及一个或多个包表示扩展模式。扩展模式包提供模型所需的那些元素，这些元素不能从NIEM参考模型中获得。通常，交换消息的根元素与更多的一般元素分离，并建模在专用于特定交换的扩展模式包中。

创建NIEM数据模型的步骤

节	细节
导入NIEM参考模型	<p>创建NIEM模型所涉及的许多活动都依赖于使用NIEM参考模型。如果您还没有这样做，请在继续之前将参考模型导入您的Enterprise Architect项目。</p> <p>有关详细信息，请参阅帮助主题下载帮助Download the NIEM Reference Model</p>
创建 NIEM-core参考包的子集	<p>在创建NIEM IEPD 时创建NIEM命名空间模式子集的原因有很多，但最重要的两个原因是：</p> <ul style="list-style-type: none"> 参考模式非常大；子集生成的模式文件要小得多，这反过来又会导致更快地验证模式 参考模式中的元素受到非常松散的约束；子集化过程允许建模者施加更严格的约束，例如限制基数和允许值，以更准确地反映实际业务需求 <p>在Enterprise Architect中，子集化过程是使用Schema 编辑器执行的。</p> <p>Schema 编辑器允许建模者从源包中选择所需类的子集，并为每个选择的类选择所需属性的子集。然后将选定的类及其减少的属性集复制到目标包。大多数情况下，源包是来自NIEM参考模型的 NIEM-core 命名空间包。在这种情况下，目标包也将是一个名为“NIEM-core”的命名空间包，但它将是您的NIEM IEPD模型的一部分。</p> <p>参考模型中的其他其它空间包，例如 Domain包，也可以以相同的方式进行子集化。</p> <p>使用Enterprise Architect的Schema 编辑器工具将 NIEM-core 参考包的子集复制到 IEPD模型中的 NIEM-core 子集包。目标是通过重用已在 NIEM-core参考模型中定义的类型和元素来模型尽可能多的数据交换。</p> <p>如果您的模型也将使用NIEM域包，则应为您使用的每个域包重复此子设置过程。</p> <p>有关更多信息，请参阅帮助主题帮助<i>NIEM with the Schema</i> 编辑器。</p>
创建扩展包	<p>创建NIEM数据模型时，目标是使用NIEM参考模型中的类型和元素来模型尽可能多的数据交换。无法通过重用现有NIEM元素建模的内容然后在“扩展”命名空间包中建模，方法是使用 NIEM-UML 配置文件中的元素创建新类型和元素，所有类型最终都派生自 XML 模式原始类型。</p> <p>NIEM Starter模型（来自模型向导工具箱）和图表Starter模型模式（来自模型工具箱）都提供了<<InformationModel>>包，其中包含各种模式使用这些包中的 PIM 图，您可以通过添加图表工具中的元素来图表不同模式的工具箱。</p> <p>建议您使用'exchange'包中的图表，组装您的exchange的高级模型，根据需要其他schema包中的类型和元素。</p> <p>大多数 IEPD 需要扩展模式来定义特定类型和属性，这些类型和属性对于正在定义的数据交换是唯一的。然而，NIEM模型没有定义特定的消息类型或结</p>

构来组装交换中的所有对象。因此，由 IEPD 的创建者编写一个扩展模式来声明根元素和消息的基本结构。根的元素汇集了交换中定义的所有对象和关联。

虽然您不需要创建单独的模式来声明根的元素和基本结构，但将特定于消息的扩展分离到“交换”模式中并将更通用的扩展分离到“扩展”模式中可能是有益的。Exchange 模式包含对消息类型或消息类型组唯一的定义。这通常只包括根元素及其类型，可能还有一些构成消息基本框架的结构元素。

将模式元素组织成“交换”和通用“扩展”组还提供了跨多个 IEPD 共享更通用模式的可能性，而“交换”模式通常特定于一个特定的 IEPD。您还可以有多个“交换”模式，以表示不同的消息类型或不同消息类型的组。

使用Schema 编辑器对NIEM进行子集化

Enterprise Architect的Schema 编辑器是一个可以大大简化从NIEM模型参考命名空间包创建子集的过程的工具。

访问

使用此处列出的任何一种方法来显示Schema 编辑器模型使用对话框。

输入新模型转换的名称，然后从“架构集”下拉列表中选择“国家信息交换模型 (NIEM) ”。

在您的项目中保存一个合适的工件模型的配置文件（在您的工件包很容易找到合适的根包）。

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：新建 > 模型转换
-----	---

创建子集模型

NIEM专家建议第一步是为您的 XML 交换创建一个UML模型，因为它允许您捕获业务需求，而不会受到NIEM中的操作方式的过度影响。

一旦您有了用于交换的UML模型的初稿，您就可以开始使用NIEM重新创建该模型。

最初，在NIEM参考模型中找到合适的类型和属性似乎是一项不可能完成的任务。随着您获得经验并熟悉NIEM模型的内容，这将变得更加容易。

您通常使用的大多数NIEM类型，例如 PersonType、OrganizationType、DocumentType或 ActivityType，都有许多属性，您通常只需要其中的几个。这是子集变得有用的地方。

如果您尝试使用他们的姓名、地址和生日来模型一个人，您可以从 NIEM-core 中选择 PersonType 和 AddressType。从这些类型中，仅选择您的模型所需的属性。

如果所选属性引用了其他类型，这些类型将自动添加到Schema 编辑器中。

当您“生成”您的子集时，Enterprise Architect创建子集所需的目标模式包，然后将所选类型及其缩减的属性集复制到目标包中。

进一步完善您的子集

创建子集后，您可以通过调整类型内属性的基数或限制属性的允许值来进一步属性它。

要调整基数或限制属性的允许值，请在Schema 编辑器的中央属性中选择该属性，然后右键单击并选择“限制此属性”。将显示“属性限制”对话框，您可以在其中调整基数或根据需要对属性应用限制。

单击“更新”以保存对模型转换配置文件的更改，然后单击“生成”以重新生成具有应用限制的子集模型。

NIEM子集化通常是一个迭代过程。使用保存的模型转换配置文件，您可以在 IEPD 开发的各个阶段根据需要重新加载、更新和重新生成子集。

使用Schema 编辑器对NIEM进行子集化

节	行动
1	打开Schema 编辑器。(见访问:功能区)

2	<p>创建一个新的Schema 编辑器配置文件。</p> <p>单击新建按钮并选择 “模型转换”。</p> <p>在打开的对话框中，指定配置文件的名称并在 “架构集” 字段中选择 “NIEM”。</p> <p>(此对话框中的 “命名空间” 字段不用于NIEM，因为NIEM使用其模型包上的标记值来指定命名空间。)</p> <p>选择一个位置来保存您的新个人资料，然后单击确定按钮。</p>
3	<p>在浏览器窗口中，在参考模型的 NIEM-core包中找到所需的 PersonType 和 AddressType 类型。</p> <p>将所需类型从浏览器窗口拖放到Schema 编辑器的 “类” 窗格中。</p>
4	<p>现在在Schema 编辑器的 “类” 窗格中选择一种类型，比如 PersonType。</p> <p>PersonType 的完整属性列表显示在 “属性” 窗格中。</p>
5	<p>使用 “属性” 列表中的复选框来选择 PersonType 的属性以在您的交换模型中使用。在这种情况下，请选中 PersonBirthDate 和 PersonName 复选框。</p> <p>当您选择这些属性时，Schema 编辑器会自动将类型 “DateType” 和 PersonNameType 添加到类列表中，因为这些类型被您刚刚选择的属性引用。</p>
6	<p>现在在 “类” 窗格中选择 “日期类型”。</p> <p>'DateType' 有四个属性，DateAccuracyAbstract、DateAugmentationPoint、DateMarginOfErrorDuration 和 DateRepresentation。这些属性中的前三个是日期元数据——它们不包含日期值。第四个，DateRepresentation，是一个抽象属性，所以它也不直接保存日期值。它用作最终保存日期值的属性的占位符。</p> <p>NIEM模型通常使用XML Schema抽象元素和替换组。</p> <p>抽象元素为子集的增加了一些复杂性，因为您需要添加抽象元素以及将替换抽象元素的那些元素。</p> <p>例如，大多数与日期相关的类型都包含抽象元素nc:DateRepresentation，可以用 nc:Date、nc:DateTime 等替代。</p>
7	<p>选择属性 DateType.DateRepresentation。</p> <p>您会注意到另一种类型 DateRepresentationPropertyHolder 已添加到 “类” 列表中。</p>
8	<p>在 “类” 列表中选择 DateRepresentationPropertyHolder。</p> <p>无类型属性 DateRepresentation 被称为替换组的 “头”。必须在替代的客户端 DateType 以及替代的供应商 DateRepresentationPropertyHolder 中选择此属性。作为替换组首的属性是为您预先选择的，因此您只需在 DateType 中选择最终将替换为 DateRepresentation 的属性。选择属性 Date:date - 它将用作实际保存数据值的 DateRepresentation。</p> <p>在涉及替换组的情况下，简单地添加抽象元素而不添加来自相关 PropertyHolder 类型的可替换元素是一个常见的错误。</p>
9	<p>通过从 PersonNameType 类中选择 PersonGivenName、PersonMiddleName 和 PersonSurName，对 PersonName 属性重复该过程。</p>
10	<p>要将您当前选择的类和属性保存到您正在创建的配置文件中，请单击更新按钮。</p> <p>这将使用您当前的选择更新配置文件，如果您需要对其执行进一步的工作，则可以在以后重新加载它。这有助于创建子集包的迭代过程。</p>
11	<p>现在单击 “生成” 选项。</p> <p>在 “架构导出” 对话框中选择 “NIEM模型子集”，然后单击 “生成” 按钮。</p> <p>导航到包含您正在构建的 IEPD 的包层次结构。选择将包含子包的父包，然后单击确定按钮。</p>

12

您在Schema 编辑器中选择的类将被复制到目标包，仅包含您选择的属性子集。

注记

- 请通读每个演练示例 - 每个示例都包含重要信息
- 支持NIEM开发的Schema 编辑器功能，协助创建 Subset Schemas；它无助于产生扩展模式

浏览示例

如果您不熟悉使用NIEM的Schema 编辑器，请花时间阅读这些示例。每个示例都包含重要信息，有助于确保您的模型使用有效的NIEM子集，最终生成有效的 XML 模式文件。

示例1：添加类和选择属性

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器来执行添加类和选择要包含在NIEM子集包中的属性的基本操作。

节	描述
1	<p>打开包含NIEM 5.0参考模型以及NIEM IEPD 类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，则打开一个新项目并使用“IE模型向导加载 参考模型和类型”（首页 从模式创建”选项卡）。</p>
2	<p>使用首页的“从模式创建”选项卡（模型向导），将NIEM 5 IEPD Starter模型的新副本添加到您的项目中。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。在生成 IEPD 时，使用该object实例的名称来命名创建 IEPD 所在的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为“工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'AircraftType'。</p> <p>将'AircraftType'拖放到Schema 编辑器的左侧列（标记为“类”）。</p> <p>您会注意到“ConveyanceType”和“ItemType”类被自动添加到类列表中。</p> <p>“ItemType”和“ConveyanceType”是派生“AircraftType”的超类型。</p>
5	<p>在“类”列中选择 AircraftType。</p> <p>您会注意到中心列“属性”显示了属于此类的属性的完整类。</p> <p>还列出了父类的属性。</p> <p>要在子集模式中包含一个属性，您只需在它旁边放置一个复选标记。</p> <p>（您应该只选择当前在“类”列表中的类的属性。</p> <p>如果您需要父类的属性，请选择类，然后选择其属性。）</p> <p>在 AircraftTailIdentification 旁边打勾。AircraftTailIdentification 的类型是 IdentificationType。</p> <p>请注意，IdentificationType 已添加到类列表中。</p> <p>Enterprise Architect自动将那些被引用为您选择的属性类型的分类器添加到“类”列表中。</p>
6	<p>在Schema 编辑器的左侧栏中选择类IdentificationType。</p> <p>在中心列中，在属性 IdentificationID 旁边放置一个复选标记。IdentificationID 的类型是“string”。</p> <p>'string' 类型是原始类型 - 它没有添加到类列表中。</p>
7	<p>现在，选择Schema 编辑器左侧栏中的类ConveyanceType。</p>

	<p>在属性 <code>ConveyanceMotorizedIndicator</code> 旁边放置一个复选标记。</p> <p>'boolean' 类型是原始类型 - 它没有添加到类列表中。</p>
8	<p>在Schema 编辑器的左侧栏中选择类 <code>ItemType</code>。</p> <p>在属性 <code>ItemMakeName</code>、<code>ItemModelName</code> 和 <code>ItemModelYearDate</code> 旁边放置复选标记。</p> <p>类型 <code>ProperNameTextType</code> 和 <code>TextType</code> 会自动添加到类列表中。 <code>TextType</code> 是 <code>ProperNameTextType</code> 的基础类。</p>
9	<p>单击更新按钮将选定的类和属性保存到配置文件，然后单击生成按钮。</p> <p>在打开的窗口中，选择“NIEM模型子集”，然后单击生成按钮。</p> <p>系统将提示您选择将在其中创建子模型的包。通常，您会选择作为 Exchange 架构包的父级的包。在 starter模型中，交换包名为“IEPD-Exchange”，其父包名为“NIEM 5模型”，尽管您可能在前面的步骤 2 中重命名了这些包。</p> <p>选择包“NIEM 5模型”，然后点击确定按钮。</p> <p>注记：当创建更复杂的模型时，您的子集可能包括来自多个不同 <<InformationModel>>包的类。Enterprise Architect的Schema 编辑器会自动创建所需的目标包，并将您正在设置子集的Classes 复制到标记值'targetNamespace' 与提取原始类的源包的标记值匹配的目标包中。子集 <<InformationModel>>包将被创建为您选择作为生成目标的包的子项。</p>
10	<p>一旦生成完成，展开目标<<InformationModel>>包。</p> <p>您将看到您选择的类及其减少的属性集。</p>

示例2：使用关联类型

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器将关联类型和它们引用的类型添加到您的NIEM子集包中。

节	描述
1	<p>打开包含NIEM 5.0参考模型以及NIEM IEPD 类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，则打开一个新项目并使用模型加载“参考模型类型”并在模型向导首页“创建模式”选项卡)。</p>
2	<p>使用模型向导 (首页“从模型创建”选项卡) 将模式5 IEPD Starter模型的新副本NIEM到您的项目。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。在生成 IEPD 时，使用该object实例的名称来命名创建 IEDP 所在的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为“工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'PersonLocationAssociationType'。</p> <p>将'PersonLocationAssociationType'拖放到Schema 编辑器的左侧列 (标记为“类”) 。</p> <p>您会注意到中心列“属性”显示 PersonLocationAssociationType.Attributes 以及关联。</p> <p>在关联 Location 和Person旁边放置复选标记。</p>

	LocationType 和 PersonType 类型会自动添加到Schema 编辑器的“类”列表中。
5	<p>类PersonLocationAssociationType 派生自超类型'nc:AssociationType'，但在这种情况下，超类型不会自动添加到类列表中。</p> <p>如果您想在生成的子集中包含超类型“nc:AssociationType”的任何属性，则必须手动将“nc:AssociationType”添加到Schema 编辑器的类列表中，然后选择所需的属性。</p> <p>如果您不想特别包含“nc:AssociationType”的属性，则无需将其添加到 Classes 列表中。</p> <p>当模式文件最终从子集包生成时，Enterprise Architect将在需要时为“nc:AssociationType”生成元素和类型定义，即使它没有显式建模。</p>
6	<p>单击更新按钮，然后单击生成按钮。</p> <p>在打开的窗口中，选择“NIEM模型子集”，然后单击生成按钮。</p> <p>系统将提示您选择将在其中创建子模型的包。通常，您会选择作为 Exchange 架构包的父级的包。在 starter模型中，交换包名为“IEPD-Exchange”，其父包名为“NIEM 5模型”，尽管您可能在前面的步骤 2 中重命名了这些包。</p> <p>选择包'NIEM 5模型'，然后点击确定按钮。</p>
7	<p>在子模型中找到名为“niem-core”的<<InformationModel>>包。在这个包中创建一个NIEM PIM 图，然后将这个包中的三个类拖放到图上。您会注意到属性“Person”和“Location”被建模为关联与类型 PersonType 和LocationType 之间关联的关联端。</p>

示例3：使用替换组和属性持有者

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器属性正确地将替换组和持有者添加到您的NIEM子集中。

节	描述
1	<p>打开包含NIEM 5.0参考模型以及NIEM IEPD 类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，则打开一个新项目并使用该模型向导加载参考模型并加载 IE 模型。</p>
2	<p>使用首页的“从模式创建”选项卡（模型向导），将NIEM 5 IEPD Starter模型的新副本添加到您的项目中。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。在生成 IEPD 时，使用该object实例的名称来命名创建 IEDP 所在的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为“工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'AircraftType'。</p> <p>将“AircraftType”拖放到Schema 编辑器的左侧列（标记为“类”）。</p> <p>您会注意到类 ConveyanceType 和 ItemType 被自动添加到类列表中。</p> <p>ItemType 和 ConveyanceType 是派生自 AircraftType 的超类型。</p>
5	<p>在Schema 编辑器的左侧栏中选择类'AircraftType'。</p>

	<p>在中心列中，在属性 AircraftWingColorAbstract 旁边放置一个复选标记（注意该属性没有指定类型）。</p> <p>类AircraftWingColorAbstractPropertyHolder 会自动添加到类列表中。</p>
6	<p>选择左侧栏中的类'AircraftWingColorAbstractPropertyHolder'。请注意，这个类还有一个名为“AircraftWingColorAbstract”的属性，它没有指定类型。该属性是为您预先选择的 - 它应该保持选中状态。</p> <p>只需在 AircraftWingColorText 旁边打勾。</p>
7	<p>在这种情况下，属性 AircraftWingColorAbstract 是替换组的头部，并提供客户端类AircraftType 和供应商类AircraftWingColorAbstractPropertyHolder 之间的连接。</p> <p>AircraftWingColorText 是将添加到 AircraftType 的实际属性（TextType 类型）。</p>
8	<p>某些 PropertyHolder 类型将具有多个属性 - 替换组头，以及许多其他属性。必须始终在客户端和供应商类中都选择作为替代组头的属性。Enterprise Architect在供应商类（PropertyHolder）中为您预先选择了这个属性。然后，您只需从供应商处选择要替换的属性来代替替换组的负责人。</p>

示例NIEM架构

此页面提供了从头到尾定义新的NIEM兼容架构的概述。

NIEM建模所需的框架包已在前面的主题中进行了描述。模型向导包（首页）“Create from模式”选项卡还提供了一个方便的起点，用于定义您的IE。将其导入您的模型时，您会发现包含IEPD类型实例的图表，其中设置了运行状态属性以显示您最可能需要设置的核心。

本节介绍从模式中获取示例IEPD并创建“Hello World”样消息的过程，其中请求基于面部图像的个性化消息。响应将是图中人物的身份和给他们的个性化信息。

导入NIEM框架包

Enterprise Architect中的NIEM建模从NIEM技术架构委员会和物件管理组NIEM-UML规范定义的标准类型开始，如此处所述。这些可从我们的可重用资产服务器和模型模式向导中获得。

要将这些导入您的模型：

- 打开首页'从模式创建'选项卡（模型向导）
- 查找蓝图'NIEM 3、4和5'
- 选择您的模型所需的包
- 单击模型(s)按钮将所选模式导入您的模型。

注记：

- 所有NIEM 5模型都需要NIEM IEPD Types包以及NIEM参考模型包之一
- 所有NIEM 3和4模型都需要NIEM MPD Types包以及NIEM参考模型包之一
- 所有NIEM 2.1模型都需要NIEM 2.1参考模型包，但不是MPD类型包，因为NIEM 2.1 MPD元素可从NIEM 2.1图表工具箱

部件	细节
NIEM框架	NIEM的强大主要来自于可用于构建自己的模式的广泛类型库。Enterprise Architect完成成为NIEM以及所有版本的NIEM 4提供了完成NIEM框架NIEM这些框架都可以从首页的“创建模式”选项卡（模型向导）中获得。本教程使用NIEM 5.0框架，因此选择该模式进行导入。
来自NIEM-UML的IEPD类型	用户定义的NIEM架构围绕IEPD构建，该IEPD为架构A使用者定义了如何使用包含的各种XSD文件以及定义了哪些消息类型。 在UML中建模时，使用UML配置文件中定义的多个类的实例创建IEPD。Enterprise Architect在一个包中提供了这些类，该包可从首页“从模式创建”选项卡（模型向导）中获得。 所有NIEM 5模型都需要这些IEPD类型，因此请选择“NIEM 5 IEPD Types”模式进行导入。

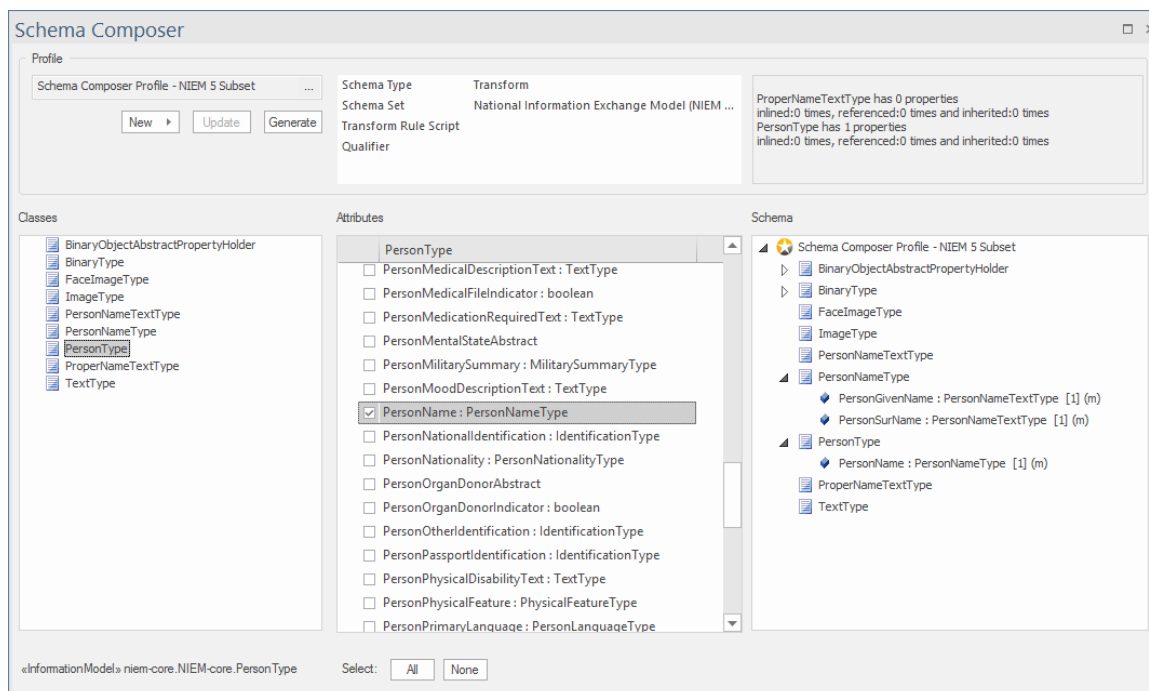
子集NIEM命名空间

工件模式包括一个Schema编辑器模型，用于指定一个子集。双击它打开Schema编辑器并开始子集化过程。

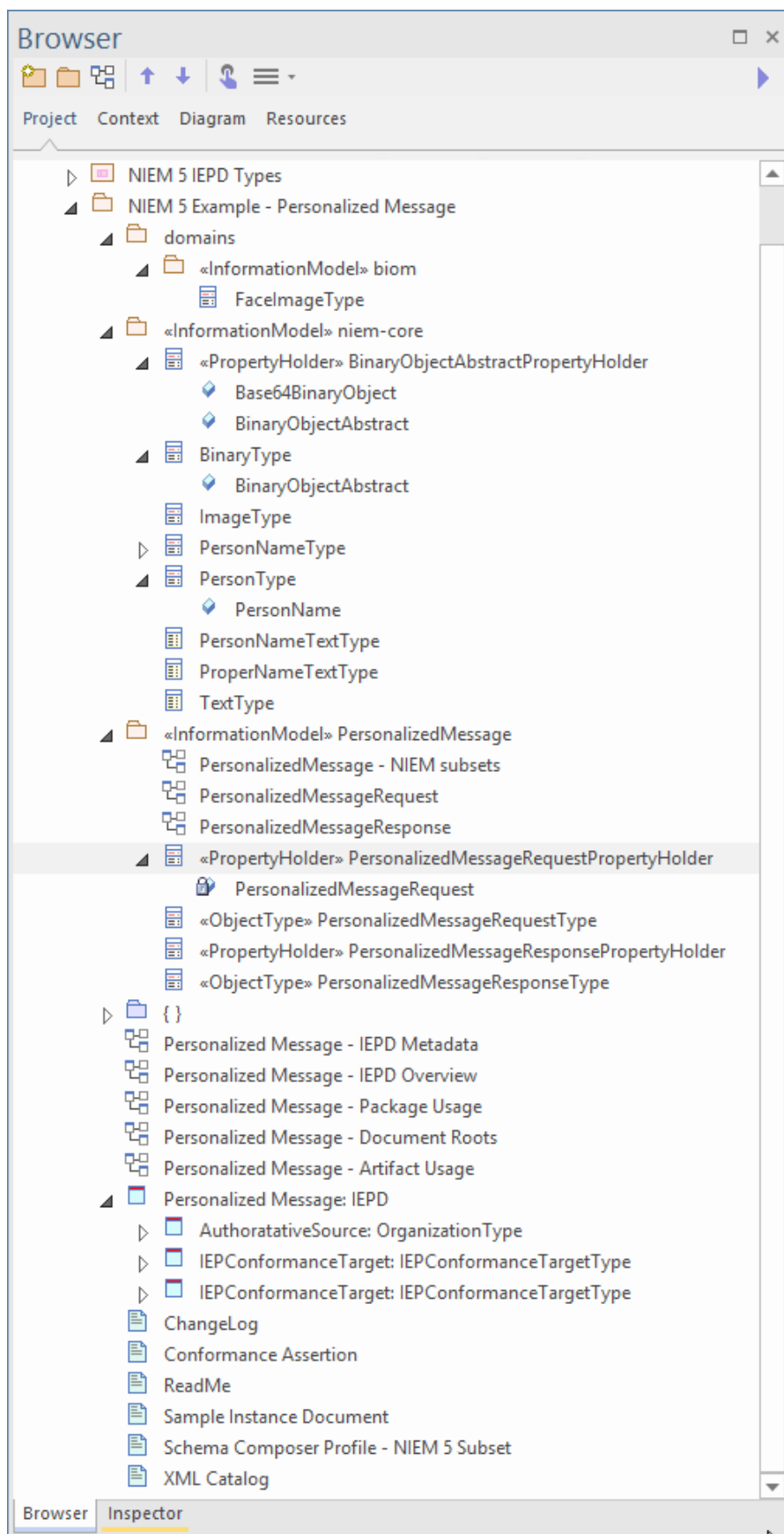
我们希望我们的请求消息发送用于面部识别的面部图像。为此，我们需要从Biometrics包中子集适当的类型。开始通过在NIEM 5.0参考模型中的Domains\Biom包中找到类型FaceImageType。将此类型拖入Schema编辑器。该类型所继承的超类型会自动添加到Schema编辑器中。

我们的响应消息需要来自'niem-core'包的PersonType。将此类型也拖到Schema编辑器上。

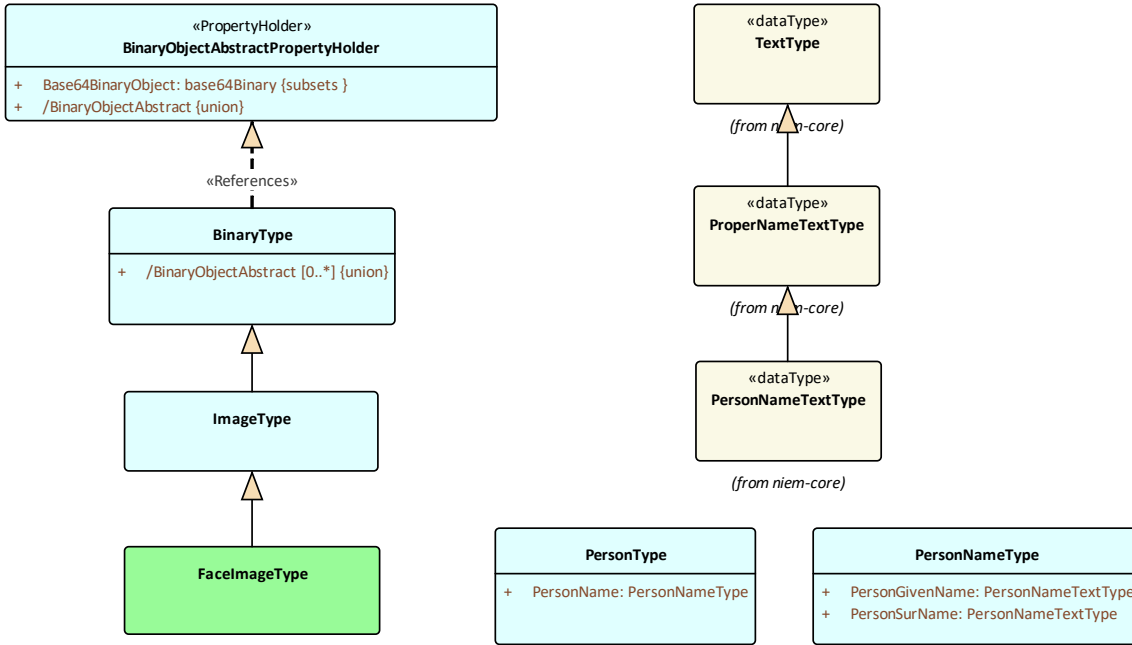
此图显示了在NIEM 5.0参考模型中跨多个命名空间选择类型和属性的子集：



一旦选择了所需的类型，您就可以生成子集。出现提示时，选择将生成子命名空间包的父包。生成后，子集包中的类应如下所示：

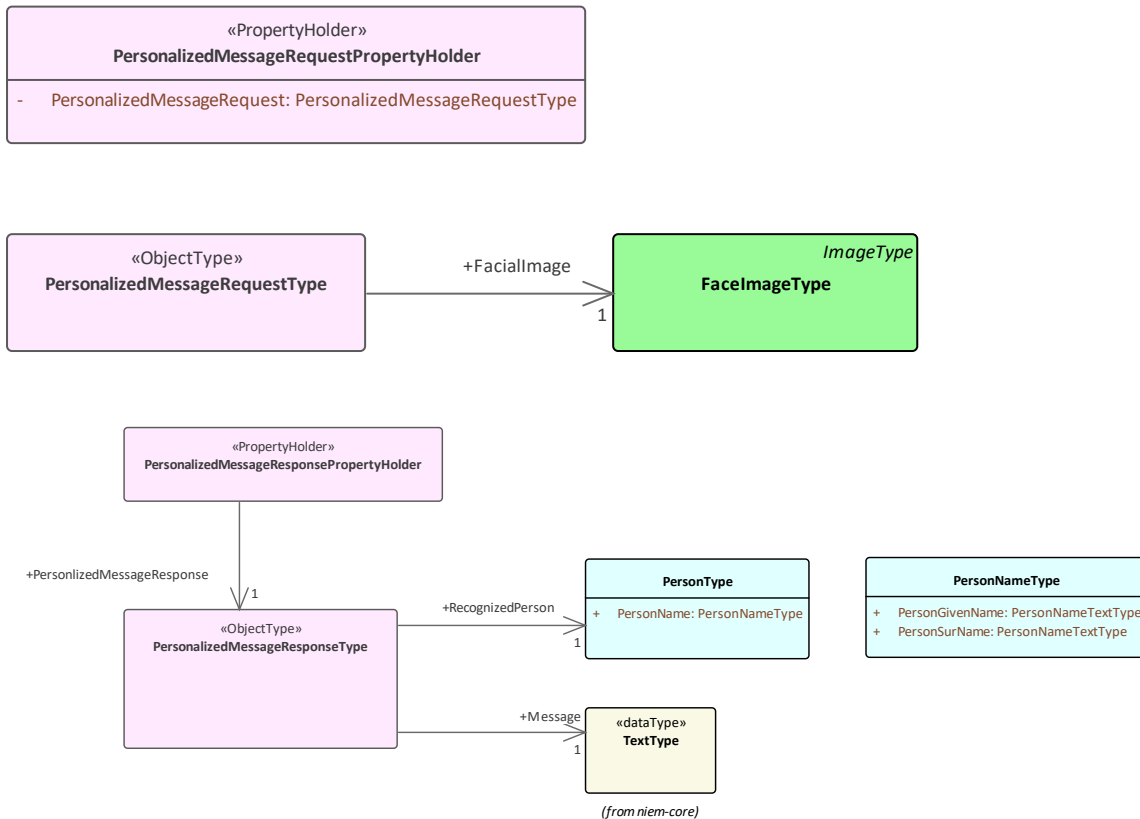


我们现在可以创建一个现在PIM 图NIEM我们所有的子类类放在该图上，以生成类似于以下内容的内容：



创建扩展类型

我们将定义两条消息，一条请求和一条响应。对于这些消息中的每一个，我们都需要定义文档根元素。这些将被建模为NIEM模式的扩展。现在我们已经定义了我们的子集包，我们可以定义这些文档根。因为我们只创建了两种简单的文档类型，所以只需要为每条消息创建一个 PropertyHolder 和 ObjectType。ObjectTypes 链接到我们 NIEM 框架中选择的类型，用于描述每条消息的内容，如下所示：

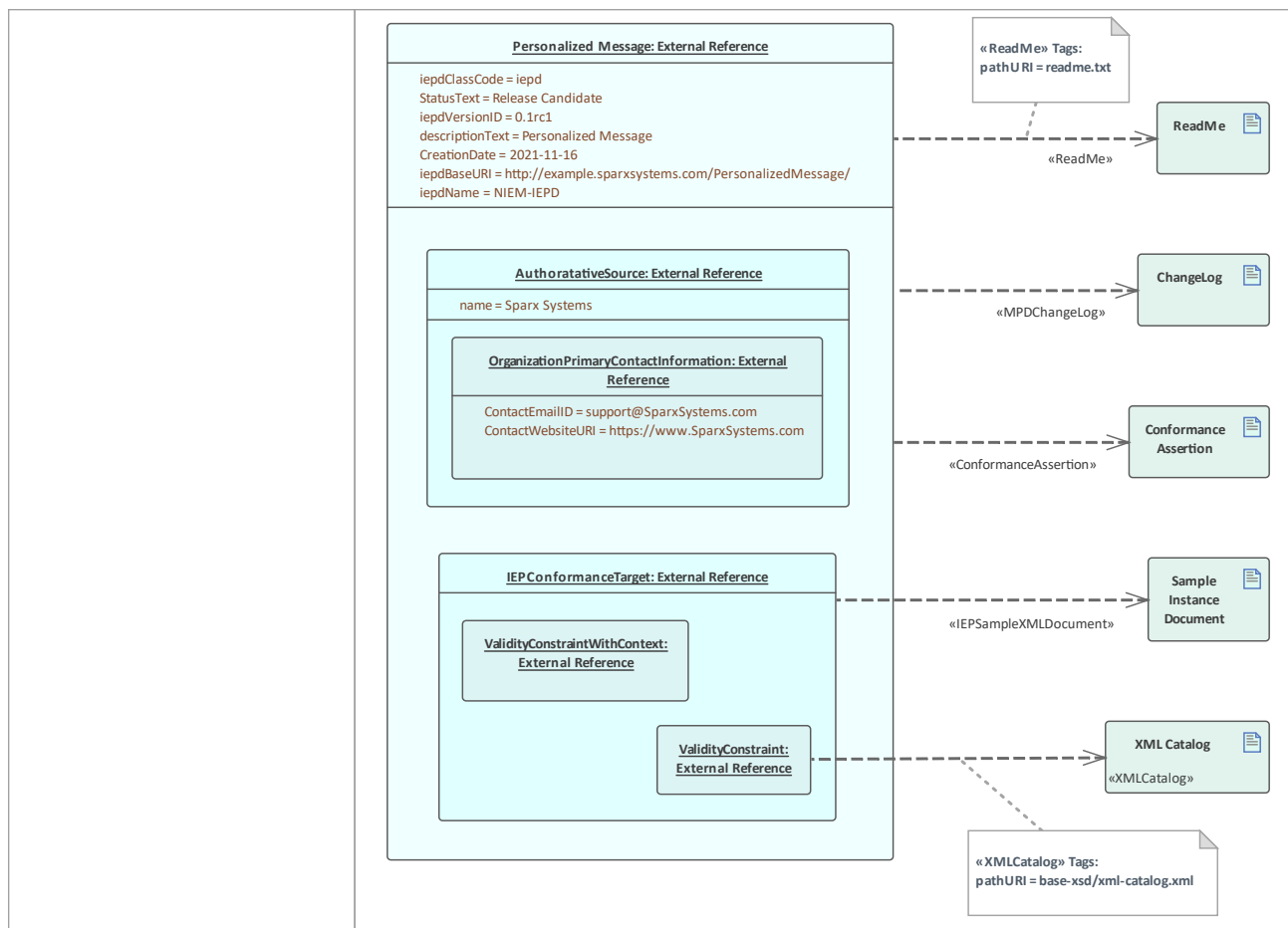


自定义 IEPD

IEPD类的实例携带标识 IEPD 的信息，它与各种其他模型工件之间的链接决定了在生成模式文件和目录文件时生成什么（以及生成到哪里）。这里描述了要点。

部件	描述
IEPD 元数据	<p>模式中的顶级object是 IEPD 类的一个实例。IEPD 的名称是物件本身的名。所有其他属性都处于object的运行状态。</p> <p>该图显示了 IEPD 在提供真实信息后的样子。</p> <div data-bbox="528 613 1294 1303" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Personalized Message: External Reference</p> <p>iepdClassCode = iepd StatusText = Release Candidate iepdVersionID = 0.1rc1 descriptionText = Personalized Message CreationDate = 2021-11-16 iepdBaseURI = http://example.sparxsystems.com/PersonalizedMessage/ iepdName = NIEM-IEPD</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">AuthoritativeSource: External Reference</p> <p>name = Sparx Systems</p> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">OrganizationPrimaryContactInformation: External Reference</p> <p>ContactEmailID = support@SparxSystems.com ContactWebsiteURI = https://www.SparxSystems.com</p> </div> </div> </div> <p>NIEM-UML 建议iepdBaseURI 的最后一部分与IEPD 的名称相匹配，并指定 iepdVersionID 将附加到iepdBaseURI 以生成生成的iepdURI。此示例遵循该约定。</p> <p>该模式将iepdClassCode 的值默认为'iepd'。这意味着 IEPD 旨在表示信息交换包文档 (IEPD)。这是最常见的 IEPD 类型，也是我们想要创建的，所以一直保留默认值。</p>
定义的文档类型	IEPD 应定义一种或多种文档类型。每一个都是名为“IEPConformanceTarget”的 IEPConformanceTargetType 实例。提供的模型模式已经包括其中之一，但我们需要第二个，如下所示：

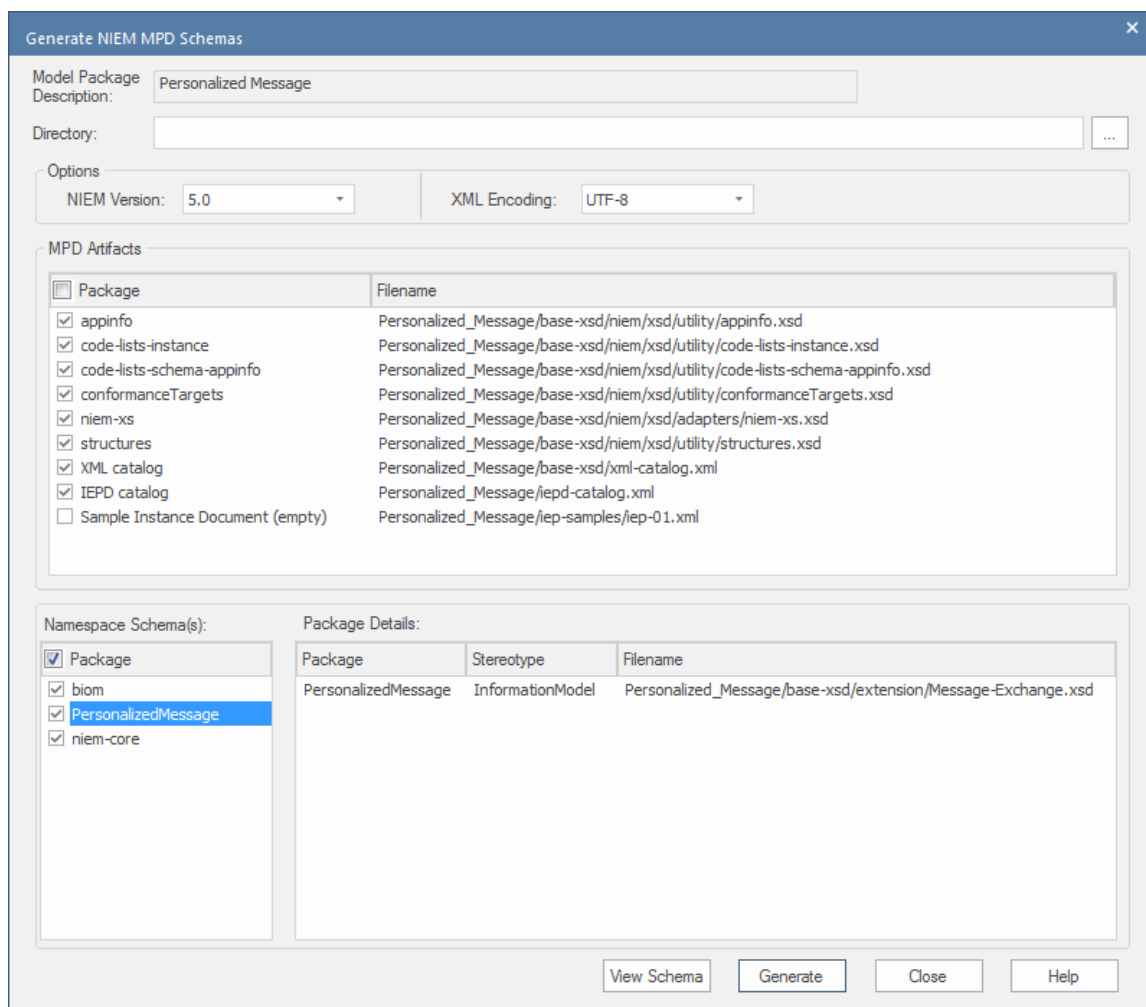
	<p>注册QualifiedNamesType的实例，其qualifiedName关系到一个PropertyHolder。这指定正在描述的文档的顶层将是来自所包含属性之一的元素。主题创建NIEM数据模型中的创建扩展包部分描述了它是如何定义的。</p>
用途包	<p>将 IEPD 实例连接到信息模型的关系指定要使用该 IEPD 生成哪些模式文件。在此示例中，我们使用来自两个不同NIEM命名空间的类型。子设置过程为每个创建了一个 InformationModel包，其中命名空间标记值与原始匹配，目的设置为子集。我们还创建了一个扩展包，我们在其中定义了我们自己的类型以及如何使用NIEM类型。</p> <p>此图显示了它的外观：</p> <p>使用的关系还指定包使用方式以及该包定义的模式的路径。</p>
附加文件	<p>NI 预计所有NIEM包都至少包含更改log和自述文件，但也支持其他几种类型的工件包。在Enterprise Architect的工件关系。与包使用一样，关系指定文件的位置。</p> <p>在此图像中，描述了每种文档类型的自述文件、变更日志和示例文档。这会将有关这些文件的信息添加到目标目录文件中。Enterprise Architect不会创建这些文件，它们的内容超出了本教程的范围。</p>



生成IEPD

要生成您的 IEPD :

- 在图表或浏览器窗口中选择 IEPD 实例规范。
- 从 特定”功能区中，选择 技术> NIEM >生成NIEM架构”选项



该对话框显示标准NIEM工件和可作为模式生成的链接命名空间列表。设置目标目录并单击“生成”按钮以创建建模的 IEPD。

导入NIEM XML Schema

除了在Enterprise Architect中生成NIEM模式外，您还可以将外部 NIEM 特定XML Schema文件作为UML模型导入（反向工程）到Enterprise Architect项目中。

访问

功能区	特定>技术> NIEM >导入NIEM架构 特定>技术> NIEM 2.1 >导入NIEM 2.1架构
上下文菜单	在浏览器窗口中： 右键包 特定 NIEM 导入NIEM架构 右键包 特定 NIEM 2.1 导入NIEM 2.1架构

导入一个NIEM特定的XML Schema

选项	行动
包	在浏览器窗口中显示当前选定包的名称，作为导入NIEM模式的包。 您可以通过单击  按钮并检查“导航员”对话框来验证您使用的是适当的包；如有必要，请选择不同的包。
目录	单击  按钮并浏览包含源NIEM架构文件的目录。单击要导入的每个文件，然后单击浏览器打开按钮。
选定的文件	列出选择导入的XML Schema文件。
导入引用的XML Schema (s)	如果您要导入“选定文件”字段中列出的任何文件所引用的任何其他XML Schema，请选中此复选框。
跳过架构如果在模型中命名空间	如果模型中已存在 XML 模式，则如果要跳过导入XML Schema，请选中此复选框。 Enterprise Architect将使用架构命名空间和名称来确定它是否存在于模型中。
图表XML Schema (s)	选中此复选框以在每个导入的命名空间包下创建类图（NIEM PIM图）。
布局创建图表	（仅在选择XML Schema（S）选项的“创建图表”时才启用。） 选中此复选框以自动布置创建的类图。
导入	单击此按钮开始导入过程。 导入进度报告在系统输出窗口的“NIEM Importer”选项卡中。当导入完成时，还会显示消息框；单击确定按钮以清除消息。
关	单击此按钮可关闭“架构导入器”对话框。

帮助	单击此按钮可显示此帮助页面。
----	----------------

注记

- Enterprise Architect使用XML Schema的 XSD导入和 XSD Include 元素中的`schemaLocation`属性来确定文件之间的依赖关系；此属性必须设置为有效的文件路径（而不是 URL），才能正确导入依赖的XML Schema
- 'Create图表for XML Schema (s)' 选项为每个导入的模式文件生成图表，但仅显示用户特别选择的模式文件的图表；它不显示引用的架构文件的图表
- 如果您导入大型架构文件，建议您取消选择“为XML Schema (s) 创建图表”选项，因为这会大大增加导入所花费的时间

地理空间模型

互联网的普及、手机的普及和定位服务的普及，导致几乎每个人在日常生活中都以某种形式与定位信息进行交互。对于政府和组织而言，将此类信息作为战略决策的一部分也变得至关重要。地理空间信息可以在Enterprise Architect中建模，也可以与其他数据集成，以形成其他工具无法实现的单一和全面的信息视图。


Enterprise Architect通过使用MDG 技术，支持地理标记语言 (GML) 应用程序模式和 ArcGIS 地理数据库的建模。这些模型的信息前体——例如社区概念模型——也可以被建模，并且可以使用可追溯性将模型连接在一起。


建模工具

工具	描述
ArcGIS配置文件 	Enterprise Architect支持为 Esri Inc. 开发的 ArcGIS 10.0 工具套件设计地理数据库。
地理标记语言 	Enterprise Architect中的地理标记语言 (GML) 是开放地理空间联盟的地理标记语言 3.3 的实现，它为Enterprise Architect中的地理特征建模功能提供了 XML 语法。

开始

Enterprise Architect将工具的广泛特征划分为多个视角。这可以帮助您聚焦于特定任务并使用您需要的工具，而不会分散其他特征的注意力。要使用ArcGIS Geodatabases或Geography Markup Language特征，您首先需要选择以下透视图：

 数据库工程 > ArcGIS

 信息交换 > 地理 GML

设置透视图可确保ArcGIS Geodatabases和Geography Markup Language图表、它们的工具箱页面和透视图的其他特征默认可用。

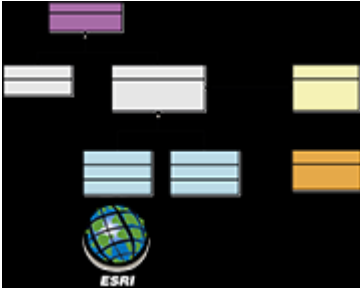
ArcGIS 地理数据库

使用 ArcGIS特征Enterprise Architect，您可以轻松地可视化地理数据库。这使您可以将在传统软件中心和工程系统中工作的团队与定义特征和域的地理空间团队统一起来。为系统或提供系统功能的组件定义战略业务规则和要求，团队可以与地理空间团队共享模型，从而创建有助于集成和降低风险的集成模型。

地理标记语言 (GML)

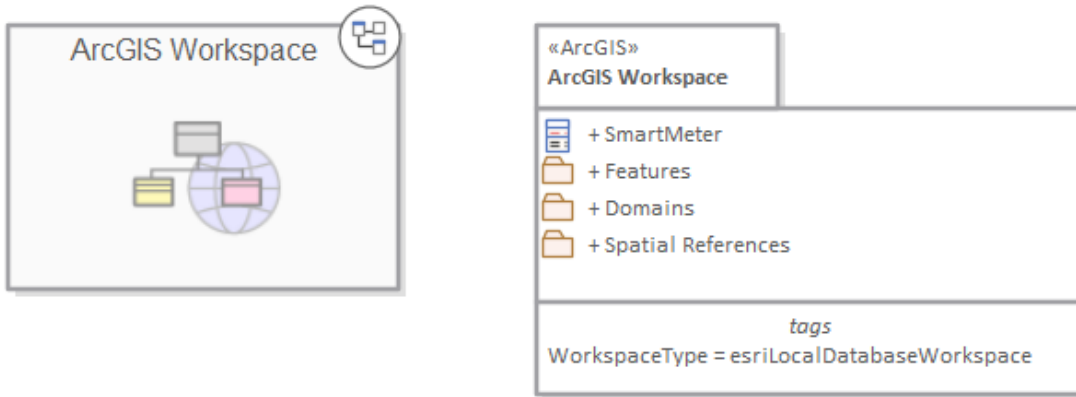
使用地理标记语言 (GML)功能，您可以模型基于组织或社区的应用程序模式。这些模型可用于显示特征之间的关系，而这些元素又可以包含任何数量的属性来限定其特征。这些可以基于定义的类型、数据类型、代码列表或枚举。

ArcGIS 地理数据库



交换、模型和可视化 ArcGIS 地理数据库

Enterprise Architect支持 ArcGIS 地理数据库的导入和导出，允许您在这个多功能协作平台中可视化特征和域。在最近的过去，系统软件开发和地理空间开发之间的学科出现了分离。在这个社会架构和数字颠覆的时代，几乎每个项目和努力都需要位置信息的某些方面，从简单的交付服务到农业、采矿、勘探、天气、房地产和灾难恢复系统。



显示导航单元的包图和包含特征域和地理空间参考的包

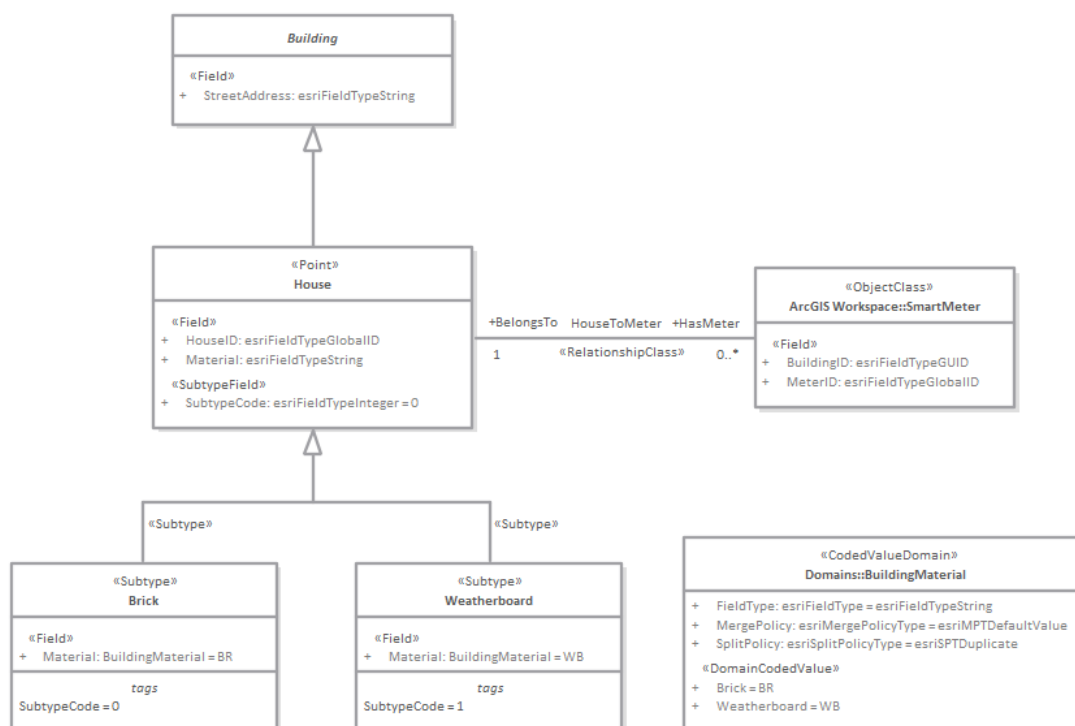
由 Esri 开发的 ArcGIS 系统支持地理数据库的开发和管理。与其他数据库一样，使用UML等标准符号对地理数据库的设计进行模型很有用。您可以使用 ArcGIS 的UML配置文件在Enterprise Architect中执行此类建模。在 Enterprise Architect中为 ArcGIS 模式建模后，您可以将模型作为 XML工作空间文档导出到 ArcGIS。您还可以通过将 ArcGIS XML工作空间文档导入Enterprise Architect来可视化现有的 ArcGIS 地理数据库方案。

示例图表

ArcGIS 图表允许您可视化构成地理数据库方案的地理特征、域和其他元素。在此示例中，建造建筑子类型化为房屋，而房屋又根据材料类型进行子类型化。房屋子类型引用了一个编码值域，图中也显示了两个域编码值：

- 砖
- 气象板

智能电表与房屋相关。该房屋属于建造类型，而建造包含 Street Address 的属性



导出 ArcGIS XML 工作空间

为地理数据库工作空间XML 文档（包含 ArcGIS 模式）建模后，可以将其导出到外部目录（使用 Publish 模型包功能），然后可以从该目录将其导入 Esri ArcCatalog。

访问

单击浏览器窗口中的 ArcGIS 原型包（您的 ArcGIS 工作空间包）。

功能区	特定>工作技术>ArcGIS>导出到ArcGIS工作空间XML或发布>模型交换>发布为...
上下文菜单	右键包 特定 ArcGIS 导出到ArcGIS工作空间XML
键盘快捷键	Ctrl+Alt+E：发布

导出工作空间

选项	行动
根包	显示选中的 ArcGIS 工作空间包的名称。
文件名	类型输入或浏览要生成 XML 文件的文件路径。
XML 类型	选择 “ArcGIS” 作为要导出包的 XML/XMI 版本。
格式 XML 输出	将输出格式化为可读的 XML（在运行结束时这需要几运行）。
写入日志文件	写一份出口活动的 log（推荐）。 log 文件保存到 XML 文件导出到的目录中。
视图 XML	单击此按钮可查看导出的 XML 文件。
导出	单击此按钮以启动 XML 导出。
关	单击此按钮可关闭此对话框。
进步	观察 XML 导出的进度。

注记

- ArcGIS 在 Enterprise Architect 的专业版、企业统一版和终极版中可用
- 在 Enterprise Architect 的企业统一版和终极版中，如果启用了安全性，则必须具有“导出 XMI”权限才能导出为 XML

- 在将您的模型导出到 ArcGIS 模式之前，您必须至少定义一个空间参考元素；空间参考元素被其他模式元素通过动态链接的标记值引用，名为 `SpatialReference`
- ArcGIS包上的 `DefaultSpatialReference` 标签用于指定可应用于工作空间中所有特征数据集和特征类的空间参考；因此，您不需要对每个特征数据集或特征类应用空间参考元素
- 如果您没有从任何特征数据集或 ArcGIS模型中的特征类中引用空间参考类，Enterprise Architect默认会为这些元素生成一个未知类型为空间参考的 XML 模式

导入 ArcGIS XML 工作空间

如果您有一个地理数据库工作空间XML 文档（包含 ArcGIS 模式），您可以将它作为UML模型导入到您的 Enterprise Architect项目中。

在运行导入之前，取消选中 首选项”窗口（开始>外观>首选项>首选项）的 对象”页面上的 按字母顺序排列特征”复选框。这可确保在Enterprise Architect中以与源中相同的顺序导入和组织字段。

访问

在浏览器窗口中单击目标包。

功能区	发布 >模型交换>导入> ArcGIS
上下文菜单	右键包 特定 ArcGIS 导入ArcGIS工作空间XML

导入地理数据库工作空间XML 文档

选项	行动
文件名	类型在或浏览要导入的 ArcGIS XML 文件的名称。
创建图表	选中复选框以在导入的包下创建类图。
在图表上隐藏系统级 ArcGIS 字段	选中复选框以隐藏这些原型属性： <ul style="list-style-type: none"> 必填项目 属性索引 空间索引 在这些刻板印象的类上： <ul style="list-style-type: none"> 观点 折线 多边形 多补丁 库表（物件类）类的 'RequiredField' 和 'AttributeIndex' 属性也是隐藏的。 This option is enabled only when the 'Create图表' checkbox is selected.
移除 GUIDs	移除 GUIDs ”特征目前对于 ArcGIS 导入是强制性的，这意味着每次导入 ArcGIS 模式时都会将元素创建为 新的”。
写入日志文件	选中复选框以写入导入活动log（推荐）。 log文件保存在要导入文件的目录中，与导入文件同名加后缀_import。log。
视图XML	单击此按钮可在导入前查看 XML。
导入	单击此按钮可导入 ArcGIS XML 文件。

关	单击此按钮可关闭此对话框。
帮助	单击此按钮可显示此帮助页面。
导入进度	此字段指示导入的进度。

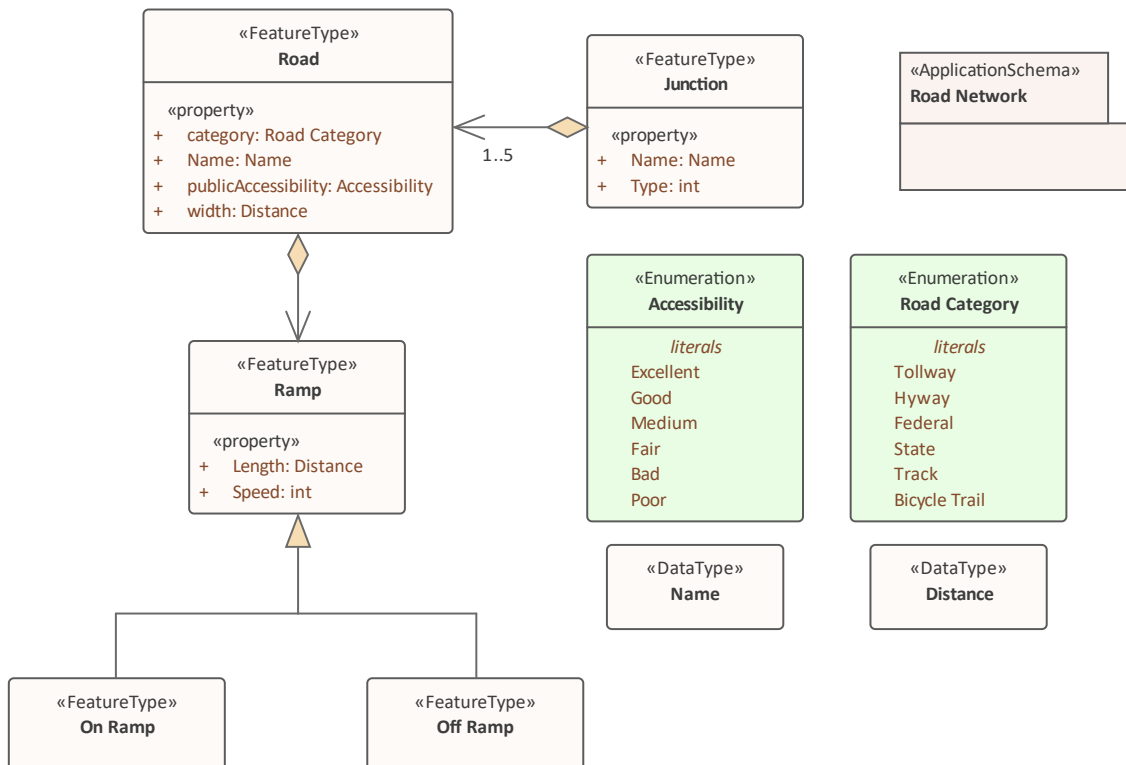
注记

- ArcGIS 在Enterprise Architect的专业版、企业统一版和终极版中可用

Geography Markup Language (GML)

模型地理特征和生成应用模式

您可以创建域中重要特征的表达性和协作模型，并使用这些模型生成符合地理标记语言 (GML) 的应用程序模式。其他应用程序可以使用这些模式。许多颠覆性力量和技术改变了我们彼此互动的方式和我们所居住的世界，涉及地理位置和特征。我们沿着公路行驶，在瞭望台停下来观看沿海特征或城市景观，我们出国旅行以观看纪念碑和建筑物，例如教堂和博物馆，我们依靠风力发电场获取能源，我们在机场起飞和降落等等。您可以使用Enterprise Architect实施的地理标记语言来模型任何感兴趣的地理特征，该语言是地理信息系统的基础，并且可以用作 Internet 上地理交易的开放交换格式。



道路的 GML 模型显示两个具有访问两个数据类型和枚举的属性的特征

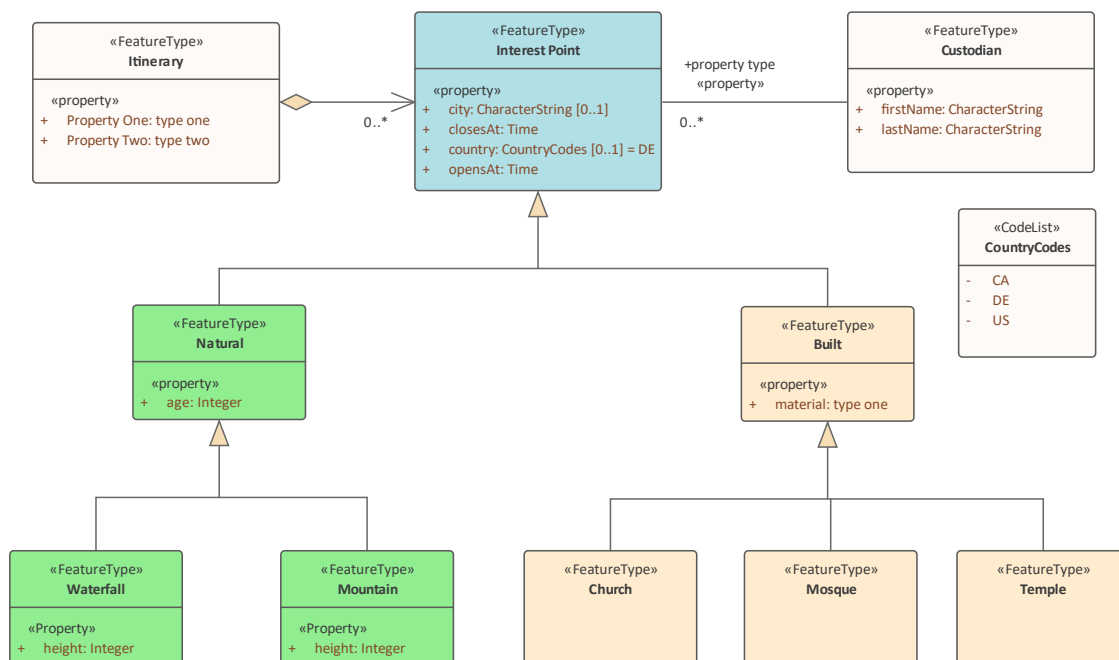
Enterprise Architect 的 GML 是 Open Geospatial Consortium 的地理标记语言 (GML) 3.3 的实现，它在版本 10 或之后的版本中为 Enterprise Architect 中的地理特征建模功能提供 XML 语法。

通过 GML，您可以：

- 为 Geography Markup Language (GML) 3.3 应用 UML 配置文件
- 利用自定义的图表类型和工具箱页面，方便有效地访问元素和与模型地理特征的关系
- 生成 GML 应用程序架构文件

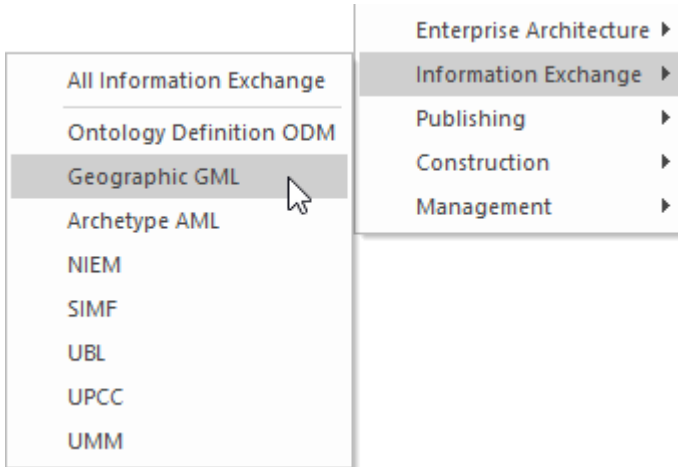
示例图表

使用地理标记语言 (GML) 功能，您可以模型基于组织或社区的应用程序模式。这些模型可用于显示特征类型之间的关系，这些特征类型包含任何数量的限定其特征的属性。这些可以基于定义的类型、数据类型、代码列表或枚举。您可以与其他地理空间同事或在管理工件的学科中从事传统系统实施工作的人员协作，这些工件包括：策略、目标、需求、数据模型、软件模型、部署描述等。



使用 GML 建模

您可以使用Enterprise Architect中的综合图表和建模功能创建 GML 模型。首先，您需要选择 GML 或 Information Exchange蓝图。蓝图是一个有用的聚焦工具功能，可确保您保持专注并专注于 GML 建模。



蓝图菜单 - GML蓝图选择

这将激活 GML 的UML配置文件，允许您使用描述您的组织或社区的元素和连接器创建模型。

访问

功能区	特定>技术> GML
上下文菜单	右键包 特定 GML

特征

特征	细节
配置文件支持	<p>通过使用以下列形式提供的内置 GML功能，您可以快速简单地开发 GML 结构：</p> <ul style="list-style-type: none"> • A GML 图表类型，可通过“新图表”对话框访问 • 工具箱中的 GML 页面，将图表概念映射到适当的原型UML元素 • “工具箱快捷菜单”和“快速链接器”中的 GML元素和关系条目
GML工具箱页面	GML工具箱页面有效地包含了模型地理特征的元素和连接器。
来自 ISO 的UML类	<p>(可选) 您可以将 ISO/TC 211 中实现的UML类下载为 XMI 文件，然后将 XMI 文件作为包含图表和标准UML类的包导入Enterprise Architect，您可以在您的模型中重复使用。</p> <ul style="list-style-type: none"> • 并非所有在 ISO/TC 211 中实现的UML类在 GML 中都有对应的映射；具有映射的类(如 GML 3.2.1 规范1所指定)在 <code>Sparx Systems > EA ></code>

	<p>Config > GML”文件夹中的可配置文件 GMLClassMapping.xml 中指定</p> <ul style="list-style-type: none">• 这些类的命名空间信息在 “Sparx Systems > EA > Config > GML 文件夹”中的可配置文件 GMLNamespaces.xml 中指定
GML 应用架构生成	<p>您在Enterprise Architect中使用 GML 创建的任何模型都可以导出为 GML 应用程序架构。</p> <p>使用 “Sparx Systems > EA > Config > GML”文件夹中的可配置文件 GMLstereotypes.xml，您可以为标准 GML 原型指定别名。GML 应用程序架构生成器也会在架构生成期间考虑这些别名。</p>

注记

- GML 在Enterprise Architect的专业版、企业统一版和终极版中可用

更多信息

通过访问以下主题，可以在更多细节中探索地理空间建模：

- [ArcGIS Geodatabases](#)
- [Geography Markup Language \(GML\)](#)

集成来自外部提供商的数据

专业云服务器可帮助您将来自外部提供商的数据集成到Enterprise Architect模型中。Enterprise Architect是一个团队合作者，通过其基于服务器的集成功能，可以帮助您创建模型，这些模型将来自各种不同工具的元素相关联，每个工具都可能包含诸如战略目标和需求之类的实体，通过实施陈述、工作包和配置项。Enterprise Architect不管理这些项目的主记录，而是充当累加器，将内容放入单个存储库并允许相关项目。A第三方提供商可以与Enterprise Architect集成，包括：

-  应用程序生命周期管理 (以前称为 HP 质量中心)
-  Jama集成
-  Jazz (与：
 - IBM Rational DOORS Next Generation 的需求管理工具
 - Rational Rhapsody设计管理 (DM)
 - Rational Team Concert更改和配置管理 (CCM)
 - 理性质量管理器 (QM))
-  Jira 和  Confluence
-  Polarion
-  Azure DevOps /团队基础服务器
-  Wrike
-  ServiceNow
-  Autodesk
-  Bugzilla
-  Salesforce
-  SharePoint
-  Dropbox 和
-  其它Enterprise Architect模型

见 [Install and Configure](#) 帮助主题以获取有关如何配置每个提供程序的信息。演练也可用于 [Walkthrough: Jira Integration](#) 和 [Walkthrough: Polarion Integration](#) 集成。

当在列表中选择来自外部供应商的项目时，该项目的元数据将显示在适当的Enterprise Architect窗口中。因此，任何属性和值类型的信息都将显示在属性窗口中，描述和评论将显示在笔记窗口中，讨论或帖子将显示在讨论和审阅窗口中。例如，如果 Jira 是外部数据源并且集成是列出用户案例，则 Jira 用户故事属性（例如优先级：中等）将出现在属性窗口中，故事描述将出现在笔记窗口中，评论将出现在讨论和审阅窗口中。

某些元数据（例如协作信息）可能不适用于所有项目和集成，但在可用的情况下，功能提供元数据的统一、交叉集成视图，使Enterprise Architect用户更容易理解数据来自多个提供商和集成，而无需离开工具或使用供应商

特定的术语。Enterprise Architect正在扮演一个累加器的角色，允许来自广泛不同来源的信息与工具中已经丰富的架构模型集相关联，创建一个视图，以了解如何可视化这些原本不相关的工具中的信息。这消除了对静态电子表格包的需要，组织传统上使用这些电子表格将成对的项目相关联，例如测试用例与业务驱动因素，或利益相关者对应用程序服务的关注，等等。

所有集成都支持将对象和元素从外部系统链接到Enterprise Architect客户端。外部数据窗口支持浏览外部提供者的项目并根据提供者的查询检索元素和对象列表。能力包括：

- 将Enterprise Architect元素链接到外部object
- 视图外部元素属性
- 视图并在某些情况下添加到外部object讨论
- 导出与当前模型相对应的WebEA URL 的链接
- 在网络浏览器中打开外部项目
- 导入元素
- 导出元素

从Enterprise Architect Release 1开始，可以将非云模型链接到在专业云服务器上配置的集成插件。

有关配置选项，请参阅云页面帮助主题。

访问

功能区	特定>工具>系统集成>打开外部数据 要查看链接到外部项目的所有元素的列表： 特定>工具>系统集成>显示所有链接项
上下文菜单	在图表或浏览器窗口中，对于已链接到外部项目的元素： 右键单击元素>特定>外部属性
键盘快捷键	Alt+1 >系统集成

特征

特征	描述
专业云服务器配置	必须在专业云服务器上配置每个外部提供程序以启用连接。可以为每个提供者进行多种配置（例如连接到两个单独的 Jira 服务器）。
授权	如果集成提供者需要授权，系统会提示您输入凭据。如果提供商支持它，将打开一个新的 Internet 浏览器窗口并提示您log集成提供者并允许Enterprise Architect访问其资源。或者，将弹出一个简单的对话框，询问您的凭据，并可选择将它们安全地存储在当前模型中。如果存储在模型中，则凭据将仅用于当前用户。
导航外部提供者	通过导航外部提供程序查找要链接到的外部项目。
外部项目详细信息	从“外部数据”窗口中选择一个外部项目以查看其属性、笔记和属性窗口、笔记窗口和讨论和审阅窗口的“讨论”选项卡中的讨论。

链接外部项	外部项目可以作为通用的 «ExternalReference» 原型或作为另一个元素类型链接到Enterprise Architect 。
添加WebEA链接	<p>WebEA的超链接可以添加到外部项目中，以便您可以从外部源快速打开WebEA元素。</p> <p>右键单击列表中的本地链接元素，然后选择“将WebEA超链接添加到外部物件”。这将使用指向在Enterprise Architect中链接到它的WebEA元素的链接更新外部项目。注记并非所有提供程序类型都有“链接”机制。如果不存在，某些提供程序可能允许将链接添加为对项目的评论。</p> <p>确保模型在模型选项中设置了有效的WebEA地址。</p>
配置	<p>每个集成提供者都带有一组默认映射值，这些映射值确定在Enterprise Architect中创建的本地元素的类型，以及将哪些字段复制到新元素。这些映射可通过“外部数据”窗口工具 菜单上的“配置”选项进行配置。</p> <p>您必须具有“配置外部数据源”权限才能访问此功能。</p>
故障排除	<ul style="list-style-type: none"> • 系统输出窗口将在尝试从外部提供者检索数据时显示任何错误 • 专业云服务器为每个外部提供者输出log文件

注记

- “集成”需要专业云服务器，并且仅适用于云模型
- “集成”目前在Enterprise Architect的企业版、统一版和终极版中提供
- 如果同时选择两个或多个元素，则可以一次对所有选定元素执行操作
- 要使WebEA链接起作用，必须为WebEA地址设置有效的URL（请参阅云页面帮助主题）

到WebEA的项目超链接

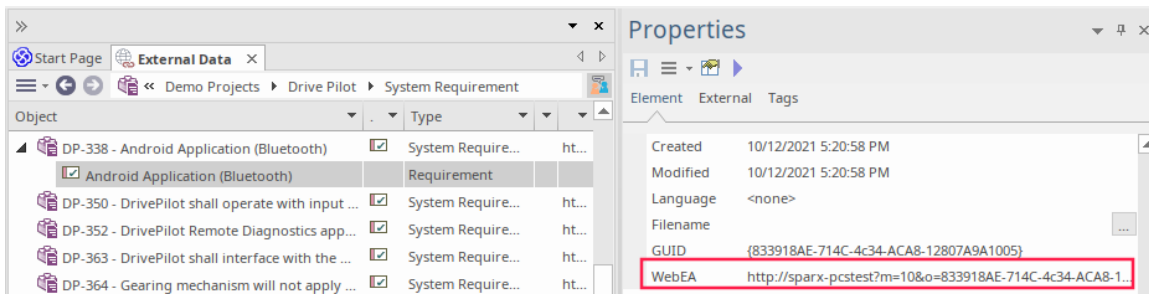
从外部项目创建的元素可以通过超链接从外部应用程序引用到WebEA上托管的元素详细信息。使用WebEA作为参考的好处包括：

- 通过网络浏览器轻松访问详细信息
- 访问该元素的最新数据
- 无需安装Enterprise Architect的本地副本

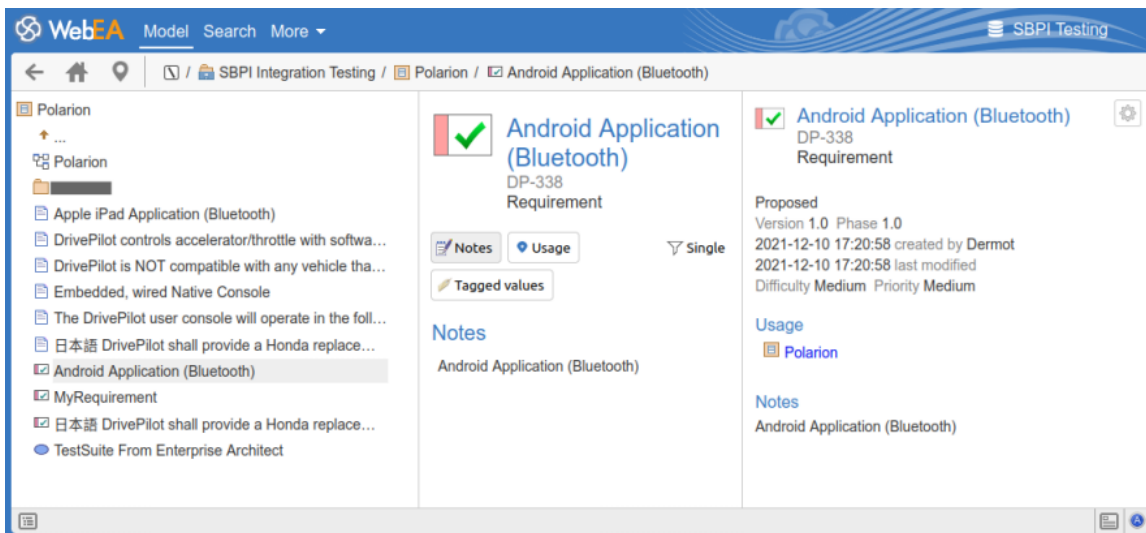
当在Enterprise Architect中创建新元素时，在外部项中设置超链接引用。例如，这里我们有一个超链接，在外部应用程序的一个项目内，指向一个元素的WebEA页面：

Role	URL
<i>external reference</i>	http://sparx-pcstest?m=10&o=BC32EA51-640F-4457-A933-E57D93952AB0

超链接引用是系统生成的，并显示在相关元素的属性窗口中：



这是从外部超链接引用的WebEA页面：



支持的

支持对WebEA的超链接引用的集成包括：

- ALM
- Azure 开发运营 (TFS)
- Confluence (附加为 “信息” 字段)

- 吉拉
- 爵士乐 (包括 Doors NG)
- 极化子
- Wrike (作为对项目的评论)

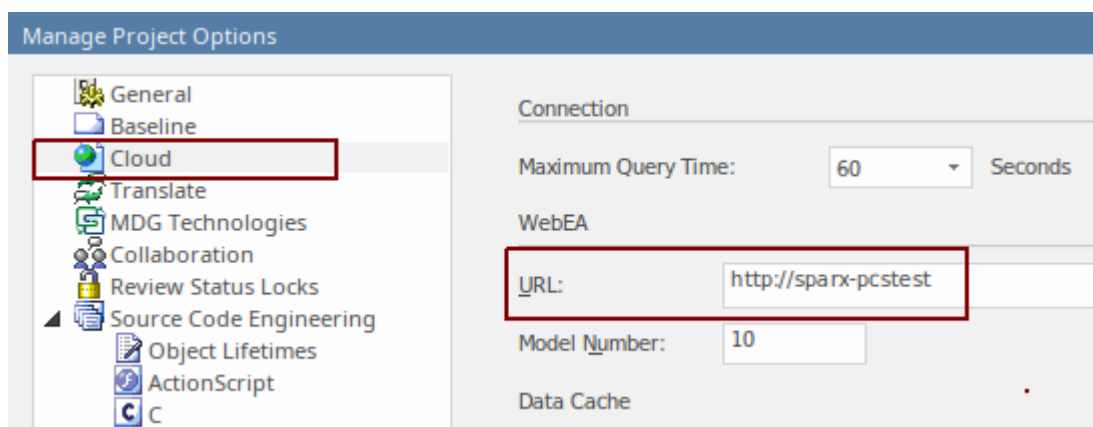
不支持的外部应用程序是：

- 投递箱
- 远程Enterprise Architect
- 立即服务

配置

该配置要求WebEA正在为该存储库运行。要启用外部超链接，您必须为该存储库设置WebEA URL 引用。基本参考定义在：

- 设置>模型>选项>云>网址



更多详情请参阅云页面帮助主题。

访问的信息

每个集成插件都基于“过滤器”或外部产品数据中的位置返回信息。一些产品，例如Enterprise Architect、Jira和DevOps，提供了一种自定义返回数据的机制，而另一些产品只是在外部提供商应用程序中的特定位置返回所有信息。

从每个提供者获取的信息

Sparx Systems开发的每个Integration插件都基于产品的外部数据中的“过滤器”或位置返回信息。有些产品（例如Enterprise Architect、Jira和TFS）提供了一种自定义返回数据的机制，而其他产品则只是将所有信息返回到数据中的特定位置。

注意：对于Enterprise Architect、Jira或TFS，必须在Enterprise Architect请求将集成插件连接到它们之前配置过滤器。

提供者	返回信息
应用程序生命周期管理器	根据缺陷、需求和测试的内部列表返回的信息。
欧特克	基于Hubs内容返回的信息 项目 文件夹。
布吉拉	根据产品内容返回的信息 部件 <组件中的所有项目>。
投递箱	根据文件夹内容返回的信息。
Enterprise Architect	当您通过集成连接到外部Enterprise Architect存储库时，在“外部数据”面包屑中，将为您提供一个带有“浏览”或“搜索”的菜单。在选择“搜索”时，它会返回本地存储库中定义的搜索列表。 选择特定搜索后，项目列表将显示来自外部存储库数据的结果。
爵士乐	根据 (DoorsNG) - 文件夹的内容返回的信息。
吉拉	显示“最喜欢的过滤器”列表。请参阅菜单选项“问题 管理过滤器”。
Salesforce	显示所有具有“列表视图”的项目类型。在默认安装中，这些包括：客户、资产、市场活动、案例、组、联系人、合同、潜在客户和机会。
立即服务	显示用户定义的过滤器列表，按它们所基于的表分组。
DevOps / 团队基础服务器	显示 DevOps / TFS 全局查询和“我的... ..”查询的列表。
写	基于Accounts内容返回的信息 文件夹。

注记

对于Enterprise Architect、Jira或DevOps，必须在Enterprise Architect请求集成插件连接到它们之前配置过滤器。

自定义 SBPI 服务

自定义 SBPI 服务是用户定义的集成插件，可以由 Enterprise Architect 的脚本或插件调用，然后可以使用响应来操作存储库数据。可以使用存储库从 Enterprise Architect 调用自定义插件。调用 SBPI 自动化接口方法。

服务的请求范围和返回给 Enterprise Architect 的范围不受限制。例如，自定义服务可以响应从第三方服务收集信息的请求，并将其返回给 Enterprise Architect 脚本。可能的用途包括通过其它调用对模型数据本身执行操作，或运行自定义构建处理模型数据。

好处

- 允许任意请求和响应
- 为模型配置后，可以从脚本或插件中调用
- 由专业云服务器自动处理的生命周期和请求转发
- 可以用多种编程语言编写，包括 C++ 或 C#

概述

要编写自己的自定义服务插件，您可以从头开始，也可以复制其中一个示例并进行修改。

安装专业云服务器时，启用“SBPI 示例”组件以包含自定义服务示例。启用后，示例文件的默认位置位于“SBPI Examples\ExampleServicePlugins”文件夹中。例如：

C:\Program Files (x86)\Sparx Systems\专业云服务器\SBPI Examples\ExampleServicePlugins

有关详细信息，请参阅[Pro Cloud Server Installation](#)帮助主题。

注记：默认情况下不启用“SBPI 示例”安装选项。如果您已经安装了没有“SBPI 示例”的专业云服务器，您可以执行完全重新安装（启用“SBPI 示例”），或使用安装程序的“更改”选项仅添加“SBPI 示例”零件。

插件可以用 C++ 或 C# 编写。

这些示例是使用 Visual Studio 2017 编写的，但这不是先决条件。

自定义服务插件必须实现 ISBPIServicePlugin 中定义的接口，该接口包含在 ISBPIServicePlugin.h（用于 C++）或 ISBPIServicePlugin.cs（用于 C#）中。

程序的一般流程是：

- 用户通过插件自动化接口在 Enterprise Architect 中执行需要来自服务插件的信息的操作
- 插件收到请求
- 插件解析请求并执行所需的任何操作（例如，调用第三方服务、运行程序、对专业云服务器进行 OSLC 调用）
- 插件通过提供的回调函数向 Enterprise Architect 发送响应；这可以是请求的实际数据或错误值
- Enterprise Architect 接收回调数据并在脚本或插件中使用它

接口

函数/类	细节
创建插件（在 C# 中不需要）	插件必须实现这个导出函数： <pre>extern "C" SBPI_SERVICE_API SBPI_SERVICE_PLUGIN CreatePlugin();</pre> 它必须返回一个指向实现 ISBPIServicePlugin 接口的类的指针。推荐的实现

	<p>是：</p> <pre>SBPI_SERVICE_PLUGIN CreatePlugin() { 返回新的 ExampleServicePlugin ; } </pre> <p>新创建的 ISBPIServicePlugin 在接收到 ISBPIServicePlugin::Release 方法时可以被删除。</p>
ISBPIServicePlugin 接口	dll插件必须实现 ISBPIServicePlugin 接口中的所有方法。

ISBPIServicePlugin 接口

ISBPIServicePlugin 接口方法

方法名称	注记
发布	仅限 C++。当不再需要接口类时，它由控制应用程序调用。该方法应该删除在 CreatePlugin()函数期间创建的类。
设置回调	仅限 C++。传入一组回调函数指针，稍后插件使用这些回调函数指针将数据传回Enterprise Architect。
设置配置	接收用户在专业云服务器中设置自定义服务插件时定义的设置。
处理请求	来自Enterprise Architect的一般请求。该插件可以执行任何所需的操作，并通过回调方法将数据或错误代码返回给Enterprise Architect。

处理请求

输入

方法	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 要执行的方法的名称。这允许插件选择要执行的操作，而不必先完全解析参数string。
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string。有关可用参数的更多信息，请参阅详细信息。

通过回调输出

[可选] Result、LogMessage 或 SetError - 提供用户反馈。

细节

这是函数的主要功能。它应该处理所有传入的“方法”请求并执行所需的操作，并通过“结果”回调返回任何数据。

示例实现

```
void ExampleServicePlugin::HandleRequest(const char* 方法, const char* 参数)
{
    LogMessage(LOG_TRACE, std::string(__FUNCTION__ + " - Method = "s + method).c_str());

    // 这个示例方法演示了如何从参数中提取各种参数类型。
    // 这是使用 jsoncpp 库完成的，但可以使用任何兼容的 JSON 库完成。
    Json::Value jsonParameters;
    如果 ( strlen ( 参数 ) )
    {
        std::stringstream(参数) >> jsonParameters;
    }

    if (std::string (method) == "DoSomething")
    {
        int myNumber = jsonParameters["myNumber"].asInt();
    }
}
```

```
双 myFloat = jsonParameters["myFloat"].asDouble();
std::string myString = jsonParameters["myString"].asString();
std::list<int> myArrayOfNumbers;
for (auto& myValue : jsonParameters["myArrayOfNumbers"])
{
    myArrayOfNumbers.push_back(myValue.asInt());
}
std::list<std::string> myArrayOfStrings;
for (auto& myValue : jsonParameters["myArrayOfStrings"])
{
    myArrayOfStrings.push_back(myValue.asString());
}

std::string result = "示例用户SBPI C++ 服务插件。DoSomething 接收参数：myNumber = " +
std::to_string(myNumber)
    + ", myfloat = " + std::to_string(myFloat)
    + ", 我的字符串 = " + 我的字符串;

// 设置结果string 。
结果(result.c_str());
}
else if (std::string (method) == "DoSomethingToElement")
{
    // 这个示例方法演示了如何响应用户在单个元素上执行任务。
    std::string elementID = jsonParameters["elementGUID"].asString();

    Result(std::string ("C++ 中的示例用户SBPI 插件。DoSomething to元素GUID : " + elementID).c_str());
}
else
{
    // 设置错误string
    SetError(std::string ("未知方法：" + std::string (method)).c_str());
}
}
```

设置回调

仅限 C++。这只需要在 C++ 中实现。示例中的代码就足够了，不需要修改。可以添加额外的错误或边界检查。

输入

参数	细节
<code>const void ** 回调函数</code>	用于将数据传回Enterprise Architect的回调函数指针数组。

通过回调输出

没有任何

细节

C++插件会在创建后很快收到这个方法。它传入一个回调函数指针数组，稍后插件使用这些回调函数指针将数据传回Enterprise Architect

示例实现

```
void ExampleServicePlugin::SetCallbacks(const void ** callbackFunctions)
{
    如果 ( 回调函数 )
    {
        结果 = (ResultPtr)callbackFunctions[0];
        SetErrorCode = (SetErrorCodePtr)callbackFunctions[ 1 ];
        SetError = (SetErrorPtr)callbackFunctions[2];
        LogMessage = (LogMessagePtr)callbackFunctions[3];
    }
}
```


设置配置

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。有关可用参数的更多信息，请参阅详细信息。

通过回调输出

[可选] LogMessage - 设置有关收到的配置设置的log消息。注意不要log敏感信息。

注记：此方法将忽略 SetError 回调。

细节

此方法接收用户在启用此自定义集成插件时输入到专业云服务器配置中的详细信息。

它包括以下详细信息：

- 在“自定义属性”部分输入的自定义项/值对设置。

代理设置：

- 代理服务器
- 代理绕过
- 代理用户名
- 代理密码

示例实现

```
void ExampleServicePlugin::SetConfiguration(const char* 参数)
{
    LogMessage(LOG_TRACE, std::string (__FUNCTION__).c_str());

    Json::Value jsonParameters;
    如果 ( strlen ( 参数 ) )
    {
        std::stringstream(参数) >> jsonParameters;
    }

    for (auto& myProperty : jsonParameters.getMemberNames())
```

```
{  
m_properties[myProperty] = jsonParameters[myProperty].asString();  
}  
}
```

示例脚本

此JavaScript脚本详细说明了如何向自定义服务插件发送简单请求：

！ INC 当地的 脚本 。 EAConstants - JavaScript

```
/*
*脚本名称：Custom Service示例
*作者：Sparx Systems
*目的：演示如何使用自定义服务插件的 SBPI 自动化接口
*日期：2022-02-28
*/

// 使用一些参数向插件发送一个简单的请求。
函数 简单请求()
{
    // 显示脚本输出窗口
    存储库。确保输出可见(《脚本》);
    Session.输出(“JavaScript自定义插件示例”);
    会话.输出输出“

=====

// 通过使用 InsertSBPIParameter 添加参数，随请求一起发送数据。
变量 打包参数 = “;

// 带有额外参数的可选数据发送
打包参数 = 存储库。InsertSBPIParameter (packedParameters, '我的号码', 25);
打包参数 = 存储库。InsertSBPIParameter (packedParameters, '我的浮动', 123.456);
打包参数 = 存储库。InsertSBPIParameter (packedParameters, '我的字符串', '你好世界');

Session.输出("向'DoSomething'方法发送简单的插件请求");
变量 回复 = SBPIRequest ('csvc', '做某事', 打包参数);
}

// 帮助函数向自定义插件发送请求并检查错误。
函数 SBPIRequest (前缀, 方法, 打包参数)
{
    // 指定插件的前缀。这是在Pro云客户端中配置的。
    变量 回复 = 存储库。CallSBPI (前缀, 方法, 打包参数);
    如果 ( 回复 == "
    {
```

```
        Session.输出('插件错误:' + 存储库。GetLastError ());
    }
    else
    {
        会话.输出('成功:' + 回应) ;
    }

    返回 回应 ;
}

函数 主要 ( )
{
    // 使用一些参数向插件发送一个简单的请求。
    简单请求 ( ) ;
}

主要 ( ) ;
```

编写自定义集成插件

我们生活在一个高度互联的世界中，虽然专业云服务器提供了大量开箱即用的集成，但可以使用任何具有标准 Web 服务接口的产品创建自定义（专有）集成。此功能将向外部工具开放 Enterprise Architect 存储库的内容，并使外部工具中的信息在 Enterprise Architect 中可用。例如，项目管理工具可以定义有助于在 Enterprise Architect 中可视化的工作页面，或者自动化测试工具可以定义与 Enterprise Architect 中的实现和规范元素相关的测试用例和测试过程。这将需要一些技术专业知识和使用多种编程语言（如 C++ 或 C#）中的一种来创建集成，但这只需完成一次，并且可以在任意数量的存储库中使用。

专业云服务器和 Enterprise Architect 将完成繁重的工作，管理员无需更改专业云服务器的任何基本安全设置，因为新的集成将通过现有端口和防火墙运行。开发者也无需编写 http 监听代码，让他们聚焦于确定和配置 RESTful API 调用以将外部项目的信息传入和传出服务器。

安装专业云服务器时，启用“SBPI 示例”组件以包含自定义集成示例。启用后，示例文件的默认位置位于“SBPI Examples\ExampleIntegrationPlugins”文件夹中。例如：

C:\Program Files (x86)\Sparx Systems\专业云服务器\SBPI Examples\ExampleIntegrationPlugins

有关详细信息，请参阅 [Pro Cloud Server Installation](#) 帮助主题。

注记：默认情况下不启用“SBPI 示例”安装选项。如果您已经安装了没有“SBPI 示例”的专业云服务器，您可以执行完全重新安装（启用“SBPI 示例”），或使用安装程序的“更改”选项仅添加“SBPI 示例”零件。

要编写自己的自定义集成插件，您可以从头开始，也可以复制其中一个示例并进行修改。插件可以用 C++ 或 C# 编写。

这些示例是使用 Visual Studio 2017 编写的，但这不是先决条件。

自定义集成插件必须实现 ISBPIIntegrationPlugin 中定义的接口，该接口包含在 ISBPIIntegrationPlugin.h（对于 C++）或 ISBPIIntegrationPlugin.cs（对于 C#）中。

程序的一般流程是：

- 用户在 Enterprise Architect 中执行需要来自集成插件的信息的操作
- 插件将接收到对适当接口方法的调用（或多次调用）
- 插件解析请求，并在需要时向实际数据提供者发出自己的请求
- 插件接收来自实际提供者的结果，并解析数据
- 插件通过提供的回调函数向 Enterprise Architect 发送响应；这可以是请求的实际数据或错误值
- Enterprise Architect 接收回调数据并显示给用户

函数/类	细节
检查版本	<p>（在 C# 中不需要。）</p> <p>输入：无符号 int 版本</p> <p>如果您的插件支持请求的版本，则返回 true。</p> <p>版本 2 在 Enterprise Architect 中的元素链接或取消链接到外部项目或修改它们时添加通知。</p> <p>版本 2 扩展版本 1，因此对于支持的版本（包括支持的版本）返回 true。</p> <p>例如返回（版本 <= 2）；</p>
创建插件	<p>（在 C# 中不需要。）</p> <p>插件必须实现这个导出函数：</p> <pre>extern "C" SBPI_API SBPI_INTEGRATION_PLUGIN CreatePlugin();</pre> <p>它必须返回一个指向实现 ISBPIIntegrationPlugin 接口的类的指针。推荐的实现是：</p> <pre>SBPI_INTEGRATION_PLUGIN CreatePlugin() {</pre>

	<pre>返回新的 ExampleIntegrationPlugin ; } 新创建的 ISBPIIntegrationPlugin 在接收到 ISBPIIntegrationPlugin::Release 方法 时可以被删除。</pre>
接口	dll插件必须实现 ISBPIIntegrationPlugin 接口中的所有方法。

ISBPIIntegrationPlugin 接口

ISBPIIntegrationPlugin 接口提供了一系列方法。

接口方法

大多数方法都有一个string参数 (C++ 中的 char* , C# 中的string) , 它被编码为 JSON 以提供各种参数。

ISBPIIntegrationPlugin 接口提供以下方法：

方法名称	注记
创建WebEA超链接	在外部项目中添加指向WebEA的链接。
通用请求	保留供将来使用。
获取授权详情	返回您的提供者需要的授权方法。支持的类型有：none、basic 和 OAuth。
获取默认字段映射	返回外部项目字段和属性如何映射到Enterprise Architect元素字段、属性和标记值的列表。
获取默认类型映射	返回外部项类型如何映射到Enterprise Architect元素类型的列表。
获取字段	返回外部提供者中已知字段/属性的列表。
获取图标	返回外部提供者的图标。
获取项目	返回所选项目的完整属性。
获取项目讨论	在主讨论窗口中的“外部”选项卡中填充对选定外部项目的评论和讨论。
获取项目列表	在收到为所选菜单级别填充“外部数据”窗口中的项目列表的请求时，此方法会填写这些项目。
GetItemNotes	使用所选外部项目的详细信息填充主注记窗口中的“外部”选项卡。
获取项目	返回所选项目的完整属性。
获取菜单列表	收到在“外部数据”窗口中填充菜单的请求，此方法将填充菜单项。
获取类型	返回外部提供者中已知类型的列表。
项目链接	(需要Enterprise Architect Release 15.2 build 1559 及更高版本。) 外部项目已链接到Enterprise Architect中的元素的通知。
项目未链接	(需要Enterprise Architect Release 15.2 build 1559 及更高版本。) 外部项目已与Enterprise Architect中的元素取消链接的通知。
物品已更新	(需要Enterprise Architect Release 15.2 build 1559 及更高版本。) Enterprise Architect中链接的元素已更新或修改的通知。

后新讨论	向外部项目添加新评论或讨论。
PostNewItem	在外部提供者中创建一个新项目。
PostOAuthCode	发出访问令牌请求以交换 OAuth 访问令牌的代码（使用 OAuth 令牌端点）。使用 AddProperty 返回 accessToken 和 refreshToken。
PostUpdateItem	更新外部提供者中的选定项目。
PostUpdateItemNotes	更新外部提供者中所选项目的笔记。
刷新OAuthToken	针对 OAuth 刷新端点执行刷新请求并返回新的 accessToken 和 refreshToken。
发布	仅限 C++。当不再需要接口类时，控制应用程序会调用此方法。该方法删除在 CreatePlugin() 函数期间创建的类。
设置授权	接收用户在 Enterprise Architect 中输入的授权信息。
SetCallbacks(const void ** callbackFunctions)	仅限 C++。传入一组回调函数指针，稍后插件使用这些回调函数指针将数据传回 Enterprise Architect。
设置配置	接收用户在专业云服务器中设置自定义集成插件时定义的设置。

创建WebEA超链接

在外部项目中添加指向WebEA的链接。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++：常量字符*• C#：string 参数A JSON string。包括要添加的WebEA链接的详细信息。

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收到外部项目的WebEA链接的请求。

一些供应商允许将外部链接添加到项目中，而其他供应商可能只允许将其添加为评论或注册部分。

输入参数string包含有关要创建的链接的信息，包括：

- "itemID" - 要更新的项目的 ID
- "webEALink" - WebEA链接的完整 URL

注意WebEA注册有效，设置 >模型> 选项 >云“功能区选项卡必须指定有效的WebEA基本链接。

获取授权详情

返回您的提供者需要的授权方法。

支持的类型有：

- 没有任何
- 基本的
- 身份验证

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++：常量字符* • C#：string 参数A JSON string。这是一个空string，保留供将来使用。

通过回调输出

- [可选] AddProperty - 指定可用的授权方法和这些方法的属性
- [可选] LogMessage 或 SetError - 提供用户反馈。

细节

对于没有授权，在这个方法中什么也不做。

Enterprise Architect仅支持 OAuth 2.0 授权码授予类型。

OAuth 授权将Enterprise Architect设置为提示以打开浏览器页面供用户log到 OAuth 提供程序（“授权端点 URI”）。

OAuth 提供者将向Enterprise Architect发回一个“代码”。Enterprise Architect然后将使用新代码调用 PostOAuthCode。

注记：如果指定了 OAuth 则优先，不允许基本授权。

OAuth 需要 OAuth 提供者的 authorizationEndpointURI 和 redirectURI。

对于Enterprise Architect，redirectURI 必须是“http://localhost:8888/oauth/callback”。

这通常需要作为允许的重定向 URI 添加到您的 OAuth 提供程序中。

基本授权会将Enterprise Architect设置为提示用户名和密码，这些用户名和密码将在每个后续请求中传回。

示例实现

基本授权：

```
AddProperty (索引, 基本, 真);
```

认证：

```
AddProperty(index, "OAuthConfiguration", "true");
```

```
AddProperty(index, "authorizationEndpointURI", " https ");
```

```
AddProperty(index, "redirectURI", "http:// localhost :8888/oauth/callback");
```

获取默认字段映射

返回外部项目字段和属性如何映射到Enterprise Architect元素字段、属性和标记值的列表。

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。保留供将来使用。

通过回调输出

AddProperty - 每个字段映射 2-3 个值。为每个映射使用唯一索引值：

- AddProperty(index, "externalField", "name");
- AddProperty (索引 , 内部字段" , 名称") ;
- [可选] AddProperty(index, "名称", "My External Name");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法返回的值用于填充外部数据配置中的字段映射。

该方法定义了外部项的字段/属性在链接外部项时如何映射到Enterprise Architect元素字段/属性/标记值的默认值。

“externalField”值应 GetFields 中指定的字段 ID 以及 GetItems 返回的字段名称相匹配。

“internalField”值应 Enterprise Architect字段名称匹配。

internalField 名称的选项有：

- '名称'
- '别名'
- '作者"
- 《注记》
- '标记值' - 这是一种特殊情况，需要 AddProperty(index, "taggedValue", "tagged value name") 指定要使用的标记值的名称
- '状态'
- '版本'
- '相'
- '关键字'
- '复杂'
- '范围'
- '多样性'

- '分类器'
- '抽象的'
- '是叶子'
- '语'
- '文件名'
- '是根'
- '是规范'
- '坚持'

示例：

- `AddProperty(index, "externalField", "modifiedDate");`
- `AddProperty(index, "internalField", "标记值");`
- `AddProperty(index, "taggedValue", "修改日期");`

获取默认类型映射

返回外部项类型如何映射到Enterprise Architect元素类型的默认列表。

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。保留供将来使用。

通过回调输出

AddProperty - 每个类型映射 2-4 个值。为每个映射使用唯一索引值：

- AddProperty(index, "externalType", "需求");
- AddProperty(index, "internalType", "需求");
- [可选] AddProperty(index, "stereotype", "document");
- [可选] AddProperty(index, "toolbox", "扩展::需求");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法返回的值用于填充外部数据配置中的类型映射。它定义了链接项目时外部项目的类型如何映射到Enterprise Architect元素类型的默认值。该映射可由Enterprise Architect中的最终用户通过编辑外部数据配置中的类型映射进行配置。

'externalType' 值应匹配 GetTypes 中指定的用户友好类型名称，以及由 GetItems 等返回的类型。

'internalType' 值应与Enterprise Architect类型名称相匹配。

或者，指定此类型来自哪个工具箱页面。这将在用户手动编辑映射时显示，使他们可以轻松地从同一工具箱页面选择另一种类型。

始终指定默认映射类型。如果找不到匹配的映射，将使用它。

- 添加属性（索引，外部类型，默认）；
- AddProperty(index, "internalType", "需求");

可选的刻板印象：

- AddProperty(index, "stereotype", "document");

指定工具箱页面以允许用户将映射配置更改为来自同一工具箱页面的类型。

- AddProperty(index, "工具箱", "扩展::需求");

如果需要，可以指定其它映射。例如：

- AddProperty(index, "externalType", "特征");
- AddProperty(index, "internalType", "特征");
- AddProperty(index, "工具箱", "扩展::需求");

获取字段

返回外部提供者中已知字段和属性的列表。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。保留供将来使用。

通过回调输出

AddProperty - 每个类型映射 2 个值。使用每个映射的唯一索引值：

- AddProperty (索引, "id", 类型) ;
- AddProperty(index, "名称", "类型");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

大多数系统都有一个 ID 或非用户友好的名称，以及一个用户友好的显示名称。Enterprise Architect 只会识别此处返回的字段。

此处返回的值将与 DefaultFieldMapping 和 GetItem 中返回的类型进行匹配。

如果可能，应该通过向外部提供者查询其已知字段的列表来动态创建此字段列表。对于不提供字段列表的提供者，可以对字段列表进行硬编码。

获取图标

返回外部提供者的图标。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++：常量字符*• C#：string 参数A JSON string。保留供将来使用。

通过回调输出

- `AddBinaryProperty` - 图标的二进制数据。
- [可选] `LogMessage` 或 `SetError` - 提供用户反馈。

细节

指定要在Enterprise Architect外部数据中显示的图标。返回的图标应为 png 格式，20x16 像素，左侧 4 列像素为空白且透明。

位深度为 32 以包含 Alpha 通道。

- `AddBinaryProperty(index, <指向二进制数据数组的指针>, iconBytesSize);`

获取项目

返回所选项目的完整属性。

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。包括 : <ul style="list-style-type: none"> • itemID - 项目的唯一 ID (在 GetItemList 中传回) • level - 请求菜单的层次结构 (从1开始表示第一级) • currentID, currentName - 当前菜单级别的 id 和名称 • levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID • levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name

通过回调输出

AddProperty - 每个菜单项多次调用 :

- AddProperty (索引 · “id” · “item1”) ;
- AddProperty(index, "name", "Item 1 ");

可选的额外属性。这些可以是自定义的 propertyID:propertyValue 对，例如 :

- AddProperty(index, "注记", " 1项示例注记");
- AddProperty(index, "modifiedDate", "2022-04-06T11:33:44");
- AddProperty(index, "type", "需求");
- AddProperty(index, "url", "http://example.com/item/ 1 ");
- AddProperty(index, "propertyX", "Item 1属性X");
- AddProperty(index, "propertyY", "Item 1属性Y");
- AddProperty(index, "propertyZ", "Item 1属性Z");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收请求以使用有关此外部项目的详细信息填充主属性列表中的“外部”选项卡。

该方法应通过调用 AddProperty 回调方法填写项目的属性。

每个项目需要两次调用 AddProperty，并且可以接受额外的可选调用来指定自定义属性。此调用不需要“索引”值。将其设置为 0。

强制属性 :

- `id` - 传入的 `id` 指定代表此菜单项的唯一 `id`；插件可以在后续调用中接收到这个 `id`（例如在请求下一个子菜单级别时）
- `name` - 菜单项的用户可显示名称

可选属性：

可选属性可以是任何 `propertyID:propertyValue` 对。对于要在属性窗口中列出的属性，属性名称必须与属性方法返回的字段值匹配。

获取项目讨论

返回所选项目的讨论和评论。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。包括： <ul style="list-style-type: none">• itemID - 项目的唯一 ID (在 GetItemList 中传回)

通过回调输出

AddProperty - 每个评论需要 3 次调用。每个评论使用一个唯一的索引值：

- AddProperty(index, " Person ", "示例1 ");
- AddProperty(index, "createdDate", "2022-04-06T12:34:56");
- AddProperty(index, "comment", "第1 Person对第1项的示例评论");
- [可选] AddProperty(index, "id", "0002"); // 见下文注记
- [可选] AddProperty(index, "parentID", "0001"); // 见下文注记

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收请求，在主讨论窗口中的“外部”选项卡中填充有关此外部项目的评论/讨论。

如果添加了“id”属性，则如果用户回复特定评论，这将被传回，从而允许线程讨论。

“parentID”将链接此评论作为对具有 parentID 的父评论的回复

获取项目列表

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。包括 : <ul style="list-style-type: none"> • level - 请求菜单的层次结构 (从1开始表示第一级) • currentID, currentName - 当前菜单级别的 id 和名称 • levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID • levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name

通过回调输出

[可选] AddProperty - 每个项目多次调用 :

- AddProperty (索引 , “id” , “myitemid”) ;
- AddProperty(index, "name", "示例3");
- AddProperty(index, "modifiedDate", "2022-04-06T11:33:44");
- AddProperty(index, "type", "需求");
- AddProperty(index, "url", "http://example.com/item/3");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收请求以填充所选菜单级别的外部数据窗口中的项目列表。此方法应通过调用 AddProperty 回调方法填充项目。

每个项目都需要五次调用 AddProperty 来填写列表中项目的详细信息。并且每个项目必须使用唯一的“索引”值作为第一个参数。

- id - 传入的 id 指定代表该项目的唯一 id ; 插件可以在后续调用中接收到这个 id (例如在请求项目详细信息或注册时)
- name - 项目的用户可显示名称
- modifiedDate - 上次修改项目的日期/时间 , 格式为 : YYYY-MM-DDTHH:MM:SS
- type - 外部提供者理解的项目类型 ; 例如缺陷、任务、需求、文档
- url - 将用户直接带到该项目的 url ; 企业架构师在选择 “在浏览器窗口中打开外部项目”时使用此项

不要为空白响应返回错误。干脆什么都不做。

GetItemNotes

返回所选项目的笔记或说明。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。包括： <ul style="list-style-type: none">• itemID - 项目的唯一 ID (在 GetItemList 中传回)

通过回调输出

- AddProperty - 注记- 项目注记的文本值；允许使用一些基本的 HTML 标记
- [可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收一个请求，以使用有关此外部项目的详细信息填充主注记窗口中的“外部”选项卡。

获取项目

返回请求项目的完整属性。

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。包括 : <ul style="list-style-type: none"> • itemIDs - 以逗号分隔的项目 ID 列表 • level - 请求菜单的层次结构 (从1开始表示第一级) • currentID, currentName - 当前菜单级别的 ID 和名称 • levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID • levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name

通过回调输出

AddProperty - 每个菜单项多次调用 :

- AddProperty (索引 · “id” · “item1”) ;
- AddProperty(index, "name", "Item 1 ");

可选的额外属性。这些可以是自定义的 propertyID:propertyValue 对，例如 :

- AddProperty(index, "注记", " 1项示例注记");
- AddProperty(index, "modifiedDate", "2022-04-06T11:33:44");
- AddProperty(index, "type", "需求");
- AddProperty(index, "url", "http://example.com/item/ 1 ");
- AddProperty(index, "propertyX", "Item 1属性X");
- AddProperty(index, "propertyY", "Item 1属性Y");
- AddProperty(index, "propertyZ", "Item 1属性Z");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

返回项目列表的属性。推荐：如果提供者有查询项目列表的方式，那么建议运行单个查询并返回结果。如果提供者无法查询多个项目，则运行多个单独的 `GetItem` 调用并连接结果。

返回的值应该与 `GetItem` 相同，但为每个不同的项目指定一个唯一的 索引“值”。

获取菜单列表

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++ : 常量字符* • C# : string 参数A JSON string 。包括 : <ul style="list-style-type: none"> • level - 请求菜单的层次结构 (从1开始表示第一级) • currentID, currentName - 当前菜单级别的 ID 和名称 • levelXID - 层次结构中所有先前级别的菜单级别 X 的 id ; 即level1ID · level2ID • levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name

通过回调输出

- AddProperty - 每个菜单项 2 次调用 : “id” 、 “name” 。
- [可选] LogMessage 或 SetError - 提供用户反馈 。

细节

此方法接收在 “外部数据” 窗口中填充菜单的请求。菜单层次结构中的每个级别都会触发一个新请求，并且参数将包括所请求的级别。

此方法应通过调用 AddProperty 回调方法填充菜单项。

菜单中的每个项目都需要两次调用 AddProperty。每个项目必须使用唯一的 “索引” 值作为第一个参数。

- id - 传入的 id 指定代表此菜单项的唯一 id ; 插件可以在后续调用中接收到这个 id (例如在请求下一个子菜单级别时)
- name - 菜单项的用户可显示名称

不要为空白响应返回错误。干脆什么都不做。

获取类型

返回外部提供者中已知类型的列表。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++：常量字符*• C#：string 参数A JSON string。保留供将来使用。

通过回调输出

AddProperty - 每个类型映射 2 个值。使用每个映射的唯一索引值：

- AddProper, "id", "工件");
- 工件 · 名称" · 索引") ;

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

大多数系统都有一个 ID 或非用户友好的名称，以及一个用户友好的显示名称。Enterprise Architect只会识别此处返回的类型。

此处返回的值将与 DefaultTypeMapping 和 GetItem 中返回的类型进行匹配。

如果可能，应该通过向外部提供者查询其已知类型的列表来动态创建此类型列表。对于不提供类型列表的提供者，可以对类型列表进行硬编码。

项目链接

(需要Enterprise Architect Release 15.2 build 1559 及更高版本。)

外部项目已链接到Enterprise Architect中的元素的通知。

输入

参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。 <ul style="list-style-type: none">• itemID - 项目的唯一 ID (在 GetItemList 中传回)
----	---

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

每当外部项目链接到Enterprise Architect中的元素时，都会调用此方法 - 创建新元素或链接到现有元素。

此方法是 ISBPIIntegrationPlugin 接口版本 2 的一部分。对于 C++插件，请确保 CheckVersion 正确处理版本 2。

项目未链接

(需要Enterprise Architect Release 15.2 build 1559 及更高版本。)

外部项目已与Enterprise Architect中的元素取消链接的通知。

输入

参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。 <ul style="list-style-type: none">• itemID - 项目的唯一 ID (在 GetItemList 中传回)• eaElementGUID - Enterprise Architect元素的GUID
----	---

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

每当Enterprise Architect中的元素与外部项目取消链接时，都会调用此方法。这可能是由于元素被删除，或者通过菜单选项“与外部物件断开连接”。

此方法是 ISBPIIntegrationPlugin 接口版本 2 的一部分。对于 C++插件，请确保 CheckVersion 正确处理版本 2。

物品已更新

(需要Enterprise Architect Release 15.2 build 1559 及更高版本。)

Enterprise Architect中链接的元素已更新或修改的通知。

输入

参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。 <ul style="list-style-type: none">• itemID - 项目的唯一 ID (在 GetItemList 中传回)
----	---

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

每当修改Enterprise Architect中的链接元素时，都会调用此方法；例如，名称更改、属性修改、注记更新。此方法是 ISBPIIntegrationPlugin 接口版本 2 的一部分。对于 C++插件，请确保 CheckVersion 正确处理版本 2。

后新讨论

向外部项目添加新评论/讨论

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string A JSON 参数string , 包括要添加的新评论的详细信息。

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收在外部项目中创建新讨论/评论的请求。输入参数string包含有关要创建的评论的信息，包括：

- "itemID" - 要更新的项目的 ID
- 作者"
- 评论"
- [可选] parentID - 使用线程评论时父评论的 ID；这是在 GetItemDiscussion 中传回的 ID。

PostNewItem

在外部提供者中创建一个新项目。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string 。包括要创建的新项目的详细信息。

通过回调输出

AddProperty - 返回新创建项目的 ID 。 Enterprise Architect不会认为创建成功，除非返回一个有效的 ID 。

- AddProperty(0, "id", "itemX");

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收在外部提供者中创建新项目的请求。输入参数string包含有关要创建的项目的信息，包括：

- 标题”
- 类型”
- 刻板印象”
- 描述”

PostOAuthCode

输入

参数	细节
代码	<ul style="list-style-type: none">• C++：常量字符*• C#：string 授权成功后返回给用户的OAuth码值。

通过回调输出

- AddProperty - 用于 OAuth 的 “accessToken”
- [可选] AddProperty - OAuth 的 “refreshToken”
- [可选] LogMessage 或 SetError - 提供用户反馈。

细节

用户成功登录 OAuth 提供程序后，Enterprise Architect 将使用 OAuth '代码' 调用此方法。发出访问令牌请求以交换 OAuth 访问令牌的代码（使用 OAuth 令牌端点）。

使用 AddProperty 返回 accessToken 和 refreshToken。

PostUpdateItem

更新外部提供者中的选定项目。可以通过 `PostUpdateItemNotes` 方法更新所选项目的注记。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++ : 常量字符*• C# : string 参数A JSON string。包括要更新的项目的详细信息。

通过回调输出

[可选] `LogMessage` 或 `SetError` - 提供用户反馈。

细节

此方法接收更新外部提供者中的项目的请求。输入参数string包含有关要创建的项目的信息，包括：

- "itemID" - 要更新的项目的 ID
- 标题"
- 类型"
- 刻板印象"
- 描述"

注记：新值实际上可能与当前值没有什么不同。

PostUpdateItemNotes

更新外部提供者中所选项目的注记。

输入

参数	细节
参数	<ul style="list-style-type: none">• C++：常量字符*• C#：string 包含要更新的项目详细信息A JSON 参数string。

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收更新外部提供者中项目的注记的请求。输入参数string包含有关要创建的项目的信息，包括：

- "itemID" - 要更新的项目的 ID
- 《注记》

刷新OAuthToken

输入

参数	细节
参数	<ul style="list-style-type: none">• C++：常量字符*• C#：string 用于获取新访问令牌的 OAuth 刷新令牌。

通过回调输出

- AddProperty - 用于 OAuth 的 “accessToken”
- [可选] AddProperty - OAuth 的 “refreshToken”
- [可选] LogMessage 或 SetError - 提供用户反馈。

细节

如果此处的任何方法通过 SetErrorCode 返回 401（未授权），则 Enterprise Architect 将尝试使用刷新令牌（如果已提供）并调用此方法。

针对 OAuth 刷新端点执行刷新请求并返回新的 accessToken 和 refreshToken。

设置授权

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++：常量字符* • C#：string 参数A JSON string。可用字段有： <ul style="list-style-type: none"> • 用户名 • 密码 • accessToken - OAuth 访问令牌

通过回调输出

[可选] LogMessage 或 SetError - 提供用户反馈。

细节

此方法接收用户在Enterprise Architect中输入的授权信息。

它由用户名：密码凭据或 OAuth 访问令牌组成。

Enterprise Architect将在每个会话开始时调用使用（例如打开系统集成、打开模型或更改用户）。使用为对外部提供者的任何调用提供的值。

不要以任何方式将这些存储为“静态”。存储为简单成员是确定的，因为类每个新会话创建此类的新实例。

示例实现

```
void ExampleIntegrationPlugin::SetAuthorisation(const char* 参数)
{
    日志消息 ( LOG_TRACE · __FUNCTION__ ) ;

    Json::Value jsonParameters;
    如果 ( strlen ( 参数 ) )
    {
        std::stringstream(参数) >> jsonParameters;
    }

    m_username = jsonParameters[ "username" ].asString(); // 如果使用基本授权。
    m_password = jsonParameters["密码"].asString();
    m_accessToken = jsonParameters["accessToken"].asString(); // 如果使用 OAuth。
```

}

设置回调

仅限 C++。这只需要在 C++ 中实现。示例中的代码就足够了，不需要修改。可以添加额外的错误或边界检查。

输入

参数	细节
<code>const void **</code> 回调函数	用于将数据传回Enterprise Architect的回调函数指针数组。

通过回调输出

没有任何

细节

C++插件会在创建后很快收到这个方法。它传入一个回调函数指针数组，稍后插件使用这些回调函数指针将数据传回Enterprise Architect

示例实现

```
void ExampleIntegrationPlugin::SetCallbacks(const void ** callbackFunctions)
{
    如果 ( 回调函数 )
    {
        AddProperty = (AddPropertyPtr)callbackFunctions[0];
        AddBinaryProperty = (AddBinaryPropertyPtr)callbackFunctions[ 1 ];
        SetErrorCode = (SetErrorCodePtr)callbackFunctions[2];
        SetError = (SetErrorPtr)callbackFunctions[3];
        LogMessage = (LogMessagePtr)callbackFunctions[4];
    }
}
```

设置配置

输入

参数	细节
参数	<ul style="list-style-type: none"> • C++：常量字符* • C#：string 参数A JSON string。有关可用参数的更多信息，请参阅详细信息。

通过回调输出

[可选] LogMessage - 设置有关收到的配置设置的log消息。注意不要log敏感信息。

注记：此方法将忽略 SetError 回调。

细节

此方法接收用户在启用此自定义集成插件时输入到专业云服务器配置中的详细信息。

它包括以下详细信息：

要连接的外部服务器：

- 服务器名称
- 服务器端口
- 服务器协议
- baseURL - 要附加到 url 的 url 文件夹

服务器设置组合形成如下 URL：<protocol>://<serverName>:<serverPort>/<baseURL>。

硬编码凭据 - 这些是可选的，可用于连接到具有通用帐户的提供商：

- 用户名
- 密码

权限 - 设置用户是否可以对外部提供者执行指定的操作：

- 允许创建项目
- 允许修改项
- 允许发表讨论

代理设置：

- 代理服务器
- 代理绕过
- 代理用户名
- 代理密码

示例实现

```
void ExampleIntegrationPlugin::SetConfiguration(const char* 参数)
{
    日志消息 ( LOG_TRACE · __FUNCTION__ ) ;

    Json::Value jsonParameters;
    如果 ( strlen ( 参数 ) )
    {
        std::stringstream(参数) >> jsonParameters;
    }

    // 将设置存储为成员变量以供以后使用。
    m_serverName = jsonParameters["serverName"].asString();
    m_serverPort = jsonParameters["serverPort"].asString();
    m_serverProtocol = jsonParameters["serverProtocol"].asString();
    m_baseURL = jsonParameters["baseURL"].asString();

    m_settingsUsername = jsonParameters[ "username" ].asString();
    m_settingsPassword = jsonParameters["密码"].asString();

    m_allowCreateItems = jsonParameters["allowCreateItems"].asString();
    m_allowModifyItems = jsonParameters["allowModifyItems"].asString();
    m_allowPostDiscussions = jsonParameters["allowPostDiscussions"].asString();

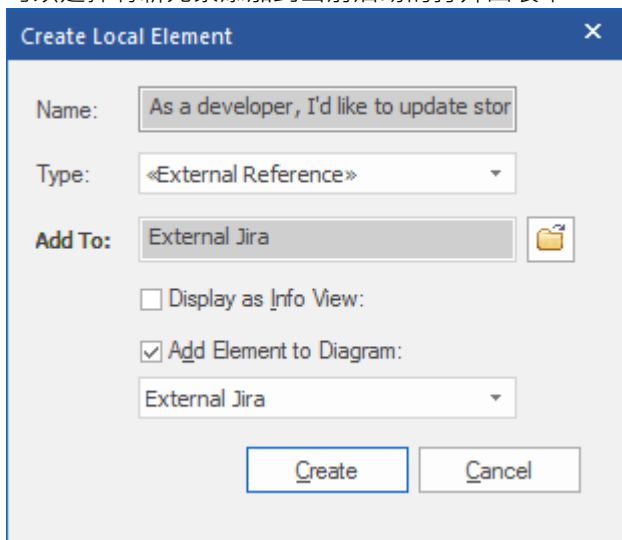
    m_proxyServer = jsonParameters["proxyServer"].asString();
    m_proxyBypass = jsonParameters["proxyBypass"].asString();
    m_proxyUsername = jsonParameters["proxyUsername"].asString();
    m_proxyPassword = jsonParameters["proxyPassword"].asString();
}
```

链接项

在Enterprise Architect中创建链接元素

要在Enterprise Architect模型中创建链接到外部项目的新元素：

1. 右键单击外部数据窗口列表中的外部项目。
2. 选择“创建本地元素”。
3. 选择默认类型或«ExternalReference»（忽略任何类型映射；请参阅[Configuration](#)帮助主题）。
4. 选择将在哪个包中创建元素。
5. 可以选择将新元素添加到当前活动的打开图表中。



本地元素属性将从“字段映射”对话框中定义的值创建，在[Configuration](#)中描述帮助话题。

链接后，本地元素将显示在外部项目下：



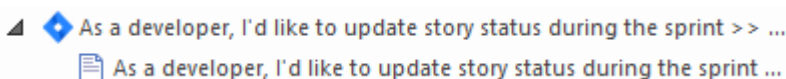
链接多个项

选择多个项目以同时创建多个元素。所有元素都将使用默认类型或«ExternalReference»创建。

- 要选择一组多个项目，请选择第一个项目，按住 shift 并选择最后一个项目。
- 要选择（或取消选择）单个项目，请按住控件并单击该项目。

链接现有的Enterprise Architect元素

Enterprise Architect中的现有元素可以链接到外部提供者中的现有项目，方法是将本地元素从浏览器窗口拖放到外部项目上。所以在我们的例子中：



将Enterprise Architect元素“作为开发者，我想在冲刺期间更新故事状态”从浏览器窗口拖放到同名的 Jira 项目上，从而在两者之间建立了联系。

在外部提供者中创建链接项目

Enterprise Architect中的现有本地元素可以通过以下方式添加到外部提供者：

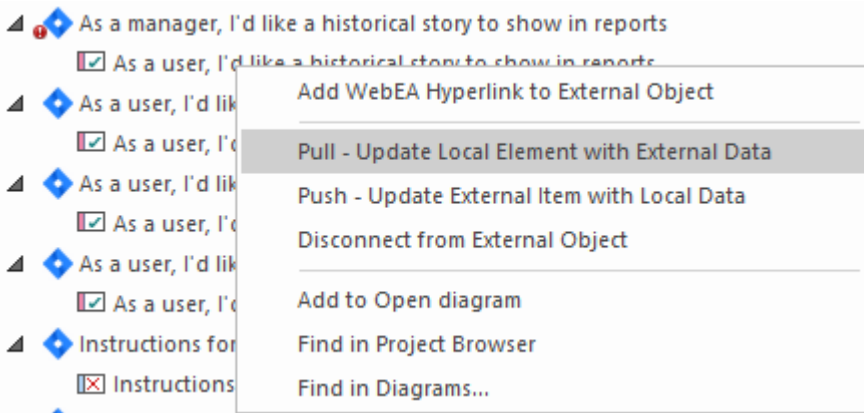
1. 选择本地Enterprise Architect元素（在浏览器窗口中）。
2. 浏览外部提供者并选择所需的目标位置。
3. 在“外部数据”菜单中，选择“创建链接项目”。

注记：

- 要创建新的外部项目，集成提供者必须配置为允许在专业云服务器上进行此操作（默认情况下不允许）
- 并非所有提供商都允许创建新项目

更新本地元素（“拉”更改）

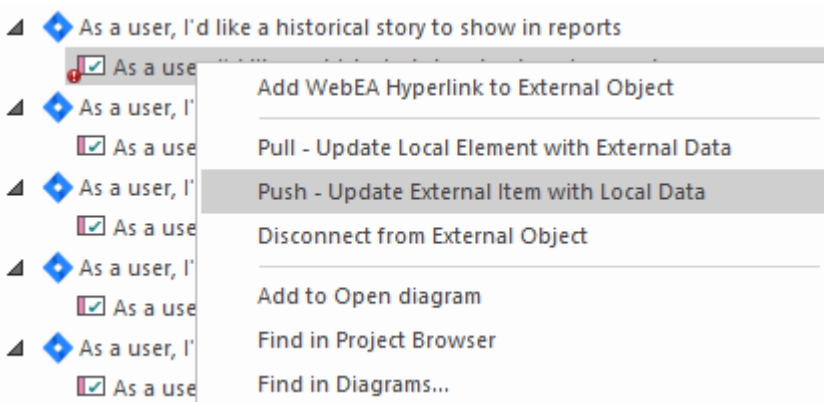
如果在创建或链接项目后本地元素或外部数据项已被修改，则“感叹号”指示符将显示在已更改项目名称旁边的图标顶部。右键单击链接的Enterprise Architect元素并选择“Pull - Update Local Element with External Data”选项。



注记：如果本地元素在创建链接后被修改，那么它的更改将被外部数据覆盖。

更新外部项目（“推送”更改）

如果在创建或链接项目后本地元素或外部数据项已被修改，则“感叹号”指示符将显示在已更改项目名称旁边的图标顶部。右键单击Enterprise Architect元素并选择“Push - Update External Item with Local Data”选项。



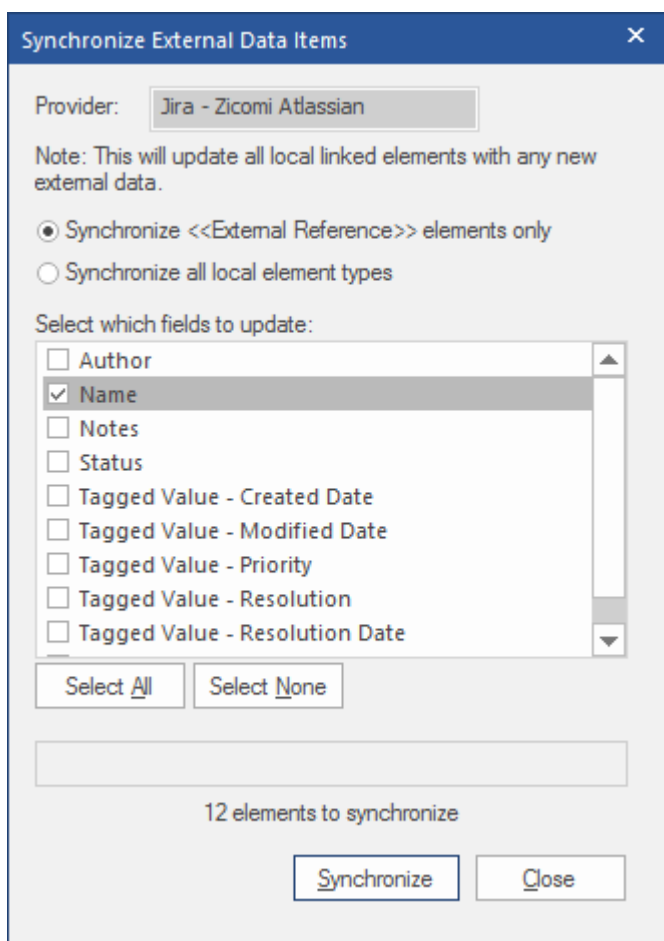
注记：

- 如果在创建链接后外部元素已被修改，则其更改将被本地数据覆盖
- 要更新外部项目，必须将集成提供者配置为允许在专业云服务器上进行此操作（默认情况下不允许）

同步所有本地元素

Enterprise Architect中的所有本地链接元素都可以使用来自外部数据提供者中链接项目的新数据快速更新。

1. 在“外部数据”窗口中，显示要同步的外部数据提供者中的数据。
2. 点击工具栏中的“汉堡包”图标，选择“同步本地链接元素”选项。
3. 在“同步外部项”对话框中，选中每个属性字段对应的复选框，这些属性字段应该使用来自外部项目的新数据进行更新。请注记，这些字段中Enterprise Architect元素中的任何本地更改都将被外部数据覆盖。
4. 窗口底部指示将更新多少本地元素（注意，此注记不清楚外部项是否有任何更改）。
5. 单击同步按钮。



外部项目详细信息

在“外部数据”窗口中选择外部项目时，将检索该项目的详细信息并显示在检验员窗口的“外部”选项卡中，以及相应属性的外部版本、注记和讨论和审阅窗口中。

检验员窗口

检验员窗口的“外部”选项卡仅在您查看外部项目时显示。它的行为方式与检验员窗口的“详细信息”选项卡大致相同，显示外部元素的关系、特征、要求、标记值、项目维护项目和相关文件。但是，“外部”选项卡仅显示存在项目的类别，并没有列出所有可能的但空的类别。

属性

单击列表中的外部项目以在属性窗口的“外部”版本中查看其属性。此窗口显示外部项目的所有可用属性，由外部提供者定义。

Properties

External

Mapped Properties

Type	Artifact
Stereotype	document
Name	Backup Process - swapping drives
Author	Howard Britten
Created Date	2016-08-26 14:57:40
Modified Date	2019-09-03 14:58:46
Status	Proposed
Complexity	Easy
Phase	1.0
Version	1.0

Other Properties

Type	Artifact
Unique Identifier	el_{E9D04385-7FDA-4f2b-824B-C57FA8DFC700}
Stereotype	document
Language	Java
WebEA URL	https://localcloud:1443?m=1&o=E9D04385-7FDA-4f2b-824B-C57FA8DFC700
Parent Resource Id...	pk_{3B47939A-6E1D-4327-B59E-5FBCB6EC526A}
Parent Resource - T...	Backup Processes
Parent Resource - I...	pk_{3B47939A-6E1D-4327-B59E-5FBCB6EC526A}
Parent Resource - T...	Package
Resource Type	Element

Properties | Toolbox

创建链接的本地元素时将使用的属性显示在顶部组“映射的属性”中，而所有其他属性都分组在“其它属性”中。可以配置属性映射，如配置帮助主题中所述。

笔记

外部项目的“笔记”或“描述”文本显示在笔记窗口的“外部”版本中。

Notes

External

Notes window showing the 'External' version and a toolbar with icons for adding, deleting, and refreshing notes.

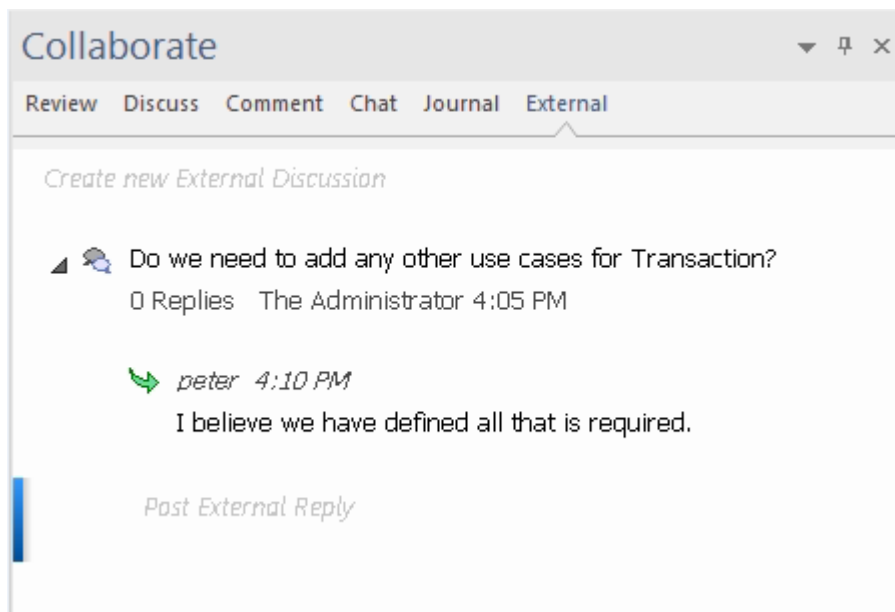
注记并非所有外部提供者都有“注记”或“描述”字段。

此版本的注记窗口有一个不同的工具栏，其中包含以下图标：

- 切换注记上的编辑锁定
- 同步Enterprise Architect视图中的注记和外部源
- 将编辑注记文本推送到外部源
- 显示在线Enterprise Architect 用户指南

讨论

从列表选择一个外部项目以查看并参与讨论和审阅窗口的“外部”选项卡中的讨论。“外部”选项卡类似“讨论”选项卡，并且行为方式相同。



可以通过双击创建新的外部讨论文本或右键单击选项卡并选择“创建新的外部讨论”菜单选项来发布新讨论。

注记：

- 要发布外部讨论，必须将集成提供者配置为允许在专业云服务器上进行此操作（默认情况下允许）
- 并非所有外部提供商都有讨论或评论

导航外部数据

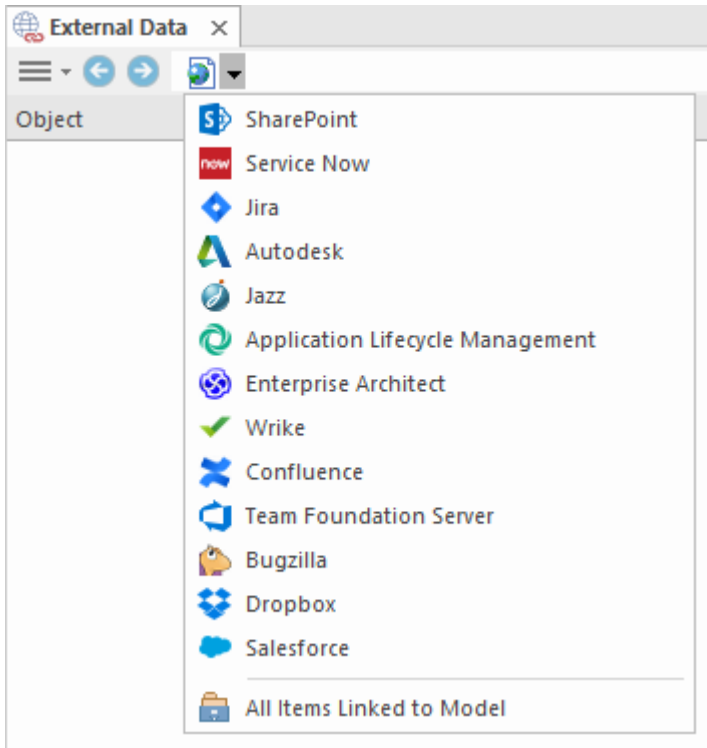
每个集成提供者都可以导航以显示可以链接到Enterprise Architect的外部项目列表。每个提供者可能会根据其存储数据的方式提供稍微不同的导航机制。有些提供简单的文件夹层次结构，而另一些则提供用户定义的过滤器。有关每个提供程序的详细信息，请参阅[Install and Configure](#)帮助主题中的集成插件返回什么数据表。

访问

功能区	特定>工具>系统集成>打开外部数据
-----	-------------------

导航层次结构

通过从右侧下拉菜单中选择提供程序类型开始导航。

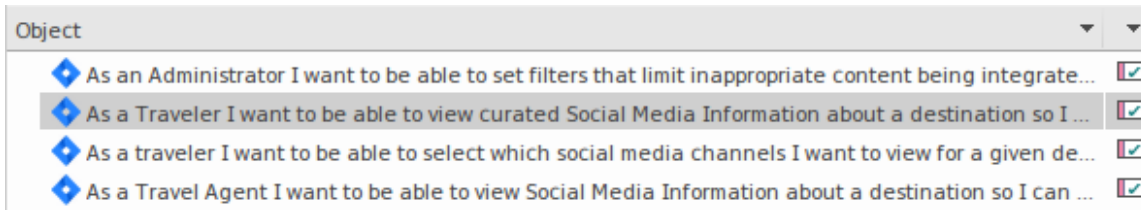


接下来，每种提供商类型都提供略有不同的导航系统；例如，Dropbox 允许浏览文件夹结构，而 Jira 提供项目列表，然后是用户查询列表。

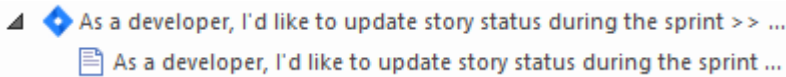


物品清单

在每个导航级别（如果可用），与导航级别对应的项目列表将填充左侧面板。



链接到外部项的任何本地Enterprise Architect元素都将显示为外部项的子项。

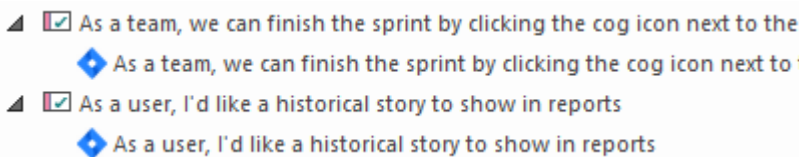


所有项链接到模型

在每个导航级别都有一个选项来显示链接到所选外部数据源的所有本地Enterprise Architect元素。选择“链接到模型的所有项”并选择从今天开始收集信息的天数（7、30或90天，或“全部”）。

在根导航级别选择“所有项链接到模型”以查看所有外部提供商的链接项。

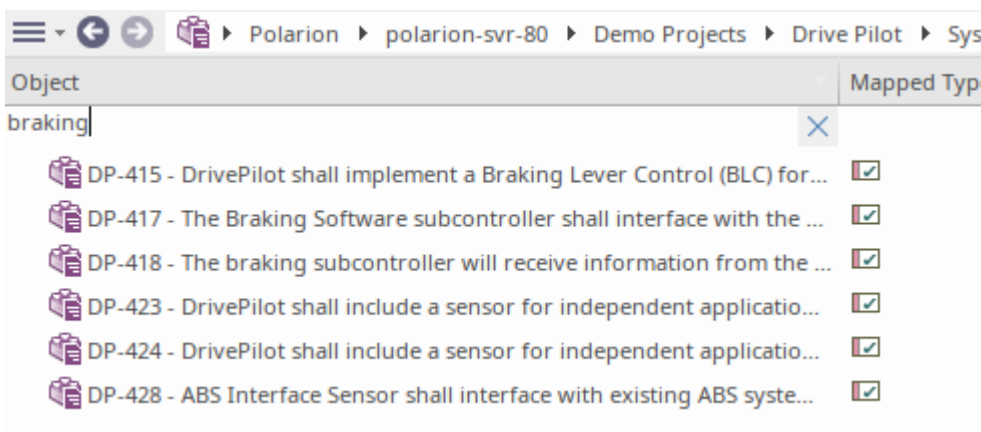
此视图不同于常规列表，因为它在顶部显示本地Enterprise Architect元素，并将外部链接项显示为本地元素的子项。所有相同上下文的菜单项都在视图中可用。



过滤器列表

可以使用过滤器栏过滤外部项目列表。要激活过滤器栏，请右键单击列表标题并选择“切换过滤器栏”。

可以通过在过滤器栏中输入文本来过滤列表。该列表将被过滤以仅显示具有包含过滤器文本的文本的那些项目。



显示项到当前上下文的项目

在集成菜单中，选择“链接到当前上下文的项目”以仅显示链接到当前选定的本地元素的外部项。也就是说，在浏览器窗口中选择一个项目并查看与其链接的外部项目。

此视图与“所有链接到模型的项”相同。

配置

每个集成提供者都带有一组默认映射值，这些值首先确定在Enterprise Architect中创建的本地元素类型，其次将哪些字段复制到新元素中。这些映射可针对每个客户端模型进行配置。

允许

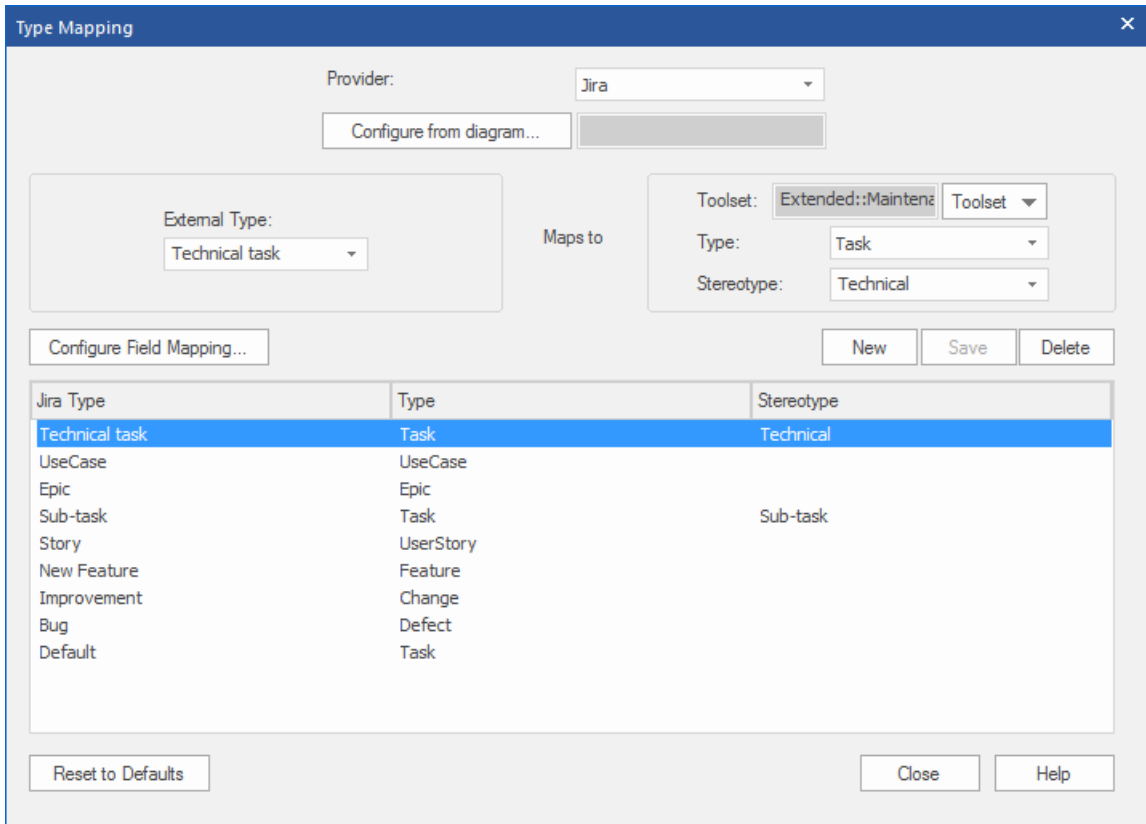
您必须具有 配置外部数据源”权限才能访问此功能。

访问

外部数据窗口工具栏	'汉堡包'菜单>配置
-----------	------------

类型映射

类型映射”对话框定义了当用户为外部元素选择 创建本地元素”选项时将创建的元素类型。



要创建新映射：

1. 单击新建按钮，然后单击 外部类型”下拉箭头并从外部源中选择要映射的元素类型。
2. 单击 工具集”下拉箭头并选择适当的Enterprise Architect工具集，然后单击 类型”下拉箭头并选择外部元素类型映射到的Enterprise Architect元素类型。

3. 如果合适，还单击“构造型”下拉箭头并选择外部元素类型映射到的扩展Enterprise Architect元素类型的构造型。
4. 单击保存按钮。

注记如果外部元素类型已被链接为 <<ExternalReference>>元素，则任何类型映射设置都将被忽略；请参阅 [Linking Items](#) 帮助主题。

字段映射

“元素字段映射”对话框定义了当用户选择“创建元素”选项时将为映射的元素更新的字段。您可以通过单击“类型映射”对话框上的配置字段映射按钮来显示此对话框，已将外部元素类型映射到Enterprise Architect元素（或选择了现有的映射元素类型）。

Jira Field	Enterprise Architect Field	Tagged Value Field
Updated	Tagged Value	Modified Date
Created	Tagged Value	Created Date
Creator - Name	Author	
Resolved	Tagged Value	Resolution Date
Resolution	Tagged Value	Resolution
Priority - Name	Tagged Value	Priority
Fix Version/s	Version	
Description	Notes	
Status - Name	Status	
Summary	Name	

该过程通常映射特定类型元素的字段；但是，如果您想为所有类型的元素映射某些字段，请单击“字段映射”对话框上的“外部类型”下拉箭头，然后选择值“默认”。

要映射字段：

1. 单击新建按钮，然后单击“外部字段”下拉箭头并从外部源中选择要映射的字段名称。
2. 单击“内部字段”下拉箭头并选择外部元素字段映射到的相应Enterprise Architect字段名称。
3. 如果内部字段是标记值字段，也请单击“标记值”下拉箭头，选择外部元素字段映射到的标记值。
4. 单击保存按钮。

注记

- 通过单击“重置为默认值”按钮，可以将每个映射重置为默认值
- 如果没有为给定元素类型定义类型映射，则将使用类型“默认”的映射

安装和配置

Sparx Systems专业云服务器产品是使用标准的 Microsoft窗口安装程序包安装的，该程序包包括许多可选组件。其中一个组件是集成插件，默认安装在 C:\Program Files (x86)\Sparx Systems\专业云服务器\SBPI\ 文件夹中（假设使用的是 64 位机器）。

需求

为了安装特定的集成插件，您必须具有：

- 授权A专业云服务器
- 对托管外部数据的服务器的物理网络访问
- 访问外部数据的用户凭据
- Enterprise Architect企业，统一或终极版，v14或更高版本
注意：Enterprise Architect试用版提供对集成的只读访问权限

包裹里有什么？

集成插件文件夹最初由集成服务器可执行文件 (SBPI.exe)、每个支持的外部产品的单独可执行文件 (*SBPI.exe)、示例配置文件和描述手动安装和配置步骤的文本文件组成。

插入	描述
集成 (SBPI) 服务器	SBPI.exe 应用程序充当专业云服务器和每个插件之间的接口，通过翻译 Enterprise Architect请求，将它们转发给适当的插件，然后将生成的响应返回给Enterprise Architect。
Application Lifecycle Manager插件	ALMSbpi.exe 插件与 MicroFocus Application Lifecycle Manager 产品（以前称为 HP Quality Center）交互。
Autodesk插件	AutodeskSbpi.exe插件与 AutoCAD 的文件和管理组件 AutoDesk 交互。
Bugzilla插件	BugzillaSbpi.exe插件与最初由 Mozilla 项目开发和使用的基于 Web 的缺陷/错误跟踪系统交互，现在根据 Mozilla 公共许可协议获得许可。
Confluence插件	ConfluenceSbpi.exe 插件与 Atlassian 的协作进行交互。它能够列出空间并链接到页面。 注：Confluence HTML 页面的内容是不同步的。
Dropbox插件	DropboxSbpi.exe 插件与 Dropbox 基于 Web 的文件托管服务交互。它能够列出 Dropbox 中的文件夹并链接到单个文件。
EA插件	插件与外部Sparx Systems的Enterprise Architect基于云的存储库交互。它能够浏览包层次结构或执行基于搜索的查询。
爵士乐插件	插件与： <ul style="list-style-type: none"> • IBM Rational DOORS Next Generation 的需求管理工具 • Rational Rhapsody设计管理 (DM) • Rational Team Concert更改和配置管理 (CCM)

	<ul style="list-style-type: none"> • 理性质量经理 (QM)
Jira插件	JiraSbpi.exe 插件与 Atlassian 的问题跟踪系统交互。它能够列出用户最喜欢的过滤器（也称为星标过滤器）。然后每个过滤器将列出过滤器返回的所有 Jira 项目。
Salesforce	SalesforceSbpi.exe 插件与Salesforce的顾客关系管理系统交互。
ServiceNow插件	ServiceNowSbpi.exe 插件与其基于云的企业管理系统的 ServiceNow 资产管理组件交互。
SharePoint插件	SharePointSbpi.exe插件与 Microsoft 基于 Web 的协作平台 SharePoint 进行交互。
Azure DevOps / TFS插件	插件与 Microsoft 的 Azure DevOps /团队基础服务器(TFS) 工作项交互
插件	WrikeSbpi.exe插件与 Wrike 的项目管理系统交互。

如何设置

集成框架由一个集成服务器 (SBPI.EXE) 应用程序组成，该应用程序启动一个或多个插件（例如 DropboxSbpi.exe 和 JiraSbpi.exe）。集成服务器和每个集成插件可以配置为与专业云服务器运行同一台机器上运行，也可以在完全不同的机器上运行。在最简单的配置中，集成服务器和所有集成插件都安装在单个服务器上。这种配置有两个主要优点：

1. 专业云服务器将在其窗口服务启动（或停止）时自动启动（和停止）所有配置的插件。
2. 云配置客户端内置的集成配置GUI，可以全面管理集成配置的方方面面；请参阅步骤 - 简单表。

但是，如果您选择在与专业云服务器不同的机器上运行集成服务器或集成插件，则必须手动配置每个单独的插件，并配置为使用正确参数作为窗口服务启动。请参阅步骤 - 手册表。

步骤 - 简单

云配置客户端中包含的集成配置 GUI 消除了配置集成服务器和数据提供者所涉及的大部分复杂性，因此这是大多数用户推荐的方法。但是，此配置方法仅限于在与专业云服务器相同的服务器上运行所有集成组件（*SBPI.exe）。

节	描述
为集成配置专业云服务器	<p>Sparx Systems专业云服务器的每个安装都可以配置为与单个集成服务器进行通信；此定义的配置选项定义为一组注册表设置，但是专业云服务器配置客户端中的“集成”选项卡允许定义和维护集成服务器选项，而无需您手动操作注册表或配置文件。</p> <p>此图显示了具有默认设置的集成服务器的定义：</p>

	 <p>或者，可以通过 WebConfig 接口配置集成服务器和插件。有关详细信息，请参阅WebConfig - Integration Plug-ins主题。</p>
<p>配置集成数据提供者</p>	<p>每个数据提供者（或插件）都被定义为 [HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\插件\{unique} 中的一系列注册表项和配置文件中的设置。同样，云配置客户端中的“集成”选项卡允许定义和维护集成数据提供者的详细信息，而无需您手动操作注册表和配置文件。</p>
<p>配置防火墙</p>	<p>为了尽量减少配置专业云服务器及其特征所需的防火墙规则，PCS 版本 4.1 现在通过普通 PCS 端口路由对集成服务器（1）的所有请求，因此如果您使用的是 PCS 4.1 或更高版本不需要额外的防火墙规则，除了 Enterprise Architect 客户端与专业云服务器通信的规则。</p> <p>对于 PCS 的版本 3 和 4，集成服务器 (SBPI.exe) 通常必须被授予通过任何本地防火墙的访问权限，以便 Enterprise Architect 客户端可以连接到它。专业云服务器安装程序将自动创建一个防火墙异常，允许任何传入请求传递给 SBPI.exe；但是，应检查和调整默认设置以适应您的环境。</p>

步骤 - 手动

重要提示：仅当集成组件将在与专业云服务器不同的运行上运行时才需要这些步骤，否则应使用步骤 - 简单表。

节	描述
<p>为集成配置专业云服务器</p>	<p>Sparx Systems 专业云服务器的每个安装都可以配置为与单个集成（或 SBPI）服务器进行通信。这些配置选项被定义为一组注册表设置。</p> <p>这是集成服务器的所有有效选项的示例：</p> <pre>[HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\服务器] 启用="真" 本地端口="dword:00001f90" "UseLegacy"="false" "参数"="-port 8080 -protocol http" 协议="https" "服务器"="localhost" "端口"="dword:00001f90" "忽略SSL错误"="真" 尝试自动发现="真" "客户端协议"="http" "ClientServer"="alternativeservername" 客户端端口="dword:00001f90"</pre> <ul style="list-style-type: none"> Enabled - true or false，表示端口服务器应该监听的端口号，哪个值应该

	<p>与参数中指定的值匹配；例如， dword:00001f90（十进制 8080）</p> <ul style="list-style-type: none"> • LocalPort - 一个十六进制值，表示当集成服务器使用选项为 FALSE 时服务器正在侦听的端口号；例如， dword:00001f90（十进制 8080） • UseLegacy - true or false，控制集成服务器是否应使用简单（false）或复杂（true）配置选项集 • 参数- 不用于专业云服务器4.0 或更高版本；在早期版本中，这表示用于启动集成服务器的参数，其中包括服务器应侦听的端口和协议 - 例如，“-port 8080 -protocol http” • 协议- http 或https，当使用旧选项为 TRUE 时，应用于与托管集成服务器的机器通信的协议；该字段与“服务器”和“端口”结合形成服务器的URL，集成服务器专业云服务器将向该服务器发送SBPI相关请求 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由专业云服务器机器解析 • 服务器- 当使用旧选项为 TRUE 时托管集成服务器的机器的名称（或 IP 号）（例如， yourdomain.com）；该字段与“Protocol”和“端口”组合形成集成服务器的URL，集成服务器会将专业云服务器请求发送到 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由专业云服务器机器解析 • 端口- 一个十六进制值，表示当集成服务器使用继承选项为 TRUE 时服务器正在侦听的端口号 - 例如， dword:端口（十进制 8080）；该字段与“协议”和“服务器”组合形成集成服务器的URL，专业云服务器将向该URL 发送服务器相关请求 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由专业云服务器机器解析 • IgnoreSSLErrors - true or false，定义当使用Legacy 选项为 TRUE 时，是否应忽略与集成服务器组件通信时发生的 SSL 相关错误 • AttemptAutoDiscovery - true or false，定义专业云服务器是否应在使用 Legacy 选项为 TRUE 时自动尝试确定Enterprise Architect客户端的网络地址并将其提供给集成服务器 • ClientProtocol - http 或https，定义协议，当使用Legacy 选项为TRUE 时，当与‘ClientServer’和‘ClientPort’组合时，形成Enterprise Architect客户端可以与集成服务器通信的可解析URL 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由Enterprise Architect客户端机器解析 • ClientServer - 定义服务器名称（或 IP 号），当与“ClientProtocol”和“ClientPort”组合时，当使用旧选项为 TRUE 时，Enterprise Architect客户端可以与集成服务器通信，形成可解析的URL 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由Enterprise Architect客户端机器解析 • ClientPort - 一个十六进制值，定义端口号，当与“ClientProtocol”和“ClientServer”组合时，当使用Legacy 选项为TRUE 时，Enterprise Architect客户端可以与集成服务器通信，形成可解析的URL；例如， dword:00001f90（十进制 8080） 注记：完成URL ({protocol}://{server-name}:{port}) 必须可由Enterprise Architect客户端机器解析 <p>注记：1 4.1版本的专业云服务器开始，集成服务器不需要有自己的防火墙规则，以便Enterprise Architect客户端可以与之通信。对于 PCS 版本 3 和 4，集成服务器(SBPI.exe) 通常需要通过任何本地防火墙的访问权限，以便客户端可以连接到它。专业云服务器安装程序会自动创建防火墙异常，允许任何传入请求通过；但是，应检查和调整默认设置以适应您的环境。</p>
配置数据提供者	<p>每个数据提供者被定义为一组注册表项：</p> <p>[HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\插件\{unique}]</p>

其中 {unique} 是数据提供者的唯一 UUID。

这是完成外部数据提供者定义的示例：

```
[HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\插件
\{853489C1-4C22-4bad-9A8E-3098D07A3FC1}]
```

自动启动"= 真"

启用"= 真"

"组"=""

"名称"=" Sparx Systems示例帐户"

"端口"=dword:00001f91

特色"="前缀"

"TypeKey"="jira"

"参数"="-port 8081 -config jr.config"

"配置"="jr1.config"

- **AutoStart** - true or false · 定义集成服务器(SBPI.exe) 是否应该为此数据提供者维护一个正在运行的进程 (*sbpi.exe)
- **已启用**- true or false · 定义集成服务器(SBPI.exe) 是否应允许将通信转发到此数据提供者
- **Group** - 一个可选值 · 当在Enterprise Architect中显示时 · 可用于将提供者排 "成组"
- **名称**——一个 友好的"项目名称 · 用于描述外部数据提供者 · 向所有 Enterprise Architect用户显示；例如 "Sparx Systems示例帐户"
- **端口**- 一个十六进制值 · 表示外部数据源期望在其上接收请求的端口号 · 该值应与在端口中指定的值匹配；例如 · dword:00001f91 (十进制 8081)
- **前缀**-对用户有意义的简短唯一名称 · 并为存储在Enterprise Architect模型中的每个链接添加前缀；例如 · jr1::10001 (其中 10001 是 Jira object ID)
- **TypeKey** - 定义当前Data提供者的提供者类型；只能使用以下受支持的值：cint、csvc、alm、ad、bug、cflu、drop、ea、jazz、jira、sf、现在、sp、tfs、wrike
- **参数**- 从 PCS 1开始弃用 · 动态参数列表是根据各个设置构建的；早期版本使用此字段来定义应该用于启动集成的参数 · 包括插件端口、协议和配置文件名
- **配置**- 从 PCS 4.1开始弃用 · 配置文件名现在是带有 "现在"的前缀；在 PCS 3 和 PCS 4 版本中 · 该字段被赋予了一个独立的值 · 该值必须是唯一的

每个外部数据提供者都需要自己的一组选项来定义配置的插件如何连接到外部数据源的详细信息。这些设置存储在 .config 文件中 · 该文件与插件的 .exe 文件位于同一位置。例如 · 使用上面的定义需要创建一个 "jr1.config" · 并包含与此类似的信息：

协议=https

服务器=example.com

端口=443

BASEURL=我的项目

用户名=

密码=

创建项目=假

修改项目=假

后讨论=真

代理=10.0.0.0:3128

代理=<本地>;10.*

	<p>IGNORESSLERORS=真</p> <ul style="list-style-type: none"> • PROTOCOL - 通信协议，http 或https • SERVER - 外部数据源的服务器名称（或 IP 号），例如 example.com • PORT - 外部数据源配置监听的端口，例如 443 • BASEURL - 当外部数据源支持多个“项目”时，BaseURL属性标识应该使用哪个；例如，对于Enterprise Architect SBPI，BaseURL 是 PCS 中定义的 DB别名 • 用户名- 可选 - 见注 1 • 密码- 可选 - 见注 1 • CREATEITEMS - 定义Enterprise Architect用户是否可以在外部数据源中创建项目 - 见注 2；默认值：false • MODIFYITEMS - 定义Enterprise Architect用户是否可以修改外部数据源中的现有项目 - 见注 2；默认值：false • POSTDISCUSSIONS - 定义Enterprise Architect用户是否可以将讨论添加到外部数据源中的项目 - 见注 2；默认值：真 • PROXY - 可选 - 代理服务器的服务器名称或IP号和端口，例如 10.0.0.0:3128 • PROXYBYPASS - 可选 - 不应该通过代理发送的 IP 号码的分号分隔列表；例如，<本地>;10.* • IGNORESSLERORS - 可选 - 忽略任何 SSL 证书错误的布尔值 <p>注1</p> <ul style="list-style-type: none"> • 如果外部服务器支持 OAuth 2 身份验证（Autodesk、Dropbox 和 Wrike），从Enterprise Architect内部访问时，将打开一个浏览器窗口并提示您log外部帐户并允许Enterprise Architect访问您的帐户 • Enterprise Architect在此过程中永远不会看到您的凭据；相反，它提供了一个唯一的令牌来访问外部帐户 • 如果外部服务器允许基本身份验证，则可以在配置文件中选择设置用户名和密码 • 如果配置文件中未指定用户名和密码，Enterprise Architect将提示您提供凭据以访问外部数据 <p>笔记2</p> <ul style="list-style-type: none"> • 并非所有外部产品都支持创建和修改其对象；例如，Dropbox 没有
<p>为外部数据提供者配置模型</p>	<p>为了让Enterprise Architect的用户连接到给定的外部数据提供者，必须配置Enterprise Architect模型和外部数据提供者模型的“绑定”。这是通过 [HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\Bindings\{unique}] 中的一系列注册表设置为每个模型/外部数据提供者组合执行的。在此string中，{unique} 是唯一标识绑定的 8 位十六进制数。例如：</p> <pre>[HKEY_USERS\DEFAULT\Software\Sparx Systems\SQLBridge\SBPI\Bindings\B6EE6851] "LocalModel"="eaexample" "插件"="{853489C1-4C22-4bad-9A8E-3098D07A3FC1}"</pre> <ul style="list-style-type: none"> • LocalModel -模型的 DB别名，例如 'eaexample' • Plugin - 外部数据源的 UUID；例如，{853489C1-4C22-4bad-9A8E-3098D07A3FC1}

特别注记

在集成服务器或提供者级别使用 SSL

如果您选择在集成服务器级别或集成提供者级别实现 HTTPS 协议，则集成可执行文件 (*SBPI.EXE) 将需要与它们在同一文件夹中的 “server.pem” 文件；因此，如果使用默认安装路径，这意味着 “server.pem” 应放在 C:\Program Files (x86)\ Sparx Systems \Future专业云服务器\SBPI\ 文件夹中。

请参阅 [Add a Port Definition](#) 帮助主题中的 SSL 证书部分和 [Self-Signed SSL Certificates](#) 帮助话题 有关如何创建有效的 “server.pem” 文件的更多信息。

Enterprise Architect的集成插件需求：

- The别名as the模型's DB
- 定义的模型在专业云服务器中配置为 启用”和 启用专业特征 (OSLC、WebEA和集成)”

集成插件

集成插件组件代表一组用于专业云服务器的插件（或扩展），使来自外部提供商的数据能够在Enterprise Architect中显示。这些组件也称为基于服务器的插件接口(服务器)。

专业云服务器的集成插件运行作为一系列本身没有图形用户界面（GUI）的可执行文件（.exe文件）运行；但是，本节的主题描述了包含在专业云服务器的配置客户端中的配置编辑器屏幕，可用于配置集成插件。

配置客户端（作为专业云服务器的一部分安装）的主屏幕包括一个“集成”选项卡，该选项卡分为两个主要部分。上半部分定义了集成服务器（或服务器）的详细信息，下半部分显示了各种“数据提供者”的详细信息；这些部分中的每一个都在本主题中进行了更细节的描述。

笔记1：集成特征仅在专业云服务器的许可版本中可用：团队、企业或令牌。

笔记2：如果您使用的是专业云服务器的Token版本，在配置集成之前，请确保您已通过Token Allocation选项启用所需的集成提供者/s；请参阅[Manage Allocations](#)帮助主题。

笔记3：如果您使用的是Enterprise Architect试用版，集成插件是只读的，不会更新任何存储的数据

笔记4：对集成服务器或插件的任何更改（通过配置客户端或手动进行）都需要重新启动专业云服务器

或者，可以通过 WebConfig 接口配置集成服务器和插件。有关详细信息，请参阅[WebConfig - Integration Plug-ins](#)帮助主题。

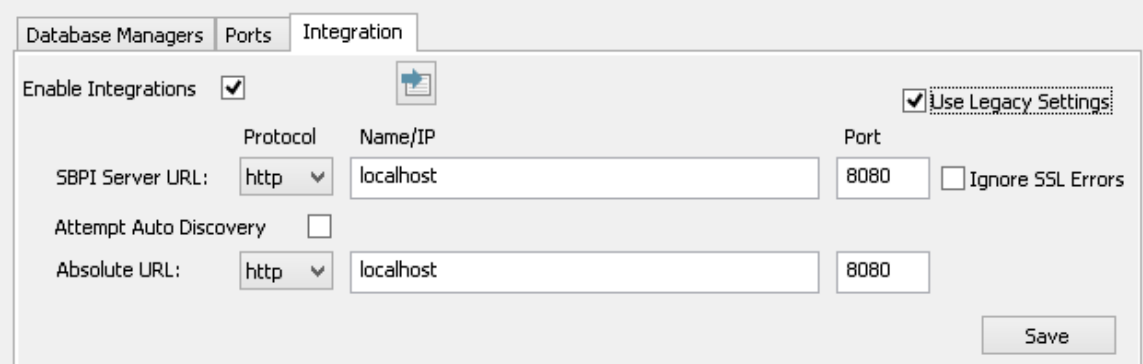
还专门针对[Walkthrough: Jira Integration](#)和[Walkthrough: Polarion Integration](#)集成提供了详细的演练。

集成服务器



Enterprise Architect对外部数据的所有请求都发送到集成服务器（通过专业云服务器），后者决定哪个数据提供者（或插件）将处理该请求。

为了让大多数用户的集成服务器配置尽可能简单，专业云服务器的 4.1 版本引入了一组简化的选项；但是，如果默认配置不适合您的环境，请选择“使用传统设置”选项以显示所有可用选项。



笔记：集成服务器设置仅在选中“启用集成”标志时是必需的。协议、名称和端口的组合必须与托管专业集成服务器（在本例中为专业云服务器）的机器相匹配，以便Enterprise Architect客户端能够与之通信。

物品	描述
启用集成	<p>默认值：未选中</p> <p>此复选框启用（或禁用）当前专业云服务器安装的集成。</p> <p>选择此值时，意味着其他字段（“协议”、“名称”、“端口”和“路径”）现在是必需的，并且如果每个字段中没有值，则无法保存集成服务器定义。</p>
加载默认	<p>此按钮将集成服务器设置的默认值加载到任何空字段中。</p> <p>注记：如果一个字段已经有一个值，那么 Load Defaults 按钮将不会改变或更新这个现有的值</p>
使用旧版设置	<p>默认值：未选中</p> <p>此复选框确定是否应显示简化或复杂的选项集。该名称是指专业云服务器的早期版本要求用户填写复杂的选项集。在更高版本的专业云服务器中，通过做出一些适用于大多数环境的假设，大大减少了所需的选项集。</p> <p>选择此值时，意味着其他字段（“服务器URL”：“协议”、“名称/IP”和“端口”）现在是必需的，并且如果每个字段中没有值，则无法保存集成服务器定义。</p>
端口	<p>默认值：8080</p> <p>此选项仅在未选择“使用旧版设置”选项时显示。它定义了集成服务器正在侦听来自专业云服务器的请求的端口。</p> <p>确保没有其他应用程序或服务已经在使用所选端口。</p>
服务器URL：协议	<p>默认值：HTTP</p> <p>此选项仅在选中“使用旧版设置”选项时显示。</p> <p>此字段与“SBPI服务器名称/IP”和“SBPI服务器端口”结合使用“SBPI服务器的协议”；这些设置形成了集成服务器的 URL，专业云服务器将向其发送 SBPI 相关请求。</p> <p>注记：完成URL ({protocol}://{server-name}:{port}) 必须可由专业云服务器机器解析。</p>
服务器URL：名称/IP	<p>默认值：localhost</p> <p>此选项仅在选中“使用旧版设置”选项时显示。</p> <p>此字段定义“SBPI服务器的机器名称或 IP”，结合“SBPI服务器协议”和“服务器专业云服务器端口”这些设置形成集成服务器的 URL，服务器将向其发送 SBPI 相关请求。</p> <p>注记：完成URL ({protocol}://{server-name}:{port}) 需要由专业云服务器机器解析。</p>
服务器URL：端口	<p>默认值：8080</p> <p>此选项仅在选中“使用旧版设置”选项时显示。</p> <p>该字段与“服务器服务器端口协议”和“端口服务器端口定义”相结合；这些设置形成了集成服务器的 URL，专业云服务器将向其发送 SBPI 相关请求。</p> <p>注记：完成URL ({protocol}://{server-name}:{port}) 需要由专业云服务器机器解析。</p>
忽略 SSL 错误	<p>默认值：未选中</p> <p>此选项仅在选中“使用旧版设置”选项时显示。</p> <p>此值定义是否应忽略与集成服务器组件通信时发生的与 SSL 相关的错误。最常见的 SSL 错误与自签名证书有关；默认情况下，客户端不将自签名证书视为安全的，因此为了连接到使用自签名证书的服务器，应选中此选项。</p>

<p>尝试自动发现</p>	<p>默认值：选中 此选项仅在选中“使用旧版设置”选项时显示。 此选项定义专业云服务器是否应自动尝试确定Enterprise Architect客户端的网络地址并将其提供给集成服务器。 此选项是在专业云服务器4中引入的。1.40;早期版本无法发现客户端的网络地址。 选中“尝试自动发现”时，设置为“后备URL”，仅在自动发现失败时使用。如果未选中“尝试自动发现”，则设置为在所有情况下都将使用的“绝对URL”。</p>
<p>绝对URL：协议</p>	<p>默认值：HTTP 此选项仅在选中“使用旧版设置”选项时显示。 该字段定义了与绝对服务器名称端口结合使用时的协议，并形成Enterprise Architect客户端可以与集成服务器通信的可解析URL。 注记：完成URL {protocol}://{server-name}:{port} 需要由Enterprise Architect客户端机器解析。 在某些环境中，服务器URL被重定向到完全不同的位置，因此为了处理这种情况，将绝对URL返回给Enterprise Architect客户端，以便它们可以与集成服务器通信。</p>
<p>绝对网址：名称/IP</p>	<p>默认值：{empty} 此选项仅在选中“使用旧版设置”选项时显示。 此字段定义服务器名称/IP，当与绝对协议结合使用时，端口形成Enterprise Architect客户端可以与SBPI服务器通信的可解析URL。 注记：完成URL {protocol}://{server-name}:{port} 需要由Enterprise Architect客户端机器解析。 在某些环境中，服务器URL被重定向到完全不同的位置，因此为了处理这种情况，将绝对URL返回给Enterprise Architect客户端，以便它们可以与服务器通信。</p>
<p>绝对URL：端口</p>	<p>默认值：8080 此选项仅在选中“使用旧版设置”选项时显示。 此字段定义端口，当与绝对协议和服务器名称结合使用时，形成Enterprise Architect客户端可以与SBPI服务器通信的可解析URL。 注记：完成URL {protocol}://{server-name}:{port} 需要由Enterprise Architect客户端机器解析。 在某些环境中，服务器URL被重定向到完全不同的位置，因此为了处理这种情况，将绝对URL返回给Enterprise Architect客户端，以便它们可以与服务器通信。</p>
<p>节省</p>	<p>此按钮保存对集成服务器设置的任何未决更改；它仅在尚未保存的更改时启用。</p>

数据提供者

单个专业云服务器可以支持任意数量A外部数据提供者，反过来，每个外部数据提供者可以用于安装许可证支持的尽可能多的模型。注记只有支持Pro的存储库（或数据库管理器）可以访问外部数据提供者。启用Pro A数据库管理器是选中了“启用专业特征（OSLC、WebEA和集成）”选项的数据库管理器。

选项	描述
----	----

定义的提供者	此控件列出了所有已定义的外部数据提供者的摘要。 双击现有项目将显示编辑数据提供者屏幕。
添加...	此按钮显示添加数据提供者屏幕以允许输入新的外部数据提供者。
编辑...	此按钮显示编辑数据提供者屏幕以允许修改现有的外部数据提供者定义。 注记： 此按钮仅在选择了数据提供者时启用。
消除	此按钮永久删除选定的数据提供者及其所有绑定。 注记： 此按钮仅在选择了数据提供者时启用。
<'插件名称'> 绑定到：	此控件显示所有已启用 Pro 的数据库管理器的检查列表（即，选择了 启用专业特征（OSLC、WebEA和集成）”选项）。选定A数据库管理员将当前数据提供者分配给给定的存储库。
选择选	此按钮是一种将所有已定义的启用 Pro 的数据库管理器分配给当前选定的数据提供者的快速简便的方法。
不选全部	此按钮是从当前选定的数据数据库提供者中取消分配所有已定义的启用数据库的管理器的快捷方式。

添加/编辑数据提供者

每当您请求创建新数据提供者或编辑现有数据提供者时，都会显示添加/编辑数据提供者屏幕。屏幕的行为在两种模式下都是相同的，一个区别是当屏幕以“编辑”模式加载时，将显示所选数据提供者的详细信息。

当您选择添加新的数据提供者时，屏幕最初显示如下：

The screenshot shows the 'Add Data Provider' dialog box with the following fields and values:

- Data Provider:**
 - Enabled:
 - Name: * (empty text box)
 - Provider: * (dropdown menu)
 - Prefix: * (empty text box)
 - Group: (empty text box)
- Integration Plugin:**
 - Auto Start:
 - Port: * 8082
- Provider Server:**
 - Protocol: * (dropdown menu)
 - Server Name/IP: * (empty text box)
 - Port: * (empty text box)
 - Base URL: (empty text box)
 - Maximum Request Time: 60 seconds
 - Authentication: Credentials (dropdown menu)
 - User Name: (empty text box)
 - Password: (empty text box)
 - Create Items:
 - Modify Items:
 - Post Discussions:
 - Ignore SSL Errors:
- Logging:**
 - File Count: 3
 - Max File Size: 1048576
 - Level: WARNING (dropdown menu)
 - Directory: C:\Program Files\Sparx Systems\Pro Cloud Server\SBPI\Logs\
- Proxy:**
 - Server Name/IP: (empty text box)
 - Bypass: (empty text box)
 - User Name: (empty text box)
 - Password: (empty text box)

Buttons: Help, OK, Cancel. Note: * Denotes a value is required

在可能的情况下，添加/编辑数据提供者屏幕将使用默认值填充每个字段，以使您更轻松；如果默认值不正确，只需覆盖它们即可。为此，每当更改“提供者”值时，许多其他字段将设置为新提供者默认值，例如数据提供者的前缀，如下所示，选择Enterprise Architect时：

注记：在定义数据提供者时，有许多必填字段（用 * 表示）需要在数据提供者可以保存之前输入值。

数据提供者设置

物品	描述
启用	默认值：未选中 此值控制集成服务器(SBPI.exe) 是否应允许将通信转发到此数据提供者。
名称	此值表示当前数据提供者的友好名称。它将显示给Enterprise Architect的用户。
提供者	该值指定当前Data提供者的提供者类型；只能使用受支持的值。如前所述，每当更改此值时，都会自动更新许多其他字段，以使用户更轻松地输入数据。
前缀	该值代表一个简短的唯一名称，用于标识当前的数据提供者。该值将针对链接到外部元素的每个Enterprise Architect元素保存。
团体	此字段仅在选择“自定义集成”提供程序类型时可用。 通常在Enterprise Architect中，集成按提供者类型（例如“Jira”或“Dropbox”）分组。对于自定义集成，根据在此字段中输入的值应用分组。
最大请求时间	此字段仅在选择“自定义服务”提供商类型时可用。 此值表示对提供者系统的调用在超时之前将等待的最长时间（以秒为单位）。当提供者系统远程或性能缓慢时，增加此值特别有用。

	<p>默认值为 60 秒。</p> <p>注记：这是专业云服务器v4.2 build 65引入的新设置。</p>
--	--

集成插件settings

这组设置定义了集成插件和集成服务器之间的关系以及两者如何交互。

物品	描述
自动开始	<p>默认值：选中</p> <p>此值控制集成服务器(SBPI.exe) 是否应为此数据提供者维护一个正在运行的进程 (*sbpi.exe)。</p> <p>对于标准集成插件，我们希望检查此值；但是，对于自定义插件，不自动启动插件可能很有用，而是手动控制它何时运行，尤其是在开发时。</p>
端口	<p>默认值：8081</p> <p>此值表示集成服务器应用于与集成通信的插件端口。集成服务器假定端口是相对于当前机器的（或 http:// localhost :{port} ）。</p> <p>每个集成插件都需要一个唯一的端口编号，当尝试保存插件的设置时将检查该编号。但是，专业云服务器只能查看专业云服务器本身及其组件在使用什么端口，与其他应用程序发生冲突的端口仍然存在。</p>
协议	<p>该值表示集成服务器应使用的协议与集成插件进行通信；在最简单的配置中，这将是 "HTTP"。</p>
DLL 路径	<p>此字段仅适用于“自定义服务”、“自定义集成”和“翻译器”提供程序类型。</p> <p>此值表示自定义 DLL 的物理路径，将由 CustomService.exe 或 CustomIntegration.exe 数据提供者进程调用。</p> <p>为方便起见，该应用程序使用 \SBPI\Custom\ 文件夹 (C:\Program Files (x86)\ Sparx Systems \专业云服务器\SBPI\Custom\) 中存在的所有 .DLL 文件填充下拉列表。</p>

自定义属性

自定义属性部分仅针对“自定义服务”类型的数据提供者显示。

物品	描述
自定义属性列表	<p>列出自定义数据提供者的当前属性。</p>
添加	<p>此按钮显示“添加属性提供者自定义属性”屏幕并允许输入新的自定义属性，定义为项目/值对。</p>
编辑	<p>此按钮显示“编辑属性提供者自定义属性”屏幕并允许修改现有自定义属性定义。</p>
删除	<p>此按钮从当前数据提供者中永久删除选定的自定义属性。</p>

提供者服务器设置

这组设置定义了外部数据提供者的详细信息，例如 Dropbox、Jira 或 TFS。集成插件将使用定义的详细信息连接到远程系统并从中检索数据，以便可以将其发送回 Enterprise Architect。协议”、服务器名称”和 端口”的组合必须由托管集成的服务器插件，在最简单的情况下是专业云服务器

物品	描述
协议	<p>该值表示集成插件应使用的协议与外部数据提供者进行通信。</p> <p>注记：大多数在线云托管提供商需要https和 443 端口。</p>
服务器名称/IP	<p>此值表示集成插件应用于与外部数据提供者通信的服务器名称或 IP 地址。</p>
端口	<p>该值表示集成应该用来与外部数据提供者通信的插件端口。</p> <p>注记: 大多数在线托管帐户都运行https，这需要通过 443端口记录。自托管服务器将取决于单个服务器配置。</p> <ul style="list-style-type: none"> • Jazz：服务器默认使用端口或https • Jira：自托管服务器默认端口是http端口8080 • Confluence：自托管服务器默认端口是http端口8090
基本网址	<p>一些外部数据提供者允许在一个位置有多个存储库的概念；对于此类数据提供者，此字段充当过滤器。</p> <p>例如，可以使用另一个Enterprise Architect模型作为外部数据源；在这种情况下，协议”、服务器”和 端口”确定专业云服务器（例如，http://别名:804），而 基础”是存储库 (myModel) 的 DB 名称。因此： http://myEAServer:804/myModel</p> <p>特定提供商的详细信息：</p> <ul style="list-style-type: none"> • Confluence：在线云托管帐户要求此为“wiki”；自托管服务器将取决于单个服务器的设置 • Jira：对于在线云托管帐户，这应该是空白的；自托管服务器将取决于单个服务器的设置 • Dropbox：这应该留空 • Enterprise Architect：这应该是要连接的模型别名
最大请求时间	<p>此值表示对提供者系统的调用在超时之前将等待的最长时间（以秒为单位）。当提供者系统远程或性能缓慢时，增加此值特别有用。</p> <p>默认值为 60 秒。</p> <p>注记：这是专业云服务器v4.2 build 65引入的新设置。</p>
验证	<p>默认值：凭据</p> <p>该值允许选择应该用于提供者身份验证类型，有效选项是 Credentials 和 OAuth。</p> <p>选择“凭据”时，预计会输入用户名称和密码。</p> <p>选择“OAuth”时，预计将在Data Provider Advanced Authentication屏幕上填写高级身份验证选项。</p> <p>注记1：此选项仅在所选提供者支持多种形式的身份验证时启用，例如：</p>

	<p>Dropbox、Jira、Salesforce、ServiceNow、SharePoint 和 Wrike</p> <p>笔记2: 此选项在专业云服务器v5.0 build 100 中引入，之前的版本仅支持基于凭证的身份验证。</p>
用户名称	<p>此值表示应该用于访问外部系统中的数据的用户名（与密码结合）。如果在此字段中定义了一个值，则当前数据提供者的所有Enterprise Architect用户将使用同一组凭据来读取外部系统。否则，如果用户名字段留空，Enterprise Architect将提示每个用户独立地获取他们的凭据。</p> <p>此值仅在选择“凭据”身份验证时显示。</p> <p>笔记：对于许多在线云帐户，用户名可以是该帐户的电子邮件地址。</p>
密码	<p>此值表示与“用户名称”字段中指定的用户名匹配的密码。</p> <p>此值仅在选择“凭据”身份验证时显示。</p> <p>笔记：对于 Atlassian云托管帐户（Jira、Confluence），常规密码的使用已替换为 API 密钥。请参阅 Atlassian 文档，了解如何获取 API 密钥并将 API 密钥粘贴到密码字段中。</p>
高级...	<p>此按钮用于显示Data Provider Advanced Authentication屏幕，仅在选择 OAuth 身份验证时可见。</p>
创建项	<p>默认值：未选中</p> <p>此复选框控制Enterprise Architect用户是否能够在外部系统中创建新项目。</p>
修改项	<p>默认值：未选中</p> <p>此复选框控制Enterprise Architect用户是否能够在外部系统中外部项目的详细信息。</p>
发表讨论	<p>默认值：选中</p> <p>此复选框控制Enterprise Architect用户是否能够在外部系统中的元素创建讨论。</p>
忽略 SSL 错误	<p>默认值：未选中</p> <p>此复选框控制集成插件是否应忽略由于与外部系统通信而发生的与 SSL 相关的错误。</p>

记录设置

这组设置定义了集成插件如何在其log文件中创建条目。由于集成插件运行用户干预即可运行，因此能够将任何消息或潜在问题的报告写入log文件对他们来说非常重要。但是，必须特别考虑确保写入此类文件不会影响性能，如果log文件持续无限制地增长，这是可能的。期望需要手动干预以确保log文件的大小和数量不会影响性能也是不合理的。由于这些原因，每个集成插件都有自己的一组log文件和配置选项来管理这些log文件。

当前log文件将始终在其文件名末尾包含一个“1”。每当集成插件启动或当前log文件的物理大小达到插件时，都会创建一个新的log文件。发生这种情况时，所有现有文件都将被“翻转”，这意味着它们的文件编号将增加并且新的“filename} 1。log”创建。如果翻转过程导致存在的文件多于“文件计数”设置允许的数量，则这些附加文件将被删除。

物品	描述
文件数	<p>默认值：3</p> <p>此值表示应为当前集成插件保留的log文件的“滚动”数量。</p>

最大文件大小	默认值：1048576 此值表示在创建新log文件之前log文件可以达到的最大大小（以字节数计）。
等级	默认值：警告 此值表示应写入log文件的最大消息级别。级别为：关闭、致命、警告、信息和系统。定义A log级别包括所有较低级别；因此，如果插件设置为信息的log级别，那么所有致命、警告和信息消息都将写入其log文件。
目录	（只读）此字段显示将保存当前数据提供者的log文件的目录的物理位置。

代理设置

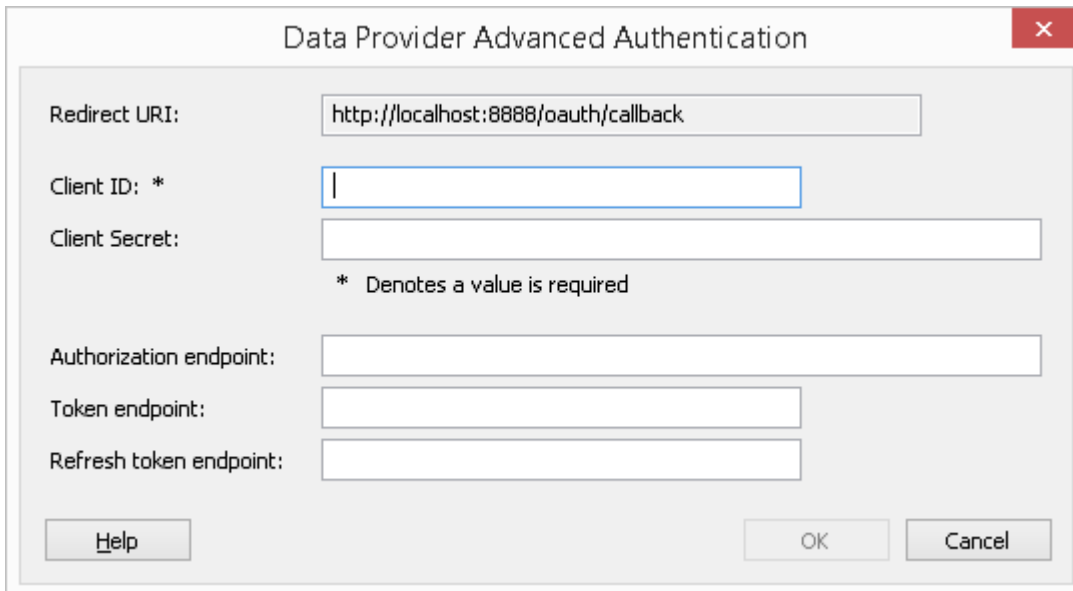
这组设置定义了当集成插件被代理服务器分隔时，它们应该如何与外部数据提供者进行通信。

物品	描述
服务器名称/IP	Proxy服务器的服务器名称（或IP）和端口号；例如，proxyserver.com:3131
旁路	该字段为某些地址提供绕过代理的机制；它接受由分号 (;) 分隔的多个值。例如，192.168.*;*.localdomain.com;www.mydomain.com
用户名称	如果代理服务器需要凭据，则此字段提供用户名。
密码	如果代理服务器需要凭据，则此字段提供用户密码。

数据提供者高级认证

此屏幕允许输入用于当前集成提供者的 OAuth 身份验证的所有设置。

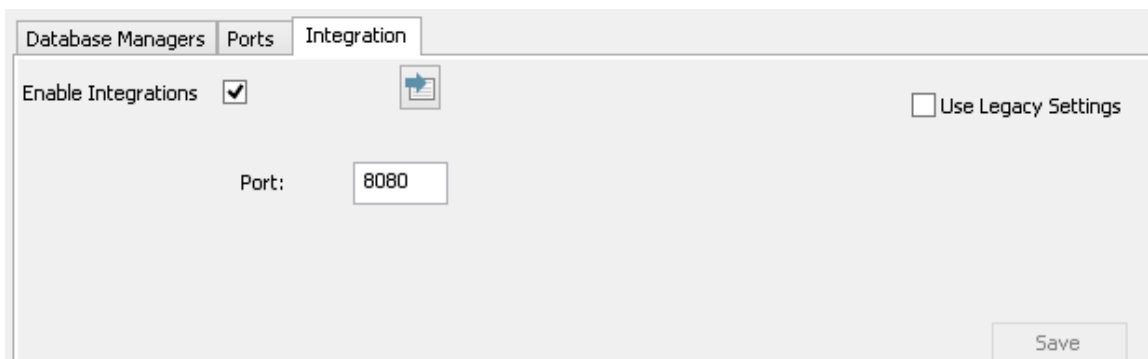
注记：OAuth 仅被部分集成提供商支持。



The dialog box titled "Data Provider Advanced Authentication" contains the following fields and controls:

- Redirect URI:
- Client ID: *
- Client Secret:
- * Denotes a value is required
- Authorization endpoint:
- Token endpoint:
- Refresh token endpoint:
- Buttons: Help, OK, Cancel

集成服务器

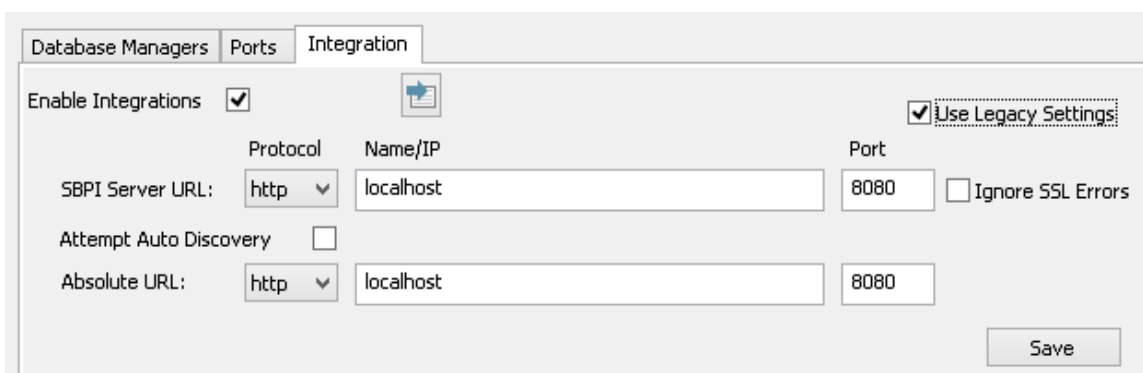


The "Integration" tab in the settings dialog shows:

- Enable Integrations:
- Use Legacy Settings:
- Port:
- Save button

Enterprise Architect对外部数据的所有请求都发送到集成服务器（通过专业云服务器），后者决定哪个数据提供者（或插件）将处理该请求。

为了让大多数用户的集成服务器配置尽可能简单，专业云服务器的 4.1 版本引入了一组简化的选项；但是，如果默认配置不适合您的环境，请选择“使用传统设置”选项以显示所有可用选项。



The "Integration" tab in the settings dialog shows the following configuration:

- Enable Integrations:
- Use Legacy Settings:
- Ignore SSL Errors:
- SBPI Server URL: Protocol: Name/IP: Port:
- Attempt Auto Discovery:
- Absolute URL: Protocol: Name/IP: Port:
- Save button

注记：集成服务器设置仅在选中“启用集成”标志时是必需的。协议、名称和端口的组合必须与托管专业集成服务器（在本例中为专业云服务器）的机器相匹配，以便Enterprise Architect客户端能够与之通信。

物品	描述
重定向 URI	此只读值显示 OAuth 系统将回调到的专业云服务器的 URI。 重要提示：在 OAuth 中定义数据提供者时，需要保存重定向 URI，以便 OAuth 知道它应该与哪个 URI 通信。
客户端编号	该值表示数据提供者在 OAuth 中的唯一标识符，有时也称为“App ID”。
客户秘密	这个值代表一个秘密，只有数据提供者应该知道，就像密码一样必不可少。 在 OAuth 系统中定义数据提供者时，客户端/应用程序 ID 需要定义一个秘密。
授权端点	授权端点是 OAuth 系统的完成 URL，专业云服务器与之通信以确定当前用户是否有权访问当前数据提供者。用户和 OAuth 之间的所有必要通信都发生在用户的浏览器中，专业云服务器会收到结果通知。 注记： 授权端点的 A 不是强制性的，因为专业云服务器有一组将被使用的已知地址。 一个例子是： https://login.microsoft.com/oauth/authorize
令牌端点	在用户被授予对系统的访问权限后，专业云服务器使用令牌端点获取访问令牌。 注记： 令牌端点可以是相对或完整 URL。
刷新令牌端点	刷新令牌端点通常与令牌端点相同，但一些提供商使用不同的 URL 来获取刷新令牌。如果留空，则将使用令牌端点。
确定	此按钮保存所有未决更改并关闭当前屏幕。 它仅在所有必填字段都有值且存在未保存的更改时启用。

演练：ServiceNow集成

本演练帮助您设置和使用专业云服务器的集成功能，将您的Enterprise Architect模型与相应的 ServiceNow库表集成。

使用与 ServiceNow 的集成，您可以：

- 将Enterprise Architect元素与 ServiceNow 中的对应元素链接
- 基于 ServiceNow 项创建Enterprise Architect元素，反之亦然
- 执行可追溯性分析

先决条件

为了设置 ServiceNow 和Enterprise Architect之间的集成，您必须首先：

- 使用带有集成插件组件的专业云服务器的许可安装
- 在托管您的Enterprise Architect模型的每个数据库管理系统的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项
- 拥有 PCS 服务器和 ServiceNow 之间的网络访问权限
- 拥有访问外部数据的用户凭据
- 正在使用Enterprise Architect企业、统一或终极版 v14 或更高版本

我们假设集成服务器和运行集成插件与专业云服务器在同一台机器上运行。

最低要求的权限

这些库表需要读取权限：

- **sys_filter** - 列出决定在Enterprise Architect中显示什么的自定义过滤器
- **sys_db_object** - 库表列表，按使用的主库表对过滤器进行分组
- **sys_dictionary** - 用于表字段的显示名称
- 自定义过滤器使用的任何表

要启用“推送”同步，用户还需要对库表的写入权限。

可以通过在表上添加访问控件（ACL）规则来启用读取访问。

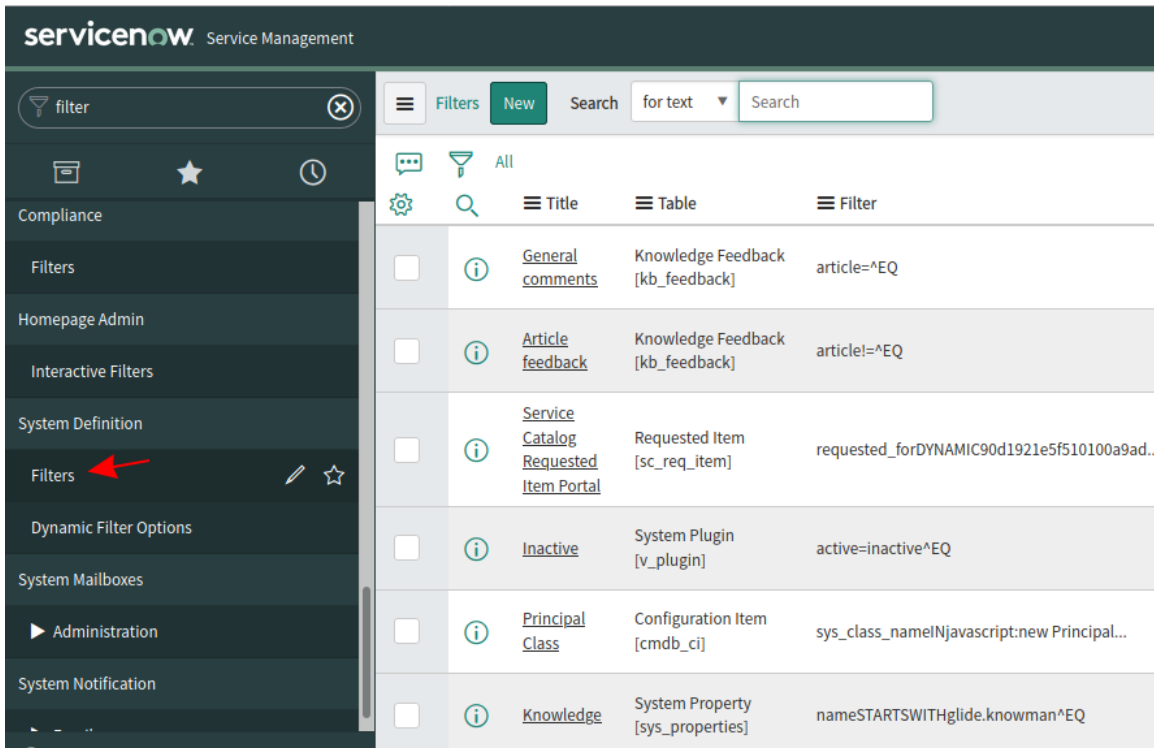
A需要第二条规则来启用对表中所有字段的读取访问权限（例如，表.*）。

访问控件规则应具有所需的角色分配，然后将该角色分配给组或个人用户。

注记：在某些库表上，需要将角色从系统管理员提升“到 安全管理员 - security_admin”才能编辑访问控件。

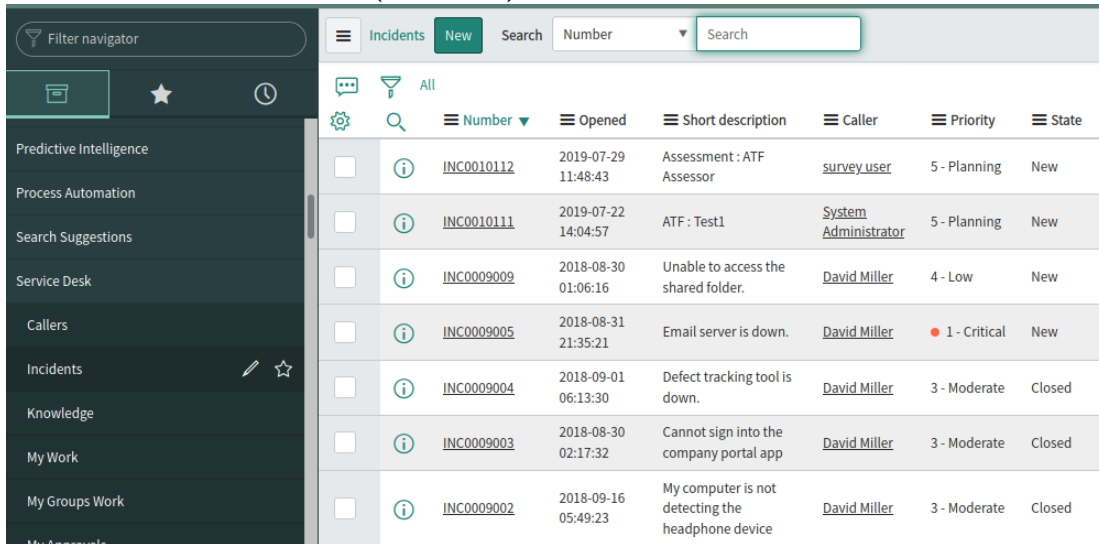
定义 ServiceNow表过滤器

ServiceNow 集成允许通过保存的表过滤器访问项目。默认情况下，ServiceNow 通常包含许多预定义的全局过滤器。您可以使用 [系统定义](#) | “浏览 ServiceNow 中所有定义的过滤器。过滤器选项。

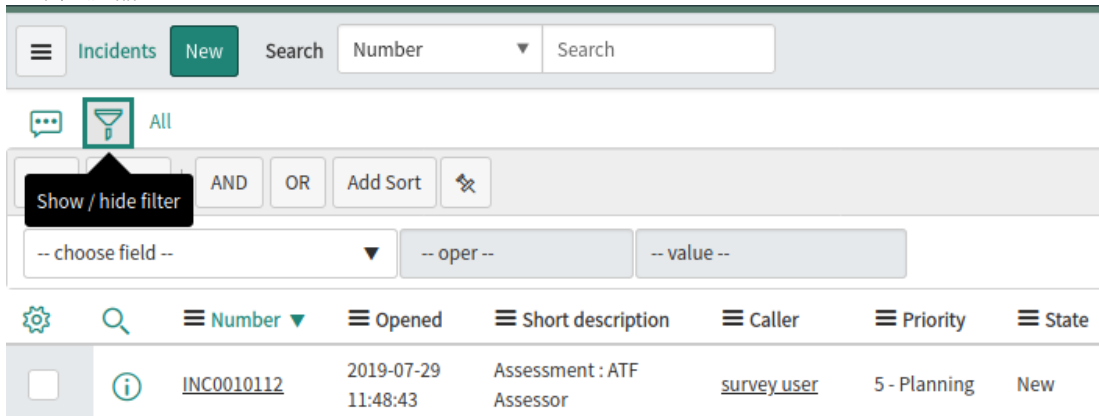


您可以按照以下步骤在 ServiceNow 中创建新过滤器：

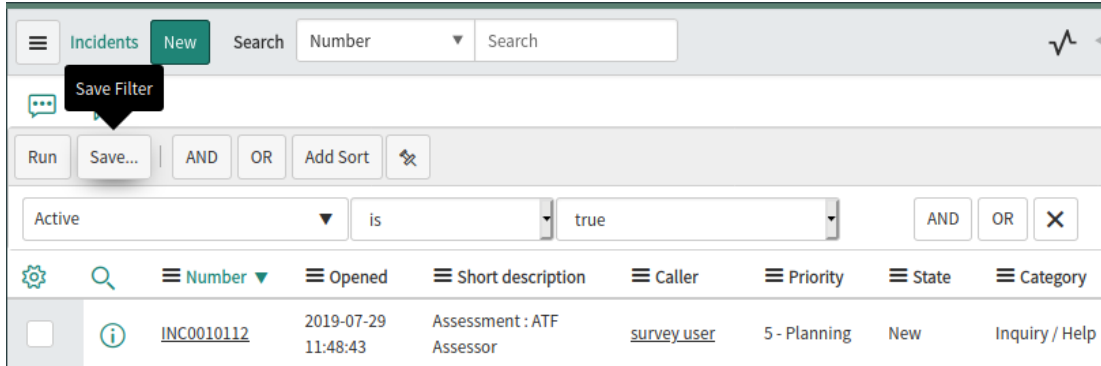
1. 在 ServiceNow 中找到相关数据（例如事件）



2. 显示过滤器栏。



- 3. 调整过滤器选项。
- 4. 保存过滤器。

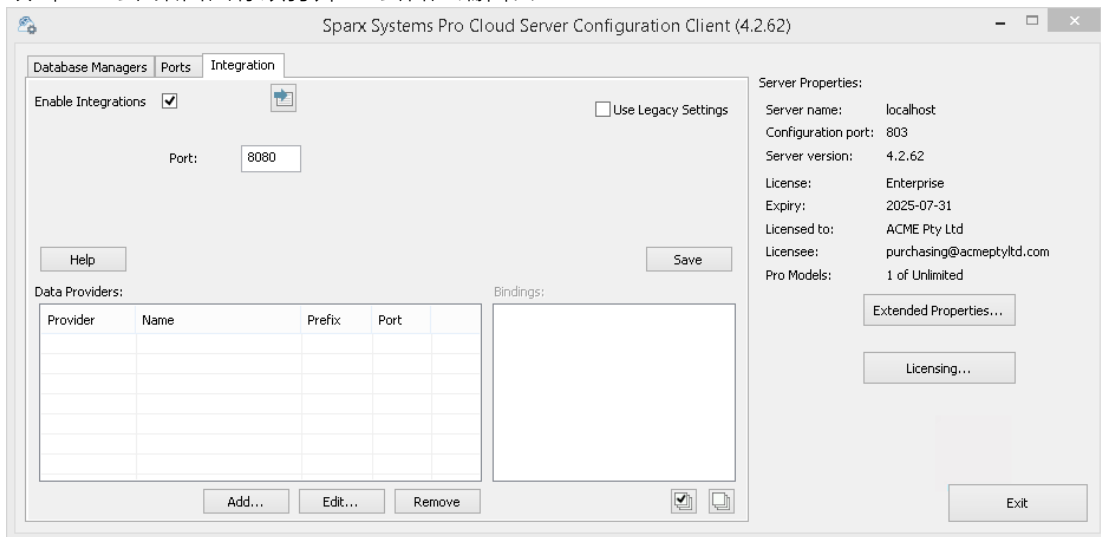


- 5. 专业云服务器的 ServiceNow 集成现在可以访问此过滤器（及其返回的项目）。

向专业云服务器和Enterprise Architect识别 ServiceNow

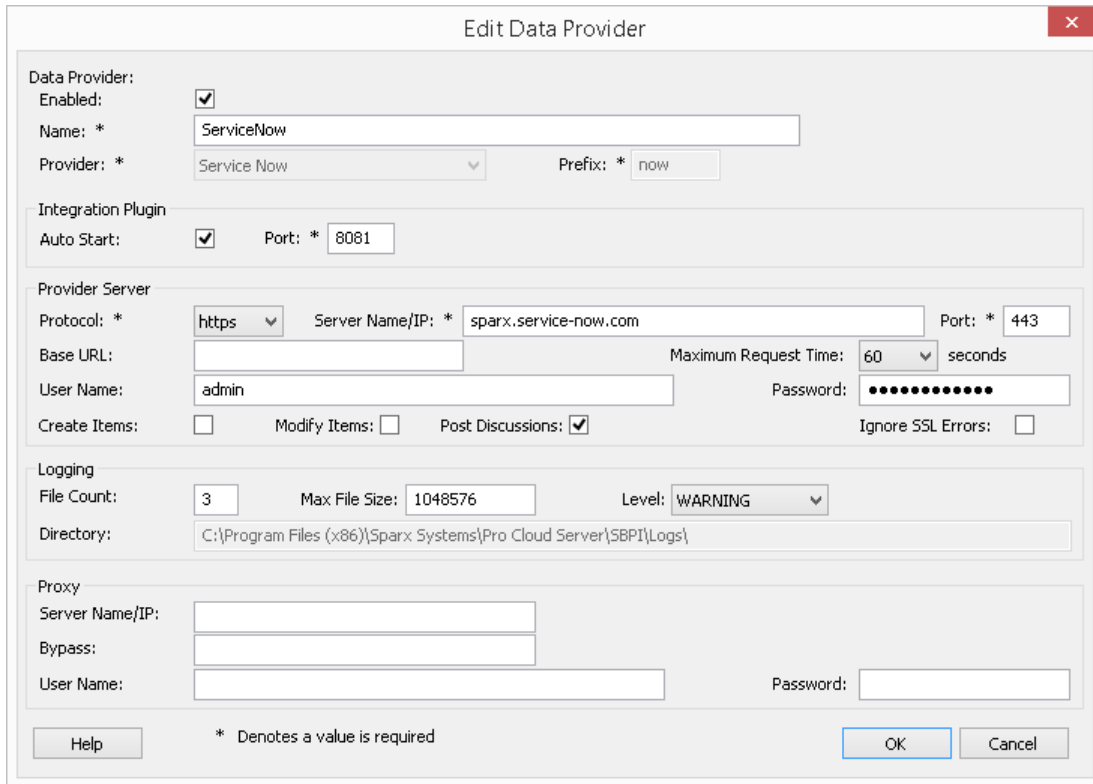
完成以下步骤：

- 1. 双击“Pro云”桌面图标以打开Pro云客户端窗口。



- 2. 选择“集成”选项卡并选择“启用集成”复选框。在“端口”字段中，输入集成服务器正在侦听来自专业云服务器的请求的端口号。
- 3. 单击添加按钮。将显示“添加数据提供者”对话框。

4. 选中“启用”复选框。
5. 在“名称”字段中，键入适当的连接名称，例如“ServiceNow”。
6. 在“提供者”字段中，单击下拉箭头并选择“立即服务”；这也会自动将“现在”添加到“前缀”字段中。
7. 在“集成插件”面板的“插件端口”将“端口”字段中，键入集成服务器用于与端口插件的端口号。
8. 在“协议”字段中，单击下拉箭头并选择“https”。
9. 在“服务器名称/IP”字段中，键入 ServiceNow 插件将用于与 ServiceNow 通信的服务器名称或 IP 地址。
10. 在“端口”字段中输入“443”。
11. 离开“基地”默认 ServiceNow 配置的 URL 字段为空白。非默认 ServiceNow 服务器配置可能需要该字段。注记将步骤 8 至 11 中的字段连接起来形成一个网址；即：`<protocol>://<server>:<port>/baseURL`（如果字段不为空，则包括 baseURL）。
12. 如果您将“用户”和“密码”字段留空，则会提示每个 Enterprise Architect 用户输入他们的个人 ServiceNow 凭据，这可以提供更好的可用性。
13. 如果您更喜欢在“用户名称”和“密码”字段中设置值，则这些值将组合使用以访问 ServiceNow 中的数据。ServiceNow 安装的所有当前 Enterprise Architect 用户将使用同一组凭据来读取外部数据。
14. 根据需要选择“创建项”和/或“修改项”复选框，以允许 ServiceNow 插件的用户使用 Enterprise Architect 在 ServiceNow 中创建和/或更新项目。
15. 不选中“忽略 SSL 错误”选项。



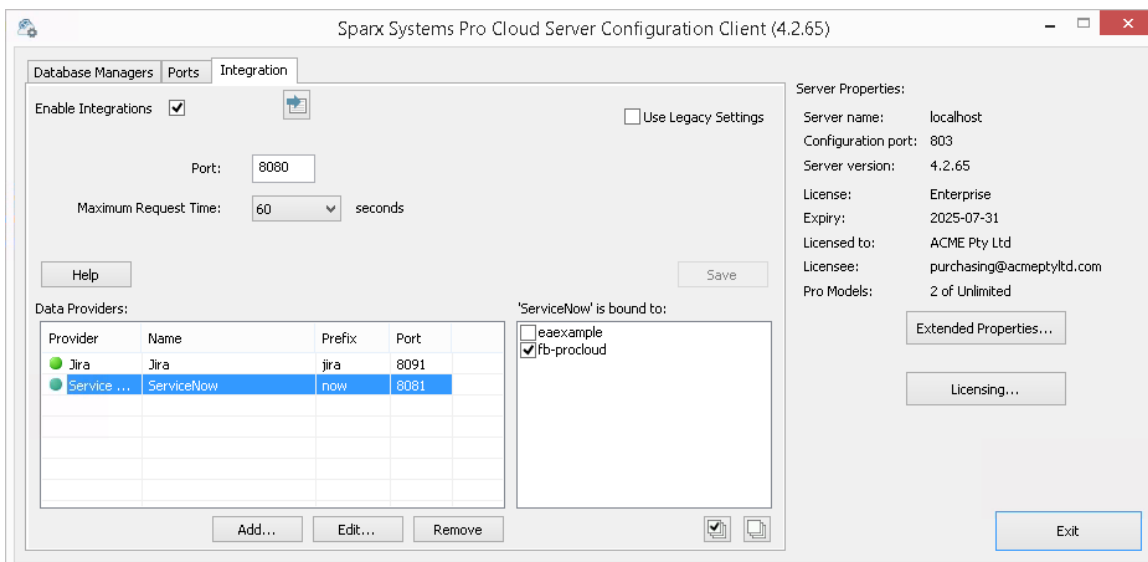
16. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的“集成”选项卡。

为 ServiceNow 识别Enterprise Architect模型

在专业云服务器配置客户端窗口的“集成”选项卡上，您现在将看到：

- 在屏幕的左下方，“数据提供者”面板将 ServiceNow 列为数据提供者
- 在屏幕的右下方，“ServiceNow”绑定到：面板显示来自启用 Pro 的数据库管理员的Enterprise Architect模型的检查列表

您可以单击要绑定到 ServiceNow 的每个Enterprise Architect模型的复选框，或单击  以一次选择所有它们。



单击退出按钮。

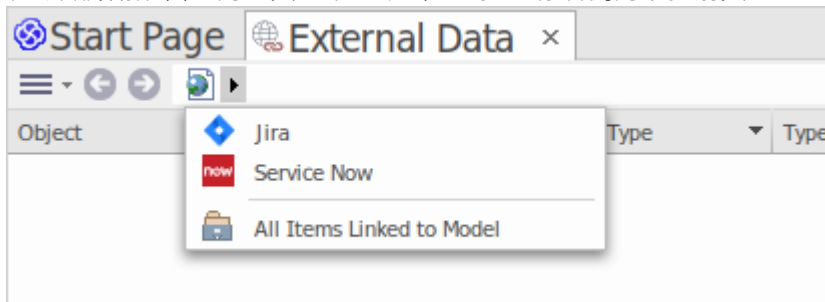
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 ServiceNow 的集成。

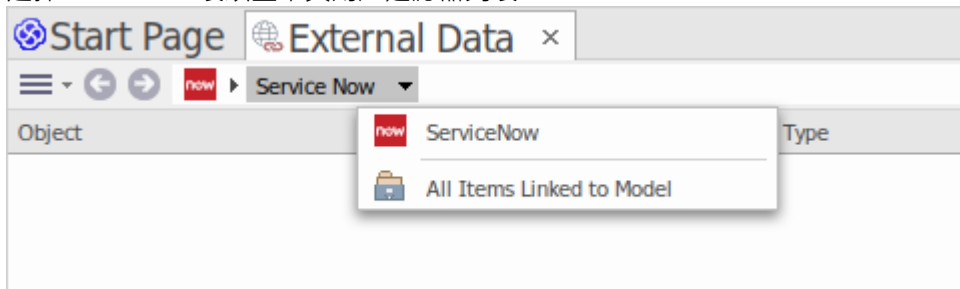
测试集成

在Enterprise Architect中打开您已识别为绑定到 ServiceNow 的模型之一。

1. 选择功能区选项'特定>工具>系统集成'。
2. 在 外部数据"窗口的工具栏中，单击 地球"图标右侧的下拉箭头。



3. 从列表中选择 立即服务"；这会将提供程序添加到工具栏中的面包屑路径。
4. 单击面包屑路径中的 "ServiceNow"以显示已配置并绑定到此模型ServiceNow 提供程序列表。如果未列出任何内容，则没有提供程序绑定到此模型。
5. 单击已设置并绑定到此模型提供者，以显示现在选择以在此Enterprise Architect模型中使用的 ServiceNow库表列表。
6. 选择 ServiceNow表以显示其用户过滤器列表。



如果此处未显示任何内容，则您可能没有足够的权限查看库表。但是，如果您还没有为您创建任何用户过滤器，则该列表仅包含 []"或信息消息。

(如果您看不到 ServiceNow 或 ServiceNow库表列表，请检查本主题中的过程，或[Integration Plug-ins](#)部分中更广泛和更详细的主题，看看是否有您遗漏的步骤或不存在的先决条件满意。您也可以查看[Troubleshooting](#)帮助主题。)

数据映射

为了让 ServiceNow 和Enterprise Architect顺利交换数据项，您必须审阅并在必要时进行编辑：

- 哪种类型的 ServiceNow 项对应于哪种类型的Enterprise Architect元素，以及
- 属性项的哪些属性对应于Enterprise Architect元素的哪些属性

这些任务在[Configuration](#)帮助主题中有完整的解释。

审阅ServiceNow Data

当您测试Enterprise Architect和 ServiceNow 之间的通信时，在“外部数据”窗口中，您打开了面包屑跟踪以显示 ServiceNow 项目的列表。

当您选择其中一个项目时，再次单击下拉箭头并从列表中选择过滤器，然后在窗口主体中列出该过滤器中的项目。

Object	M...	Type
I can't launch my VPN client since the last software upda...		Incident
Rain is leaking on main DNS Server		Incident
How do I create a sub-folder		Incident
Sales forecast spreadsheet is READ ONLY		Incident
Can't launch 64-bit Windows 7 virtual machine		Incident
I need a replacement iPhone, please		Incident
Need to add more memory to laptop		Incident
Please remove the latest hotfix from my PC		Incident

如果此列表太长，您可以单击窗口工具栏中的“汉堡包”图标，选择以下选项：

- “链接项”仅显示链接到Enterprise Architect元素的那些 ServiceNow 项
- “元素接到当前上下文”仅显示链接到当前选定的Enterprise Architect的 ServiceNow项
- “未链接的项目”仅显示尚未链接到Enterprise Architect元素的那些 ServiceNow 项目

如有必要，选择适当的选项，然后浏览所选 ServiceNow 项目的内容。每个属性窗口、笔记窗口和检验员窗口都有单独的选项卡或窗口版本以显示所选项目的任何属性、笔记和特征（分别）。这些在 [External Item Details](#) 帮助主题。

使用 ServiceNow 项目和Enterprise Architect元素

在Enterprise Architect和 ServiceNow 之间创建工作通信后，您可以：

- 在Enterprise Architect中创建一个链接到 ServiceNow 项的新元素
- 在Enterprise Architect中创建一个链接到元素的新 ServiceNow 项
- 将Enterprise Architect中的现有元素链接到 ServiceNow 项
- 使用对Enterprise Architect元素的更改来更新 ServiceNow 项
- 通过对 ServiceNow 项的更改更新Enterprise Architect元素
- 使用对 ServiceNow 项目的任何更改来更新所有链接的Enterprise Architect元素

这些操作都在[Linking Items](#)帮助主题中进行了解释。

演练：Jira集成

本演练可帮助您设置和使用专业云服务器的集成功能，通过云安装选项将您的Enterprise Architect模型与相应的Jira Software项目集成。

使用与Jira的集成，您可以：

- 将Enterprise Architect元素与Jira中的对应元素链接起来
- 基于Jira项目创建Enterprise Architect元素，反之亦然
- 在需要时同步Enterprise Architect和Jira之间的更改
- 执行可追溯性分析
- 从Enterprise Architect向Jira项目添加评论

当您的系统上安装了专业云服务器时，它会提供一个提示来安装可选的集成插件组件；默认情况下，该组件安装在C:\Program Files (x86)\Sparx Systems\专业云服务器\SBPI\文件夹中（假设使用的是64位计算机）。集成插件组件包括：

- 集成（SBPI）服务器（sbpi.exe）
- 许多集成工具中的每一个的插件.exe文件，包括用于Jira的工具（JiraSbpi.exe）

JiraSbpi.exe插件与Atlassian的问题跟踪系统交互。它能够列出用户最喜欢的过滤器（也称为星标过滤器）。然后每个过滤器将列出过滤器返回的所有Jira项目。在Enterprise Architect请求集成插件应该连接到它们之前，这些过滤器在Jira中的收藏夹文件夹中为每个登录的用户配置。您只能查看属于过滤器的那些Jira项目，因此请在Jira中创建适当的过滤器。例如，如果您想将Enterprise Architect需求链接到Jira用户案例，请创建一个过滤器来显示相关用户案例。

Jira本身有两个安装选项，Jira云和Jira服务器。Jira服务器选项正在逐步淘汰，因此这些过程假定您正在使用Jira云。但是，如果您正在使用Jira服务器，则过程中有一些注释可以适应这种情况。

访问令牌

如果Jira托管在Atlassian服务器上而不是本地托管，则它需要使用在Atlassian站点上创建的TokenID，用于登录Atlassian的单个预定义用户集。请参阅Atlassian网站页面上的令牌-一般审查访问审阅(https)和关于创建令牌的讨论(https云/api-tokens-938839638.html)。请参阅下面“了解更多”下的链接。

先决条件

为了在Jira和Enterprise Architect之间建立集成，您必须首先：

- 如前所述，使用带有集成插件组件的专业云服务器的许可安装
- 在托管您的Enterprise Architect模型的每个数据库管理员系统的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项
- 在PCS服务器和托管外部Jira安装及其数据的Jira服务器之间具有物理网络访问权限
- 拥有访问外部数据的用户凭据
- 正如刚才讨论的那样，在Jira中设置一个包含适当过滤器的“收藏夹”文件夹
- 正在使用Enterprise Architect企业、统一或终极版v14或更高版本

我们将假设：

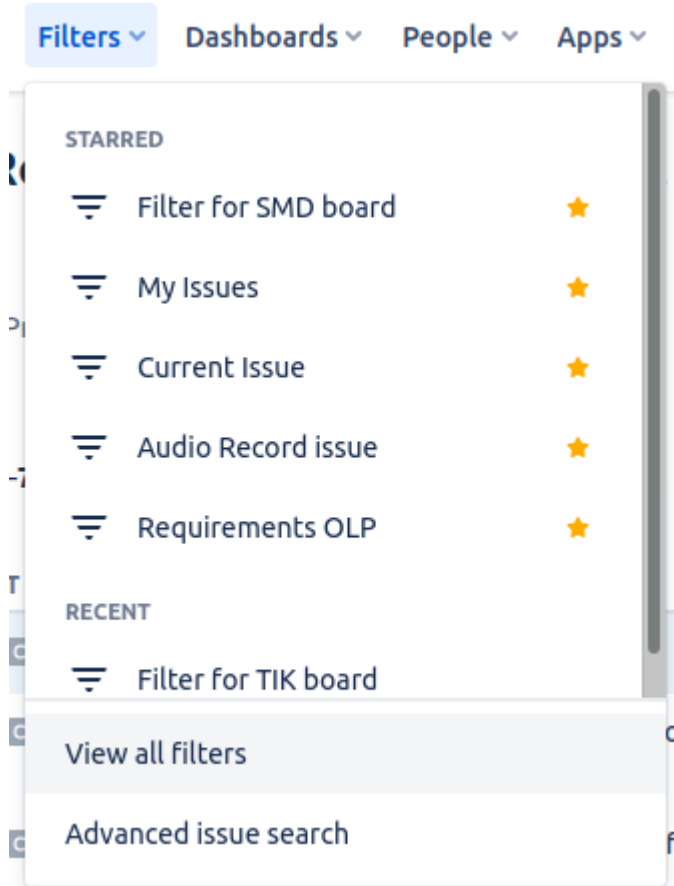
- 集成服务器和运行集成插件与专业云服务器在同一台机器上运行
- 您正在使用专业云服务器配置客户端中的“集成”选项卡定义集成服务器选项，而不是通过手动操作注册表或配置文件或使用WebConfig界面

在 Jira 中定义星标过滤器

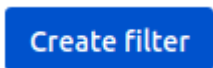
专业云服务器的 Jira 集成允许访问由 Jira 加星标过滤器返回的项目。在使用集成之前，您应该确保已设置一些加星标的过滤器以返回您希望在Enterprise Architect中可访问的项目。

要在 Jira 中定义星标过滤器，请执行以下步骤：

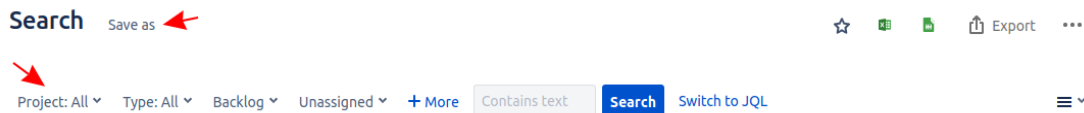
1. 在 Jira 中，选择 过滤器”下拉菜单，然后选择 视图所有过滤器”。



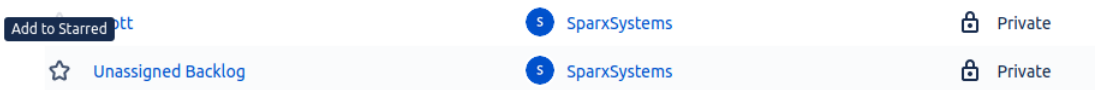
2. 在过滤器列表的右上角，单击 创建过滤器”按钮。



3. 调整过滤器/搜索设置（例如项目和类型），然后使用 另存为”选项保存新过滤器。



4. 现在返回所有过滤器列表（ 过滤器|视图所有过滤器”）。在过滤器列表中，您可以使用星形图标将过滤器添加到已加星标的过滤器中。

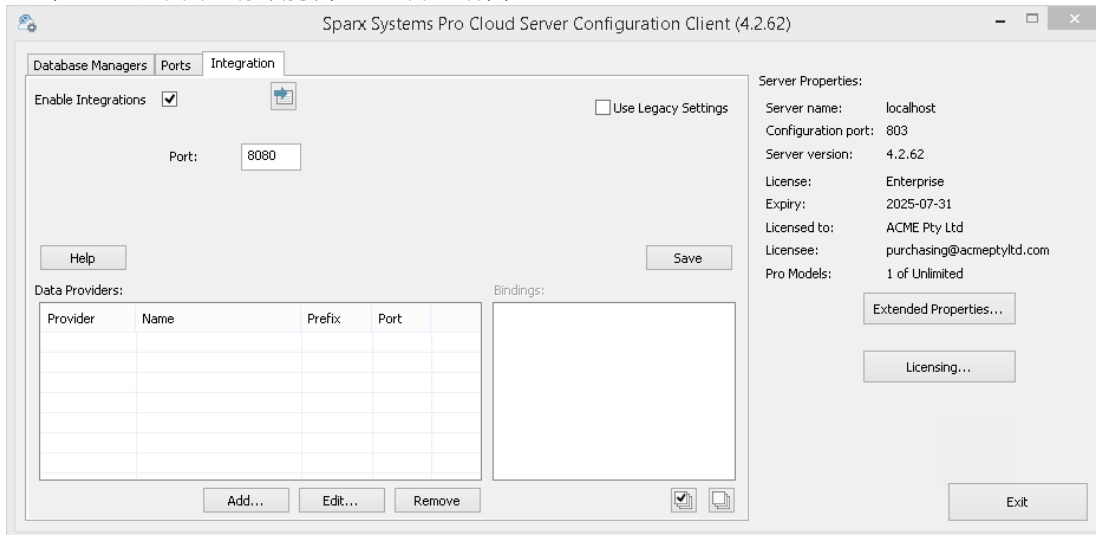


5. 专业云服务器的现在集成现在应该可以访问此过滤器（及其返回的项目）

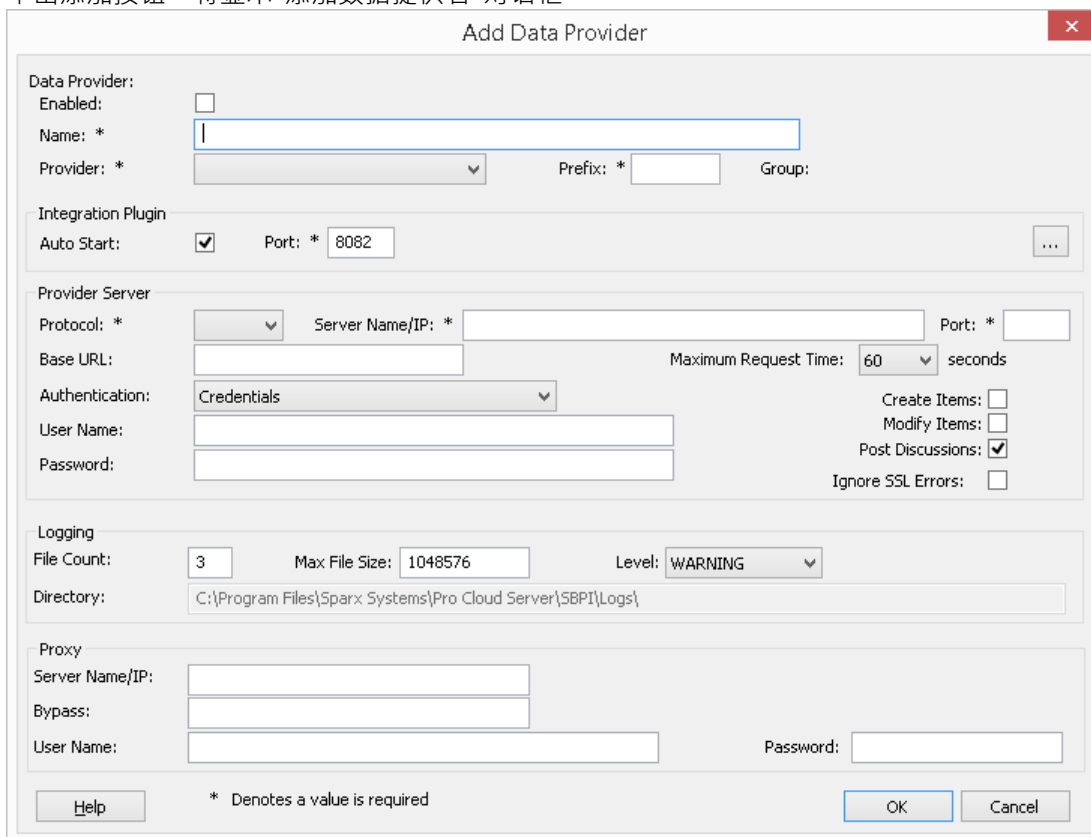
将 Jira 识别为专业云服务器和Enterprise Architect

完成以下步骤：

1. 双击“Pro云”桌面图标以打开Pro云客户端窗口。



2. 选择“集成”选项卡并选择“启用集成”复选框。在“端口”字段中，输入集成服务器正在侦听来自专业云服务器的请求的端口号。
3. 单击添加按钮。将显示“添加数据提供者”对话框。



4. 选中“启用”复选框。
5. 在“名称”字段中，输入适当的连接名称，例如“Jira”。
6. 在“提供者”字段中，单击下拉箭头并选择“Jira”；这也会自动将“Jira”添加到“前缀”字段中。
7. 在“集成插件”面板的“插件端口”字段中，键入集成服务器用于与端口通信的端口号。
8. 在“协议”字段中，单击下拉箭头并选择“Jira”插件将用于与“Jira”通信的协议。
 - 对于“Jira云”默认安装，这必须是“HTTPS”
 - 对于“Jira服务器”安装，默认值为“HTTP”

9. 在“服务器名称/IP”字段中，键入 Jira 插件将用于与 Jira 通信的服务器名称或 IP 地址。
 - 对于 Jira 云默认安装，这必须是 <account>.atlassian.net
 - 对于 Jira 服务器安装，默认值为 <服务器或主机的名称>
10. 在“插件端口”字段中，输入端口将用于与 Jira 通信的端口号。
 - 对于 Jira 云默认安装，这必须是 '443'
 - 对于 Jira 服务器安装，默认值为 '8080'
11. 离开“基地”对于 Jira 云和默认的 Jira 服务器安装，URL 字段为空白。非默认 Jira 服务器配置可能需要该字段。注记将步骤 8 至 11 中的字段连接起来形成一个网址；即：<protocol>://<server>:<port>/baseURL（如果字段不为空，则包括 baseURL）。
12. 如果您将“用户”和“密码”字段留空，则会提示每个 Enterprise Architect 用户输入他们的个人 Jira 凭据，这样可以提供更好的可用性。
13. 如果您更喜欢在“用户名称”和“密码”字段中设置值，这些值将组合使用以访问 Jira 中的数据。Jira 安装的所有当前 Enterprise Architect 用户将使用同一组凭据来读取外部数据。
14. 根据需要选择“创建项”、“修改项”和/或“发布讨论”复选框，以允许 Jira 插件的用户使用 Enterprise Architect 在 Jira 中创建和/或更新项目和/或创建讨论帖子。
15. 如果您希望 Jira 插件忽略由于与 Jira 通信而发生的与 SSL 相关的错误，请选中“忽略 SSL 错误”复选框。

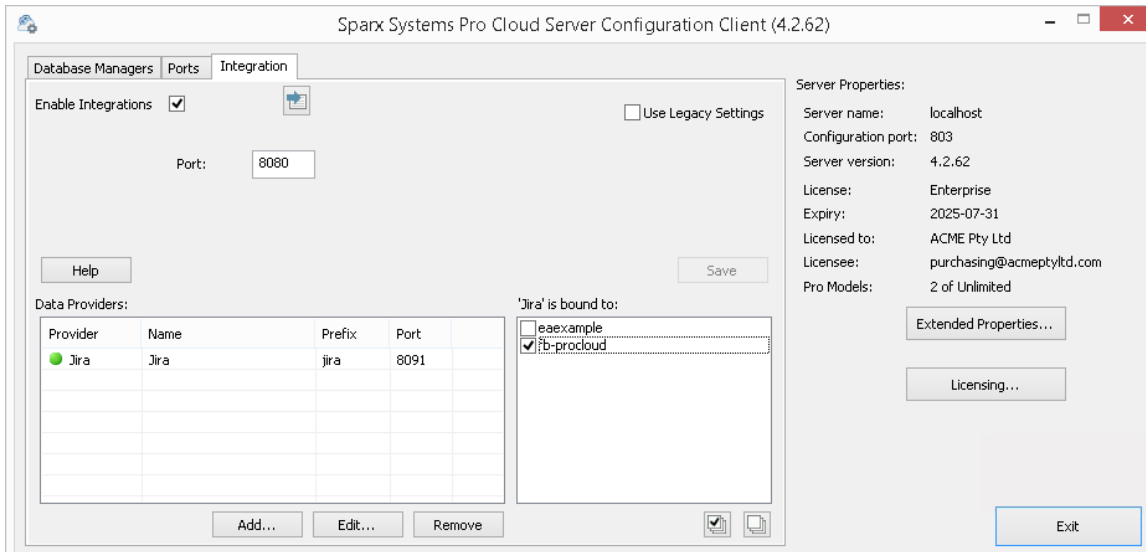
16. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的“集成”选项卡。

为 Jira 识别 Enterprise Architect 模型

在专业云服务器配置客户端窗口的“集成”选项卡上，您现在将看到：

- 在屏幕的左下方，“数据提供者”面板将 Jira 列为数据提供者
- 在屏幕的右下方，“Jira”绑定到：面板显示来自启用 Pro 的数据库管理员的 Enterprise Architect 模型检查列表

您可以单击要绑定到 Jira 的每个 Enterprise Architect 模型的复选框，或单击  以一次选择所有它们。



单击退出按钮。

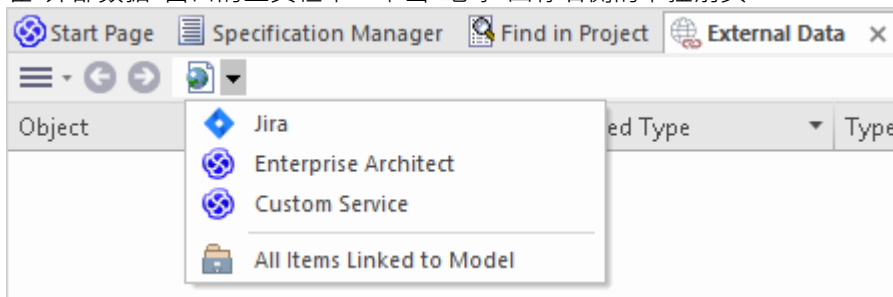
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 Jira 的集成。

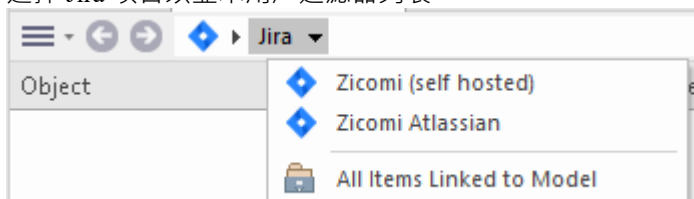
测试集成

在Enterprise Architect中打开您在列表中选择的模型之一：“Jira”绑定到。S参见上图。

1. 选择功能区选项特定>工具>系统集成。
2. 在“外部数据”窗口的工具栏中，单击“地球”图标右侧的下拉箭头。



3. 从列表中选择“Jira”；这会将提供程序添加到工具栏中的面包屑路径。
4. 单击面包屑路径中的“Jira”以显示已配置并绑定到此模型的 Jira 供应商列表。如果未列出任何内容，则没有提供程序绑定到此模型。
5. 单击已设置并绑定到此模型提供者，以显示 Jira 项目列表，这些项目现在可供选择以在此Enterprise Architect模型中使用。
6. 选择 Jira 项目以显示用户过滤器列表。



如果此处未显示任何内容，则您可能没有足够的权限来查看项目。但是，如果您还没有为您创建任何用户过滤器，则该列表仅包含“”或信息消息。

(如果您看不到 Jira 或 Jira 项目列表，请检查本主题中的过程，或[Integration Plug-ins](#)部分中更广泛和更详细的主题，看看是否有您遗漏的步骤或不满足的先决条件。您也可以查看[Troubleshooting](#)帮助主题。)

数据映射

为了让 Jira 和 Enterprise Architect 顺利交换数据项，您必须审阅并在必要时进行编辑：

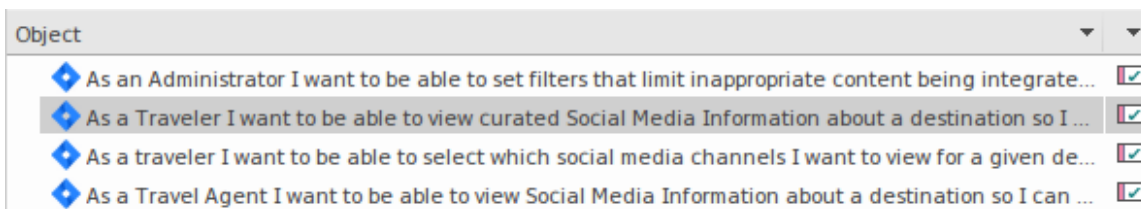
- 哪种类型的 Jira 项对应于哪种类型的 Enterprise Architect 元素，以及
- 属性项的哪些属性对应于 Enterprise Architect 元素的哪些属性。

这些任务在配置帮助主题中有完整的解释，该主题使用 [Configuration](#) 作为示例数据提供者。

审阅 Jira Data

当您测试 Enterprise Architect 和 Jira 之间的通信时，在 External Data 窗口中，您打开了面包屑跟踪以显示 Jira 项目列表。

当您选择其中一个项目时，再次单击下拉箭头并从列表中选择过滤器，然后在窗口主体中列出该过滤器中的项目。



如果此列表太长，您可以单击窗口工具栏中的“汉堡包”图标，选择以下选项：

- “链接项”仅显示那些链接到 Enterprise Architect 元素的 Jira 项目
- “链接到当前上下文的元素项”仅显示链接到当前选定的 Enterprise Architect 的 Jira 项目
- “未链接的项目”仅显示尚未链接到 Enterprise Architect 元素的那些 Jira 项目

如有必要，选择适当的选项，然后浏览所选 Jira 项目的内容。讨论和审阅窗口的每个属性窗口、笔记窗口、检验员窗口和讨论选项卡都有单独的选项卡或窗口版本，以显示所选项目的任何属性、笔记、特征和讨论（分别）。这些在 [External Item Details](#) 帮助主题。

使用 Jira 项目和 Enterprise Architect 元素

在 Enterprise Architect 和 Jira 之间创建工作通信后，您可以：

- 在 Enterprise Architect 中创建一个链接到 Jira 项目的新元素
- 在 Enterprise Architect 中创建一个链接到元素的新 Jira 项目
- 将 Enterprise Architect 中的现有元素链接到 Jira 项目
- 使用 Enterprise Architect 元素的更改更新 Jira 项
- 使用对 Jira 项的更改来更新 Enterprise Architect 元素
- 使用对 Jira 项目的任何更改更新所有链接的 Enterprise Architect 元素

这些操作都在 [Linking Items](#) 帮助主题中进行了说明，该主题使用帮助作为示例数据提供者。

演练：DevOps集成

本演练可帮助您设置和使用专业云服务器的 DevOps/团队基础服务器集成。

注记：2019 年微软团队基础服务器更名为 Azure DevOps。更具体地说，云托管的 Visual Studio Team Services (VSTS) 已重命名为 Azure DevOps Services，本地 Visual Studio 团队基础服务器(TFS) 已重命名为 Azure DevOps 服务器。

专业云服务器集成与 Azure DevOps (云托管和本地) 和团队基础服务器(TFS) 兼容。本演练将主要使用当前名称“DevOps”，但在专业云服务器和Enterprise Architect中，该插件通常被称为团队基础服务器或 TFS。

使用与 DevOps 的集成，您可以：

- 将Enterprise Architect元素与 DevOps 中的对应元素联系起来
- 基于 DevOps 项创建Enterprise Architect元素
- 在需要时同步Enterprise Architect和 DevOps 之间的更改
- 执行可追溯性分析

先决条件

为了在 DevOps 和Enterprise Architect之间建立集成，您必须首先：

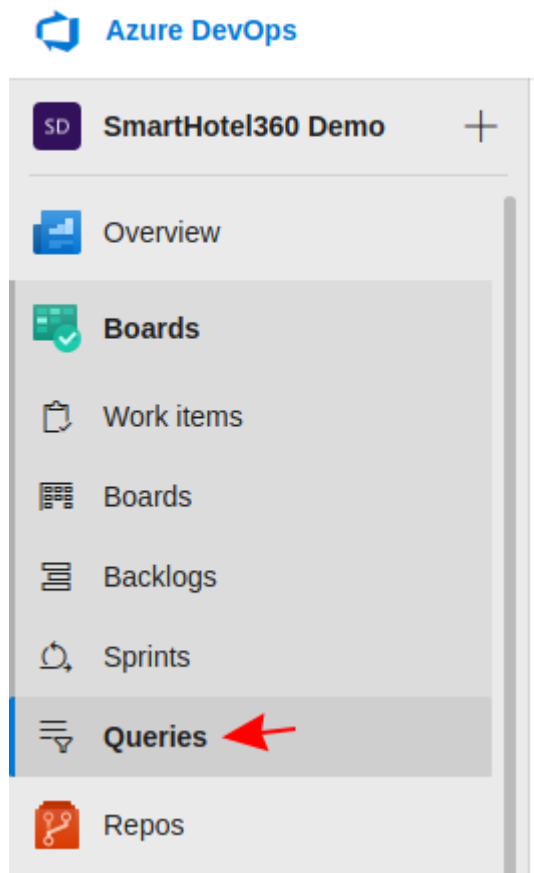
- 使用带有集成插件组件的专业云服务器的许可安装
- 在托管您的Enterprise Architect模型的每个数据库管理员系统的专业云服务器配置中选择“启用专业特征 (OSLC、WebEA和集成)”选项
- 拥有 PCS 服务器和 DevOps 之间的网络访问权限
- 拥有访问外部数据的用户凭据
- 正在使用Enterprise Architect企业、统一或终极版 v14 或更高版本

我们将假设集成服务器和运行集成插件与专业云服务器在同一台机器上运行

定义 DevOps 查询

DevOps 集成允许通过在 DevOps 中定义的查询访问工作项。您可以查看现有查询并创建新查询，如下所述。

1. 登录到 Azure DevOps 并选择您的项目。
2. 从左侧面板中选择“Boards”，然后选择“Queries”。



3. 要查看 DevOps 集成可访问的所有查询，请单击“全部”。查询将分为“我的查询”和“共享查询”。

Queries

Favorites **All** | + New query | New folder | Import Work Items

Title

My Queries

Assigned to me

Followed work items

Shared Queries

All Items_WI

BacklogBoard WI

4. 可以A “新查询”按钮创建新查询。

+ New query

5. 调整查询的过滤选项后，单击“保存查询”按钮以保存查询。

Queries > My Queries

Results **Editor** Charts | ▶ Run query + New ▾ 📁 Save query ↶ Revert chType of query Flat list of work items

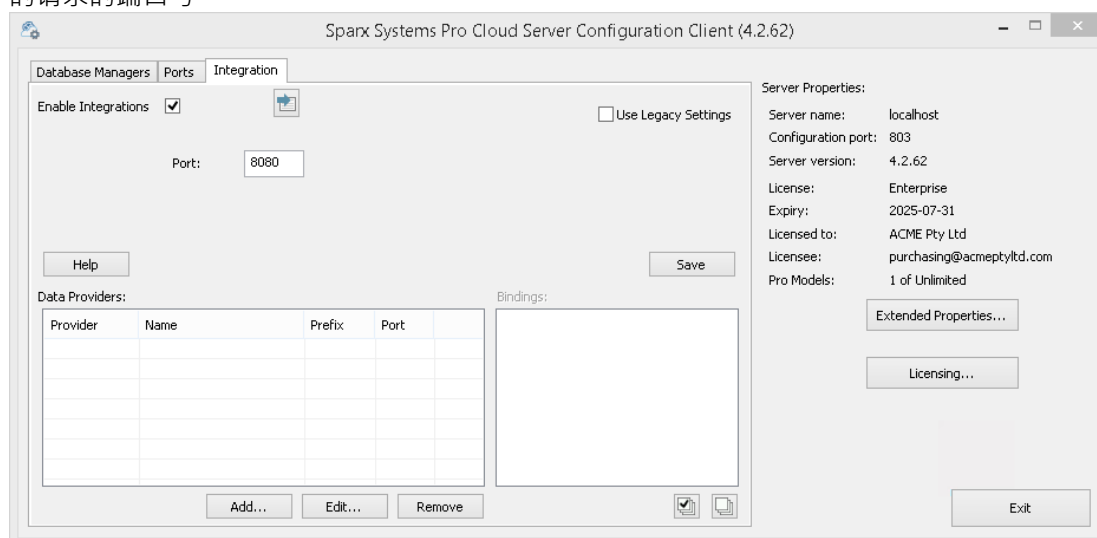
Filters for top level work items

	And/Or	Field*	Operator
+ X <input type="checkbox"/>		Work Item Type	=
+ X <input type="checkbox"/>	And ▾	State	=
+ Add new clause			

向专业云服务器和Enterprise Architect确定 DevOps

完成以下步骤：

1. 双击“Pro云”桌面图标以打开Pro云客户端窗口。
2. 选择“集成”选项卡并选择“启用集成”复选框。在“端口”字段中，输入集成服务器正在侦听来自专业云服务器的请求的端口号。



3. 单击添加按钮。将显示“添加数据提供者”对话框。

4. 选中“启用”复选框。
5. 在“名称”字段中，键入适当的连接名称，例如“DevOps”。
6. 在“提供者”字段中，单击下拉箭头并选择“团队基础服务器”；这也会自动将“tfs”添加到“前缀”字段中。
7. 在“集成插件”面板的“插件集成服务器端口”端口号。
8. 如果您使用云托管的“Azure DevOps Services”，请填写提供者服务器字段，如下所示：

协议：https

服务器名称/IP：dev.azure.com

端口：443

基本 URL：输入 DevOps 组织名称。这可以在 DevOps URL 的末尾找到。例如，如果您的 url 是 https，则在“基本 URL”字段中输入“org1”。

用户名称：将此字段留空。使用 Enterprise Architect 中的集成时，将提示用户输入凭据。

密码：将此字段留空。

创建项/修改项/发布讨论：根据需要启用这些复选框，以允许 DevOps 插件的用户使用 Enterprise Architect 在 DevOps 中创建和/或更新项目和/或创建讨论帖子。

忽略 SSL 错误：不选中此选项。

如果您使用的是本地 **“Azure DevOps服务器”/TFS**，请填写提供者服务器字段，如下所示：

协议：选择 http 或 https

服务器名称/IP：输入您的服务器名称或 IP 地址

端口：输入 DevOps/TFS 配置使用的端口号。

Base URL：输入 DevOps/TFS url 的最后一个组成部分（服务器名称/端口之后的所有内容）。E.g. tfs/DefaultCollection

用户名称：将此字段留空。当使用Enterprise Architect中的集成时，将提示用户输入他们的用户名。

密码：将此字段留空。使用Enterprise Architect中的集成时，将提示用户输入密码。在这种情况下，密码“是指个人访问令牌 (PAT)。个人访问令牌是在 DevOps 中通过 设置 | 个人访问令牌的选项。有关详细信息，请参阅 DevOps 文档。

如果您更喜欢在“用户名称”和“密码”字段中设置值，这些值将组合使用以访问 DevOps 中的数据；但是，DevOps 安装的所有当前Enterprise Architect用户将使用同一组凭据来读取外部数据。

创建项/修改项/发布讨论：根据需要启用这些复选框，以允许 DevOps插件的用户使用Enterprise Architect在 DevOps 中创建和/或更新项目和/或创建讨论帖子。

忽略 SSL 错误：不选中此选项。

注记 · Protocol、服务器、端口和 Base URL 字段被连接起来形成一个 web 地址；即：
`<protocol>://<server>:<port>/baseURL`（如果字段不为空，则包括 baseURL）。

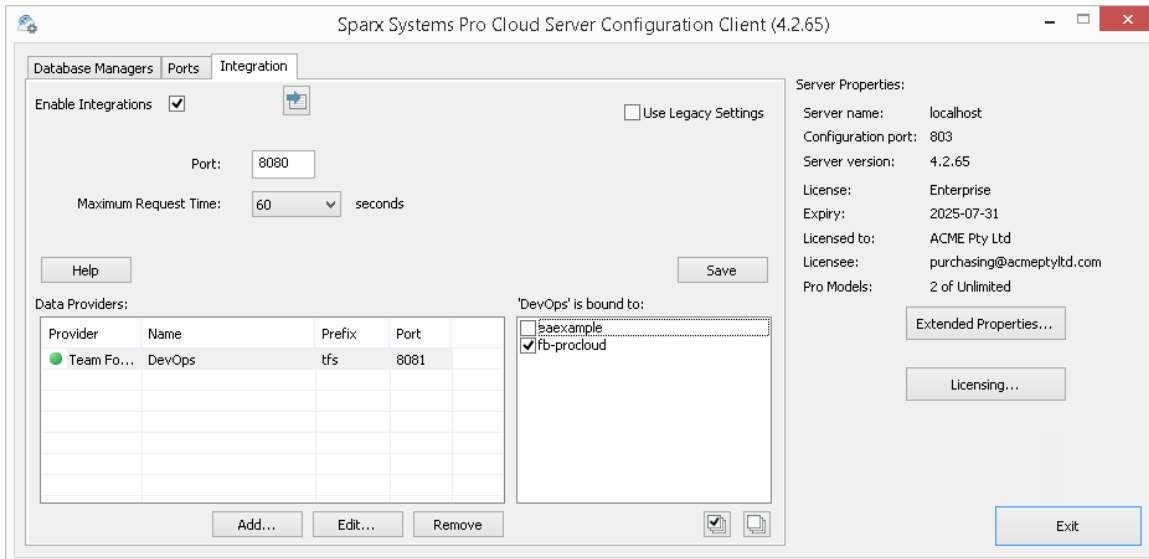
- 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的“集成”选项卡。

为 DevOps 识别 Enterprise Architect 模型

在专业云服务器配置客户端窗口的“集成”选项卡上，您现在将看到：

- 在屏幕的左下方，“数据提供者”面板将团队基础服务器/DevOps 列为数据提供者
- 在屏幕的右下角，“DevOps”绑定到：面板显示来自启用 Pro 的数据库管理员的 Enterprise Architect 模型的检查列表

您可以单击要绑定到 DevOps 的每个 Enterprise Architect 模型的复选框，或者单击  以一次选择所有它们。



单击退出按钮。

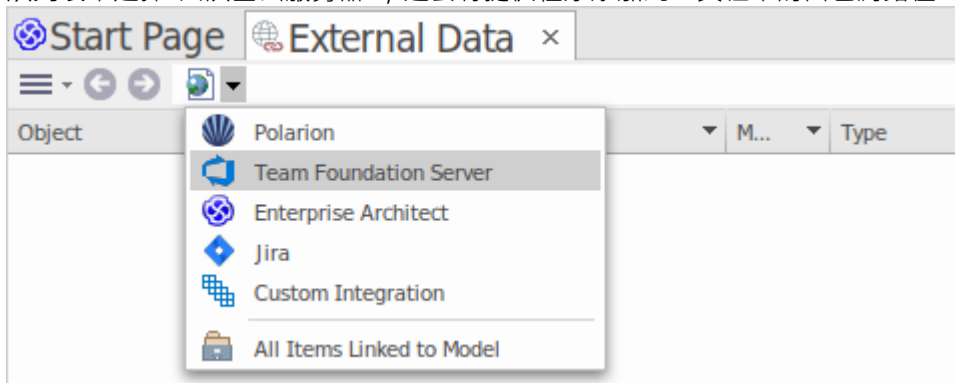
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 DevOps 的集成。

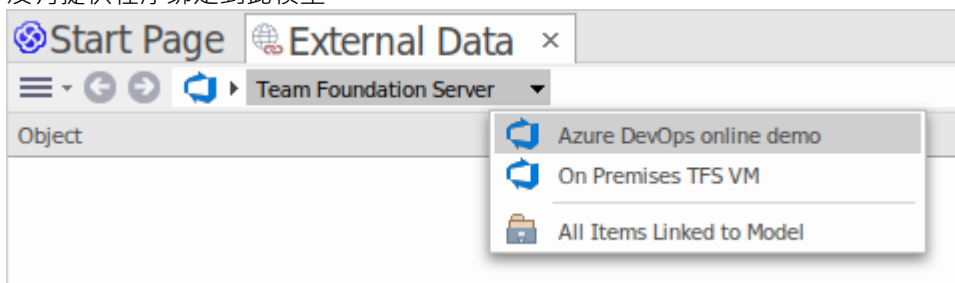
测试集成

在Enterprise Architect中打开您已确定为绑定到 DevOps 的模型之一。

1. 选择功能区选项特定>工具>系统集成。
2. 在“外部数据”窗口的工具栏中，单击“地球”图标右侧的下拉箭头。
3. 从列表中选择“团队基础服务器”；这会将提供程序添加到工具栏中的面包屑路径。



4. 单击面包屑路径中的“DevOps”以显示已配置并绑定到此模型DevOps 提供商列表。如果未列出任何内容，则没有提供程序绑定到此模型。



5. 单击 DevOps 中定义的项目之一。
6. 选择“共享查询”或“我的查询”

7. 选择一个定义的查询以在下面的列表中显示该查询返回的所有项目。

(如果您看不到团队基础服务器、DevOps 项目或任何定义的查询，请检查本主题中的过程，或[Integration Plug-ins](#)部分中更广泛和更详细的主题，看看是否有您遗漏的步骤或不满足的先决条件。您也可以查看[Troubleshooting](#)帮助主题。)

数据映射

为了让 DevOps 和 Enterprise Architect 顺利交换数据项，您必须审阅并 - 在必要时 - 编辑：

- 哪种类型的 DevOps 项对应于哪种类型的 Enterprise Architect 元素，以及
- 属性项的哪些属性对应于 Enterprise Architect 元素的哪些属性。

这些任务在[Configuration](#)帮助主题中有完整的解释。

审阅 DevOps 数据

当您测试 Enterprise Architect 和 DevOps 之间的通信时，在 External Data 窗口中，您打开了面包屑路径以显示 DevOps 项目的列表，然后选择“我的查询”或“共享查询”并从列表中选择查询。然后工作项会列在窗口的正文中。

Object	M...	Type	Stereotype	URL
Capture guest details to Cosmos DB	IT	Task		https://dev.azure.com/sparx0704/3df7f
Implement to search for loyalty membership	IT	Task		https://dev.azure.com/sparx0704/3df7f
Search for an existing reservation	IT	Task		https://dev.azure.com/sparx0704/3df7f
Implement off-season discount	IT	Task		https://dev.azure.com/sparx0704/3df7f
Generate complimentary parking coupons	IT	Task		https://dev.azure.com/sparx0704/3df7f
Generate complimentary parking coupons		Task		

如果此列表太长，您可以点击窗口工具栏中的“汉堡包”图标，选择以下选项：

- “链接项”仅显示那些链接到 Enterprise Architect 元素的 DevOps 项目
- “项 to Current Context”仅显示链接到当前选定的 Enterprise Architect 元素的 DevOps 项
- “未链接的项目”仅显示尚未链接到 Enterprise Architect 元素的那些 DevOps 项目

如有必要，选择适当的选项，然后探索所选 DevOps 项目的内容。每个属性窗口、笔记窗口和检验员窗口都有单独的选项卡或窗口版本以显示所选项目的任何属性、笔记和特征（分别）。这些在[External Item Details](#)帮助主题。

使用 DevOps 项目和 Enterprise Architect 元素

在 Enterprise Architect 和 DevOps 之间建立了有效的沟通后，您可以：

- 在 Enterprise Architect 中创建一个链接到 DevOps 项目的新元素
- 在 Enterprise Architect 中创建一个链接到元素的新 DevOps 项
- 将 Enterprise Architect 中的现有元素链接到 DevOps 项
- 使用对 Enterprise Architect 元素的更改来更新 DevOps 项
- 通过对 DevOps 项的更改更新 Enterprise Architect 元素
- 使用对 DevOps 项目的任何更改更新所有链接的 Enterprise Architect 元素

这些操作都在[Linking Items](#)帮助主题中进行了解释。

演练：Polarion集成

先决条件

为了在 Polarion 和Enterprise Architect之间建立集成，您必须首先：

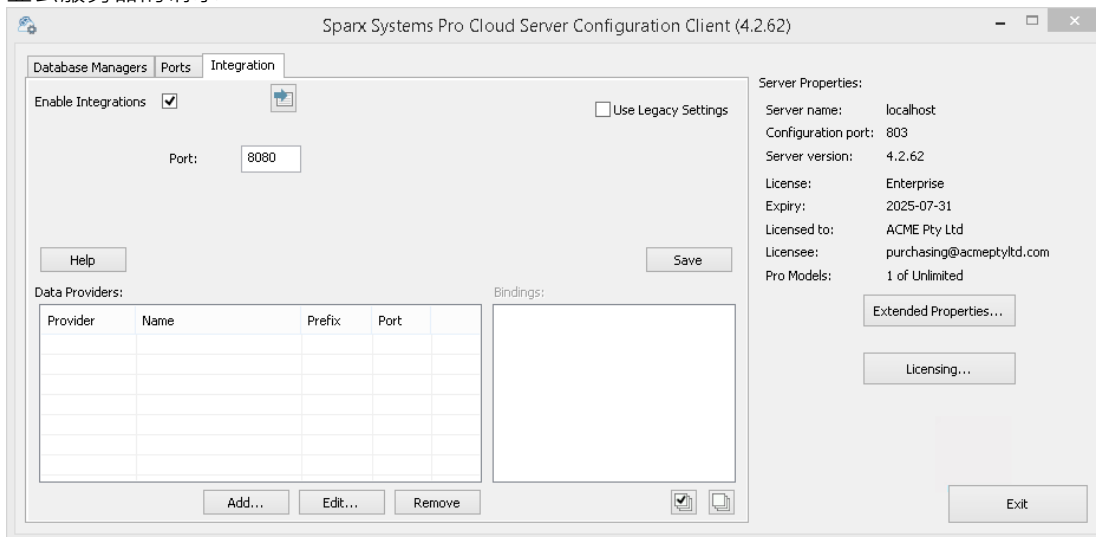
- 使用插件集成组件的专业云服务器（版本或更高版本）的许可安装
- 在托管您的Enterprise Architect模型的每个数据库管理员系统的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项
- 在 PCS 服务器和 Polarion 服务器之间具有物理网络访问权限
- 拥有访问外部数据的用户凭据
- 正在使用Enterprise Architect企业、统一或终极版 v15.2 或更高版本

我们假设集成服务器和运行集成插件与专业云服务器在同一台机器上运行

向专业云服务器和Enterprise Architect识别 Polarion

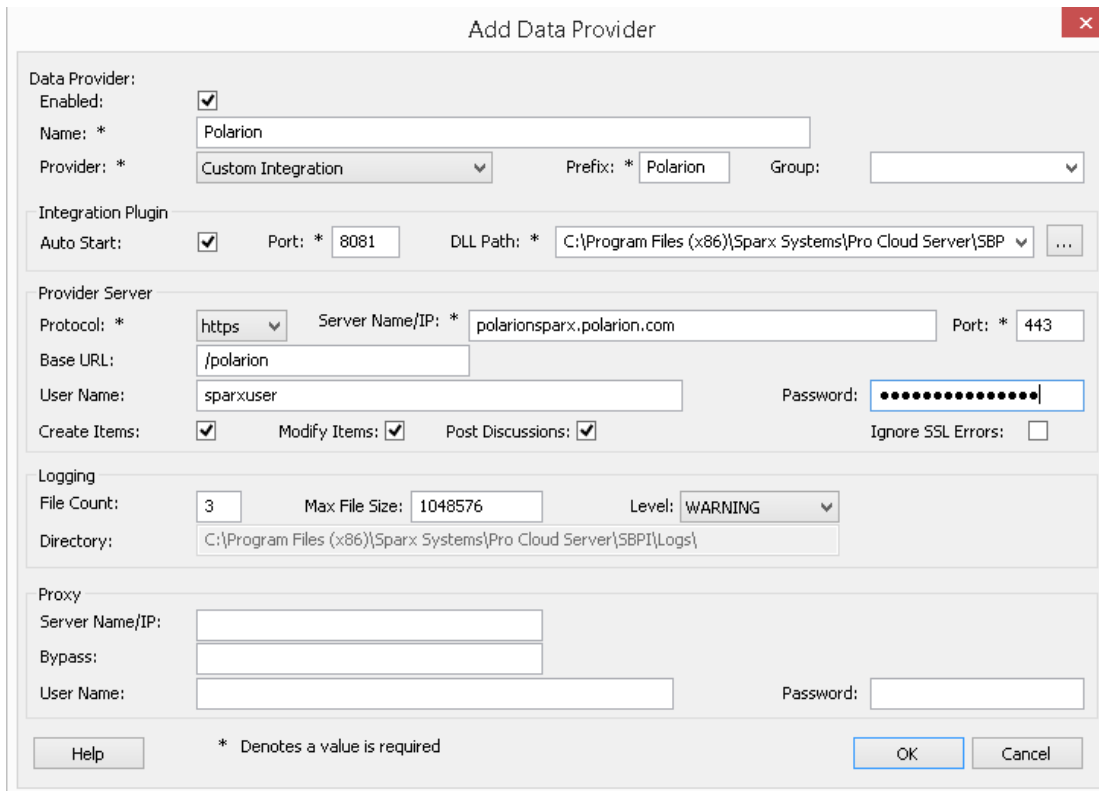
完成以下步骤：

1. 双击“Pro云Client”桌面图标以打开Pro云Client窗口。
2. 选择“集成”选项卡并选择“启用集成”复选框。在“端口”字段中输入端口的数量，该集成服务器将侦听来自专业云服务器的请求。



3. 单击添加按钮。将显示“添加数据提供者”对话框。

4. 选中“启用”复选框。
5. 在“名称”字段中，输入适当的连接名称，例如“Polarion”。
6. 在“提供者”字段中，单击下拉箭头并选择“自定义集成”。
7. 在“前缀”字段中，输入适当的前缀，例如“Polarion”。
8. 在“集成插件”面板的“插件端口”字段中，键入集成服务器用于与端口通信的插件。
9. 单击“DLL 路径”字段旁边的省略号按钮。浏览并选择“PolarionSbpi.dll”文件。它位于专业云服务器安装文件夹中。E : C:\Program Files (x86) Sparx Systems \专业云服务器\SBPI\PolarionSbpi.dll
10. 步骤 10 至 13 中的字段构成您的 Polarion 网址的组成部分；即：<protocol>://<server>:<port>/baseURL
在“协议”字段中，单击下拉箭头并选择 Polarion 插件将用于与 Polarion 通信的协议。
11. 在“服务器名称/IP”字段中，键入 Polarion 插件将用于与 Polarion 通信的服务器名称或 IP 地址。
12. 在“插件端口”字段中输入端口将用于与 Polarion 通信的端口号。
13. 在“基地 URL”字段输入 Polarion URL 的最后部分（服务器名称和端口之后的任何内容）
14. 如果您将“用户”和“密码”字段留空，则会提示每个 Enterprise Architect 用户输入他们的个人 Polarion 凭据，这可以提供更好的可用性。
15. 如果您更喜欢在“用户名称”和“密码”字段中设置值，这些值将被组合使用以访问 Polarion 中的数据。Polarion 安装的所有当前 Enterprise Architect 用户将使用同一组凭据来读取外部数据。
16. 根据需要选择“创建项”、“修改项”和/或“发布讨论”复选框，以允许 Polarion 插件的用户使用 Enterprise Architect 在 Polarion 中创建和/或更新项目和/或创建讨论帖子。
17. 如果您希望 Polarion 插件忽略由于与 Polarion 通信而发生的与 SSL 相关的错误（例如，如果您使用自签名 SSL 证书），请选中“忽略 SSL 错误”复选框。



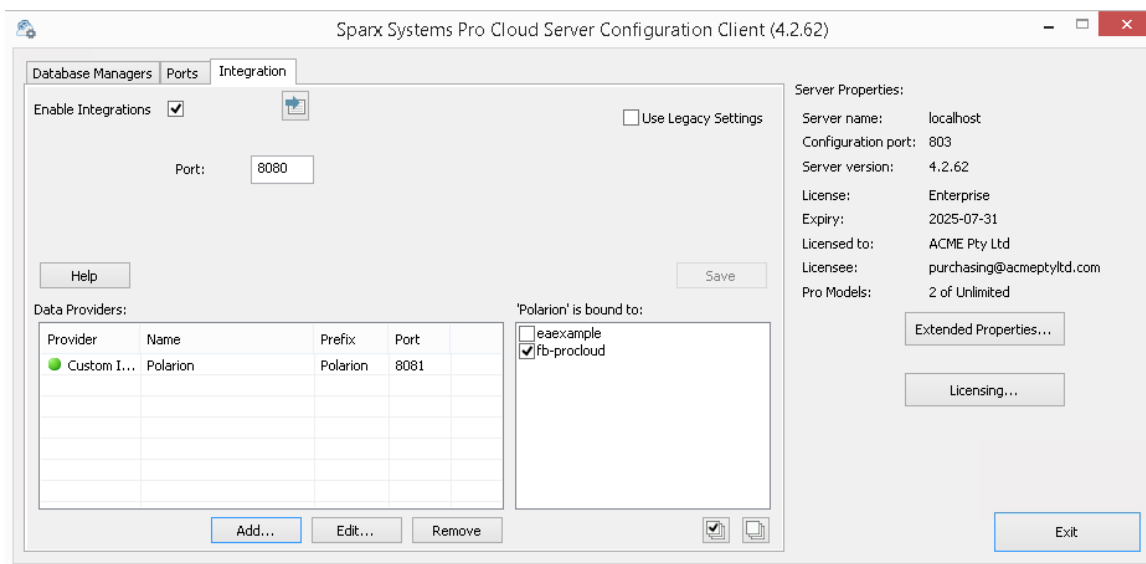
18. 您可以将其其他字段留空或设置为其默认值。点击确定按钮完成配置。这会让您返回到专业云服务器配置客户端窗口的“集成”选项卡。

为 Polarion 识别Enterprise Architect模型

在专业云服务器配置客户端窗口的“集成”选项卡上，您现在将看到：

- 在屏幕的左下方，“Data Providers”面板将 Customfor Polarion 列为数据集成提供者
- 在屏幕的右下方，“Polarion’ 绑定到：’面板显示来自启用 Pro 的数据库管理员的Enterprise Architect模型的检查列表

您可以单击要绑定到 Polarion 的每个Enterprise Architect模型的复选框，或单击  以一次选择所有这些模型。



单击退出按钮。

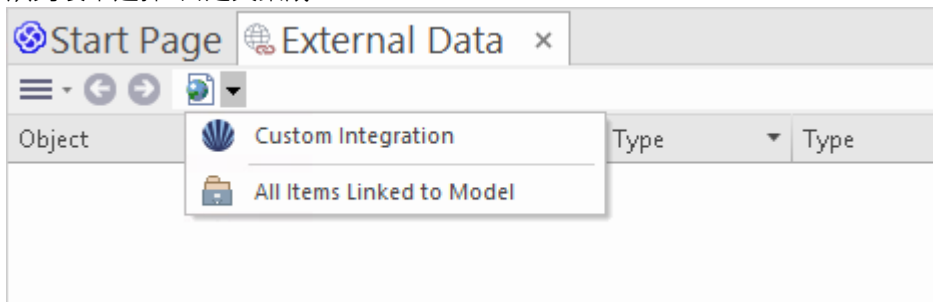
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 Polarion 的集成。

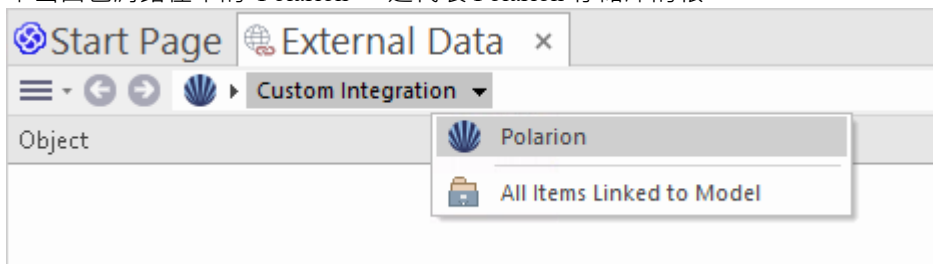
测试集成

在Enterprise Architect中打开您已识别为绑定到 Polarion 的模型之一。

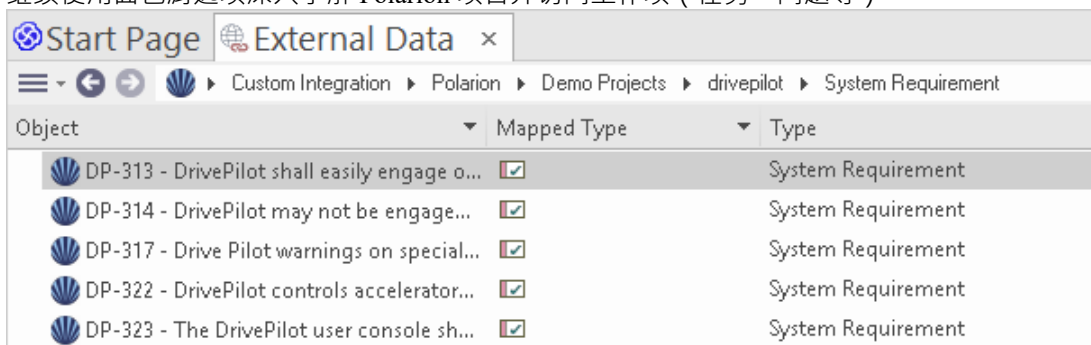
1. 选择功能区选项特定>工具>系统集成。
2. 在“外部数据”窗口的工具栏中，单击“地球”图标右侧的下拉箭头。
3. 从列表中选择“自定义集成”。



4. 单击面包屑路径中的“Polarion”，这代表 Polarion 存储库的根。



5. 继续使用面包屑选项深入了解 Polarion 项目并访问工作项（任务、问题等）。



数据映射

为了使 Polarion 和Enterprise Architect顺利交换数据项，您必须审阅并 - 在必要时 - 编辑：

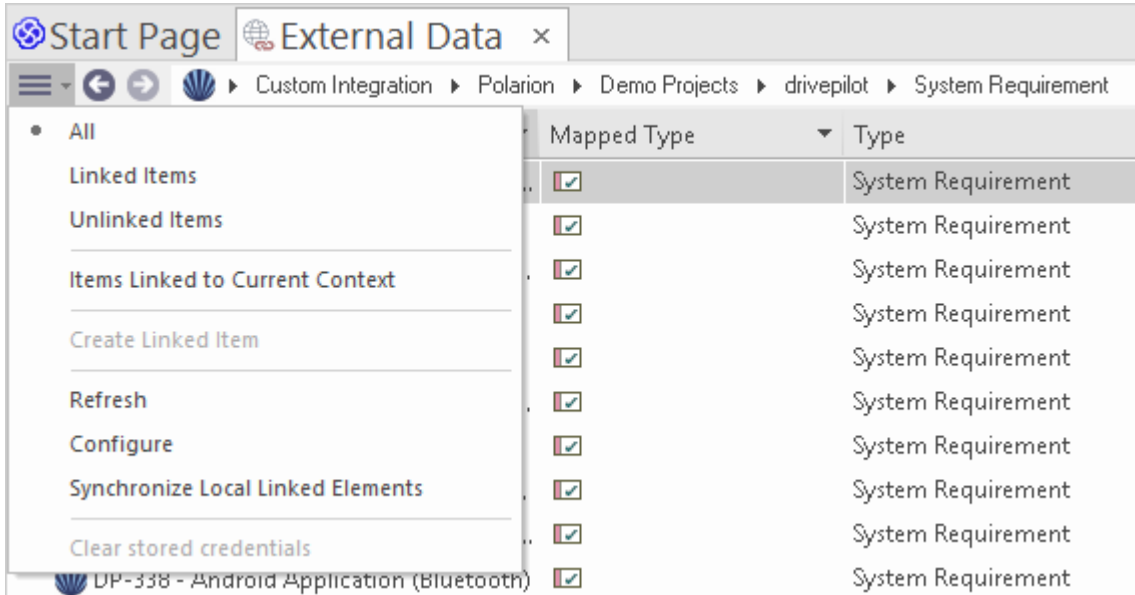
- 什么类型的 Polarion 项目对应于什么类型的Enterprise Architect元素，以及
- 属性项的哪些属性对应于Enterprise Architect元素的哪些属性。

这些任务在[Configuration](#)帮助主题中有完整的解释。

审阅Polarion Data

当您测试Enterprise Architect和Polarion 之间的通信时，在External Data 窗口中，您使用面包屑路径查看工作项列表（例如任务）。

如果此列表太长，您可以点击窗口工具栏中的“汉堡包”图标。

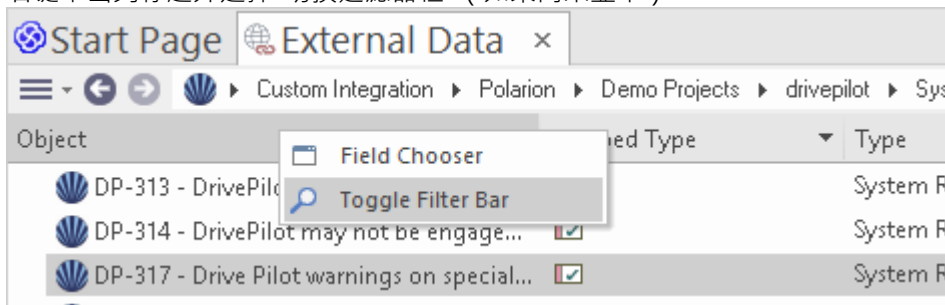


此菜单提供以下选项：

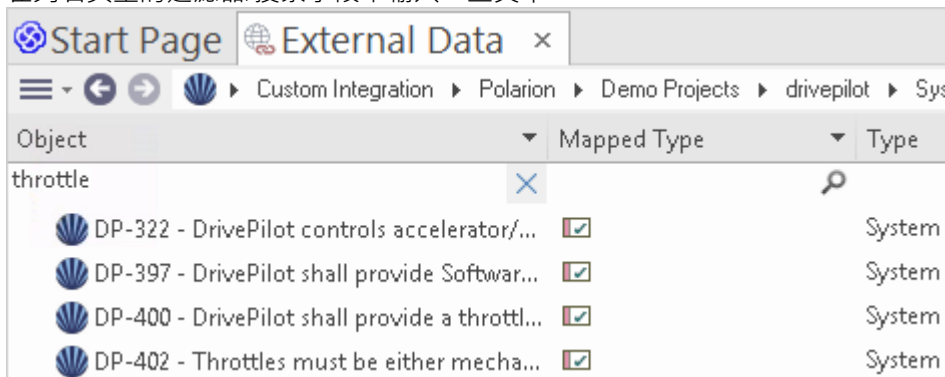
- “链接项”仅显示那些链接到Enterprise Architect元素的 Polarion 项目
- “未链接的项目”仅显示尚未链接到Enterprise Architect元素的 Polarion 项目，或
- “链接到当前上下文的元素项”仅显示链接到当前选定的Enterprise Architect的 Polarion 项目

也可以将基于文本的过滤器应用于外部数据表内容。

1. 右键单击列标题并选择“切换过滤器栏”（如果尚未显示）。



2. 在列名类型的过滤器/搜索字段中输入一些文本。



从列表中选择项目时，每个属性窗口、笔记窗口、检验员窗口和讨论和审阅窗口的“讨论”选项卡都有单独的选

项卡或窗口版本，以显示任何属性、注记、特征和讨论（分别）选定的项目。这些在 [External Item Details](#) 帮助主题。

使用 Polarion 项目和Enterprise Architect元素

在Enterprise Architect和 Polarion 之间建立了有效的沟通后，您可以：

- 在Enterprise Architect中创建一个链接到 Polarion 项目的新元素
- 在Enterprise Architect中创建一个链接到元素的新 Polarion 项目
- 将Enterprise Architect中的现有元素链接到 Polarion 项目
- 使用Enterprise Architect元素的更改更新 Polarion 项目
- 通过更改 Polarion 项目更新Enterprise Architect元素
- 使用 Polarion 项目的任何更改更新所有链接的Enterprise Architect元素

这些操作都在[Linking Items](#)帮助主题中进行了解释。

注记，当从Enterprise Architect创建 Polarion 项目时，它被放置在 Polarion Work项组中（作为要完成的新工作）。

演练：Enterprise Architect集成

本演练帮助您设置和使用专业云服务器的集成功能，通过云安装选项将一个Enterprise Architect模型与另一个模型集成。

使用与Enterprise Architect集成，您可以：

- 将Enterprise Architect元素与外部Enterprise Architect模型中的对应元素链接起来
- 根据外部元素创建本地元素，反之亦然
- 在需要时同步本地和外部元素之间的更改
- 执行可追溯性分析
- 向外部模型中的元素添加讨论

先决条件

为了设置集成Enterprise Architect，您必须首先：

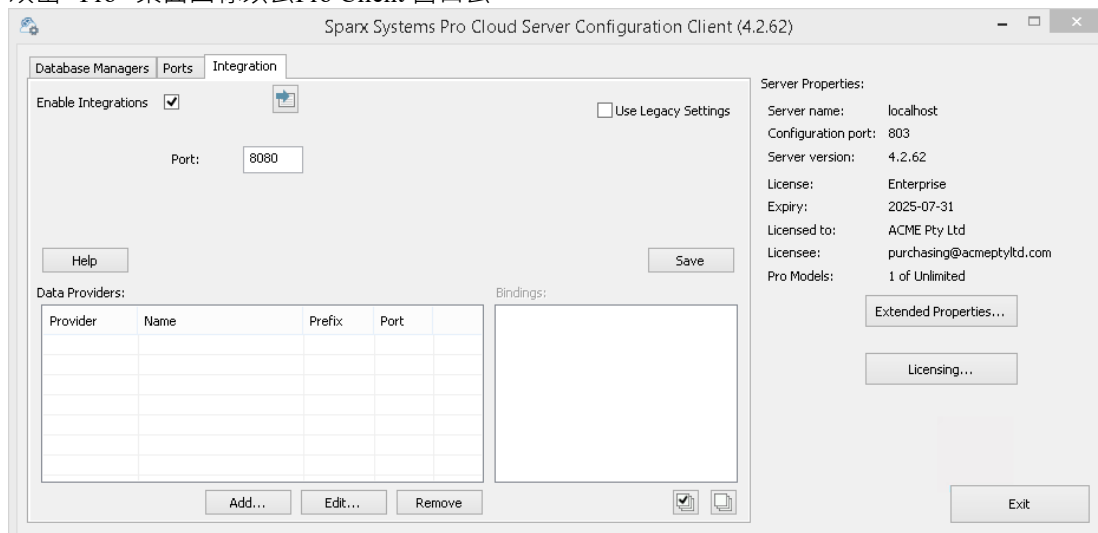
- 正如刚刚讨论的那样，使用带有集成插件组件的专业云服务器的许可安装
- 源和目标模型都应该可以通过专业云服务器访问
- 在每个模型/数据库管理器的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项。
- 在PCS服务器和托管外部EA模型PCS服务器之间具有物理网络访问权限（这仅适用于使用不同/外部PCS托管目标模型的情况）
- 拥有访问目标模型的用户凭据
- 使用Enterprise Architect企业、统一或终极版 v14 或更高版本

对于本演练，我们假设您使用专业云服务器配置客户端中的“集成”选项卡定义集成服务器选项，而不是通过手动操作注册表或配置文件或使用WebConfig界面。

配置Enterprise Architect集成以访问目标模型

完成以下步骤：

1. 双击“Pro”桌面图标以云Pro Client窗口云



2. 选择“集成”选项卡并选中“启用集成”复选框。在“端口”字段中键入端口号集成服务器将侦听来自专业云服务器的请求。这可以是您选择的任何有效/可用端口号。在大多数情况下，可以使用默认值 8080。
注记：您可以使用窗口资源监视器 (resmon.exe) 实用程序来帮助识别其他应用程序和服务已在使用的 监听

端口”。Pro云配置客户端的“端口”选项卡上有打开资源监视器的按钮。

- 单击添加按钮。显示“添加数据提供者”对话框。

- 选择“启用”复选框。
- 在“名称”字段中，键入适当的连接名称。此集成将链接到特定目标模型，因此最好使用有助于识别该模型的名称。对于此示例，我们将使用通用名称“EA目标模型”。
- 在“提供者”字段中，单击下拉箭头并选择“Enterprise Architect”；这也会自动将“EA”添加到“前缀”字段。
- 在“集成插件”面板的“端口”字段中，键入集成服务器将用于与Enterprise Architect插件通信的端口号。这可以是您选择的任何有效/可用端口号。通常可以使用默认值（此对话框分配一个高于8080的数字，该数字尚未分配给另一个集成提供商）。
- 注记，以下四个字段对应于协议、服务器、端口和模型名称，您在通过Enterprise Architect中的云连接选项访问目标模型时将输入这些字段。
- 在“协议”字段中，单击下拉箭头并选择“http”或“https”。
- 在“服务器名称/IP”字段中，键入运行配置为访问目标模型的专业云服务器的计算机的服务器名称或IP地址。通常源和目标模型都在当前机器的专业云服务器（您正在配置集成的地方）上定义。在这种情况下，您可以在此字段中输入“localhost”。
- 在“端口”字段中输入用于访问目标模型端口号。
- 在“基地URL”字段输入目标模型名称，就像您通过Enterprise Architect连接到它时所做的那样。
- 如果目标模型启用了用户安全，请在“用户”和“密码”字段中输入一些有效凭据。
- 根据需要选择“创建项”、“修改项”和/或“发布讨论”复选框。这些选项允许用户创建或修改元素，或在目标模型中发表讨论。
- 如有必要（例如，您正在使用带有自签名证书的HTTPS），请选中“忽略SSL错误”选项。

16. 您可以将其他字段留空或设置为默认值。单击确定按钮以完成配置。这将返回到专业云服务器配置客户端窗口的“集成”选项卡。
17. 如果您想与多个不同的目标模型集成，您可以通过重复上述步骤（从第 3 步开始）来实现，为每个目标模型定义一个提供者。

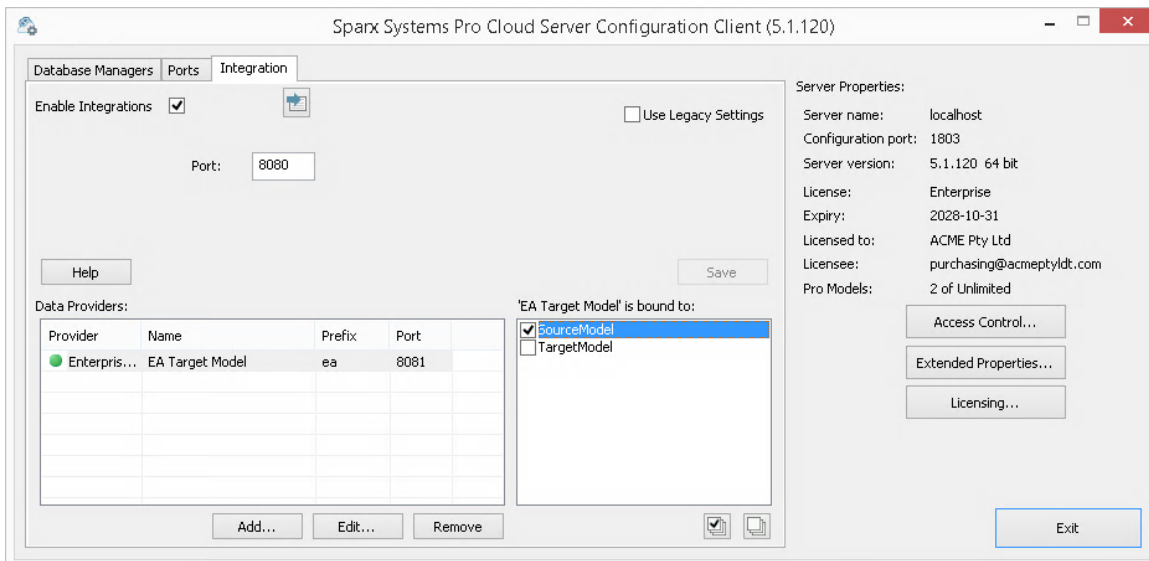
配置哪些源模型可以访问Enterprise Architect集成

在专业云服务器配置客户端窗口的“集成”选项卡上，您将现在看到：

- 在屏幕的左下角，“数据提供者”面板将Enterprise Architect列为数据提供者
- 在屏幕的右下角，“Enterprise Architect绑定到：”面板显示来自启用 Pro 的数据库管理员的Enterprise Architect模型检查列表

您可以单击要绑定到Enterprise Architect集成的每个Enterprise Architect模型复选框（授予对目标模型访问权限

），或者单击  一次选择所有模型。



单击退出按钮。

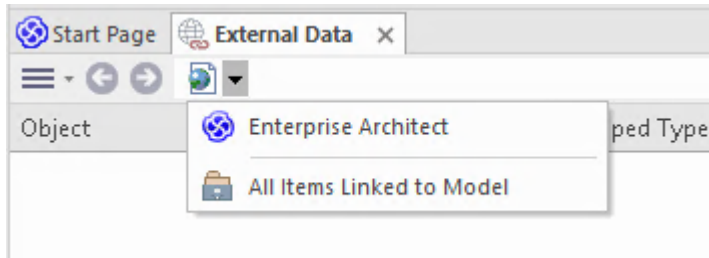
重新启动 PCS 服务以应用所做的更改。

您现在可以进入您的Enterprise Architect 源”模型之一并检查 外部 目标”模型集成。

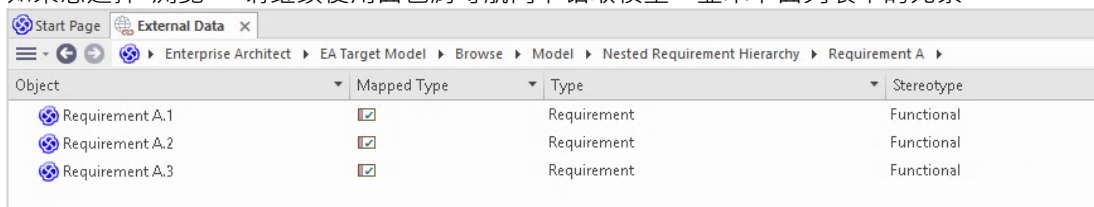
测试整合

在Enterprise Architect中打开您绑定到外部 目标”模型 源”模型之一。

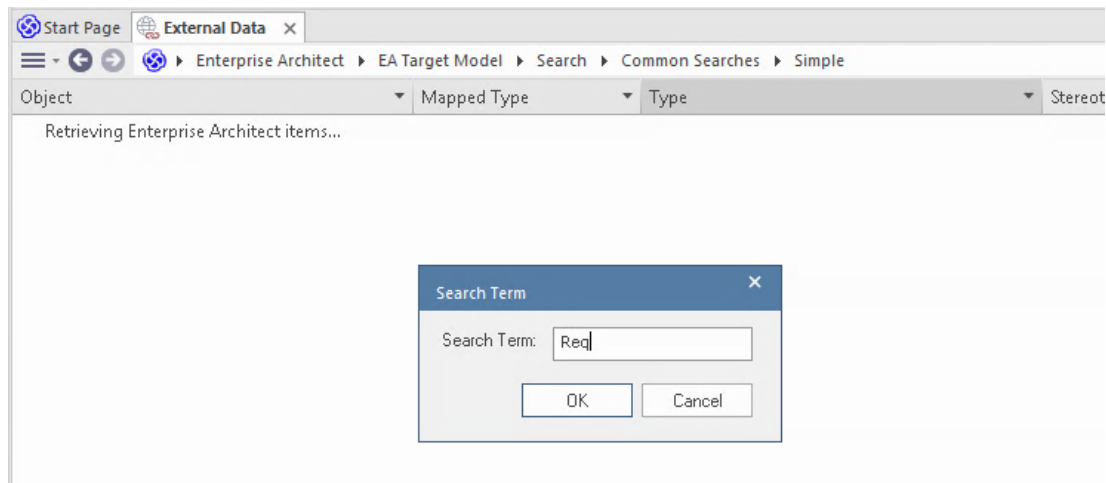
1. 选择功能区选项'特定>工具>系统集成'。
2. 在 外部数据”窗口的工具栏中，单击 地球”图标右侧的下拉箭头。



3. 从列表中选择 “Enterprise Architect ”；这会将提供程序添加到工具栏中的面包屑路径。
4. 单击面包屑路径中的集成名称（例如 EA目标模型”），然后单击 浏览”或 搜索”。
5. 如果您选择 浏览”，请继续使用面包屑导航向下钻取模型，显示下面列表中的元素。

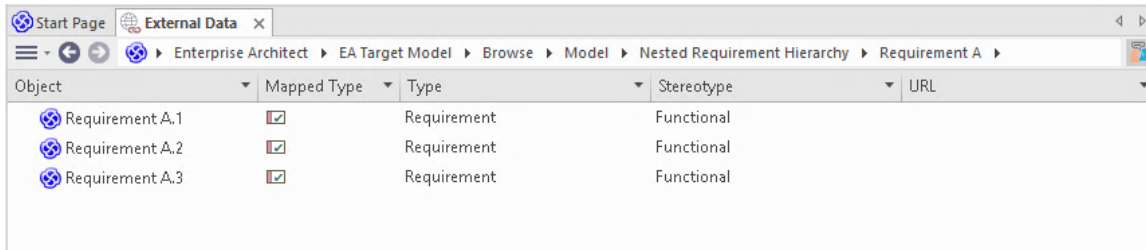


6. 如果您选择搜索，请选择一个搜索组（例如 公共搜索”）、一个搜索定义（例如 简单”），然后在提示中输入一个搜索团队。外部模型中与搜索词匹配的所有元素都将在下方列出。



审阅外部模型数据

当您测试Enterprise Architect和外部模型之间的通信时，在外部数据窗口中，您浏览或搜索元素列表。



如果这个列表太长，您可以点击窗口工具栏中的“汉堡包”图标并选择选项，例如：

- “链接项”仅显示外部模型中 当前模型中的元素链接的那些元素。
- “项链接到当前上下文”仅显示外部模型中的元素，这些元素链接到当前选择的Enterprise Architect元素或
- ‘Unlinked items’ 仅显示来自外部模型元素，这些元素尚未链接到当前模型中的元素。

如有必要，选择适当的选项，然后从外部模型中探索所选元素的内容。每个属性窗口、笔记窗口和检验员窗口都有单独的选项卡或窗口版本，以显示所选项目的任何属性、笔记和特征（分别）。这些说明在 [External Item Details](#)帮助主题。

使用Enterprise Architect元素和来自外部模型元素

在当前打开的模型和外部模型之间创建工作通信后，您可以：

- 在Enterprise Architect中创建一个新的元素链接到一个来自外部模型的元素
- 在外部模型中创建一个新的元素，它链接到当前模型中的一个元素
- 将Enterprise Architect中的现有元素链接到外部模型中的元素
- 使用对当前模型中元素的更改更新外部模型中的元素
- 使用外部模型中元素的更改更新当前模型中的元素
- 使用对外部模型中元素的任何更改更新当前模型中的所有链接元素

这些操作都在[Linking Items](#)帮助主题中进行了解释。

故障排除

集成服务器消息

在定义集成服务器设置期间可能会显示某些错误消息；大多数都是不言自明的。此表描述了最常见的错误消息。

错误信息	描述
请为 [field_names] 输入一个值。集成服务器只有在所有必填字段都有值时才能保存（启用时）	原因：选中“启用”标志后，所有剩余的集成服务器字段都是必需的。当一个或多个集成服务器字段为空并且已选中“启用”标志时，会出现此消息。 [field_names] 表示以逗号分隔的空字段名称列表的占位符。
SBPI.EXE 的可执行文件名是预期的	原因：“可执行路径”字段有一个值，但它不包括文件名“..\SBPI.exe”，这是唯一可以使用有效文件名。
[full_path] 指定的集成服务器路径不存在或无效	原因：“可执行路径”字段有值，但指定值的路径部分无效或在专业云服务器机器上不存在。
指定的端口是 [duplicate_plugin_name] 的副本	原因：“端口”字段已分配一个端口号，但它与另一个集成数据提供者使用的端口相同，该端口由名称 [duplicate_plugin_name] 标识。

集成数据提供者消息

在定义集成数据提供者的过程中可能会显示某些错误消息；大多数都是不言自明的。此表描述了最常见的错误消息。

错误信息	描述
请为 [field_names] 输入一个值。如果没有为所有必填字段分配值，则无法保存集成提供者。	原因：为了最大限度地减少定义无效数据提供者的机会，应用程序确保所有必填字段都被赋予了一个值。 [field_names] 表示以逗号分隔的空字段名称列表的占位符。
端口需要在所有 Provider 中唯一，[new_port] 的端口值已被 [other_provider] 使用	原因：已经为当前的Data提供者定义了A端口号；但是，该值与另一个提供商的端口重复。
提供程序的指定执行路径不存在或无效， [full_path]	原因：已为数据提供者的“执行路径”分配了一个值，但在本地计算机上找不到路径或文件名或两者。
指定的端口是集成服务器端口的副本	原因：为当前Data提供者定义了A端口号，但该端口值与集成服务器使用的端口重复。
无法将配置文件 [old_filename] 重命名为 [new_filename]	原因：当用户更改现有数据提供者的配置文件名时，云服务器必须将旧文件名重命名为新文件名。如果重命名任务不成功（如果文件正在被另一个进程使用，则可能发生），将显示此消息，在这种情况下，手动干预是解决此问

	题的唯一方法。
前缀在所有 Provider 中必须是唯一的， [new_prefix] 的前缀值已被 [other_provider] 使用	原因：前缀为当前数据A提供者定义了前缀，但该值与另一个提供者使用的前缀重复。
配置文件名需要在所有 Providers 中唯一， [new_filename] 的配置文件名已被 [other_provider] 使用	原因：已为当前数据提供者定义了A配置文件名，但该值与另一个提供者使用的配置文件名重复。

一般故障排除

此表提供一般建议，以帮助识别和解决专业云服务器集成组件的常见问题。

问题描述
<p>问题：对服务器或数据提供者的集成定义进行更改后，Enterprise Architect用户没有注意到任何差异。</p> <p>解决方法：更改集成配置后，专业云服务器是否重启？如果没有，请重启专业云服务器。</p>
<p>问题：Enterprise Architect（专业云服务器重启后）导航面包屑中未列出新配置A集成数据提供者。</p> <p>解决方法：确认当前打开的模型有数据提供者的捆绑入口。有关详细信息，请参阅Integration Plug-ins帮助主题的Data Providers表中的'帮助'字段。</p>
<p>问题：在Enterprise Architect的集成导航面包屑中选择项目似乎没有效果。</p> <p>解决方案：系统输出窗口可能出现错误（或警告），因此请确保在使用外部数据窗口时显示系统输出窗口。</p>
<p>问题：似乎正在发生一般错误；但是，系统输出窗口中没有显示任何有用的信息。</p> <p>解决方案：每个集成数据提供者都可以将详细条目写入log文件；为确保写入最详细的信息，请将数据提供者的log级别更改为SYSTEM（重新启动专业云服务器），然后在Enterprise Architect中重试相同的操作。然后审阅数据提供者的log文件；即专业云服务器上的C:\Program Files (x86)\Sparx Systems\专业云服务器\专业云服务器\Logs。</p>
<p>问题：从插件收到的一般错误或空白错误。</p> <p>解决方案：没有其他服务应用程序使用与选择主集成配置中定义的端口相同的选项卡。如果另一个程序正在使用相同的端口，则错误消息将来自该应用程序，而不是专业云服务器或SBPI。</p> <p>打开A端口列表和使用它们的应用程序可以在窗口的“资源监视器”中找到。</p>

