



**ENTERPRISE ARCHITECT**

用户指南系列

# 整合来自外部供应商的数据

Author: Sparx Systems

Date: 13/11/2024

Version: 6.0

创建于  **ENTERPRISE  
ARCHITECT**

# 目录

集成来自外部提供商的数据	4
安装和配置	7
集成插件	14
添加/编辑数据提供者	18
数据提供者高级认证	24
演练：ServiceNow集成	26
演练：Jira集成	33
演练：DevOps集成	39
演练：Jama集成	48
演练：Polarion集成	54
演练：Enterprise Architect集成	60
故障排除	65
访问的信息	67
导航外部数据	68
外部项目详细信息	71
配置	74
链接项	77
到WebEA的项目超链接	82
编写自定义集成插件	84
ISBPIIntegrationPlugin 接口	86
创建WebEA超链接	88
获取授权详情	89
获取默认字段映射	91
获取默认类型映射	93
获取字段	95
获取图标	96
获取项目	97
获取项目讨论	99
获取项目列表	100
GetItemNotes	101
获取项目	102
获取菜单列表	103
获取类型	104
项目链接	105
项目未链接	106
物品已更新	107
后新讨论	108
PostNewItem	109
PostOAuthCode	110
PostUpdateItem	111
PostUpdateItemNotes	112
刷新OAuthToken	113
设置授权	114
设置回调	116
设置配置	117
自定义 SBPI 服务	119
ISBPIServicePlugin 接口	121

处理请求	122
设置回调	124
设置配置	125
示例脚本	127

# 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>信息交流>系统集成
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”原型或另一种元素类型链接到Enterprise Architect。

	可以同时链接整个树结构层次 - 导入Enterprise Architect同时保持树结构层次。
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，某些提供商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>
将图表图像附加到外部项目	<p>一旦将链接元素添加到图表中，就可以上传相关图表图像并附加到外部项目。</p> <p>右键单击列表中的本地链接元素，然后选择“将图表图像附加到外部项目”。如果图表图像发生变化，则将更新所有现有图像附件，并上传任何新图像。</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"> <li>系统输出窗口将在尝试从外部提供商检索数据时显示任何错误</li> <li>专业云服务器为每个外部提供商输出log文件</li> </ul>

## 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）

# 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：

- 获得许可的专业云服务器A
- PCS服务器与托管外部数据的服务器之间的网络访问
- 访问外部数据的用户凭证
- 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。

插件	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。 注记 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"> <li>• IBM Rational DOORS Next Generation 的需求管理工具</li> <li>• Rational Rhapsody 设计管理 (DM)</li> <li>• Rational Team Concert 更改和配置管理 (CCM)</li> </ul>

	<ul style="list-style-type: none"> <li>• 理性质量经理 (QM)</li> </ul>
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。

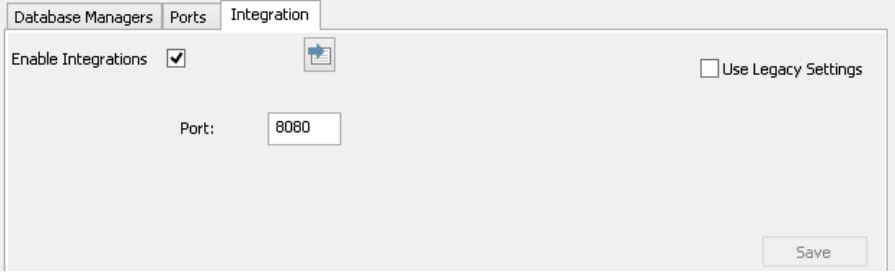
但是，如果您选择在与专业云服务器不同的机器上运行集成服务器或集成插件，则必须手动配置每个单独的插件，并配置为使用正确参数作为窗口服务启动。请参阅步骤 - 手册表。

## 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系列注册表设置，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<a href="#">WebConfig - Integration Plug-ins</a>主题。</p>
配置集成数据提供者	<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>
配置防火墙	<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
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（或 SBPI）服务器进行通信。这些配置选项被定义为一系列注册表设置。</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"> <li>• <b>Enabled</b> - true or false，表示端口服务器应该监听的端口号，哪个值应该</li> </ul>

	<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"> <li>• <b>LocalPort</b> - 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）</li> <li>• <b>UseLegacy</b> - 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</li> <li>• <b>参数</b>- 不用于专业云服务器4.0 1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”</li> <li>• <b>协议</b>- 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</li> <li>• <b>服务器</b>- 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</li> <li>• <b>端口</b>- 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</li> <li>• <b>IgnoreSSLErrors</b> - 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</li> <li>• <b>AttemptAutoDiscovery</b> - 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</li> <li>• <b>ClientProtocol</b> - 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</li> <li>• <b>ClientServer</b> - 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</li> <li>• <b>ClientPort</b> - 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</li> </ul> <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) 通常需要被授予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>IGNORESSLERRORS=真</p> <ul style="list-style-type: none"> <li>• <b>PROTOCOL</b> - 通信协议，http 或https</li> <li>• <b>SERVER</b> - 外部数据源的服务器名称（或 IP 号），例如 example.com</li> <li>• <b>PORT</b> - 外部数据源配置监听的端口，例如 443</li> <li>• <b>BASEURL</b> - 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</li> <li>• <b>用户名</b>- 可选 - 见注 1</li> <li>• <b>密码</b>- 可选 - 见注 1</li> <li>• <b>CREATEITEMS</b> - 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</li> <li>• <b>MODIFYITEMS</b> - 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</li> <li>• <b>POSTDISCUSSIONS</b> - 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</li> <li>• <b>PROXY</b> - 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</li> <li>• <b>PROXYBYPASS</b> - 可选 - 不应该通过代理发送的 IP 号码的分号分隔列表；例如，&lt;本地&gt;;10.*</li> <li>• <b>IGNORESSLERRORS</b> - 可选 - 忽略任何 SSL 证书错误的布尔值</li> </ul> <p><b>注1</b></p> <ul style="list-style-type: none"> <li>• 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</li> <li>• 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</li> <li>• 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</li> <li>• 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</li> </ul> <p><b>笔记2</b></p> <ul style="list-style-type: none"> <li>• 并非所有外部产品都支持创建和修改其对象；例如，Dropbox 没有</li> </ul>
<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"> <li>• LocalModel -模型的 DB别名，例如 'eaexample'</li> <li>• Plugin - 外部数据源的 UUID；例如，{853489C1-4C22-4bad-9A8E-3098D07A3FC1}</li> </ul>

### 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。

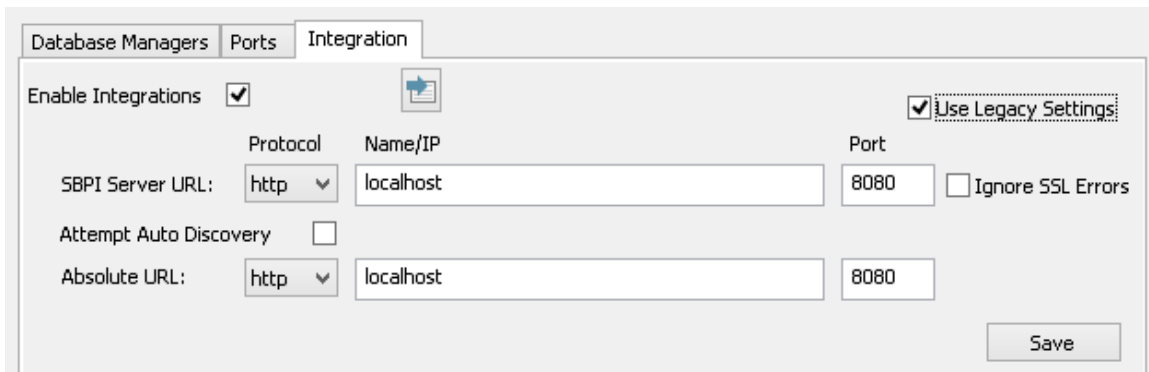
还专门针对[Walkthrough: Jira Integration](#)和[Walkthrough: Polarion Integration](#)集成提供了详细的演练。

## 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。



**注意**：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><b>注记：</b>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><b>注记：</b>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 \$BPI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><b>注记：</b>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><b>注记：</b>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“备用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；它仅在在有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。 <b>注记：</b> 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。 <b>注记：</b> 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: \*
  - Provider: \* (dropdown menu)
  - Prefix: \* (text field)
  - Group: (text field)
- Integration Plugin:**
  - Auto Start:
  - Port: \* 8082
- Provider Server:**
  - Protocol: \* (dropdown menu)
  - Server Name/IP: \* (text field)
  - Port: \* (text field)
  - Base URL: (text field)
  - Maximum Request Time: 60 seconds
  - Authentication: Credentials (dropdown menu)
  - User Name: (text field)
  - Password: (text field)
  - 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: (text field)
  - Bypass: (text field)
  - User Name: (text field)
  - Password: (text field)

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"> <li>• Jazz：服务器默认使用端口或https</li> <li>• Jira：自托管服务器默认端口是http端口8080</li> <li>• Confluence：自托管服务器默认端口是http端口8090</li> </ul>
基本网址	<p>一些外部数据提供商允许在单个位置存在多个存储库的概念；对于此类数据提供者，该字段充当过滤器。</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://myEAserver:804），而“基本 URL”是存储库（myModel）的 DB别名。因此： http://myEAserver:804/myModel</p> <p>特定提供者的详细信息：</p> <ul style="list-style-type: none"> <li>• Confluence：在线云托管帐户要求将其设置为“wiki”；自托管服务器将依赖于单个服务器的设置</li> <li>• Jira：对于在线云托管帐户，此项应为空白；自托管服务器将取决于单个服务器的设置</li> <li>• Dropbox：此项应留空</li> <li>• Enterprise Architect：这应该是目标Enterprise Architect模型的“模型别名”</li> </ul>
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<a href="#">Data Provider Advanced Authentication</a>屏幕上填写高</p>

	<p>级身份验证选项。</p> <p><b>注记1</b>：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： Dropbox、Jira、Salesforce、ServiceNow、SharePoint 和 Wrike</p> <p><b>注记2</b>：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	<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>注记：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”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><b>注记</b>：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 <a href="#">Data Provider Advanced Authentication</a> 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用户是否能够针对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	默认值：3 此值表示应为当前集成插件保留的log文件的“滚动”数量。
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

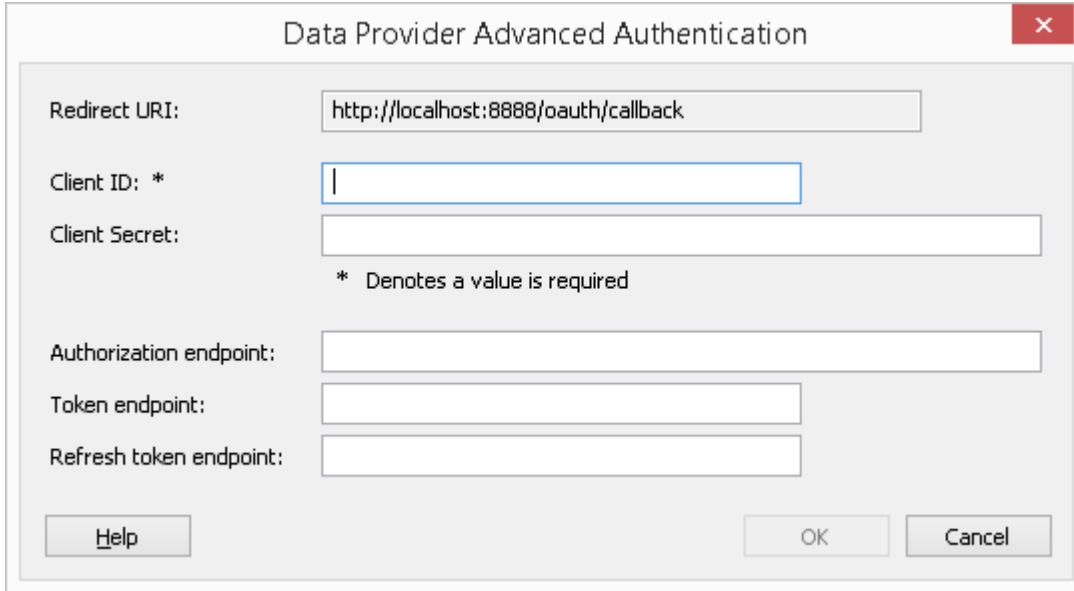
这组设置定义了当集成插件被代理服务器分隔时，它们应该如何与外部数据提供者进行通信。

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。

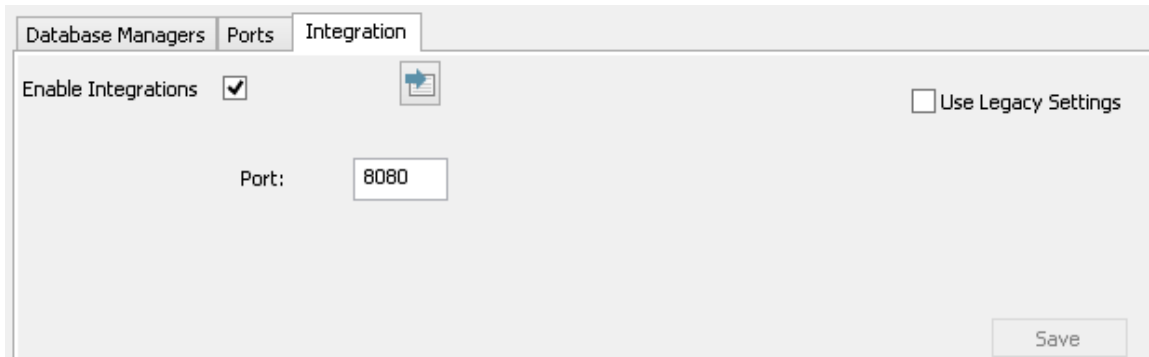
注记：OAuth 仅被部分集成提供商支持。



The dialog box is titled "Data Provider Advanced Authentication" and 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

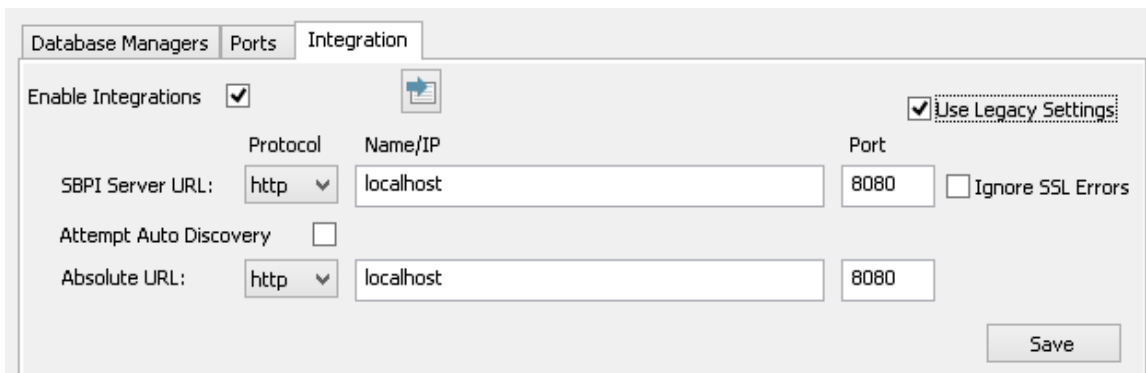


The dialog box shows the "Integration" tab with the following settings:

- 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 dialog box shows the "Integration" tab with advanced settings:

- Enable Integrations:
- Use Legacy Settings:
- Ignore SSL Errors:
- Attempt Auto Discovery:
- SBPI Server URL: Protocol  Name/IP  Port
- 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。 <b>注记：</b> 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： <a href="https://login.microsoft.com/oauth/authorize">https://login.microsoft.com/oauth/authorize</a>
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。 <b>注记：</b> 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。

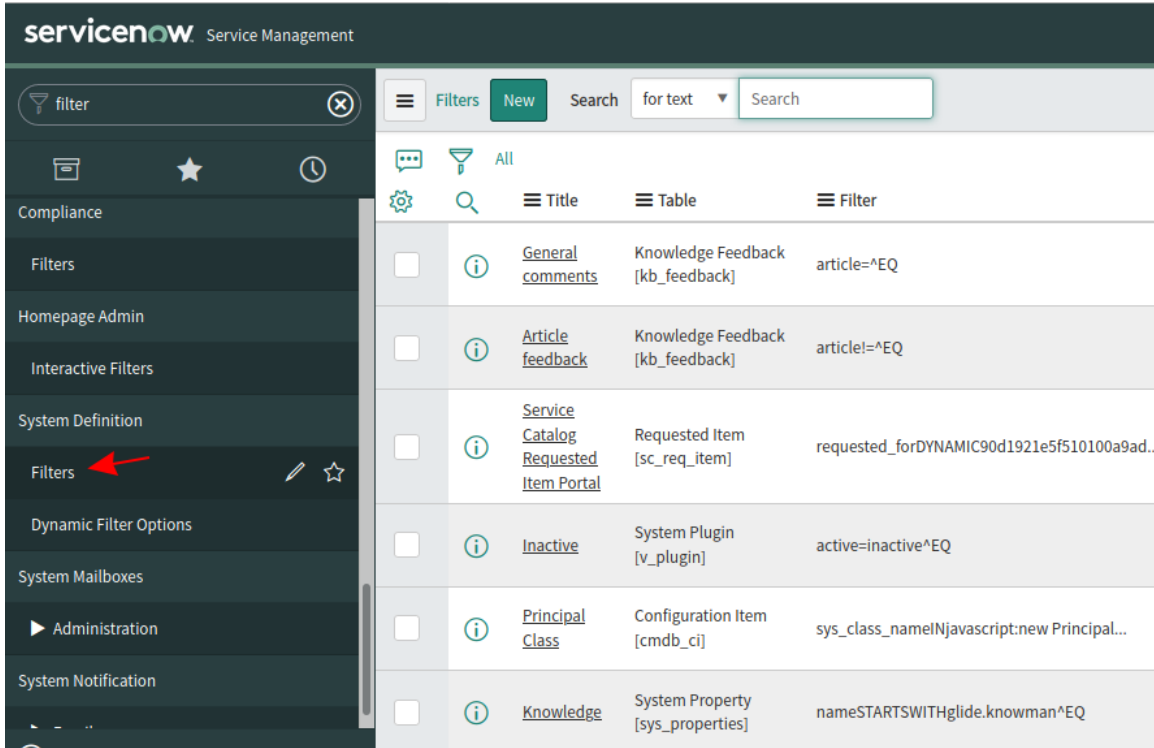
A需要第二条规则来启用对表中所有字段的读取访问权限（例如，表.\*）。

访问控件规则应具有所需的角色分配，然后可以将该角色分配给组或个人用户。

注意：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。

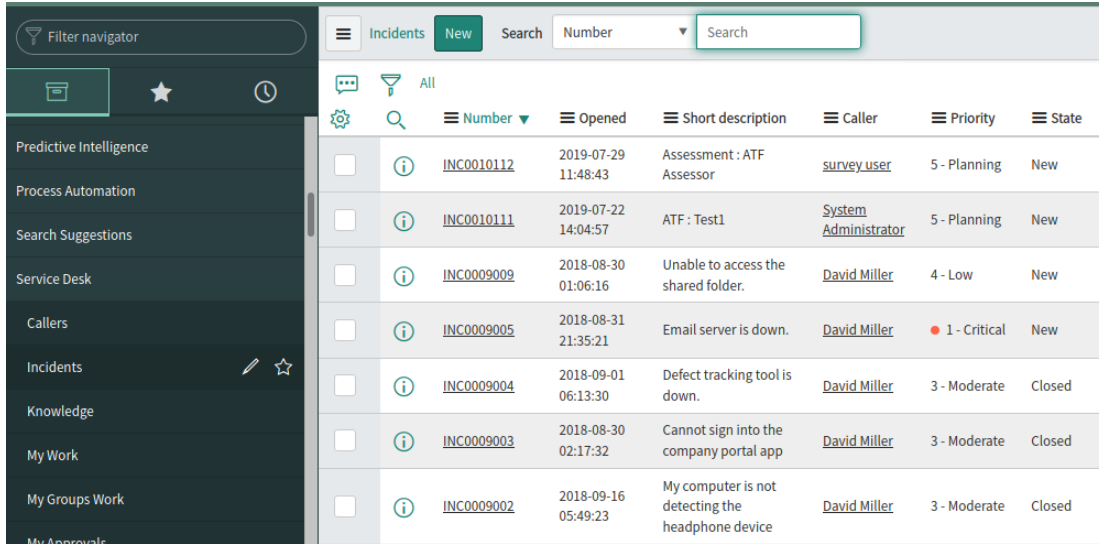
### 定义 ServiceNow表过滤器

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”。

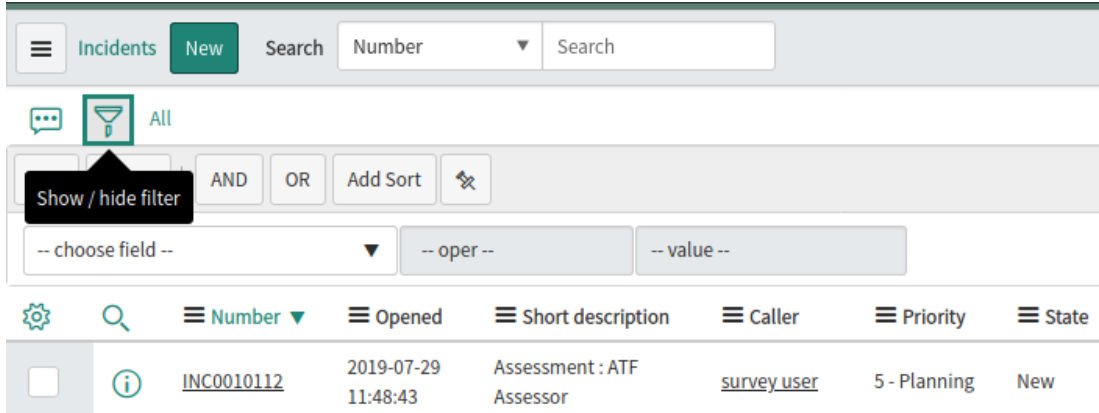


您可以按照以下步骤在 ServiceNow 中创建新过滤器：

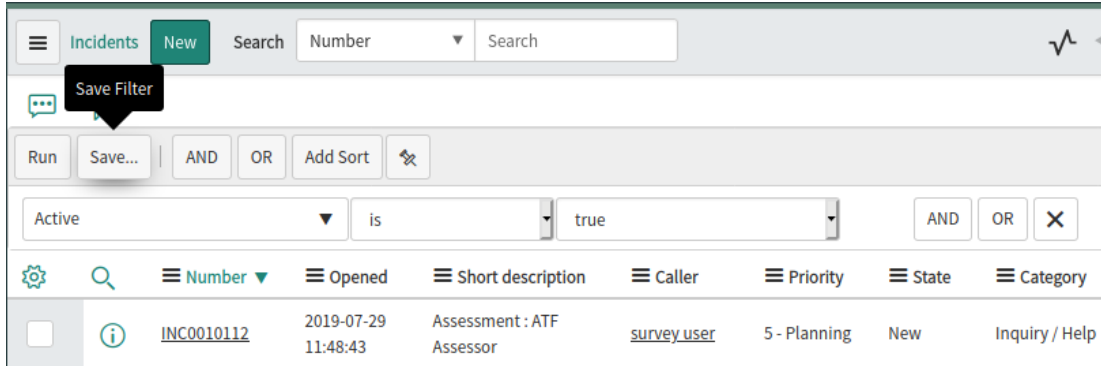
1. 在 ServiceNow 中找到相关数据（例如事件）



2. 显示过滤器栏。



- 3. 调整过滤器选项。
- 4. 保存过滤器。

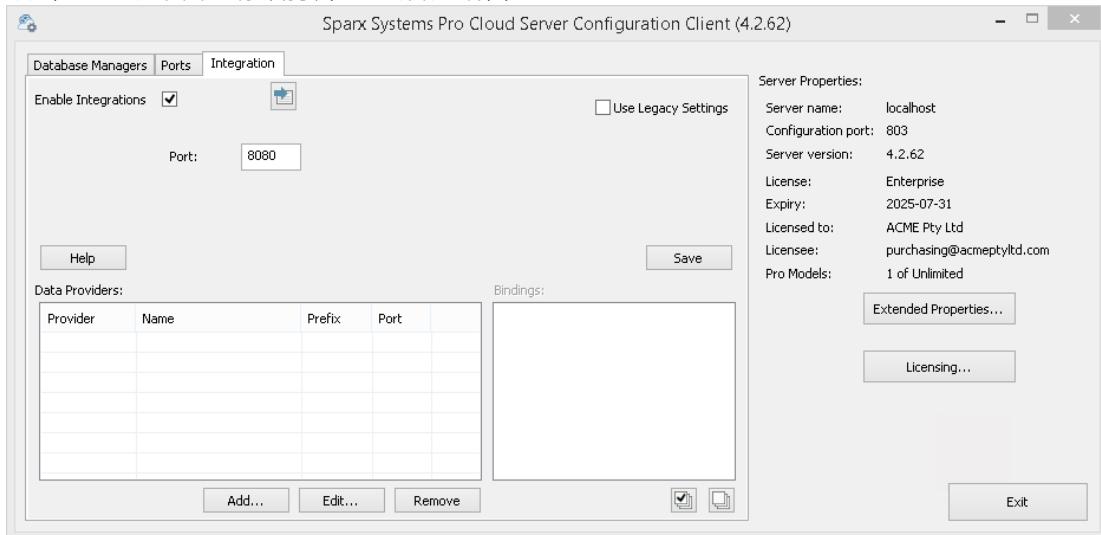


- 5. 专业云服务器的 ServiceNow 集成现在可以访问此过滤器（及其返回的项目）。

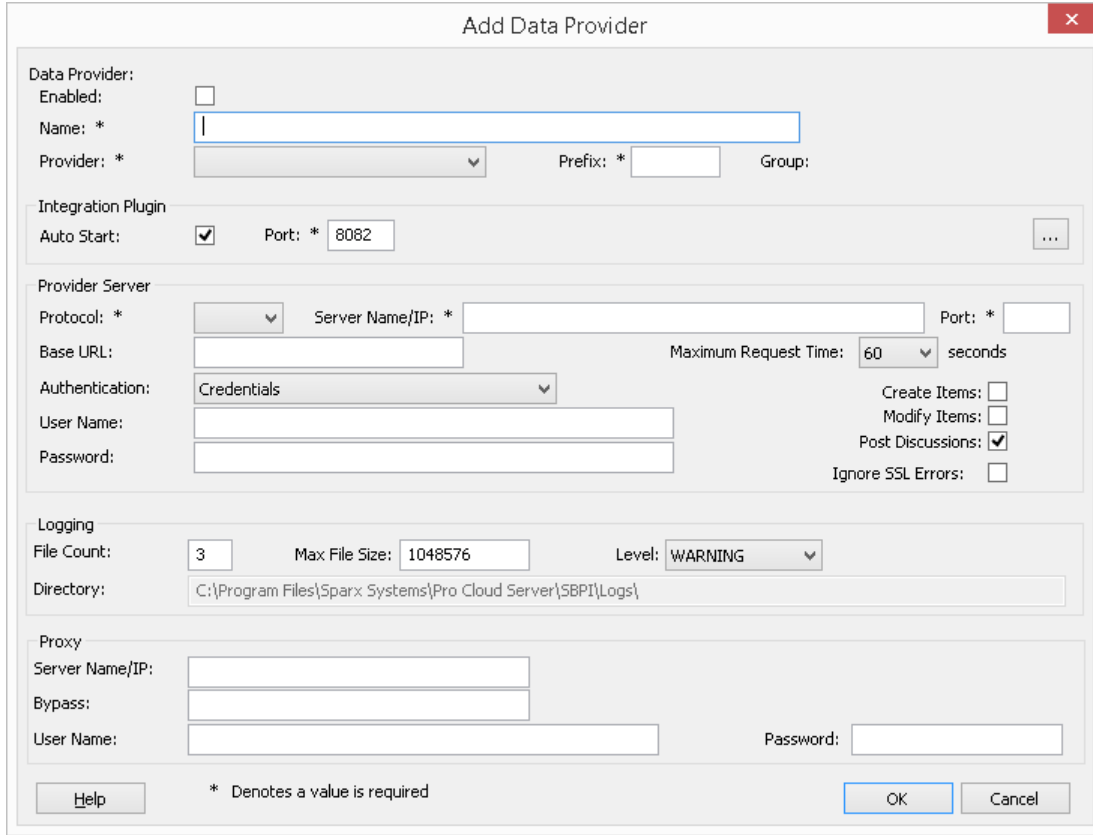
## 向专业云服务器和Enterprise Architect识别 ServiceNow

完成以下步骤：

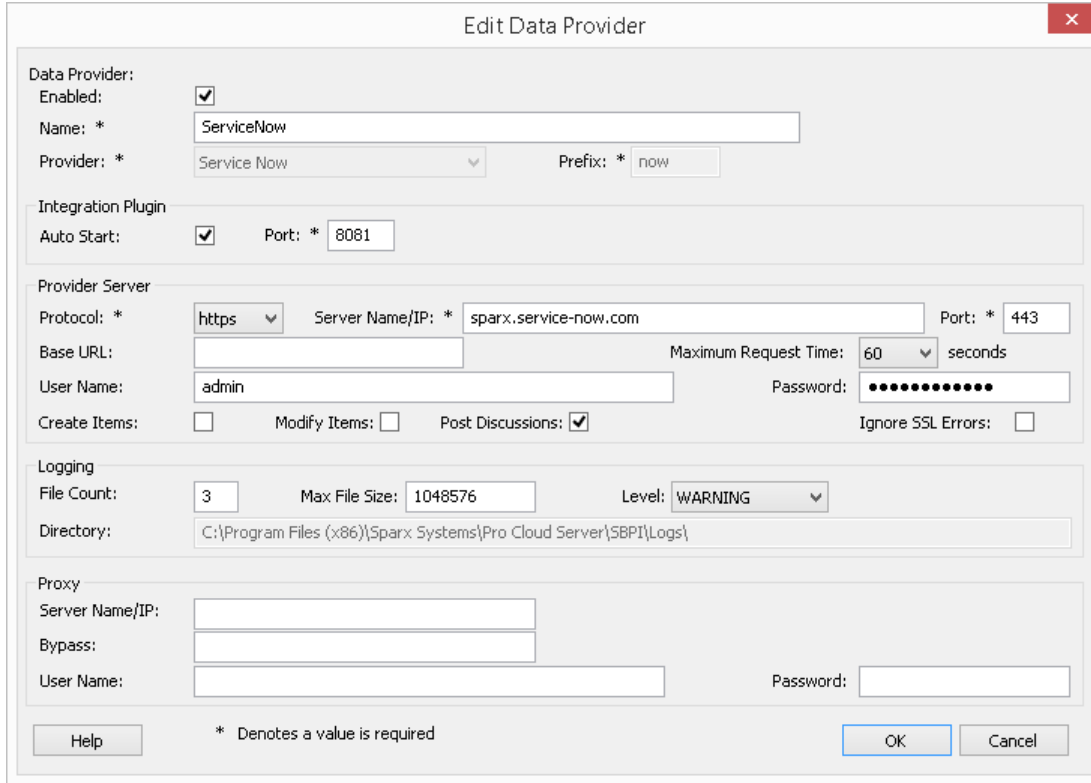
- 1. 双击“Pro云”桌面图标以打开Pro云客户端窗口。



- 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“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。




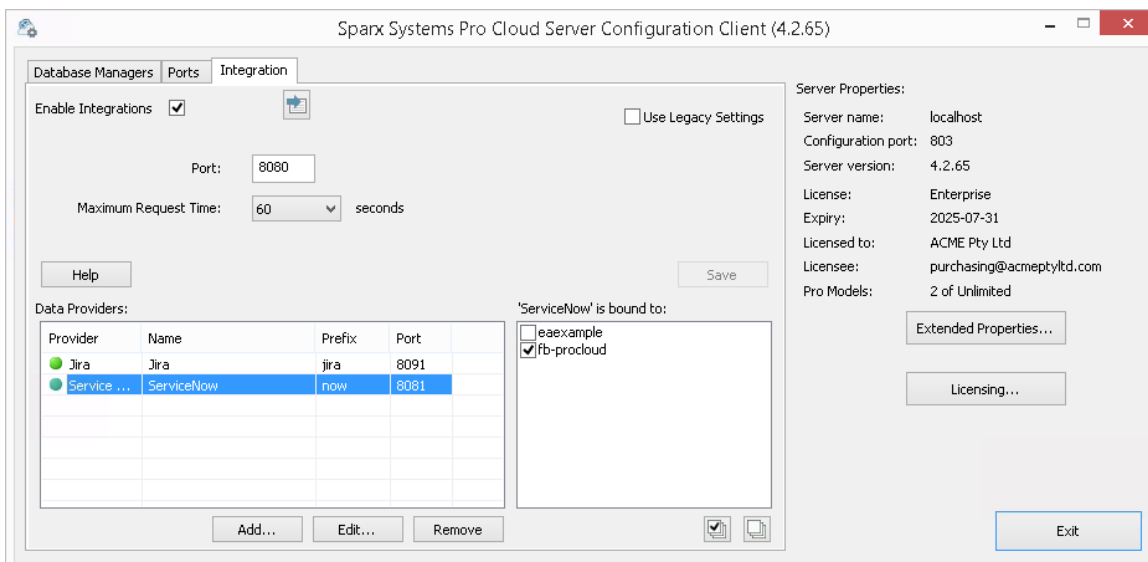
16. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的“集成”选项卡。

## 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。

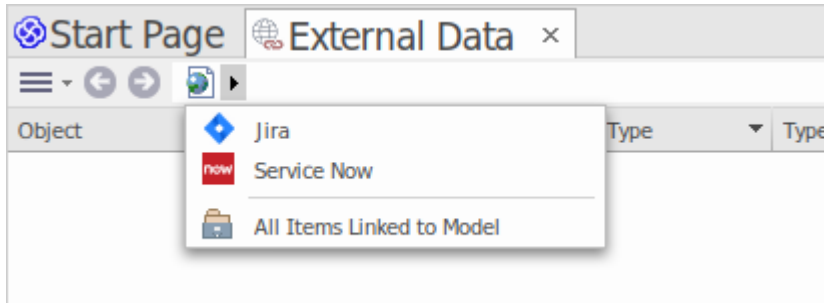
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 ServiceNow 的集成。

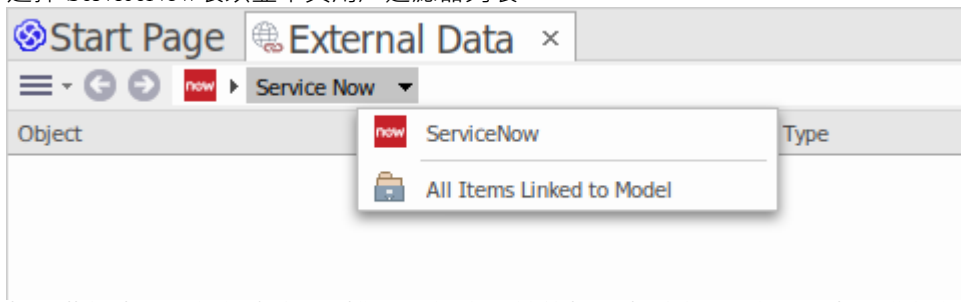
## 测试集成

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：

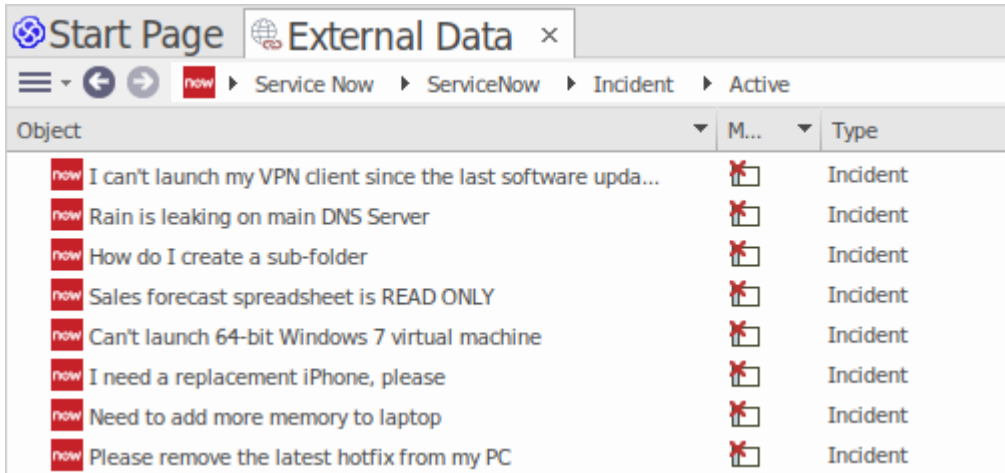
- 哪种类型的 ServiceNow 项对应于哪种类型的Enterprise Architect元素，以及
- 属性项的哪些属性对应于Enterprise Architect元素的哪些属性

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。



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://api-tokens-938839638.html)。请参阅下面“了解更多”下的链接。

### 先决条件

为了建立Jira和Enterprise Architect之间的集成，您必须首先：

- 正如刚才所讨论的，将专业云服务器的许可安装与集成插件组件一起使用
- 在托管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服务器之间具有网络访问
- 拥有访问外部数据的用户凭证
- 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
- 使用Enterprise Architect企业、统一或终极版本v14或更高版本

我们假设：

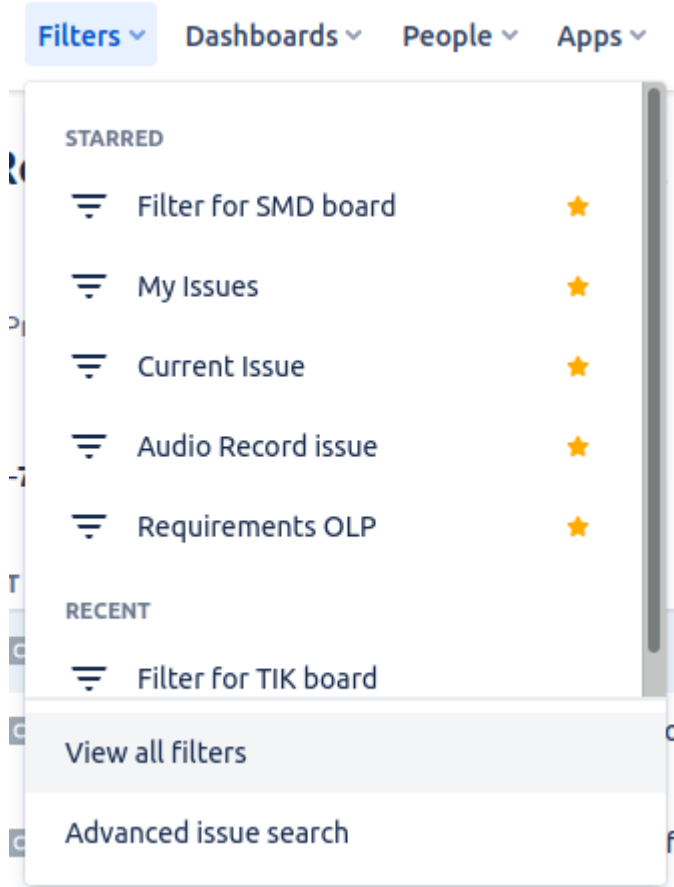
- 集成服务器和Jira集成插件与专业云服务器在同一台机器上运行
- 您使用专业云服务器配置客户端中的“集成服务器”选项卡来定义集成服务器选项，而不是通过手动操作注册表或配置文件或使用WebConfig界面

## 在 Jira 中定义星标过滤器

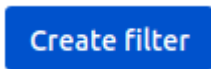
专业云服务器的 Jira 集成允许访问由 Jira 加星标过滤器返回的项目。在使用集成之前，您应该确保已设置一些加星标的过滤器以返回您希望在 Enterprise Architect 中可访问的项目。

要在 Jira 中定义星标过滤器，请执行以下步骤：

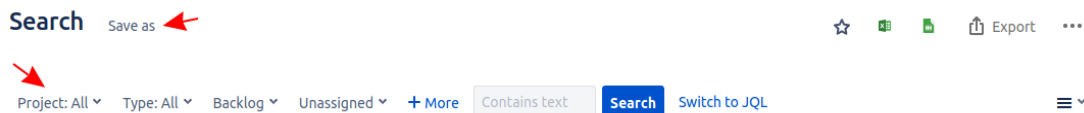
1. 在 Jira 中，选择“过滤器”下拉菜单，然后选择“视图所有过滤器”。



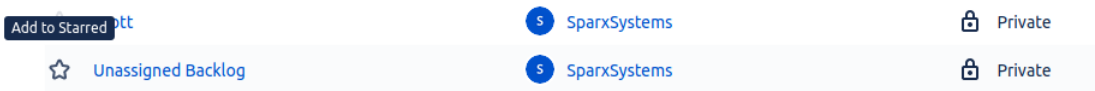
2. 在过滤器列表的右上角，单击“创建过滤器”按钮。



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。

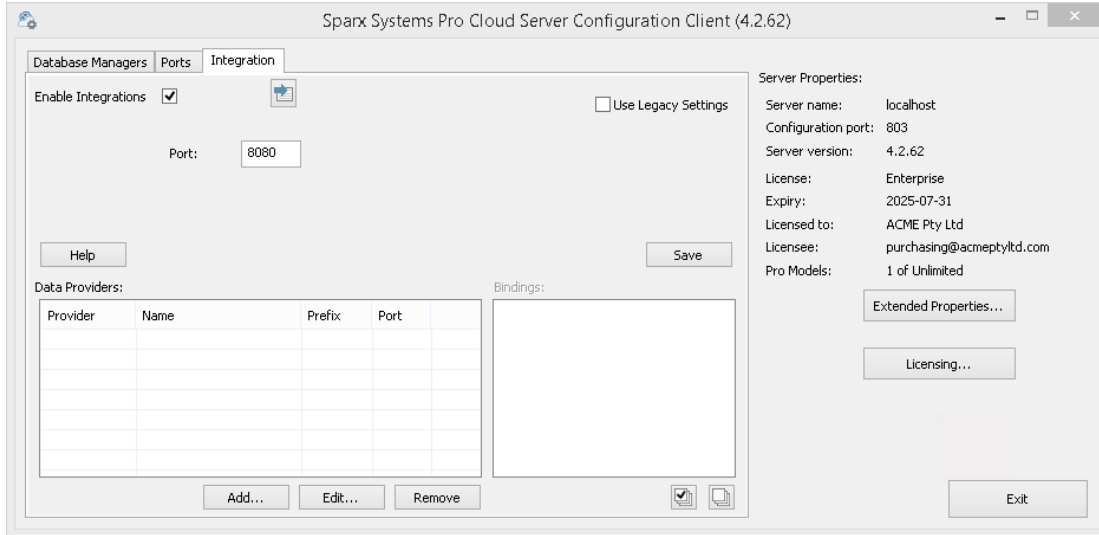


5. 专业云服务器的现在集成现在应该可以访问此过滤器（及其返回的项目）

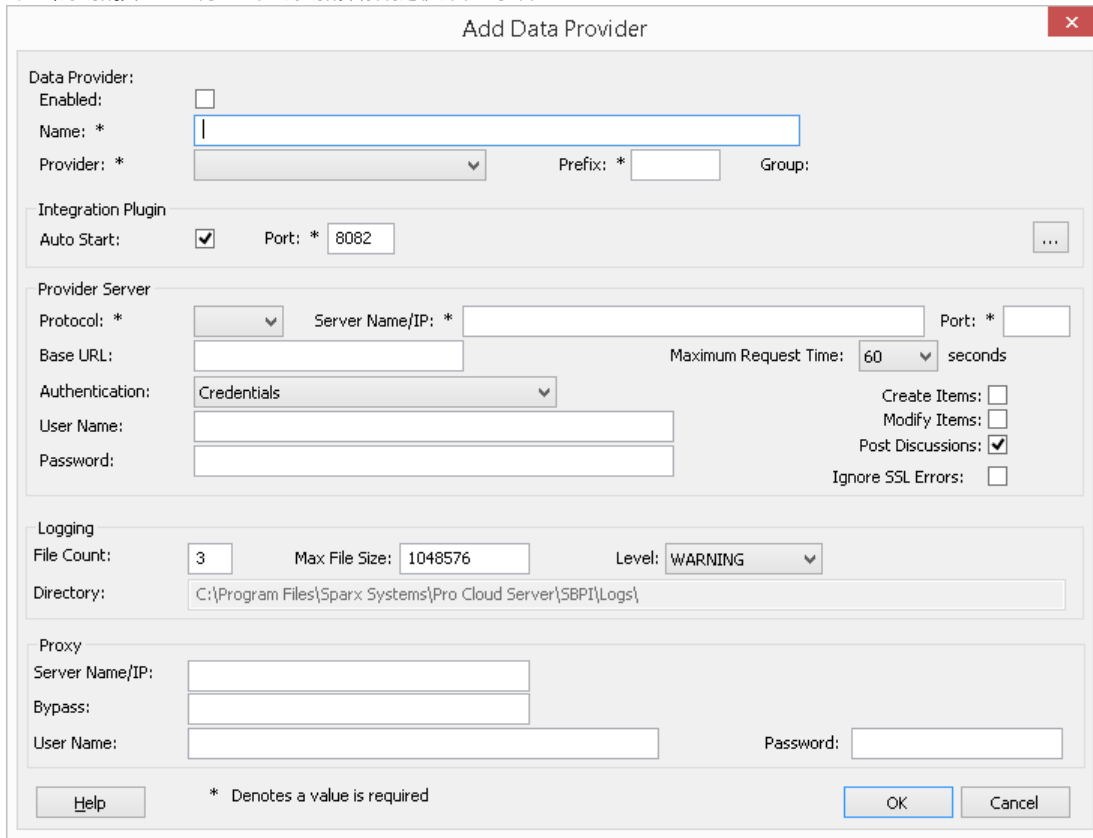
## 将 Jira 识别为专业云服务器和 Enterprise Architect

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。

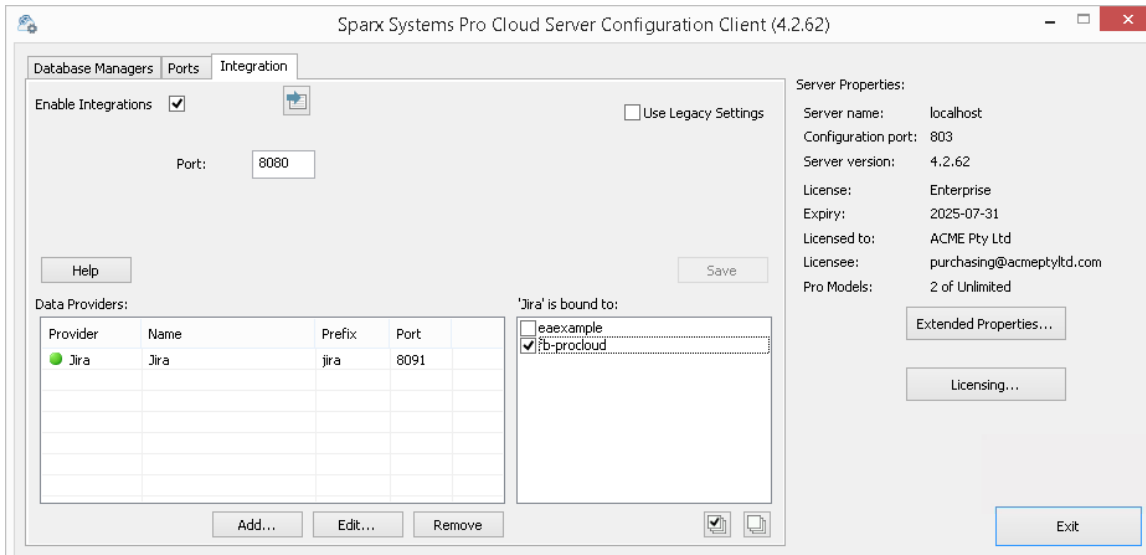
16. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的 集成”选项卡。

## 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。

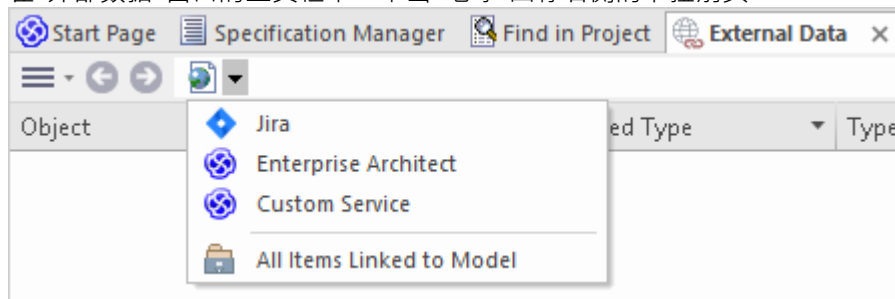
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 Jira 的集成。

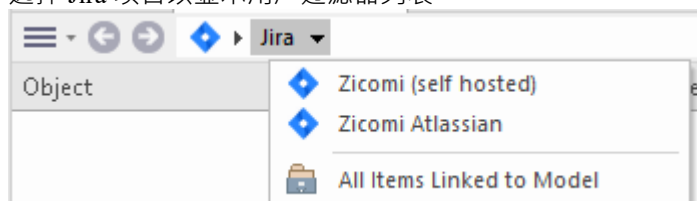
## 测试集成

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：

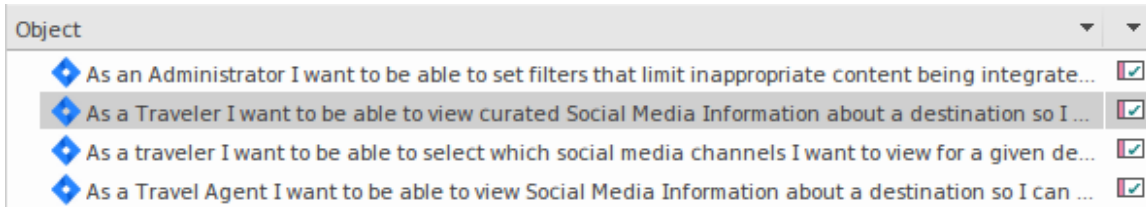
- 哪种类型的 Jira 项对应于哪种类型的Enterprise Architect元素，以及
- 属性项的哪些属性对应于Enterprise Architect元素的哪些属性。

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 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：

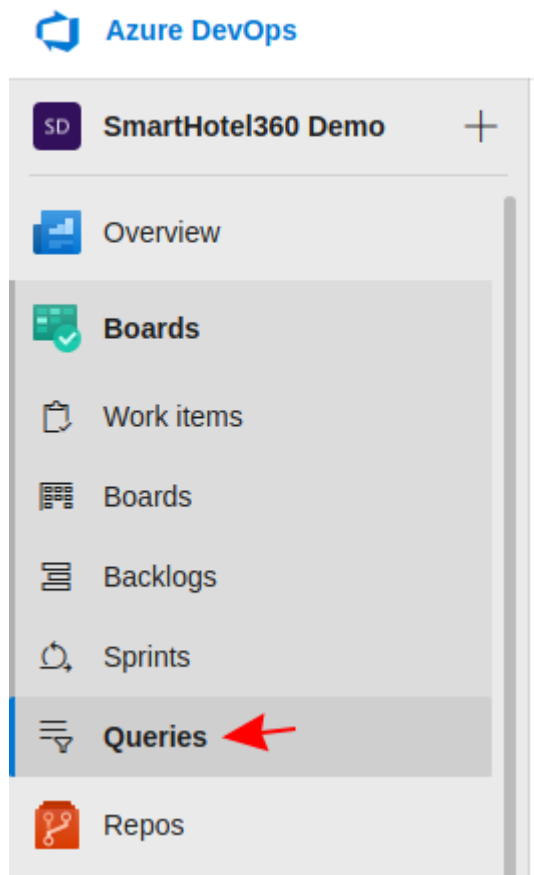
- 使用带有集成插件组件的专业云服务器的许可安装
- 在托管您的Enterprise Architect模型的每个数据库管理员系统的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项
- 拥有 PCS 服务器和 DevOps 之间的网络访问权限
- 拥有访问外部数据的用户凭据
- 正在使用Enterprise Architect企业、统一或终极版 v14 或更高版本

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

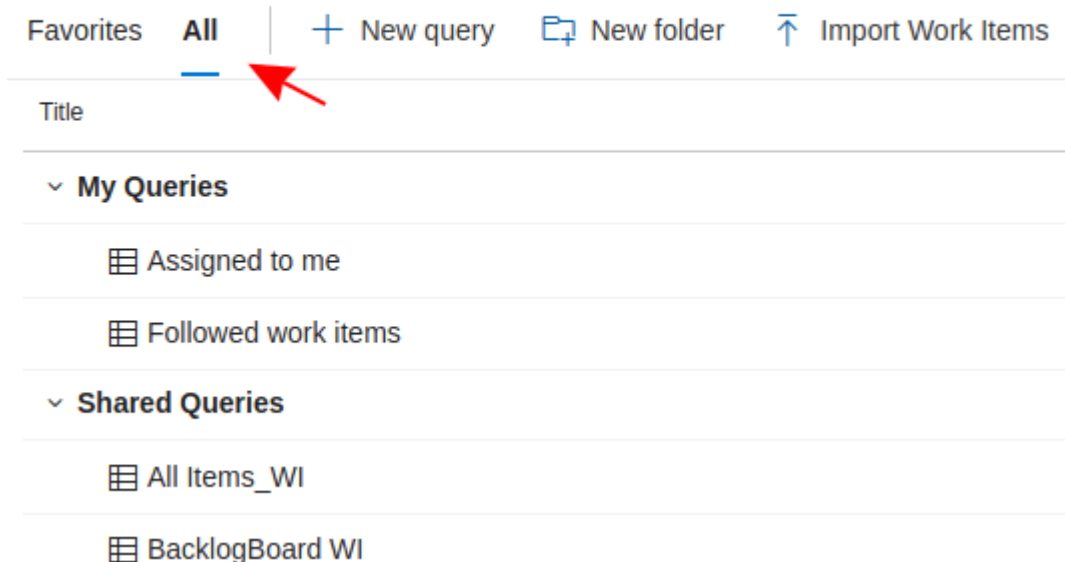
DevOps 集成允许通过在 DevOps 中定义的查询访问工作项。您可以查看现有查询并创建新查询，如下所述。

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



4. 可以A 新查询”按钮创建新查询。



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 ch

Type of query 📄 Flat list of work items

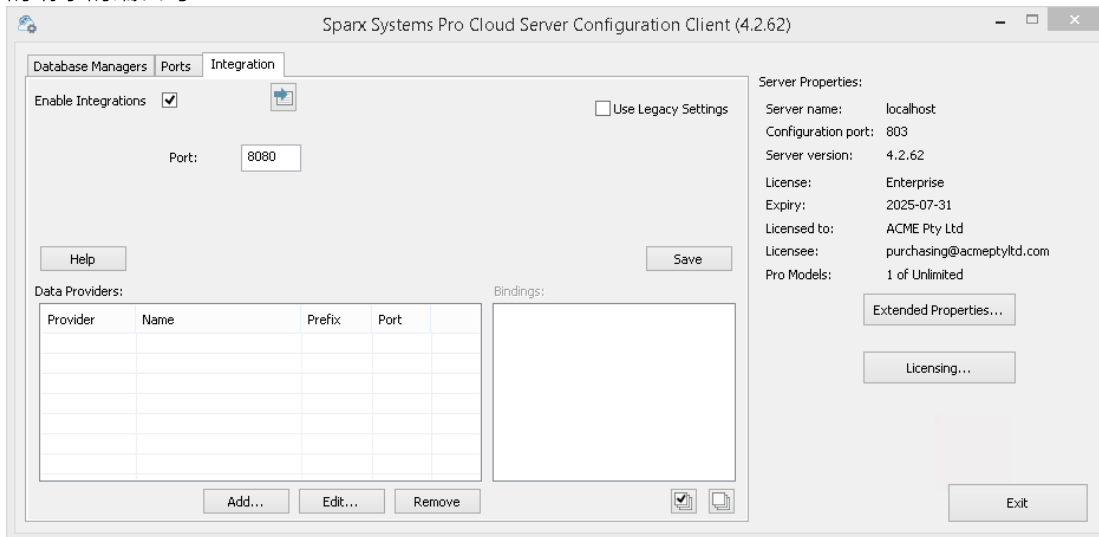
Filters for top level work items

	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）。

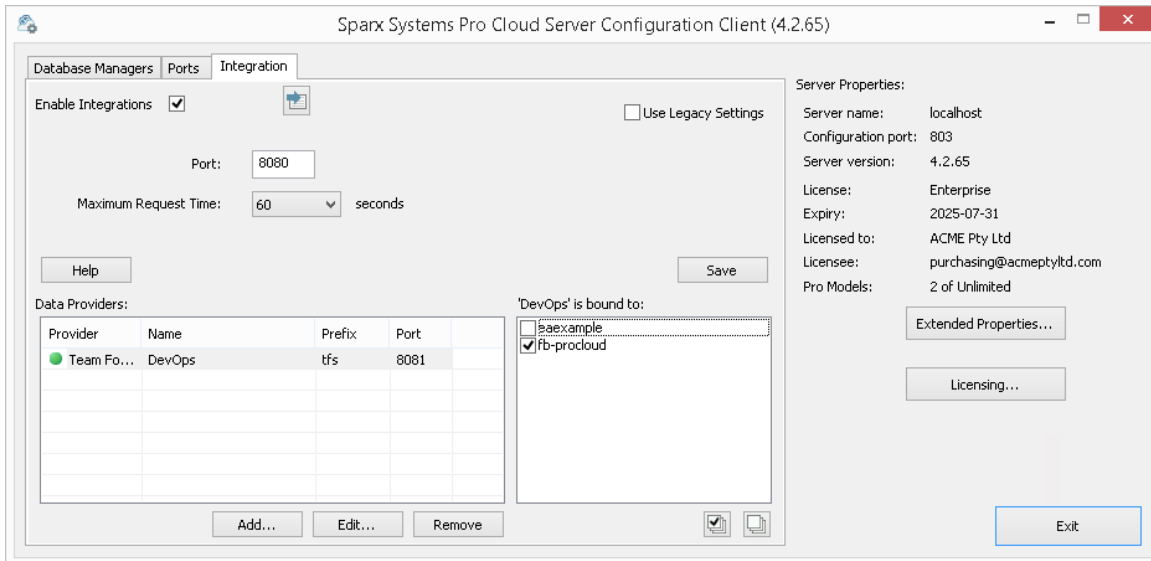
9. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会将您返回到专业云服务器配置客户端窗口的“集成”选项卡。

## 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。

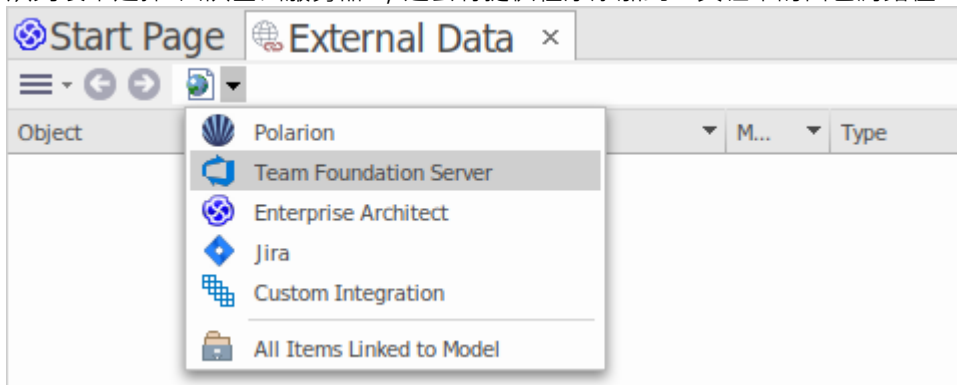
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 DevOps 的集成。

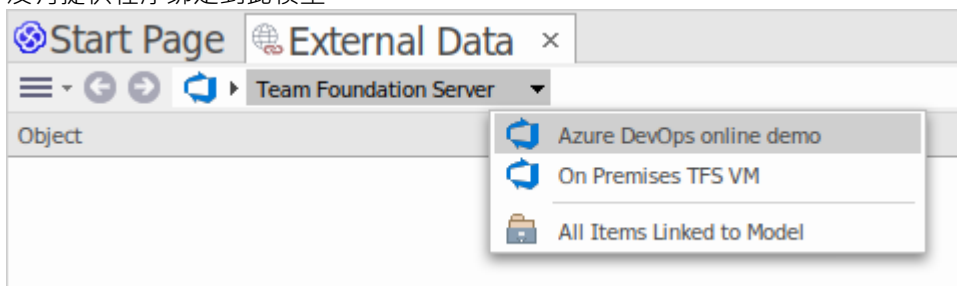
### 测试集成

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”并从列表中选择一个查询。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
IT 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：Jama集成

## 先决条件

为了建立 Jama 和Enterprise Architect之间的集成，您必须首先：

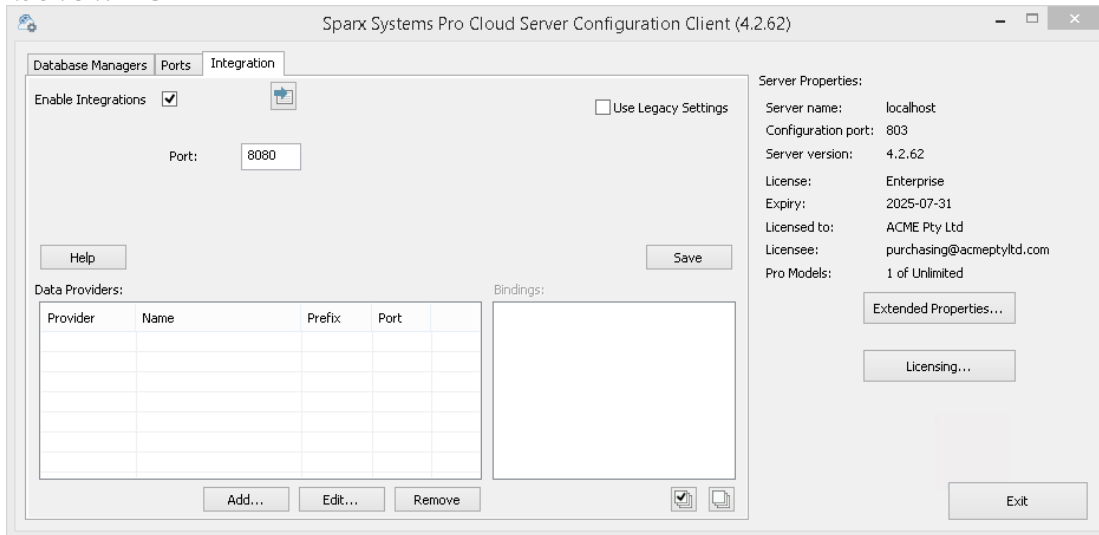
- 将专业云服务器（版本5.0或更高版本）的许可安装与集成插件组件一起使用
- 在专业云服务器配置中为托管Enterprise Architect模型的每个数据库管理员系统选择 启用专业特征（OSLC、WebEA和集成）”选项
- PCS 服务器和 Jama 服务器之间有网络访问
- 拥有访问外部数据的用户凭证
- 使用Enterprise Architect企业、统一或终极版 v16.0 或更高版本

我们假设集成服务器和 Jama集成插件与专业云服务器在同一台机器上运行

## 确定 Jama 到专业云服务器和Enterprise Architect

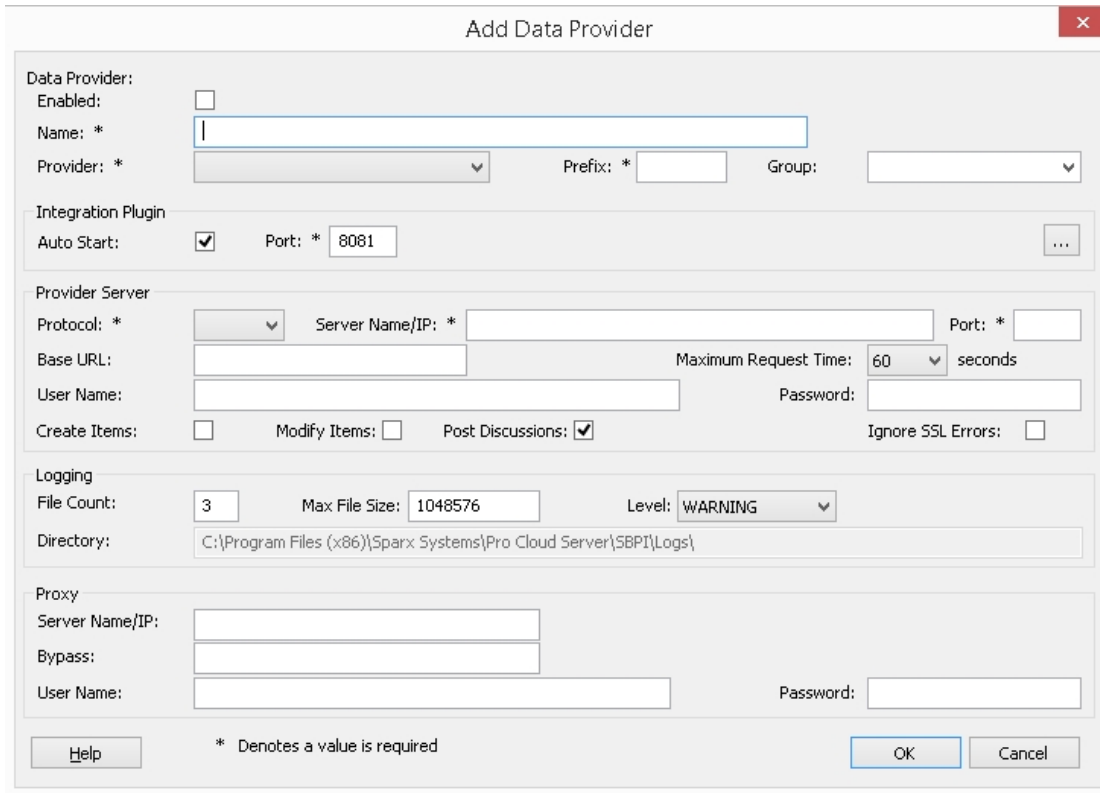
请按照以下步骤操作：

1. 双击 “Pro云Config Client”桌面图标，打开Pro云Config 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，输入集成服务器将侦听来自专业云服务器的请求的端口号。

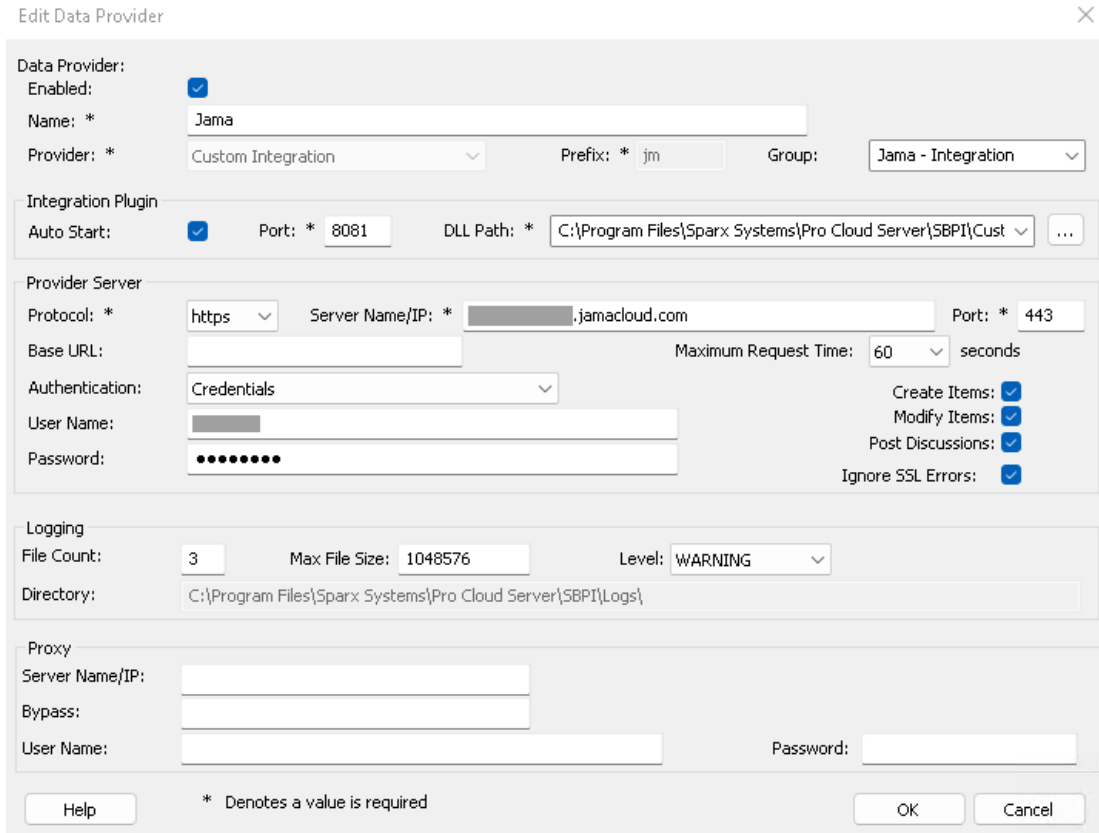


3. 单击添加按钮。将显示 “添加数据提供者”对话框。





4. 选中 已启用“复选框”。
5. 在 名称”字段中，输入适当的连接名称，例如 *Jama*”。
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 *Jama*’。
8. 在 组”字段中输入浏览 *Jama* 集成时的根级别名称（例如 *Jama -集成*”）。
9. 在 集成插件”面板的 端口”字段中，键入集成服务器将用于与 *Jama*插件进行通信的端口号。
10. 单击 DLL 路径”字段旁边的省略号按钮。浏览并选择 *JamaIntegrationPlugin.dll*”文件。它位于 专业云服务器安装文件夹内。 *E C:\Program Files\ Sparx Systems \专业云服务器\SBPI\JamaIntegrationPlugin.dll*
11. 步骤 11 至 14 中的字段构成了您的 *Jama* 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 *Jama*插件用于与 *Jama* 通信的协议。
12. 在 服务器名称/IP”字段中，键入 *Jama*插件用于与 *Jama* 通信的服务器名称或 IP 地址。
13. 在 端口’字段中输入 *Jama*插件用于与 *Jama* 通信的端口号。
14. 基地 “URL”字段可以留空。
15. 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 *Jama* 中的数据。*Jama* 安装的所有当前 Enterprise Architect用户都将使用同一套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 *Jama*插件的用户使用Enterprise Architect在 *Jama* 中创建和/或更新项目和/或创建讨论帖子。
17. 如果您希望 *Jama*插件忽略与 *Jama* 通信时发生的 SSL 相关错误（例如，如果您使用自签名 SSL 证书），请选择 忽略 SSL 错误”复选框。
18. 如果您使用代理服务器访问 *Jama*，请在代理部分输入相关详细信息：  
**服务器名称/IP** - 代理服务器的服务器名称（或 IP）和端口号；例如， *proxyserver.com:3131*  
**绕过** - 此字段为某些地址提供绕过代理的机制；它接受以分号 (;) 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证，此字段提供用户密码。




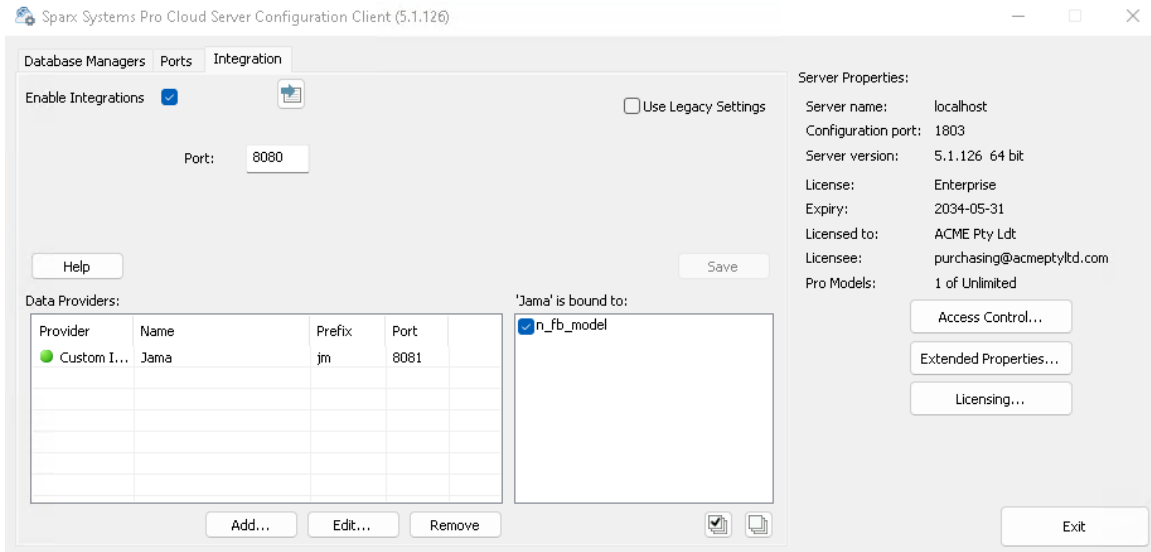
19. 您可以将其他字段留空或设置为其默认值。单击确定按钮完成配置。这将返回到专业云服务器配置客户端窗口的“集成”选项卡。

## 向 Jama 确认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：

- 屏幕左下方的“数据提供商”面板将集成Jama 列为数据提供者
- 在屏幕的右下角，'Jama'绑定到：'面板显示来自您的Pro-enabled数据库管理员的Enterprise Architect模型的检查清单

您可以单击要绑定到 Jama 的每个Enterprise Architect模型对应的复选框，也可以单击  一次性选择所有模型。



单击退出按钮。

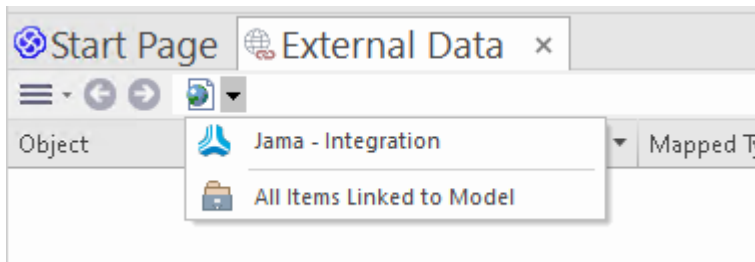
重新启动 PCS 服务以应用所做的更改。

您现在可以进入您的一个Enterprise Architect模型并检查与 Jama 的集成。

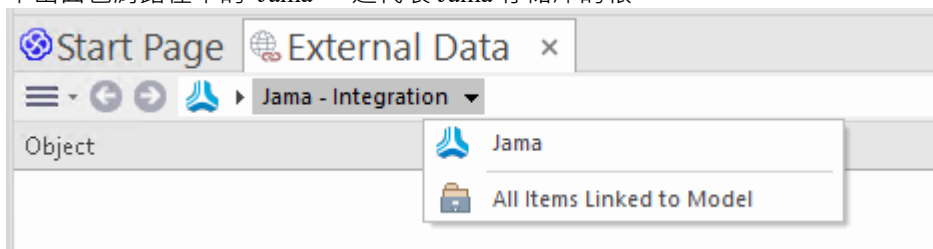
## 测试集成

在Enterprise Architect中打开您已确定与 Jama 绑定的模型之一。

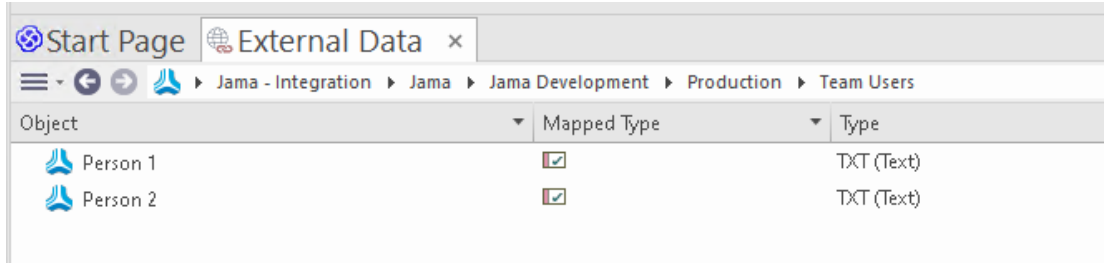
1. 选择功能区选项特定> 工具">系统集成"。
2. 在外部数据窗口的工具栏中，单击地球"图标右侧的下拉箭头。
3. 从列表中选择集成组（例如 Jama集成"或 Custom集成"）。这是您在 添加数据提供者"对话框中输入的组"。



4. 单击面包屑路径中的 'Jama'，这代表 Jama 存储库的根。



5. 继续使用面包屑选项深入了解 Jama 项目并访问工作项（需求、问题等）。

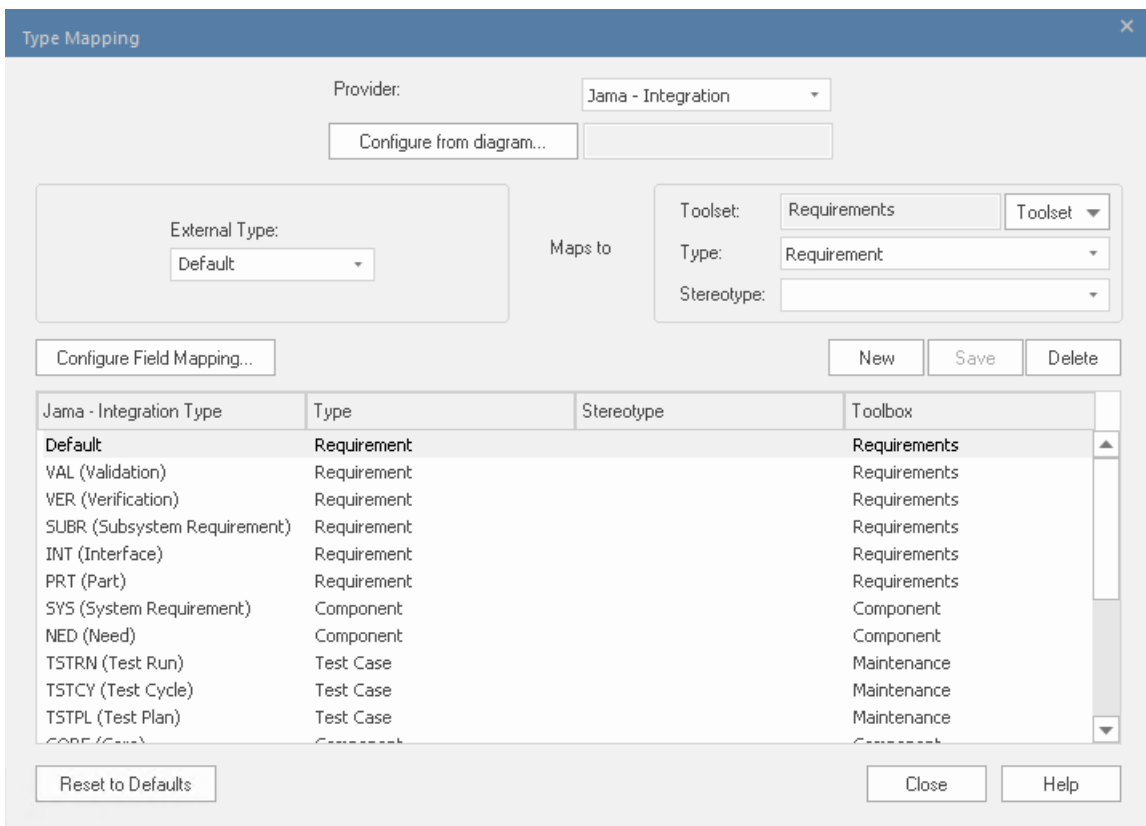


Object	Mapped Type	Type
Person 1	<input checked="" type="checkbox"/>	TXT (Text)
Person 2	<input checked="" type="checkbox"/>	TXT (Text)

### 数据映射

为了使 Jama 和Enterprise Architect能够顺利交换数据项，您应该审阅并（如有必要）编辑：

- 哪种类型的 Jama 项目对应哪种类型的Enterprise Architect元素，以及
- Jama 项目的哪些属性对应于Enterprise Architect元素的哪些属性。



Provider: Jama - Integration

External Type: Default

Maps to: Requirement

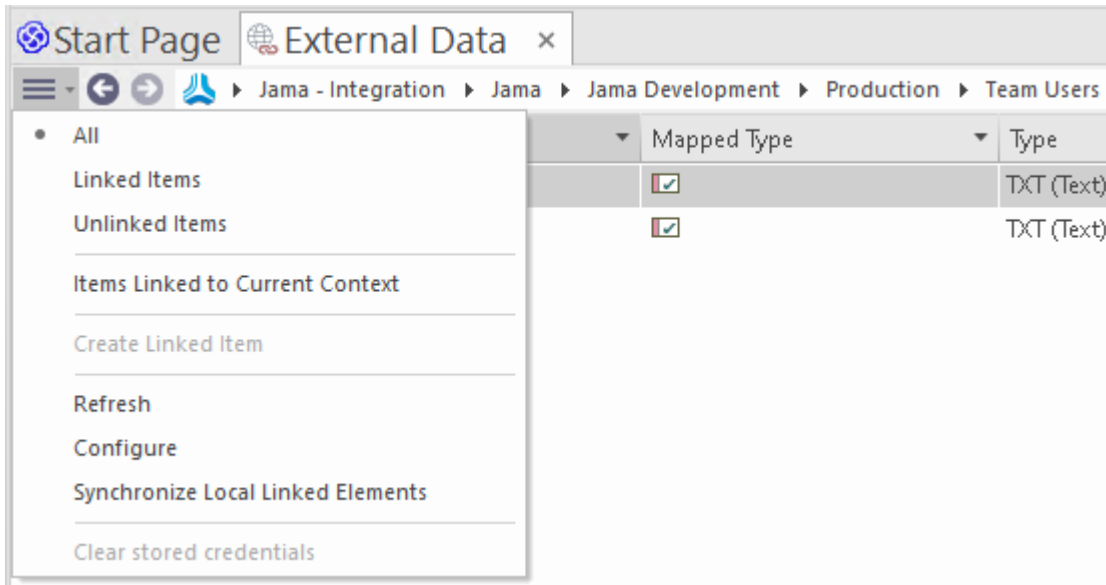
Toolset: Requirements

Jama - Integration Type	Type	Stereotype	Toolbox
Default	Requirement		Requirements
VAL (Validation)	Requirement		Requirements
VER (Verification)	Requirement		Requirements
SUBR (Subsystem Requirement)	Requirement		Requirements
INT (Interface)	Requirement		Requirements
PRT (Part)	Requirement		Requirements
SYS (System Requirement)	Component		Component
NED (Need)	Component		Component
TSTRN (Test Run)	Test Case		Maintenance
TSTCY (Test Cycle)	Test Case		Maintenance
TSTPL (Test Plan)	Test Case		Maintenance
CORE (Core)	Component		Component

这些任务在[Configuration](#)帮助主题中有完整解释。

### 审阅Jama Data

当您测试Enterprise Architect与 Jama 之间的通信时，在“外部数据”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，您可以单击窗口工具栏中的“汉堡包”图标。



此菜单提供以下选项：

- '项' 仅显示与Enterprise Architect元素链接的 Jama 项目
- 未链接项目“仅显示尚未链接到Enterprise Architect元素的 Jama 项目，或
- 链接到当前上下文的项”仅显示链接到当前选定的Enterprise Architect元素Jama 项

还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 “Toggle过滤器Bar” (如果它尚未显示)。
2. 在列名下方的过滤器/搜索字段中类型一些文本。

从列表中选择一个项目时，属性窗口、笔记窗口、检验员窗口和讨论与审阅窗口的“讨论”选项卡均有单独的选项卡或窗口版本，分别显示所选项目的任何属性、笔记、特征和讨论。这些内容在 [External Item Details](#)帮助主题。

## 使用 Jama 项目和Enterprise Architect元素

在Enterprise Architect和 Jama 之间建立工作沟通后，您可以：

- 在Enterprise Architect中创建一个链接到 Jama 项目的新元素
- 在Enterprise Architect中创建一个链接到元素的新 Jama 项目
- 将Enterprise Architect中的现有元素链接到 Jama 项目
- 使用对Enterprise Architect元素的更改来更新 Jama 项目 (‘推送’更改)
- 使用对 Jama 项的更改来更新Enterprise Architect元素 (‘拉动’更改)
- 使用对 Jama 项目的任何更改来更新所有链接的Enterprise Architect元素 (‘同步全部’)

这些操作均在[Linking Items](#)帮助主题中进行了解释。

# 演练：Polarion集成

## 先决条件

为了建立 Polarion 和Enterprise Architect之间的集成，您必须首先：

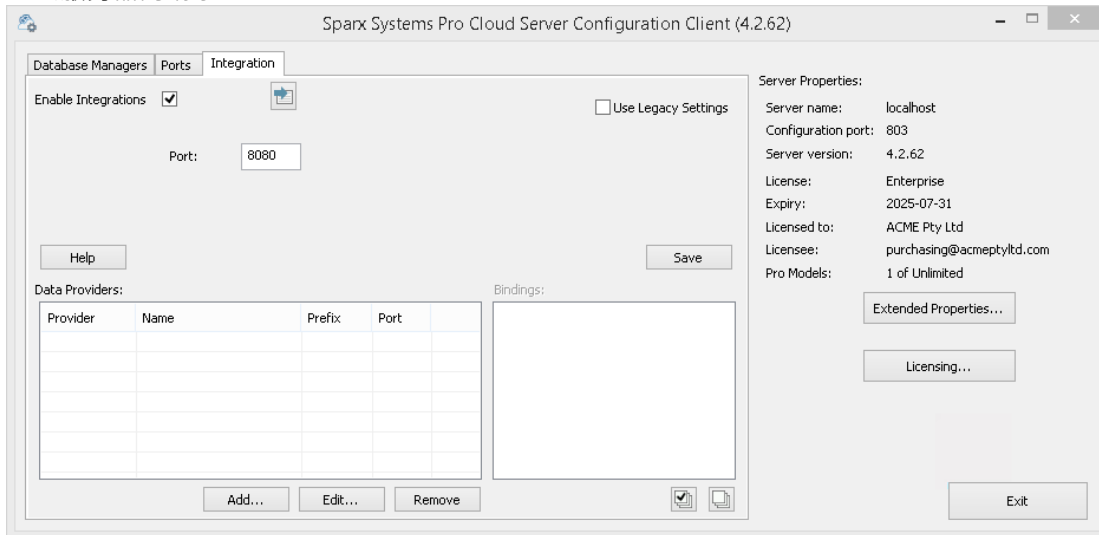
- 将专业云服务器（版本4.2.62或更高版本）的许可安装与集成插件组件一起使用
- 在专业云服务器配置中为托管Enterprise Architect模型的每个数据库管理员系统选择“启用专业特征（OSLC、WebEA和集成）”选项
- PCS 服务器和 Polarion 服务器之间有网络访问
- 拥有访问外部数据的用户凭证
- 使用Enterprise Architect企业、统一或终极版 v15.2 或更高版本

我们假设集成服务器和Polarion集成插件与专业云服务器在同一台机器上运行

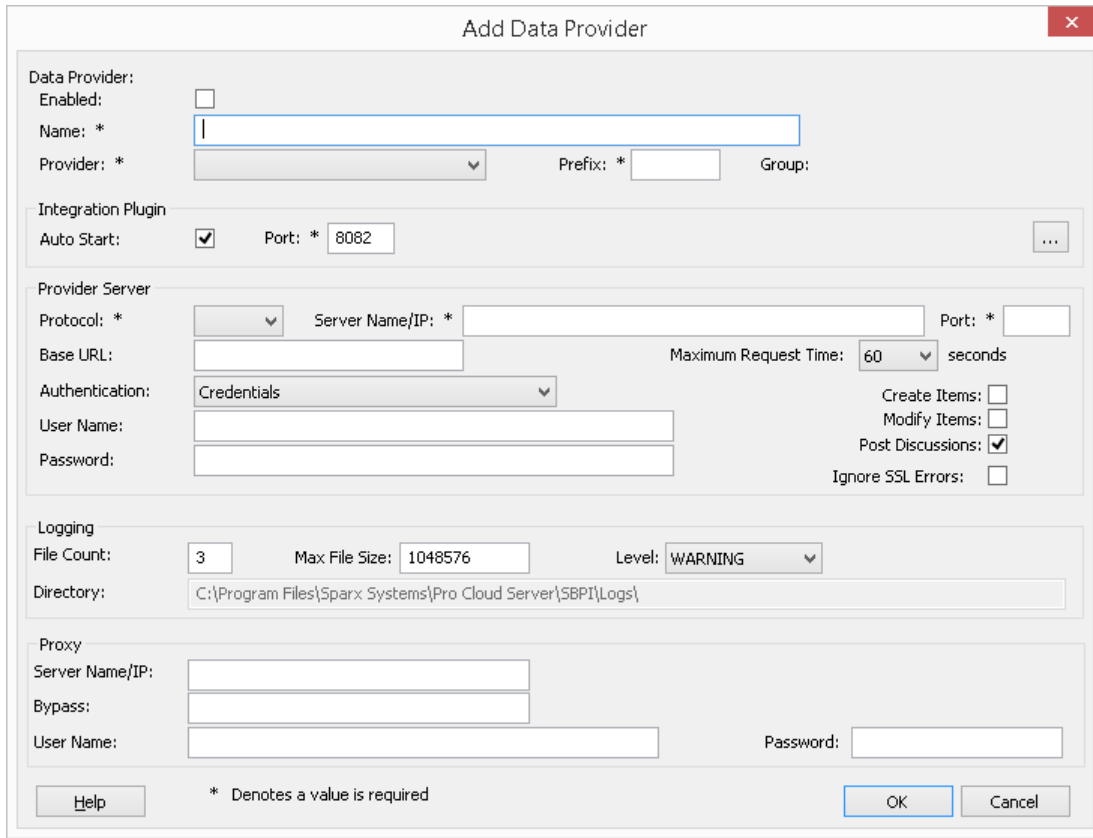
## 向专业云服务器和Enterprise Architect识别 Polarion

完成以下步骤：

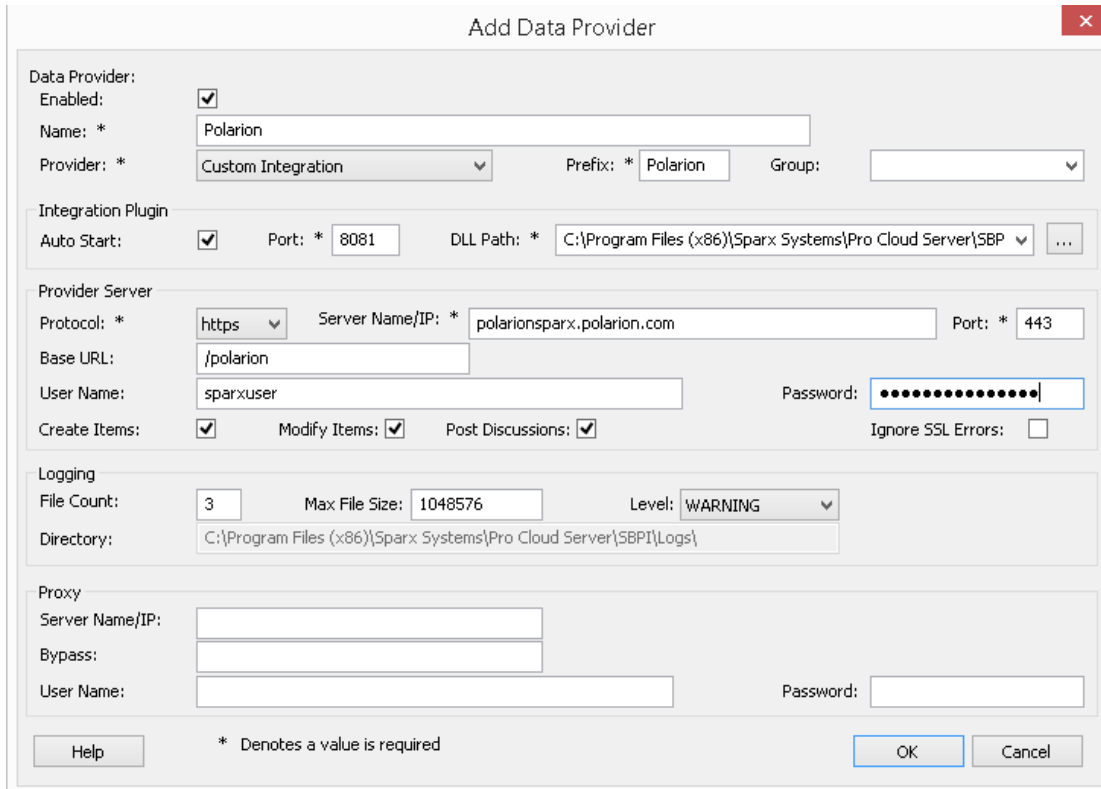
1. 双击“Pro云Client”桌面图标以打开Pro云Client 窗口。
2. 选择“集成”选项卡并选择“启用集成”复选框。在“端口”字段中输入端口的数量，该集成服务器将侦听来自专业云服务器的请求。



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 插件端口将”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。



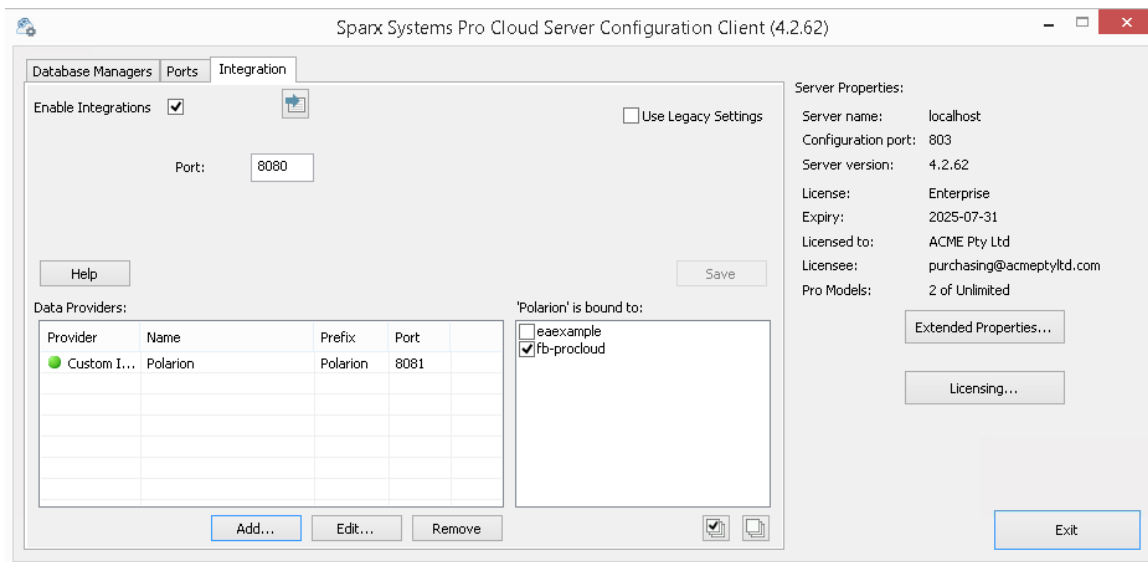
18. 您可以将其他字段留空或设置为其默认值。点击确定按钮完成配置。这会让您返回到专业云服务器配置客户端窗口的“集成”选项卡。

## 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。

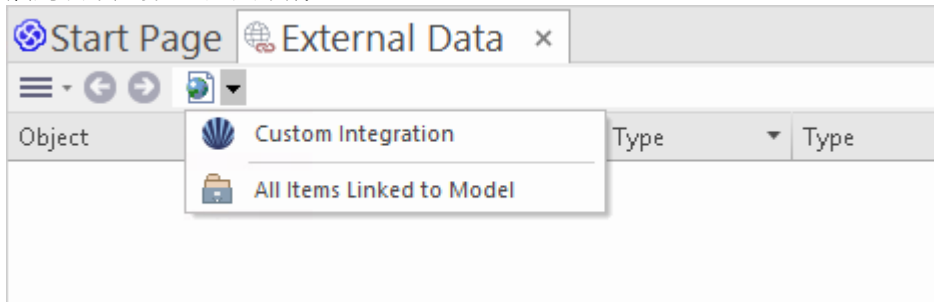
重新启动 PCS 服务以应用您所做的更改。

您现在可以进入您的Enterprise Architect模型之一并检查与 Polarion 的集成。

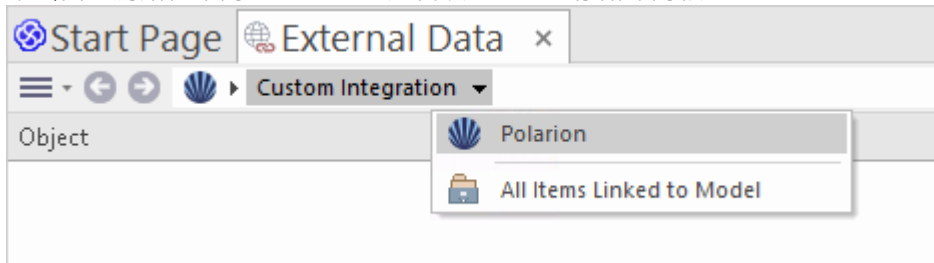
## 测试集成

在Enterprise Architect中打开您已识别为绑定到 Polarion 的模型之一。

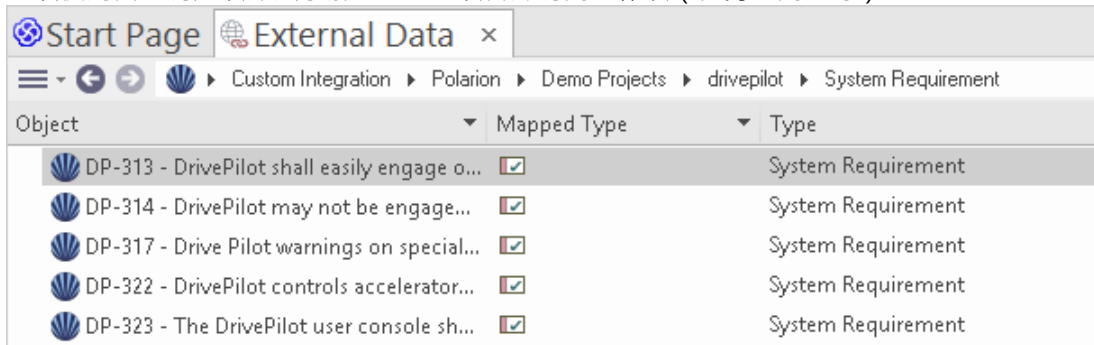
1. 选择功能区选项特定>工具>系统集成。
2. 在“外部数据”窗口的工具栏中，单击“地球”图标右侧的下拉箭头。
3. 从列表中选择“自定义集成”。



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：

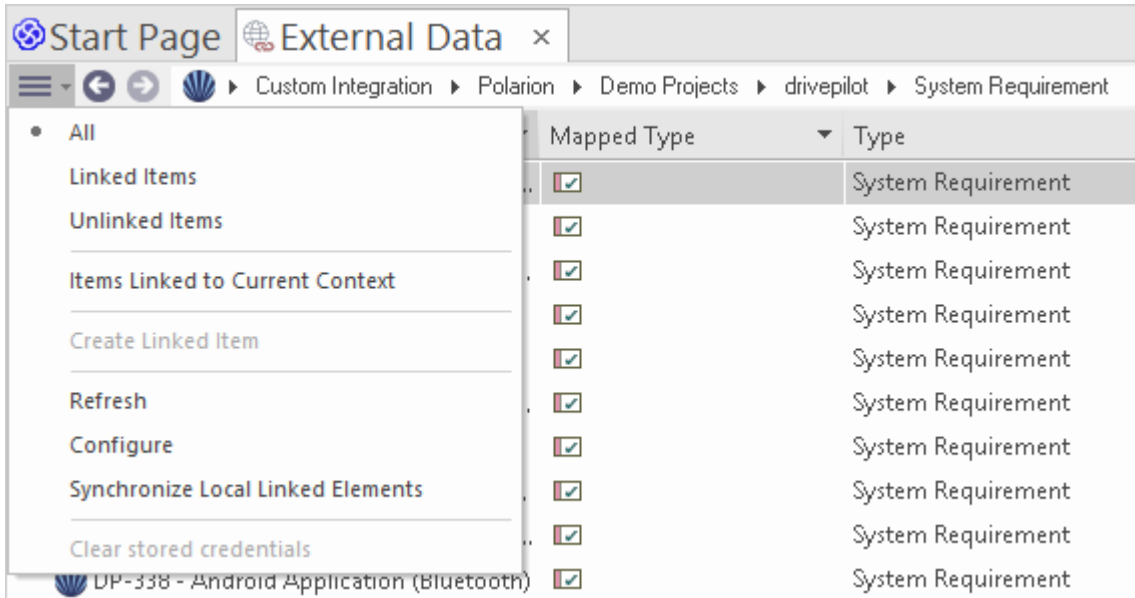
- 什么类型的 Polarion 项目对应于什么类型的Enterprise Architect元素，以及
- 属性项的哪些属性对应于Enterprise Architect元素的哪些属性。

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。

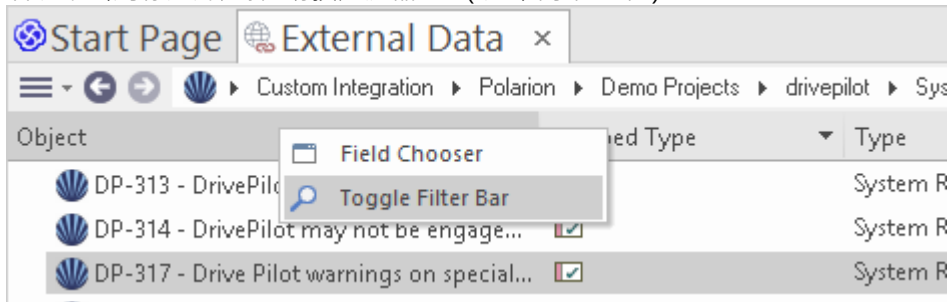


此菜单提供以下选项：

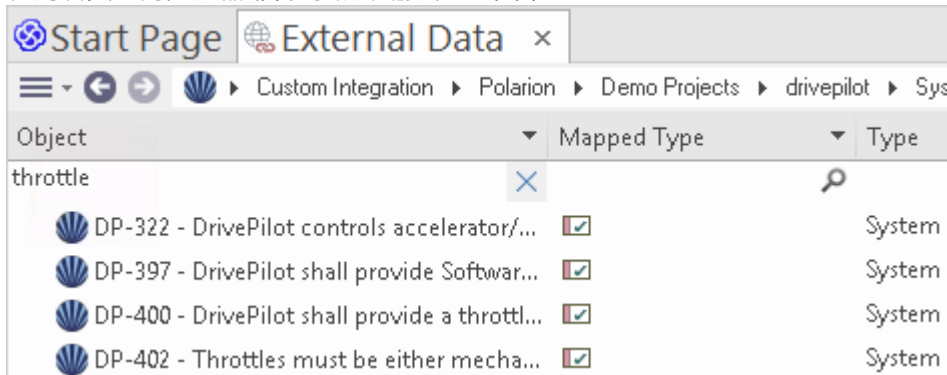
- “链接项”仅显示那些链接到Enterprise Architect元素的 Polarion 项目
- “未链接的项目”仅显示尚未链接到Enterprise Architect元素的 Polarion 项目，或
- “链接到当前上下文的元素项”仅显示链接到当前选定的Enterprise Architect的 Polarion 项目

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：

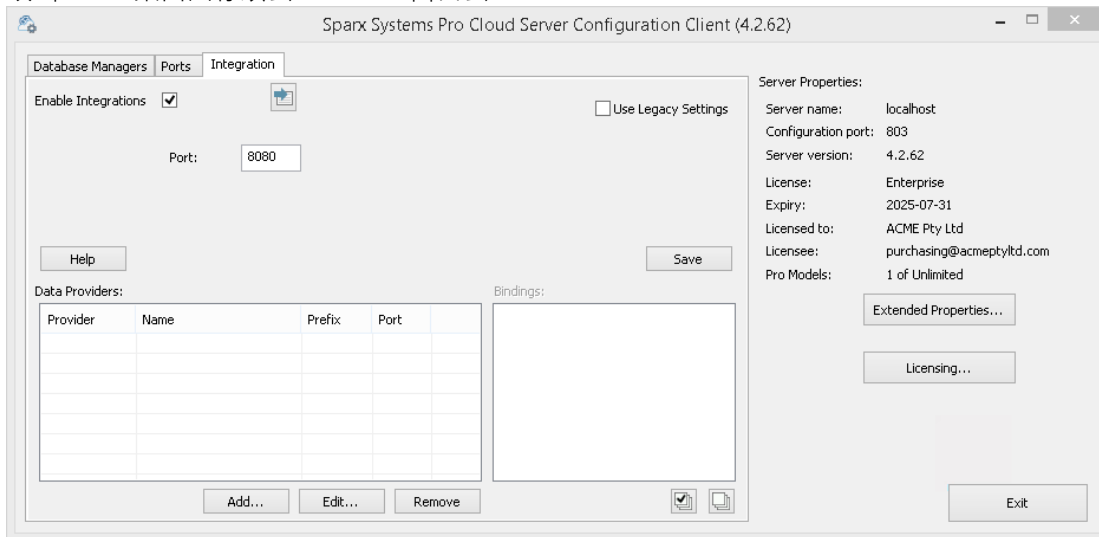
- 正如刚才所讨论的，将专业云服务器的许可安装与集成插件组件一起使用
- 源和目标模型都应该可以通过专业云服务器访问
- 在每个模型/数据库管理器的专业云服务器配置中选择“启用专业特征（OSLC、WebEA和集成）”选项。
- 在PCS服务器和托管外部EA模型PCS服务器之间具有网络访问（这仅适用于使用不同/外部PCS托管目标模型情况）
- 拥有访问目标模型的用户凭证
- 使用Enterprise Architect企业、统一或终极版本 v14 或更高版本

在本演练中，我们假设您使用专业云服务器配置客户端中的“集成”选项卡来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：

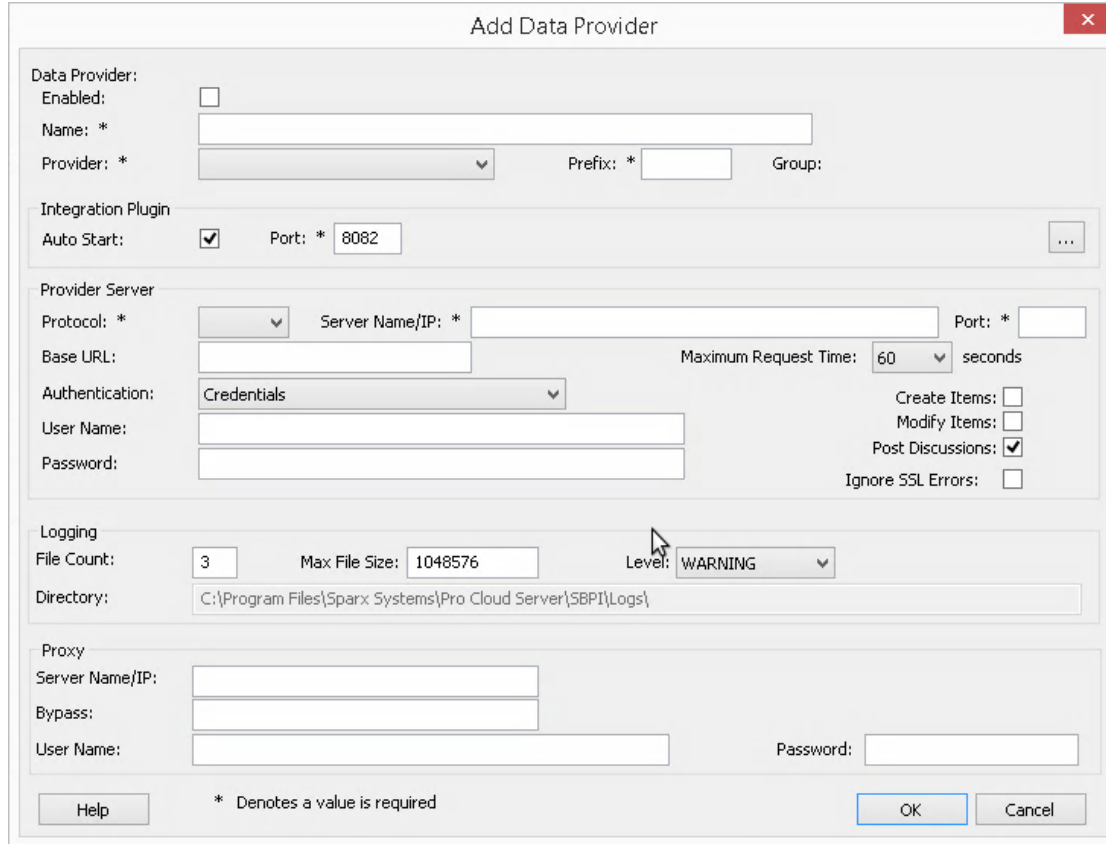
1. 双击“Pro”桌面图标以云Pro Client窗口云



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。

- 3. 单击添加按钮。显示“添加数据提供者”对话框。



- 4. 选择“启用”复选框。
- 5. 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”。
- 6. 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。
- 7. 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）。
- 8. 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。
- 9. 在“协议”字段中，单击下拉箭头并选择“http”或“https”。
- 10. 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”。
- 11. 在“端口”字段中输入用于访问目标模型端口号。
- 12. 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。
- 13. 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。
- 14. 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。
- 15. 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。

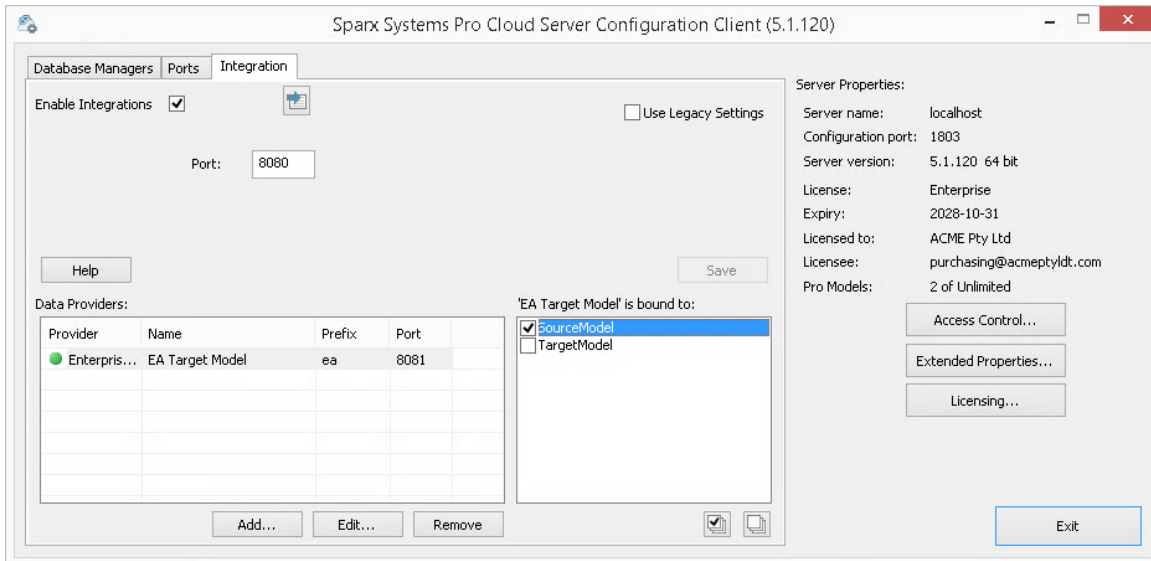
## 配置哪些源模型可以访问Enterprise Architect集成

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。

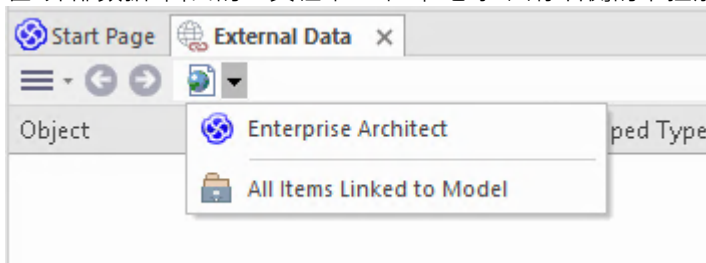
重新启动 PCS 服务以应用所做的更改。

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。

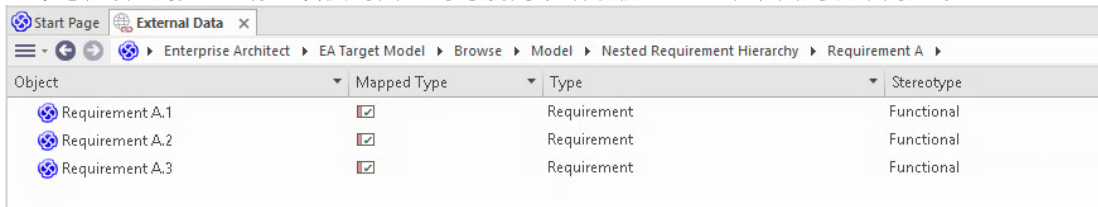
### 测试整合

在Enterprise Architect中打开您绑定到外部 目标”模型 源”模型之一。

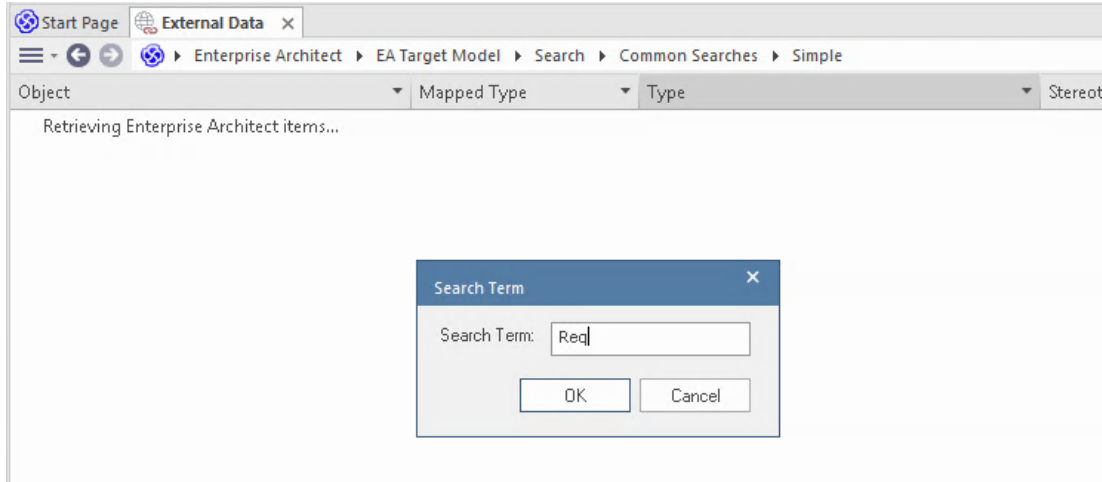
1. 选择功能区选项'特定>工具>系统集成'。
2. 在 外部数据”窗口的工具栏中，单击 地球”图标右侧的下拉箭头。



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。

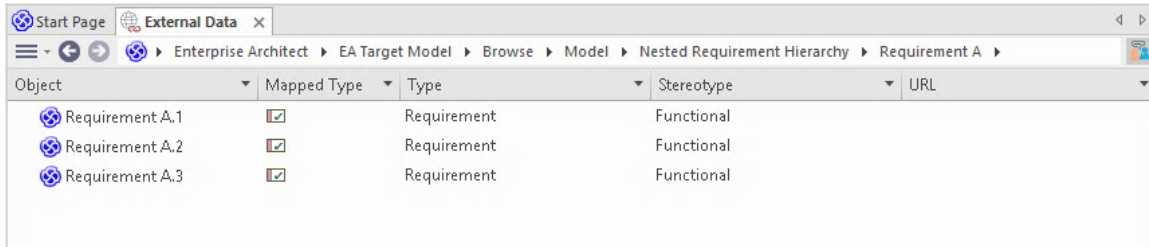


6. 如果您选择搜索，请选择一个搜索组（例如 公共搜索”）、一个搜索定义（例如 简单”），然后在提示中输入一个搜索团队。外部模型中与搜索词匹配的所有元素都将在下方列出。



### 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

本页提供的信息可帮助用户解决在设置集成服务器设置时可能遇到的问题。

## 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
<p>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）</p>	<p>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。</p>
<p>SBPI.EXE 的可执行文件名是预期的</p>	<p>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。</p>
<p>[full_path] 指定的集成服务器路径不存在或无效</p>	<p>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。</p>
<p>指定的端口是 [duplicate_plugin_name] 的副本</p>	<p>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。</p>

## 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
<p>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。</p>	<p>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。</p>
<p>端口需要在所有 Provider 中唯一，[new_port] 的端口值已被 [other_provider] 使用</p>	<p>原因：已经为当前的Data提供者定义了一个端口号；但是，该值与另一个提供者的端口重复。</p>
<p>提供程序的指定执行路径不存在或无效， [full_path]</p>	<p>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。</p>
<p>指定的端口是集成服务器端口的副本</p>	<p>原因：为当前Data提供者定义了一个端口号，但该端口值与集成服务器使用的端口重复。</p>
<p>无法将配置文件</p>	<p>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</p>

[old_filename] 重命名为 [new_filename]	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<a href="#">Integration Plug-ins</a>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>

# 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	返回的信息
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。

# 导航外部数据

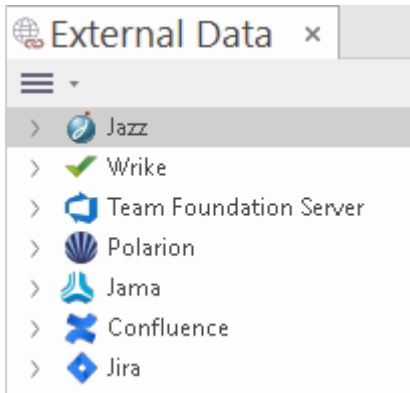
每个集成提供者都可以导航以显示可以链接到Enterprise Architect的外部项目列表。每个提供者可能会根据其存储数据的方式提供稍微不同的导航机制。有些提供简单的文件夹层次结构，而另一些则提供用户定义的过滤器。有关每个提供程序的详细信息，请参阅[Install and Configure](#)帮助主题中的集成插件返回什么数据表。

## 访问

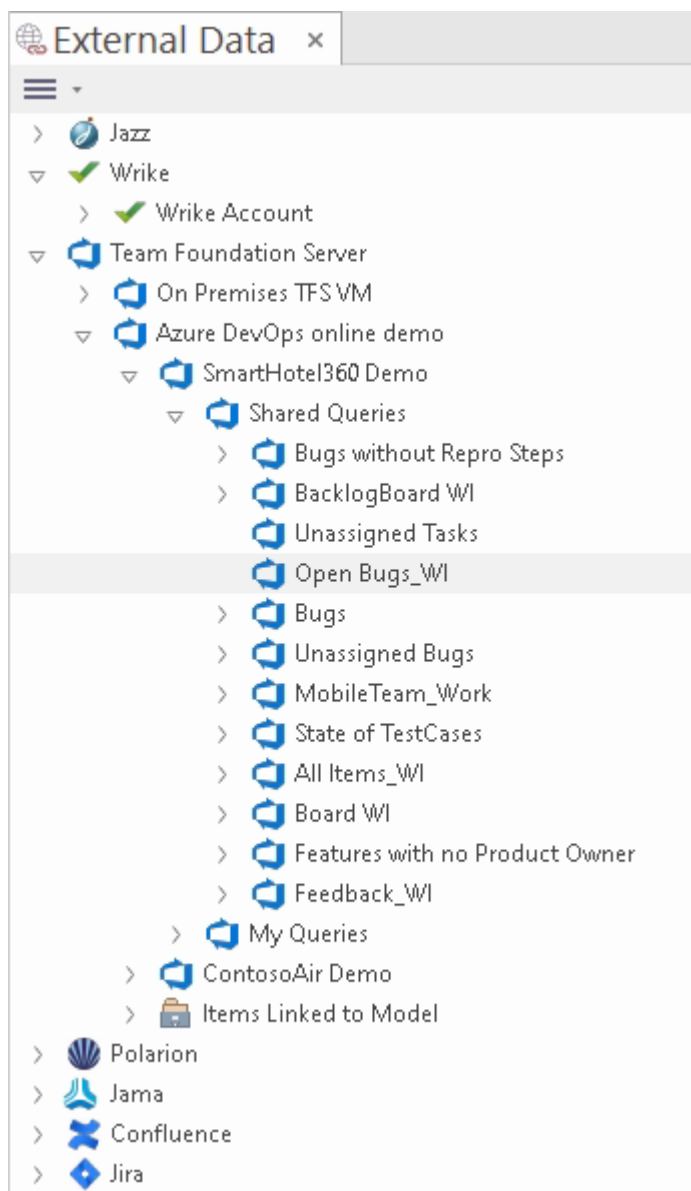
功能区	特定>信息交流>系统集成
-----	--------------

## 浏览层次结构

从☰菜单中选择提供商类型开始导航。

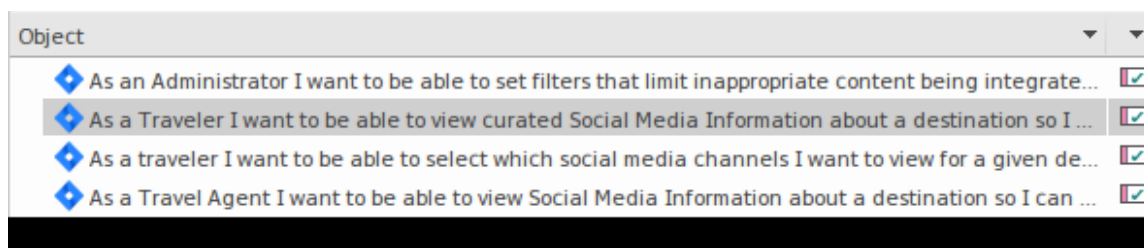


每个提供商提供的导航系统略有不同，因此具体导航取决于您使用的提供商。例如，Dropbox 允许浏览文件夹结构，而 Jira 提供项目列表，后面跟着用户查询列表。



## 物品清单

在每个导航级别，如果可用，对应于导航级别的项目列表将填充左侧面板。



任何链接到外部项的本地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。

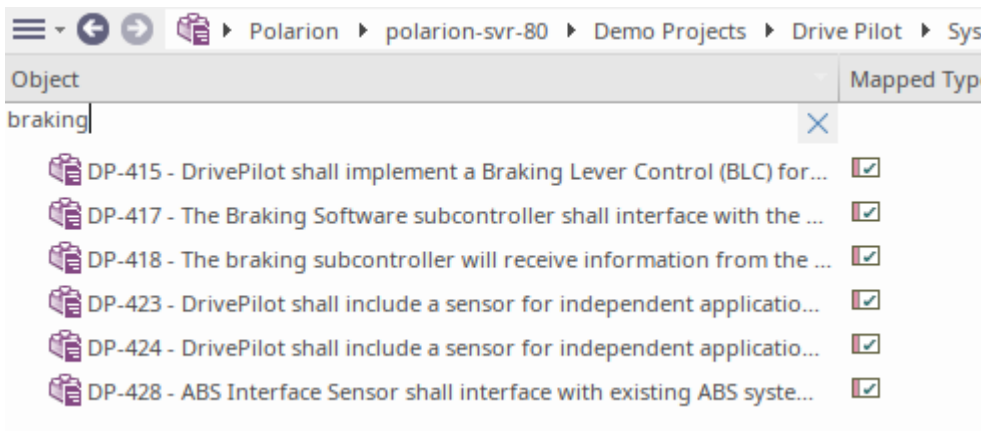
此视图不同于常规列表，因为它在顶部显示本地Enterprise Architect元素，并将外部链接项显示为本地元素的子项。所有相同上下文的菜单项都在视图中可用。

- As a team, we can finish the sprint by clicking the cog icon next to the
  - ◆ As a team, we can finish the sprint by clicking the cog icon next to :
- As a user, I'd like a historical story to show in reports
  - ◆ As a user, I'd like a historical story to show in reports

## 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。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。

## 外部项目详细信息

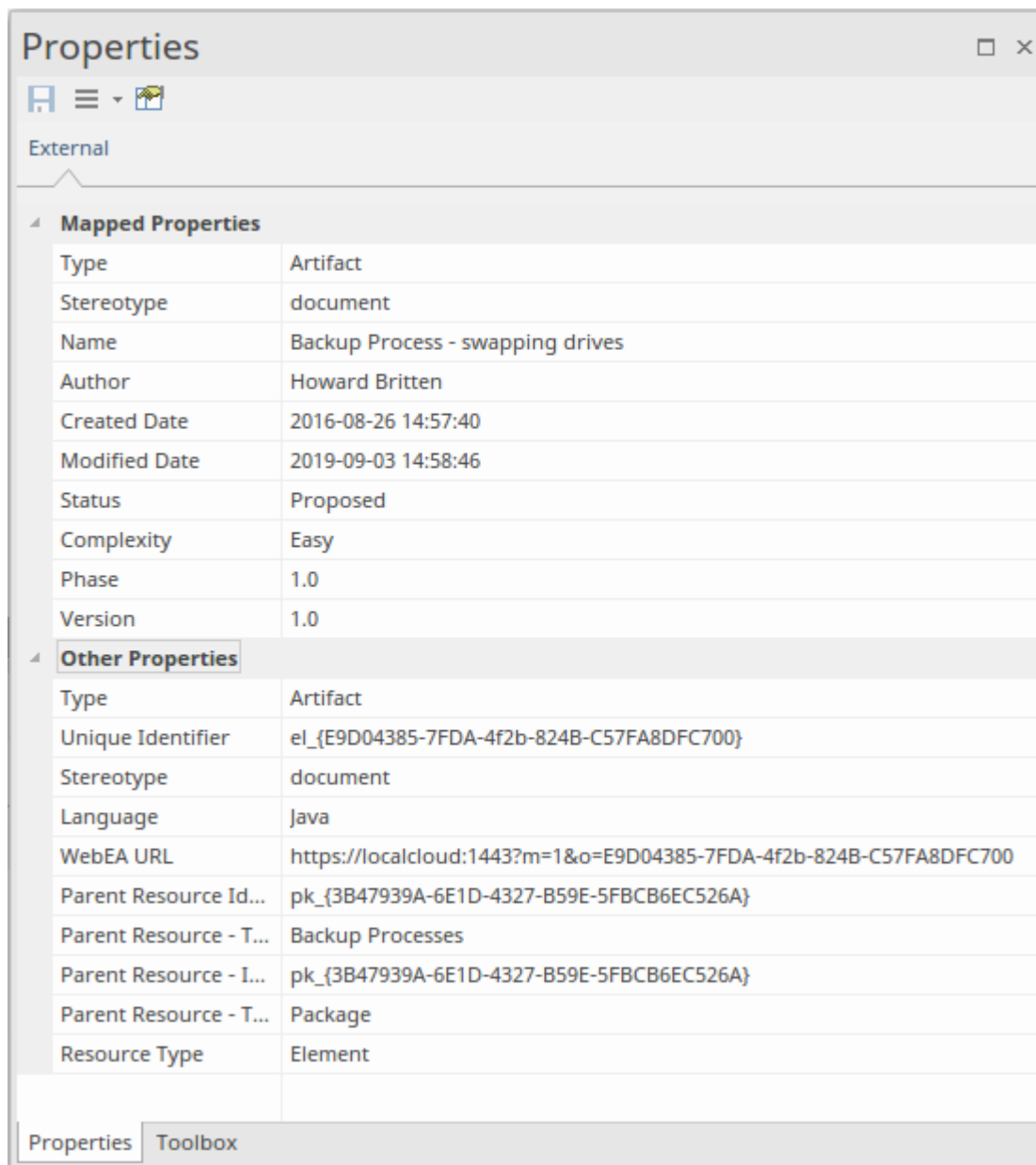
在“外部数据”窗口中选择外部项目时，将检索该项目的详细信息并显示在检验员窗口的“外部”选项卡中，以及相应属性的外部版本、注记和讨论和审阅窗口中。

### 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

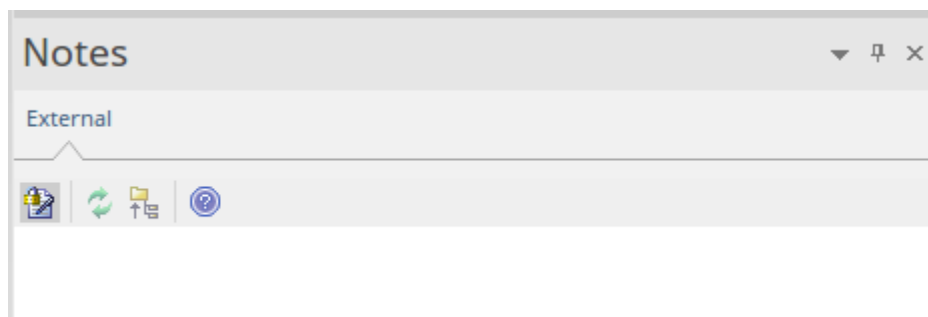
单击列表中的外部项目以在属性窗口的“外部”版本中查看其属性。此窗口显示外部项目的所有可用属性，由外部提供者定义。



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。

## 注记

外部项目的“注记”或“描述”文本显示在注记窗口的“外部”版本中。





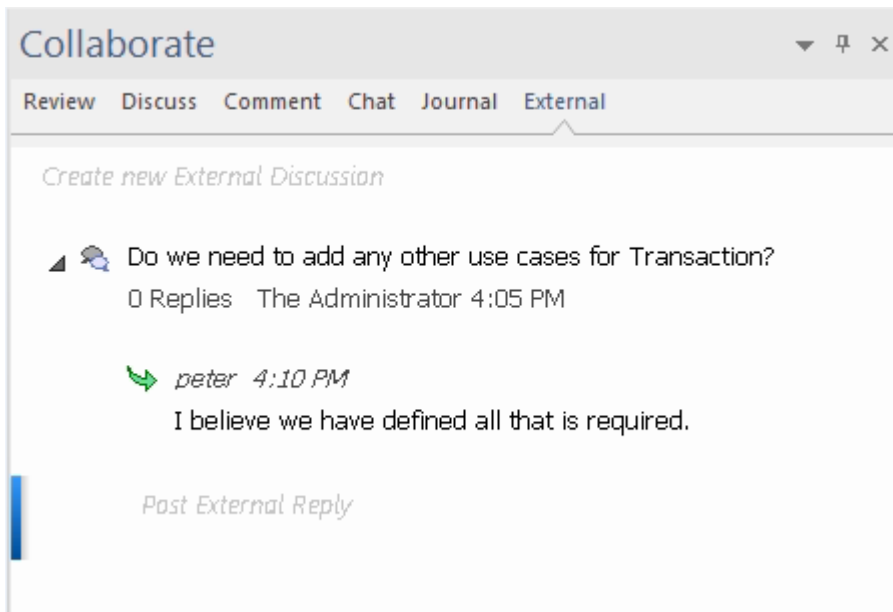
注记并非所有外部提供者都有“注记”或“描述”字段。

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

# 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

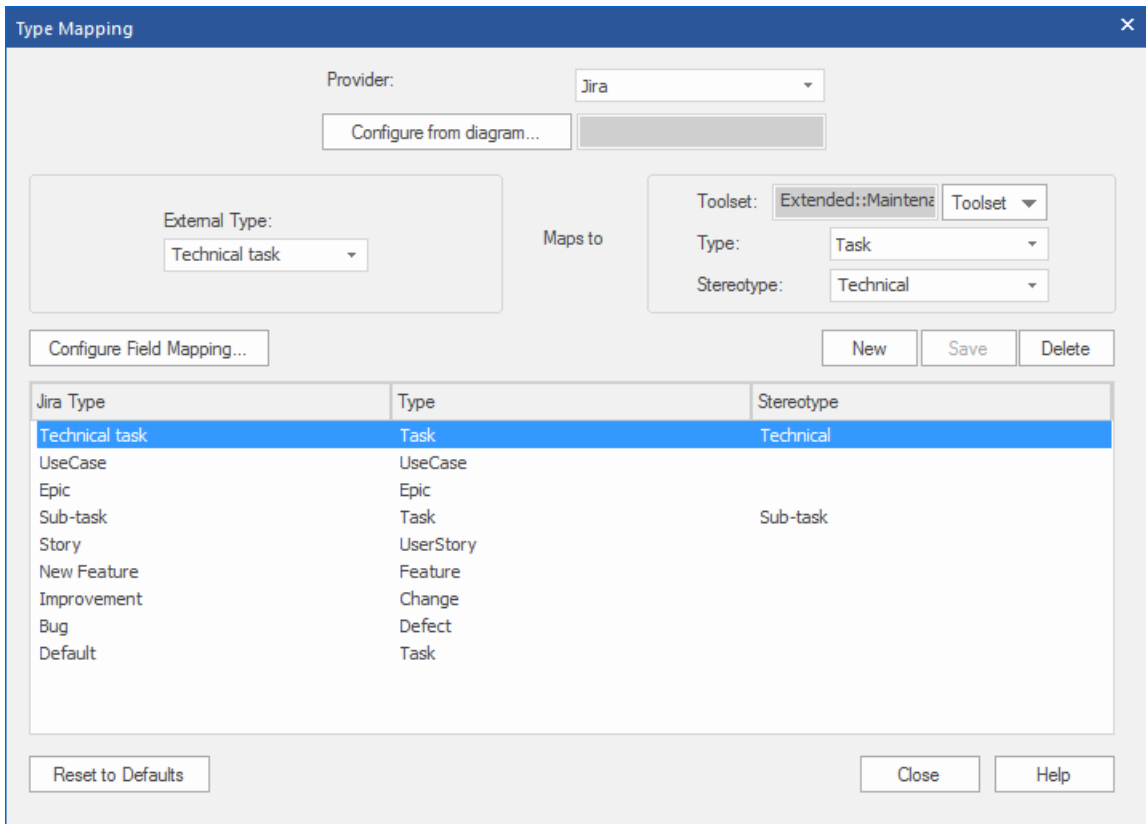
您必须具有 配置外部数据源”权限才能访问此功能。

## 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：

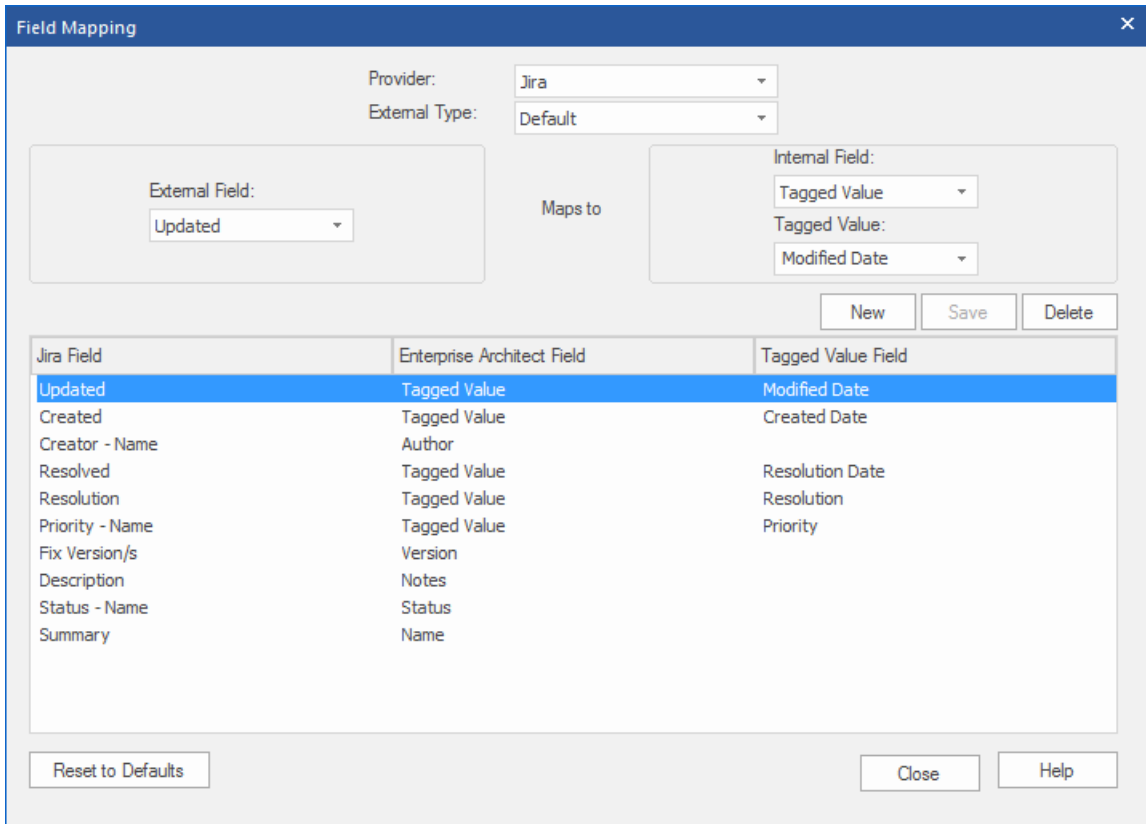
1. 单击新建按钮，然后单击 外部类型”下拉箭头并从外部源中选择要映射的元素类型。
2. 单击 工具集”下拉箭头并选择适当的Enterprise Architect工具集，然后单击 类型”下拉箭头并选择外部元素类型映射到的Enterprise Architect元素类型。

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

无元素字段映射”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）。



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

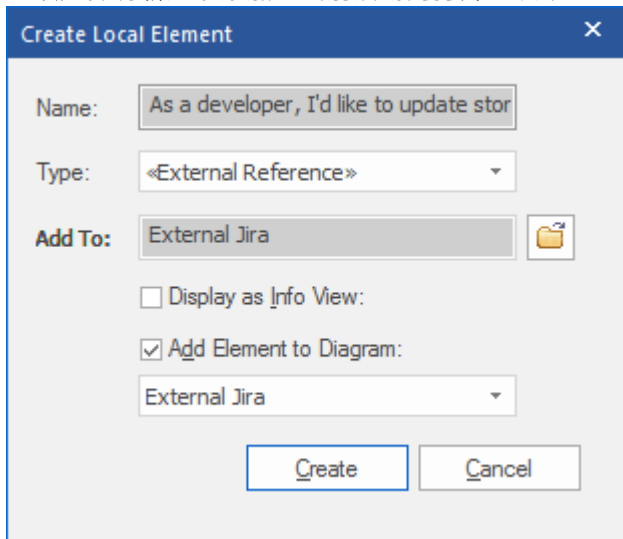


# 链接项

## 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：



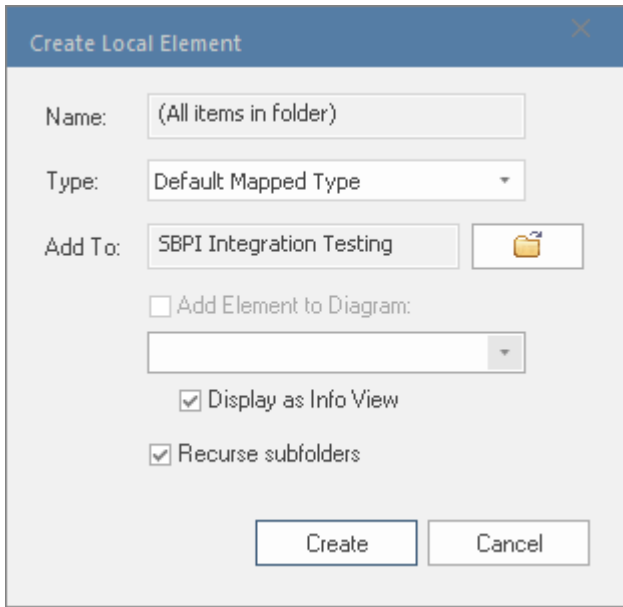
## 链接多个项

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。

## 链接整个文件夹

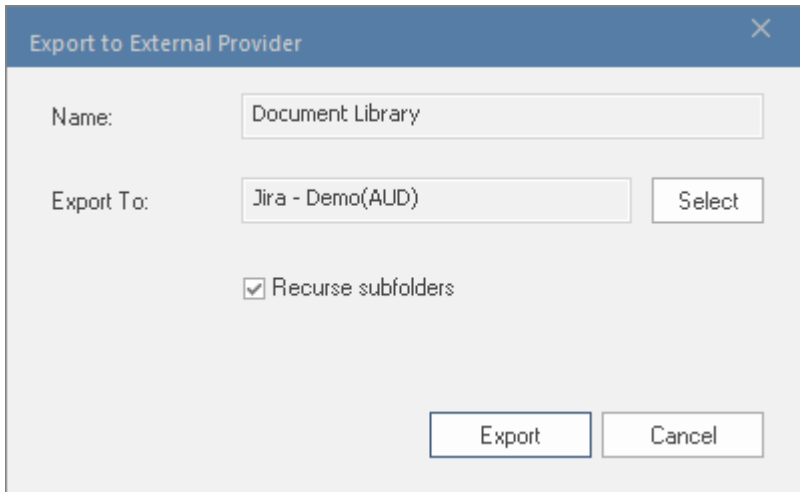
可以同时链接整个树结构层次 - 导入Enterprise Architect同时保持树结构层次。  
右键单击外部树层次结构中的外部文件夹，然后选择“链接整个文件夹”。



### 在外部提供者中创建并链接整个Enterprise Architect包

只需一步即可将整个Enterprise Architect包（包括所有子包和元素）链接并添加到外部提供商。

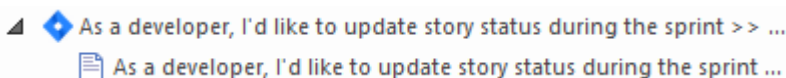
- 在Enterprise Architect浏览器中右键单击一个包，然后选择**专门化 > 导出到外部提供者**



- 选择要导出到哪个外部提供商和文件夹
- 要导出所有子包和子项目，请勾选“递归子文件夹”复选框
- 点击导出

### 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：

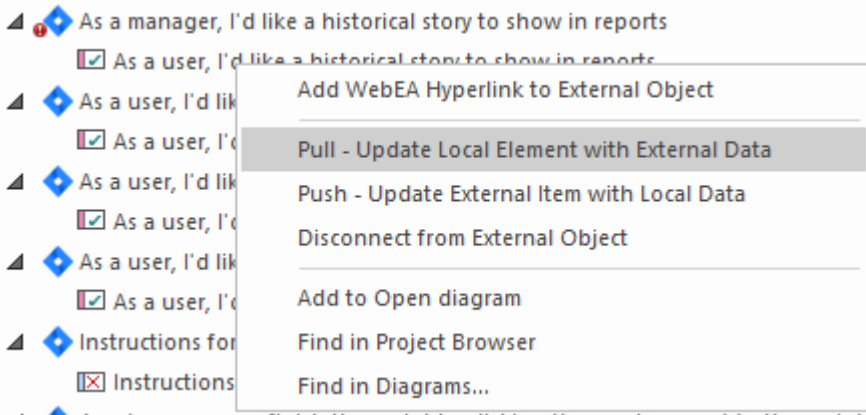
1. 选择本地Enterprise Architect元素（在浏览器窗口中）。
2. 浏览外部提供者并选择所需的目标位置。
3. 在“外部数据”菜单中，选择“创建链接项目”。

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）

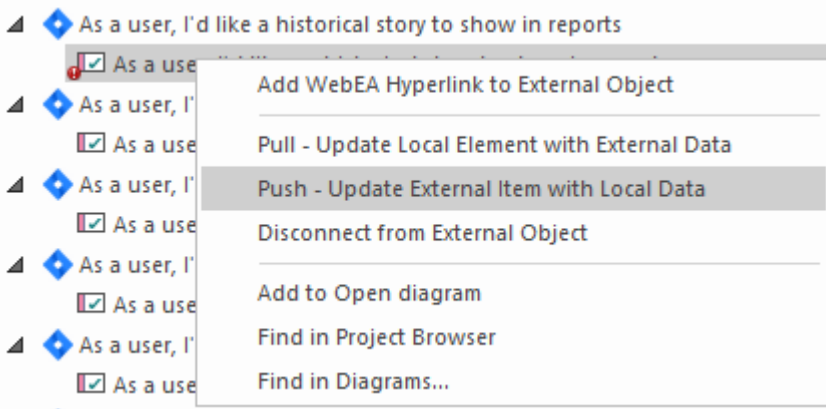
如果在创建或链接项目后本地元素或外部数据项已被修改，则“感叹号”指示符将显示在已更改项目名称旁边的图标顶部。右键单击链接的Enterprise Architect元素并选择“Pull - Update Local Element with External Data”选项。



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“推送 - 使用本地数据更新外部项目”选项。



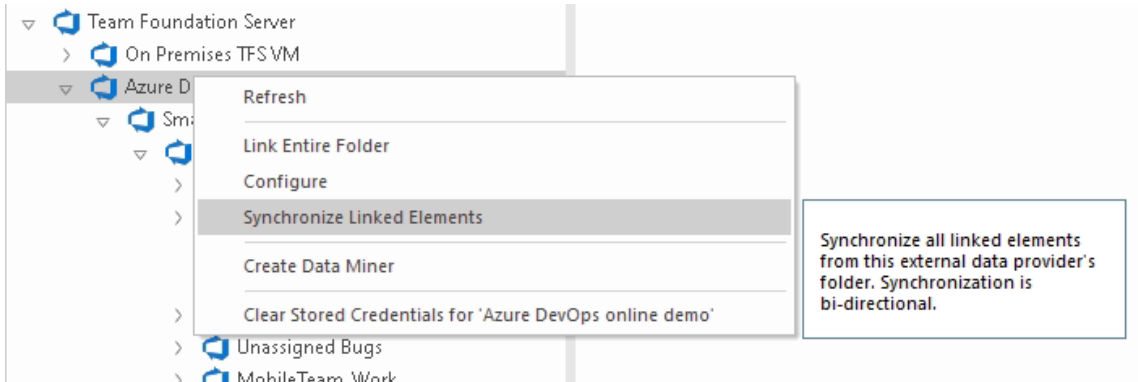
注记：

- 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）

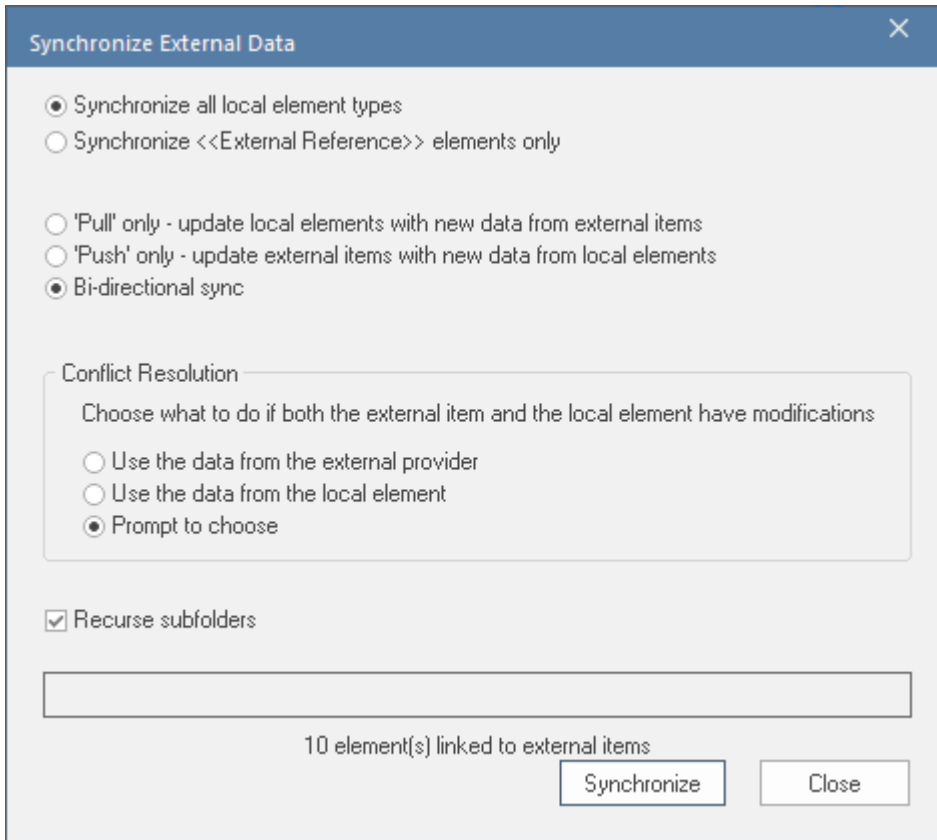
### 同步所有本地元素

Enterprise Architect中的所有本地链接元素都可以使用外部提供者中链接项的新数据进行快速更新。

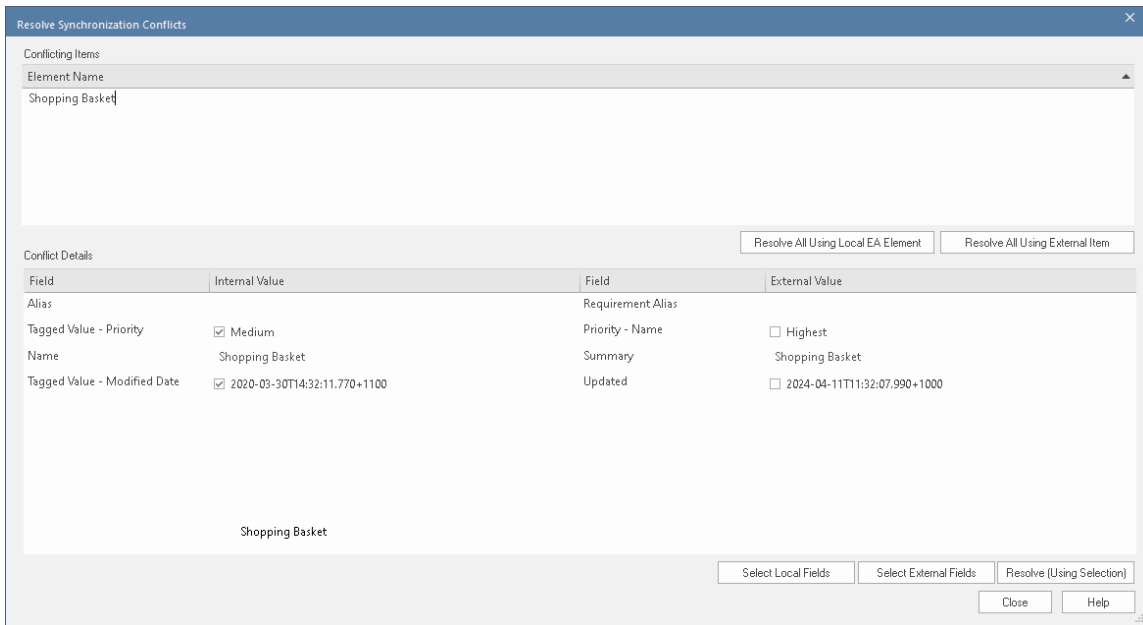
- 在树状层次结构中的外部提供程序上单击鼠标右键，并选择 **同步链接元素**，或者单击工具栏中的 汉堡包 图标，然后选择 **同步本地链接元素** 选项。







- 选择同步所有本地元素或仅同步具有 <<外部引用>> 构造型的元素
- 选择将所有外部更改“拉”入 EA，或将所有更改从 EA “推送”到外部提供商，或者执行“双向”同步，这将同时更新 EA 和外部项目，具体取决于哪个已更新。
- 如果选择“双向同步”，那么在双方都更新的情况下，还有另一种选择：要么让外部提供商覆盖更改，要么让 EA 覆盖外部更改，或者打开冲突解决窗口以允许为任何一方选择单独的字段



- 窗口底部指示可能更新多少个本地元素（注记此时尚不清楚外部项目是否有任何更改）。
- 单击同步按钮。

# 到WebEA的项目超链接

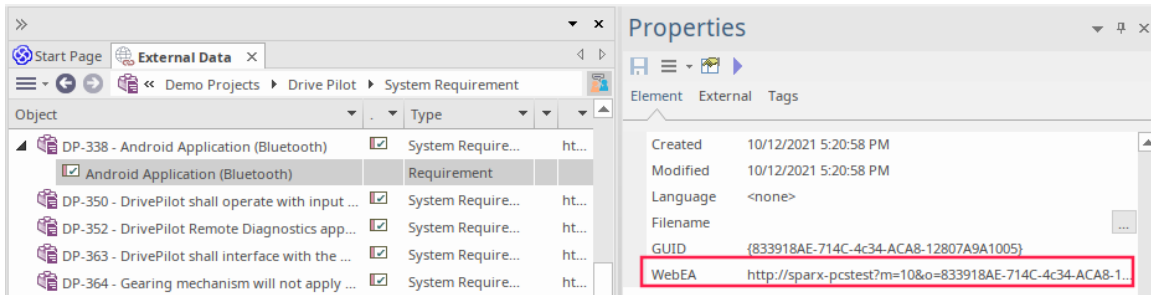
从外部项目创建的元素可以通过超链接从外部应用程序引用到WebEA上托管的元素详细信息。使用WebEA作为参考的好处包括：

- 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

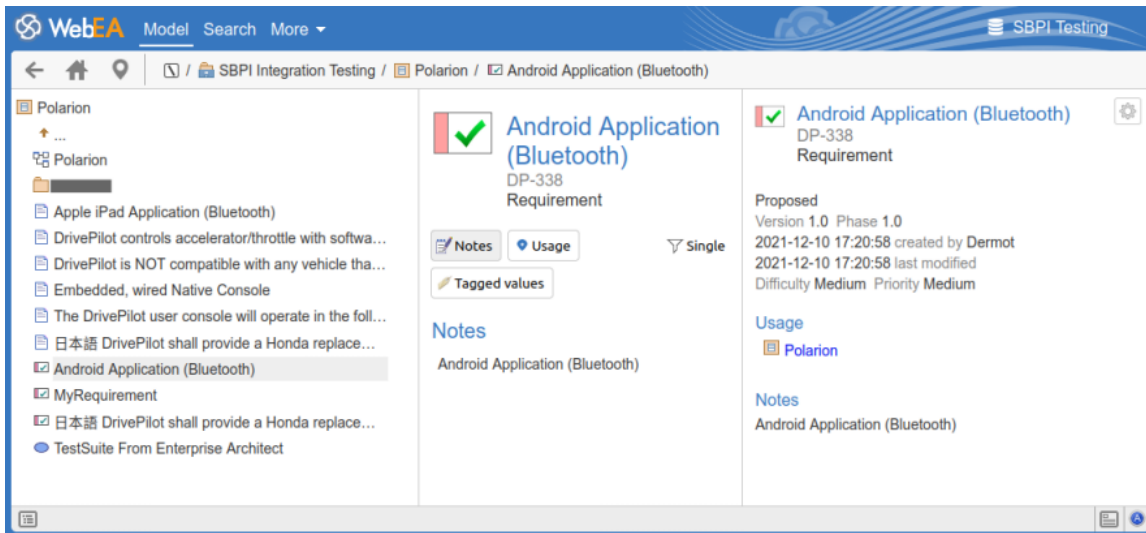
当在Enterprise Architect中创建新元素时，在外部项中设置超链接引用。例如，这里我们有一个超链接，在外部应用程序的一个项目内，指向一个元素的WebEA页面：

Role	URL
<i>external reference</i>	<a href="http://sparx-pcstest?m=10&amp;o=BC32EA51-640F-4457-A933-E57D93952AB0">http://sparx-pcstest?m=10&amp;o=BC32EA51-640F-4457-A933-E57D93952AB0</a>

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 )

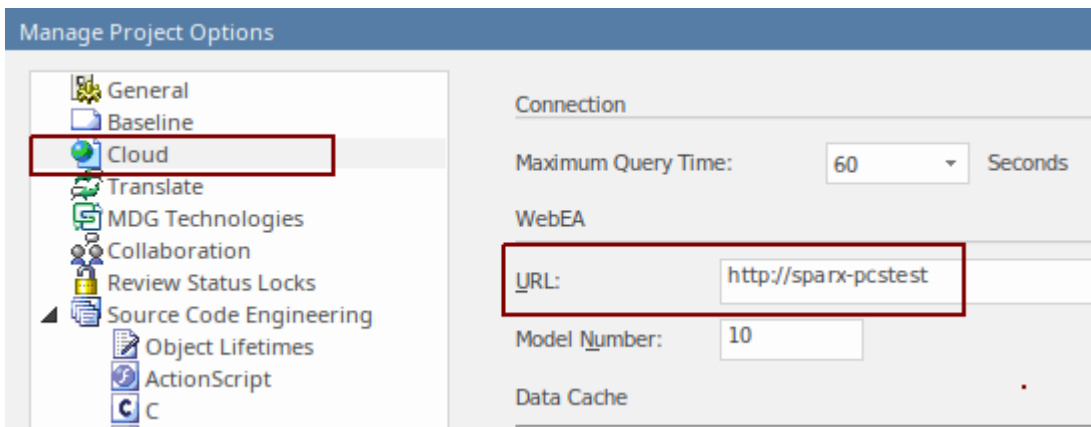
不支持的外部应用程序是：

- 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。

# 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来使用多种编程语言（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 &lt;= 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后，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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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记部分。

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注记有效，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"> <li>• C++：常量字符*</li> <li>• C#：string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。包括 : <ul style="list-style-type: none"> <li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li> <li>• level - 请求菜单的层次结构 ( 从1开始表示第一级 )</li> <li>• currentID, currentName - 当前菜单级别的 id 和名称</li> <li>• levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID</li> <li>• levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name</li> </ul>

## 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, "笔记", " 1项示例笔记");
- 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项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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 参数A JSON string 。包括： <ul style="list-style-type: none"><li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li></ul>

### 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。包括 : <ul style="list-style-type: none"> <li>• level - 请求菜单的层次结构 ( 从1开始表示第一级 )</li> <li>• currentID, currentName - 当前菜单级别的 id 和名称</li> <li>• levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID</li> <li>• levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name</li> </ul>

## 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。包括 : <ul style="list-style-type: none"> <li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li> </ul>

### 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笔记窗口中的“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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。包括 : <ul style="list-style-type: none"> <li>• itemIDs - 以逗号分隔的项目 ID 列表</li> <li>• level - 请求菜单的层次结构 ( 从1开始表示第一级 )</li> <li>• currentID, currentName - 当前菜单级别的 ID 和名称</li> <li>• levelXID - 层次结构中所有先前级别的菜单级别 X 的 ID ; 即level1ID · level2ID</li> <li>• levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name</li> </ul>

## 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, "笔记", " 1项示例笔记");
- 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。包括 : <ul style="list-style-type: none"> <li>• level - 请求菜单的层次结构 ( 从1开始表示第一级 )</li> <li>• currentID, currentName - 当前菜单级别的 ID 和名称</li> <li>• levelXID - 层次结构中所有先前级别的菜单级别 X 的 id ; 即level1ID · level2ID</li> <li>• levelXName - 层次结构中所有先前级别的菜单级别 X 的名称 ; 即 level1Name · level2Name</li> </ul>

## 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 参数A JSON string 。 <ul style="list-style-type: none"><li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li></ul>
----	---

### 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> <p>参数A JSON string 。</p> <ul style="list-style-type: none"> <li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li> <li>• eaElementGUID - Enterprise Architect元素的GUID</li> </ul>
----	--

### 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 参数A JSON string 。 <ul style="list-style-type: none"><li>• itemID - 项目的唯一 ID ( 在 GetItemList 中传回 )</li></ul>
----	---

### 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 参数A JSON string 。可用字段有 : <ul style="list-style-type: none"> <li>• 用户名</li> <li>• 密码</li> <li>• accessToken - OAuth 访问令牌</li> </ul>

## 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

该函数用于指定回调函数数组，用于从SBPI集成插件将数据传回Enterprise Architect。  
仅限 C++。这只需要用 C++ 实现。示例中的代码足够了，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
const void ** 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

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。

## 输入

参数	细节
参数	<ul style="list-style-type: none"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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();
}
```

# 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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"> <li>• C++ : 常量字符*</li> <li>• C# : string</li> </ul> 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

Parameter	Details
<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

This method receives the details that the user inputs into the Pro Cloud Server configuration when enabling this Custom Service Plug-in.

## 输入

Parameter	Details
参数	<ul style="list-style-type: none"><li>• C++ : 常量字符*</li><li>• C# : string</li></ul> 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 ( ) ;
```



