EA Protocol handler

Disclaimer

1.1 Introduction

The EA protocol handler enables access to arbitrary elements, packages and diagrams in a Sparx Systems Enterprise Architect model repository via a simple URL.
Potential applications for this access method include:

- Reference items from requirements tools to enable traceability
- Exchange model information with stakeholders
- Link model elements and diagrams from corporate web sites, intranets or Wiki installations
- Easily deploy a shared (database) repository to users without the need to guide them through wizards and/or connection strings

1.1.1 System Requirements

- Windows2000/XP/Vista
- Sparx Systems Enterprise Architect®[2] version 7.1 or higher
- Microsoft .NET Framework 2.0

Note: The protocol handler and addin have been tested with Microsoft .NET framework 2.0, however it might run with higher versions, too.

1.2 Installation/Deinstallation

Installation is done via the included installer program. It automatically installs the software in the EA root directory (eg. C:\Program Files\Sparx Systems\EA\CanonicPlugins) and sets the required registry entries.
Deinstallation is accomplished by running the installer again or by running the removal program from the installation directory.
Depending on the provider of the installer a basic set of preconfigured connection strings might already be provided.

1.3 Basic usage

1.3.1 Using the ea:// protocol

EA protocol handler accesses diagrams, packages and elements by following a URL from your web browser, text processor, file explorer, etc.

The software installs a new protocol handler in the operating system thus enabling the use of a ea:// URL. The common URL syntax is as follows:

```
ea://default|<model>?[c=<connectstring>]&d=<diagram guid>|e=<element guid>|p=<package guid>
```

Each reference starts with the ea:// protocol name followed by a model name. Model names can be preconfigured ones if desired (see configuration below).
If a connection string is given with the c="…" command then this connection string is used to connect to the EA repository. This connection string is encrypted and base64 encoded.
Otherwise the connection string configured as `default` or as `<model>` is considered.

The model element reference is given as guid in either short or expanded form:

```
{8759-F3EF2FB6-4b30AFA0-3025C17E-77DB} is equivalent to 8759F3EF2FB64b30AFA03025C17E77DB
```

[2] Enterprise Architect is a trademark of Sparx Systems Pty Ltd. (http://www.sparxsystems.com)

Accessing a diagram (d=) leads to selecting it in the element browser and showing it.
Packages and elements are given with the p=… respectively e=… parameter and are not opened but selected in the project browser. There can always be only one model element or diagram referenced.

Examples:

```
ea://model1?c=%2fgH%2bjil1ZOsHDun78LPBQlRSjB%2fJGN%2fc9RFs6xEVfsqnUqntn5aJYBDaebMdyumonnRjbN
K%2fc9g%2bC0VtuW87WQ%3d%3d&e=8759F3EF2FB64b30AFA03025C17E77DB
```

Opens an element with guid `{8759-F3EF2FB6-4b30AFA0-3025C17-E77DB}` in the given repository.

```
ea://model1?&e=8759F3EF2FB64b30AFA03025C17E77DB
```

Opens an element with guid `{8759-F3EF2FB6-4b30AFA0-3025C17-E77DB}` in the model repository configured as model1.

```
ea://default?&e=8759F3EF2FB64b30AFA03025C17E77DB
```

Opens an element with guid `{8759-F3EF2FB6-4b30AFA0-3025C17-E77DB}` in the configured default model repository.

1.3.2 Acquiring links to EA objects- the EA Add-In

The purpose of the Add-In is to provide a non intrusive facility to generate valid EA Protocol Handler URLs. As per the original design intention of the EA Protocol Handler, the Add-In will generate URLs for the following artefacts:

- Diagrams
- Packages
- Elements

The Add-In is separate to the EA Protocol Handler insomuch as they need not be installed at the same time. Installation and deinstallation of the add-in is performed by calling the included installation program.

The Add-In facilitates creation of the URLs whereas the EA Protocol Handler actions upon them.

As per all EA Add-Ins, the EA Protocol Handler Add-In provides a menu accessible from the Add-Ins menu. This provides the root for all menu items of the Add-In.

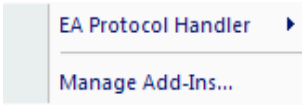

Figure 1: EA Add-Ins Menu

Expanding the main menu will present a pop-up with 5 items. All menu items support keyboard accelerators to aid keyboard users as best as possible.

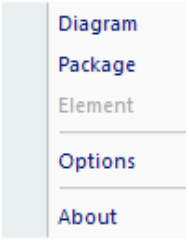

Figure 2: Menu items are enabled/disabled based on active selection

The availability of the top 3 menu items is controlled by the active selection within EA. The following rules are applied:
Diagram is enabled (processed in order of precedence):

- Whenever a Diagram is open in the Workspace.
- Whenever the selected item in the Project Browser is a Diagram.

Package should always be enabled:

- Whenever there is an item selected in the Project Browser. The item need not be a Package type – the Add-In will use the object's hierarchy to find the Package type the selected item belongs to.

Element is enabled:

- Whenever there is an item selected in the Project Browser that is a UML Element.

Whenever any of the above menu items are selected the Add-In will generate a EA Protocol Handler URL and place it on the clipboard.
The remaining two options are always enabled.
Additionally, the Add-In menu is presented in the standard EA context menu. In this scenario the only menu items available are those pertinent to the function of the Add-In.

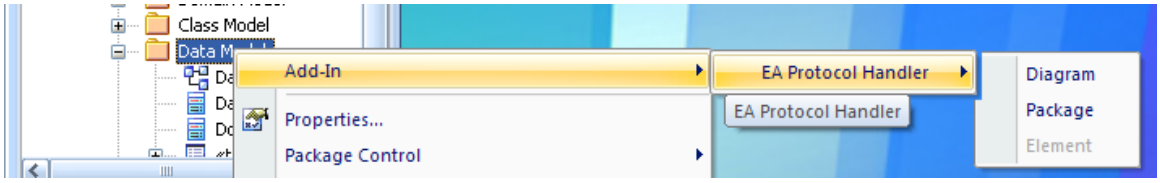

Figure 3: Context Menu from Project Browser

1.4 Configuration

1.4.1 EA protocol handler

The software comes with a configuration file in the Enterprise Architect installation folder.
It allows to preconfigure connection strings for accessing different repositories as well as defining a default connector.
The format is the following:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="default" value="model1"/>
  </appSettings>
  <connectionStrings>
    <add name="model1"
connectionString="DBType=1;Connect=Provider=SQLOLEDB.1;Password=xxx;Persi
st Security Info=True;User ID=yyyyy;Initial Catalog=model1;Data
Source=myServer;"/>
    <add name="vlmodelWAN" connectionString="
connectionString="DBType=1;Connect=Provider=SQLOLEDB.1;Password=xxx;Persi
st Security Info=True;User ID=yyyyy;Initial Catalog=model1;Data
Source=myServer;Lazy=1"/>
  </connectionStrings>
</configuration>
```

In the connectionStrings section several model repositories can be defined and assigned a shortcut name. This name is then referenced in the protocol URL string as model source (eg. `ea://model1?.....`)
The default value in the appSettings section defines a default model which is referenced if the URL is called with the “default” model name (eg. `ea://default?...`). Thus the user can decide which connection method is used under various scenarios (eg. working in the office or via VPN).
Note that configure connection strings are not base64 encoded in contrast to directly referencing them in the URL connection string (`ea://model?c=…`).

1.4.2 EA Add-In

The options dialog controls the per user settings for the Add-In. The following diagram shows the default configuration for the Add-In.

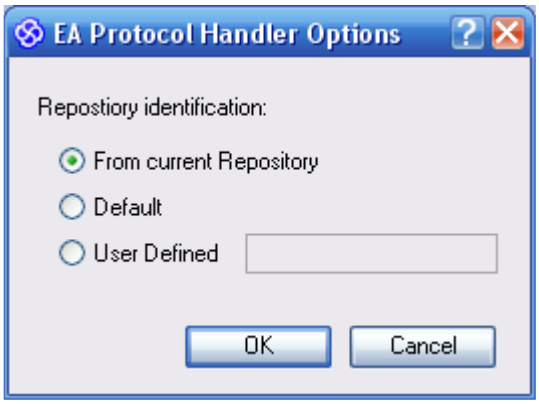

Figure 4: Options dialog

The EA Protocol Handler supports URLs with 3 different formats. Each format differs by what data is passed.

From current Repository
Defines a URL structure that includes the Repository name (as known by EA), a connection string (that's encrypted) and the id of the artefact. This option is recommended when a repository's connection is using integrated security.

Default
Defines a URL structure that includes the id of the artefact within a 'default' Repository as defined locally to the EA Protocol Handler. This option is recommended when there is a single repository.

User Defined
Defines a URL structure that includes the id of the artefact within a Repository identified by the provided 'key' value. The EA Protocol Handler will manage these 'keys' locally.

OK will write the user preferences to Isolated Storage which is an area on disk where .NET stores application data. When the Add-In is next reloaded (as part of the EA process), previous preferences will be retrieved and applied.
Contextual help is provided using the 'What Is' help button on the window title bar.