

EXTENDING EA WITH CUSTOM SCRIPTS TO CATER FOR SPECIFIC NEEDS AND USER REQUESTS

LONDON EAUG 2018



Guillaume FINANCE

 Blog: www.umlchannel.com

 Add-in: www.eautils.com

 Twitter: @umlchannel



CONTENT

01. INTRODUCTION

02. EA SCRIPTS OVERVIEW

03. SCRIPTS HELPERS

04. EA-MATIC TRIGGERED SCRIPTS

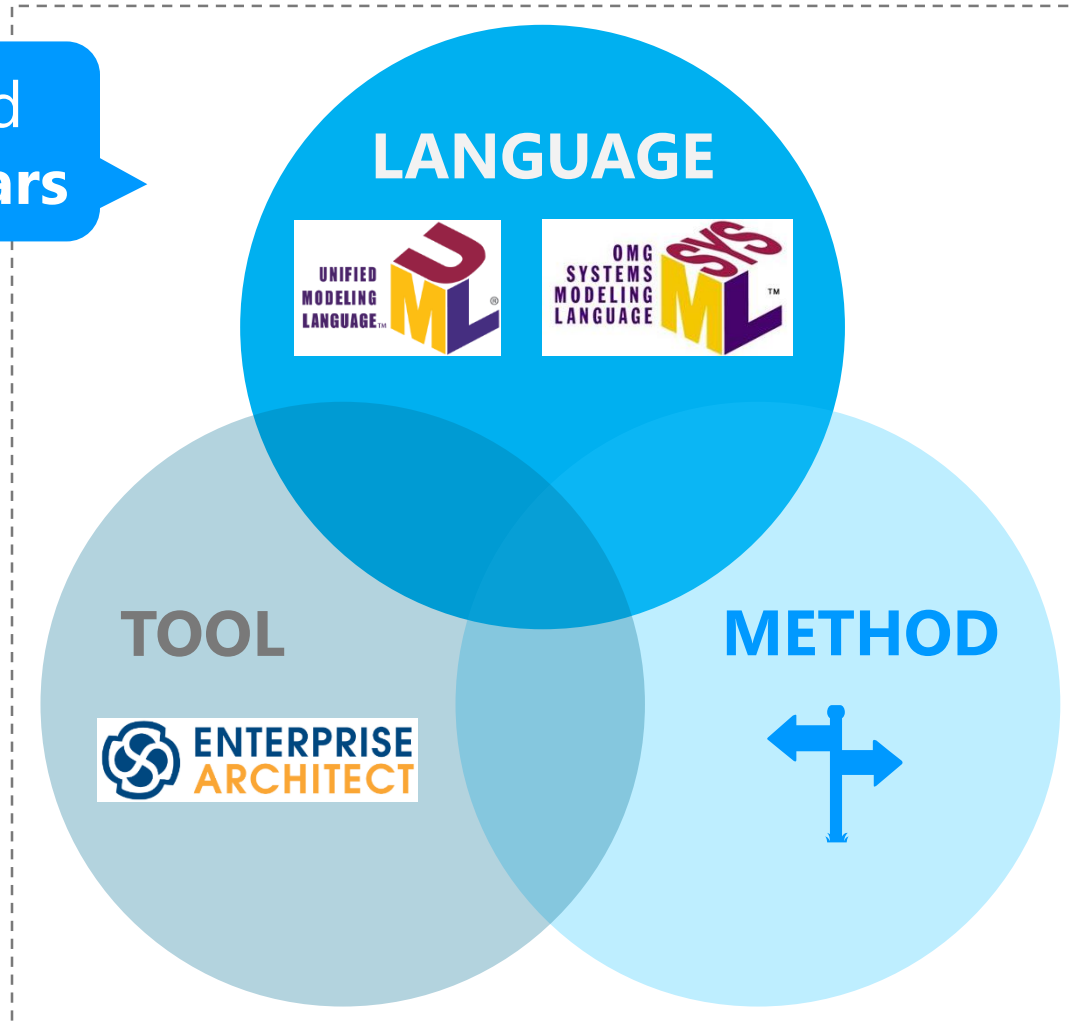
05. DEBUG & HINTS

06. DEMONSTRATION

INTRODUCTION

EA tool customization

Model-based
approach **pillars**

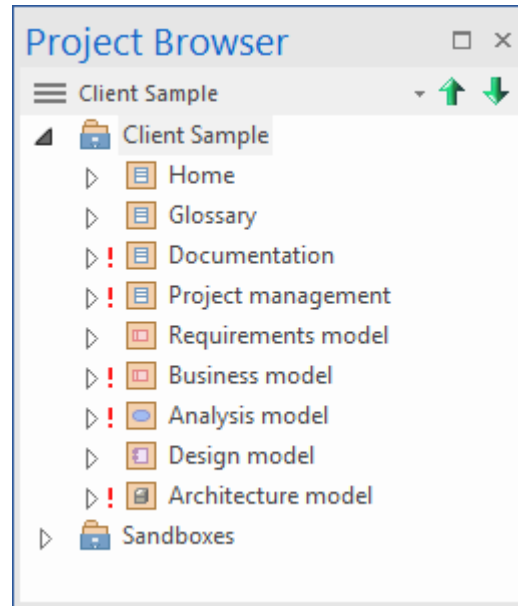


Tool customization

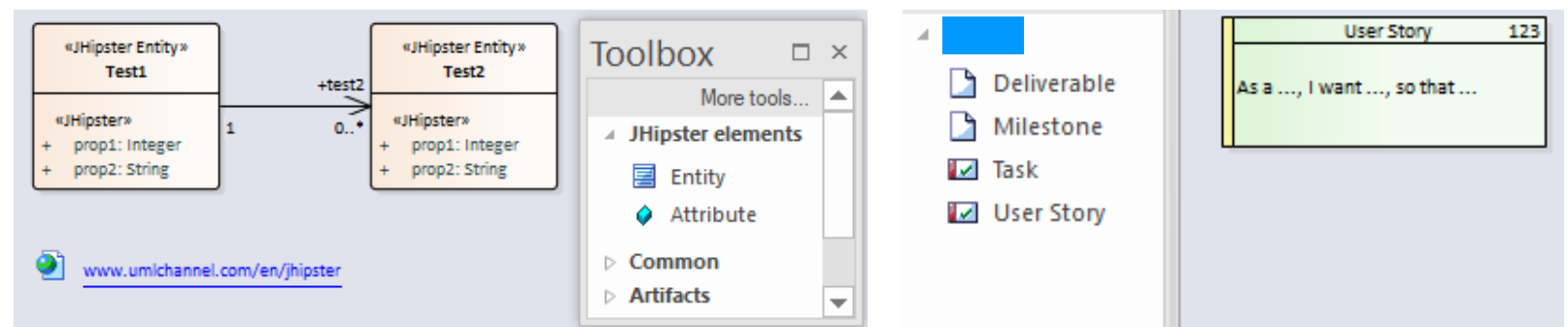
INTRODUCTION

EA tool customization

Custom project structure



Stereotypes/UML Profiles, toolbars



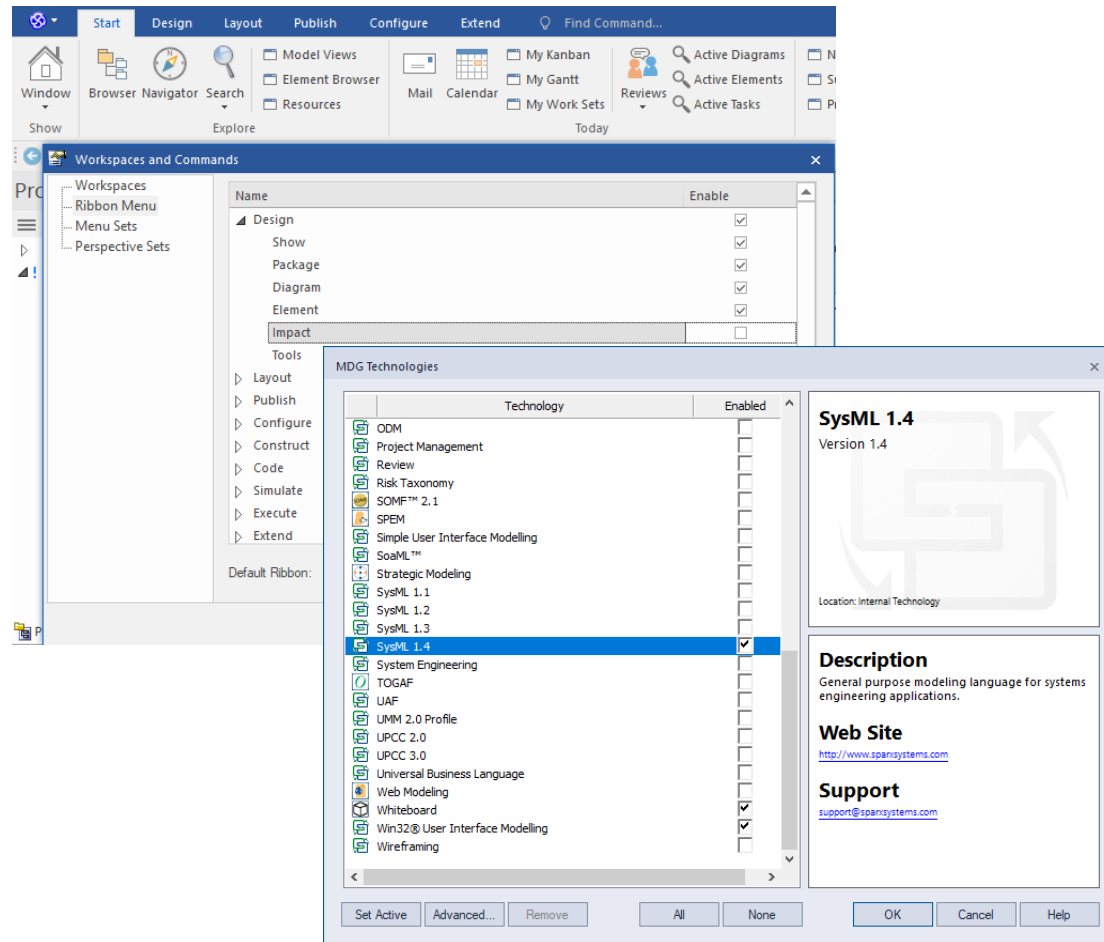
Custom searches

123	Find User Story by ID	123
Drag a column header here to group by that column.		
UserStory_ID	UserStory_Name	
123	As a ..., I want ..., so that ...	

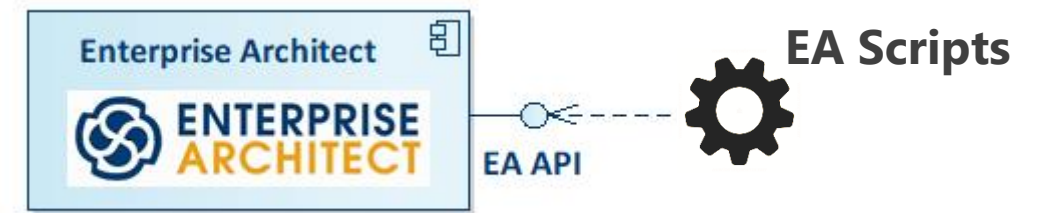
INTRODUCTION

EA tool customization

Custom EA User Interface (13.5)



New features using EA API...



INTRODUCTION

Why do we need scripts?

 Replace repetitive tasks performed by users 

 Work with Excel 

 Custom Excel publication

 Update details in Excel

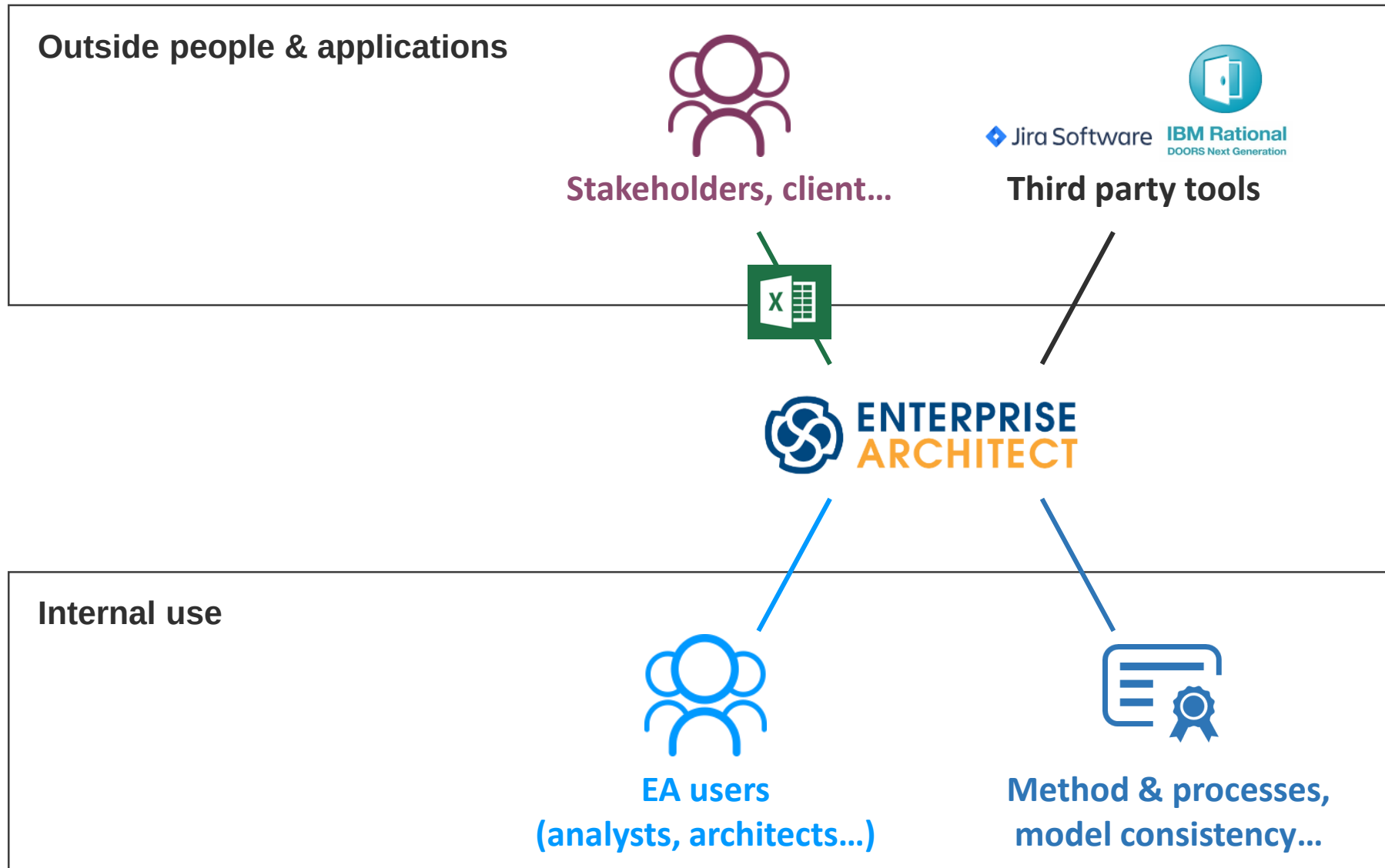
 Custom tools and solutions integration 

 Model QA 

 And much more!

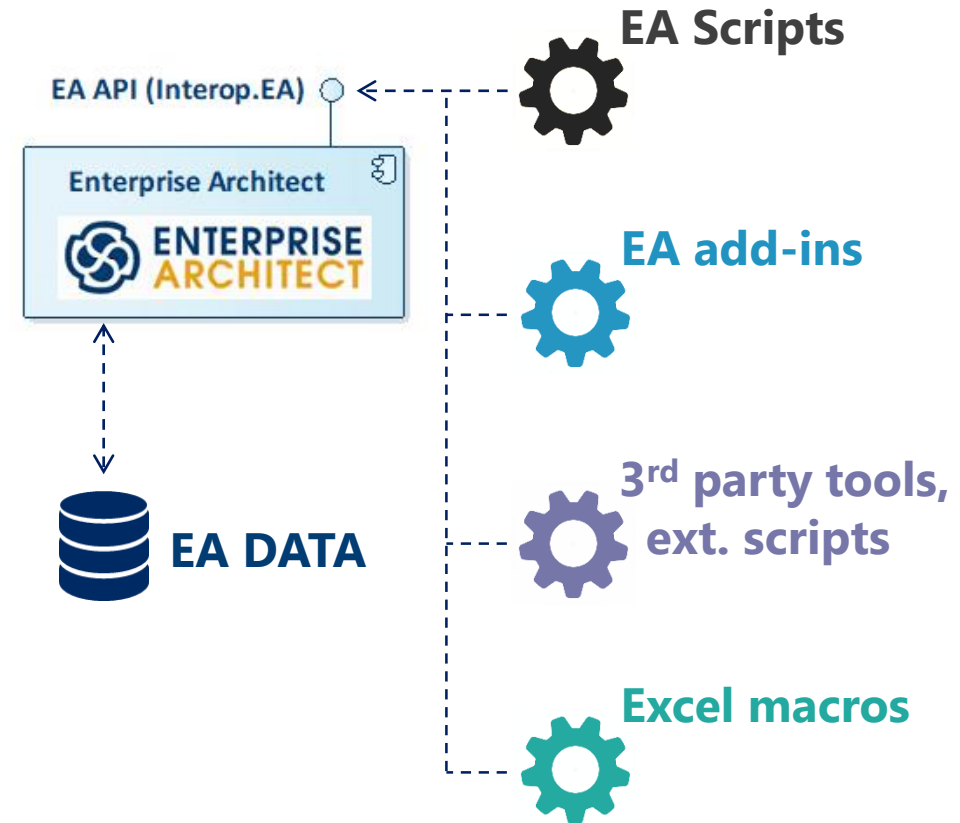
INTRODUCTION

Why do we need scripts?



INTRODUCTION

Alternatives in using EA API



Installation

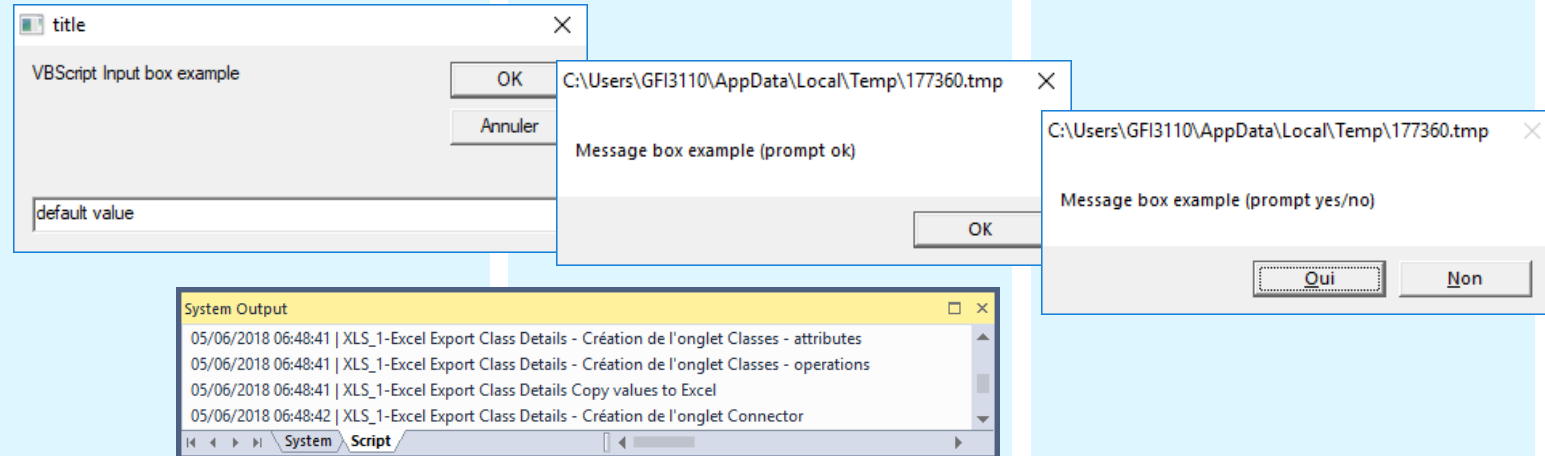
➤ None - import file in the project, available for all users

User Interface

➤ Message or Input box, system output...

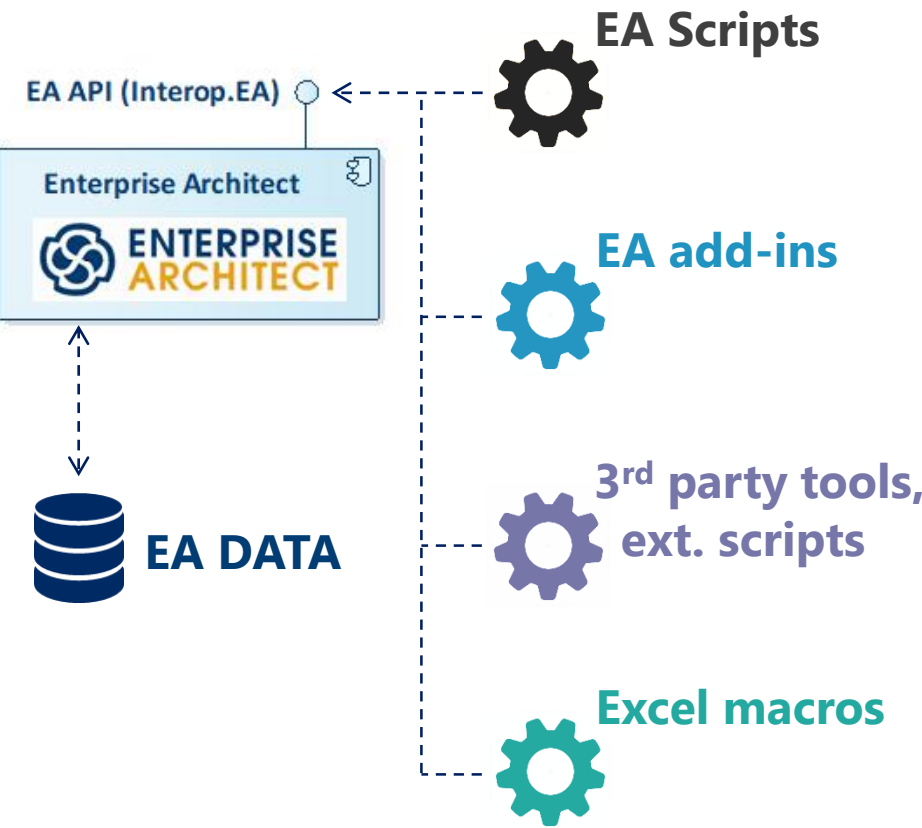
Language(s)

➤ VBScript, JavaScript, JScript



INTRODUCTION

Alternatives in using EA API



Installation

- None - import file in the project, available for all users
- Installation on all clients, available to use with any project

User Interface

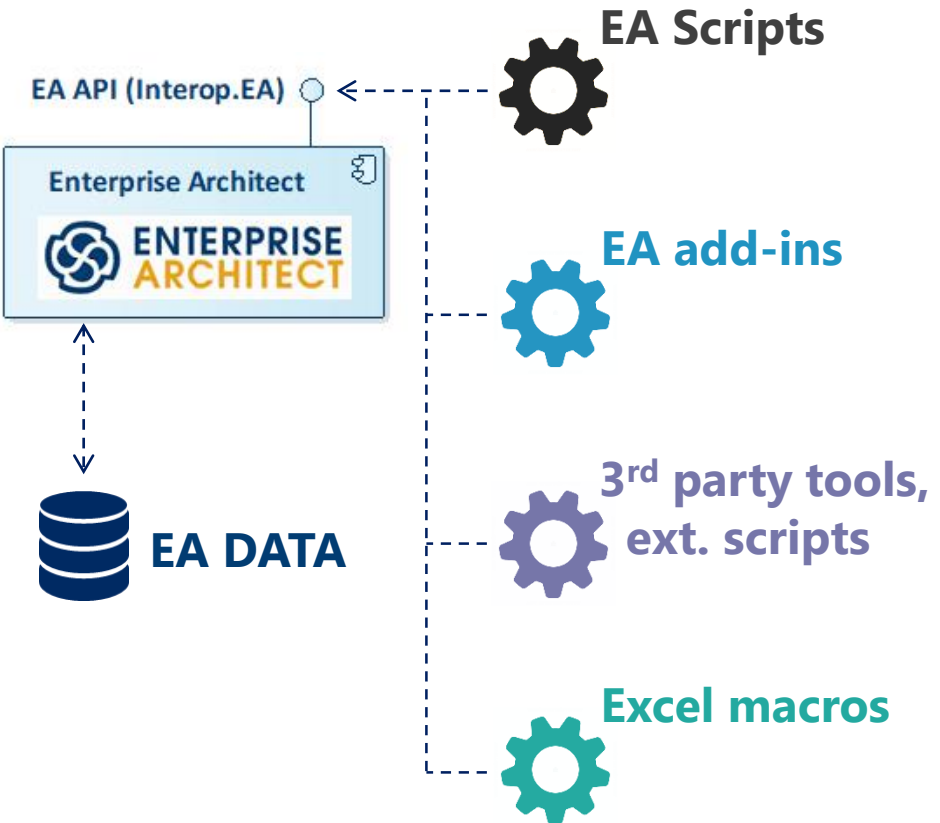
- Message or Input box, system output...
- Enhanced windows, system output...

Language(s)

- VBScript, JavaScript, JScript
- C#, VB.Net, Java, C++...

INTRODUCTION

Alternatives in using EA API



Installation

- None - import file in the project, available for all users
- Installation on all clients, available to use with any project
- Optional

User Interface

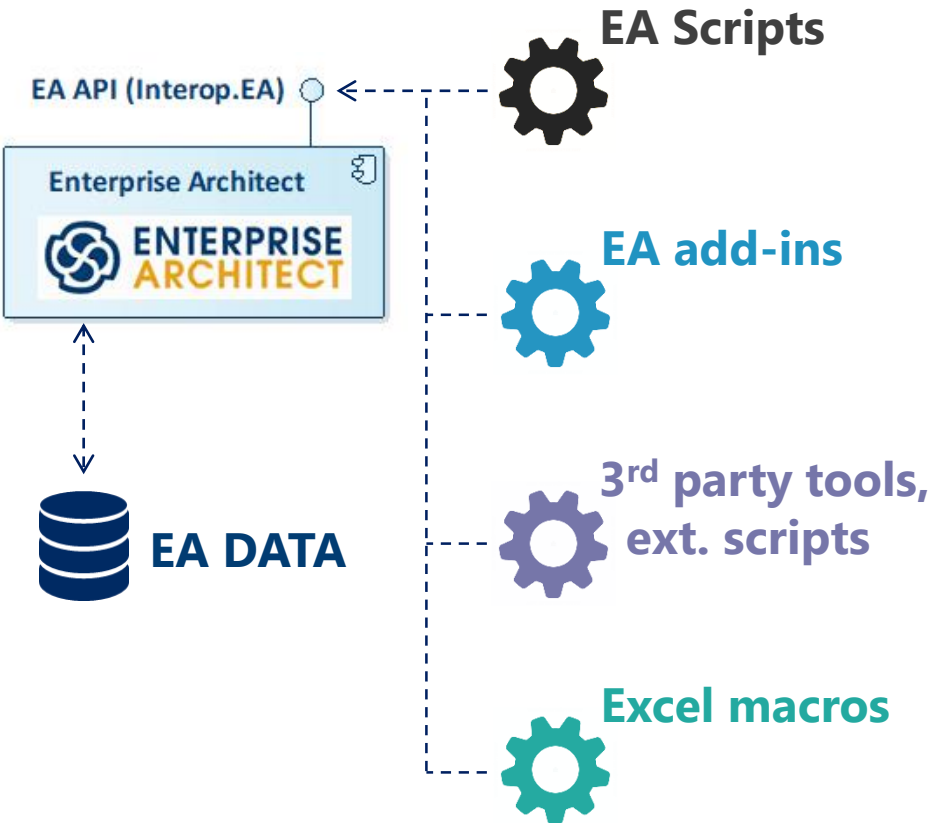
- Message or Input box, system output...
- Enhanced windows, system output...
- Linked with the chosen language

Language(s)

- VBScript, JavaScript, JScript
- C#, VB.Net, Java, C++...
- C#, VB.Net, Java, C++, Python, Perl...

INTRODUCTION

Alternatives in using EA API



Installation

- None - import file in the project, available for all users
- Installation on all clients, available to use with any project
- Optional
- n/a (Excel required)

User Interface

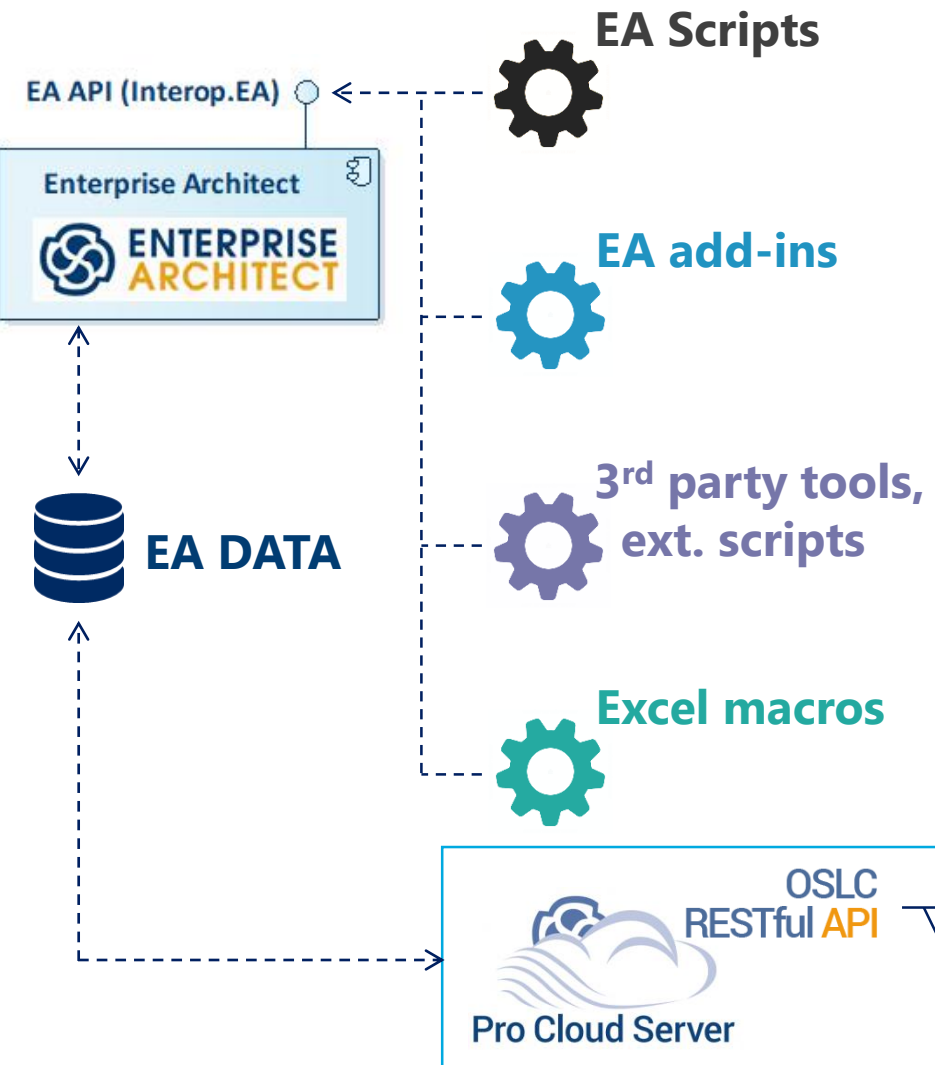
- Message or Input box, system output...
- Enhanced windows, system output...
- Linked with the chosen language
- Excel

Language(s)

- VBScript, JavaScript, JScript
- C#, VB.Net, Java, C++...
- C#, VB.Net, Java, C++, Python, Perl...
- Excel VBA

INTRODUCTION

Alternatives in using EA API

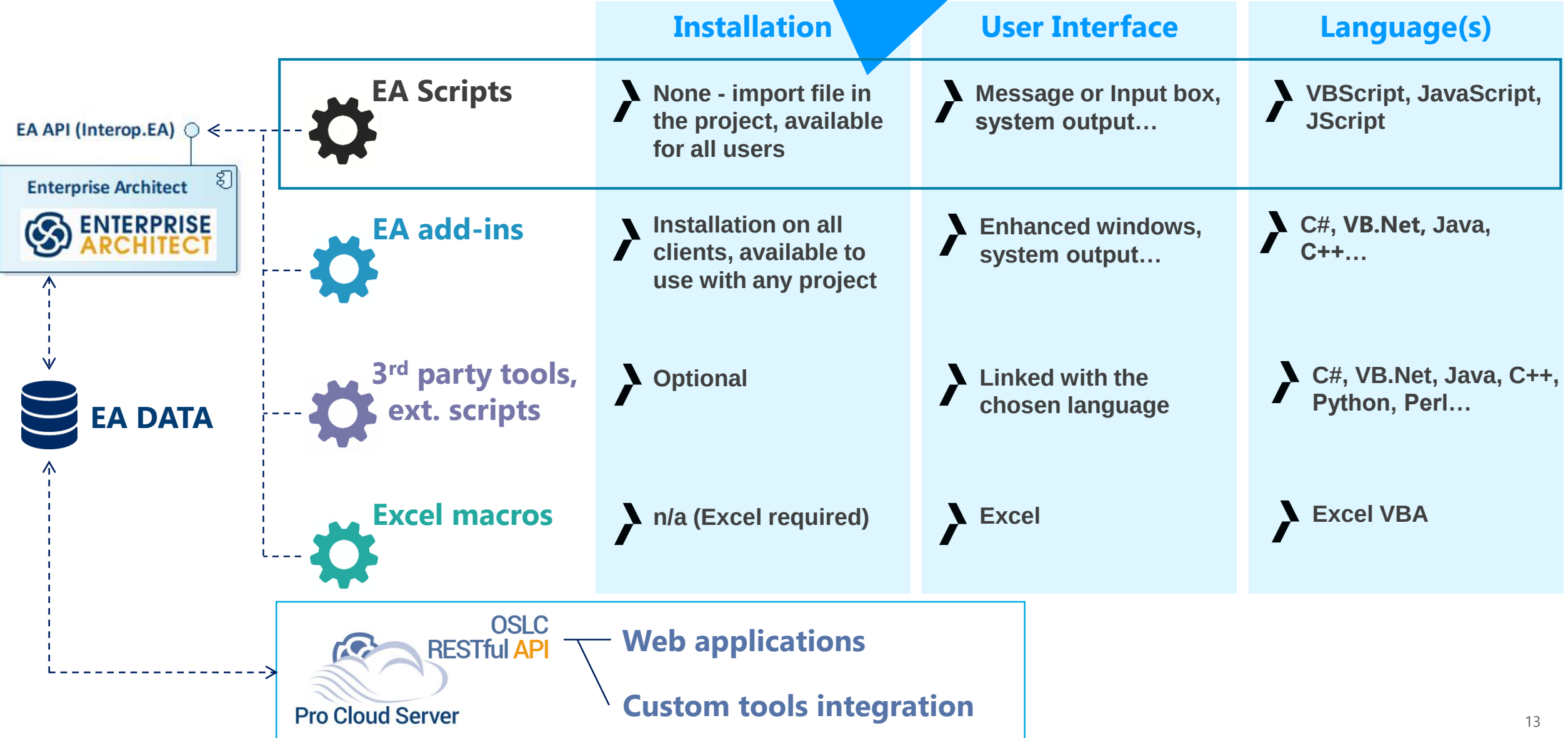


Installation	User Interface	Language(s)
None - import file in the project, available for all users	Message or Input box, system output...	VBScript, JavaScript, JScript
Installation on all clients, available to use with any project	Enhanced windows, system output...	C#, VB.Net, Java, C++...
Optional	Linked with the chosen language	C#, VB.Net, Java, C++, Python, Perl...
n/a (Excel required)	Excel	Excel VBA

INTRODUCTION

Alternatives in using EA API

Compatible with frequent releases for a large group of users



CONTENT

01. INTRODUCTION

02. EA SCRIPTS OVERVIEW

03. SCRIPTS HELPERS

04. EA-MATIC TRIGGERED SCRIPTS

05. DEBUG & HINTS

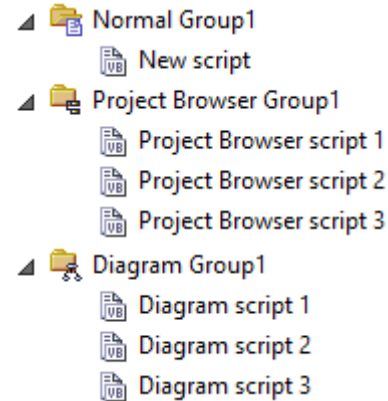
06. DEMONSTRATION

EA SCRIPTS OVERVIEW

Script Group Types

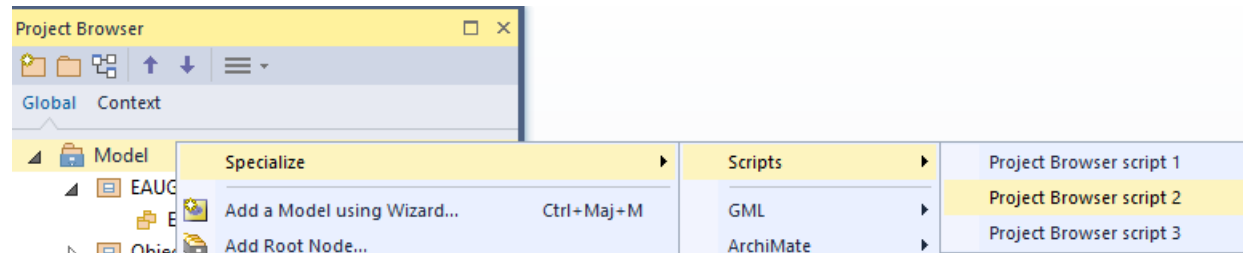
A script is defined under a group

Main group types

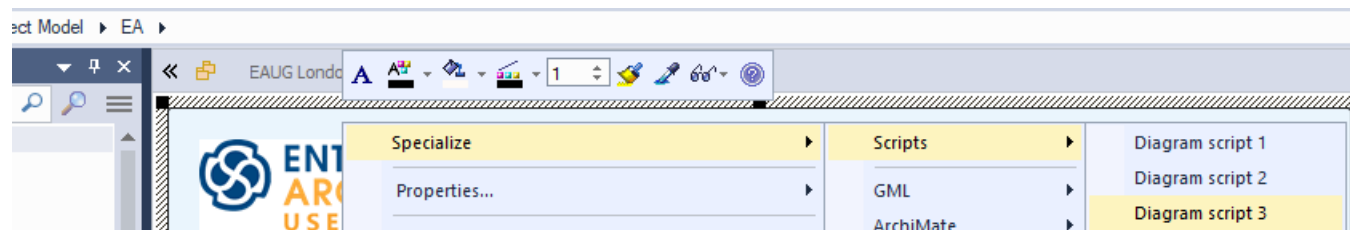


Scripts under a **Normal Group** are available from the Scripting view

Scripts under a **Project Browser Group** are available from the Scripting view or via a **right click in the Project Browser**



Scripts under a **Diagram Group** are available from the Scripting view or via a **right click in a diagram**



EA SCRIPTS OVERVIEW

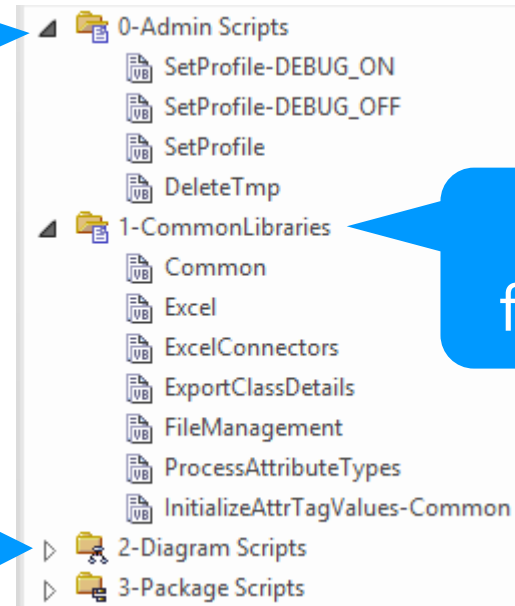
Scripts organization

 Use Groups to organize your scripts

 Example:

Reserved for the scripts administrator/dev team

Users scripts, available from a diagram or the Project Browser



Libraries of reusable functions and routines

EA SCRIPTS OVERVIEW

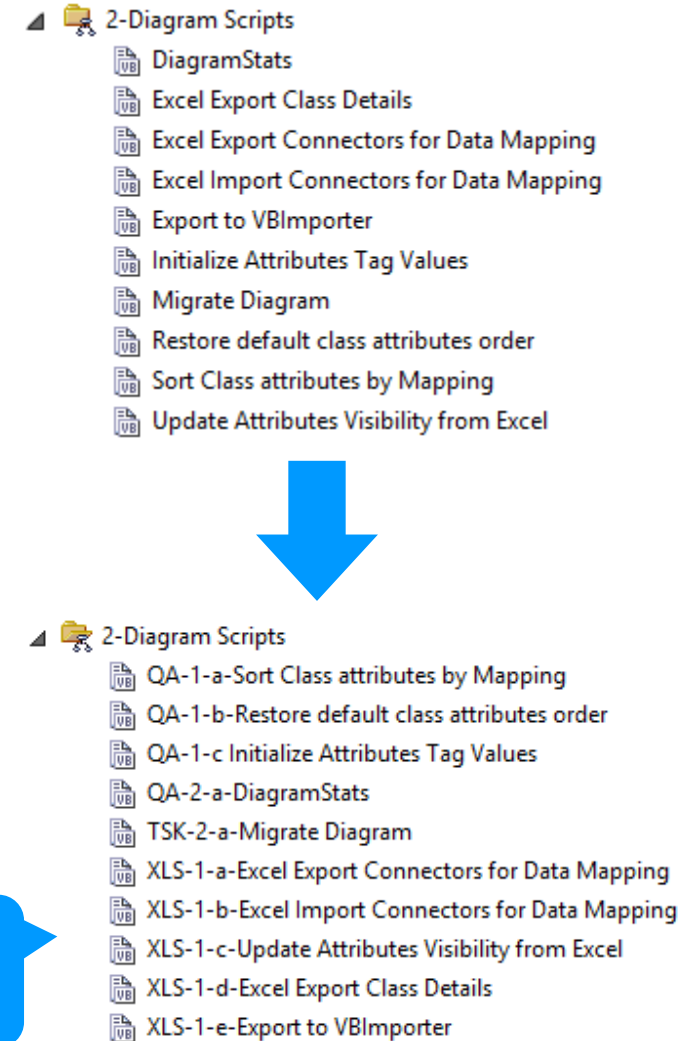
Naming conventions

Define prefixes to group and identify script functions

Example:

- QA = Quality related (QA-1 = cleaning, QA-2 = stats, QA-3 = controls)
- SEC = Users, groups and permissions
- TSK = Automated tasks (TSK-1 = generate, TSK-2 = migrate, TSK-3 = update)
- XLS = Excel related scripts (XLS-1 = import/export, XLS-2 = others)
- XT = Extract EA information

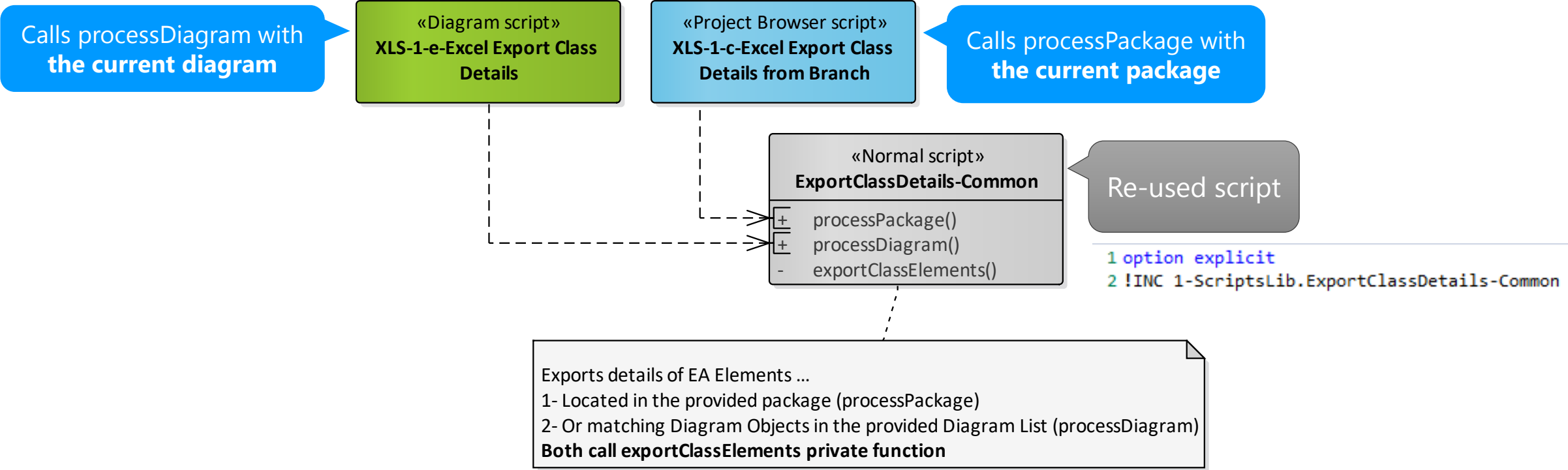
New order



EA SCRIPTS OVERVIEW

Re-used script functions

Project Browser and Diagram scripts + Common script



EA SCRIPTS OVERVIEW

EA API Methods

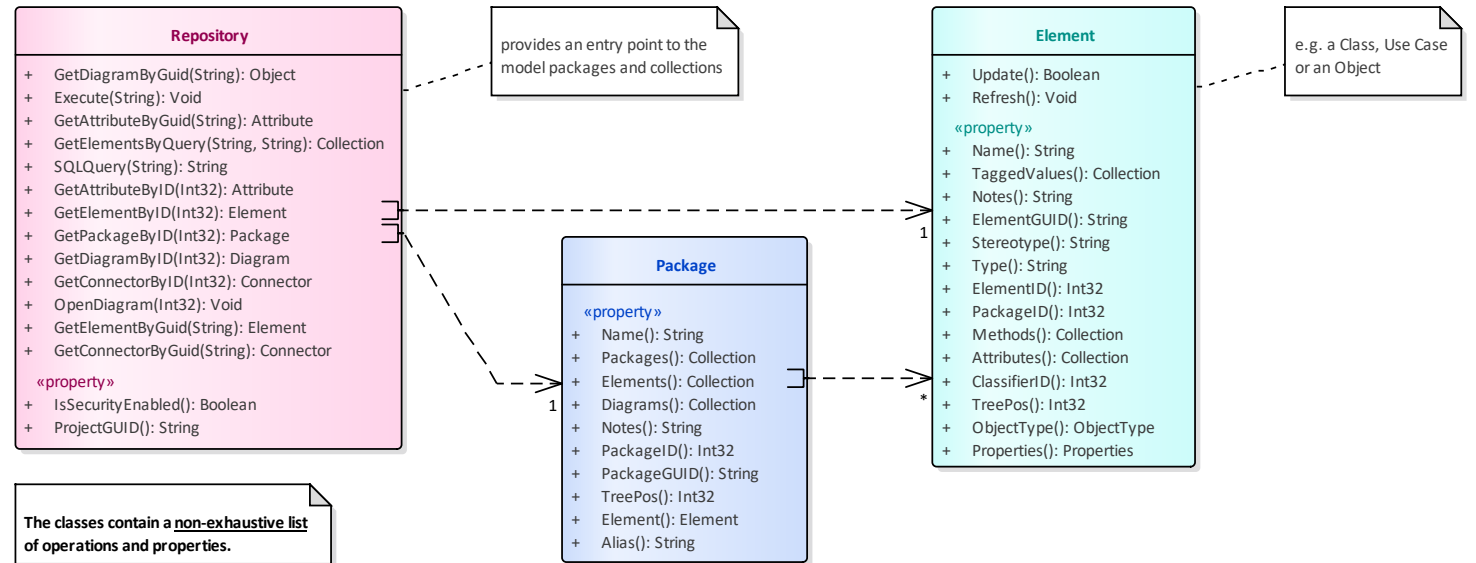
EA Automation Interface provides a way of accessing the internals of Enterprise Architect models

EA API Object Model

Repository

Package

Element, Connector, Diagram, etc.



Using Object methods - VBScript examples:

set curElement = Repository.GetElementByID(selectedDiagObject.ElementID): element matching a diagram object

set parentPkg = Repository.GetPackageByID(curElement.PackageID): element's parent package

EA SCRIPTS OVERVIEW

Run SQL queries against the model

SQL queries can be used ...

➤ When a query or update cannot be achieved via a method.

➤ To select specific elements, preventing time consuming loops.

e.g. select *datatype stereotyped Classes* with the *name* matching the *given variable*;

```
SELECT Object_ID FROM t_object WHERE (Object_Type = 'Class' and (Stereotype = 'datatype'))  
and name = '"&attributeType&"'
```

➤ To address performance issues related. Example with a bulk creation of ~1000 attributes :

API method call execution time:  **120 minutes**

```
set elementAttr = elt.Attributes.AddNew("&attrName&", "")  
elementAttr.Update()
```

API method call with a SQL insert execution time:  **4 minutes**

```
insert into t_attribute(object_id, name, ea_guid) values ...
```

EA SCRIPTS OVERVIEW

Use SQL queries

Repository object available methods:

Repository.GetElementSet(query,2) : *returns a collection of EA elements according to the SQL query*

```
11 sub main
12   Dim SQL, className
13   dim elements AS EA.Collection
14   dim matchingElement AS EA.Element
15   className = "Demo"
16   SQL = "select object_id from t_object where Object_Type='Class' and Name='Demo'"
17   set elements = Repository.GetElementSet(SQL,2)
18   Session.Output "Number of classes found: " & elements.Count
19   for each matchingElement in elements
20     Session.Output "Class found: " & matchingElement.Name & " | " & matchingElement.ElementGUID
21   next
22 end sub
```

System Output

Number of classes found: 3
Class found: Demo | {58CD86E0-875F-42b7-8ABB-0FBA1ABB80CD}
Class found: Demo | {DB070719-3CC9-46f1-BA9B-6C0EC076BFEF}
Class found: Demo | {C8D88A83-495A-40c1-B991-E87B0C0D9A6C}

EA SCRIPTS OVERVIEW

Use SQL queries

Repository object available methods:

- `Repository.GetElementSet(query, 2)` : returns a collection of EA elements according to the SQL query
- `Repository.SQLQuery(query)` : returns an XML formatted string value (SELECT queries only)

```
11 sub main
12   Dim SQL, className
13   dim queryResult
14   dim matchingElement AS EA.Element
15   className = "Demo"
16   SQL = "select t_attribute.ea_guid, t_object.Name as ClassName, t_attribute.Name, t_attribute.Type "
17   SQL = SQL & "from t_attribute inner join t_object on t_attribute.Object_ID = t_object.Object_ID "
18   SQL = SQL & "where t_object.Object_Type='Class' and t_object.Name='" & className & "'"
19   queryResult = Repository.SQLQuery(SQL)
20   Session.Output queryResult
21 end sub
```

```
xml version="1.0"
ADATA version="1.0" exporter="Enterprise Architect">
  <Dataset_0>
    <Data>
      <Row>
        <ea_guid>{DB2F2F3A-602F-47a8-B1E1-6F93AA8D4D7A}</ea_guid>
        <classname>Demo</classname>
        <name>attrB</name>
        <type>String</type>
      </Row>
      <Row>
        <ea_guid>{9985613A-259B-427c-8F58-5234428BA8B3}</ea_guid>
        <classname>Demo</classname>
        <name>attrA</name>
        <type>String</type>
      </Row>
      <Row>
        <ea_guid>{D1A8BC1C-F319-47b6-B4B6-02749120C89E}</ea_guid>
        <classname>Demo</classname>
        <name>attr2</name>
        <type>Integer</type>
      </Row>
      <Row>
        <ea_guid>{9F405C1D-90F9-4b4b-A011-C8A159EBB1DD}</ea_guid>
        <classname>Demo</classname>
        <name>attr1</name>
        <type>Boolean</type>
      </Row>
    </Data>
  </Dataset_0>
</EADATA>
```

EA SCRIPTS OVERVIEW

Use SQL queries

Repository object available methods:

- `Repository.GetElementSet(query, 2)` : **returns** a collection of EA elements according to the SQL query
- `Repository.SQLQuery(query)` : **returns** an XML formatted string value (SELECT queries only)
- `Repository.Execute(query)` : **runs** a query (UPDATE/DELETE/INSERT)

```
SQL = "select t_attribute.ea_guid, t_object.Name as ClassName, t_attribute.Name, t_attribute.Type "  
SQL = SQL & "from t_attribute inner join t_object on t_attribute.Object_ID = t_object.Object_ID "  
SQL = SQL & "where t_object.Object_Type='Class' and t_object.Name='" & className & "'" "  
queryResult = Repository.SQLQuery(SQL)  
Session.Output queryResult  
set xmlDom = XMLParseXML(queryResult)  
attributesList = XMLGetNodeTextArray (xmlDom, "//ea_guid")  
for each attItem in attributesList  
    Session.Output "Updating attribute with guid " & attItem  
    SQL = "update t_attribute set Type='String' where ea_guid='" & attItem & "'" "  
    if (bDebugMode) then  
        Session.Output SQL  
    end if  
    queryResult = Repository.Execute(SQL)  
next
```

The diagram illustrates the effect of the SQL query on the 'Demo : Features' table. The top table shows the initial state, and the bottom table shows the state after the update.

Name	Type	Scope
attr1	Boolean	Private
attr2	Integer	Private
New Attribute...		

Name	Type	Scope
attr1	String	Private
attr2	String	Private
New Attribute...		

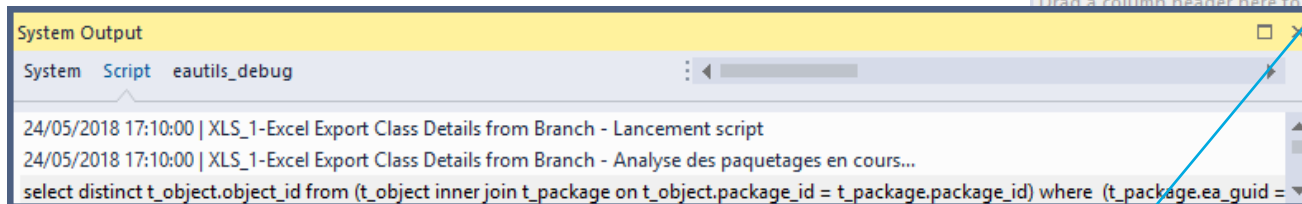
EA SCRIPTS OVERVIEW

Use SQL queries

- Be careful when running UPDATE/DELETE/INSERT queries, **it can delete or break your model!**
- Queries must be compatible with the used DBMS (MySQL, SQL Server, Postgres, Oracle, Jet/EAP)
 - e.g. apply the right **asterisk wild card** → * vs %
 - Use **Repository.RepositoryType** method to get the DB name e.g. POSTGRES

➤ Use EA SQL Scratch pad

- To build queries (IntelliSense available: Ctrl+Space)
- To test and fix queries from the DEBUG System Output



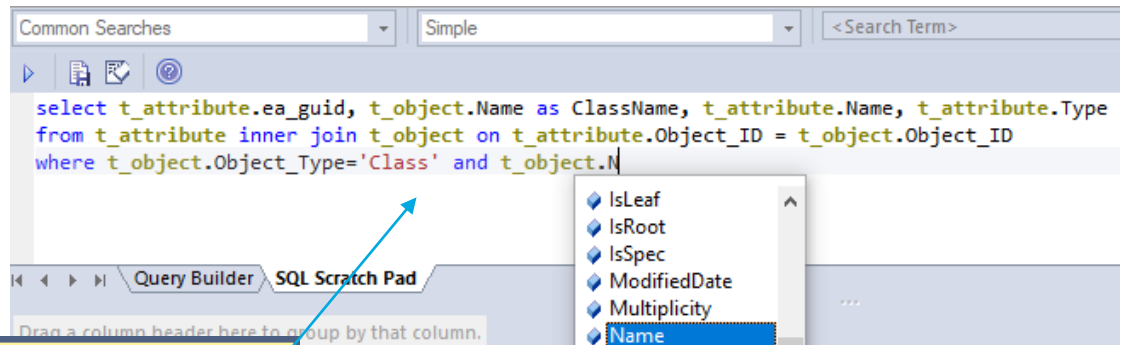
System Output

System Script eautils_debug

24/05/2018 17:10:00 | XLS_1-Excel Export Class Details from Branch - Lancement script

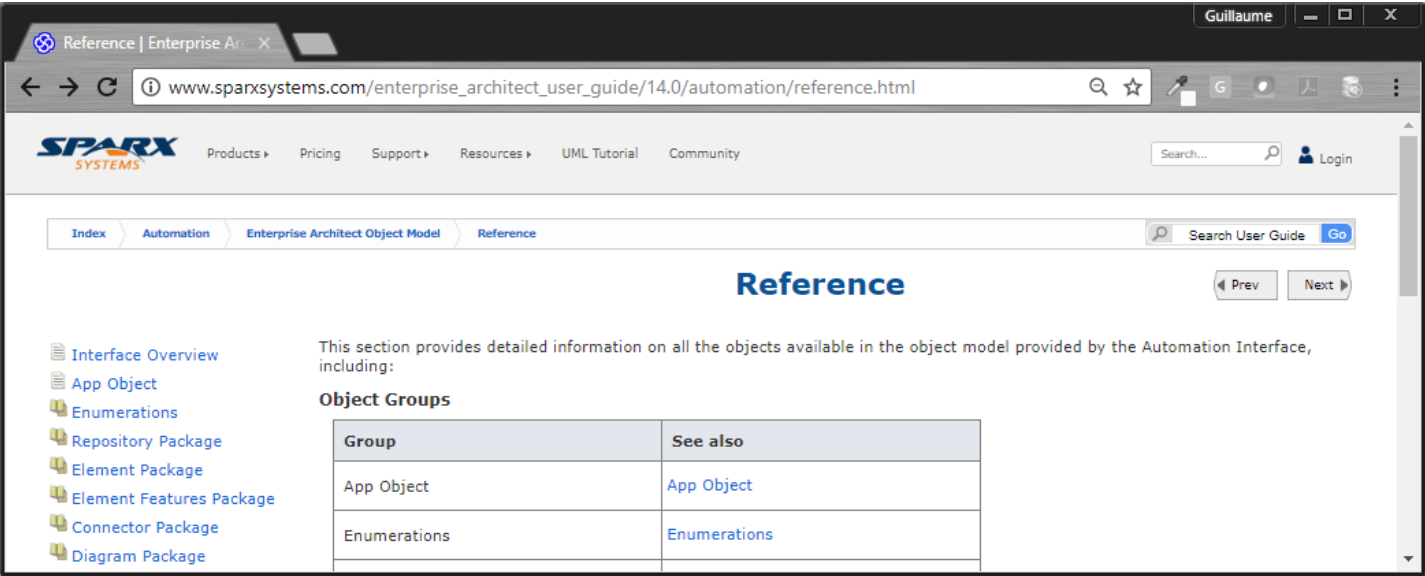
24/05/2018 17:10:00 | XLS_1-Excel Export Class Details from Branch - Analyse des paquetages en cours...

```
select distinct t_object.object_id from (t_object inner join t_package on t_object.package_id = t_package.package_id) where (t_package.ea_guid =
```



Object Model API: see EA User Guide

http://www.sparxsystems.com/enterprise_architect_user_guide/14.0/automation/reference.html



EA DB schema: generate a reverse (98 tables)

t_object
«column»
*PK object_id: integer
object_type: varchar(255)
diagram_id: integer = 0
name: varchar(255)
alias: varchar(255)
author: varchar(255)

t_attribute
«column»
object_id: integer = 0
name: varchar(255)
scope: varchar(50)
stereotype: varchar(50)
containment: varchar(50)
isstatic: integer = 0

t_diagram
«column»
*PK diagram_id: integer
package_id: integer = 1
parentid: integer = 0
diagram_type: varchar(50)
name: varchar(255)
version: varchar(50) = '1.0'

CONTENT

01. INTRODUCTION

02. EA SCRIPTS OVERVIEW

03. SCRIPTS HELPERS

04. EA-MATIC TRIGGERED SCRIPTS

05. DEBUG & HINTS

06. DEMONSTRATION

SCRIPTS HELPERS

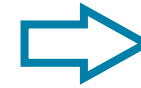
For the developer



Scripting variables used

```
bShowOutput = false  
'bShowOutput = true
```

Debug Mode



```
if (bShowOutput) then  
    Session.Output SQL  
end if
```

Prompt for a Folder?

```
sOutputPath = ""  
'sOutputPath = "C:\Users\GFI\Desktop"
```

Script name for
the System Output

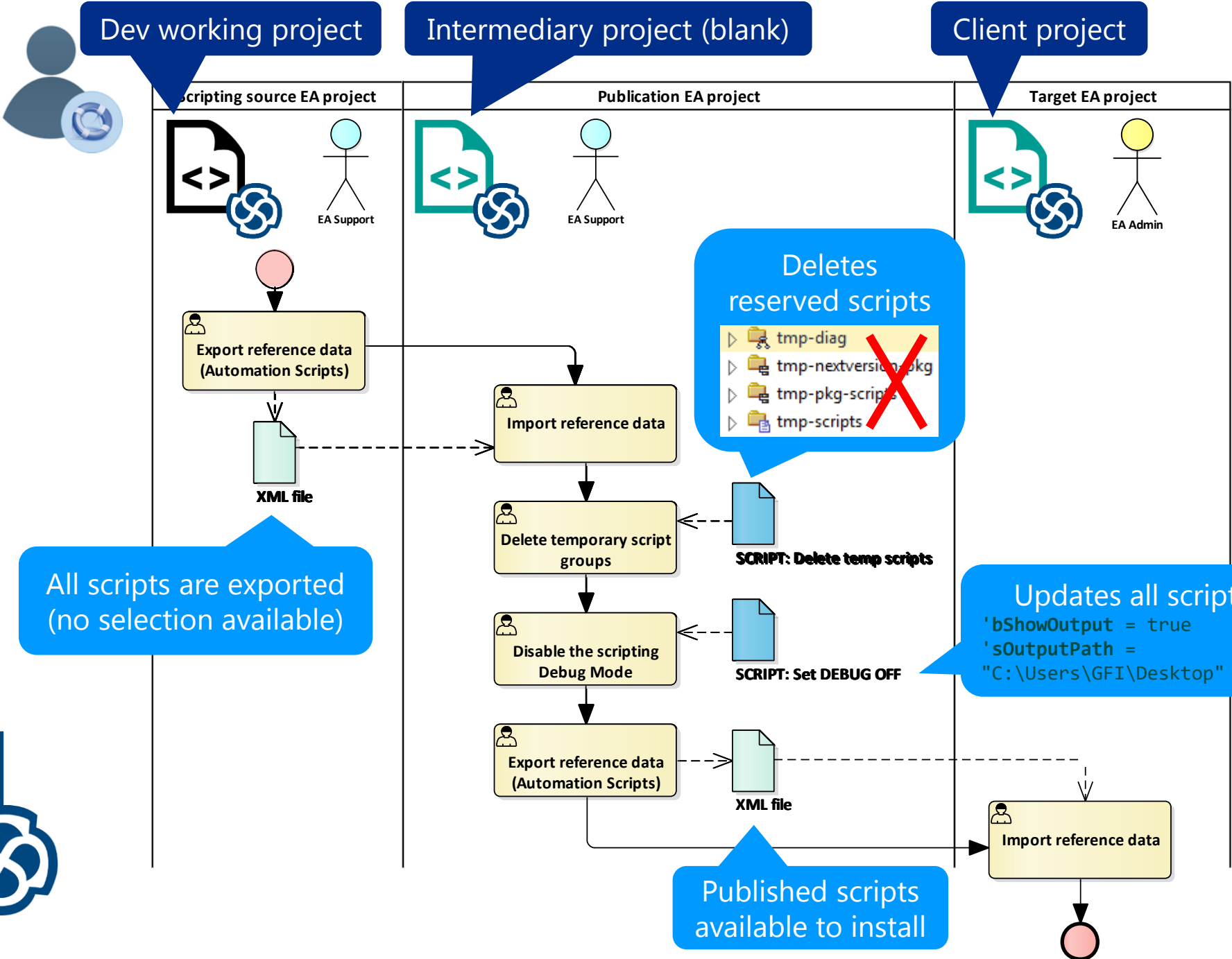
```
dim outHeader  
outHeader = " | XLS_1-Excel Export Class Details from Branch"
```



```
Session.Output Now & outHeader & " - Starting"
```

SCRIPTS HELPERS
For the developer

Publication process :



SCRIPTS HELPERS

For the users



 Store the last selected folder for the input or output file(s)

 Save time with the option to reuse the previous file location

 Apply scripts filters

 Workaround for the lack of sub menus with 15+ scripts

 Built-in help

DEMO



CONTENT

01. INTRODUCTION
02. EA SCRIPTS OVERVIEW
03. SCRIPTS HELPERS
04. **EA-MATIC TRIGGERED SCRIPTS**
05. DEBUG & HINTS
06. DEMONSTRATION

EA-MATIC ADD-IN

Automatically triggered scripts for users



ea-Matic is a free add-in by Geert Bellekens.

bellekens.com/ea-matic

EA broadcast events are sent to all add-ins.

e.g. **EAOnPostNewElement** is received when a new element is created

ea-Matic can trigger a script with an associated event.

Replaces users repetitive tasks.

Add-in installed only for the target users.

"Magic keyword"

Associated event

```
'EA-Matic
'a new connector has been created.
function EA_OnPostNewConnector(Info)
'get the connector id
dim connectorID, newConnector
connectorID = Info.Get("ConnectorID")
set newConnector = Repository.GetConnectorByID(connectorID)
MsgBox "New connector type: " & newConnector.Type
end function
```

DEMO



CONTENT



01. INTRODUCTION

02. EA SCRIPTS **OVERVIEW**

03. SCRIPTS **HELPERS**

04. EA-MATIC TRIGGERED SCRIPTS

05. **DEBUG & HINTS**

06. DEMONSTRATION

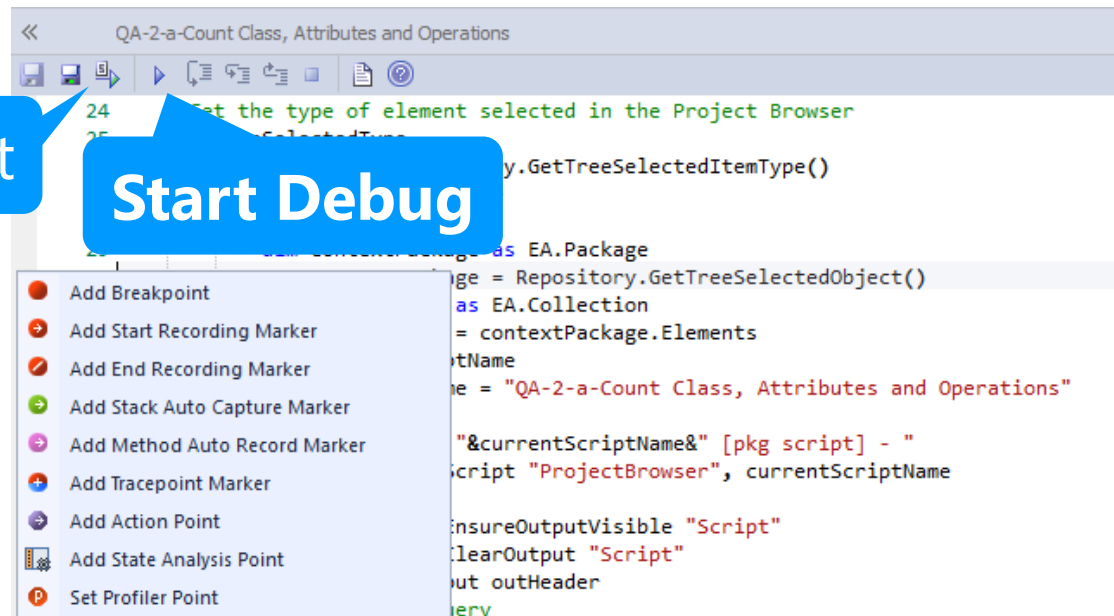
SCRIPT DEBUGGING

Principles

- Script debugging aids in the development and maintenance of scripts.
- Set Breakpoints to step over/into/out.
- View local variables values at the time of execution.

Run Script

Start Debug



HINTS

Code editor key shortcuts

➤ Ctrl+F2 to set bookmarks, F2/Shift+F2 go to next or previous

```
16 sub OnProjectBrowserScript()  
17     Dim bShowOutput  
18     'true = afficher les mess
```

➤ Ctrl+Shift+C = Toggle line comment on selection

```
25 ' dim treeSelectedType  
26 ' treeSelectedType = Repository.GetTreeSele  
27 ' select case treeSelectedType  
28 '     case otPackage
```

➤ Ctrl+Space = Intellisense

```
!3 Repository.  
!4 ' Ge  
!5 dim  
!6 tree  
!7 sele  
ActivateDiagram  
ActivatePerspective  
ActivateTab  
ActivateTechnology
```

➤ Shift+Alt+Return = Fullscreen

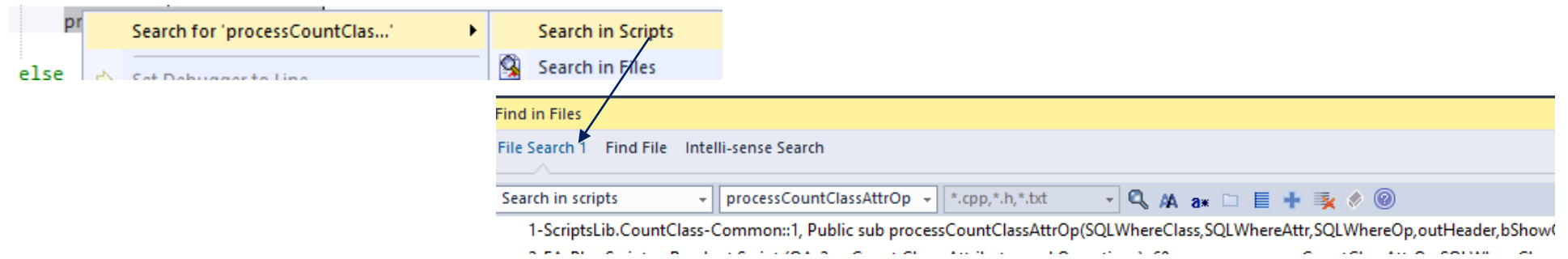
➤ Ctrl+U = to lowercase, Ctrl+Shift+U = to uppercase

➤ Ctrl+keypad(-) Zoom out, Ctrl+keypad(+)
Zoom in, Ctrl+keypad(/) restore

HINTS

Various

Search in Scripts



Use Sparx EA built-in libraries

```
3 !INC Local Scripts.EAConstants-VBScript
4 !INC EAScriptLib.VBScript-XML
```

CONTENT

01. INTRODUCTION

02. EA SCRIPTS OVERVIEW

03. SCRIPTS HELPERS

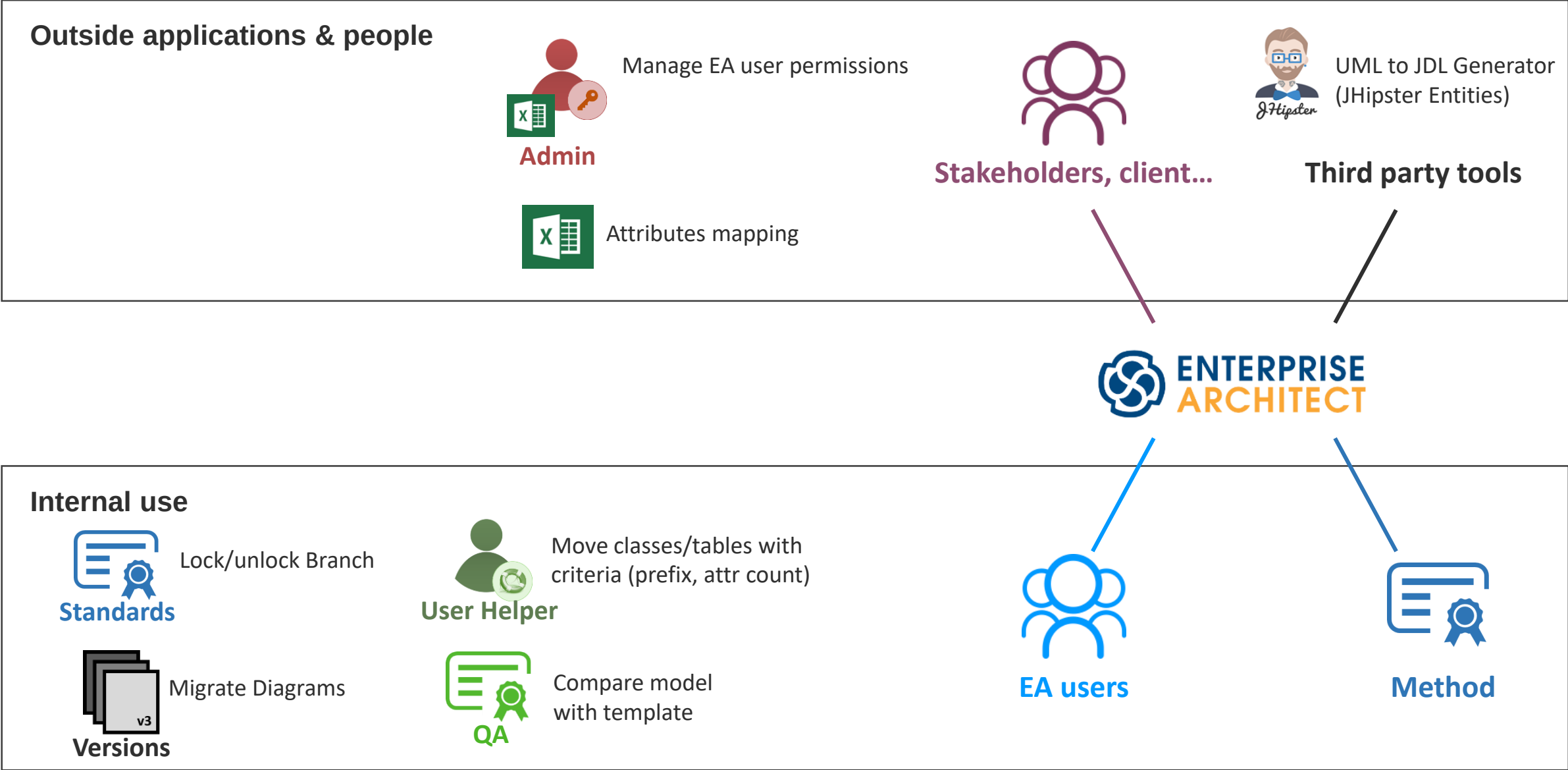
04. EA-MATIC TRIGGERED SCRIPTS

05. DEBUG & HINTS

06. DEMONSTRATION

DEMONSTRATION

Main topics



DEMONSTRATION

Main topics



EA User Group London 2018

Extending EA with custom scripts

Author: Guillaume FINANCE | Blog: www.umlchannel.com | EA add-in: www.eautils.com

SCRIPTS DEMONSTRATION

Lock/unlock branch
Use read only mode (not avail via the UI)

Schema.org Data Model



Standards

Compare a model with its original template
> mark populated / used models with *

Template Compare



QA

Migrate diagrams between versions
reverse on new releases with the need to maintain custom diagrams before deleting old versions

Migrate Diagrams



versions

EA Security management from Excel
Context: with large number of EA projects to maintain, this Excel file let the Admin update the group permissions and assigned users in a consistent manner.

[EA Security & users management](#)



User Permissions

Attributes mapping
Excel import/export on attributes to map Data Model (e.g. Application compliance with a standard)

LDM/Schema.org mapping



Sparx UML to JHipster JDL integration

JHipster 2 EA integration



Isolate classes based on a prefix/suffix or number of attributes in a package
> useful for large DB Reverse (>100 tables + >50 columns on tables)

DBReverse1



User Helper

DEMO



CONTACT

Guillaume FINANCE

-  Blog: www.umlchannel.com
-  Add-in: www.eautils.com
-  Twitter: @umlchannel
-  LinkedIn: fr.linkedin.com/in/guillaumefinance

