OpenEdge Architect Template Customization

"Progress Development Studio for OpenEdge®"
in OpenEdge 11

Mike Fechner, Director, Consultingwerk Ltd.
mike.fechner@consultingwerk.de

Mike Fechner, Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on OpenEdge and .NET
- Located in Cologne, Germany
- Customers in Germany, EMEA, USA
- Vendor of tools and consulting programs (specialized on GUI for .NET, Modernization)
- 23 years of Progress experience (V5 … V11.2)
- Passionate OpenEdge Architect user since 10.1A
- But I'm not a Java programmer…

OpenEdge® application modernization solutions

- WinKit
- **SmartComponent Library**
- Dynamics4.NET

- Tools can be used together or separately
- Shared common code base

- SmartComponents.Mobile
- SmartComponents.Web
- SmartBPMAdapter for OpenEdge BPM/Savvion

Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

OpenEdge Architect

- Eclipse based, extensible
- Available since 10.1A (continuously enhanced since then)
- Alternative to the AppBuilder / OpenEdge Studio
- Requires a separate license (included in PSDN SDK), parallel installation possible
- AppBuilder, ADM1/2 and Dynamics integrated
- Extensible using 3rd party plug-ins (commercial or community based)
- Windows only

Reasons to migrate to OpenEdge Architect

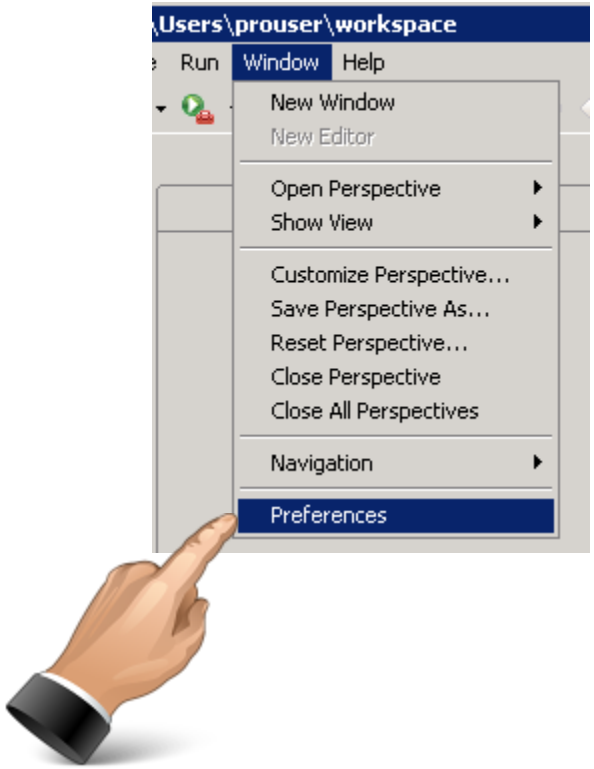
- GUI for .NET
  - Visual Designer, Class Browser
- object-oriented coding
  - no class template in AppBuilder, class browser
- AppServer Development (10.2B, 11.x)
- OE Mobile
- REST Adapter
- OpenEdge BPM

Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

Workspace preferences in OEA

- Main menu-bar, Window -> Preferences



Workspace preferences in OEA

- General – Editors – Text Editors
  - Insert spaces for Tabs, Tab width, show line numbers, …
- General – Search
  - Reuse editors … turned off
- OpenEdge Architect – Editor
  - As you type: Case (upper or lower), case keywords, depending on your style: expand keywords

Workspace preferences in OEA

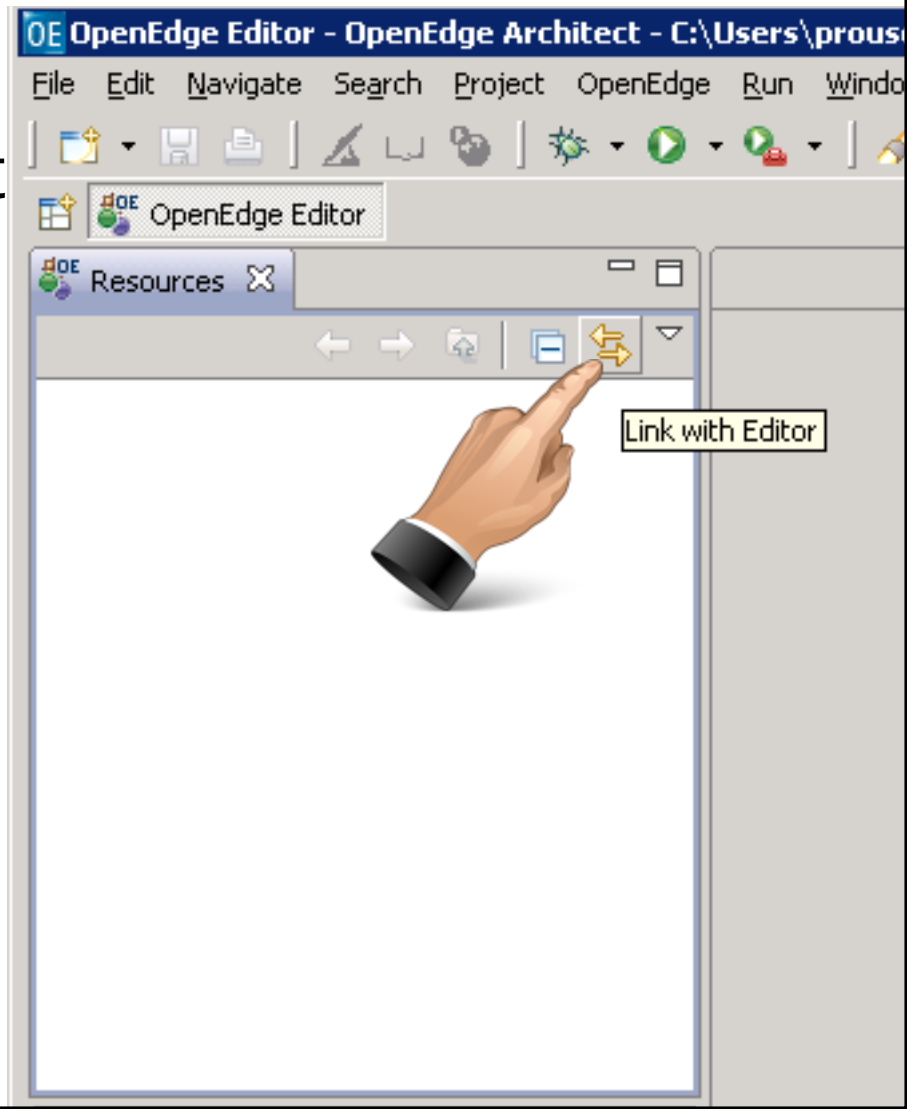
- General – Workspace
  - Refresh automatically (on fast machines, local source code)
- Refreshing causes Eclipse to permanently search the file system for new, deleted, modified files
- Sync‘s Eclipses view of the world to the file system
- May cause recompile
- Files not in sync will be skipped on compile, search, etc… ☹
- Alternative is manual refresh (e.g. F5 key)

Workspace preferences in OEA

- OpenEdge Architect – Editor – Build
  - Auto check syntax (while typing)
  - Maximum number of errors reported per build: much higher than 100 (e.g. 100,000) when you have
    - templates or scratch files
    - migration routines that typically never compile
    - just never cleaned up source code (Copy of test.p, test_backup.p)

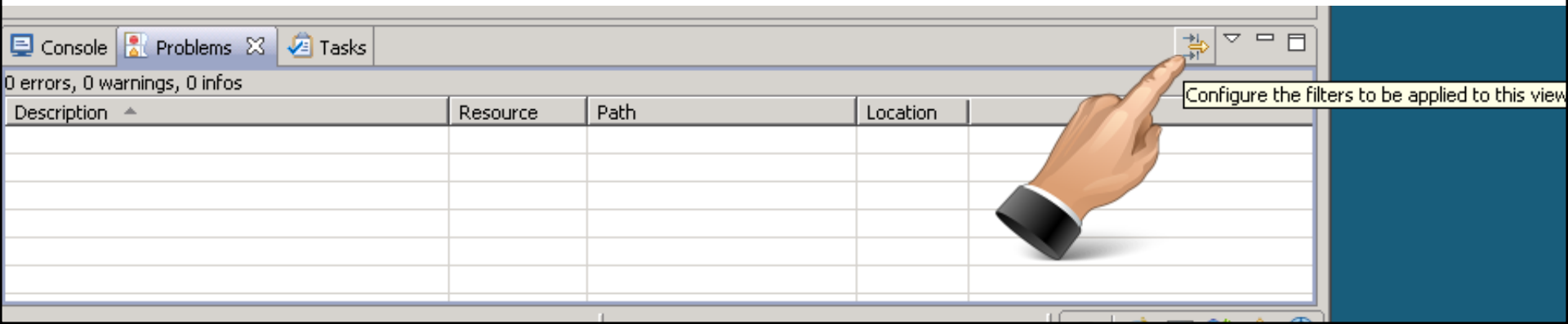
Link with Editor (Resource view)

- Keeps resource view (file browser) and current editor in sync when switching editor tabs



Error filters

- Use error filters dialog to configure filters
  - default (show all) usually not helpful on large projects
  - my preference is ‘selected element and its children‘
  - or use working sets



Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

OpenEdge Editor Macros

- Defined in the Workspace Preferences
- Progress OpenEdge → Editor → Templates (Macros)
- Macros can be inserted automatically when you type the name followed by the space key
- Macros may also require the content assist dialog (CTRL-SPACE)

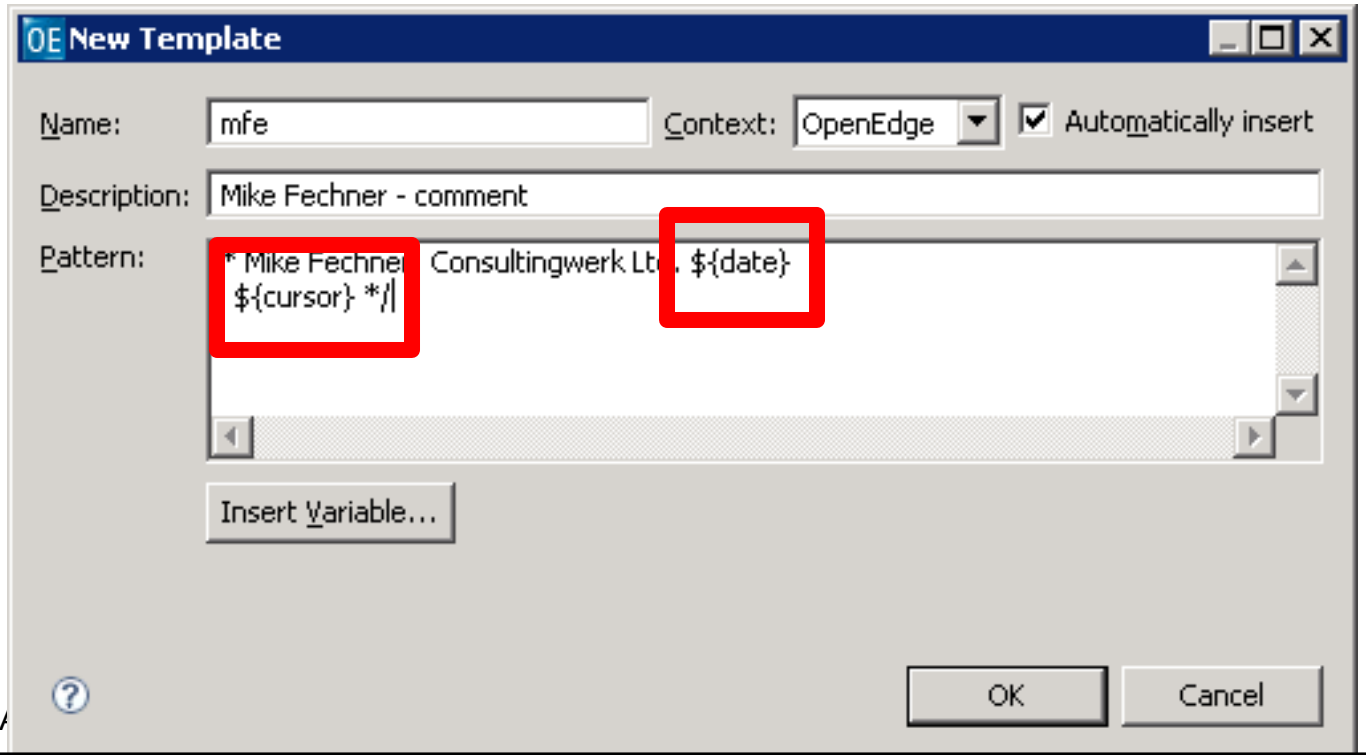
OpenEdge Editor Macros

- Progress preinstalls a set of Macros
- DVCH: DEFINE VARIABLE AS CHARACTER
- MES: MESSAGE VIEW-AS ALERT-BOX
- &FR {&FRAME-NAME}
- …

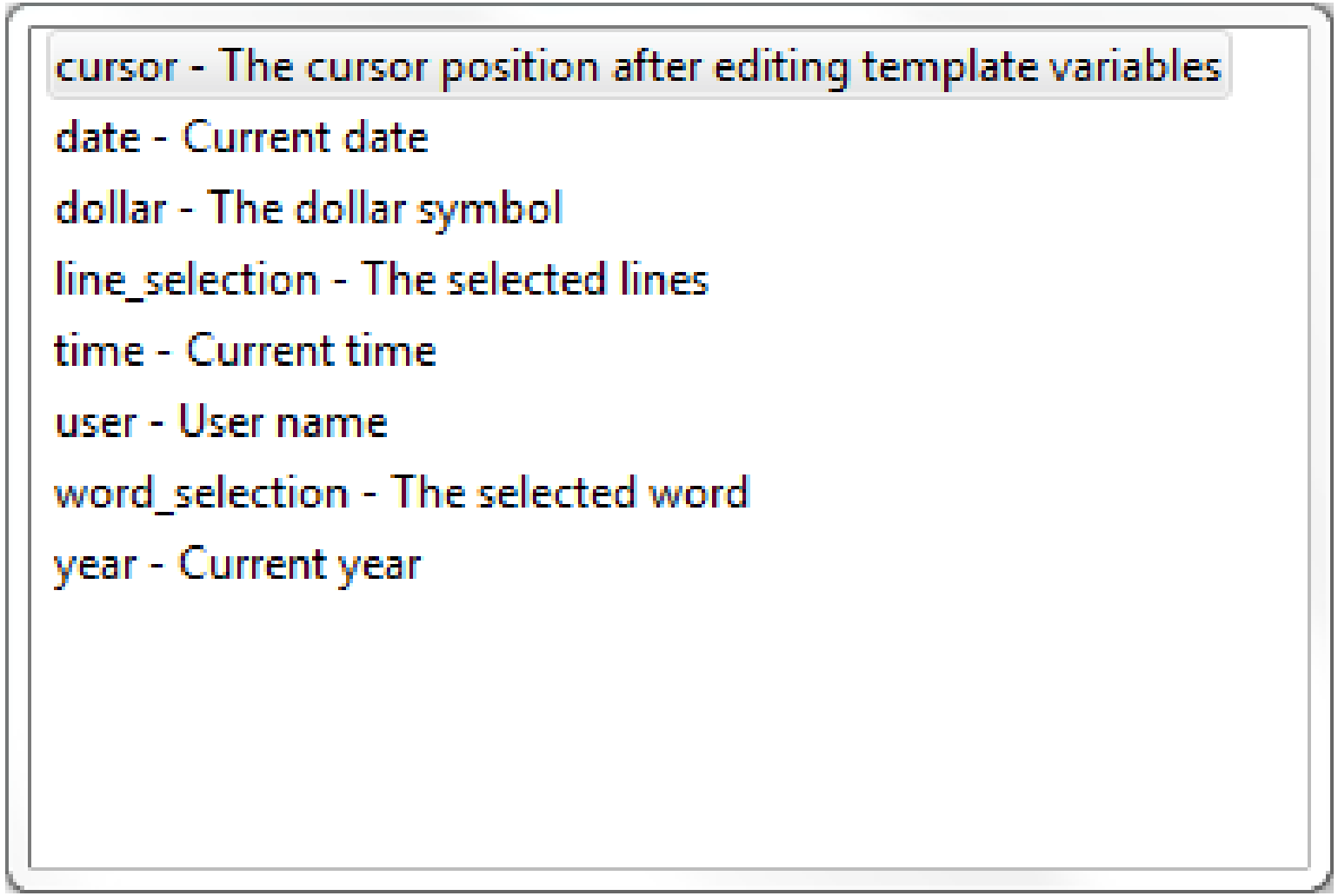
- MES Macro requires some polishing…

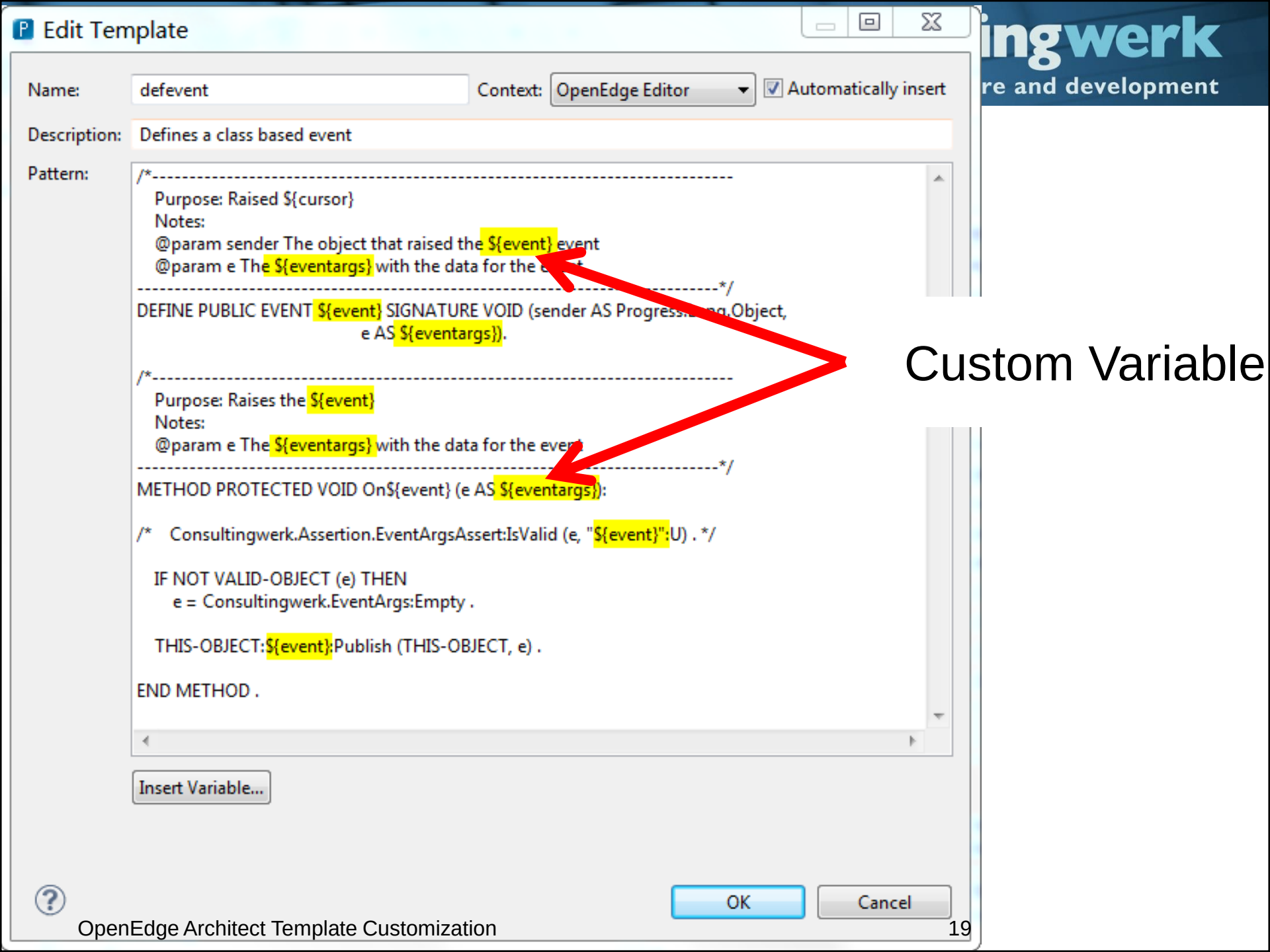
OpenEdge Editor Macros

- Macros may contain variables that will be replaced when inserted to the editor
- ${cursor} defines where the cursor should remain



Macro variables




Edit Template
Name:
defevent
Context:
OpenEdge Editor
Automatically insert
Description:
Defines a class based event
Pattern:
/*--
Purpose: Raised ${cursor}
Notes:
@param sender The object that raised the ${event} event
@param e The ${eventargs} with the data for the
--*/
DEFINE PUBLIC EVENT ${event} SIGNATURE VOID (sender AS Progress.Lang.Object,
e AS ${eventargs}).
/*--
Purpose: Raises the ${event}
Notes:
@param e The ${eventargs} with the data for the event
--*/
METHOD PROTECTED VOID On${event} (e AS ${eventargs}):
/* Consultingwerk.Assertion.EventArgsAssert:IsValid (e, "${event}":U) . */
IF NOT VALID-OBJECT (e) THEN
e = Consultingwerk.EventArgs:Empty .
THIS-OBJECT:${event}:Publish (THIS-OBJECT, e) .
END METHOD .
Insert Variable...
OK
Cancel
Custom Variable

Exporting and Importing Macros

- Macros are stored in the workspace metadata
- Exporting and importing allows reusing in different workspaces and sharing with other people
- Import and Export Buttons in the Macro Dialog

Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

Setting the Author name for Templates

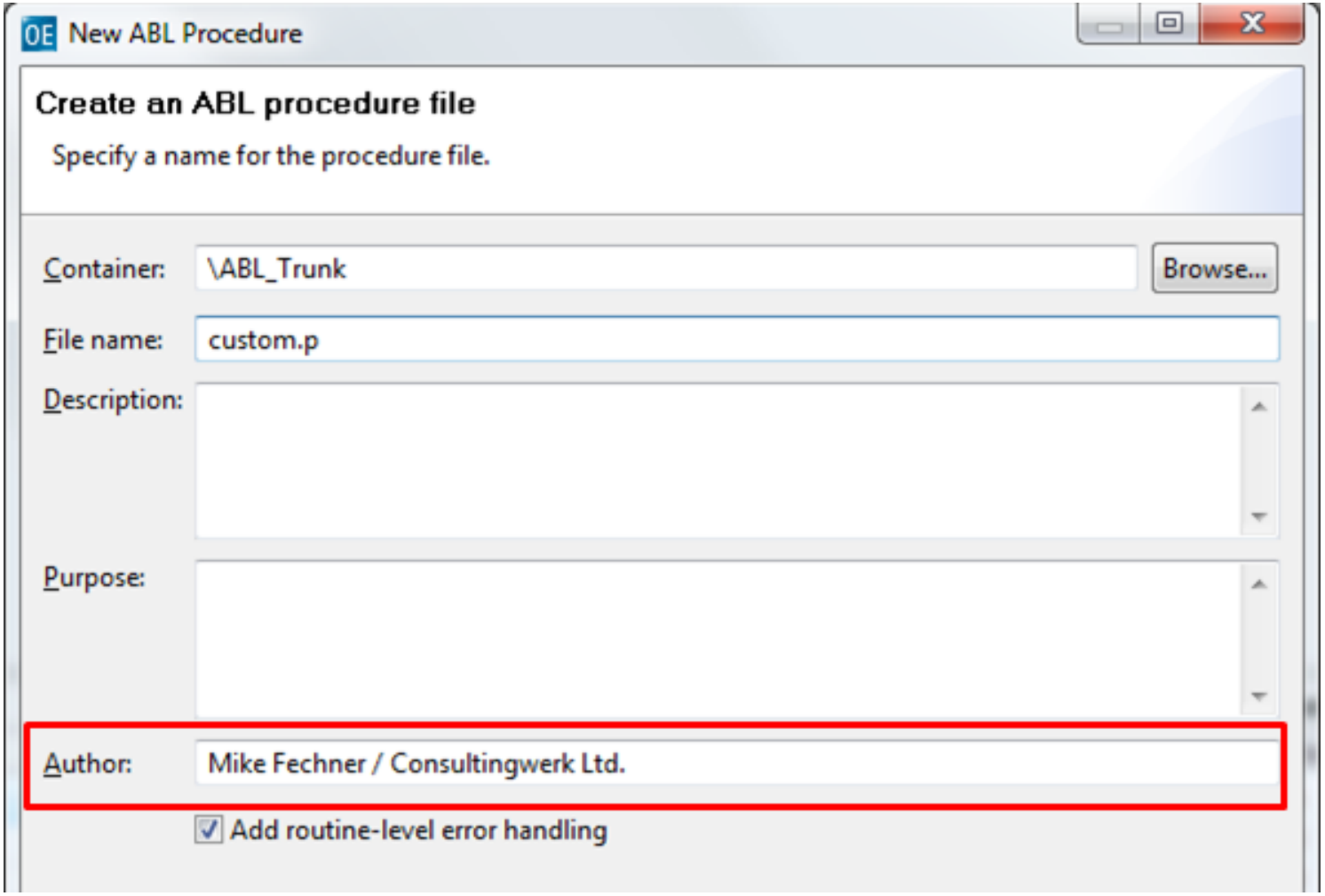
- Some of the Templates and Wizards use the **windows login name** as the Author reference
  - Mike
  - mikefe
  - fec878
  - Administrator
  - Progress

Setting the Author name for Templates

- Use vmargs **–Duser.name Startup parameter** to override this
- http://blog.consultingwerk.de/consultingwerkblog/2011/08/customizing-openedge-architect-templates-made-easy-personalization/

```
C:\Progress\OpenEdge102B\oeide\eclipse\eclipse.exe -showlocation -vm "C:\Progress\OpenEdge102B\jre\bin\javaw.exe" -vmargs -Duser.name="Mike Fechner / Consultingwerk Ltd."
```

Setting the Author name for Templates

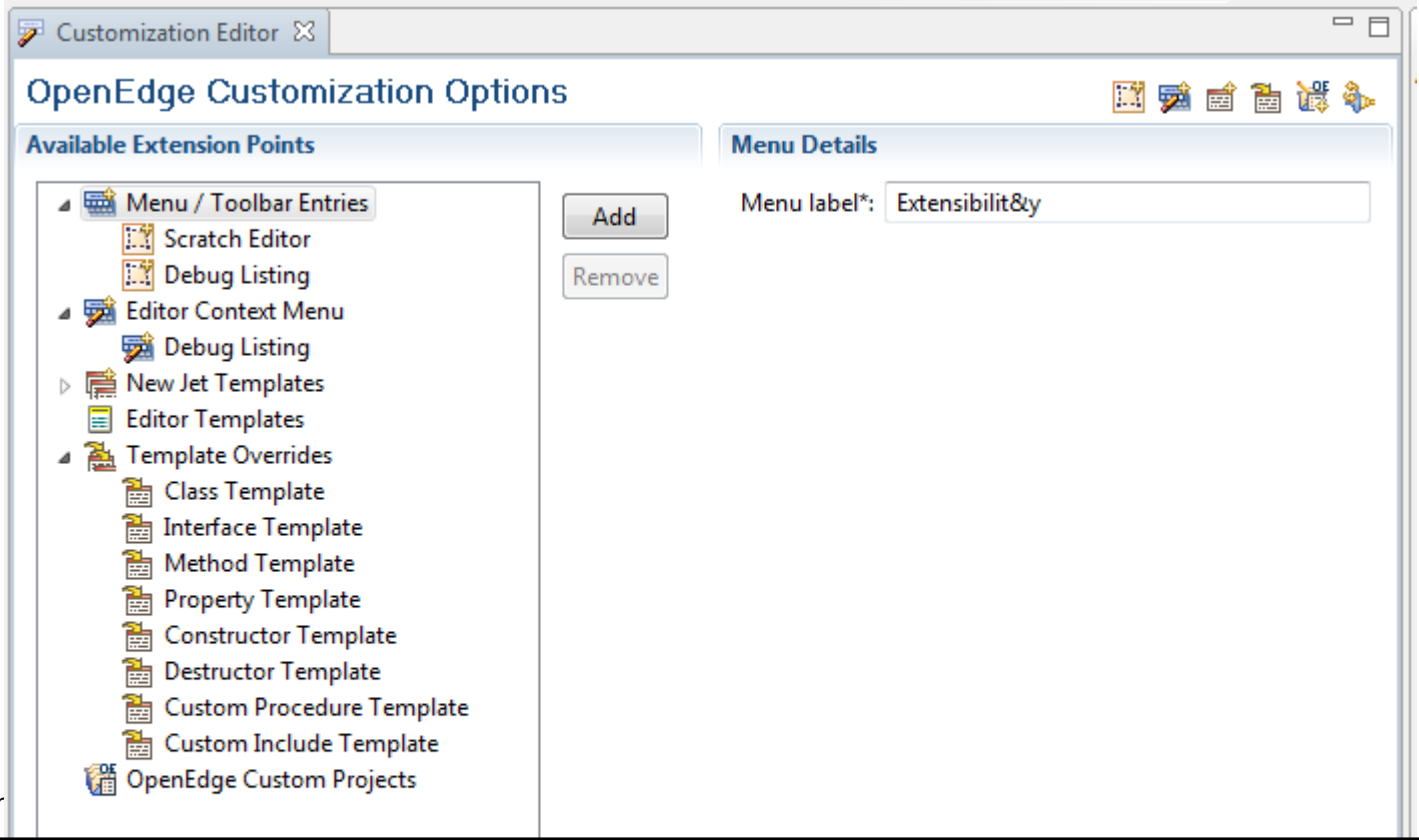


Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

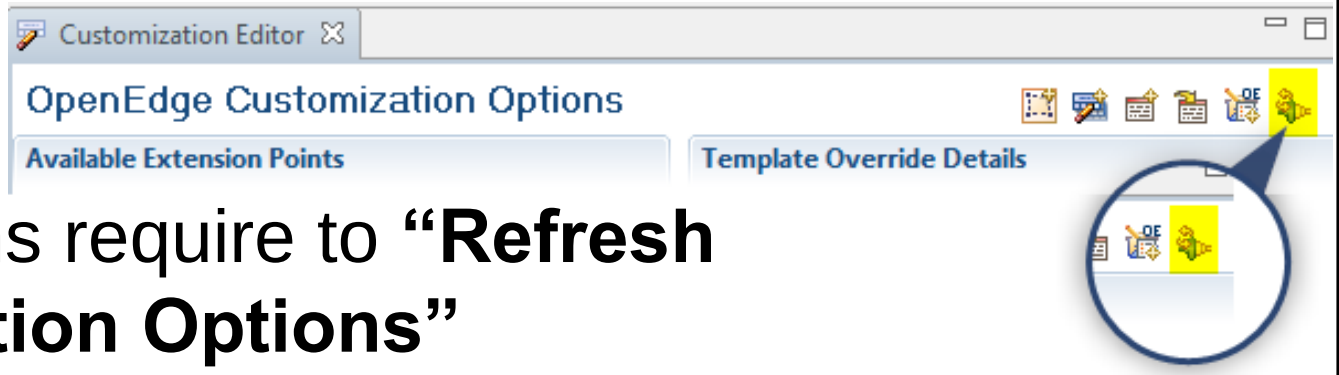
Progress Customization Editor

- In the OpenEdge Editor Perspective
- OpenEdge -> Tools -> Customization Editor



Progress Customization Editor

- ProTools for PDSOE:
  - Menu / Toolbar Entries, Editor Context Menu
- Template Overrides
- Custom Project Templates

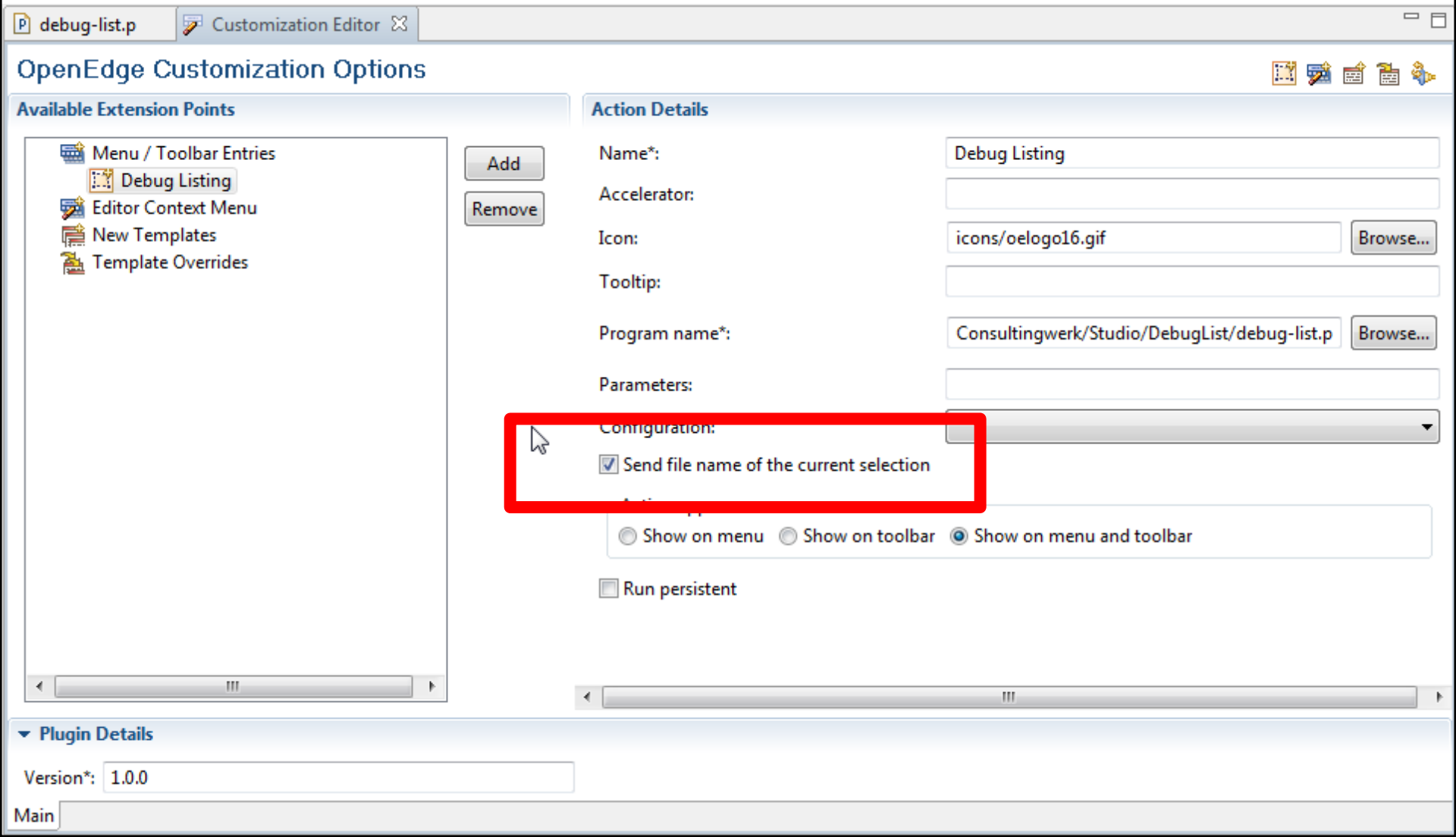


- Modifications require to **“Refresh Customization Options”**
- When prompted for resetting the perspective, choose no!

Opening files form ABL code

- Useful for Menu / Toolbar Entries and Editor Context Menu
- Public API‘s defined in adecomm/oeideservice.i
- Supports querying basic project properties, opening views for ABL windows (like OEUnit does) and opening files
- getProjectName ()
- openEditor ()

Sample Open Debug Listing



Sample Open Debug Listing

```
ROUTINE-LEVEL ON ERROR UNDO, THROW.

DEFINE INPUT  PARAMETER pcFileName AS CHARACTER NO-UNDO.

{adecomm/oeideservice.i}

/* ***************************  Main Block  *************************** */

DEFINE VARIABLE cFileName AS CHARACTER          NO-UNDO .
DEFINE VARIABLE cProject  AS CHARACTER          NO-UNDO .
DEFINE VARIABLE oFileInfo AS System.IO.FileInfo NO-UNDO .

DEFINE VARIABLE cDebugList AS CHARACTER NO-UNDO.

ASSIGN cProject  = getProjectName ()
       cFileName = SUBSTRING (pcFileName, 2)

       oFileInfo = NEW System.IO.FileInfo (cFileName)

       cDebugList = /*createLinkedFile ("":U, ".p")*/
                    SESSION:TEMP-DIRECTORY + "debuglist-":U + oFileInfo:Name.

COMPILE VALUE (cFileName)
    SAVE = FALSE
    DEBUG-LIST VALUE (cDebugList)
    NO-ERROR .

openEditor
     (?,
      cDebugList,
      "UNTITLED",
      ?) .
```

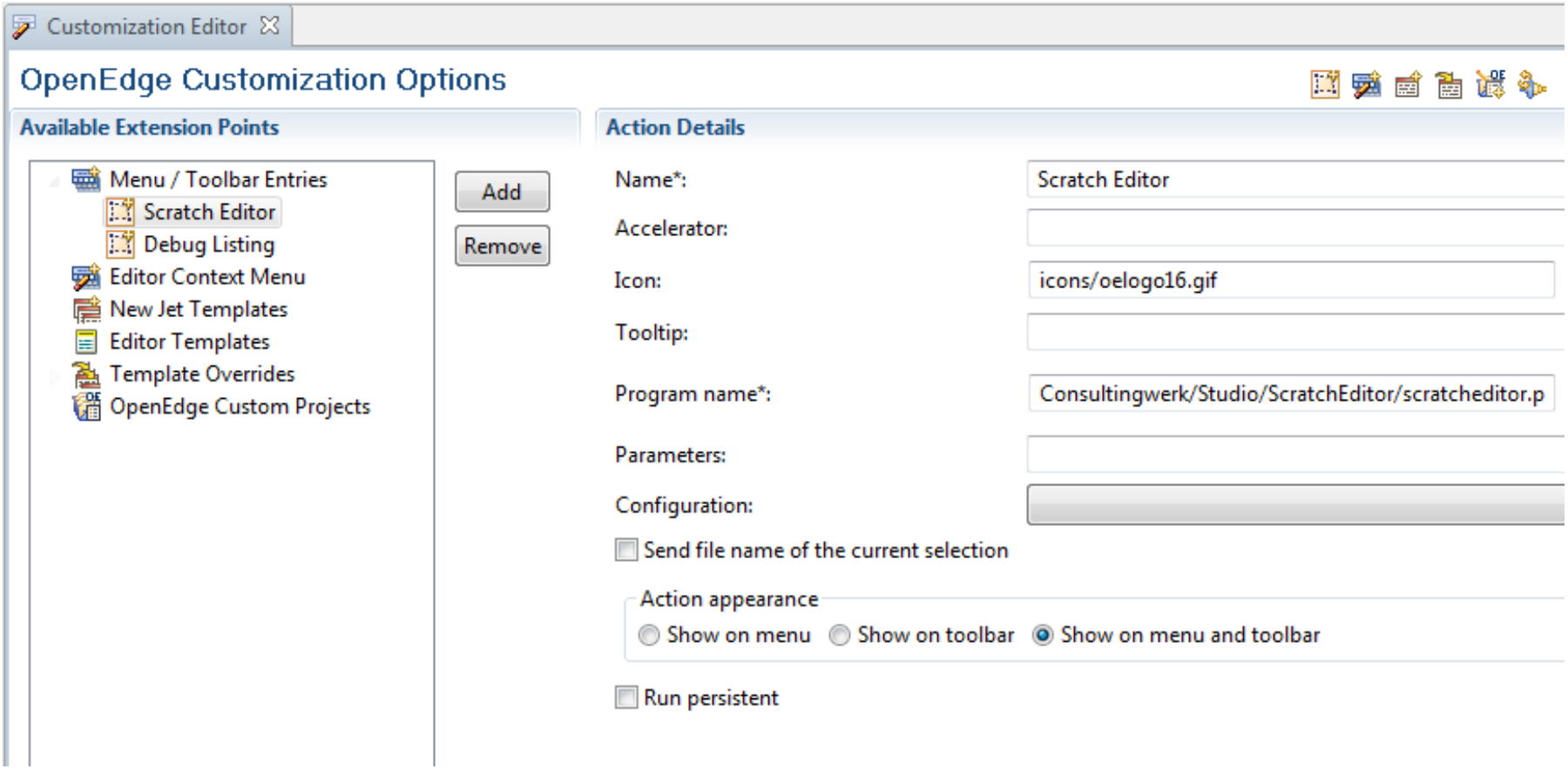
Function prototype defined in oeideservice.i

ABL Scratchpad

- There is finally an ABL scratch pad in 11.2
- But it's a view, not an Editor – so here's my workaround that also works on 10.2B
- Basically creates a copy of an empty file

- Could also create a (dynamic) copy of a file with content

ABL Scratchpad



```
/* ***************************  Definitions  ***************************** */

ROUTINE-LEVEL ON ERROR UNDO, THROW.

{adecomm/oeideservice.i}

/* ***************************  Main Block  ***************************** */

FILE-INFO:FILE-NAME = "Consultingwerk/Studio/ScratchEditor/empty.p":U .

IF FILE-INFO:FULL-PATHNAME > "":U THEN .
ELSE DO:
    MESSAGE "Unable to find scratch editor template empty.p"
        VIEW-AS ALERT-BOX ERROR .

    RETURN .
END.

openEditor
     (?,
      FILE-INFO:FULL-PATHNAME,
      "UNTITLED":U,
      ?) .
```

Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

JET Templates

- Java Emitter Templates
- Dynamic Template format
- Text files that are converted into Java source
- Compiled Java classes executed when you create a new file or new block of code
- http://www.eclipse.org/articles/Article-JET/jet_tutorial1.html
- http://www.eclipse.org/articles/Article-JET2/jet_tutorial2.html

JET Templates

- Anyone remembers SpeedScript (Web Speed)?
- It's similar (html + 4GL → pure 4GL → R-Code)

- Templates based on a skeleton file
  - Java file that contains header of generated Java class
- Typically contains import statements (Java version of USING statement)

JET Template

- Text file
- Contains header

customj.p | class.cjet

```
<%@ jet package="de.consultingwerk.ide.templates"
        imports="com.openedge.core.metadata.ClassInfo com.openedge.co
        class="CustomClassTemplate"
        skeleton="generator.skeleton"%>
```

JET Escape sequences

- Escape sequence for a single Java expression (value)

```
<%= data.getClassName() %>
```

- Escape sequence for a block of Java statements

```
<%
for (Iterator iter = fields.iterator(); iter.hasNext();)
{           Field field = (Field)
            iter.next();%><%=OETemplateUtil.generateField(field, NL)%>
```

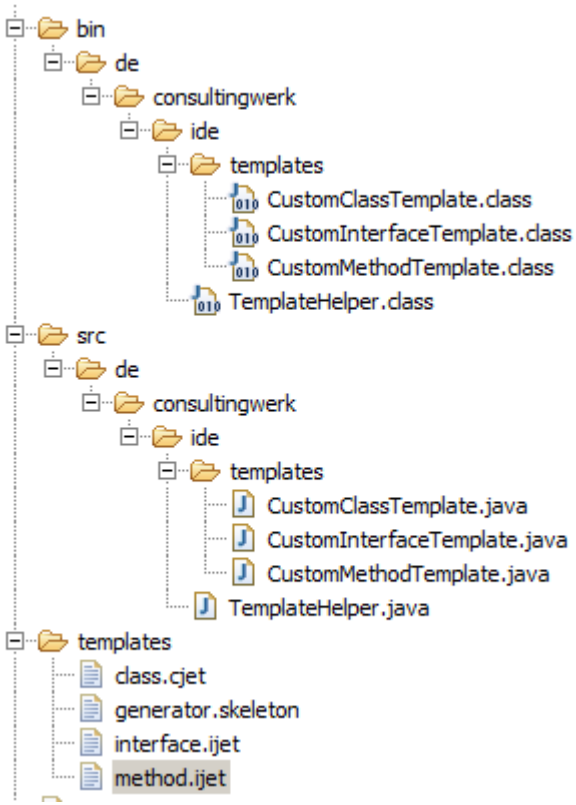
- Can be nested, just like normal Java code

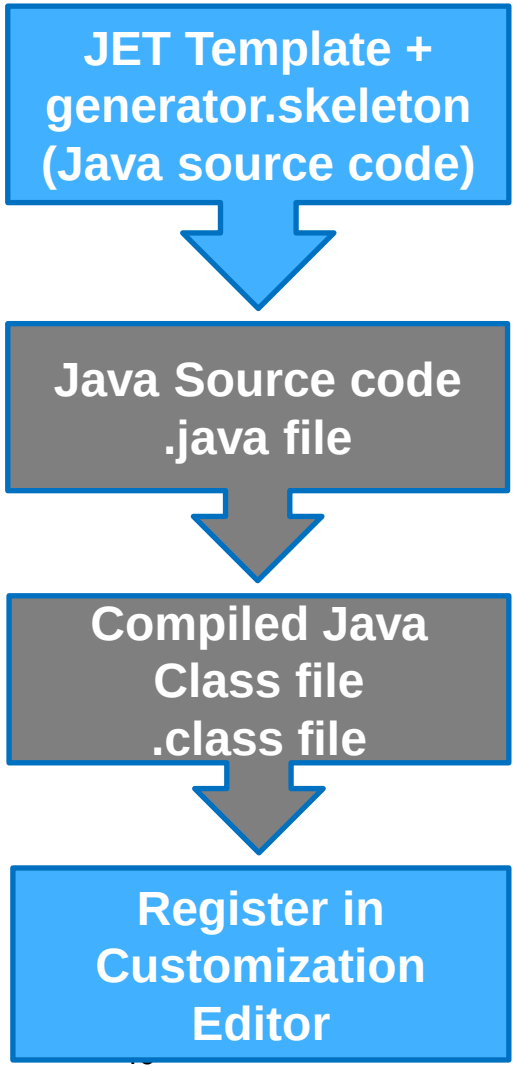
JET Templates

- Jet Templates are executed with an input parameter
- That‘s the input from the Wizard that executes them
- Data in input parameter dependent on the type of template (class, interface, method, etc.)

Template generation sequence

- Sequence executed whenever source file is modified





Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

Setting up JET for OpenEdge

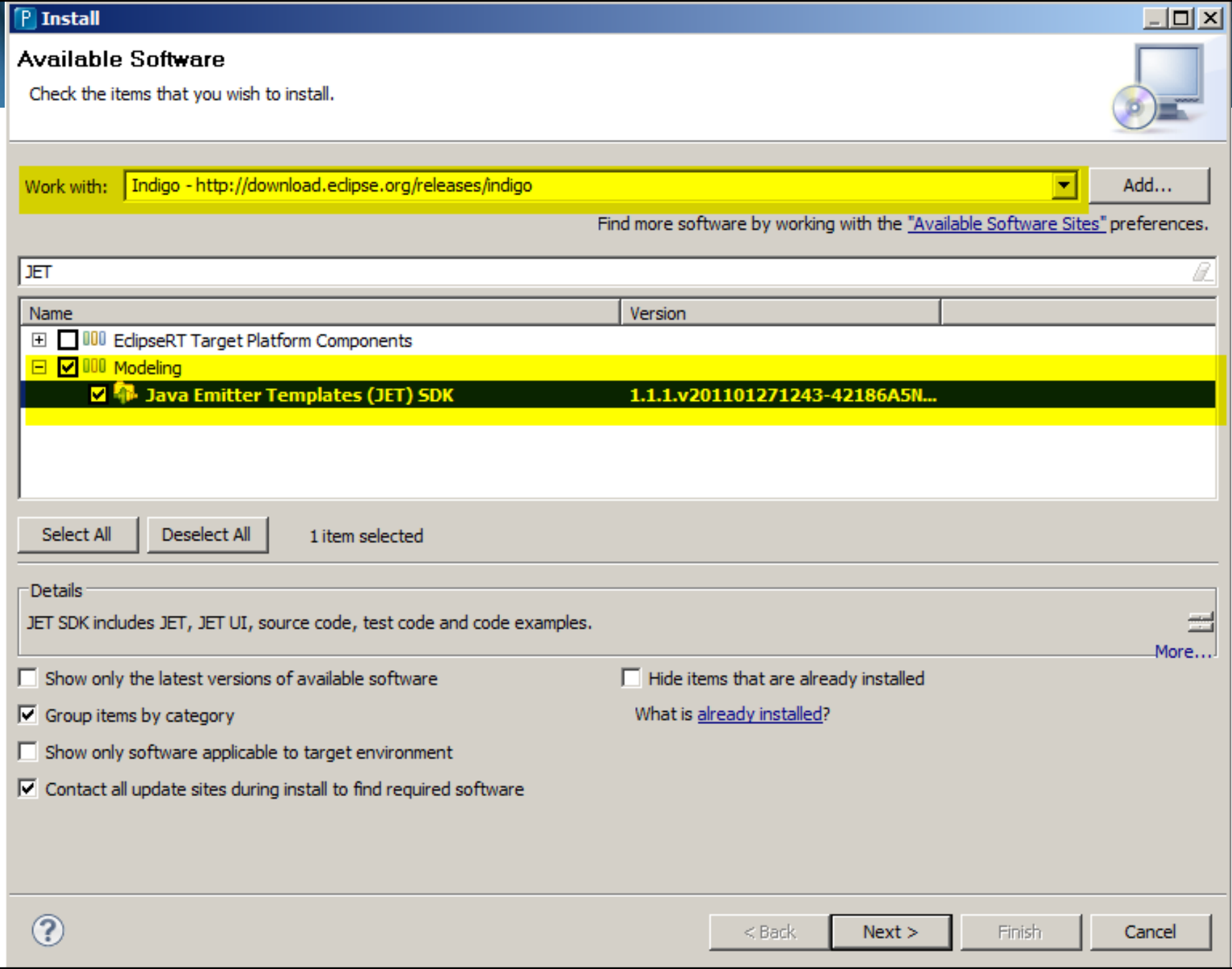
- http://communities.progress.com/pcom/docs/DOC-16387
- Documentation a bit outdated (10.1B)
- Works with 10.2B with a little tweaking and creativity
- Trivial to use on 11.2
- Use existing templates as a starting point, create copies of files in DLC\oeide\eclipse\plugins\com.openedge.pdt.text_*version*\templates

Required Plugins

- Writing custom JET Templates requires the following Eclipse Plugins/Frameworks:
  - JDT – Java Development Tools
  - EMF – Eclipse Modeling Framework
- Use version appropriate for your OpenEdge Architect installation … (Eclipse version used)
- Templates created with OE10.2B work fine in newer versions of PDSOE
- Templates can be used by others without JDT and EMF

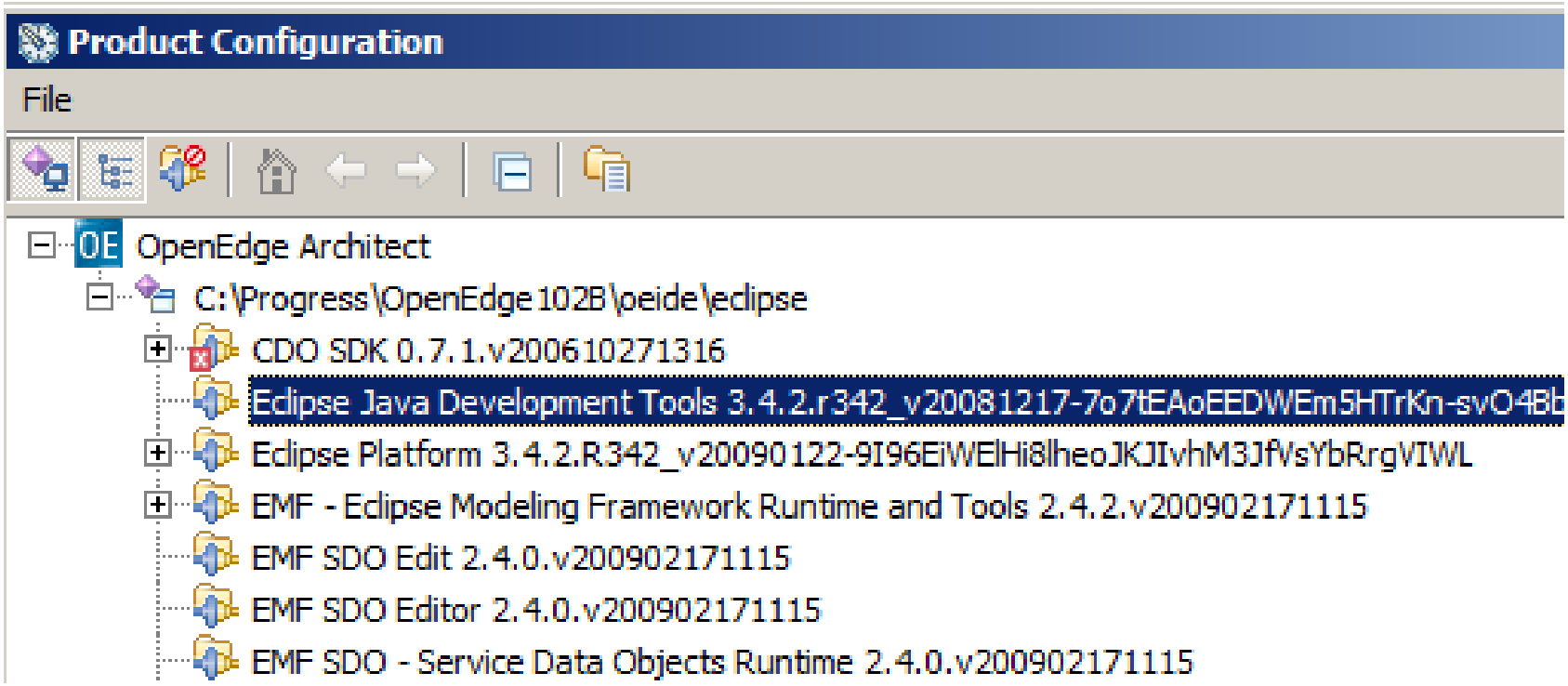
Required plugins

- Plugins can be installed into OEA/PDSOE using Software Updates
- PDSOE in 11.2 is based on Eclipse Indigo which is Eclipse 3.7
- PDSEO in 11.0, 11.1 is based on Eclipse Helios which is Eclipse 3.6
- OEA in 10.2B is based on Eclipse Ganymede which is 3.4
- JET and EMF preinstalled in 11.2, only need to install the JET SDK


Install
Available Software
Check the items that you wish to install.
Work with:
Indigo - http://download.eclipse.org/releases/indigo
Add...
Find more software by working with the "Available Software Sites" preferences.
JET
Name
Version
EclipseRT Target Platform Components
Modeling
Java Emitter Templates (JET) SDK
1.1.1.v201101271243-42186A5N...
Select All
Deselect All
1 item selected
Details
JET SDK includes JET, JET UI, source code, test code and code examples.
More...
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

Eclipse Plugins

- Eclipse Plugins for my 10.2B installation



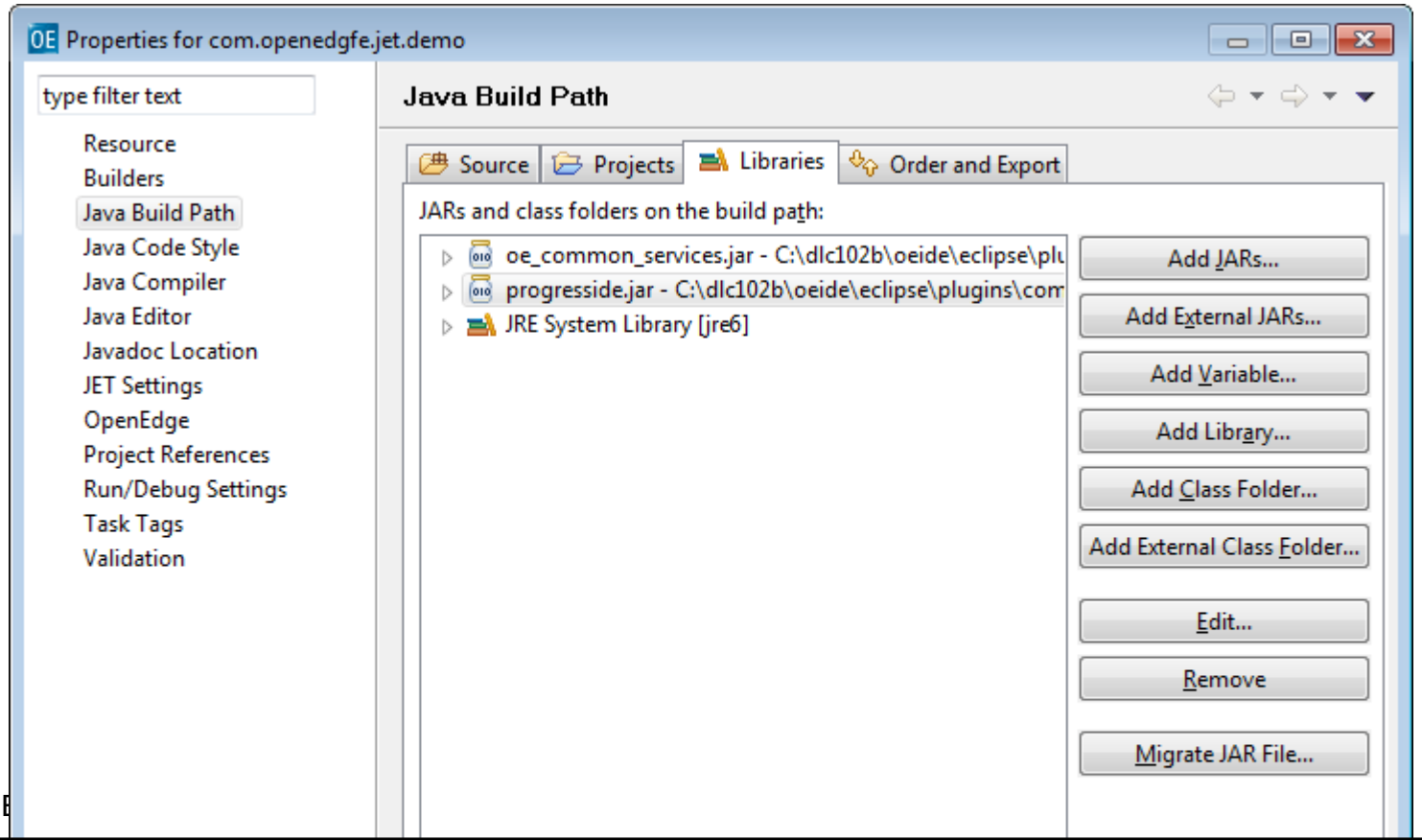
Setting the Java Build Path

- Add the following two external JAR files for OpenEdge 11.2:

- C:\Progress\DeveloperStudio3.7\eclipse\plugins\com.openedge.pdt.core_11.2.1.00**oe_common_services.jar**
- C:\Progress\DeveloperStudio3.7\eclipse\plugins\com.openedge.pdt.text_11.2.1.00**progresside.jar**

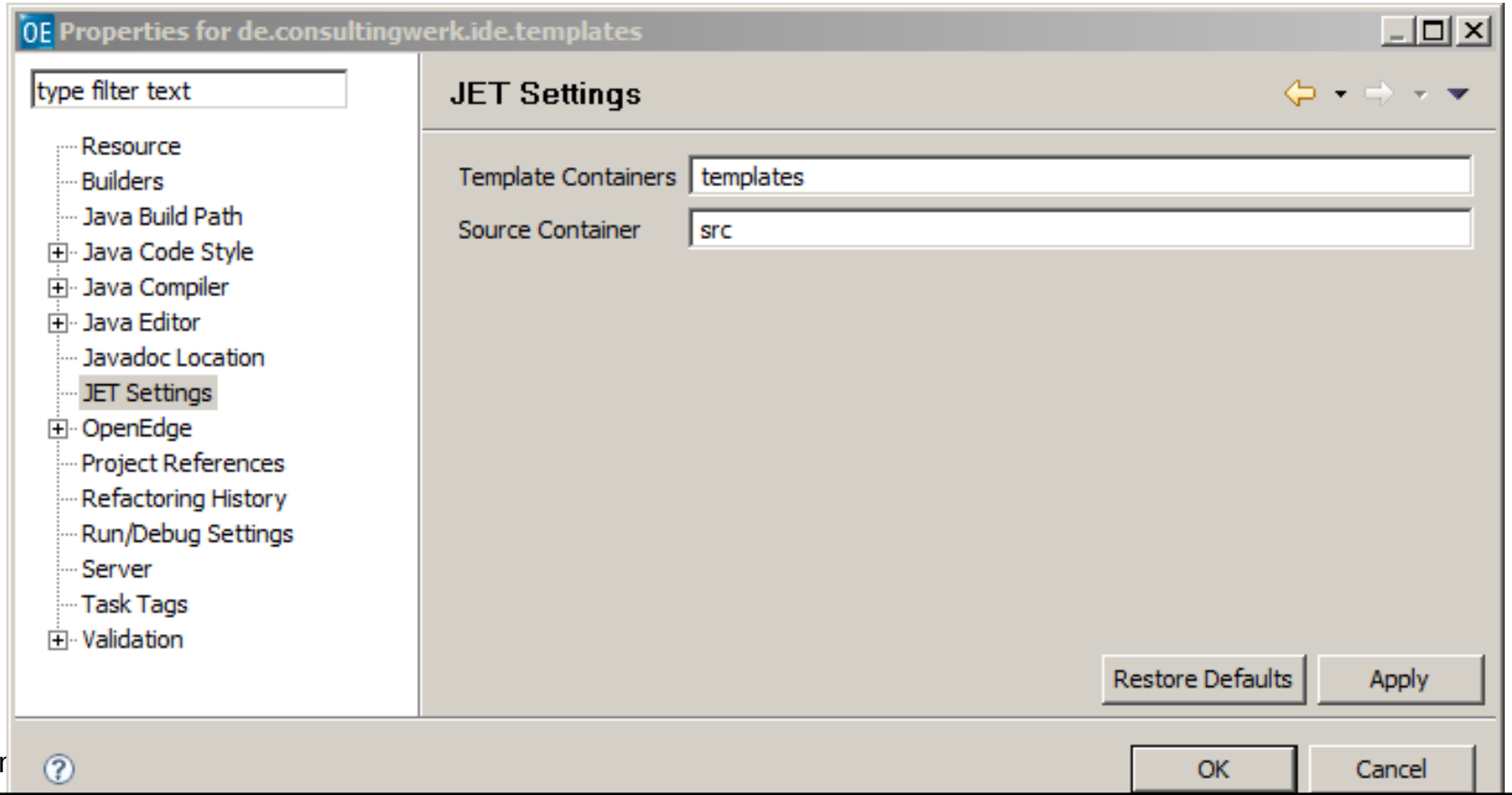
Setting the Java Build Path

- Java Build Path from white paper needs additional .jar file in 10.2B/11.x: **progresside.jar**



Editing .jetproperties

- In 10.2B with the plugins installed the JET properties are part of the project properties



Alternatively editing .jetproperties file

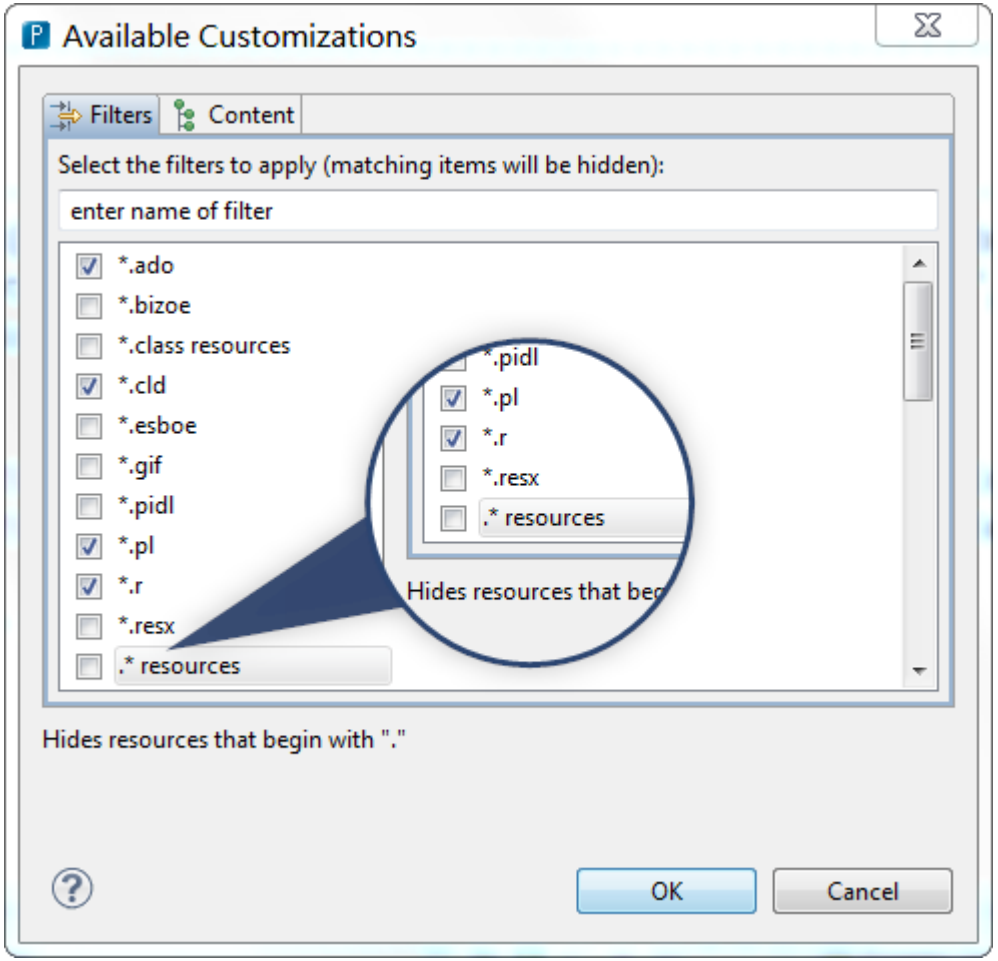
- Text/XML file in project root
- Change source container setting to: src
- Restart PDSOE when done

.jetproperties

```
<?xml version="1.0" encoding="UTF-8"?>
<jet-settings>
    <template-container>templates</template-container>
    <source-container>src</source-container>
</jet-settings>
```

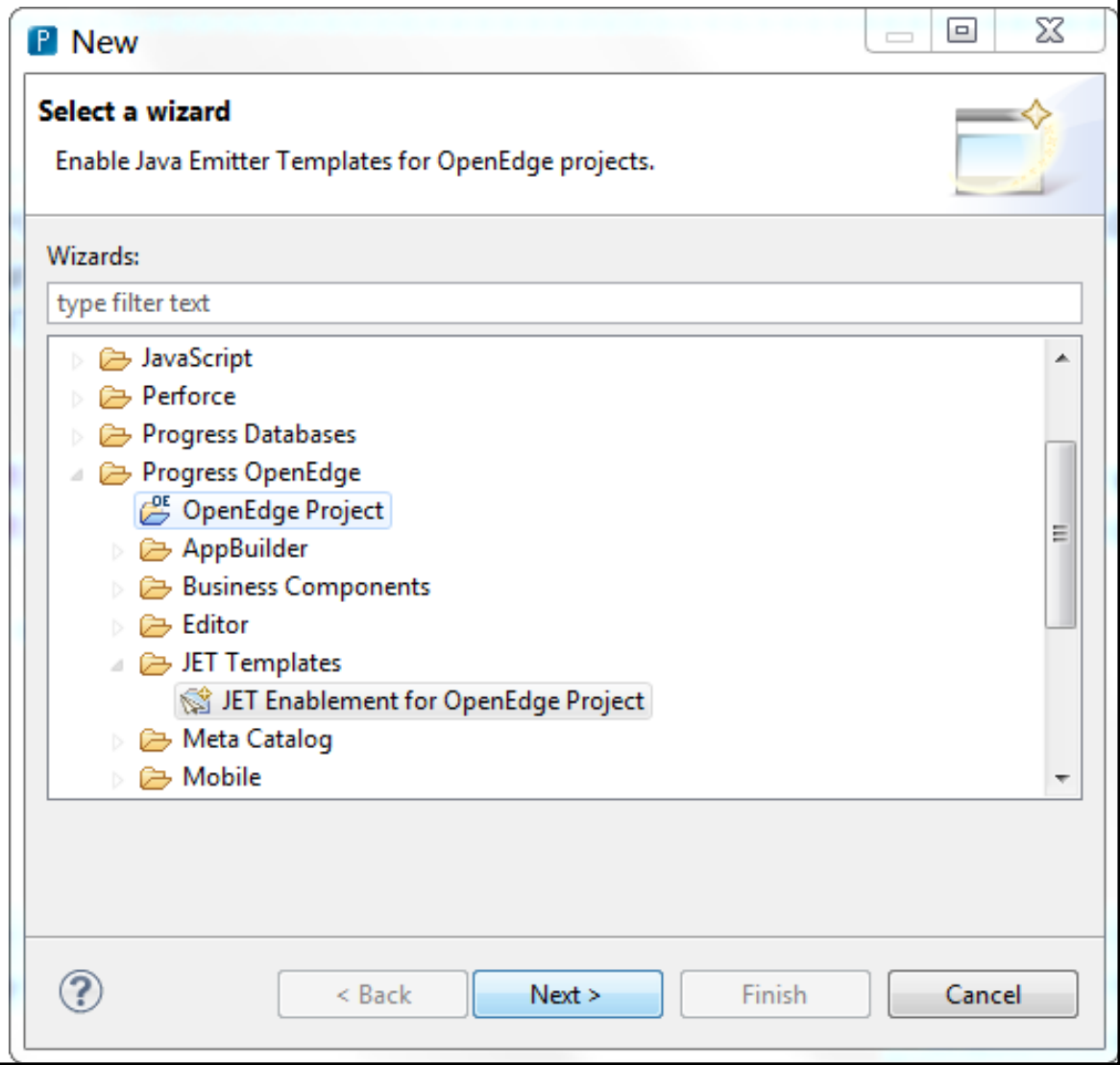
Alternatively editing .jetproperties file

- . resources are hidden in the resource view by default
- Resource view context menu -> Customize View



Enable JET for OpenEdge Project

- File -> New



Agenda

- A (short) Review of OpenEdge Architect
- (Personal) Preferences in OpenEdge Architect
- OpenEdge Editor Macros
- Setting the Author Name used by Templates
- Progress Customization Options
- JET Templates
- Installing Plugins for JET in OEA/PDSOE
- Customizing OpenEdge Architect Templates

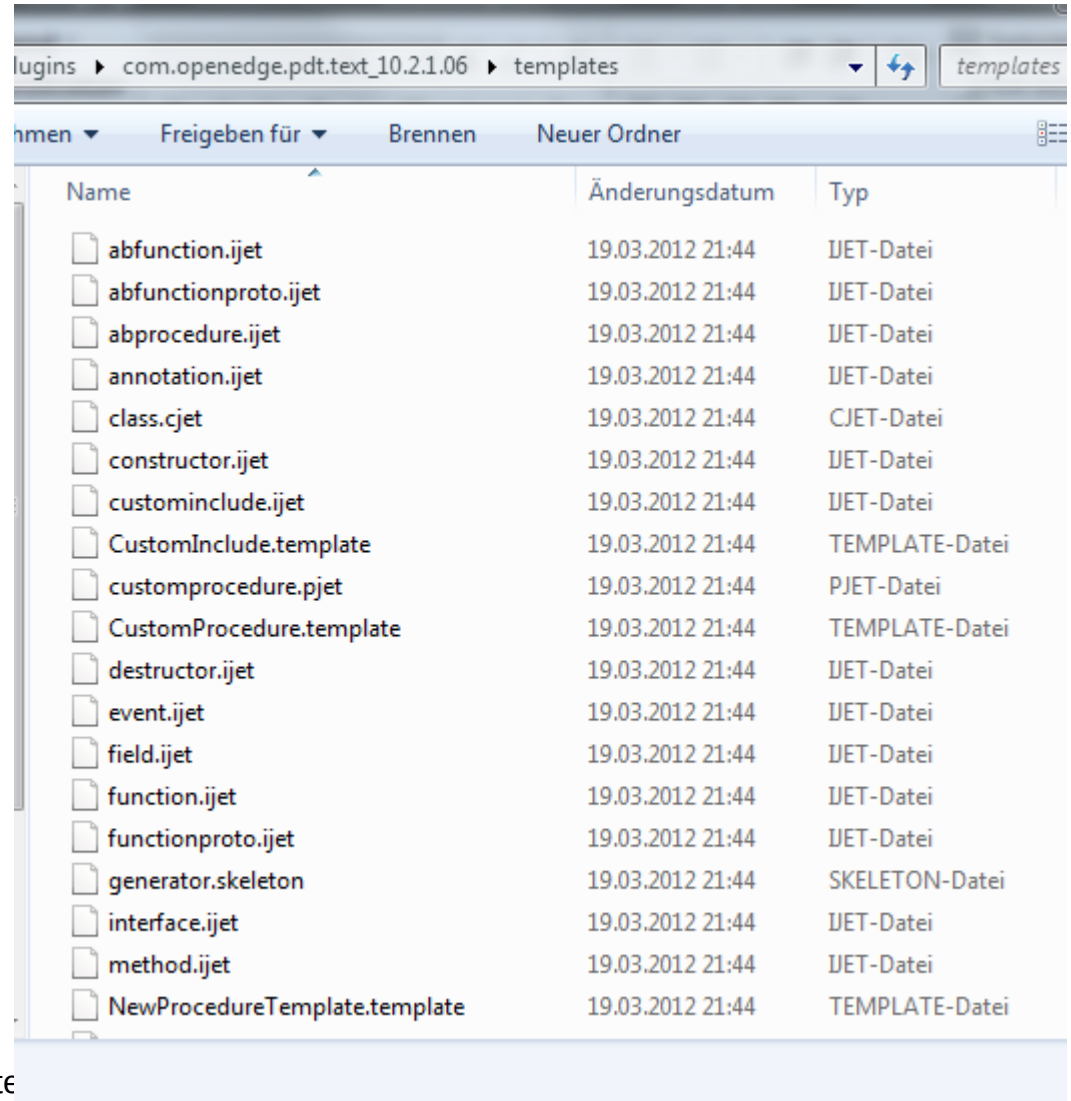
Customizing Templates for OEA/PDSOE

- OEA/PDSOE templates based on JET
- Overriding existing templates for new files, e.g. modifying the file header comment
- Overriding existing templates for new code blocks, e.g. new internal procedure wizard, new method, new property
- Additional new file templates
  - Currently only supported for .p file extension (functional limitation)

Customizing Templates for OEA/PDSOE

- Typically custom templates based on Progress supplied templates
  - C:\Progress\DeveloperStudio3.7\eclipse\plugins**com.openedge.pdt.text_11.2.1.00**\templates
  - C:\Progress\OpenEdge102B\oeide\eclipse\plugins**com.openedge.pdt.text_10.2.1.06**\templates
- Contains also the skeleton file

OpenEdge Architect Templates



Modifying templates

- Optionally create custom skeleton file, that provides standard includes etc.
- Set custom class name (template name) in <%@ header
- Set custom package name in the <%@ header

- Start writing template ☺

customj.p | class.cjet | generator.skeleton

```
<%@ jet package="de.consultingwerk.ide.templates"
                imports="com.openedge.core.metadata.ClassInfo com.openedg
                class="CustomClassTemplate"
                skeleton="generator.skeleton"%>
 <%
    ClassInfo data = (ClassInfo) argument.getObj("wizardinfo");

    Set events = data.getEvents();
    List methods = data.getMethods();
    Set properties = data.getProperties();
    Set fields = data.getFields();
    Set constructors = data.getConstructors();
    Destructor destructor = data.getDestructor(
    boolean ldAtBeginning = true;
    boolean ldAtEnd = true;
    boolean lIsEnum = (data.getSuperClassName().equals("Consult ngwerk.Enu
 %>
/*********************************************************************
 * Copyright (C) 2006-2013 by Consultingwerk Ltd. ("CW") -           *
 * www.consultingwerk.de and other contributors as listed            *
 * below.  All Rights Reserved.                                      *
 *                                                                   *
 *  Software is distributed on an "AS IS", WITHOUT WARRANTY OF ANY   *
 *   KIND, either express or implied.                                *
 *                                                                   *
 *  Contributors:                                                    *
 *                                                                   *
 *********************************************************************/
```

Non-escaped text will just show up in the template

Sample „Enum“ template

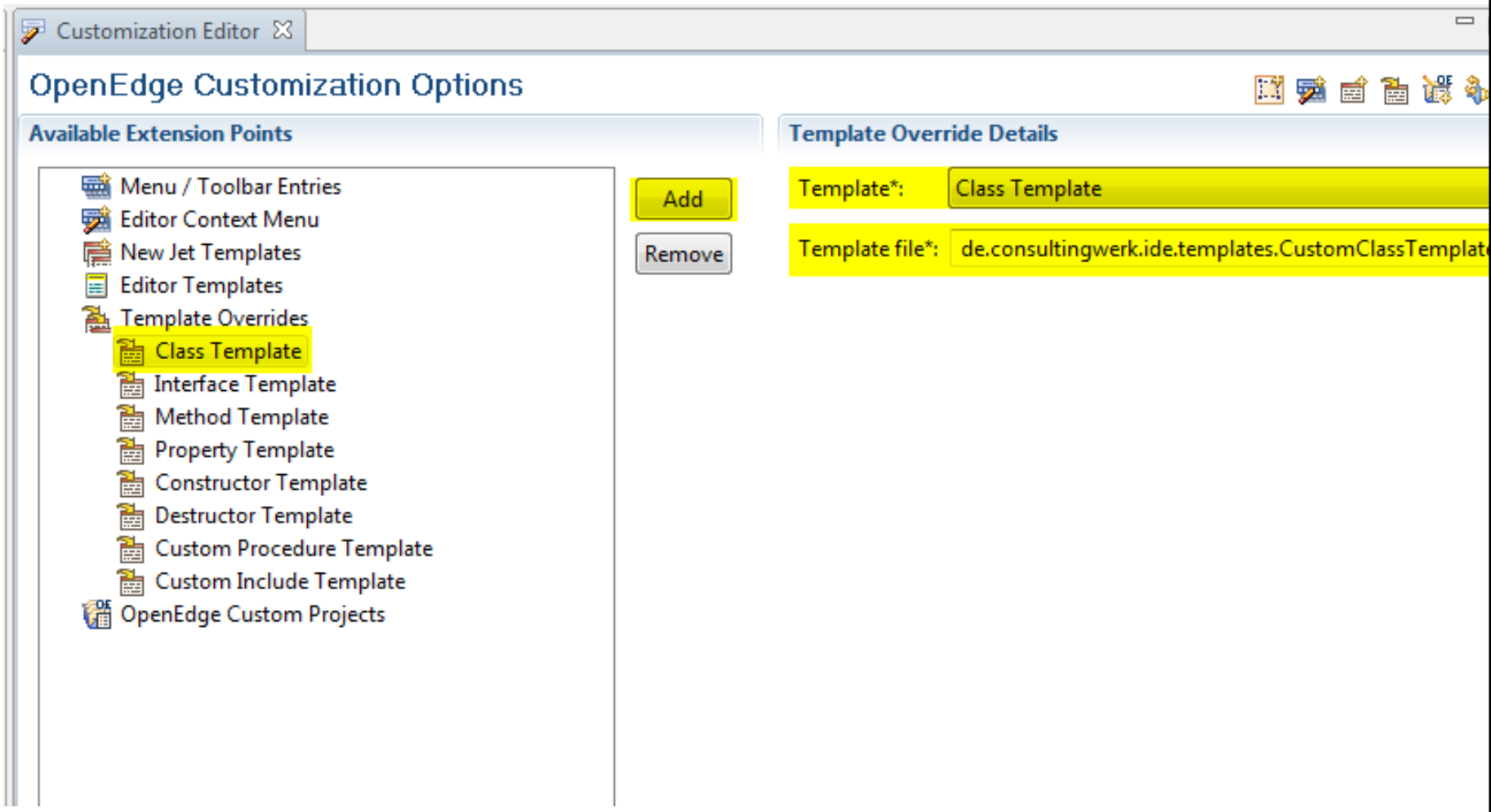
- Based on the class template
- Generate additional code based on Super-Class

```
<%
   ClassInfo data = (ClassInfo) argument.getObj("wizardinfo");

   Set events = data.getEvents();
   List methods = data.getMethods();
   Set properties = data.getProperties();
   Set fields = data.getFields();
   Set constructors = data.getConstructors();
   Destructor destructor = data.getDestructor();
   boolean ldAtBeginning = true;
   boolean ldAtEnd = true;
   boolean lIsEnum = (data.getSuperClassName().equals("Consultingwerk.Enum"));
%>
```

```
<%= lIsEnum ? "    {Consultingwerk/EnumMember.i FirstValue 0 " + data.getClassName() + "}"
<%= lIsEnum ? " " + NL : "" %><%= lIsEnum ? "    /*----------------------------------------
```

Registering your custom templates

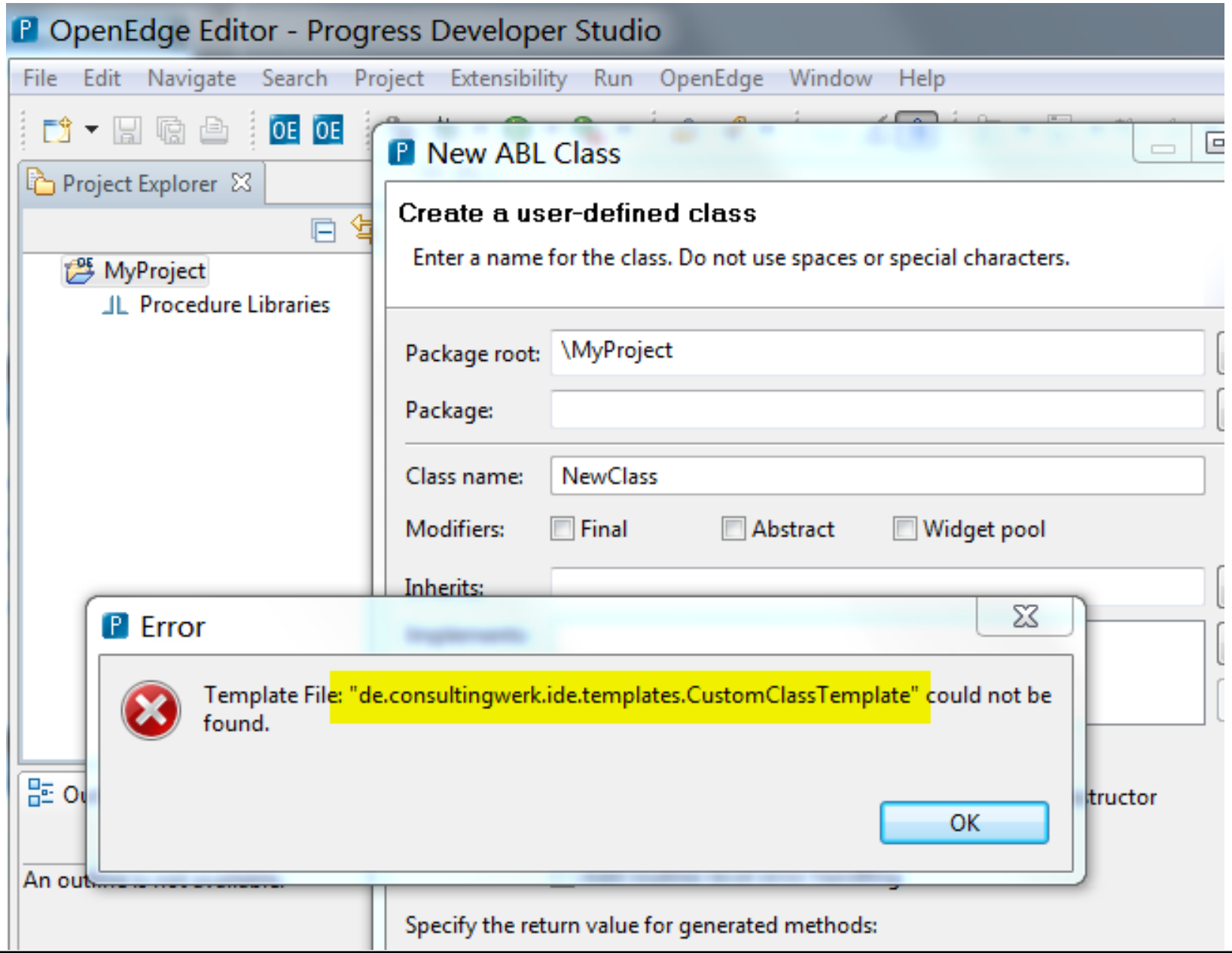


Demo

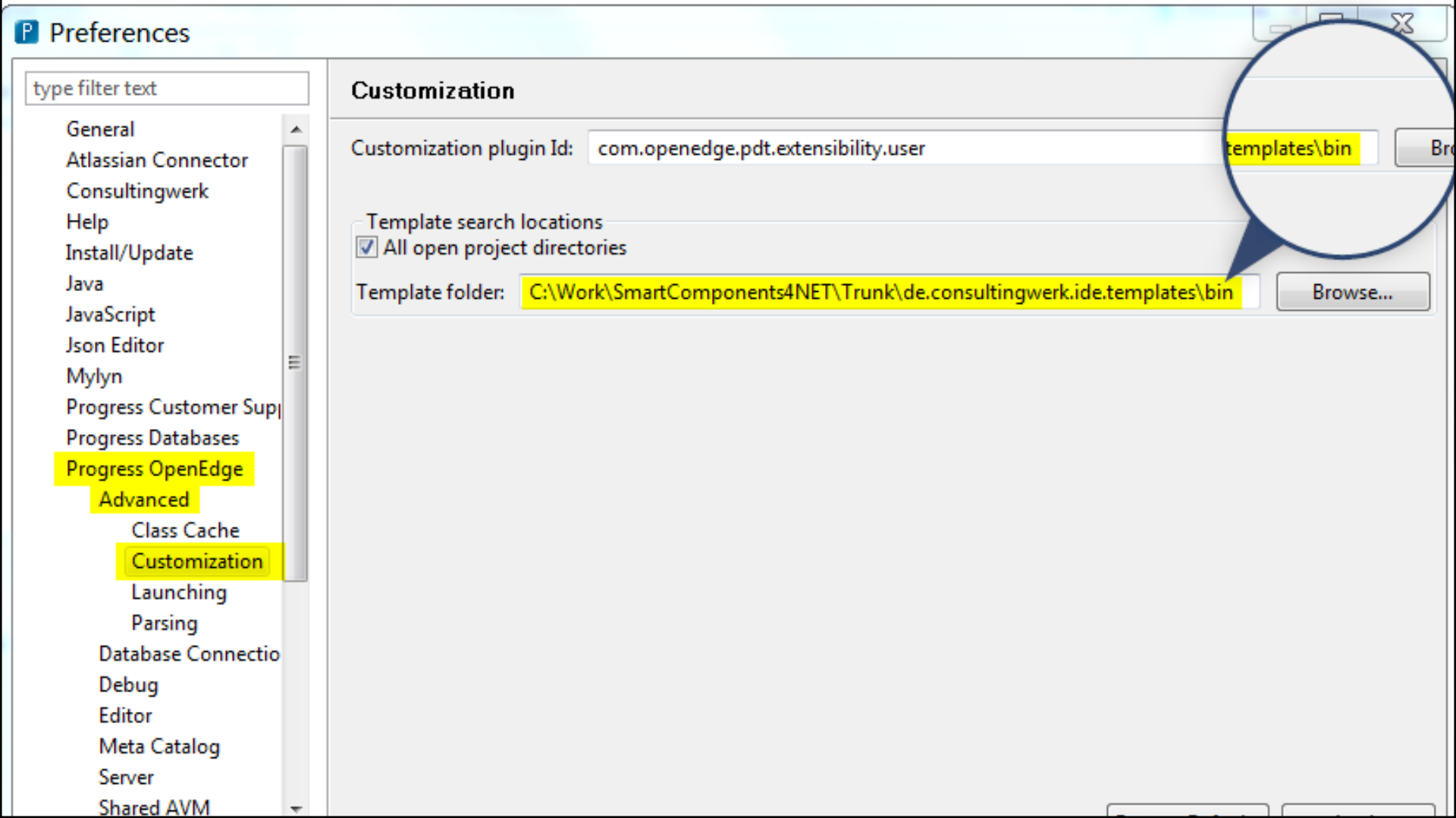
- Template customization environment
- Customized method template
- Customized new class template
- Review Java functions
- Review code in JET Template that checks for class properties
- Review usage of custom Templates in OpenEdge Architect

Customization settings

- Settings are persisted per OEA/PDSOE installation, not per workspace
- No need to re-customize in every workspace!
- When registering custom templates, they need to be made available in every workspace
  - Path to custom templates is set in the workspace (see later)


OpenEdge Editor - Progress Developer Studio
File Edit Navigate Search Project Extensibility Run OpenEdge Window Help
Project Explorer
MyProject
Procedure Libraries
New ABL Class
Create a user-defined class
Enter a name for the class. Do not use spaces or special characters.
Package root: \MyProject
Package:
Class name: NewClass
Modifiers: Final Abstract Widget pool
Inherits:
Error
Template File: "de.consultingwerk.ide.templates.CustomClassTemplate" could not be found.
OK
Specify the return value for generated methods:

Template binary location in Preferences



Using templates on other workspaces

- Just provide the contents of your bin folder
- I prefer the SCM tool for that

More info…

- Getting started videos
  http://communities.progress.com/pcom/docs/DOC-101140
- Top OpenEdge Architect FAQ, Sunil Belgaonkar
  http://communities.progress.com/pcom/community/psdn/exchange/trackprod6
- Consultingwerk blog – openedge-architect tag
  http://blog.consultingwerk.de/consultingwerkblog/tag/openedge-architect/

Questions

Thank you!