

Visual Studio Code for everyday ABL development

Peter Judge, Consultingwerk



Peter Judge

- Senior Architect at Consultingwerk
- Writing 4GL since 1996, working on a variety of frameworks and applications. More recently have worked on a lot of integration-y stuff: Authentication Gateway, HTTP Client, Web Handlers. Dabble in PASOE migrations.
- Active participator in Progress communities, PUGs and other events



Consultingwerk Software Services Ltd.

- Independent IT consulting organization
- Focusing on **OpenEdge** and **related technology**
- Located in Cologne, Germany, subsidiaries in UK, USA and Romania
- Customers in Europe, North America, Australia and South Africa
- Vendor of developer tools and consulting services
- Specialized in GUI for .NET, Angular, OO, Software Architecture, Application Integration
- Experts in OpenEdge Application Modernization



Services Portfolio, Progress Software

- OpenEdge (ABL, Developer Tools, Database, PASOE, ...)
- Telerik DevCraft (.NET, Kendo UI, Angular, ...), Telerik Reporting
- OpenEdge UltraControls (Infragistics .NET)
- Telerik Sitefinity CMS (incl. integration with OpenEdge applications)
- Kinvey Plattform, NativeScript
- Corticon BRMS
- WhatsUp Gold infrastructure-, network- and application monitoring
- Kemp Loadmaster
- ...

Services Portfolio, related products

- Protop Database Monitoring
- Combit List & Label
- Web frameworks, e.g. Angular
- .NET
- Java
- ElasticSearch, Lucene
- Amazon AWS, Azure
- DevOps, Docker, Jenkins, ANT, Gradle, JIRA, ...
- ...

Agenda

- Intro to Visual Studio Code
- Extensions
- Snippets and templates



What is Visual Studio Code (VS Code)?



- Free (as in beer and as in code) open-source code editor available on Windows, Linux, MacOS
- Core functionality extended via extensions, available through a central repository
- First-class support for SCM tools (Git, Perforce, etc)
- Remote access using SSH, WSL on Windows, Docker
- Flexible and powerful settings storage and sharing
- History
 - Visual Studio Code was first announced on April 29, 2015 by Microsoft at the 2015 Build conference
 - Public release on April 14, 2016
 - Language server protocol defined as part of VSCode

<https://code.visualstudio.com/>

Search files by name (append : to go to line or @ to go to symbol)

Command palette

Go to File Ctrl + P

Show and Run Commands > Ctrl + Shift + P

Search for Text %

Go to Symbol in Editor @ Ctrl + Shift + O

Start Debugging debug

Run Task task

More ?

recently opened

- ErrorHelper.cls ABL • Consultingwerk\Util
- HttpStatus.cls ABL • Consultingwerk\Util
- FileHelper.cls ABL • Consultingwerk\Util
- test-read-xml.p ABL • Test\SCL4599
- openedge-project.json ABL
- SmartObjectInstanceViewerControl.cls ABL • Consultingwerk\Windows\Framework\Repository\Ob...
- oeablSecurity.properties smartpas_stream • conf
- InternalWebRouter.cls C:\Work_STREAM\SmartComponentLibrary\Develop128\ADE Sourcecode\src...
- domain.csv smartpas_stream
- CustomerBusinessEntity.cls ABL • Consultingwerk\SmartComponentLibrary\OERA\Src\src\2000

Editor

```
21
22
23 @lowercase.
24
25 block-level-on-error-undo, -throw.
26
27 {-Consultingwerk/products.i-}
28
29 using Consultingwerk.* ..... from propath..
30 using Consultingwerk.Assertion.* ..... from propath..
31 using Consultingwerk.Exceptions.* ..... from propath..
32 using Consultingwerk.Framework.* ..... from propath..
33 using Consultingwerk.Framework.Collections.* from propath..
34 using Consultingwerk.Util.* ..... from propath..
35 using Consultingwerk.Util.Forms.* ..... from propath..
36 &if-not-proversion-begins "10.2" &then
37 using Progress.Json.ObjectModel.* ..... from propath..
38 &endif
39 using Progress.Lang.* ..... from propath..
40
41 class Consultingwerk Util ErrorHelper
```

Panel

Status bar

PROBLEMS 147 OUTPUT TEST RESULTS TERMINAL

c:\Work_STREAM\SmartComponentLibrary\Develop128\ABL>

Activity bar

Side bar

EXPLORER

OUTLINE

- Consultingwerk.Util.ErrorHelper Consultingwerk.Util.ErrorHelper
 - AddErrorMessage(1CHAR pcMessage, 1CHAR pcText, 1CHAR pcFi
 - AddErrorMessage(1CLOB pcMessage, 1CHAR pcText, 1CHAR pcFi
 - AddJsonErrorDetails(1Progress.Lang.Error poError, 1Progress.Json
 - AddJsonErrorDetails(1Progress.Lang.Stop poStopCondition, 1Prog
 - cCallerStackTrace : CHARACTER
 - CurrentStacktrace : CHARACTER
 - DatasetErrorStrings(1HDL phDataset) : CHARACTER
 - DeRegisterErrorHandler(1Consultingwerk.Framework.IErrorHandle
 - DeRegisterStopConditionHandler(1Consultingwerk.Framework.ISt
 - ErrorHelper() : VOID
 - ErrorHelper() : VOID
 - ErrorMessageDialogAllowed : LOGICAL
 - ErrorMessageFormType : CHARACTER
 - ErrorTitle(1Progress.Lang.Error poError) : CHARACTER

DEVELOP128 (WORKSPACE)

- ABL
 - Consultingwerk
 - Assertion
 - BusinessEntityDesigner
 - CommonUi
 - Design
 - Exceptions
 - Forms
 - Framework
 - iOSAdapter
 - OERA
 - SmartComponents
 - SmartComponentsDemo
 - SmartFramework
 - SmartUnit
 - Studio
 - Templates
 - Util
- TASK EXPLORER
- TIMELINE
- ANT TARGET RUNNER
- OPENEDGE DOCUMENTS
- SONARLINT ISSUES

Workspaces

- Multi-root workspaces allow >1 folder as a logical group
 - File with `.code-workspace` extension
 - Should be checked in to SCM
- Can have common settings, tasks and launch configurations

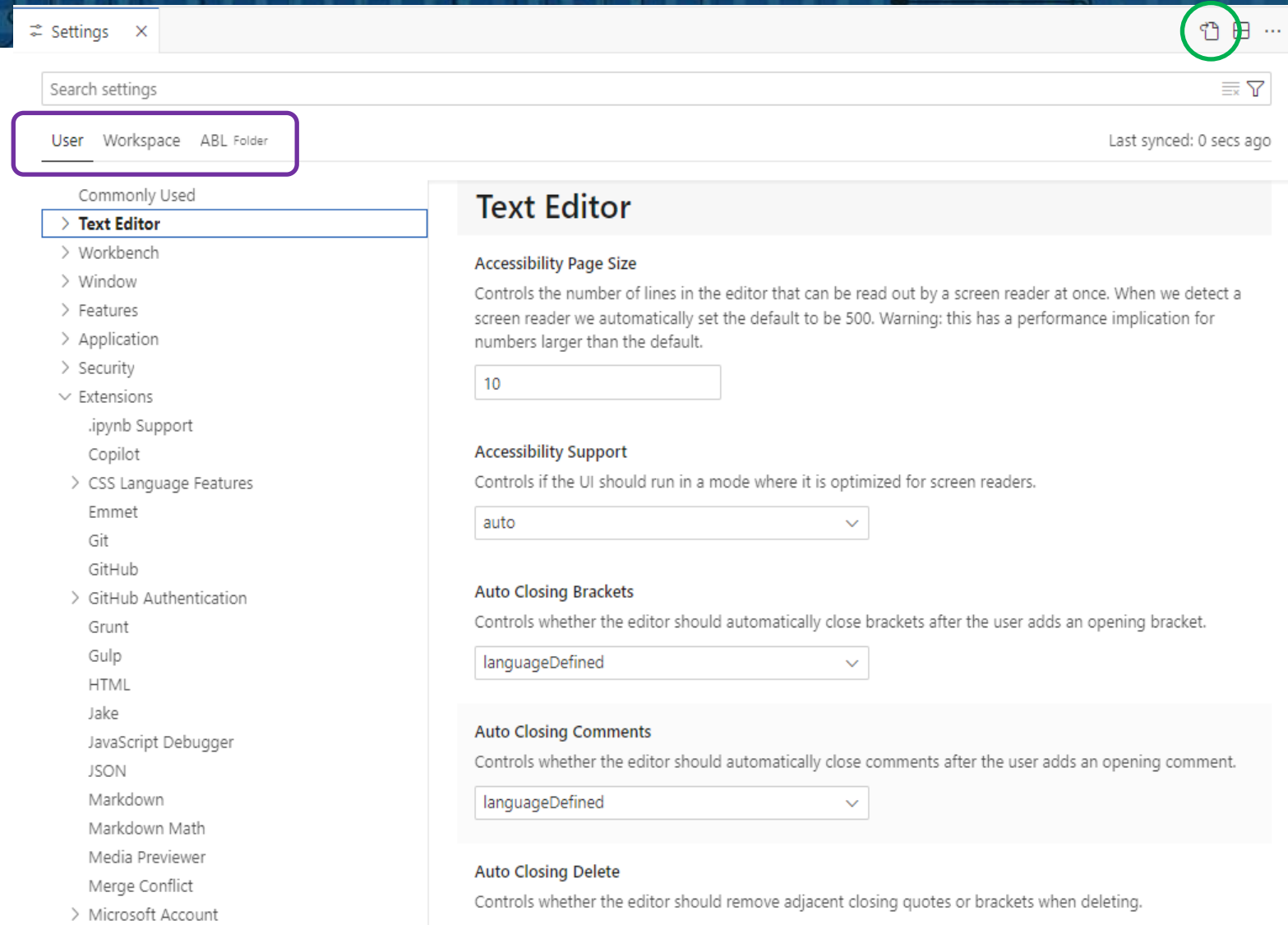
```

{
  "folders": [
    { "name": "SmartComponentLibrary", "path": "ABL" },
    { "name": "Scratch", "path": "Scratch" },
    { "name": "UnitTests", "path": "UnitTests" },
    { "name": "Templates", "path": "Templates" },
    { "name": "DB", "path": "DB" },
    { "name": "Docker", "path": "Docker" }
  ],
  "settings": {
    "abl.configuration.runtimes": [...],
    "files.exclude": {...},
    "files.encoding": "iso88591",
    "files.trimTrailingWhitespace": true,
    "files.associations": {...},
    "cSpell.words": [],
    "cSpell.ignoreWords": [],
    "editor.formatOnPaste": true,
    "diffEditor.codeLens": true,
    "abl.defaultProject": "SmartComponentLibrary",
    "templates.folder": "./Templates",
    "ant.home": "C:/Progress/OpenEdge128_64/ant",
    "ant.buildFileNames": "annotations.xml,build.xml",
    "ant.executable": "ant -lib c:/Progress/OpenEdge128_64/PCT/pct.jar",
    "terminal.integrated.enablePersistentSessions": true
  },
  "extensions": {
    "recommendations": [...]
  }
}

```

Settings

- Settings saved in 1 of 3 locations, as JSON
- User settings can be synchronized across multiple machines
- Storage
 - Windows
`%APPDATA%\Code\User\settings.json`
 - Linux
`$HOME/.config/Code/User/settings.json`




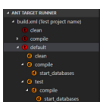





Agenda






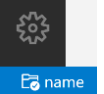

- Intro to Visual Studio Code
- Extensions
- Snippets and templates



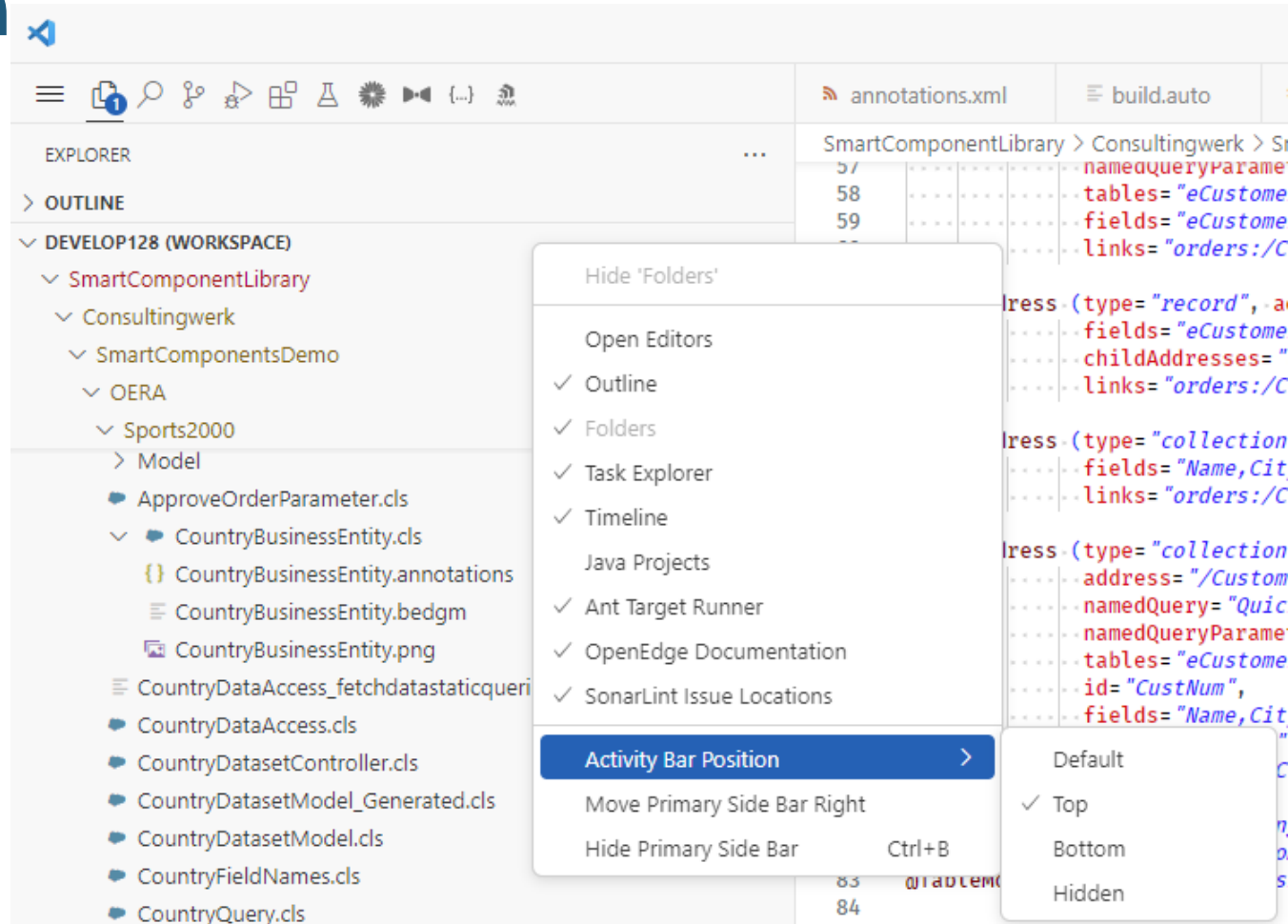
Extensions - ABL

| | | | |
|--|---------------------------|--------------------|--|
|  | OpenEdge ABL | Riverside Software | Syntax highlighting, compilation, debugging, outline views, documentation, code completion and much more |
|  | ABL Developer Pack (meta) | Carl Verbiest | Bundle of ABL extensions |
|  | ABLUnit Test Runner | Ken Herring | Runs ABLUnit unit tests |
|  | Ant Target Runner | Nick Heap | Runs Ant targets (eg PCT tasks/targets) |
|  | Pro Peek | Baltic Amadeus | Visualizes ABL profiler output |
|  | Pro Bro | Baltic Amadeus | Database table, field viewer |
|  | SonarLint for ABL | Riverside Software | Connects ABL extension with a SonarCube server |

Extensions – a few others

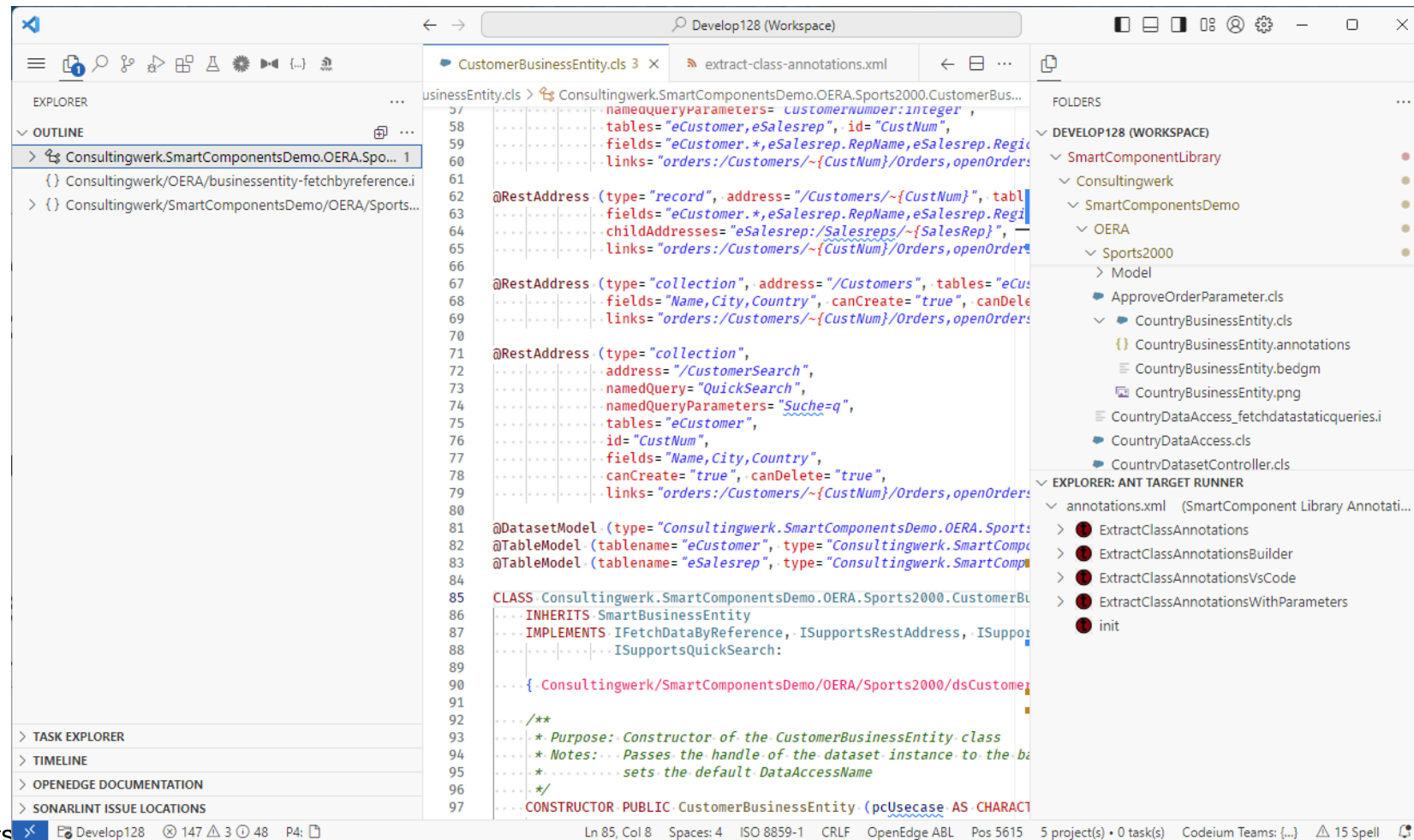
| | | |
|---|------------------------------|--|
|  | Code Spell | Spellchecker |
|  | File Templates | Templates for new files (ie classes, enums, etc) |
|  | Jenkins Jack | View running and historical Jenkins jobs |
|  | Rainbow CSV | Pretty-print CSV files; format as table |
|  | vscode-position | Position to file offset |
|  | Workspace Name in Status Bar | What it says ... |
|  | Indent-rainbow | Colorful highlights of indentation |

Activity Bar position



Move explorer views

- Move activity bar
 - Top means you lose the settings button



Extensions

- Ant Test Runner: make sure you have an OS-compliant terminal (eg Command Prompt not WSL on Windows)

Some useful settings

```
"editor.autoIndent": "keep",
"editor.formatOnPaste": true,
"editor.fontFamily": "'Fira Code', monospace",
"editor.fontLigatures": true,
"editor.minimap.enabled": false,
"editor.renderWhitespace": "all",
"editor.stickyScroll.enabled": true,
"editor.wordBasedSuggestions": "off",
"explorer.fileNesting.enabled": true,
"explorer.fileNesting.patterns": {"*.cls": "${capture}.annotations"},
"files.exclude"
"files.associations"
"files.trimFinalNewlines": true,
"files.trimTrailingWhitespace": true,
"templates.author": "Peter Judge",
"outline.collapseItems": "alwaysCollapse",
"window.menuBarVisibility": "compact",
```

Keyboard shortcuts

| | |
|-------------------|--|
| F1, SHIFT-CTRL-P | Open command palette |
| SHIFT-CTRL-O | Open 'editor symbols' (methods/functions) |
| CTRL-P | Open file in workspace |
| ALT-SHIFT-<click> | Start block/column editing |
| SHIFT-CTRL-K | Delete line |
| CTRL-O | Open (external) file |
| CTRL-K, CTRL-W | Close all editors |
| | |
| CTRL-K, CTRL-S | See (and configure) all keyboard shortcuts |

Running code

- The ABL extension provides a couple of "run" events
 - ABL: Run with `prowin`
 - ABL: Run with `_progres -b` (keyboard shortcut of F2)

Demo

- Code basics
- Extensions
- Keymaps
- etc

Agenda

- Intro to Visual Studio Code
- Extensions
- Snippets and templates



Tab stops

Standard code : snippets

- Snippets
 - Built-in / native to VS Code
 - Like 'macros' in PDSOE, chunks of code in which you can replace tokens with e.g. your own variable names
 - Can have predefined choices
 - Stored as files in project `.vscode/` folder with `.code-snippets` extension
- Use <https://snippet-generator.app/?description=&tabtrigger=&snippet=&mode=vscode> to generate your own snippets

```
"DV-PR": {
  "prefix": "dvar",
  "body": [
    "define variable
    ${1|c,lc,i,l,h,de,da,dt,ra,ro,m,o|}${2:name} as
    ${3|character,longchar,integer,int64,logical,handle,decimal,date,datetime,datetime-
    tz,raw,rowid,memptr,Progress.Lang.Object|} no-undo."
  ],
  "description": "define variable"
},

"DV-IN": {
  "prefix": "dvin",
  "body": [
    "define variable i${1:name} as integer no-
    undo.${0}"
  ],
  "description": "define variable as integer"
},
```

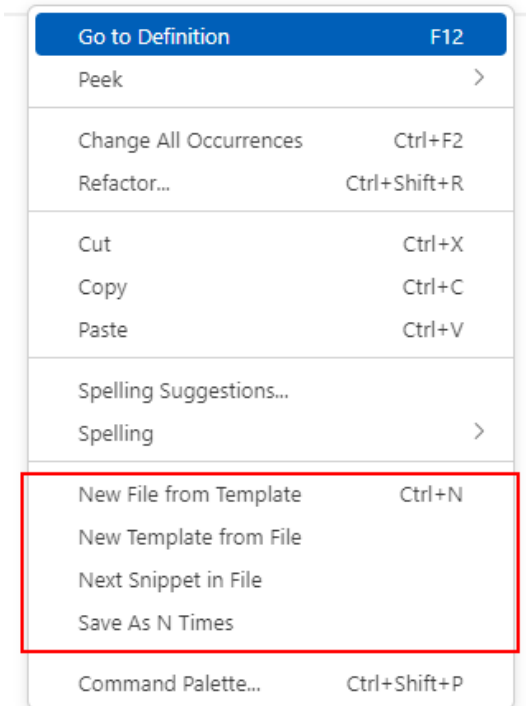
<https://code.visualstudio.com/docs/editor/userdefinedsnippets>

Standard code: templates

- Snippets could be used for whole programs, but usually you want things templated
- Not native functionality in VSCode, but there's an extension or two available ...
- We use the File Templates extension by rioj7
<https://marketplace.visualstudio.com/items?itemName=rioj7.vscode-file-templates>
 - Simply install from VS Code
 - Includes snippets with templates

File Templates

- Create multiple templates at user, workspace or folder level
 - Folder may be useful for keeping templates for unit test classes separate from logic class templates
- Extension overrides default CTRL-N behaviour and adds *New File from Template* option to context menu in editor and explorer outline
 - Standard "new file" behaviours are still available File menu items
- Can create multiple files at once from a file (Save as N Times)



Defining file templates

- Templates typically in `.vscode/templates` folder
 - May be in other places too, see extension doc
- Templates contain tokens / variables that can be replaced by user input or derived from the file name, so that we can
 - Convert file name to a class name
 - Convert a relative file path to a package name
 - Insert timestamps, with formatting
 - Insert the developer's name

Example template – ABL-Class.cls

```

##@@##      ${input#Class file name (incl. extension)#}
/*-----
File       : ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}
Purpose    : ${input#Class purpose#}
Syntax     :
Description:
Author(s)  : ${author} / Consultingwerk Ltd.
Created    : ${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}
Notes      :
-----*/

block-level on error undo, throw.

{Consultingwerk/products.i}

using Consultingwerk.*           from propath.
using ${relativeFile|find=(cls$)|replace=|flags=g|find=\/|replace=.|flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|} from
propath.

class ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}:

    /**
     * Purpose: Default constructor for ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.|flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
     * Notes:
     */
    constructor public ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.|flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
    ():
    end constructor.

end class.

```

Example template – ABL-Class.cls

Prompts for user input

```

##@###    ${input#Class file name (incl. extension)#}
/*-----
File      : ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}
Purpose   : ${input#Class purpose#}
Syntax    :
Description :
Author(s) : ${author} / Consultingwerk Ltd.
Created   : ${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}
Notes     :
-----*/

block-level on error undo, throw.

{Consultingwerk/products.i}

using Consultingwerk.*           from propath.
using ${relativeFile|find=(cls$)|replace=|flags=g|find=\/|replace=.|flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|} from
propath.

class ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}:

    /**
     * Purpose: Default constructor for ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=#flags=g#|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
     * Notes:
     */
    constructor public ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=#flags=g#|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
    ():
    end constructor.

end class.

```

Example template – ABL-Class.cls

Derive class name and package from relative path

```

##@##      ${input#Class file name (incl. extension)#}
/*-----
File       : ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=.#flags=g#}
Purpose    : ${input#Class purpose#}
Syntax     :
Description:
Author(s)  : ${author} / Consultingwerk Ltd.
Created    : ${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}
Notes      :
-----*/

block-level on error undo, throw.

{Consultingwerk/products.i}

using Consultingwerk.*                from propath.
using ${relativeFile|find=(cls$)|replace=|flags=g|find=\/|replace=.|flags=g|find=(\[a-zA-Z0-9#%_]+)(\.)$|replace=.*|flags=g|} from
propath.

class ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=.#flags=g#}:

    /**
     * Purpose: Default constructor for ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.#flags=g|find=(\[a-zA-Z0-9#%_]+)(\.)$|replace=.*|flags=g|}
     * Notes:
     */
    constructor public ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.#flags=g|find=(\[a-zA-Z0-9#%_]+)(\.)$|replace=.*|flags=g|}
    ():
    end constructor.

end class.

```


Example template – ABL-Class.cls

Insert developer name from settings

```

##@##      ${input#Class file name (incl. extension)#}
/*-----
File       : ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}
Purpose    : ${input#Class purpose#}
Syntax     :
Description:
Author(s)  : ${author} / Consultingwerk Ltd.
Created    : ${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}
Notes      :
-----*/
block-level on error undo, throw.

{Consultingwerk/products.i}

using Consultingwerk.*           from propath.
using ${relativeFile|find=(cls$)|replace=|flags=g|find=\/|replace=.|flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|} from
propath.

class ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=#flags=g#}:

    /**
     * Purpose: Default constructor for ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=#flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
     * Notes:
     */
    constructor public ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=#flags=g|find=(\[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
    ():
    end constructor.

end class.

```

Example template – ABL-Class.cls

Print date, formatted as
CCYY-MM-DD

```

##@##      ${input#Class file name (incl. extension)#}
/*-----
File       : ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=.#flags=g#}
Purpose    : ${input#Class purpose#}
Syntax     :
Description:
Author(s)  : ${author} / Consultingwerk Ltd.
Created    : ${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}
Notes      :
-----*/

block-level on error undo, throw.

{Consultingwerk/products.i}

using Consultingwerk.*           from propath.
using ${relativeFile|find=(cls$)|replace=|flags=g|find=\/|replace=.|flags=g|find=(\.[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|} from
propath.

class ${relativeFile#find=(\.cls$)#replace=#flags=g#find=#replace=.#flags=g#}:

    /**
     * Purpose: Default constructor for ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.#flags=g|find=(\.[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
     * Notes:
     */
    constructor public ${relativeFile|find=(\.cls$)|replace=|flags=g|find=#replace=.#flags=g|find=(\.[a-zA-Z0-9#$_]*\.)$|replace=.*|flags=g|}
    ():
    end constructor.

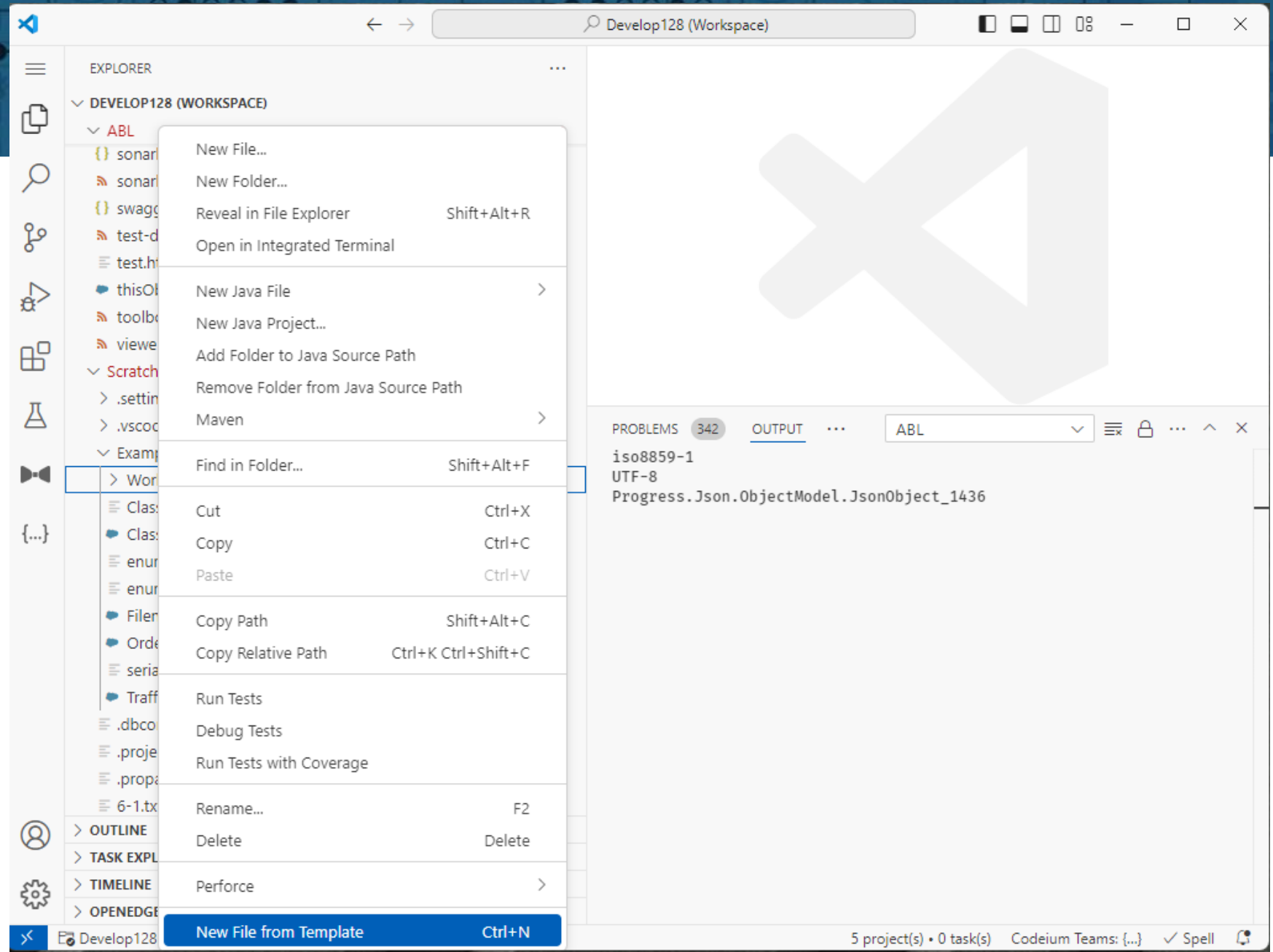
end class.

```

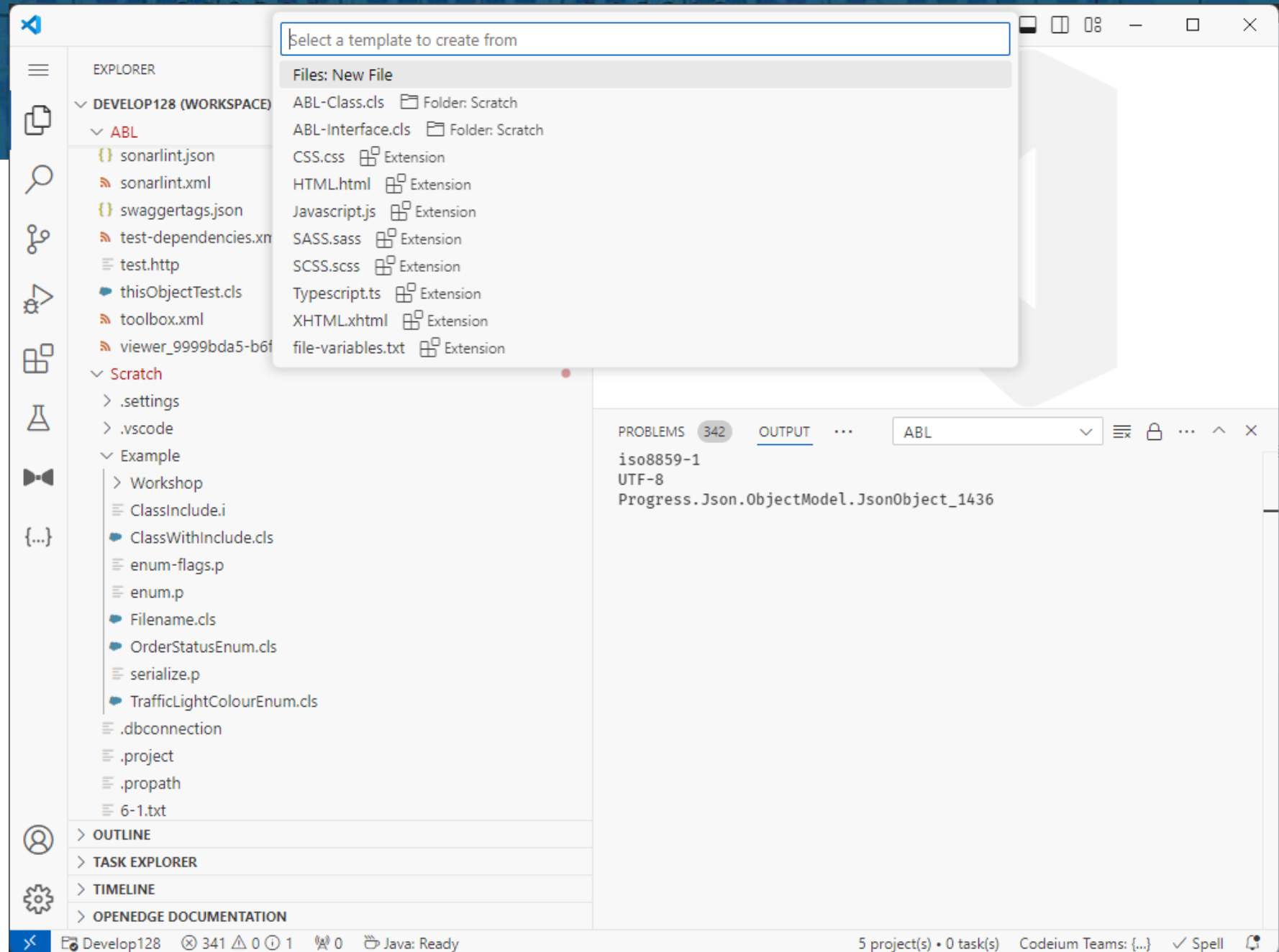
Tokens

- `${author}` is replaced by the value of `templates.author` in a settings file
 - Standard settings for folder, workspace or user
- Timestamps can be formatted as desired
 - Inline: `${dateTimeFormat#options={"year":"numeric","month":"2-digit","day":"2-digit"}#template=${year}-${month}-${day}#}`
- Regex-like functions exist manipulate various values

Example



Example



Example

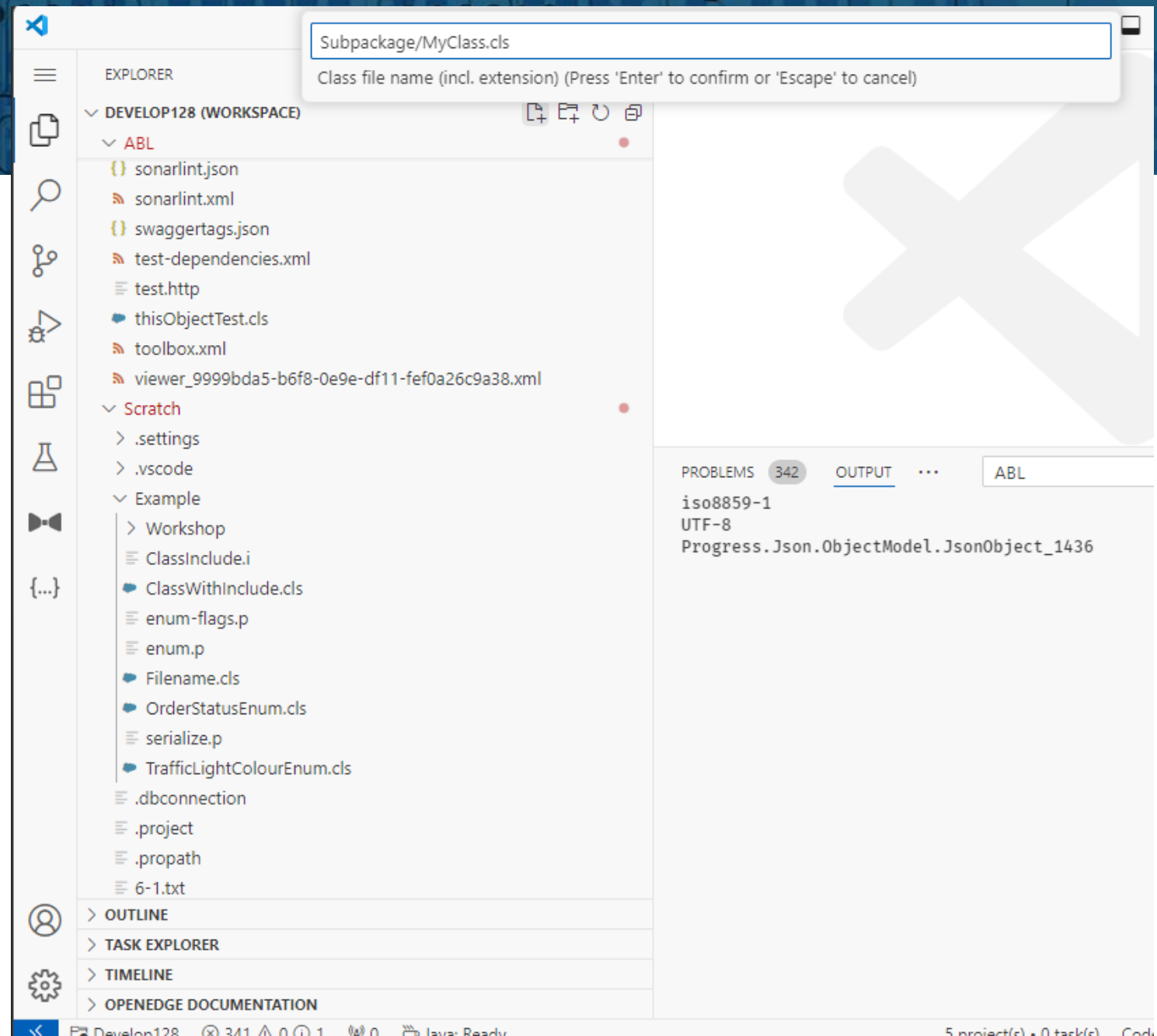
The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays a workspace named 'DEVELOP128 (WORKSPACE)' containing a folder 'ABL' with files like 'sonarlint.json', 'sonarlint.xml', 'swaggertags.json', 'test-dependencies.xml', 'test.http', 'thisObjectTest.cls', 'toolbox.xml', and 'viewer_9999bda5-b6f8-0e9e-df11-fef0a26c9a38.xml'. Below 'ABL' is a 'Scratch' folder with subfolders '.settings', '.vscode', and 'Example'. The 'Example' folder contains 'Workshop' and several class files: 'ClassInclude.i', 'ClassWithInclude.cls', 'enum-flags.p', 'enum.p', 'Filename.cls', 'OrderStatusEnum.cls', 'serialize.p', and 'TrafficLightColourEnum.cls'. Other files include '.dbconnection', '.project', '.propath', and '6-1.txt'. At the bottom of the sidebar are sections for 'OUTLINE', 'TASK EXPLORER', 'TIMELINE', and 'OPENEDGE DOCUMENTATION'.

A dialog box is open at the top center, titled 'Class file name (incl. extension) (Press 'Enter' to confirm or 'Escape' to cancel)'. It contains an empty text input field.

The PROBLEMS panel at the bottom right shows 342 errors. The selected error is 'iso8859-1 UTF-8 Progress.Json.ObjectModel.JsonObject_1436'. The status bar at the bottom indicates '5 project(s) • 0 task(s) Codeium Teams: {...} ✓ Spell'.

Example

- Paths are relative to where you start
 - In this case
Example/Workshop
- File created in
Example/Workshop/Subpackage



Example

The screenshot shows the Visual Studio Code interface. A dialog box is open at the top with the text "A class for the demo" and "Class purpose (Press 'Enter' to confirm or 'Escape' to cancel)". The Explorer view on the left shows a workspace named "DEVELOP128 (WORKSPACE)" with several folders and files. The "Scratch" folder is expanded, showing a subfolder "Workshop" which contains a subfolder "Subpackage" with a file "MyClass.cls" selected. The Code Editor shows the content of "MyClass.cls" with the following code:

```
1 /*****  
2 * Copyright (c) 2024-2024 by Consultingwerk Ltd. ("CW") -  
3 * www.consultingwerk.de and other contributors as listed  
4 * below. All Rights Reserved.  
5 *  
6 * Software is distributed on an "AS IS", WITHOUT WARRANTY OF ANY  
7 * KIND, either express or implied.  
8 *  
9 * Contributors:  
10 *  
11 *****/  
12 /*  
13 File : MyClass  
14 Purpose : ${input#Class purpose#}  
15 Syntax :
```

The PROBLEMS view at the bottom shows 344 errors, with the first one being "iso8859-1 UTF-8 Progress.Json.ObjectModel.JsonObject_1436".

Example

```
MyClass.cls
Scratch > Example > Workshop > Subpackage > MyClass.cls > ...
1  /*****
2  * Copyright (c) 2024-2024 by Consultingwerk Ltd. ("CW") - .....*
3  * www.consultingwerk.de and other contributors as listed .....*
4  * below. All Rights Reserved. ....*
5  * .....*
6  * Software is distributed on an "AS IS", WITHOUT WARRANTY OF ANY .....*
7  * KIND, either express or implied. ....*
8  * .....*
9  * Contributors: .....*
10 * .....*
11 *****/
12 /*
13 ... File ..... : MyClass
14 ... Purpose ..... : A class for the demo
15 ... Syntax ..... :
16 ... Description :
17 ... Author(s) ... : Peter Judge / Consultingwerk Ltd.
18 ... Created ..... : 2024-06-05
19 ... Notes ..... :
20 .....*/
21
22 @lowercase.
23
24 block-level on error undo, throw.
25
26 {Consultingwerk/products.i}
27
28 using Consultingwerk.* ..... from propath.
29 using Consultingwerk.Assertion.* ..... from propath.
30 using Consultingwerk.Util.* ..... from propath.
31 using Example.Workshop.Subpackage.* from propath.
32
33 class Example.Workshop.Subpackage.MyClass:
34
35     /**
36     * Purpose: Default constructor for MyClass
37     * Notes:
38     */
39     constructor public MyClass():
40     end constructor.
41
42 end class.
43
```

Conclusion

- VS Code is a feasible alternative to PDSOE for backend development
 - Fast
 - Multi-platform
 - Very many extensions
- The platform and extensions release frequently

Other sessions

- VS Code and ABL (Gilles Querret)
- ABL Formatter in Visual Studio Code (Paulius Kuprevičius)

Questions



Consultingwerk

software architecture and development