

June 15-16



Acuminator, Visual Studio and other developer tools

Developer productivity and solution quality validation

Sergey Nikomarov

Senior System Developer Acumatica

Agenda

Acuminator

Acuminator 3.0.0 – new features

Visual Studio

> Overview

- Code Editor Appearance
- Writing Code

Code Analysis

Integration with external tools

Other Developer Tools

- Visual Studio alternatives
- Examples of useful Visual Studio plugins

xt.AddScreenConfigurationFor(screen -=> StateIdentifierIs<status>() AddDefaultFlow(flow·=> .WithFlowStates(fss·=> fss.Add(initialState, flowState -=> flowState fss.Add<State.hold>(flowState -> ...); fss.Add<State.open>(flowState -=>...); fss.Add<State.confirmed>(flowState => ...) fss.Add<State.partiallyInvoiced>(flowStat fss.Add<State.invoiced>(flowState·=>...) fss.Add<State.completed>(flowState -=>WithTransitions(transitions -=> transitions.AddGroupFrom(initialState, ts transitions.AddGroupFrom<State.hold>(ts -= transitions.AddGroupFrom<State.open>(ts-= ts.Add(t'=>'t.To<State.hold>().IsTrig ts.Add(t'=>'t.To<State.confirmed>().J }); transitions.AddGroupFrom<State.confirmed> ts.Add(t+=>+t.To<State.open>().IsTrig ts.Add(t'=>'t.To<State.invoiced>().Is ts.Add(t+=>+t.To<State.partiallyInvoi</pre> ⇒ }); transitions.AddGroupFrom<State.partiallyJ transitions.AddGroupFrom<State.invoiced> transitions.AddGroupFrom<State.completed

Acuminator



New Features Overview

- □ Support of Visual Studio 2022
- □ Code analysis enhancements
- □ Code Map displays more information
- □ Suppression mechanism improvements
- Bugfixes



Acuminator: Static Code Analysis



Overview

- Throwing exception in row persisted graph event is now allowed if one of the following statements is true:
 - The graph is a processing graph
 - The exception type is one of the allowed exception types or their descendant:
 - PX.Data.PXRowPersistedException
 - PX.Data.PXLockViolationException
 - .NET exceptions from the System namespace: NotImplementedException, NotSupportedException, ArgumentException
- Underscores are now allowed in DAC extension names
- PX1008 diagnostic is significantly extended
- □ PX1088 diagnostic is entirely removed



Capturing reference to the screen graph in the long run delegate is a common error

// Reference to the graph is captured by the delegate because // GraphInstanceMethod is an instance method of the graph PXLongOperation.StartOperation(this, delegate() { GraphInstanceMethod(myDac); });

Capturing reference to the screen graph will cause synchronous delegate execution and different tricky issues including data consistency problems.

Here is a correct way to use graph instance methods in long run delegates

```
PXLongOperation.StartOperation(this, delegate()
{
    MyGraph graph = PXGraph.CreateInstance<MyGraph>();
    graph.GraphInstanceMethod(myDac);
});
```



PX1008 Diagnostic

In previous versions of Acuminator performed search for captured graph reference in processing delegates

□ Now this diagnostic also validates long run delegates





Acuminator: Code Map



Code Map Improvements



Different icons for DACs, DAC extensions, graphs, and graph extensions

The Code Map window layout now is remembered by Visual Studio

DAC nodes now indicate whether the node represents a DAC or a DAC extension





Base Overrides Category

- For graphs and graph extensions
 overrides of virtual type members
 are now displayed under the new
 Base Overrides node.
- The Persist method overrides
 are indicated with a special
 diskette icon





Graph

Initialization and Activation Category

□ New types of nodes represent:

- Graph instance and static constructors for graphs and graph extensions
- The IsActive method for graph and DAC extensions
- Initialization and Activation category node will contain these new nodes





Visual Studio



Visual Studio in comparison with other tools





Visual Studio features

- Read code with syntax highlighting and code regions folding
- □ Write code with intelligent code completion, code snippets, multiple carets, keyboard shortcuts
- □ Search your code with different search mechanisms
- □ Check your code with code analysis diagnostics
- Debug your code step by step, check objects in memory and modify their state
- □ Collaborate with your teammates with "Live Share" feature
- □ Modify your code during debug with "Hot reload" feature
- □ Profile your code with different diagnostic tools
- □ Integration with Version Control Systems like Git and other external tools
- □ Support for unit tests
- □ Highly configurable and extendable with many available plugins
- □ Many other things C# interpreter, navigation to decompiled sources, etc.



Visual Studio: Code Editor Appearance



Syntax Highlighting



public override Guid? NoteID

Search Options (Ctrl+E)	P	Show setting	_{s for:} Code E	ditor is chosen	in th	ne VS window j	picker	
Environment	~	Text Editor					✓ <u>U</u> se Defaults	
General Accounts AutoRecover		Eont (bold tyr Cascadia Mc	ve indicates fixed-v no Font f	width fonts): amily and size (<u>Size:</u> ✓ 10 ✓			
Documents Extensions Find and Replace		Display items User Membe	rs - Extension Met	thods	✓ <u>C</u> ustom			
Fonts and Colors Import and Export Settings International Settings		User Member User Member User Member User Member	rs - Fields rs - Labels rs - Locals rs - Methods	C# extension methods are selected		ltem bac <u>kg</u> round: Default	Color Picker	
Keyboard Preview Features Product Updates Tabs and Windows		User Member User Member User Member User Types -	rs - Namespaces rs - Parameters rs - Properties Classes	uno occesso a		Bold Turn o	on Bold Font Preview	
Task List Terminal Trust Settings	~	User Types - User Types - User Types -	Jelegates Enums Interfaces		× >	ij = I::	oO(0xB81l);	

Visual Studio highlights common C# syntax elements

It supports custom syntax highlighting

Visual Studio allows you to override default colors, backgrounds and fonts for any colorable code element

You can access syntax highlighting settings via "Tools" \rightarrow "Options" menu. Here look for the "Environment" \rightarrow "Fonts and Colors" section

Overrides are stored separately for each Visual Studio Theme

You can always reset color to default one



Code Outlining

111	' <summary> The identifier of the Note obje</summary>	ct associated	with	the	document			
[P)	(Searchable] 🛑 Collapsed attribute (Acuminator)				🔶 .			
[РХ	Note(ShowInReferenceSelector = true, Selector	<pre>tor = typeof(</pre>		Co	llapsed			
÷	Search2<			coi	mment			
÷	ARInvoice.refNbr,							
÷	InnerJoinSingleTable <arregister, on<="">,</arregister,>							
÷	→ And <>>,							
÷	InnerJoinSingleTable<>>,	Collapsed BQL						
÷	Where2<>,	parts (Acumina	tor)					
+	And <>>,							
÷	OrderBy <>>))]							
66 re	ferences 0 changes 0 authors, 0 changes							
pub	lic override Guid? NoteID 👍 Collapsed cu	rly braces						
#en	dregion							
Ref	RefNoteID Collapsed region							

ג רΩ נוֹם	Cut Copy Paste Insert Guid	Ctrl+X Ctrl+C Ctrl+V Ctrl+K, Ctrl+Space	Hide Selection Toggle Outlining Expansion Toggle All Outlining Stop Outlining	Ctrl+M, Ctrl+H Ctrl+M, M Ctrl+M, L Ctrl+M, P
	Annotation	×	Stop Hiding Current	Ctrl+M, Ctrl+U
	Outlining	•	Collapse to Definitions	Ctrl+M, O

Visual Studio supports **code outlining** (or **code folding**)

You can collapse some code fragments to get a better overview of the overall structure:

- curly braces
- #regions
- XML doc comments

Acuminator adds outlining for attributes and BQL parts

A tooltip with contents is displayed on mouse hover

Code editor context menu contains commands that can hide the selected text or manage the outlining for the whole file



Code Outlining Settings

□ C# outlining settings in Visual Studio options: "Text Editor" \rightarrow "C#" \rightarrow "Advanced" \rightarrow "Outlining"

□ Enable "Collapse #regions when collapsing to definitions" for better experience with DACs

Options					
Search Options (Ctrl+E)	Suggest usings for types in NuGet packages				
 ▷ Environment > Projects and Solutions > Source Control > Work Items 	Quick Actions Compute Quick Actions asynchronously (experimental, requires restart)				
 Text Editor 	Highlighting				
General Advanced Code Cleanup	 Highlight references to symbol under cursor Highlight related keywords under cursor 				
File Extension All Languages ASP.NET Web Forms Basic 	Outlining Image: Outlining mode when files open Image: Outlining mode when files open				
▲ C# General Scroll Bars Tabs	 Show outlining for declaration level constructs Show outlining for code level constructs Show outlining for comments and preprocessor regions 				
Advanced	Collapse #regions when collapsing to definitions				



General Settings

Options		
Search Options (Ctrl+E)	٩	Settings
Source Control	^	✓ Drag and drop text editing
Work Items		Select subword on double click
 Text Editor 		 Automatic delimiter highlighting
General		Track changes
Advanced		Auto-detect UTF-8 encoding without sign
Code Cleanup		Follow project coding conventions
File Extension		Enable mouse click to perform Go to Defi
All Languages		Use modifier key: Ctrl V
ASP.NET Web Forms		
▷ Basic		Open definition in peek view
⊿ C#		D' I
General		Display
Scroll Bars		View writespace
labs		Show bidirectional text control characters
Advanced		Selection margin
Code Style		✓ Indicator margin
Intellisense		Fightight current line Show structure quide lines
▷ C/C++		Show structure guide lines
CSS		Show file health indicator
Django Template		Show the react indicator
Dockernie		Line spacing 1.0 v
> HOI		Show editing context in the editor
		✓ Line/Col
		Selections
 Javaschpt/Typeschpt ISON 		✓ Insert/Overwrite
N LESS		✓ Tab/Space
N Diain Text		✓ Line endings

- □ Code editor is highly configurable
- Different indicators, controls, lines and margins can be hidden
- □ Text display can be customized
- Default values are sensible

nature

finition

Some interesting settings:

- Drag and drop of the selected text can be disabled
- Subword selection can be enabled
- Disable .editorConfig files support for your Visual Studio
- Configure Go to Definition with mouse click
 - Enable peek view
- Enable the display of tabs and whitespaces



C# Settings

- Many languages have their own section with language specific settings
- □ All Languages allows to configure common settings like indentation for all languages

Interesting C# specific settings:

- Enable virtual space at the end of the line ٠
- Enable word wrap ٠
- **Disable line numbers** ٠

Options	
Search Options (Ctrl+E) > Environment > Projects and Solutions > Source Control > Work Items Text Editor General Advanced Code Cleanup File Extension All Languages ASP.NET Web Forms Basic C# General Scroll Bars 	atement completion] Auto list members



Scroll Bar

Search Options (Ctrl+E) Vork Items Text Editor General Advanced Code Cleanup File Extension All Languages ASP.NET Web Forms Basic C# General Scroll Bars	 Show <u>h</u>orizontal scroll bar Show <u>v</u>ertical scroll bar Display Show <u>a</u>nnotations over vertical scroll bar Show <u>c</u>hanges Show marks Show <u>e</u>rrors Show caret position Behavior Use <u>b</u>ar mode for vertical scroll bar Use <u>map mode for vertical scroll bar</u>
General Scroll Bars Tabs Advanced ✓ Code Style General ▷ Formatting	 ○ Use <u>b</u>ar mode for vertical scroll bar ● Use <u>map</u> mode for vertical scroll bar ☑ Show Preview <u>T</u>ooltip <u>S</u>ource overview: Wide ~

- □ Scroll bars can be configured independently for any language or in a uniform way for all languages
- □ All Languages allows to configure common settings like indentation for all languages
- □ A map mode for vertical scroll bar will draw a small overview of the opened file in the vertical scroll bar
- □ If the preview tooltip is enabled, it will display a code fragment from the file on mouse hover





Line Separators

Line Separators are gray horizontal lines that separate different types and type members

There must be at least one empty lines between adjacent code elements.



Switch to enable Line Separators is located in the Outlining settings

Environment	^	Outlining
Projects and Solutions		Enter outlining mode when files open
Source Control		Show procedure line separators
Vork Items		
▲ lext Editor		Show outlining for declaration level constructs
General		Show outlining for code level constructs
Advanced		Show outlining for comments and preprocessor regions
Code Cleanup File Extension		Collapse #regions when collapsing to definitions
rile extension		



Document Tabs



Search Options (Ctrl+E)	IVIAX_IMUM TAD WIGTN: 213 Image: Show Close button on document tabs Image: Show Close button in the document well Image: Colorize document tabs by: Image: Project matching
Documents Extensions Find and Replace Fonts and Colors Import and Export Settings	Pinned Tabs ✓ Show pinned tabs in a separate row ✓ Show pin button in unpinned tabs ✓ Maintain pin status if document is removed from well
International Settings Keyboard Preview Features Product Updates Tabs and Windows Task List Terminal	Preview Tab ✓ Allow new files to be opened in the preview tab ✓ Preview selected files in Solution Explorer (Alt+click to avoid previewing) ✓ Preview selected files in Find Results ✓ Preview selected files in Navigate To

- **Document Tabs** tabs with opened documents that allow you to switch between them
- Document Tabs support behavior of any Visual Studio tool window (drag and drop, docking, etc.)
- Document Tabs layout can be changed. You can place them on top of the editor, to the left or to the right
- Settings are located in "Options" \rightarrow "Environment" \rightarrow "Tabs and Windows"
- Document Tabs can be colorized based on document's project of file extension
- Document Tabs can be pinned. When Document Tabs are located on top of the code editor Pinned Tabs can be placed in a separate row





Visual Studio: Writing Code



Overview

Different ways to boost our code writing skills:

- □ Code Completion
- □ Clever suggestions made with ML models
- □ Refactorings (renaming) and Code Fixes
- □ Code Snippets
- □ Keyboard Shortcuts
- Multiple Carets
- □ Code generation
- Automatic Formatting and Code Cleanup
- □ Other things



IntelliSense Code Completion

□ IntelliSense – advanced code completion VS functionality

- □ IntelliSense suggests possible options while you type
- Suggestions may include types, local variables, method parameters, type members, extensions methods and code snippets
- □ The list of variants depends on the context. It is affected by the programming language, framework, type of the document, caret location and some other things.
- Settings for C# are stored in Visual Studio options:
 - "Text Editor" \rightarrow "C#" \rightarrow "General"
 - "Text Editor" \rightarrow "C#" \rightarrow "IntelliSense"
- IntelliSense commands can be accessed in "Edit" \rightarrow "IntelliSense" top level menu





IntelliCode Model Training

٧	/iew	Git	Project	Build	Debug	Test	Analyz	e	Tools	Extensions	Window	Help	Search (C
	2	Solution	n Explorer		Ctrl	+W, S		J	-	▶ Start -	⊳ @ -	🖓 🕞	👼 🚽 🗛
🖣		Git Char	nges		Ctrl	+0, Ctrl	+G						
🤞	P (Git Repo	ository		Ctrl	+0, Ctrl	+R						
11	•												
								-					
		Other W	/indows				•	2	HiveSe	erver2 Output	Window		
		Toolbars	5				۰.	Ŷ	Applic	ation Insights	Search		
ĸ	ĸ	Full Scre	en		Shif	t+Alt+E	inter	Ŷ	IntelliC	Code			

Visual Studio IntelliCode

Team Completions

Team Completions

✓ Show details

✓ I would like to train IntelliCode models on my currently open codebase: PX.Objects.HackathonDemo

Custom completions for PX.Objects.HackathonDemo

C# custom model

Model status: Not started

Learn patterns

- IntelliCode machine learning Visual Studio functionality that relies on predictive model trained on GitHub open-source repos
- IntelliCode trains ML models on your code base to provide context-specific suggestions

 □ IntelliCode model training functionality is located in: "View" → "Other Windows" → "IntelliCode"

- □ Trained models can be shared with your team
- Retraining can be done as a step in the CI pipeline



IntelliCode ML Features

- □ Al-assisted IntelliSense IntelliSense code completion extension that predicts API that are most likely to be used
 - Suggestions are marked with a star and placed first in the list of available variants
 - Argument completion for C#
 - Suggestions based on your own code for C#
- Refactoring suggestions for C# track your edits, detect repetitive actions and offer to apply these actions in similar places
- □ Whole line completions predict the next chunk of your code and present it as a gray text inline prediction
 - Available only in Visual Studio 2022
 - Can be configured by pressing lightbulb at the bottom right of the editor
- .editorconfig file generation with code style and formatting conventions inferred from the C# code base







Keyboard Shortcuts

- □ Keyboard shortcuts allow user to quickly perform some action by pressing a combination of keys on keyboard
- □ Well-known examples:
 - Copy / Paste with Ctrl + C / Ctrl +V •
 - Cancel with Ctrl + Z •
 - Search / Replace with Ctrl + F / Ctrl + H •
- Most Visual Studio commands support shortcuts and have a default shortcut assigned
- Plugins may provide their own shortcuts
- Custom shortcuts can be assigned to Visual Studio command even if it does not have a default one
- □ Shortcut settings can be accessed in Visual Studio options:
 - "Environment" \rightarrow "Keyboard"

Apply the following additional keyboard mapping scheme:							
Visual C# 2005		~	R <u>e</u> set				
Show <u>c</u> ommands con	aining:						
Edit.Dup	Type in the searchbox to	look fo	or a command				
Edit.Duplicate							
Select a comm	and to assign shortcut command: Current shortc	ut for t	he command				
Ctrl+D, Ctrl+D (Globa	l)	~	<u>R</u> emove				
Use <u>n</u> ew shortcut in:	Press shortcut keys:						
Global Scope	New Shortcut		A <u>s</u> sign				
Shortcut currently used by:							
Comman	d already using the enter	red sho	rtcut 🗸 🗸 🗸				



Useful Keyboard Shortcuts

Duplicate selected line with **Duplicate** command

- Can be accessed with "Edit" \rightarrow "Duplicate" •
- Look for **Edit.Duplicate** command in shortcut settings to assign ٠ a shortcut
- Change indent of selected line
 - Increase indent with Tab ٠
 - Decrease indent with Shift + Tab ٠
- Call IntelliSense code completion suggestions with Ctrl + Space
- Move selected lines up and down
 - Move up with Alt + Up arrow ٠
 - Move down with Alt + Down arrow ٠
- Other scenarios
 - Custom shortcuts for Visual Studio extensions ٠
 - Text selection ٠
 - Multiple Carets •
 - Other

Edit	View	Git	Project	Build	Debug	Test				
	Go To									
	Find and Replace									
	Go To Bas	Home								
5	Undo			Ctrl	+Z					
Q.	Redo			Ctrl						
5	Undo Las									
Q	Redo Last	Globa	l Action							
X	Cut			Ctrl	۰X					
ď	Сору			Ctrl						
Ĝ	Paste			Ctrl						
Ĝ	Show Clip	board	History	Ctrl	+Shift+V					
	Paste Spe	cial				•				
Ъ	Duplicate	()		Ctrl	+E, V					



Multiple Carets

- Multiple Carets allows you to edit multiple places of your code by creating multiple carets and applying changes to all of them simultaneously
- Multiple Carets works with the text selected in the code editor. It finds occurrences of the same text in the document and creates extra carets for them
- Carets can be created for all occurrences of the selected text or only for some of them
- Multiple Carets can be found in "Edit" \rightarrow "Multiple Carets"
- Shortcuts are much more convenient
 - Insert Carets at All Matching **Shift + Alt +**; ٠
 - Insert Next Matching Caret Shift + Alt + . ٠

Edit	View	Git	Project	Build	Debug	Test	Analyze	Tools	Extensions	Window	Help
	Go To Find and	Replac	e			•	Any CPU	•	▶ Start - 🕻	> @-	📭 E
	Go To Bas	e		Alt+	Home						
5	Undo			Ctrl+	Z					- 9	\$PX.O
୯	Redo			Ctrl+	٠Y						
	Undo Last	t Globa	I Action								
	Redo Last Global Action										
	IntelliSens	se				۲					
	Multiple (Carets				۲	Insert	Next Ma	tching Caret	Shift	+Alt+.
L	Refactor					۲	Insert	Carets at	All Matching	Shift	+Alt+;
	IntelliCod	e sugg	estions			۲	Remo	ove Last C		Shift	+Alt+,
	Next Met	hod					Move	Last Care	et Down	Shift	+Alt+/
	Previous I	Metho	d				Rotat	e Primary	Caret Down		



Code Snippets Overview

- **Code Snippets** small blocks of reusable code that can be quickly inserted into your document
- □ Snippets can contain placeholders with different names that you replace with your code.
- The replacement will be done for every instance of the same placeholder in the inserted snippet
- Two types of code snippets
 - **Expansion** snippets inserted at the cursor position ٠
 - Surround With snippets that surround selected code ٠
- Extra features:
 - Generate cases for every member of enum in a code snippet for the **switch** statement ٠
 - Add missing using for namespaces ٠
 - Add references to assemblies from Global Assembly Cache (GAC) ٠



Managing Code Snippets

Code Snippets Manager	?	\times
Language:		
CSharp ~		
Location:		
C:\Users\snikomarov\Documents\Visual Studio 2022\Code Snippets\Visual C#\My Code Snippets		
 %LCID% My Code Snippets NetFX30 Office Development Other Refactoring Test U-SQL Visual C# 		
Add Remove		
Import OK	Cance	1

- □ Visual Studio comes with a lot of code snippets for different programming languages and frameworks
- □ You can write your own code snippets and import them in Visual Studio
- □ Code Snippets are XML files with **.snippet** extension

□ Custom code snippets are stored in Visual Studio subfolder in current user documents: <Current User Folder>\Documents\<Visual Studio version>

There should be a dedicated subfolder: \Code Snippets\Visual C#\My Code Snippets

You can manage code snippets in the "Code Snippets Manager" "Tools" \rightarrow "Code Snippets Manager"



Acumatica Code Snippets

Acumatica Code Snippets – code snippets developed for Acumatica Framework

□ Acumatica Code Snippets include templates for

- DAC •
- DAC fields ٠
- Graph field and row events ٠
 - Name Convention Signature 0
 - **Generic Signature** 0
 - Short Generic Signature for field events 0
- Available on GitHub: <u>https://github.com/Acumatica/CodeSnippets</u>
- Acumatica Code Snippets can be imported into Visual Studio manually
- Acuminator next version will include Acumatica Code Snippets collection





C# Interactive

C# interactive is a useful tool for quickly running a piece of code

□ C# Interactive is based on C# read-evaluate-print-loop (REPL) functionality

 \Box Located in "View" \rightarrow "Other Windows" \rightarrow "C# Interactive"

□ You can run it either line by line or from a script

```
C# Interactive
(5 ≦ ↑ ↓
  Microsoft (R) Visual C# Interactive Compiler version 4.1.0-5.22165.10 ()
  Loading context from 'CSharpInteractive.rsp'.
  Type "#help" for more information.
  > using System.Reflection;
  > var type = typeof(List<>);
  > type.FullName
   "System.Collections.Generic.List`1"
  > Enumerable.Range(1,10)
  RangeIterator { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }
  >
```



Visual Studio: Code Analysis



Overview

□ Code analysis for C# and VB in modern versions of Visual Studio is based on Roslyn framework

- Library authors can provide custom diagnostics verifying that their API is used correctly
 - □ xUnit
 - □ Acuminator
- Advanced general-purpose diagnostics developed by community as open-source projects
- Many ways to suppress a particular error pragma directives, special attributes, project settings, .editorconfig and .ruleset files
- □ Two ways to add custom diagnostics to the solution
 - Install Visual Studio extension to use diagnostics only for yourself
 - Add a Nuget package with diagnostics to the project to enforce rules for everyone in the team



Code Analysis Settings

- □ Each language has its own code analysis settings
 - C# settings can be found in "Text Editor" → "C#" → "Advanced" → "Analysis"
- □ Some settings affect Visual Studio performance
 - Background analysis scope
 - Run code analysis in a separate process in VS 2022
 - Use 64-bit process for code analysis in VS 2019
- Enable built-in decompiler with "Enable navigation to decompiled sources"
- Try experimental "Display diagnostics inline" feature

Analysis									
Run background code analysis for:									
Open documents v									
Display diagnostics inline (experimental)									
I at the end of the line of code	errors in code								
\bigcirc on the right edge of the editor window		Perform	ance						
Enable navigation to decompiled sources	Decompiler								
 Always use default symbol servers for navigation 	tion								
Enable 'pull' diagnostics (experimental, requi	res restart)								
Run code analysis in separate process (require	es restart)								
Show "Remove Unused References" command in Solution Explorer (experimental)									
Enable file logging for diagnostics (logged in '%Temp%\Roslyn' folder)									
Skip analyzers for implicitly triggered builds									

2 references | 0 changes | 0 authors, 0 changes public class SOOrderWithHold : SOOrderWithTotal

PX1009: Multiple levels of inheritance are not supported for PXCacheExtension



Project specific settings

- .Net projects have project specific code analysis settings
- Some code analysis settings can be accessed in project "Properties" window
- Code Analysis section is related to a legacy FxCop code analysis and is deprecated
- Build section contains analysis settings
 - Warning level. For higher levels compiler ٠ provides more warnings
 - **Disabled warnings** ٠
 - Treat warnings as errors •
- More settings can be specified directly in the project file

Application	Configuration: Active (Debug)	Platform: Active (Any CPU)
Build*		
Build Events	General	
Debug	Conditional compilation symbols:	
Resources	Define DEBUG constant	
Services	Define TRACE constant	
Settings	Platform target:	Any CPII
Reference Paths		
Signing		
Code Analysis	Allow unsate code	
Deprecated old	Errors and warnings	
code analysis	Warning level:	4 ~
	Suppress warnings:	
	Treat warnings as errors	
	○ None	
	Specific warnings:	



Manually Run Code Analysis on Solution

- Commands to run code analysis manually are located in "Analyze" \rightarrow "Run Code Analysis"
- You can run code analysis for the current project of for the entire solution
- All errors are listed in the "Error List" window
- Works in latest versions of Visual Studio 2019 and 2022

Ana	lyze Tools Extensions Windo	w	Help	Search (Ctrl+Q)		R	PX.Objects.Ha		
	Code Cleanup	•	Ep	🗟 🚽 Align by	¥ 🚽 🖢 🕼	1 1	II 🛛 🖓 🖓		
	Run Code Analysis	►	0	n Solution					
	Build and Suppress Active Issues			Run Code Analysis on PX.Objects.HackathonDemo					
	Configure Code Analysis		лоојес	сы наскаснопренно.	SOOTGETWICHTOIC	1			
	Calculate Code Metrics								
	Windows	Þ							



Integration with Other Tools



Different Integration Approaches

- □ Integrations built-in into Visual Studio
 - Git and other Version Control Systems
 - SQL Server
 - Unity and other game engines
 - Unit test frameworks
- □ Integrations provided by plugins
 - "Open with XXX" plugins that allow you to open file in a different tool
 - Support for different frameworks and tools
- □ Add tool to Visual Studio External Tools list located in "Tools" → "External Tools"
- Specify external tool as a designer for files with specific extension

External Tools		? ×
Menu contents:		
Create &GUID		Add
		Delete
		Move Up
		Move Down
Title:	Create &GUID	
Command:	C:\Program Files\Microsoft Visual St	udio\2022\F
Arguments:	\$(CurLine) \$(CurText)	
Initial directory:	C:\Program Files\Microsoft Visual St	udio\2022\P
Use Output window	Prompt for argume	nts
Treat output as Unicod	e Close on exit	
	OK Cancel	Apply



External Tool Editor



Open With - AP612000.rpx		?	\times
Choose the program you want to use to open this file:		Add external t	ool
Common Language Editor Supporting TextMate Bundles (Default)	^	Add	
JSON Editor			
JSON Editor with Encoding		Demovie	
Automatic Editor Selector (XML)		Kentove	
XML (Text) Editor			
XML (Text) Editor with Encoding		Set as Defaul	t
Source Code (Text) Editor			
Source Code (Text) Editor With Encoding		Set selected	
HTML Editor		adhean a dafa	
HTML Editor with Encoding		editor as deta	iuit
HTML (Web Forms) Editor		one for files v	vith
HTML (Web Forms) Editor with Encoding		this extensior	1
CSS Editor			
CSS Editor with Encoding			
SCSS Editor			
SCSS Editor with Encoding	×		
Available Editors	OK	Cancel	

- Visual Studio allows you to choose which tool should be used to open files with specific extension
- □ The editor selection dialog is located in "Solution Explorer" window in the context menu
- Select required file, make right mouse button click on it and select "Open With..." command
- There are many available editors shipped with Visual Studio that can be used to open the file
 - External tools can be added to available editors
- One of the editors can be selected as a default editor for files with the same extension
- Report Designer can be set as a default editor for Acumatica .rpx reports



Visual Studio Git Integration

- □ **Git** the most popular version control system used by many projects for collaborative development
- Git is a console application but there are many UI clients
- □ Visual Studio Git Integration is another UI client for Git
- Repository settings, branches and changes are displayed
- □ Simple Git workflow actions can be used from Visual Studio



Git Changes - Acumatica						- +	- ×
2021r212 Branch Selector		-	⇒	\downarrow	$\underline{\uparrow}$	Ĵ	
1↓ <u>0 outgoing / 0 incoming</u>	Cor	nmon	Act	tion	s To	oolb	ar
Enter a message <required> Comm</required>	it message						
Commit All 👻 🗌 Amend Butte	on to commit o	r stas	h c	han	ges		
▲ Changes (5)						+	
🔺 🗁 C:\Repos2\AcumaticaExtraRepo	\WebSites\Pure						
▲ PX.Objects\CR							
C# OpportunityMaint.cs							М
🔺 🗁 Site							
🔺 🗁 App_Data							
🔺 🗁 SnapshotConfigs							
🗋 Settings and Busine	s Accounts.esc						М
🞵 CustomizationStatus.xi	nl						М
🔺 🚞 Pages\El							
EISC2200.aspx							Α
C# EISC2200.aspx.cs							Α



Git Settings

Git settings are located in Visual Studio options in "Source Control" section

Two sets of settings – global settings for all repositories and local repository settings

Options						?	\times
Search Options (Ctrl+E)		increasing actions	nepository settings init orientide the grobal setting ratios.				
Search Options (Ctrl+E) Environment Projects and Solutions Source Control Plug-in Selection Git Global Settings Git Repository Settings General		User name: Email: Prune remote bran Rebase local branc	Sergey Nikomarov snikomarov@acumatic nches during fetch: Tru th when pulling: Un	ue v nset v			
Remotes Work Items Text Editor Debugging Performance Tools NET Core Debugging with WSL Acuminator Azure Data Lake Azure Service Authentication Container Tools	`	Tools Diff Tool: Visual St Merge Tool: Visua Git files Ignore file: /.gitigr Attributes file: /.gi	udio Global I Studio Global nore Edit tattributes Edit				~
					ОК	Cance	el



Other Developer Tools



Alternatives to Visual Studio

- Jet Brains tools
 - Rider IDE
 - dotTrace and dotMemory for profiling
 - dotPeek decompiler
- □ Visual Studio Code
- □ Notepad++

Rider
dotTrace
dotMemory
dotMemory
dotPeek
Visual Studio Code





Useful Visual Studio Extension

- **ReSharper** and **OzCode**
- Code Analysis extensions
 - **Roslynator** 500+ analyzers, refactorings and code fixes for C# ٠
 - **StyleCop** rules to enforce style and consistency rules for source code ٠
- Visual Studio Spell Checker and other spell-checking extensions
- **Debug Attach Manager** for easy attaching to IIS processes
 - **ReAttach** as alternative ٠
- Syntax highlight extensions **Viasfora**, **Codinion** and many others
- Productivity extensions
 - Open in XXX extensions Open in Visual Studio Code, Open in Notepad++ ٠
 - Match Margin / Match Margin 2022 to display occurrences of the selected word in scroll bar ٠
 - **Code alignment** aligns your code vertically to improve readability ٠
- Atomineer Pro Documentation for clever generation of XML doc comments
 - **GhostDoc** as alternative ٠



Summary

Tools that boost developer productivity, ensure product quality and take care of routine tasks

- □ Acuminator new version with support of Visual Studio 2022, improved code checks and new data displayed on Code Map
- Visual Studio know your code editor!
 - Customize appearance ٠
 - Enable advanced features and optimize performance ٠
 - Write code faster •
 - Configure code analysis ٠
 - Integrate with other tools ٠

Other Tools – useful plugins and alternatives to Visual Studio



Resources – Acuminator

Acuminator Release Notes:

https://github.com/Acumatica/Acuminator/blob/dev/docs/ReleaseNotes.md

Overview of all diagnostics:

https://github.com/Acumatica/Acuminator/blob/dev/docs/Summary.md

Acuminator is an open-source project. The list of issues: https://github.com/Acumatica/Acuminator/issues



Resources – Visual Studio

Visual Studio Blog <u>https://devblogs.microsoft.com/visualstudio/</u>

Editor features overview

https://docs.microsoft.com/en-us/visualstudio/ide/writing-code-in-the-code-and-text-editor?view=vs-2022

□ Keyboard shortcuts

https://docs.microsoft.com/en-us/visualstudio/ide/identifying-and-customizing-keyboard-shortcuts-in-visualstudio?view=vs-2022

□ Tips for C# developers

https://docs.microsoft.com/en-us/visualstudio/ide/csharp-developer-productivity?view=vs-2022

IntelliCode

https://docs.microsoft.com/en-us/visualstudio/intellicode/intellicode-visual-studio

□ Code Snippets

- C# Code Snippets <u>https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-code-snippets?view=vs-2022</u>
- Snippets for Acumatica <u>https://github.com/Acumatica/CodeSnippets</u>



June 15-16



Thank You!

www.acumatica.com/developers