

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

- Writing Code Effectively in Visual Studio
- Code Analysis
- Integration with external tools

Other Developer Tools

- Visual Studio alternatives
- Examples of useful Visual Studio plugins

		Scree	enCont	figur	atio	nFor(scre	en ·=	>	
		denti	ifier	Is <st< th=""><th>atus</th><th>>()</th><th></th><th></th><th></th><th></th></st<>	atus	>()				
		aultF	low(flow·	=>					
		W								
	.Wit	thFlo	wStat	tes <mark>(f</mark>	ss:=	>				
		fss.	. Add (:	initi	alSt	ate,	flow	State	e-=>-	flowS
		fss.	Add<	State	.hol	d>(f]	lowSt	ate∵	=>);
		fss.	Add<	State	.ope	n>(f]	lowSt	ate∵	=>);
		fss.	Add<	State	.con	firme	ed>(f	lowS	tate ·	=>
		fss.	Add<	State	.par	tiall	lyInv	oice	d>(fl	owSta
		fss.	Add<	State	.inv	oiced	>(fl	owSta	ate·=	>)
		fss.	Add<	State	.com	plete	ed>(f	lowS	tate ·	=>
	.Wit	thTra	ansit	ions(tran	sitic	ns:=	>		
		trar	isiti	ons.A	ddGr	oupFr	om(i	niti	alSta	te,∙t
		trar	nsiti	ons.A	ddGr	oupFr	om <s< th=""><th>tate</th><th>.hold</th><th>>(ts∙</th></s<>	tate	.hold	>(ts∙
		trar	nsiti	ons.A	ddGr	oupFr	om <s< th=""><th>tate</th><th>.open</th><th>>(ts∙</th></s<>	tate	.open	>(ts∙
		{								
		÷								IsTri
		7	ts.A	dd(t∙	=> · t	.To <s< th=""><th>itate</th><th>.con</th><th>firme</th><th>d>().</th></s<>	itate	.con	firme	d>().
		});								
			isiti	ons.A	ddGr	oupFr	om <s< th=""><th>tate</th><th>.conf</th><th>irmed</th></s<>	tate	.conf	irmed
		{								
		7						-		IsTri
		>								>().I
		7	ts.A	dd(t∙	=> · t	.To <s< th=""><th>state</th><th>.par</th><th>tiall</th><th>yInvo</th></s<>	state	.par	tiall	yInvo
		});								
						-			-	ially
										iced>
		trar	nsiti	ons.A	ddGr	oupFr	·om <s< th=""><th>tate</th><th>.comp</th><th>leted</th></s<>	tate	.comp	leted
	}))									

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

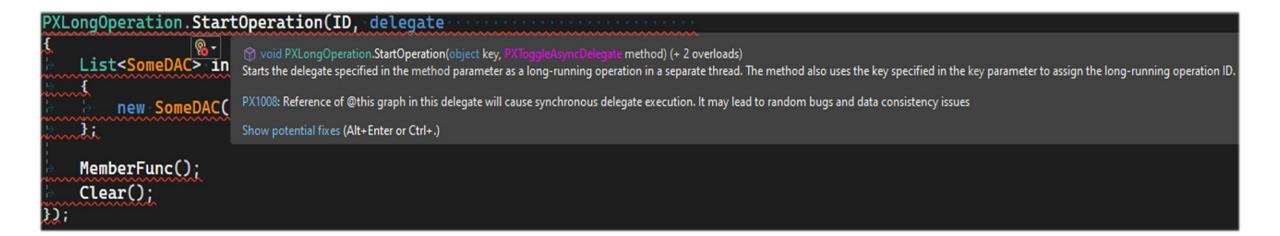
```
PXLongOperation.StartOperation(this, delegate()
{
    MyGraph graph = PXGraph.CreateInstance<MyGraph>();
    graph.GraphInstanceMethod(myDac);
});
```



PX1008 Diagnostic

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

□ Now this diagnostic also validates long run delegates

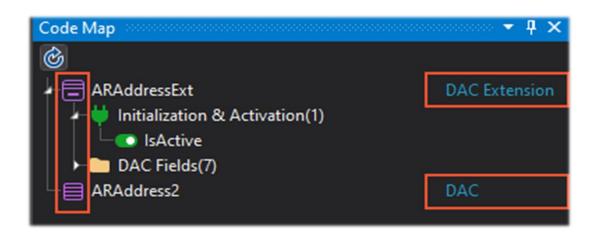




Acuminator: Code Map



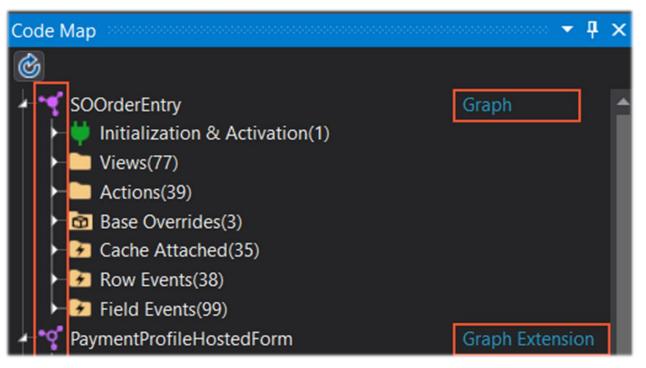
Code Map Improvements



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

The Code Map window layout now is remembered by Visual Studio

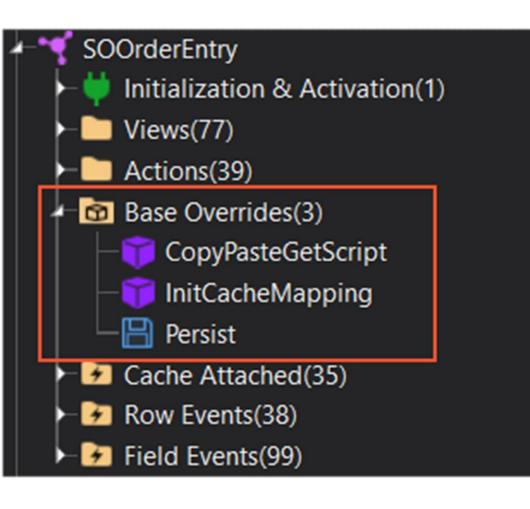
DAC nodes now indicate whether the node represents a DAC or a DAC extension





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



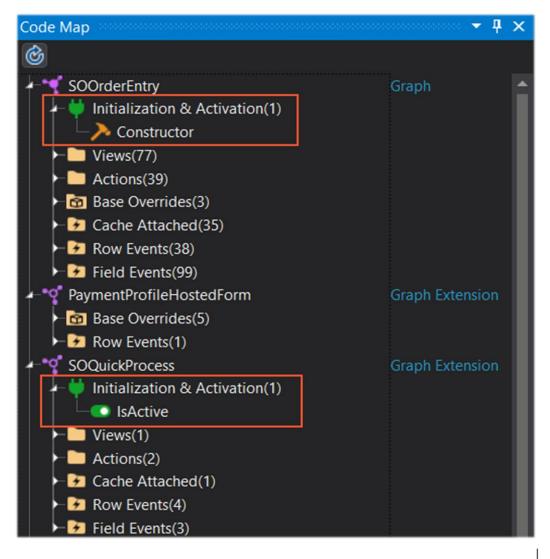


Graph

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: Writing Code Effectively



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



Visual Studio Options

Тоо	ls	Extensions	Window	Help	Search (Ctrl+Q)						
	Ge	t Tools and Fea	tures								
	M	anage Preview I	Features								
	At	omineer Pro Do	cumentatio	on	•						
1	Re	sX Manager									
Ÿ⊇	Co	nnect to Datab	ase								
별극	Connect to Server										
\$ >	Ad	Add SharePoint Connection									
	sc	L Server			•						
	Da	ta Lake			+						
	Co	de Snippets Ma	anager		Ctrl+K, Ctrl+B						
	Ch	oose Toolbox I	tems								
	Nu	ıGet Package N	lanager		۲						
	Ру	thon			۲						
	Cr	eate GUID									
œ	W	CF Service Conf	iguration E	ditor							
ď	Cl	ear MEF Compo	onent Cach	e							
8		ow MEF Errors									
	Ex	ternal Tools									
	Th	eme			۲						
	Co	mmand Line			•						
	Im	port and Expor	t Settings								
	Cu	istomize									
÷\$	OF	otions									

Visual Studio options are located in: "Tools" \rightarrow "Options..."

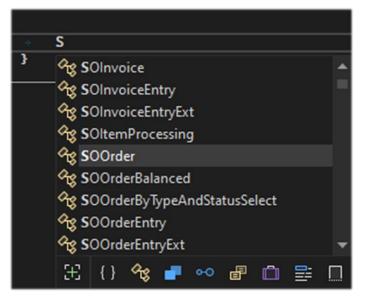
م ^	Analysis Run background code analysis for: Open documents	Â
^	Open documents v	
	 Display diagnostics inline (experimental) at the end of the line of code on the right edge of the editor window Enable navigation to decompiled sources Always use default symbol servers for navigation Enable 'pull' diagnostics (experimental, requires restart) Run code analysis in separate process (requires restart) Show "Remove Unused References" command in Solution Explorer (experir Enable file logging for diagnostics (logged in '%Temp%\Roslyn' folder) Skip analyzers for implicitly triggered builds 	
	•	 Always use default symbol servers for navigation Enable 'pull' diagnostics (experimental, requires restart) Run code analysis in separate process (requires restart) Show "Remove Unused References" command in Solution Explorer (experim Enable file logging for diagnostics (logged in '%Temp%\Roslyn' folder) Skip analyzers for implicitly triggered builds



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" → "C#" → "General"
 - "Text Editor" → "C#" → "IntelliSense"
- IntelliSense commands can be accessed in "Edit" → "IntelliSense" top level menu





IntelliCode Model Training

Viev	v Git Project	Build	Debug	Test	Analyze	e	Tools	Extensions	Window	Help	Search (C
	Solution Explorer		Ctrl	+W, S		J	•	Start -	> @ -]	0/ 6	🗖 _ 🗛
5	Git Changes		Ctrl	+0, Ctrl	+G						<u>w</u>
5	Git Repository		Ctrl	+0, Ctrl	+R						
^											
	Other Windows				•	0.1	Line	-	Window		
	Other Windows					03		erver2 Output			
	Toolbars				۰ -	Ŷ	Applic	ation Insights	Search		
кл КУ	Full Screen		Shif	t+Alt+E	Enter	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

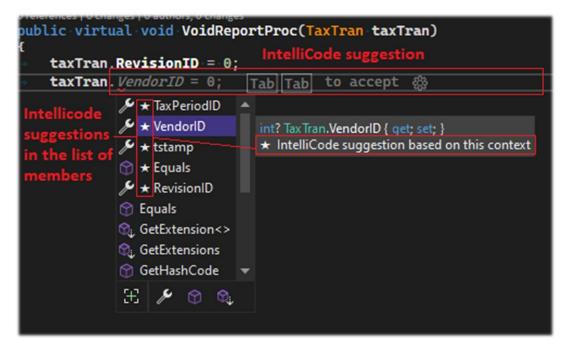
 □ IntelliCode model training functionality is located in: "View" → "Other Windows" → "IntelliCode"

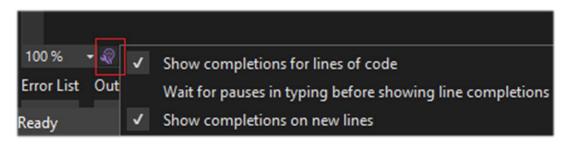
- □ Trained models can be shared with your team
- Retraining can be done as a step in the CI pipeline



IntelliCode ML Features

- □ **AI-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 ٠ list of available variants
 - Argument completion for C# ٠
 - Suggestions based on your own code for C# ٠
- **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 ٠ 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"

Visual C# 2005		\sim	R <u>e</u> set
Show <u>c</u> ommand	ls containing:		
Edit.Dup	Type in the searchbox to I	look f	or a command
Edit.Duplicate			
Select a co	ommand to assign shortcut		
	ommand to assign shortcut ected command: Current shortcu	t for t	he command
	ected command: Current shortcu	t for t	he command
Shor <u>t</u> cuts for sel	ected command: Current shortcu (Global)	t for t	
Shor <u>t</u> cuts for sel Ctrl+D, Ctrl+D	ected command: Current shortcu (Global) It in: <u>Press shortcut keys:</u>	t for t	
Shor <u>t</u> cuts for sel Ctrl+D, Ctrl+D Use <u>n</u> ew shortcu	ected command: Current shortcu (Global) It in: Press shortcut keys: De V New Shortcut	t for t	<u>R</u> emove



Useful Keyboard Shortcuts

Duplicate selected line with **Duplicate** command

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

Edit	View	Git	Project	Build	Debug	Test
	Go To					•
	Find and F	Replac	e			•
	Go To Bas	e		Alt+	Home	
5	Undo			Ctrl+	۰Z	
6	Redo			Ctrl+		
5	Undo Last	Globa	I Action			
Q	Redo Last	Globa	l Action			
X	Cut			Ctrl+	٠X	
ď	Сору			Ctrl+	+C	
Ĝ	Paste			Ctrl+	۰V	
Ĝ	Show Clip	board	History	Ctrl+	-Shift+V	
	Paste Spe	cial			,	•
ď	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" → "Multiple Carets"
- □ Shortcuts are much more convenient
 - Insert Carets at All Matching Shift + Alt + ;
 - Insert Next Matching Caret Shift + Alt + .

Edit	View Git	Project	Build	Debug	Test	Analyze	Tools	Extensions	Window Help
	Go To				۲	Any CPU	÷	Start -	> 🤄 - 📭 🗖
	Find and Replace	2			×				
	Go To Base		Alt+	Home					
5	Undo		Ctrl+	٠Z					- 🕫 PX.O
6	Redo		Ctrl+	۲Υ					
5	Undo Last Globa	I Action							
Q	Redo Last Global	Action							
	IntelliSense				۲				
	Multiple Carets				•	Insert	Next Ma	tching Caret	Shift+Alt+.
·	Refactor				۲	Insert	Carets at	All Matching	Shift+Alt+;
	IntelliCode sugg	estions			۲	Remo	ove Last C		Shift+Alt+,
	Next Method					Move	Last Car	et Down	Shift+Alt+/
	Previous Method	1				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
 - Surround With snippets that surround selected code
- □ Extra features:
 - Generate cases for every member of enum in a code snippet for the **switch** statement
 - Add missing using for namespaces
 - Add references to assemblies from Global Assembly Cache (GAC)



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" → "Code Snippets Manager"

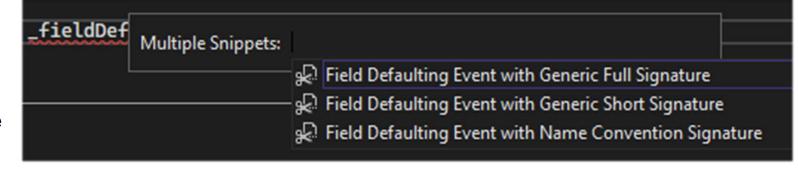


Acumatica Code Snippets

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			Ý				
Display diagnostics inline (experimental)	Display found						
 at the end of the line of code 	Display found errors in code						
\bigcirc on the right edge of the editor window	lenois in code	Perform	ance				
Enable navigation to decompiled sources	Decompiler						
Always use default symbol servers for naviga	tion						
Enable 'pull' diagnostics (experimental, requi	res restart)						
Run code analysis in separate process (requir	es restart)						
Show "Remove Unused References" comman	nd in Solution Explor	rer (experin	nental)				
Enable file logging for diagnostics (logged in	'%Temp%\Roslyn' f	older)					
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 Debug Resources Services Settings	General Conditional compilation symbols: Define DEBUG constant Define TRACE constant Platform target:	Any CPU V
Reference Paths Signing Code Analysis	 Prefer 32-bit Allow unsafe code Optimize code 	Any Cro
Deprecated old code analysis	Errors and warnings Warning level: Suppress warnings:	4 ~
	Treat warnings as errors None All 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

Analyze Tools Extensions Window	Help	Search (Ctrl+Q)	م	PX.Objects.Ha
Code Cleanup 🔸		6 🔚 🚽 Align by	🛨 🛃 🔚 🕼 🗏	🧏 🔲 🖓 🖓
Run Code Analysis 🔹 🕨	•	On Solution		
Build and Suppress Active Issues	·	Run Code Analysis on	PX.Objects.Hackathor	Demo -
Configure Code Analysis	7.01	јестьпаскатнопоетно.	SOOIderwithinioid	
Calculate Code Metrics	•			
Windows •				



Integration with Other Tools



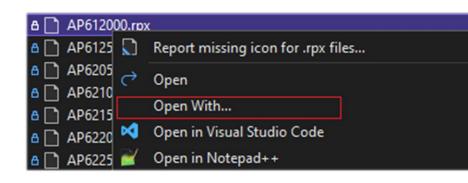
Different Integration Approaches

- □ 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\F
Use Output window	Prompt for argume	nts
Treat output as Unicod	e Close on exit	
	OK Cancel	Apply



External Tool Editor



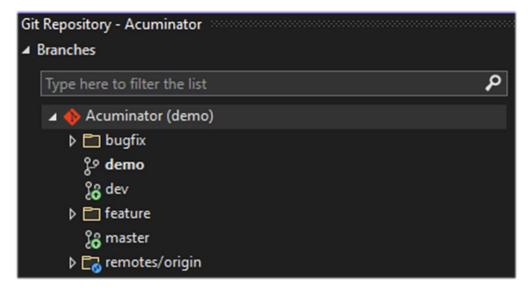
Open With - AP612000.rpx		? X
		Add external tool
Choose the program you want to use to open this file:		. 🕂
Common Language Editor Supporting TextMate Bundles (Default)	^	Add
JSON Editor		
JSON Editor with Encoding		Remove
Automatic Editor Selector (XML)		Kentove
XML (Text) Editor		Catalog Default
XML (Text) Editor with Encoding		Set as Default
Source Code (Text) Editor		
Source Code (Text) Editor With Encoding		Set selected
HTML Editor		editor as default
HTML Editor with Encoding		one for files with
HTML (Web Forms) Editor		
HTML (Web Forms) Editor with Encoding		this extension
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	▼ -⊐ X		
2021r212 Branch Se	elector - ↓↓ 1 C ···		
1 <u>0 outgoing / 0 incoming</u>	Common Actions Toolbar		
Enter a message <required></required>	Commit message		
Commit All 🔻 🗌 Amend	Button to commit or stash changes		
▲ Changes (5)	+		
🔺 🗁 C:\Repos2\Acumatical	xtraRepo\WebSites\Pure		
🔺 🗁 PX.Objects\CR			
C# OpportunityMai	nt.cs M		
🔺 🗁 Site			
🔺 🗁 App_Data			
🔺 🗁 SnapshotCon	figs		
🗋 Settings ar	nd Business Accounts.esc M		
💭 Customizatio	nStatus.xml M		
🔺 🗁 Pages\El			
EISC2200.aspx A			
C# EISC2200.asp>	A.cs		



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)	P	increasing according	min or enfact the grobal secting ratios.		^
Search Options (Ctrl+E) Environment Projects and Solutions Source Control Plug-in Selection Git Global Settings Git Repository Settings General Remotes Work Items Text Editor Debugging Performance Tools NET Core Debugging with WSL Acuminator Azure Data Lake		User name: Email: Prune remote bran Rebase local brand Tools Diff Tool: Visual St Merge Tool: Visual Git files Ignore file: /.gitigr	Sergey Nikomarov snikomarov@acumatica.com ches during fetch: True h when pulling: Unset udio Global Studio Global		
 Azure Service Authentication Container Tools 	~	Attributes file: /.gi	tattributes Edit		~
			ОК	Cance	I



Other Developer Tools



Alternatives to Visual Studio

Visual Studio Code

□ Notepad++

Jet Brains tools

- Rider IDE
- dotTrace and dotMemory for profiling
- dotPeek decompiler

Visual Studio Code



Rider dotTrace dotMemory dotMemory



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

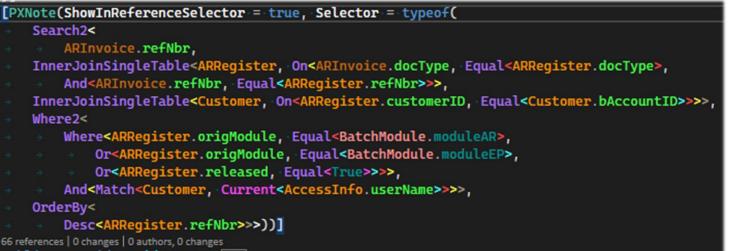




Visual Studio: Code Editor Appearance



Syntax Highlighting



public override Guid? NoteID

Search Options (Ctrl+E)	P	Show settings for:	Code Editor is chosen	in th	ne VS window picke	er
Environment	^	Text Editor			~	<u>U</u> se Defaults
General Accounts AutoRecover		Eont (bold type indicate Cascadia Mono	es fixed-width fonts): Font family and size c	can b	e customized 🗸	Size:
Documents Extensions Find and Replace		Display items: User Members - Extens		^	Item fo <u>r</u> eground:	<u>C</u> ustom
Fonts and Colors Import and Export Settings International Settings		User Members - Fields User Members - Labels User Members - Locals User Members - Metho	s C# extension s methods		Item bac <u>kg</u> round: Co	lor Picker _{Custom}
Keyboard Preview Features Product Updates Tabs and Windows		User Members - Name User Members - Param User Members - Proper User Types - Classes	spaces neters rties		Bold Turn on Bo	old Font Preview
Task List Terminal Trust Settings	~	User Types - Delegates User Types - Enums User Types - Interfaces		`	ij = I::oO(0	9xB81l);

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

11	/ <summary> The identifier of the Note obje</summary>	ect	associated	with	the	document.
[P]	XSearchable] 🛑 Collapsed attribute (Acuminator)					. 🔶 .
[P)	<pre>KNote(ShowInReferenceSelector = true, Selec</pre>	tor	= typeof(CO	llapsed
\rightarrow	Search2<				cor	mment
\rightarrow	ARInvoice.refNbr,					
	InnerJoinSingleTable <arregister, on<="">,</arregister,>	,				
÷.	→ And <>>,					
\rightarrow	InnerJoinSingleTable<>>,		llapsed BQL			
\rightarrow	Where2<>,	pa	rts (Acuminat	tor)		
÷ •	→ And<>>,					
\rightarrow	OrderBy<>>))]					
66 r	eferences 0 changes 0 authors, 0 changes					
pul	public override Guid? NoteID Collapsed curly braces					
#er	#endregion					
Re	RefNoteID 🛑 Collapsed region					

ג ה ה	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	۱.	Stop Hiding Current	Ctrl+M, Ctrl+U
	Outlining	Þ	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

Search Options (Ctrl+E)	Suggest usings for types in NuGet packages
 Environment Projects and Solutions Source Control Work Items 	 Add missing using directives on paste Quick Actions Compute Quick Actions asynchronously (experimental, requires restart)
Text Editor General Advanced Code Cleanup	Highlighting ✓ Highlight references to symbol under cursor ✓ Highlight related keywords under cursor
File Extension ▷ All Languages ▷ ASP.NET Web Forms ▷ Basic	Outlining Image: Contract of the second s
✓ 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 Code Style	Collapse #regions when collapsing to definitions



General Settings

Options	
Search Options (Ctrl+E)	Q
 Source Control Work Items Text Editor General Advanced Code Cleanup File Extension All Languages ASP.NET Web Forms Basic C# 	 Settings Drag and drop text editing Select subword on double click Automatic delimiter highlighting Track changes Auto-detect UTF-8 encoding without signature Follow project coding conventions Enable mouse click to perform Go to Definition Use modifier key: Ctrl v Open definition in peek view
General Scroll Bars Tabs Advanced ▷ Code Style IntelliSense ▷ C/C++ ▷ CSS ▷ Django Template ▷ Dockerfile ▷ F#	Display View whitespace Show bidirectional text control characters Selection margin Indicator margin Highlight current line Show structure guide lines Show error squiggles Show file health indicator Line spacing 1.0 •
 ▷ HQL ▷ HTML ▷ JavaScript/TypeScript ▷ JSON ▷ LESS ▷ Plain Text 	 Show editing context in the editor Line/Col Selections Insert/Overwrite Tab/Space 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

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

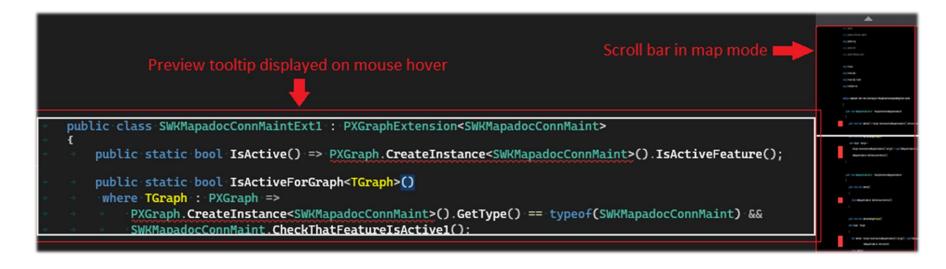
Options	
 ▷ Environment ▷ Projects and Solutions ▷ Source Control ▷ Work Items ▲ Text Editor General Advanced Code Cleanup File Extension ▷ All Languages ▷ ASP.NET Web Forms ▷ Basic ▲ C# 	atement completion Auto list members Hide advanced members Parameter information ettings Enable virtual space Word wrap Show visual glyphs for word wrap Line numbers Enable single-click URL navigation Navigation bar Automatic brace completion Apply Cut or Copy to blank lines when there is no selection



Scroll Bar

Search Options (Ctrl+E)	P	Show <u>h</u> orizontal scroll bar
Work Items Text Editor General Advanced Code Cleanup File Extension All Languages	^	 ✓ Show vertical scroll bar Display ✓ Show annotations over vertical scroll bar ✓ Show changes ✓ Show marks ✓ Show errors
ASP.NET Web Forms Basic C# General Scroll Bars]	 ✓ Show caret position Behavior O Use bar mode for vertical scroll bar ● Use map mode for vertical scroll bar
Tabs Advanced ✓ Code Style General ▷ Formatting	~	Show Preview <u>I</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

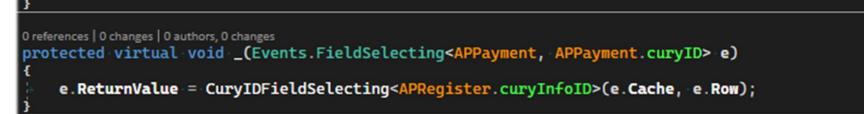




Line Separators

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.

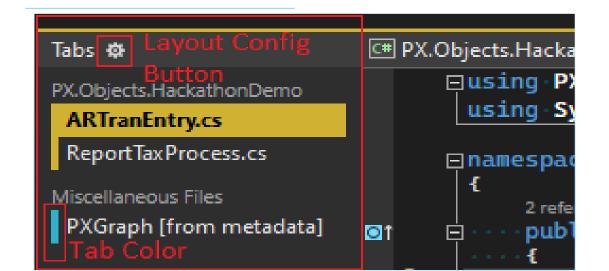


Switch to enable Line Separators is located in the Outlining settings

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



Document Tabs



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

- **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



