



# Acuminator

Acumatica-specific extension for Visual Studio



**Vladimir Panchenko**

Backend Team Lead

Acumatica

[vpanchenko@Acumatica.com](mailto:vpanchenko@Acumatica.com)

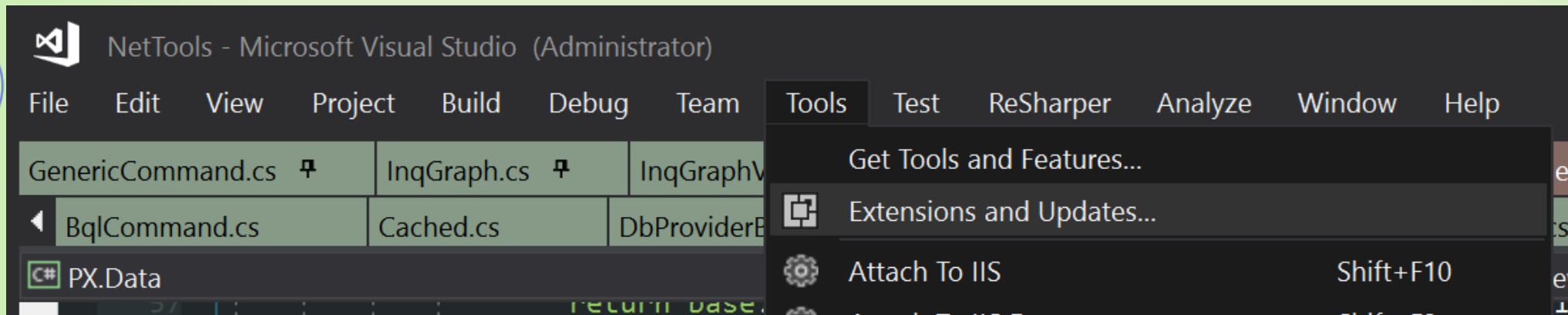
# What is Acuminator

- Extension (VSIX) for Visual Studio created specifically for the developers who use Acumatica Platform
- Supports both Visual Studio 2015 and 2017

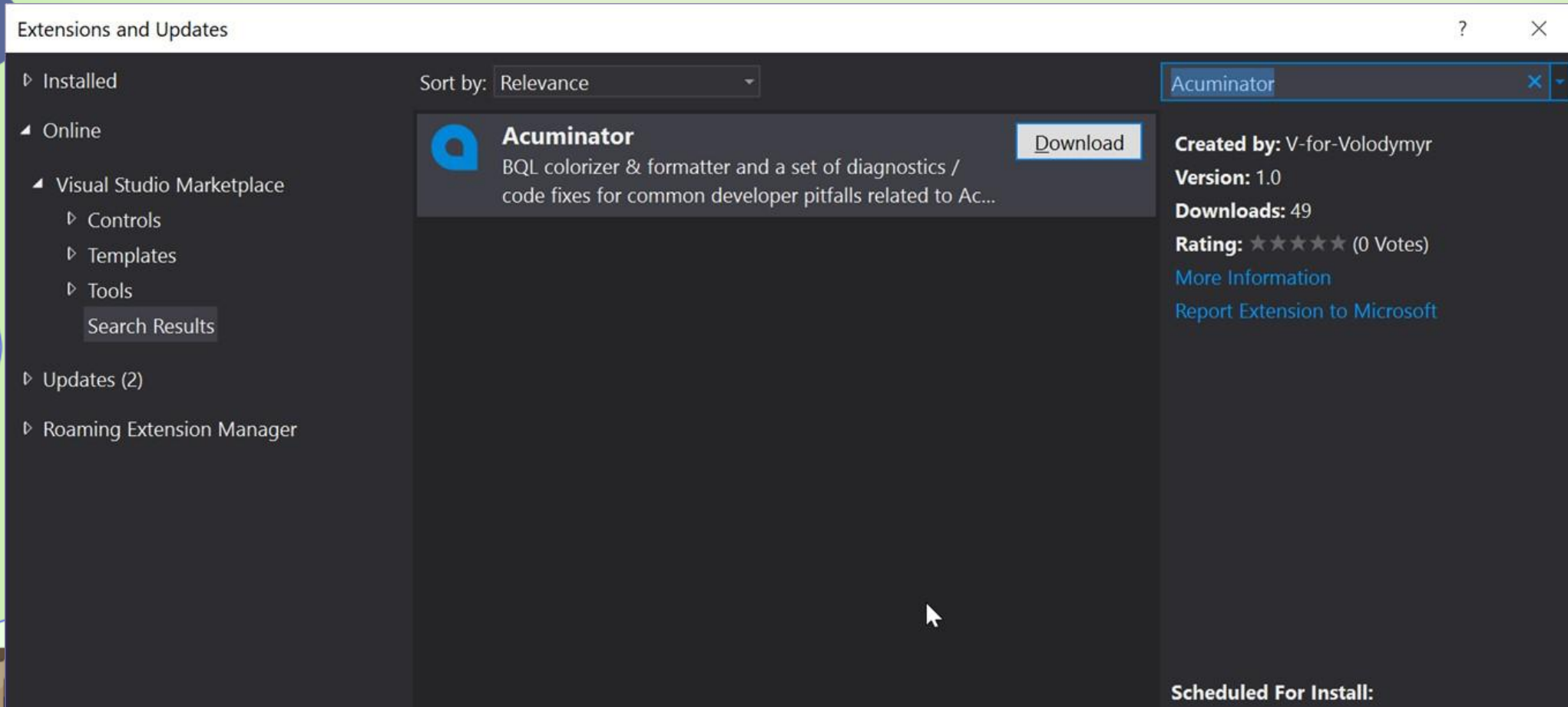


# How to install

- Open Visual Studio
- Go to Tools → Extensions and Updates...



# How to install



The screenshot shows the 'Extensions and Updates' window in Visual Studio. The left sidebar contains a tree view with categories: Installed, Online, Visual Studio Marketplace (with sub-items: Controls, Templates, Tools, Search Results), Updates (2), and Roaming Extension Manager. The main area displays search results for 'Acuminator', sorted by Relevance. The extension card for 'Acuminator' includes a blue circular icon, the name 'Acuminator', a description 'BQL colorizer & formatter and a set of diagnostics / code fixes for common developer pitfalls related to Ac...', and a 'Download' button. To the right of the card, a summary panel provides details: 'Created by: V-for-Volodymyr', 'Version: 1.0', 'Downloads: 49', 'Rating: ★★★★★ (0 Votes)', and links for 'More Information' and 'Report Extension to Microsoft'. At the bottom right, the text 'Scheduled For Install:' is visible.

Extensions and Updates

Sort by: Relevance

Acuminator

**Acuminator**  
BQL colorizer & formatter and a set of diagnostics / code fixes for common developer pitfalls related to Ac...

Download

Created by: V-for-Volodymyr  
Version: 1.0  
Downloads: 49  
Rating: ★★★★★ (0 Votes)  
[More Information](#)  
[Report Extension to Microsoft](#)

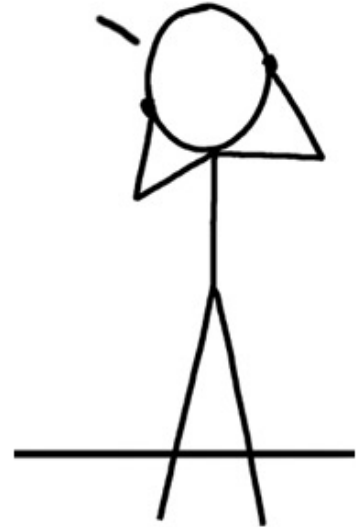
Scheduled For Install:



# Long & complex BQL queries are hard to read

```
public PXProcessingJoin<BalancedARDocument, LeftJoin<ARInvoice,
  On<ARInvoice.docType, Equal<BalancedARDocument.docType>,
  And<ARInvoice.refNbr, Equal<BalancedARDocument.refNbr>>>,
  LeftJoin<ARPayment, On<ARPayment.docType,
  Equal<BalancedARDocument.docType>, And<ARPayment.refNbr,
  Equal<BalancedARDocument.refNbr>>>, InnerJoinSingleTable<Customer,
  On<Customer.bAccountID, Equal<BalancedARDocument.customerID>>,
  LeftJoin<ARAdjust, On<ARAdjust.adjgDocType,
  Equal<BalancedARDocument.docType>, And<ARAdjust.adjgRefNbr,
  Equal<BalancedARDocument.refNbr>, And<ARAdjust.adjNbr,
  Equal<BalancedARDocument.adjCntr>, And<ARAdjust.hold,
  Equal<boolFalse>>>>>>>>>>, Where2<Match<Customer,
  Current<AccessInfo.userName>>, And<ARRegister.hold, Equal<boolFalse>,
  And<ARRegister.voided, Equal<boolFalse>, And<ARRegister.scheduled,
  Equal<boolFalse>, And<Where<BalancedARDocument.released, Equal<boolFalse>,
  And<BalancedARDocument.origModule, Equal<GL.BatchModule.moduleAR>,
  Or<BalancedARDocument.openDoc, Equal<boolTrue>, And<ARAdjust.adjRefNbr,
  IsNotNull>>>>>>>>>> ARDocumentList;
```

HOLY MOLY!  
INNER JOIN  
ON WHAT...  
WHERE... WHY?!



# Let's format them!

```
public PXProcessingJoin<
  BalancedARDocument,
  LeftJoin<ARInvoice,
    On<ARInvoice.docType, Equal<BalancedARDocument.docType>,
    And<ARInvoice.refNbr, Equal<BalancedARDocument.refNbr>>>>,
  LeftJoin<ARPayment,
    On<ARPayment.docType, Equal<BalancedARDocument.docType>,
    And<ARPayment.refNbr, Equal<BalancedARDocument.refNbr>>>>,
  InnerJoinSingleTable<Customer,
    On<Customer.bAccountID, Equal<BalancedARDocument.customerID>>,
  LeftJoin<ARAdjust,
    On<ARAdjust.adjgDocType, Equal<BalancedARDocument.docType>,
    And<ARAdjust.adjgRefNbr, Equal<BalancedARDocument.refNbr>,
    And<ARAdjust.adjNbr, Equal<BalancedARDocument.adjCntr>,
    And<ARAdjust.hold, Equal<boolFalse>>>>>>>>>,
  Where2<Match<Customer, Current<AccessInfo.userName>>,
    And<ARRegister.hold, Equal<boolFalse>,
    And<ARRegister.voided, Equal<boolFalse>,
    And<ARRegister.scheduled, Equal<boolFalse>,
    And<Where<BalancedARDocument.released, Equal<boolFalse>,
      And<BalancedARDocument.origModule, Equal<GL.BatchModule.moduleAR>,
      Or<BalancedARDocument.openDoc, Equal<boolTrue>,
      And<ARAdjust.adjRefNbr, IsNotNull>>>>>>>>>>>>
  ARDocumentList;
```

HMM... NOW IT  
MAKES SENSE...



# ...and add some color!

```
public PXProcessingJoin<
    BalancedARDocument,
    LeftJoin<ARInvoice,
        On<ARInvoice.docType, Equal<BalancedARDocument.docType>,
        And<ARInvoice.refNbr, Equal<BalancedARDocument.refNbr>>>>,
    LeftJoin<ARPayment,
        On<ARPayment.docType, Equal<BalancedARDocument.docType>,
        And<ARPayment.refNbr, Equal<BalancedARDocument.refNbr>>>>,
    InnerJoinSingleTable<Customer,
        On<Customer.bAccountID, Equal<BalancedARDocument.customerID>>>,
    LeftJoin<ARAdjust,
        On<ARAdjust.adjgDocType, Equal<BalancedARDocument.docType>,
        And<ARAdjust.adjgRefNbr, Equal<BalancedARDocument.refNbr>,
        And<ARAdjust.adjNbr, Equal<BalancedARDocument.adjCntr>,
        And<ARAdjust.hold, Equal<boolFalse>>>>>>>>>>,
    Where2<Match<Customer, Current<AccessInfo.userName>>>,
        And<ARRegister.hold, Equal<boolFalse>,
        And<ARRegister.voided, Equal<boolFalse>,
        And<ARRegister.scheduled, Equal<boolFalse>,
        And<Where<BalancedARDocument.released, Equal<boolFalse>,
            And<BalancedARDocument.origModule, Equal<GL.BatchModule.moduleAR>,
            Or<BalancedARDocument.openDoc, Equal<boolTrue>,
            And<ARAdjust.adjRefNbr, IsNotNull>>>>>>>>>>>>>>>>
    ARDocumentList;
```

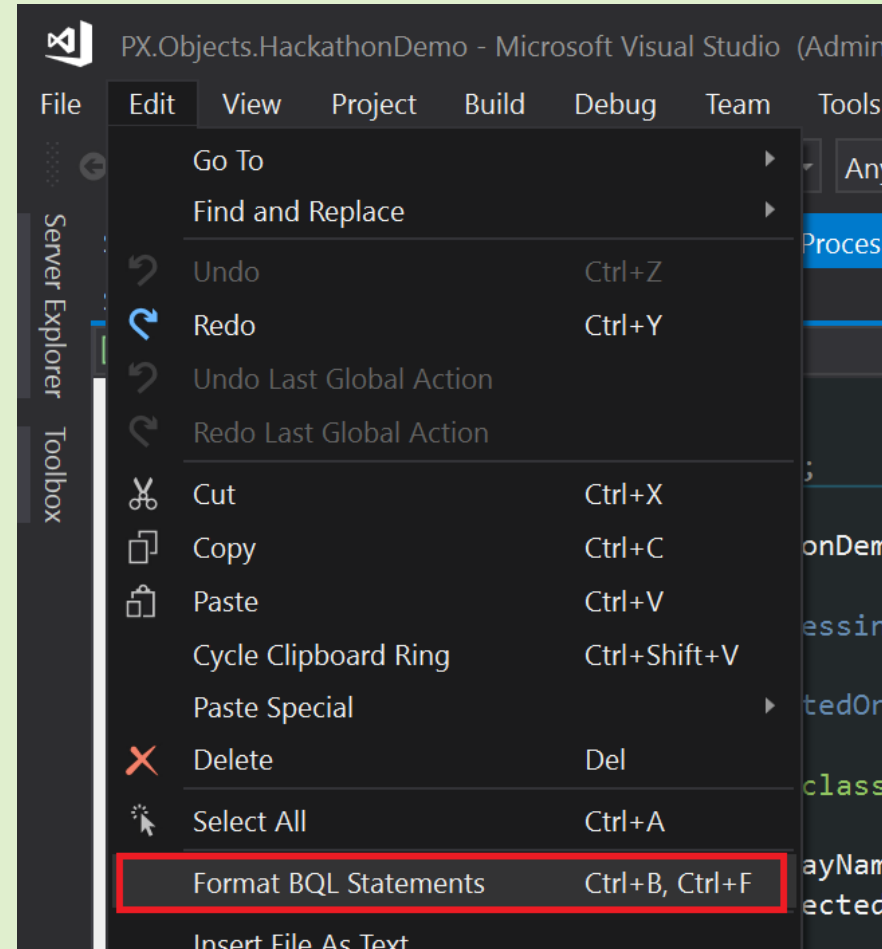
OH YEAH!





# BQL Formatter

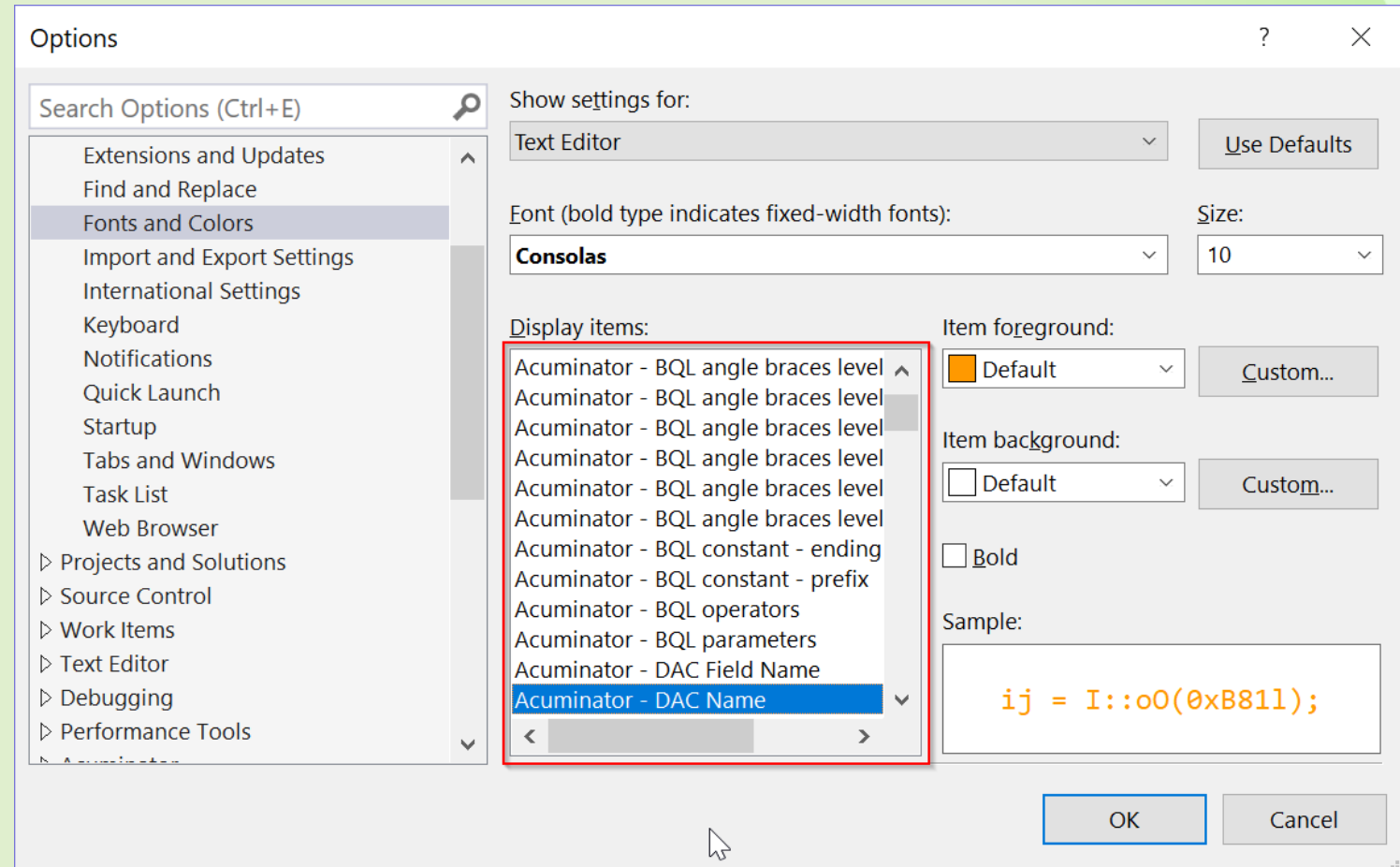
- Edit → Format BQL Statements (Ctrl+B, Ctrl+F)
- Works both for entire document or for selection



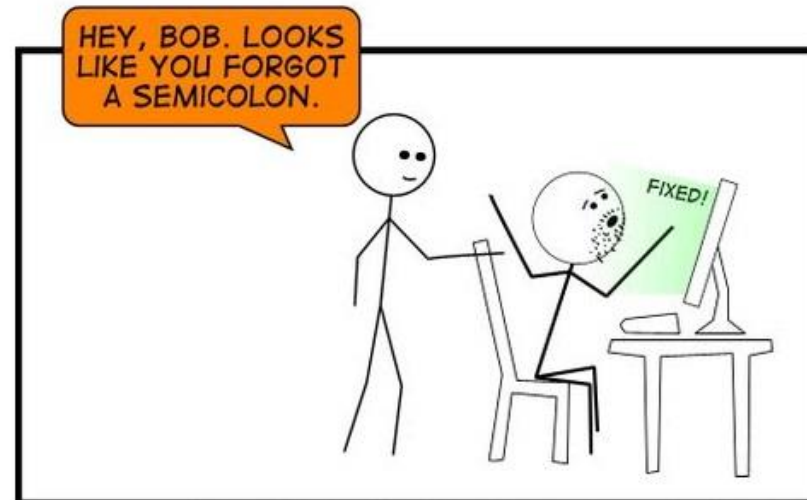
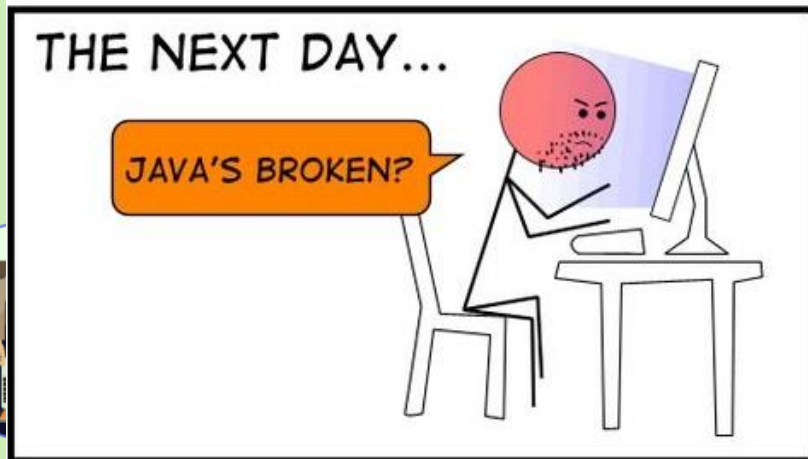
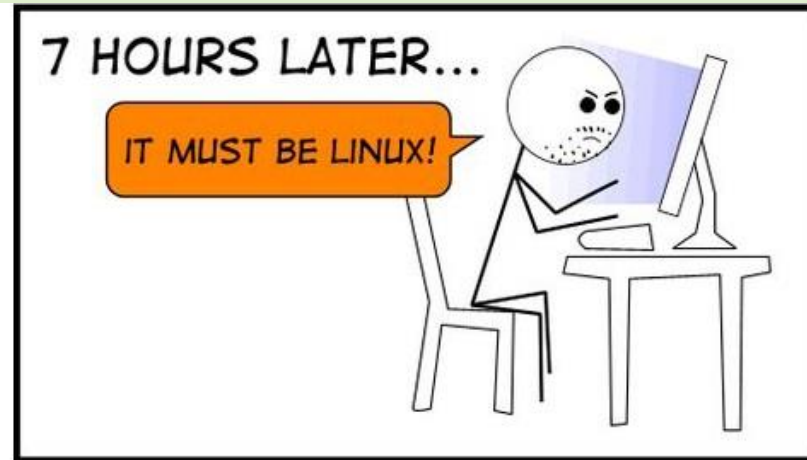
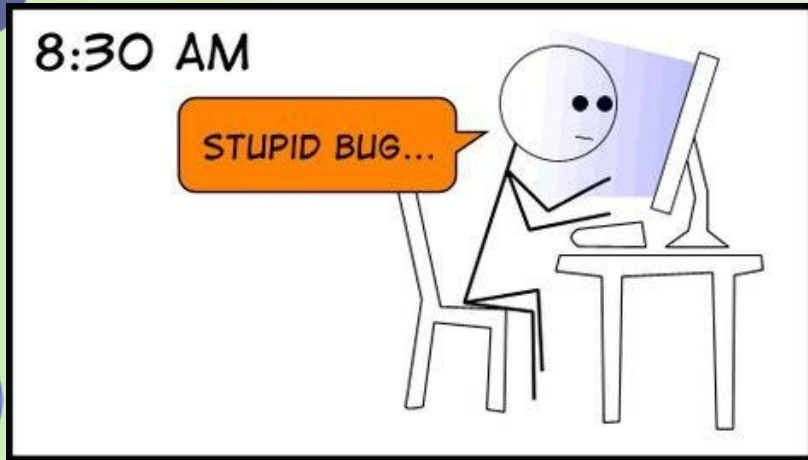


# BQL Colorizer

- Supports both dark and light color themes
- You can customize the colors using Tools → Options window (Environment → Fonts and Colors section)



# Every developer faces those kind of problems...



# What Acuminator suggests in this area

- Set of Acumatica xRP-specific diagnostics and code fixes
- Fully integrated with Visual Studio code analysis engine
- Runs analysis and shows suggestions/warnings/errors “on the fly” as you write the code
- Provides quick fixes for errors found in the code
- Allows you to run global analysis on existing solutions

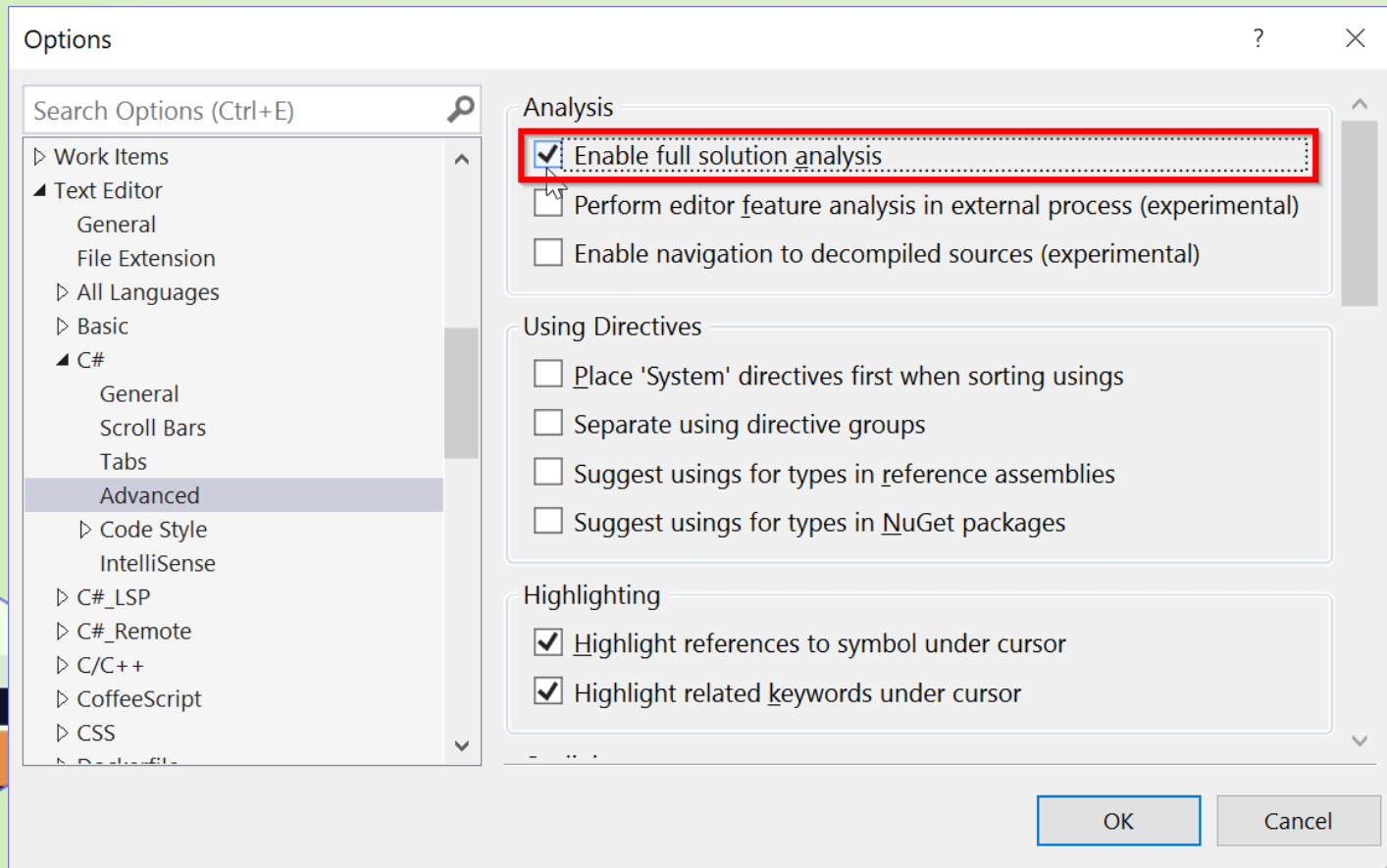


# Live Demo



# How to analyze existing solution

- Tools → Options → Text Editor → C# → Advanced → Enable Full Solution Analysis



# Acuminator is Open Source

- Has been published to GitHub
- Anyone can extend it, add new analyzers / code fixes, take a look at the code, etc.
- Repository: <https://github.com/Acumatica/Acuminator>



# Future Plans

- More static code checks and code fixes
- Shortcut for navigation between view declaration and view delegate, PXAction declaration and its handler
- Context help by F1 for Acumatica-related classes
- Static code analyzers and code fixes as NuGet package
- Feel free to suggest new ideas on <http://feedback.acumatica.com>





# Thank You!

<https://adn.Acumatica.com>

