SYNERGEX

TECH DAYS 2019

Developing in Visual Studio

Developing in Visual Studio

- Fundamental capabilities
- What’s new in Visual Studio 2019
- Synergy DBL Integration for Visual Studio (SDI)
- Migrating development to Visual Studio
- Tips & tricks

Visual Studio - Fundamental Capabilities

Visual Studio >

SYNERGEX

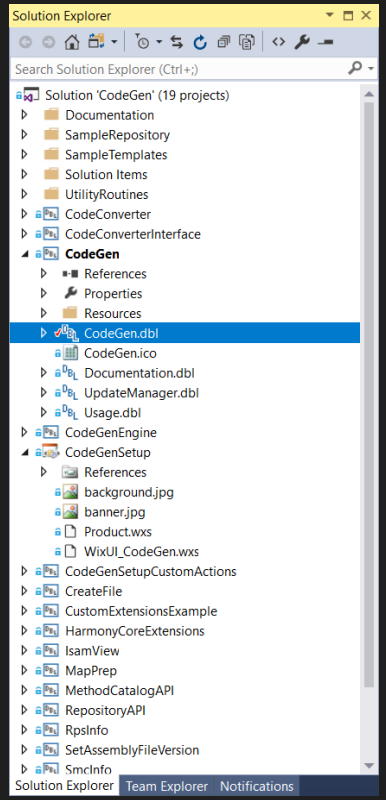
Everything in One Place

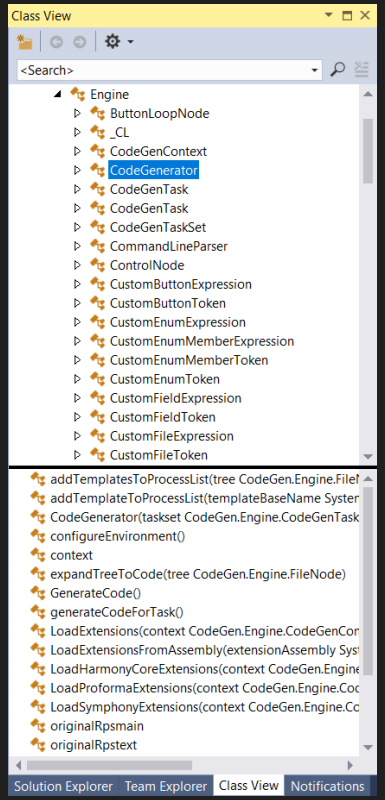
- Development models
  - Synergy .NET
    - .NET Framework
    - .NET Core
  - Traditional Synergy
    - Windows
    - Linux / UNIX
    - OpenVMS
- Project templates
  - Pre-configured starter solutions for a wide range of projects / targets
- Mix & match in a single solution
  - Project types
    - Library
    - Desktop
    - Web
    - Device
  - Architectures
    - Windows
    - Linux
    - Android / iOS
  - Languages
    - Synergy .NET
    - Traditional Synergy
    - Non-Synergy

SYNERGEX

Project Management & Navigation

- Solution Explorer
  - Projects, individual items & solution folders
  - Search: filter by keyword / type
  - Keyboard shortcuts
  - Folder view
- Class View … navigate via
  - Assembly, namespace, class, member, etc.
  - Searchable





SYNERGEX

Advanced Editor Capabilities

- Language-sensitive color coded editor
- IntelliSense
  - Completion lists & parameter info
  - Code snippets
  - Code generation
    - Implement interface, implement abstract base class, etc.
- Code Navigation
  - GOTO definition
  - GOTO references, and MUCH more
- Productivity tips
  - Quick actions (light bulb & screwdriver)
- Advanced search and replace
  - File, files, project, solution, folder

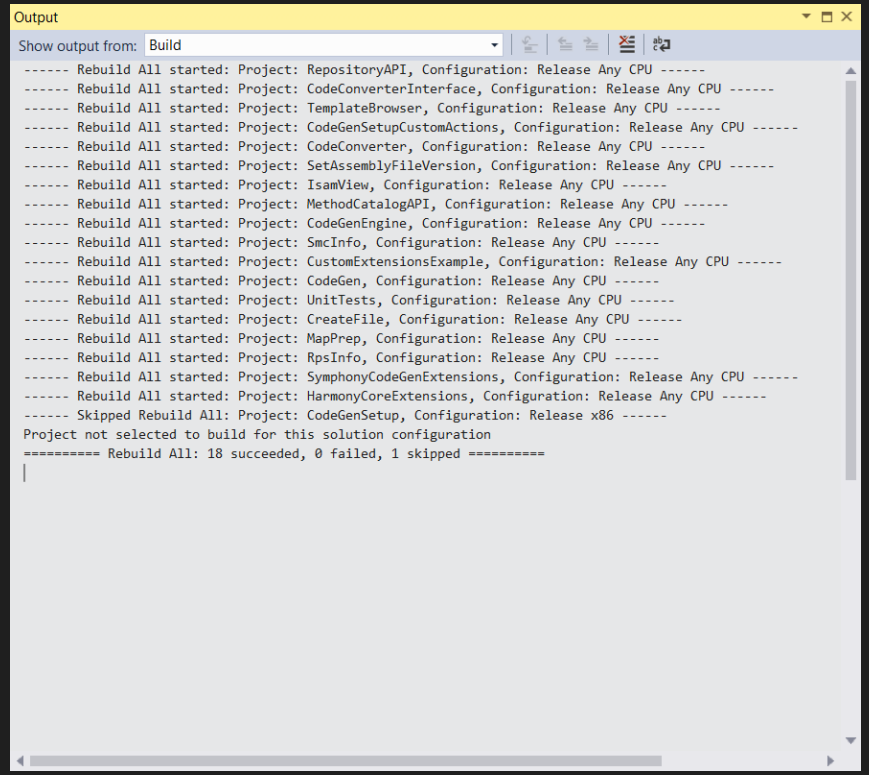
CustomersController.dbl

```
25  namespace Services.Controllers
26
27      ;;; <summary>
28      ;;; OData controller for Customers
29      ;;; </summary>
30      public partial class CustomersController extends ODataController
31
32          public readwrite property DBContext, @Services.Models.DBContext
33          public readwrite property ServiceProvider, @IServiceProvider
34
35          ;;; <summary>
36          ;;; Constructs a new instance of CustomersController
37          ;;; </summary>
38          ;;; <param name="aDbContext">Database context instance (DI)</param>
39          ;;; <param name="aServiceProvider">Service provider instance (DI)</param>
40          public method CustomersController
41              aDbContext, @Services.Models.DBContext
42              aServiceProvider, @IServiceProvider
43          proc
44              this.DBContext = aDbContext
45              this.ServiceProvider = aServiceProvider
46          endmethod
47
48          {ODataRoute("Customers")}
49          {EnableQuery(MaxExpansionDepth=4)}
50          ;;; <summary>
51          ;;; Get all Customers
52          ;;; </summary>
53          ;;; <returns>Returns an IActionResult indicating the status of the operati
54          public method GetCustomers, @IActionResult
55          proc
56              mreturn Ok(DBContext.Customers.AsNoTracking())
57          endmethod
```

SYNERGEX

Dependency-Based Build

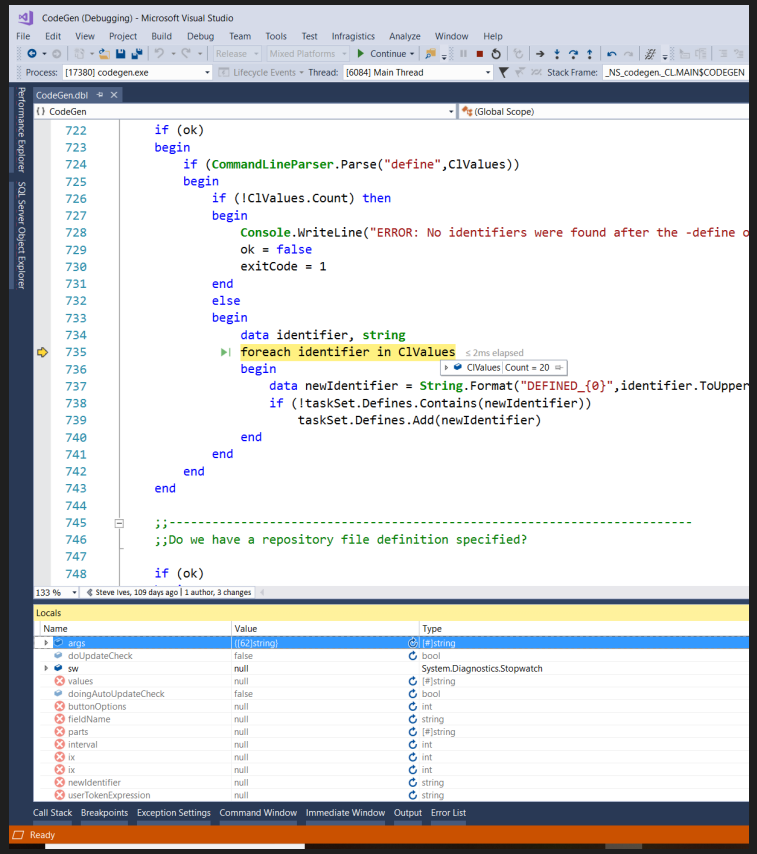
- Knows how to build your code
  - No more shell scripts & batch files
  - Only build what needs to be built
- Driven by MSBUILD
  - Define projects for OLBs, ELBs and class libraries
  - “Reference” those projects from others
  - MSBUILD does the rest
- Build Only Project
  - Don’t build dependencies
  - Traditional Synergy and .NET
- Working on CRC-based improvements for traditional Synergy ELBs



SYNERGEX

In-Line Debugging

- Debug display integrated with editor
- Yellow arrow indicates next statement
  - Hover-over variables for value
  - Locals window
  - Immediate window
- Keyboard & mouse control
  - Start debugging F5
  - Step over F10
  - Step into F11
  - Step out Shift + F11
  - Back to next statement Alt + Num *
  - Stop debugging Shift + F5



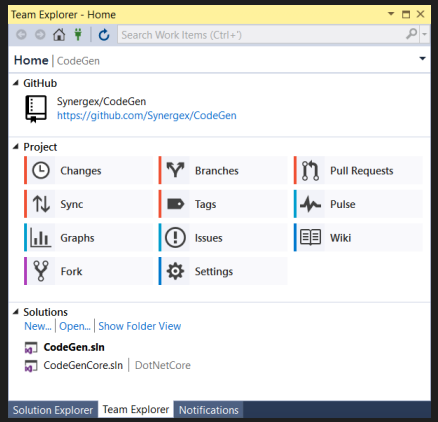
SYNERGEX

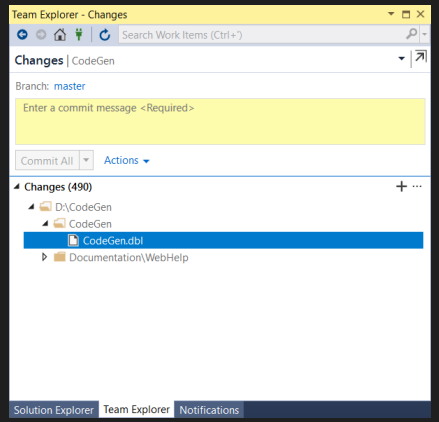
Integrated Source Control

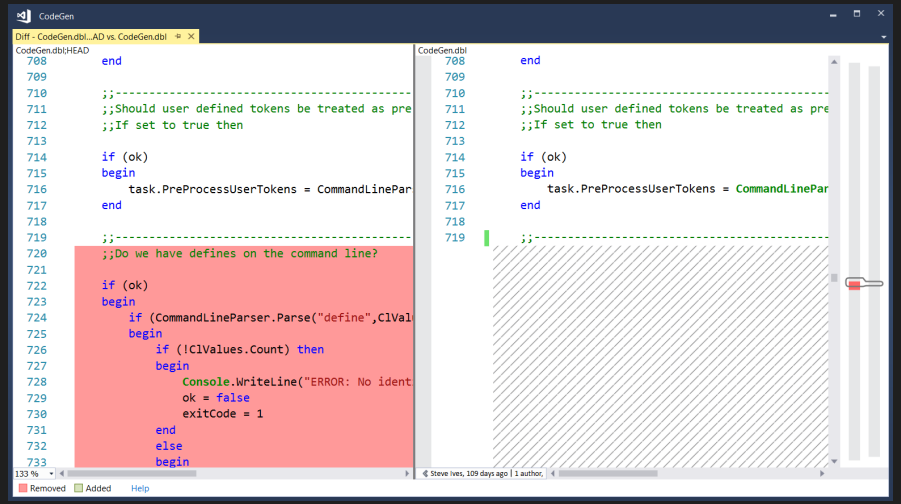
- Out of the box
- Team foundation server
- Install option (Git)
- Default in 2019

- Free plugins for most other SCM products

- Access all significant SCM functionality via Team Explorer (TFS & Git)
- Or custom UI (others)



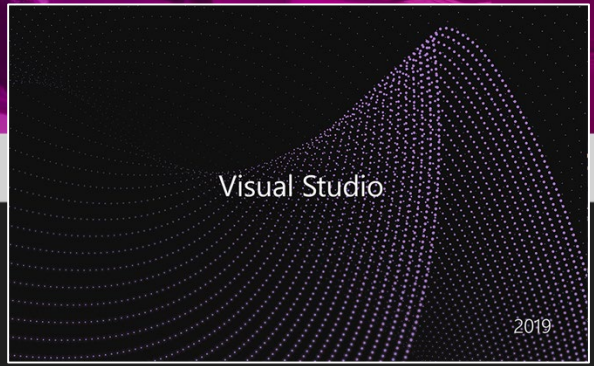




SYNERGEX

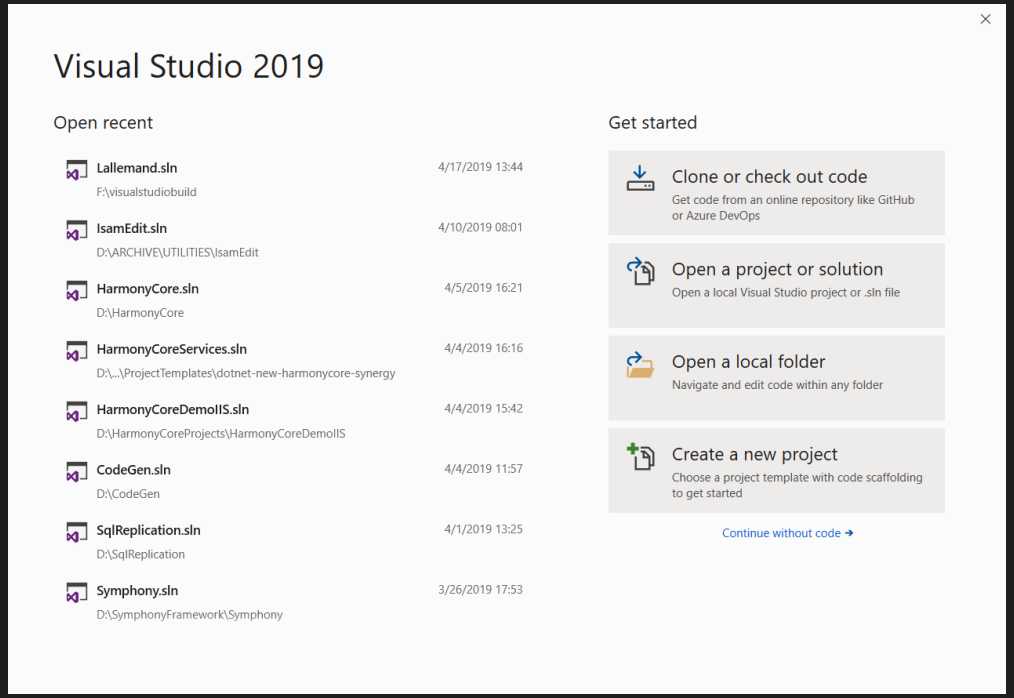
SYNERGEX

TECH DAYS 2019



What’s new in Visual Studio 2019

New Launch Dialog Replaces Start Screen

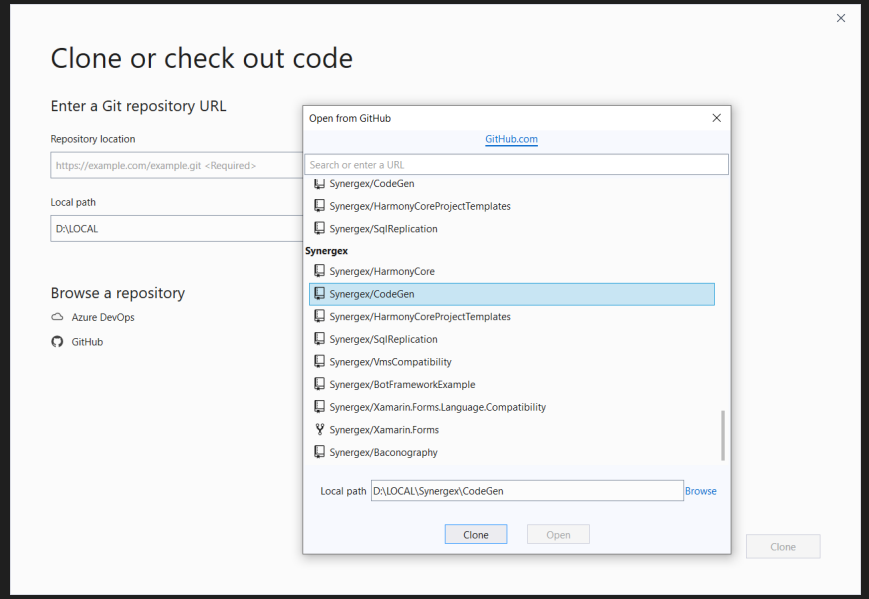


- Recent solutions
- More options for getting started with new things
- Can be disabled in Tools > Options > Environment > Startup

SYNERGEX

Clone or Checkout Code

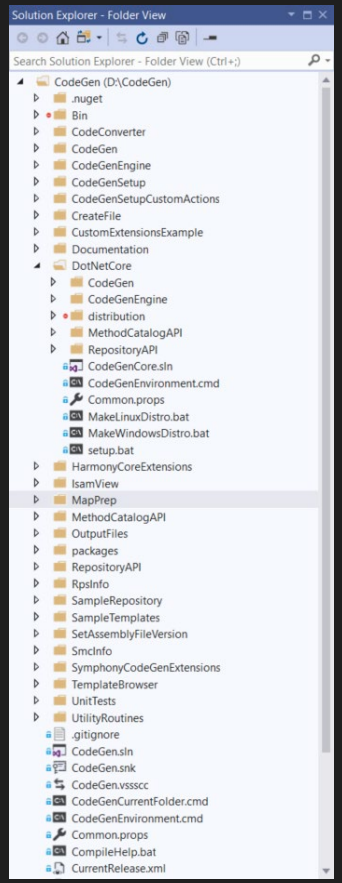
- Directly access / clone SCM repositories
- Browse support for Microsoft services
  - Azure DevOps
  - GitHub
- Many less steps than previous mechanisms



SYNERGEX

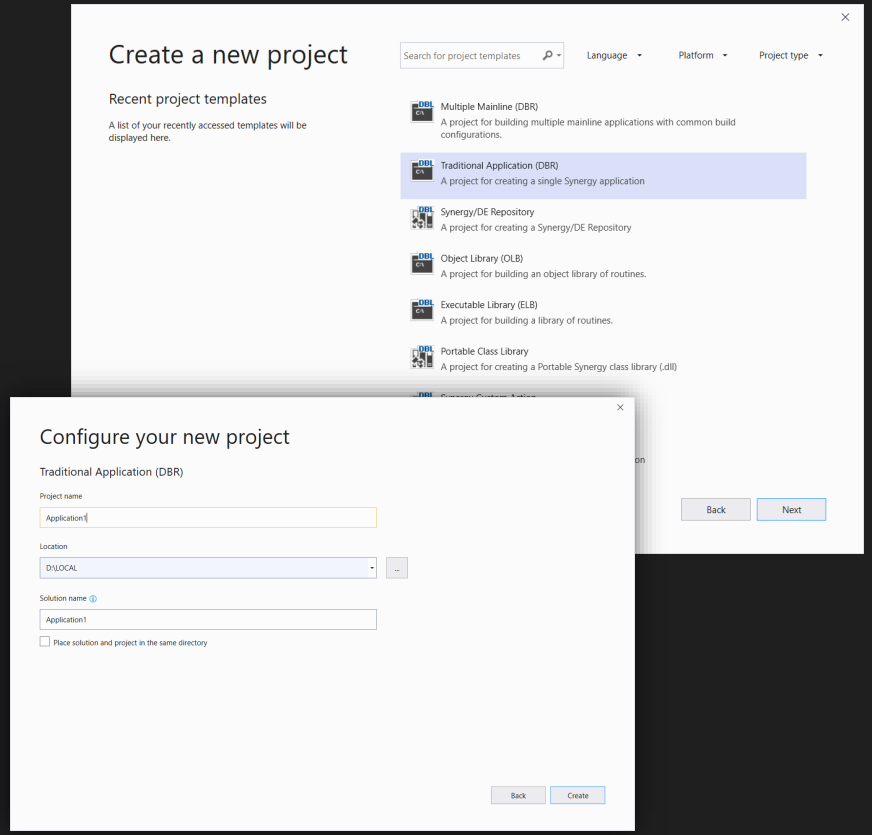
Open a Local Folder

- Directly open a folder instead of a solution
  - Puts Solution Explorer in Folder mode
- Working in non .NET environments
  - Web development with HTML5 / CSS / JS
- Browsing around, independent of projects
- Double-click to open solution
  - Solution Explorer can switch back and forth between folder and solution view
  - Toolbar button



SYNERGEX

Creating New Projects

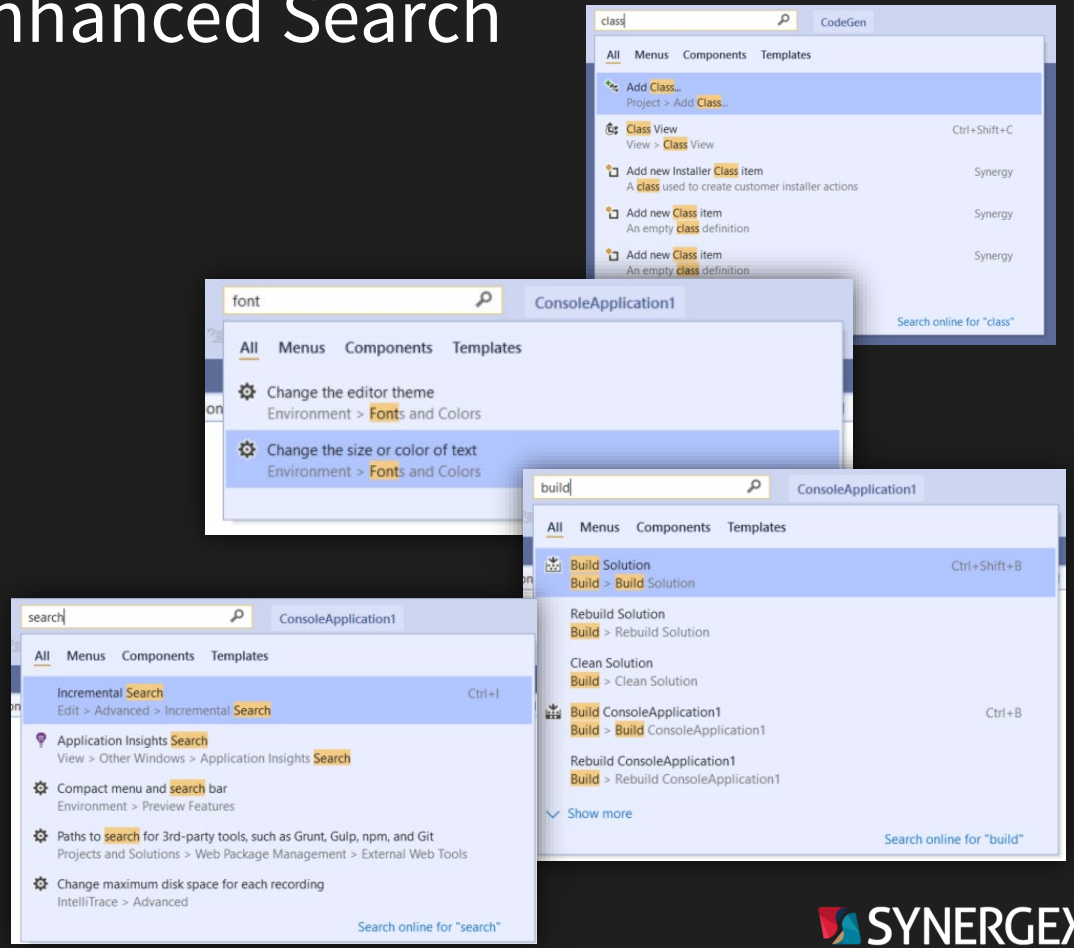


- Replaces the old project templates dialog
- Filters (top-right)
  - Language
  - Platform
  - Project Type
- Multi-page wizard UI
  - Select project language & type
  - Configure project options
- Currently no 3rd party extensibility
  - No “Synergy” language filter
  - Search for “synergy” or scroll to bottom!

SYNERGEX

Visual Studio 2019: Enhanced Search

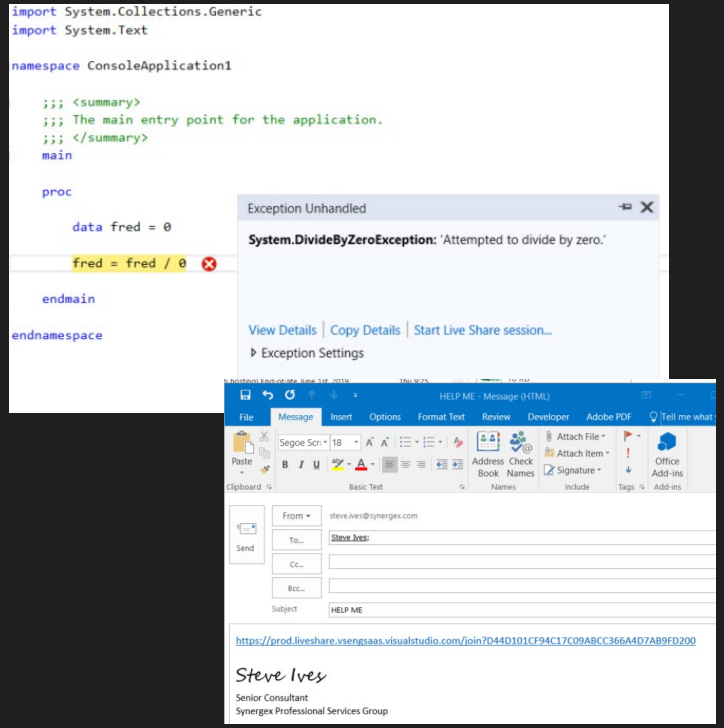
- Search Visual Studio
  - Menu items
  - Templates
  - Tool windows
  - Settings, etc.
- Fuzzy search logic
- Replaces Quick Launch, significantly enhanced
- Example: “Class”
  - Add new class



SYNERGEX

Visual Studio 2019: Collaboration

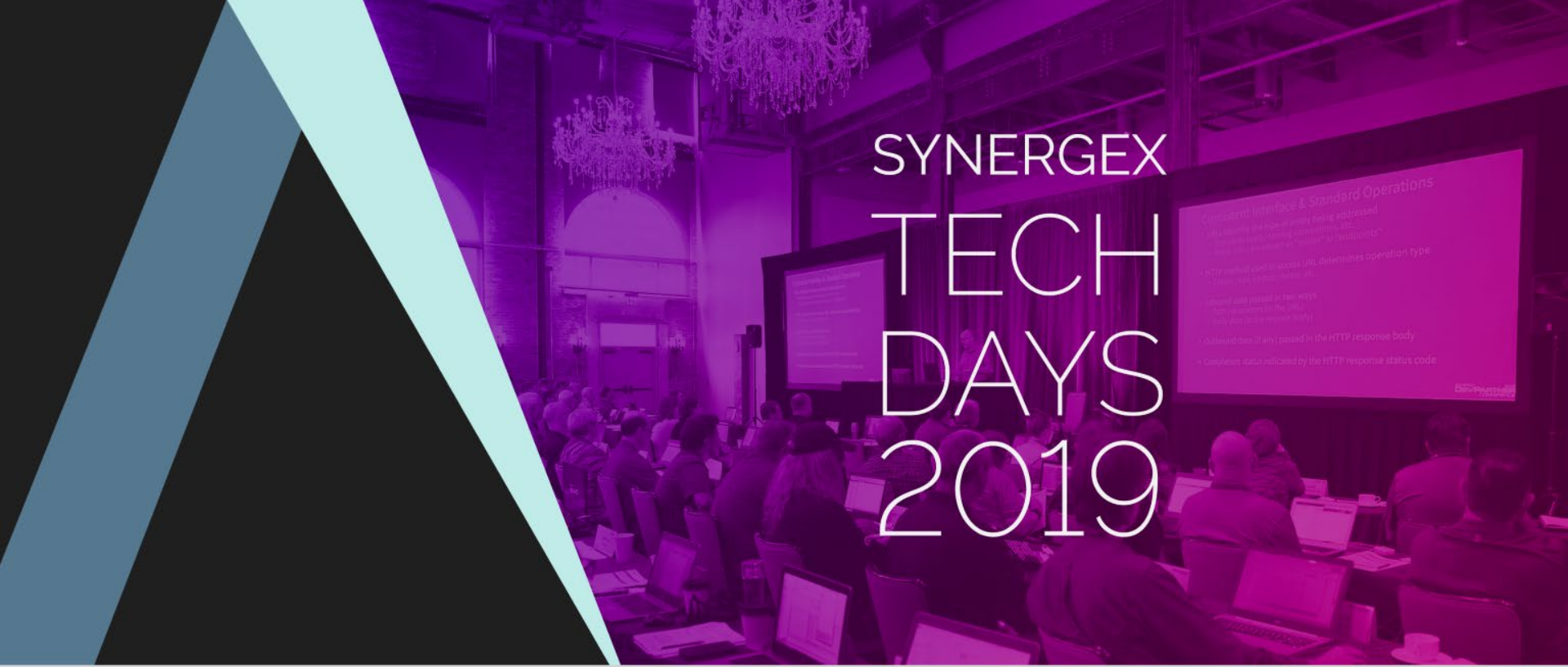
- Start Live Share session
  - Allows other developer(s) to view & interact with your Visual Studio session
  - Both can work in real time
  - Shared debugging experience
- Both developers must have Visual Studio 2019
- Integrated code reviews



SYNERGEX

Visual Studio 2019: Miscellaneous Changes

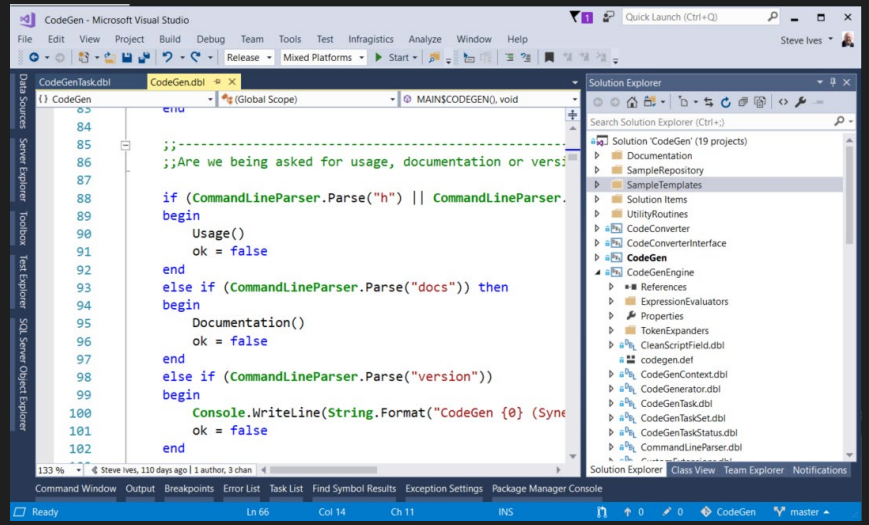
- GitHub extension now installed by default
- New extension “Pull Requests for Visual Studio”
  - Adds pull request processing into Team Explorer
- Per-monitor awareness (PMA)
  - Code now appears crisp and clear in any monitor display scale factor and DPI (dots per inch) configuration, including across multiple monitors
- Moderate speed improvements (maybe 5-10%)
  - Mainly in solution load times
- VS19 can coexist with 17



Synergy DBL Integration for Visual Studio (SDI)

Synergy Development in Visual Studio

- Full-fidelity Synergy development in the Visual Studio IDE
  - Synergy .NET
  - .NET Core
  - Traditional Synergy
- Edit, build, test, run, debug, SCM
- Develop on Windows, deploy anywhere
  - Binary compatibility with Linux
- Developer productivity
- Attract to new developers



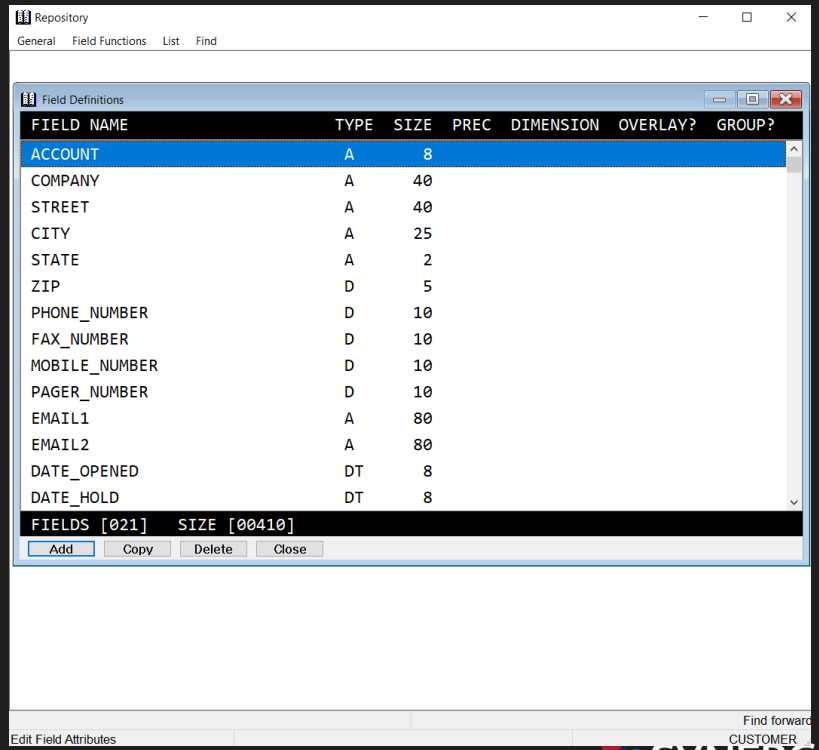
SYNERGEX

Synergy/DE Project Templates

- Synergy .NET Applications
  - Windows Forms Application
  - WPF Application
  - Windows Service
  - Console App
- Synergy .NET Libraries
  - Class library
  - Portable Class Library
  - Windows Forms Control Library
  - WPF User Control Library
  - Interop Project
  - Unit Test Project
  - WCF Service Library
  - Synergy Custom Action (for installers)
- .NET Standard
  - Class Library
- .NET Core
  - Class Library
  - Console App
- Traditional Synergy
  - Object Library (OLB)
  - Executable Library (ELB)
  - Traditional Application (DBR)
  - Multiple Mainline (DBR)
  - Synergy/DE Repository

Repository Projects

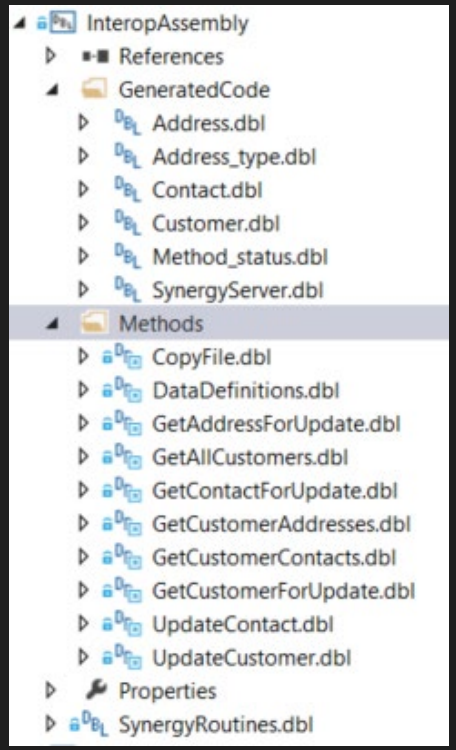
- Project designed to manage & expose your repository
- Two modes of operation
  - Schema first
    - You edit the schema
    - Visual Studio loads the schema to Repository ISAM files
  - UI first
    - You manage the repository ISAM files via the UI
    - Visual Studio exports the schema
- Reference project from others to access repository data



SYNERGEX

Interop Projects

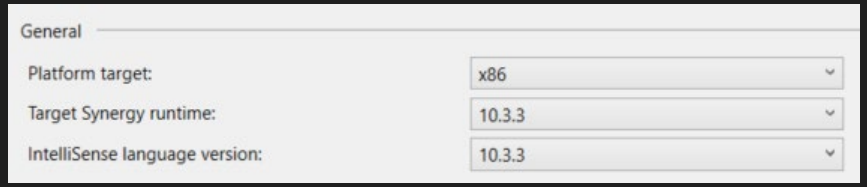
- Migration from xfServerPlus to native .NET
- Input
  - Synergy methods previously exposed via *xf*ServerPlus
  - Must have {xfMethod} and {xfParameter} attributes as necessary
- Output
  - Synergy .NET classes that WRAP the method code
  - External API is <u>IDENTICAL</u> to C# code produced by GENCS
  - All code runs <u>IN-PROCESS</u>



SYNERGEX

Runtime Version Targeting

- Targeting specific versions of the Synergy runtime
  - Prevents use of features requiring runtime support in a later version



- ALWAYS use the latest development tools!

- Traditional Synergy
  - Target back to 9.5.1

- Synergy .NET
  - Target back to 10.1.1

SYNERGEX

Environment Variables

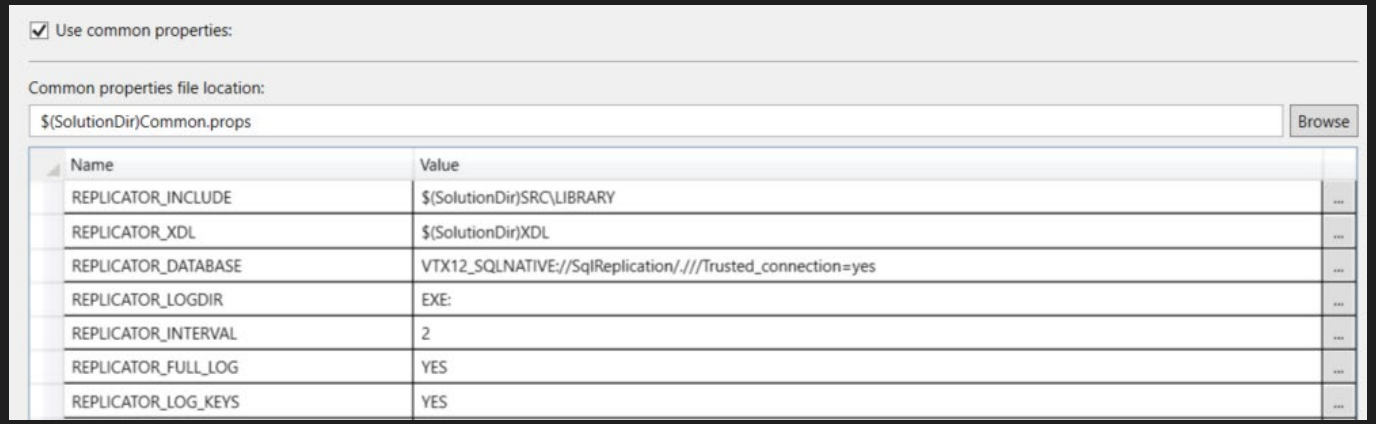
Project environment variables:

Name	Value
EXE	$(SolutionDir)EXE
DAT	$(SolutionDir)DATA
APPOPTS	1,4,6,17
*	

- Synergy code tends to rely heavily on environment variables
- Mechanism to embed definitions into the development environment
  - No need for complex environment setup batch files
- Project specific, used at project COMPILE time

SYNERGEX

Common Properties

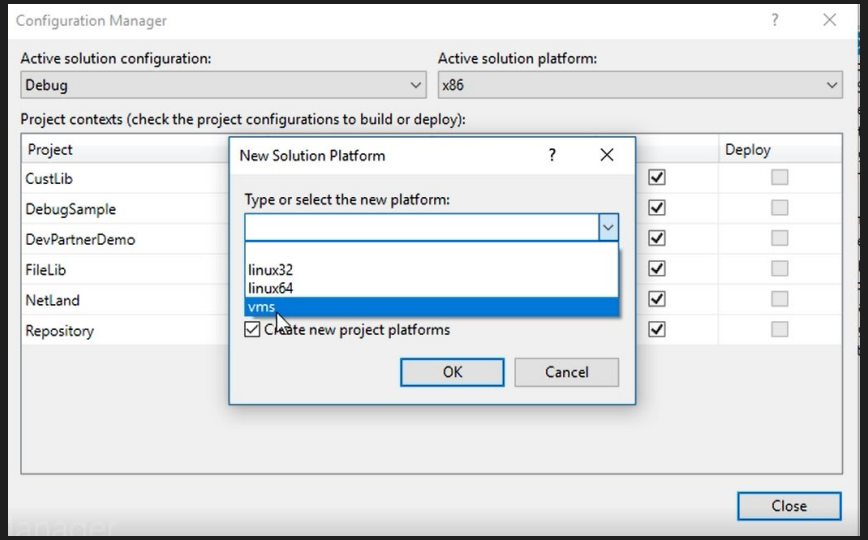


- Alternate way to define environment variables in the development environment
- Shared between multiple (or all) Synergy projects (projects opt-in)
- Used at COMPILE time, and also runtime when run from Visual Studio

SYNERGEX

Defining Target Platforms

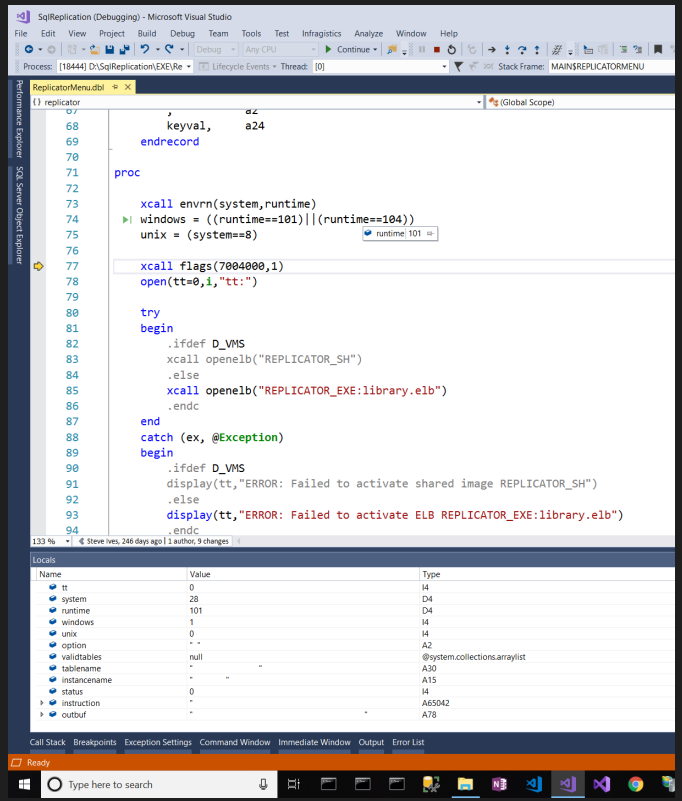
- Use Configuration Manager to define additional platforms
  - Linux 32-bit (linux32)
  - Linux 64 (linux64)
  - OpenVMS (vms)
- Define custom settings for that target in project properties (build, etc.)
- Compiler behaves as if running on the designated system
  - Platform specific defines (OS_VMS etc.)
  - Platform specific API's

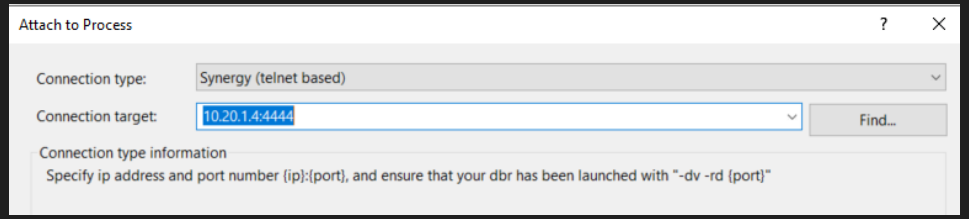


SYNERGEX

Traditional Synergy Debugging

- Same experience as .NET debugging
- Drives the original traditional Synergy debugger via it's Telnet capability
  - dbr –rd <port>:timeout
  - Telnet to the port before the timeout
- Remote debugging of processes on other systems
  - Debugging code on Unix, Linux or OpenVMS
  - Debugging *xf*ServerPlus methods
  - Accessed via Debug > Attach to Process





SYNERGEX

New to Visual Studio in Synergy 11

- Performance improvements & bug fixes
  - Lots and lots and lots of them!
- Synergy options dialog improvements
  - Improved code formatting options & display
  - Synergy file extensions now implemented as a list
- Support “Visual Studio Build Tools”
  - Alternate install for build servers
  - No Visual Studio IDE integration
- Settings export / import now includes Synergy settings

- Dropped support for Visual Studio 2015

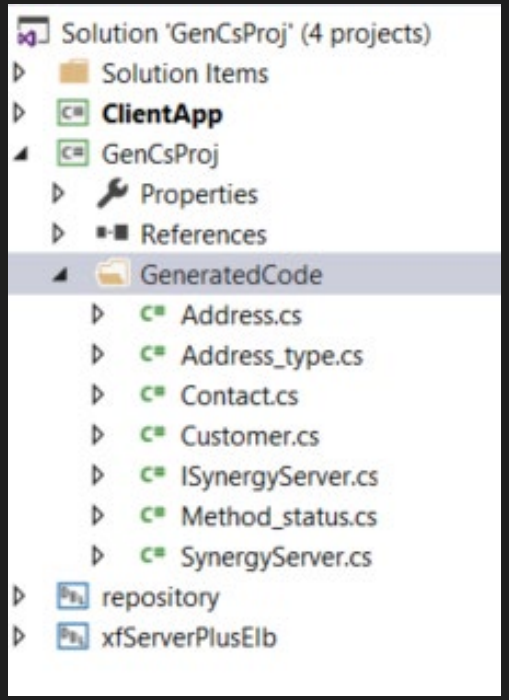
Coming Soon: *xf*NetLink .NET Projects

- Conceptually similar to Workbench xfNetLink .NET Assembly Project

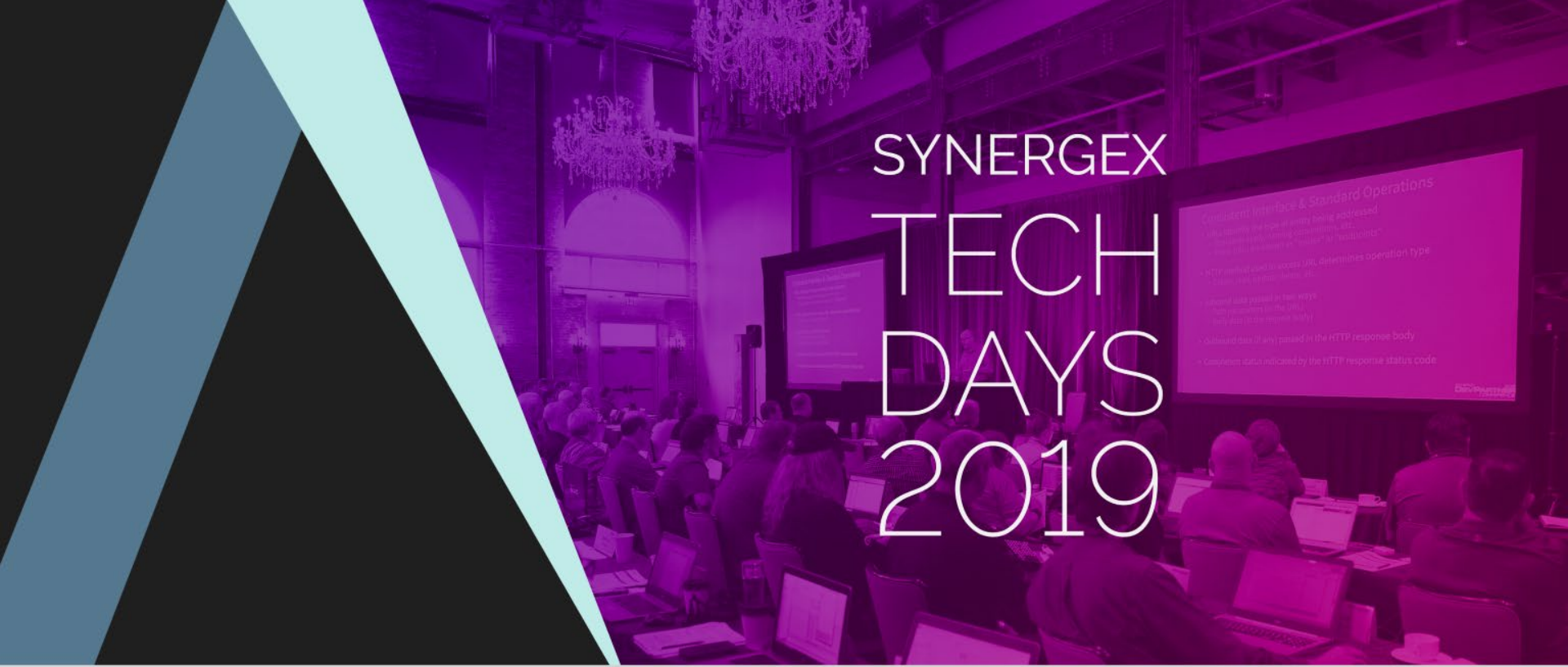
- Inputs
  - Synergy Method Catalog
    - In ISAM or XML form
  - Repository
  - Metadata
    - Interface(s), # classes, etc.

- Output
  - xfNetLink .NET C# code
  - Builds client assembly

- Prototype exists, looking for feedback on functionality & demand



SYNERGEX



Migrating Development to Visual Studio

Initial Project Setup

- If you're using Workbench AND using "modern" workbench projects
  - Synergy/DE Object Library, Executable Library & Application
  - SYN2VS utility can do most of the work for you
  - Converts Workbench workspace & projects to Visual Studio Solution & projects

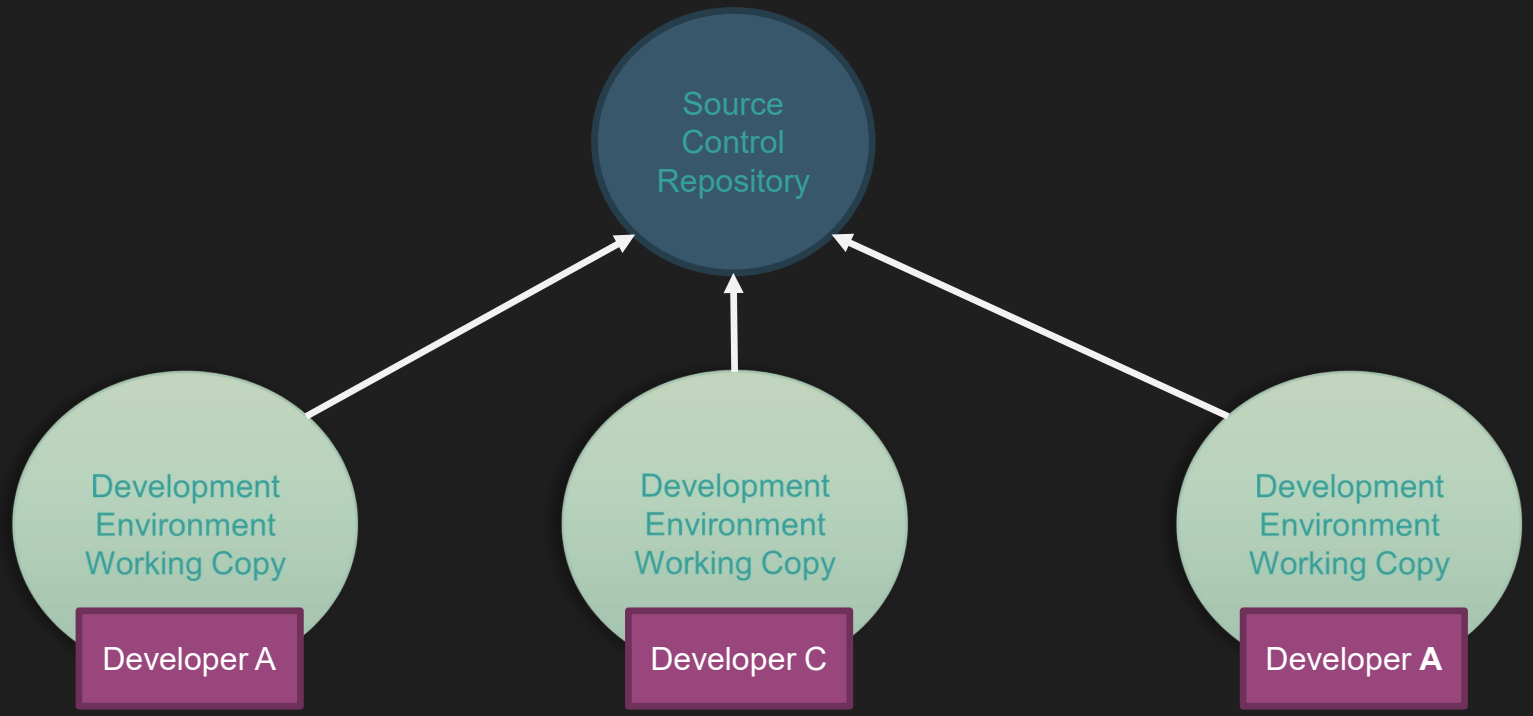
- Otherwise
  - Manually create solutions and projects
  - Not as bad as it sounds, may be able to leverage file lists from project files and / or build scripts
  - Depending on # projects, might take a few days

Migrating Development to Visual Studio

- Building in Visual Studio REQUIRES strong prototyping
  - -qrelax may be used
  - This is the hardest part, and there is no getting around it!

- Other typical issues
  - Circular references
    - Prototype-only References can help
  - Non-Windows platform-specific code
    - Custom platform targets resolve some
  - Learning curve
    - New processes & procedures
  - Developer reluctance
    - We've always done it THIS way and I don't WANT to change!

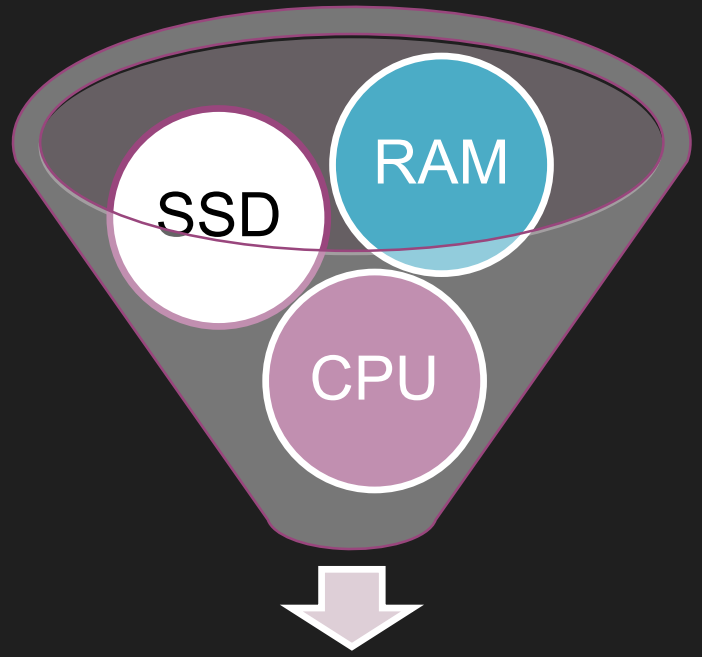
Visual Studio Development Environment



SYNERGEX

Tips & Tricks

THE TOP TIP



DEVELOPER PRODUCTIVITY

- Minimum 8GB RAM
- Minimum QUAD core I5 / I7
- Solid State Drives

SYNERGEX

Configure the Environment the way YOU want it!

Options

Synergy

- Text Editor
  - Synergy DBL
    - General
    - Scroll Bars
    - Tabs
    - Advanced
    - Formatting
    - IntelliSense

Set back to defaults: Style 1 | Style 2

Basic Blocks: Indent contents of block

Comments: Leave first column alone

Conditional Blocks: Align begin/end to block

Continuation Characters: Align to containing statement

Directives: Align to first column | Indent .include

Labels: Align to containing routine

Lone Begin/End Block: Indent one level

Routines: Indent contents of block

On Document Reformat:

- [ ] Indent code line comment to fixed column number (0)
- [ ] Align data division field elements

Comma Placement: Next to variable

Tab Stops Between Elements: 1

```
namespace myNamespace
;first column comment
    subroutine mySub
        param ,i4
        param2, i4
        record
            var ,i4
            var2, i4
    proc ;code line comment
        begin
            if (1 .or.
            &    true)
            begin
                using (var) select
                (param),
                    nop
                endusing
```

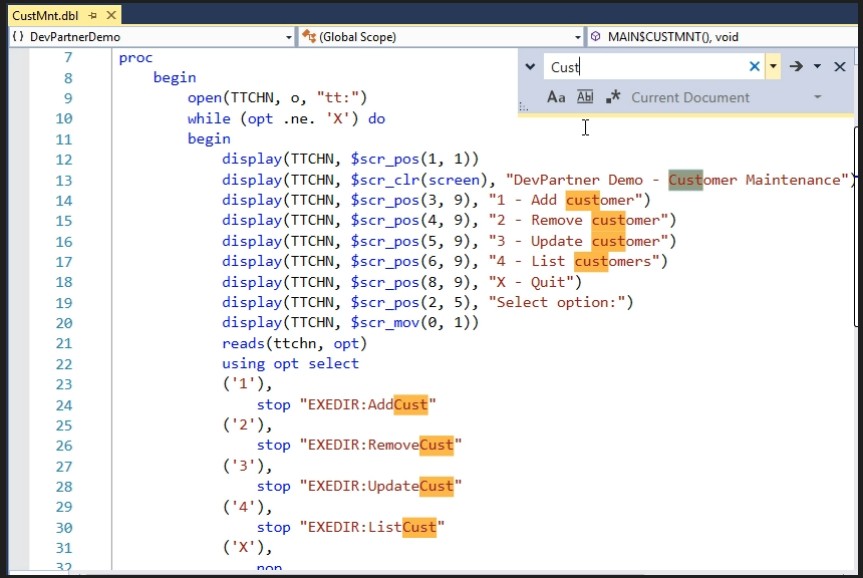
OK | Cancel

SYNERGEX

EDITOR

Finding Things in Code: Text-based

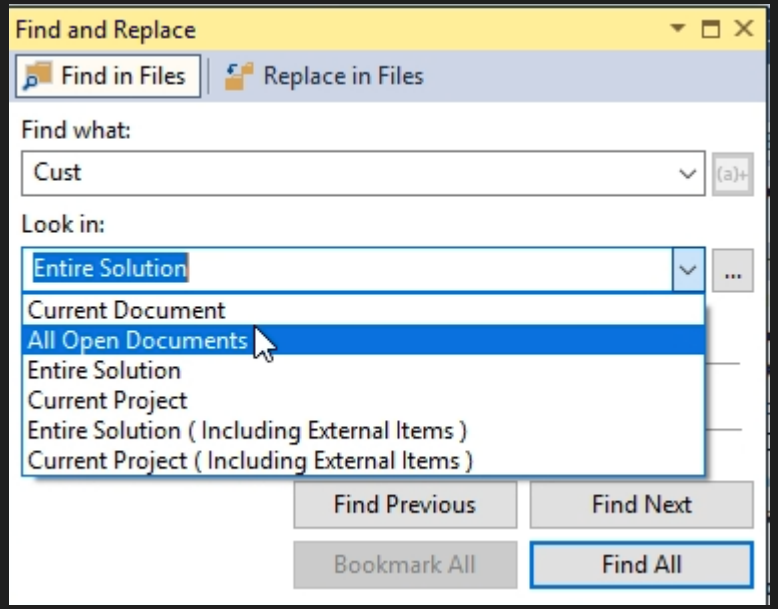
- In-line find [& replace]
  - Scope from selection up to solution
  - Regular expression support
  - In-line Find
    - Ctrl + F
  - In-line Find & Replace
    - Ctrl + H



SYNERGEX

Finding Things in Code : Text-based

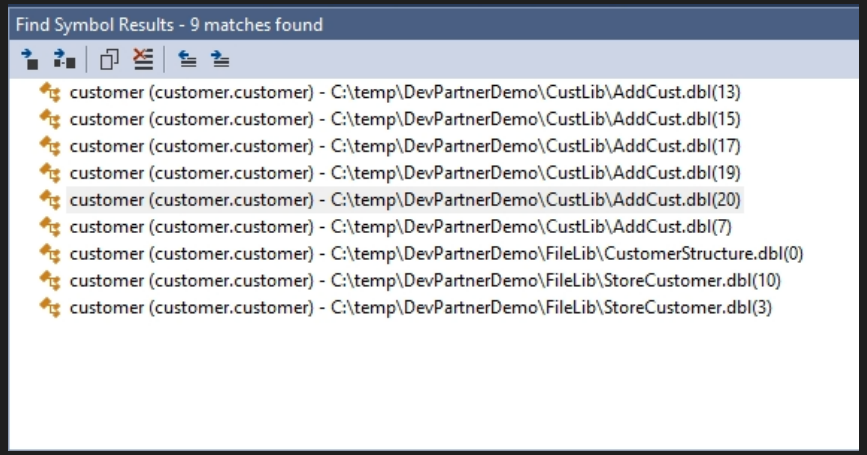
- Dialog-driven find [& replace]
  - More options than in-line search
    - Scope from selection up to folder(s)
    - Regular expression support
    - Case support
    - Whole word support
    - Multiple search results options
  - Find dialog
    - Ctrl + Shift + F
  - Find & replace dialog
    - Ctrl + Shift + H



SYNERGEX

Finding Things in Code: Symbol Based

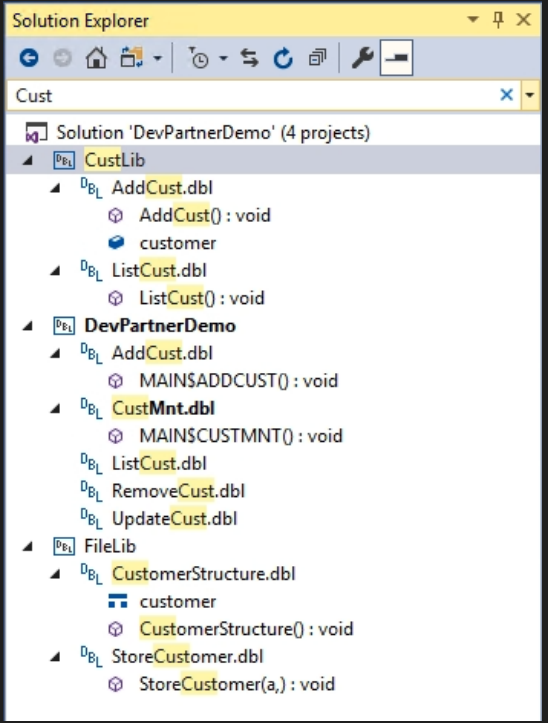
- Find All References (Shift + F12)
  - Locate every known declaration and usage of a symbol
  - Results displayed in a toolbar
  - Double-click to navigate



SYNERGEX

Finding Things in Code : Symbol Based

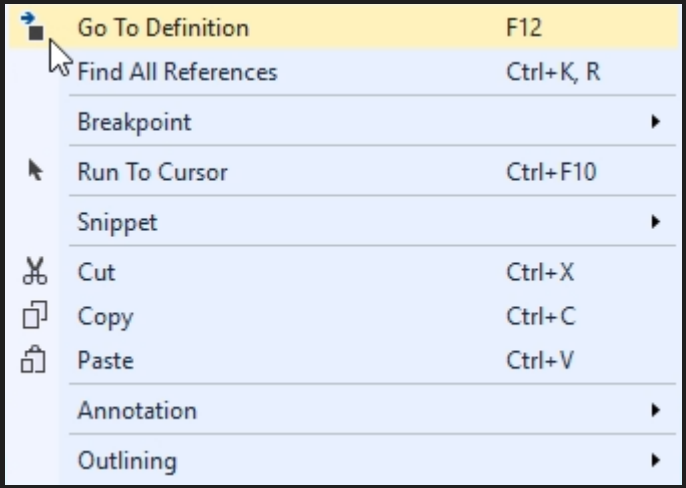
- Solution Explorer search (Ctrl+;)
  - Text-based find within symbol information & file names
  - Partial matches work fine
  - Displayed inline in Solution Explorer
  - Double-click to navigate



SYNERGEX

Finding Things in Code : Symbol Based

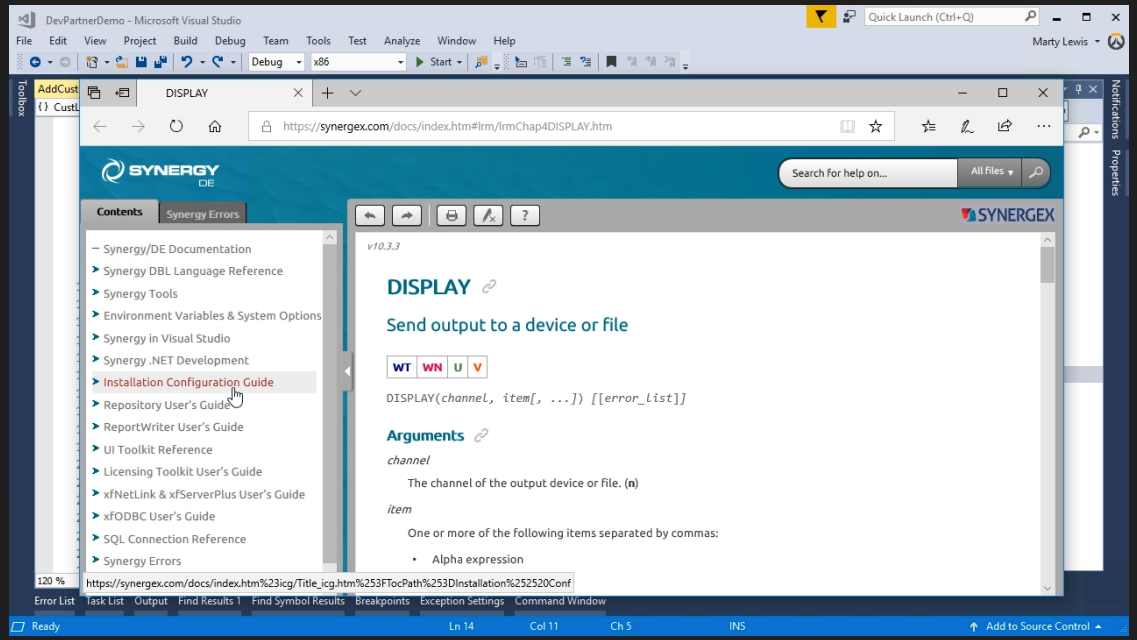
- Go To Definition (F12)
  - Navigating through code quickly
- Try it with
  - Local data
  - Records
  - Fields
  - Routines
  - Include files
  - Repository structures



SYNERGEX

Finding Things About Code

- GOTO Help (F1/F12)
  - Recognizes Synergy built-ins
  - Navigates to web docs for the selected item



SYNERGEX

Maximizing IntelliSense

- Keyword completion
  - Variable names
  - Types
  - Routine names
  - Special “new” support
  - Trigger via Ctrl + Space

```
19          reads(TTCHN, mycust.address)
20          display(TTCHN, $scr_pos(6, 5), "Phone number: ")
21          reads(TTCHN, mycust.phone)
22          my
23                                        )
24      end
25      xreturn
26  endsubroutine
```

mycust
SSQL_DID_MYSQL

```
main
    record
        myList, @List<string>
proc
    myList = new
endmain

endnamespace
```

ch, sort, and manipulate lists.To browse

line
LinkedList<T>
LinkedListNode<T>
Linq
List<String>
List<T>
ll_close
ll_open
ll_process

SYNERGEX

Maximizing IntelliSense

- List members
  - Fields in a record
  - Members in a class
  - Triggered by pressing period after a symbol name

```
20          display(TTCHN, $scr_pos(6, 5), "Phone number: ")
21          reads(TTCHN, mycust.phone)
22          mycust.
23          xcall S                    cust)
24      end
25      xreturn
26  endsubroutine
27
```

address
companyname
contact
phone
priority

- Routine signature help
  - Triggered by initial open paren
  - Re-trigger via Ctrl + Shift + Space

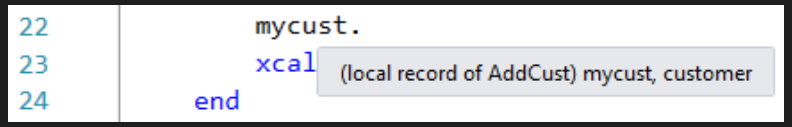
```
11      record
12          myList, @List<string>
13  proc
14      myList = new List<String>()
15      myList.FindAll(
16  endmain
```

(method) List<String>.FindAll(**match Predicate<String>**), List<String>

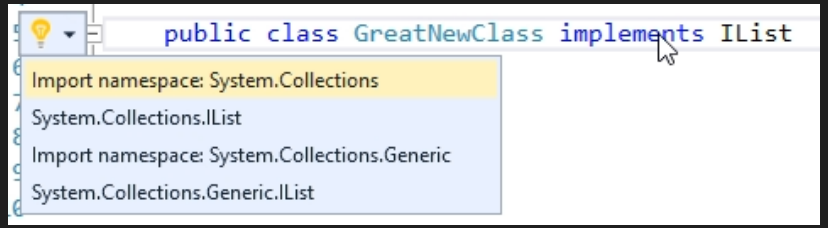
SYNERGEX

Maximizing IntelliSense

- Quick Info
  - Mouse hover-over
  - Keyboard trigger via Ctrl + K + I



- Quick Action (Lightbulb, smart tag)
  - Offered when we think we have figured out something clever to do to your code
    - Import missing namespace
    - Implement interface
  - Trigger with Ctrl + .



SYNERGEX

Maximizing IntelliSense

- Snippets
  - Keywords that expand to code
  - Lots included
    - namespace, class, begin, property, record, propauto, method, subroutine, function, try … to name a few
  - You can add your own snippets

```
3   namespace MyNamespace
4       public class AnotherGreatClass
5           public method AnotherGreatClass
6               endparams
7           proc
8               begin
9                   try
10                  begin
11
12                  end
13                  catch (e, @Exception)
14                  begin
15
16                  end
17                  endtry
18              end
19          endmethod
20      endclass
21  endnamespace
```

SYNERGEX

More Editor Features

- Block editing
  - Processing of vertical selections
  - Super useful in languages with data divisions
  - Triggered via Alt + Drag Mouse

```
main
    record
        fname   ,a20
        lname   ,a20
        mname   ,a20
proc
    xcall MyInputRoutine()
```

- Paired keyword matching
  - Examples
    - BEGIN/END
    - SUBROUTIBE/PROC/END, etc.
  - Great for navigating massive scopes
  - Triggered via Ctrl + }

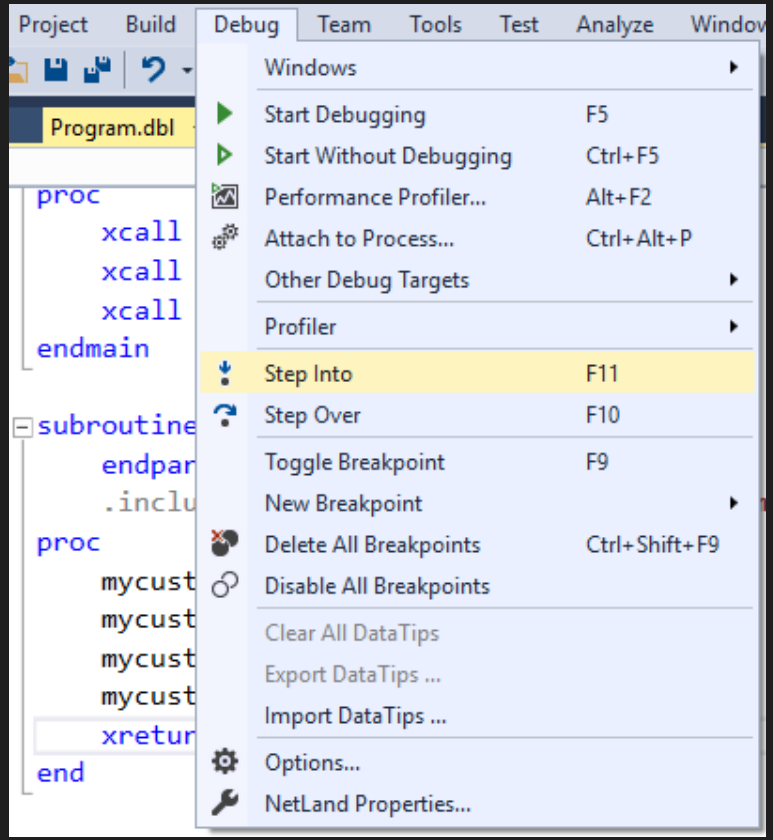
```
main
    record
        fname   ,a20
        lname   ,a20
        mname   ,a20
proc
    xcall MyInputRoutine()
endmain
```

SYNERGEX

DEBUGGER

Debugging Basics

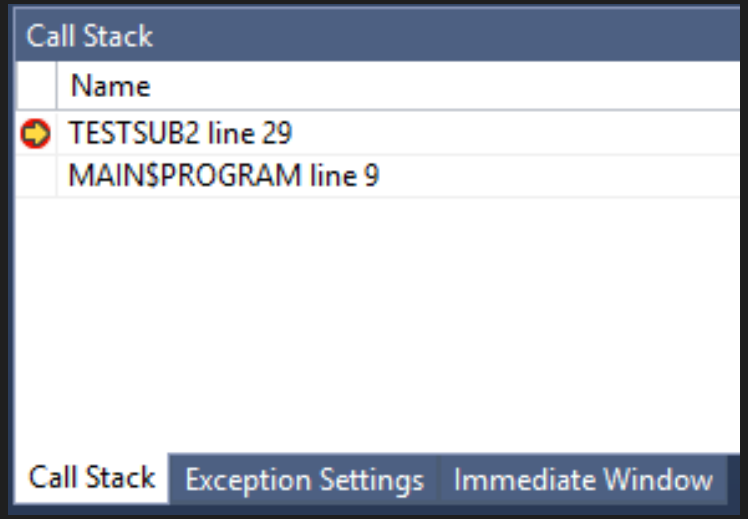
- Stepping
  - In F11
  - Out Shift + F11
  - Over F10
- Evaluating
  - Hover-over (& pinning)
  - Watch window
  - Locals window
- Breakpoints
  - Conditional breakpoints

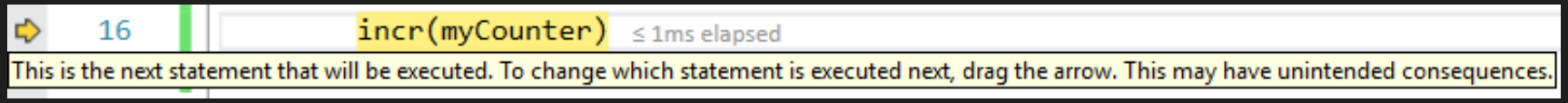


SYNERGEX

Debugging Basics

- Call stack
  - Navigating the stack
- Altering execution flow
  - In .NET only
  - Just drag your execution pointer
  - Forward and backward

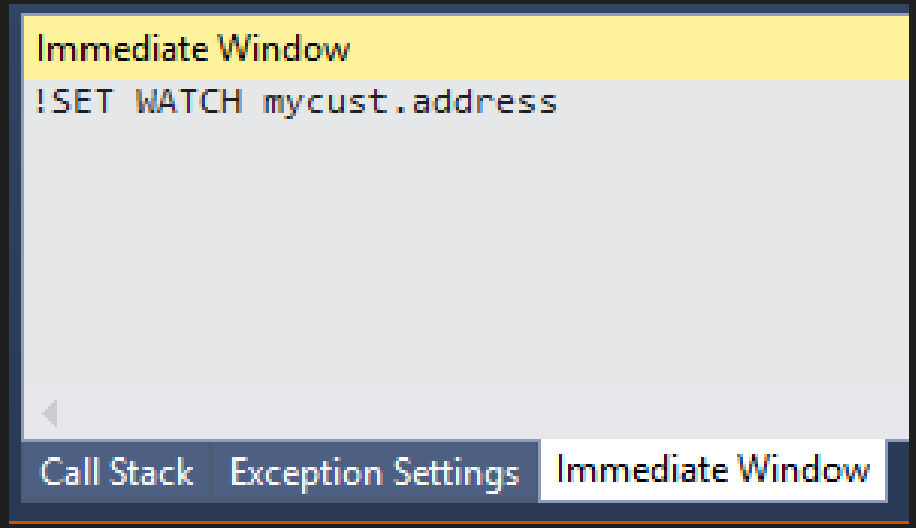




SYNERGEX

Immediate Window

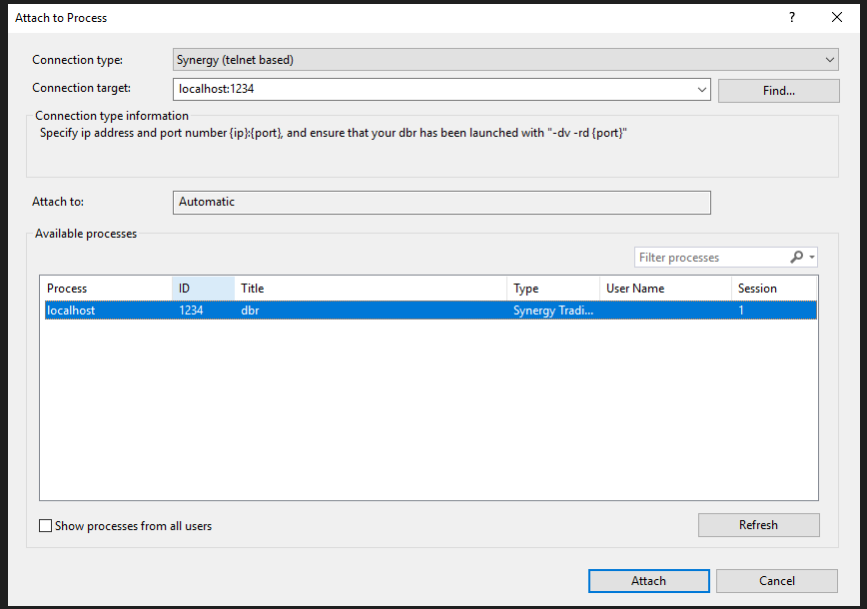
- A CLI to the debugger
  - Which is a GUI on top of
    - A transport layer on top of
      - A CLI debugger…
- Supports
  - !SET WATCH
  - !DEPOSIT
  - !SET TRAP
    - Available on property page
  - !SET UNINITIALIZED
  - !SHOW
- Does only what's on the box



SYNERGEX

Remote Debugging

- Attach to a traditional Synergy process via the Telnet debugger
  - Windows & UNIX
    - dbr -dv -rd <port> <program>
  - VMS
    - $ DEFINE DBG_RMT "-rd <port>:<timeout>"
    - $ RUN <program>
  - Back in VS:
    - Debug > Attach to Process
    - Synergy (telnet-based)
    - <machine/ip>:<port>
- Full-fidelity debugging!

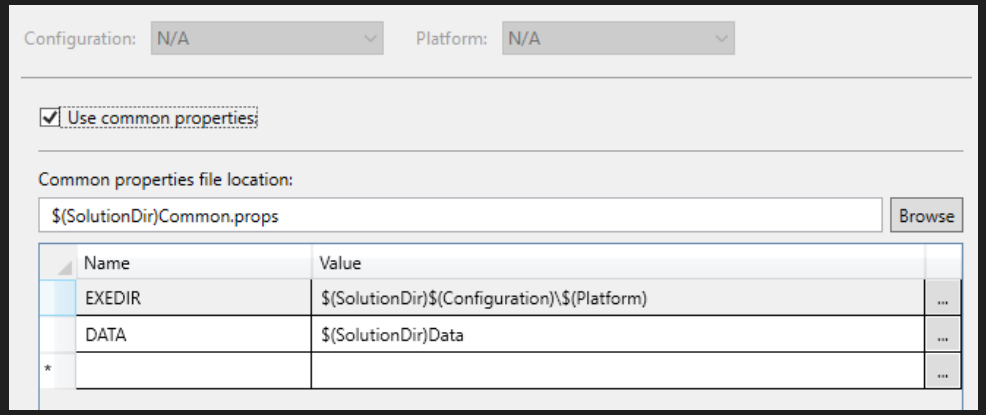


SYNERGEX

PROJECTS

Environment Variables

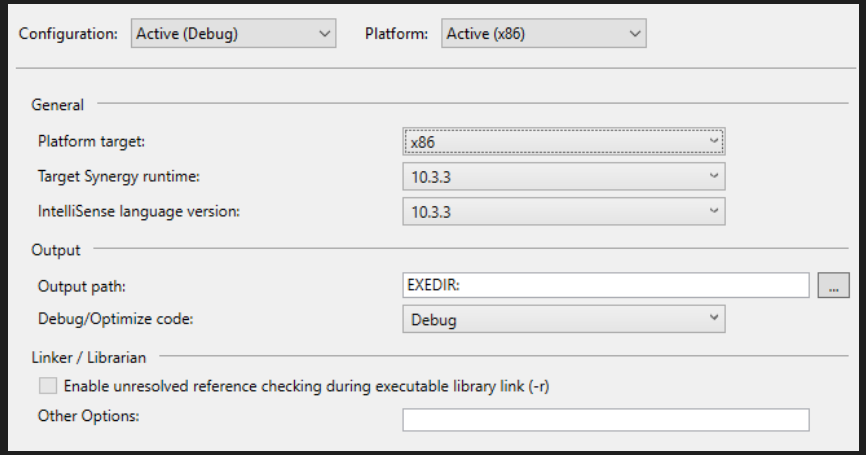
- The first thing to get right
- Common props
  - Sharable configuration across multiple Synergy projects
- Environment variables
  - Specific to one project
  - “Always wins”
- Compile-time, not run-time
  - Need to provide runtime environment for your apps



SYNERGEX

Property Pages: Build

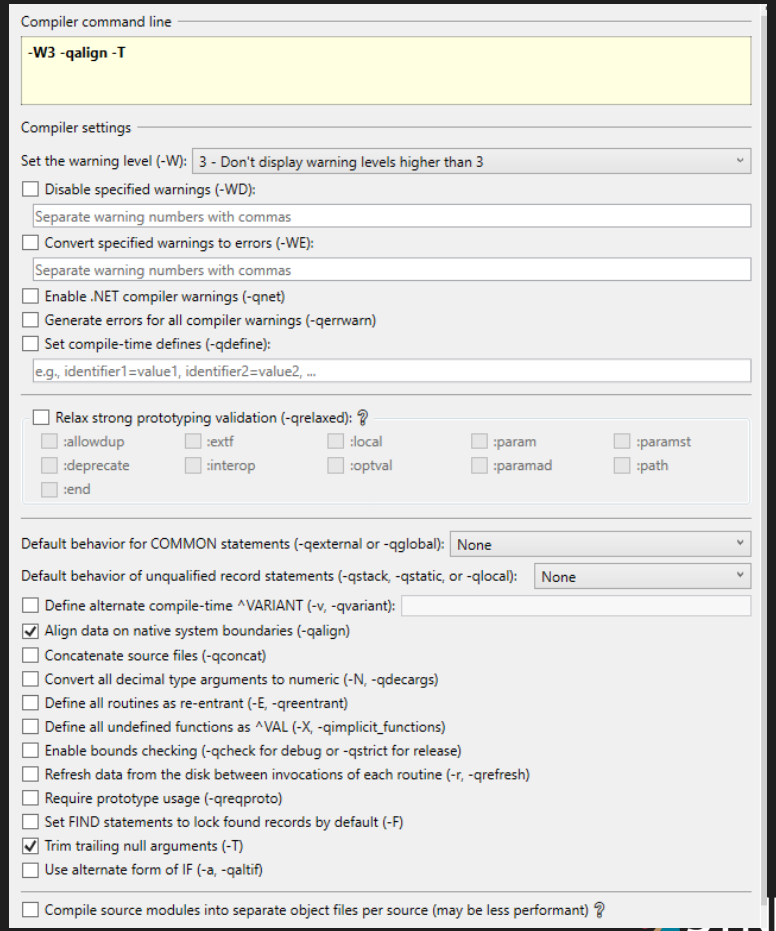
- Build platform target
  - The real target of your config
- Target runtime version
  - Synergy runtime backtargeting
- Debug / Release
- Output path
  - EXEDIR: by default



SYNERGEX

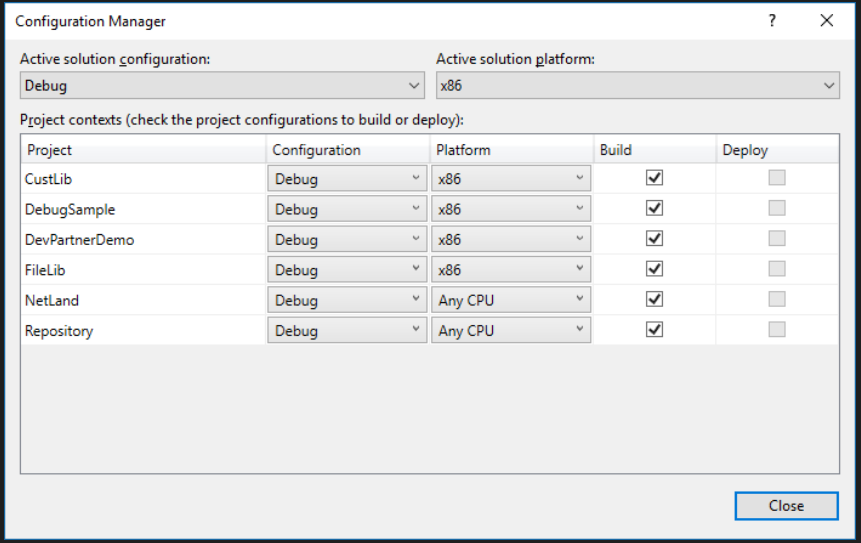
Property Pages - Compile

- -qrelaxed
  - :paramad
  - :paramst
  - :allowdup
- Getting less relaxed
  - -qreqproto
    - Oh yeah, even more strict
- Build a little or build a lot
  - Single DBO versus many DBO



Configuration Manager

- Choose how configurations fit together
  - Generate new configurations
  - Configure build environments for Linux or OpenVMS
- Manage project build settings
  - Only build the projects relevant to your configuration



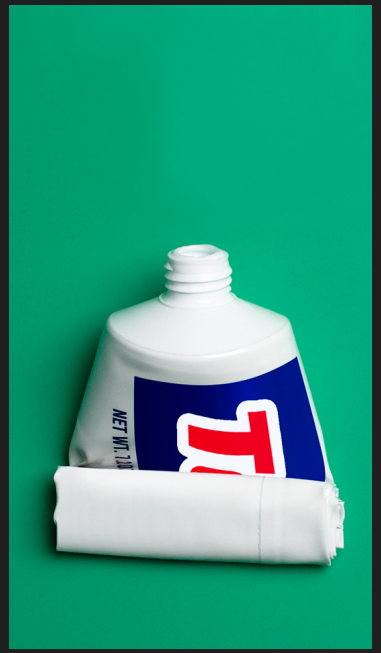
SYNERGEX

Developing for OpenVMS

- OpenVMS compatibility library
  - https://github.com/Synergex/VmsCompatibility
- Open source library providing stubs for common VMS routines
  - Some implemented to mock behavior
- Another tool in your toolkit
- Include in your solution and add a reference

Getting the Most out of Visual Studio

- Take the time to set things up the way YOU want them!
- Prototyping can take some effort, but it's ALWAYS worth it in the end!
- Recognize there will be a learning curve
- Always use a SCM tool
  - And have developers work in isolation
- KEEP EVERYTHING UP TO DATE
- Don't skimp on hardware!
- Let us know if you need help



Who has the first question?

SYNERGEX