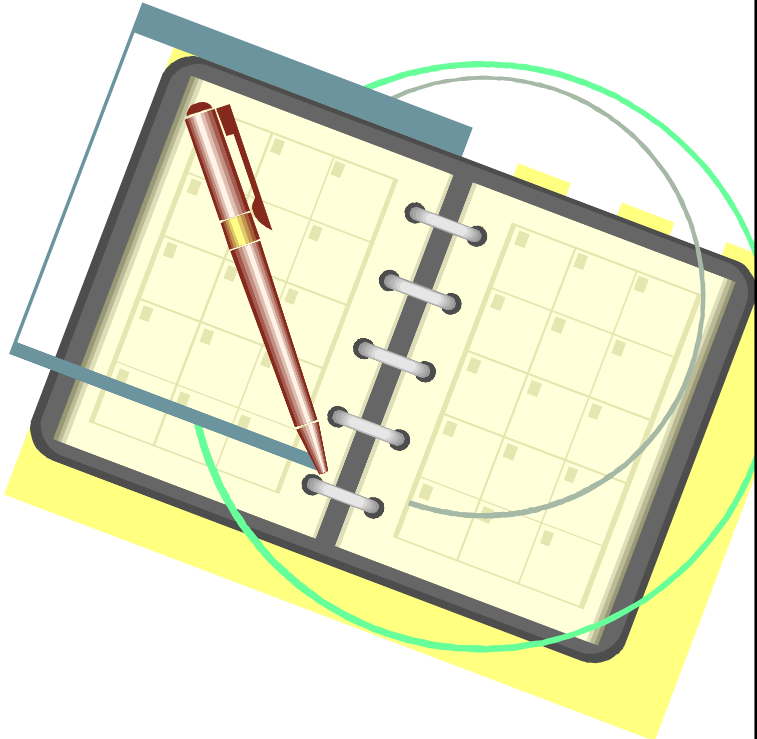
gmStudio Workshop

Mark Juras
Great Migrations LLC
mark@greatmigrations.com

Workshop Agenda

- Objectives, Expectations, and Context
- Methodology Concepts
- Technology Concepts
- Demonstrations and Labs
- Onsite Smart Start

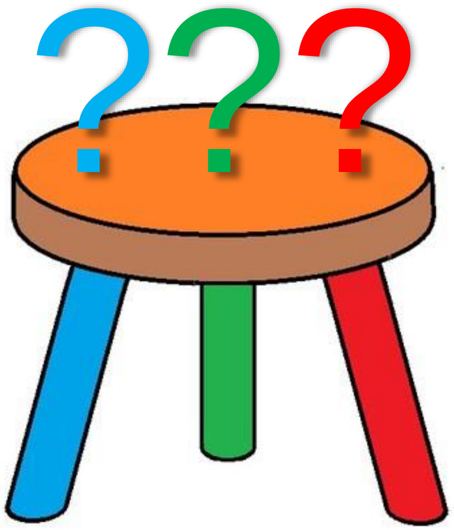


Objectives and Expectations

- Objectives
  - Successfully upgrade VB6/ASP systems to .NET
  - Understand how Great Migrations can help you
- Expectations
  - VB6/ASP systems = ?
  - Successfully = ?
  - Upgrade = ?
  - .NET = ?
- Additional Reading
  - greatmigrations.com/pubs/gmStudioPricing.pdf
  - greatmigrations.com/resources/myth-busters.aspx

Three Factors

- As-Is: what do you have?
- To-Be: what do you want?
- How: how will you get there?
  - Priorities
  - People
  - Process
  - Productivity
  - Proof



Accuracy requires details

details

details

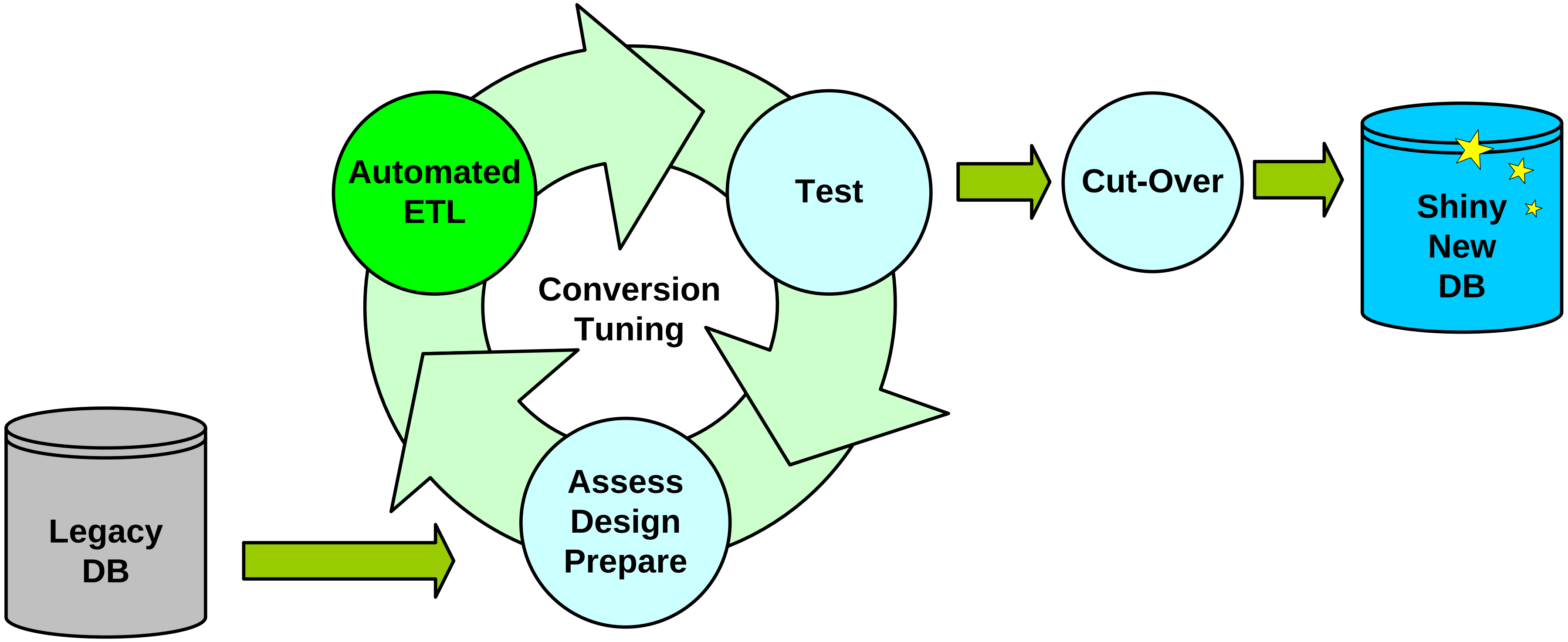
details

details…….

Estimation

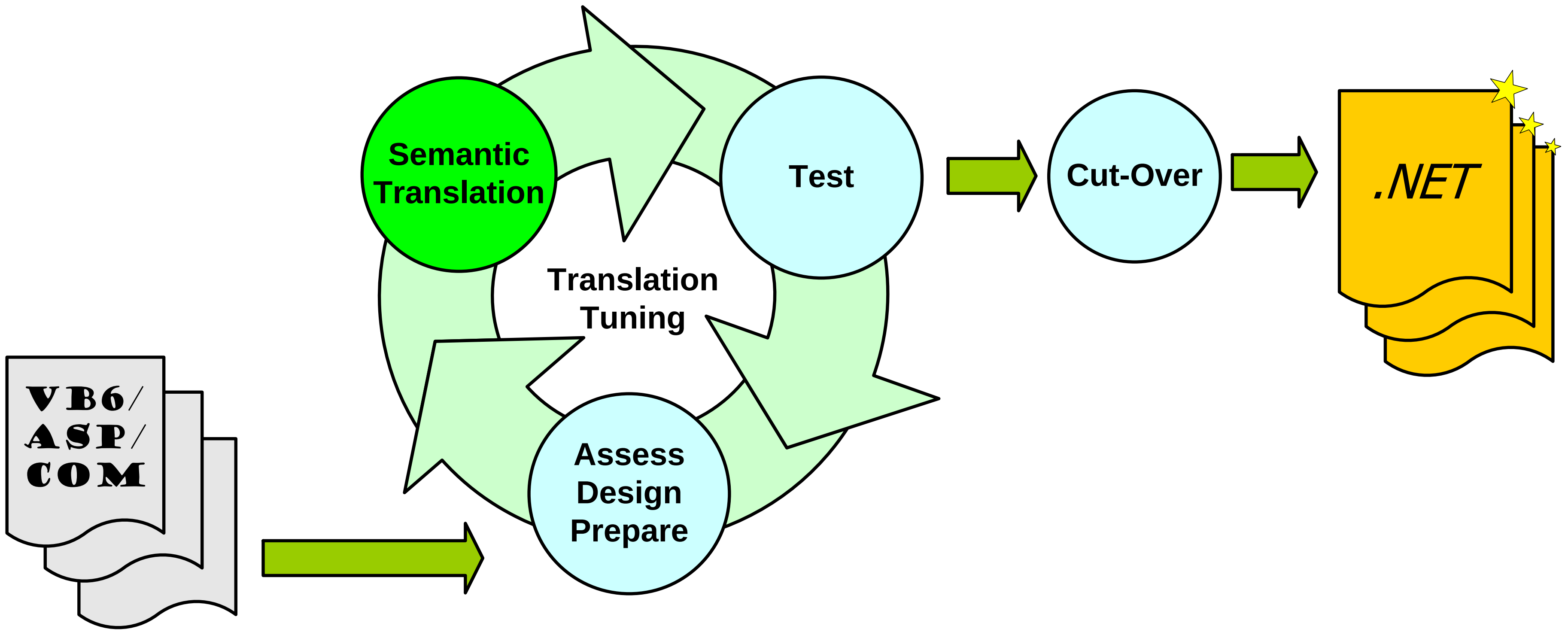
- What is the Source/Target Architecture? Do you have a target architecture in mind?
  - Inter-related VBPs?
  - Shared Code Files?
  - Code to be removed?
  - Specific new components to incorporate?
- What is the Maintenance Process? (SDLC, schedule, team structure/size/capabilities)
- What is your Source/Target Platform? OS upgrade? Multiple-OS?
- What is the desired migration team? (testers, developers, internal/external partners)
- What is your expected process for project status tracking and management?
- What is your specific acceptance criteria for deliverables?
- Do you plan to change functionality during the migration?
- Are you interested in having GM develop automated unit tests?
- What is your Development Process (unit testing? Other tools?)
- What is your SCM Process? (Version control, tools, standards)
- What is your Deployment Process?
- What is your Test Process? (team, environment, data, automation)

Methodology: Concept



The Tool-Assisted Data Conversion

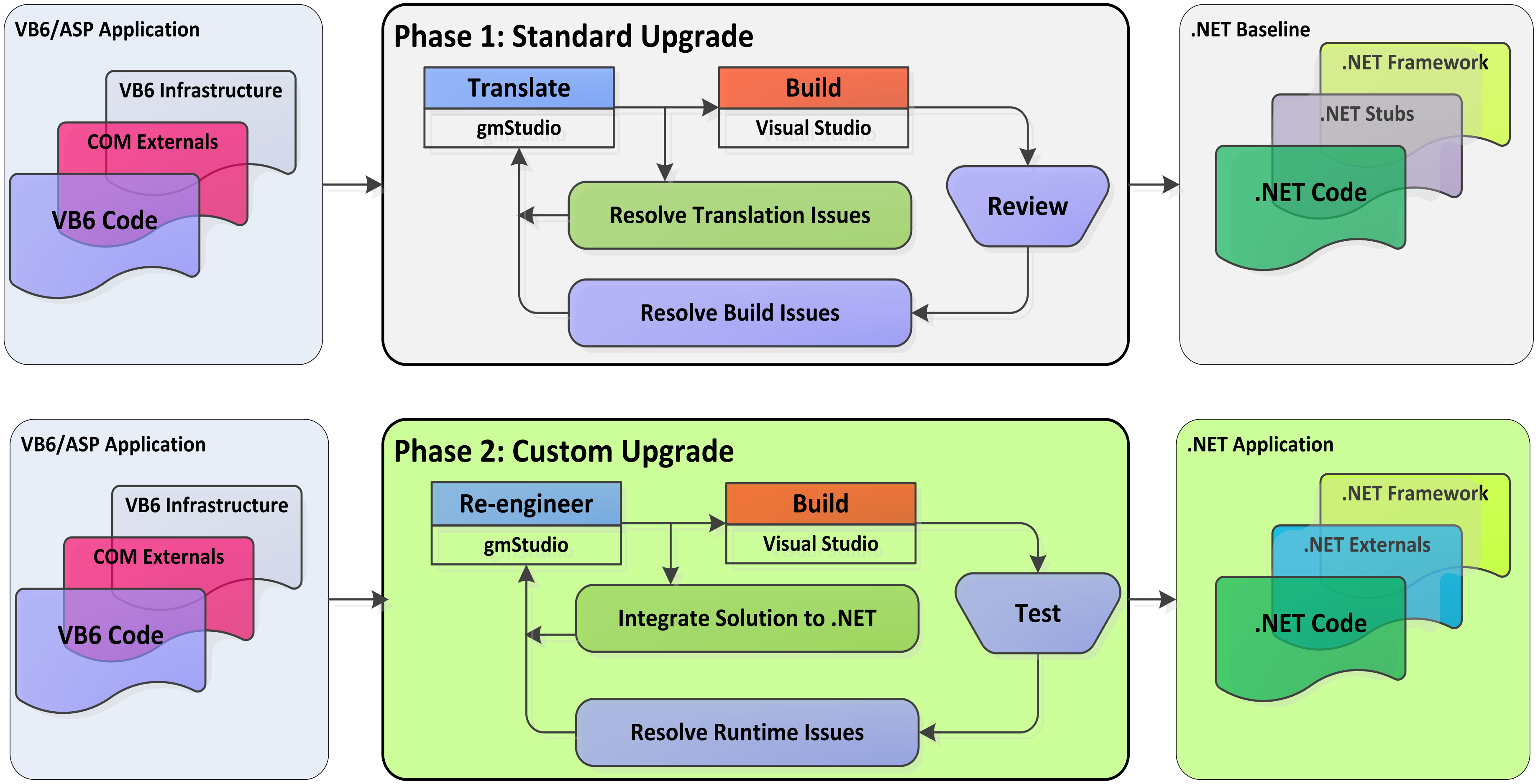
Methodology: Concept



The Tool-Assisted Rewrite

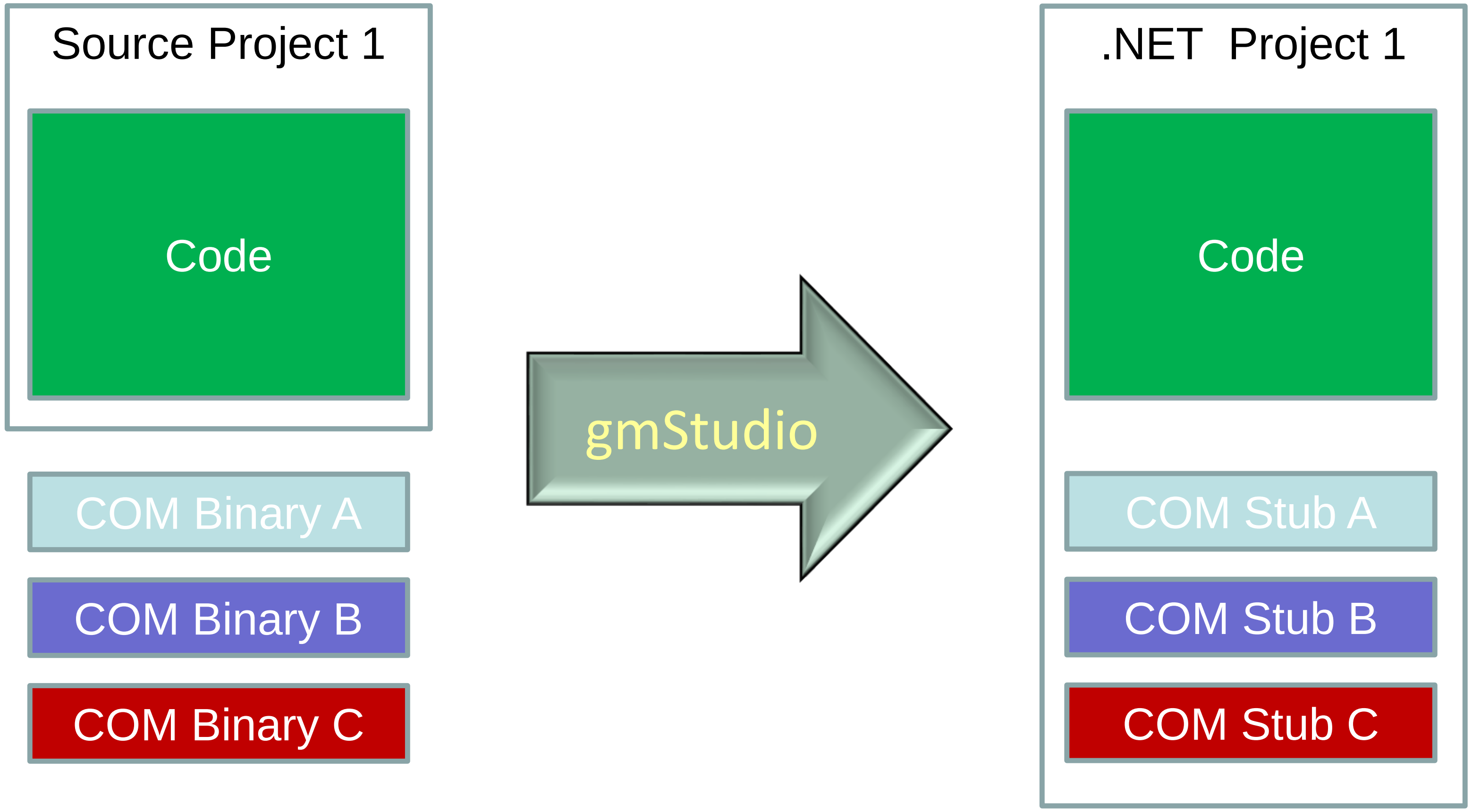
Agile Iterative Scalable Repeatable Measureable Improvable

Methodology: Phases



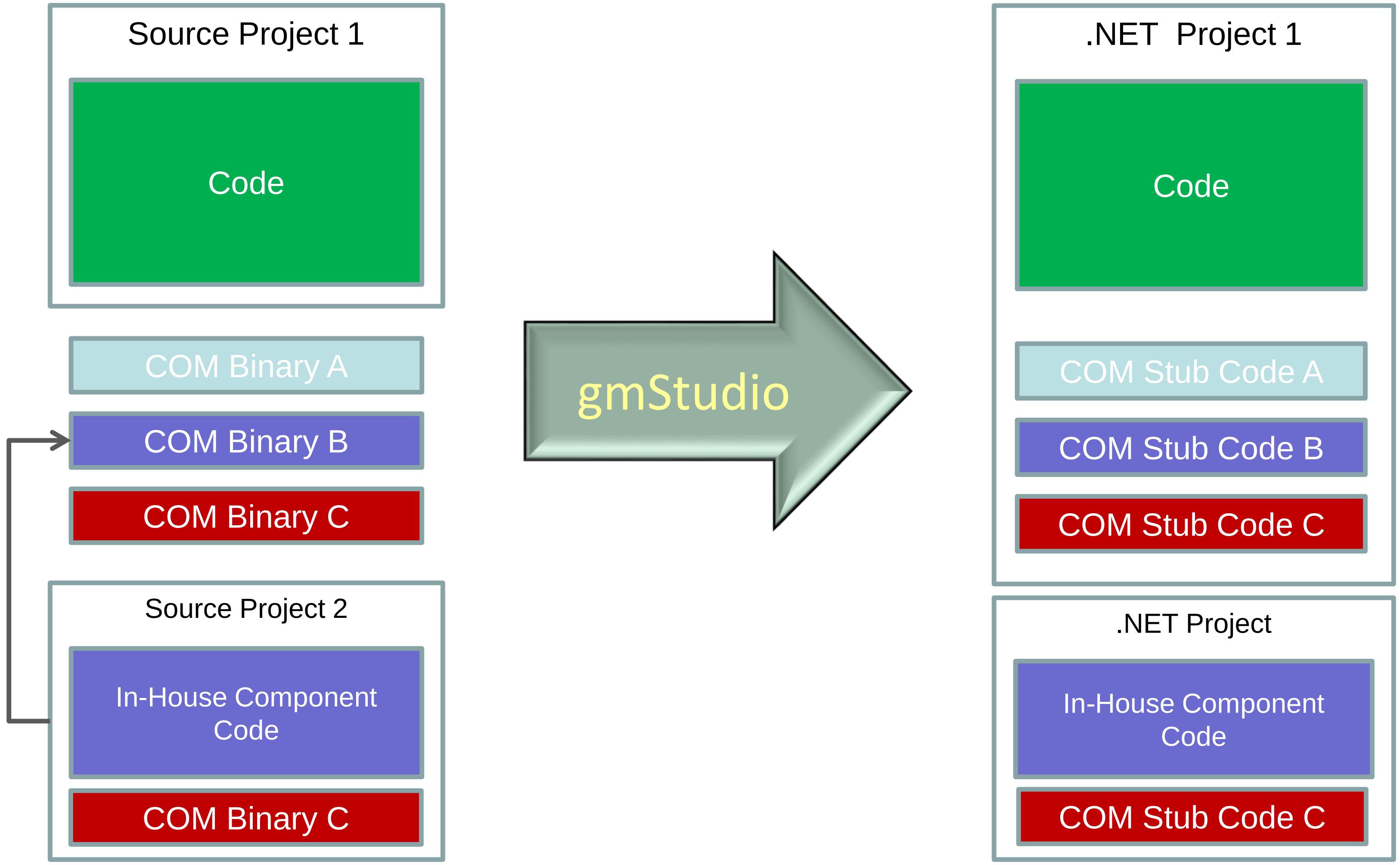
Incremental Upgrade: Step 1

Single, Standalone Translation



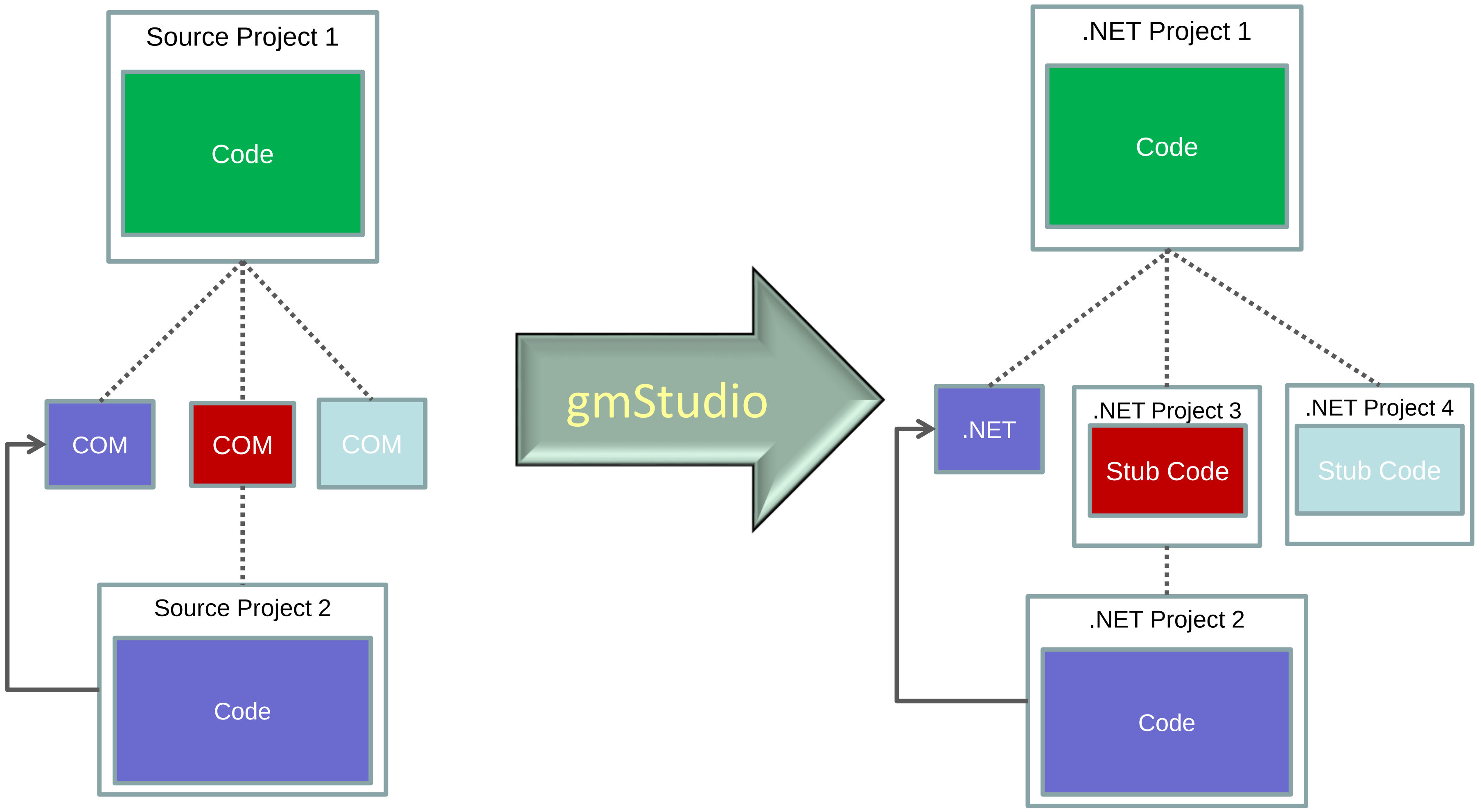
Incremental Upgrade: Step 2

Multiple, Standalone Translations



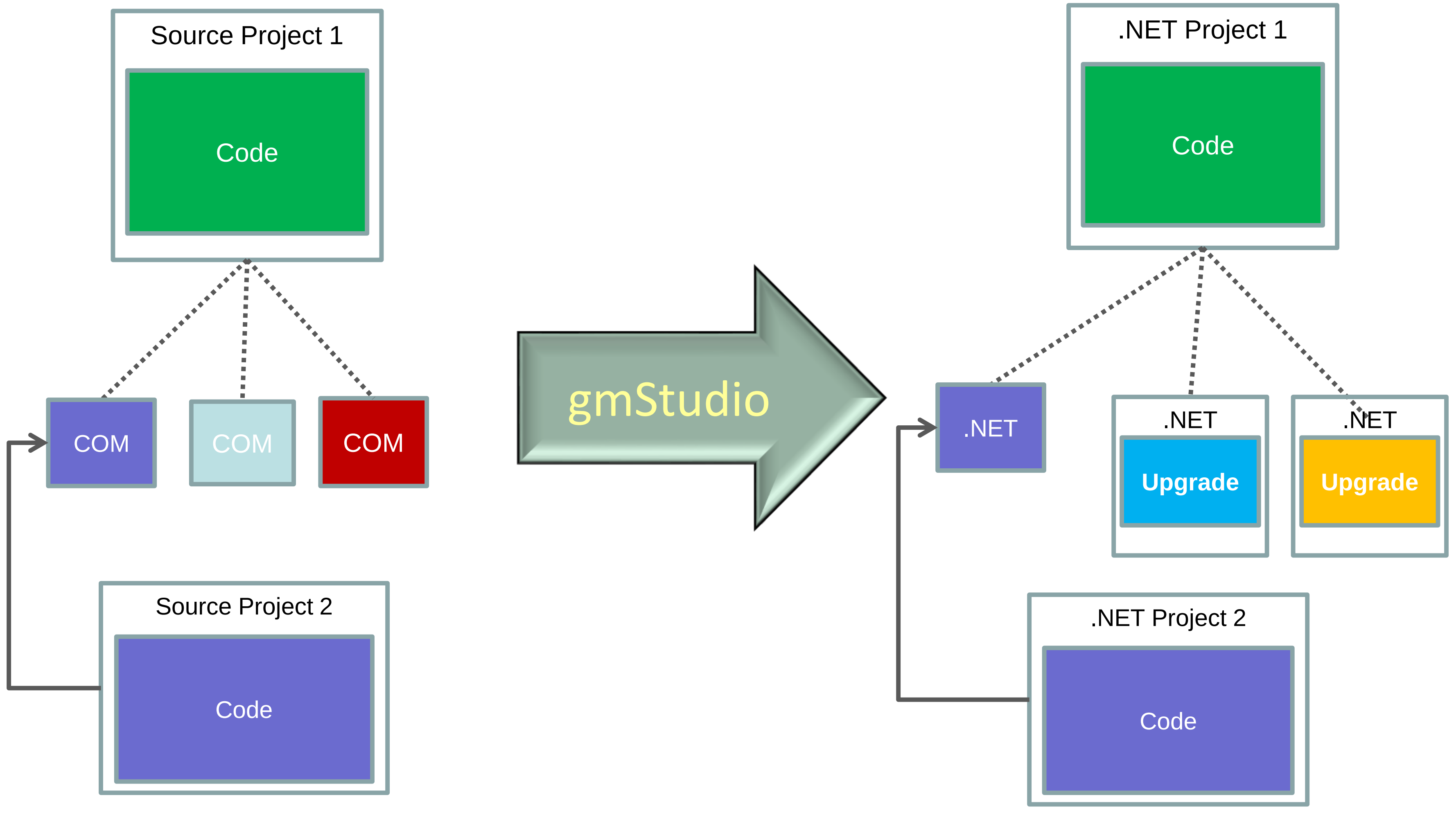
Incremental Upgrade: Step 3

Multiple, Integrated Translations

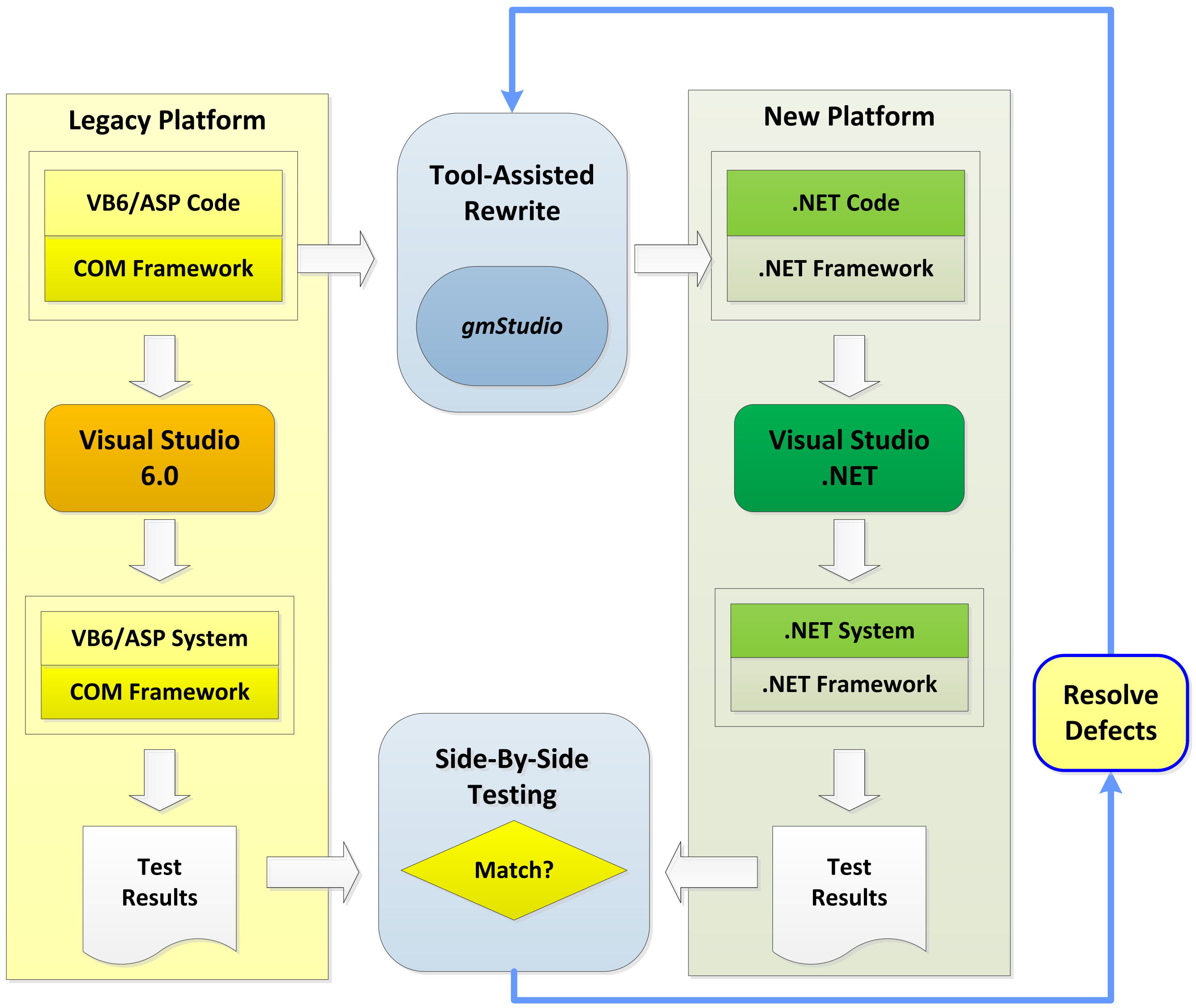


Incremental Upgrade: Step 4

Multiple, Integrated Upgraded Translations



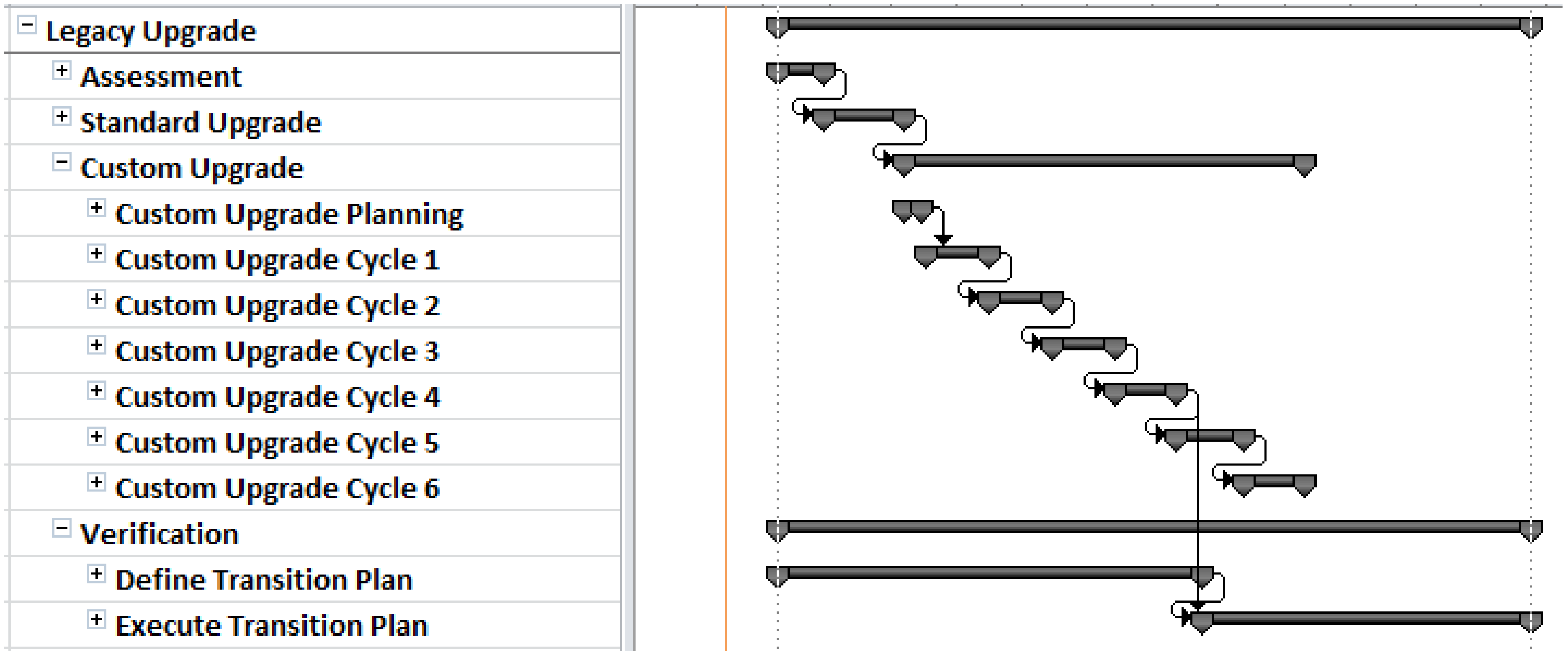
Methodology: Side-by-Side Testing



Methodology: Milestones

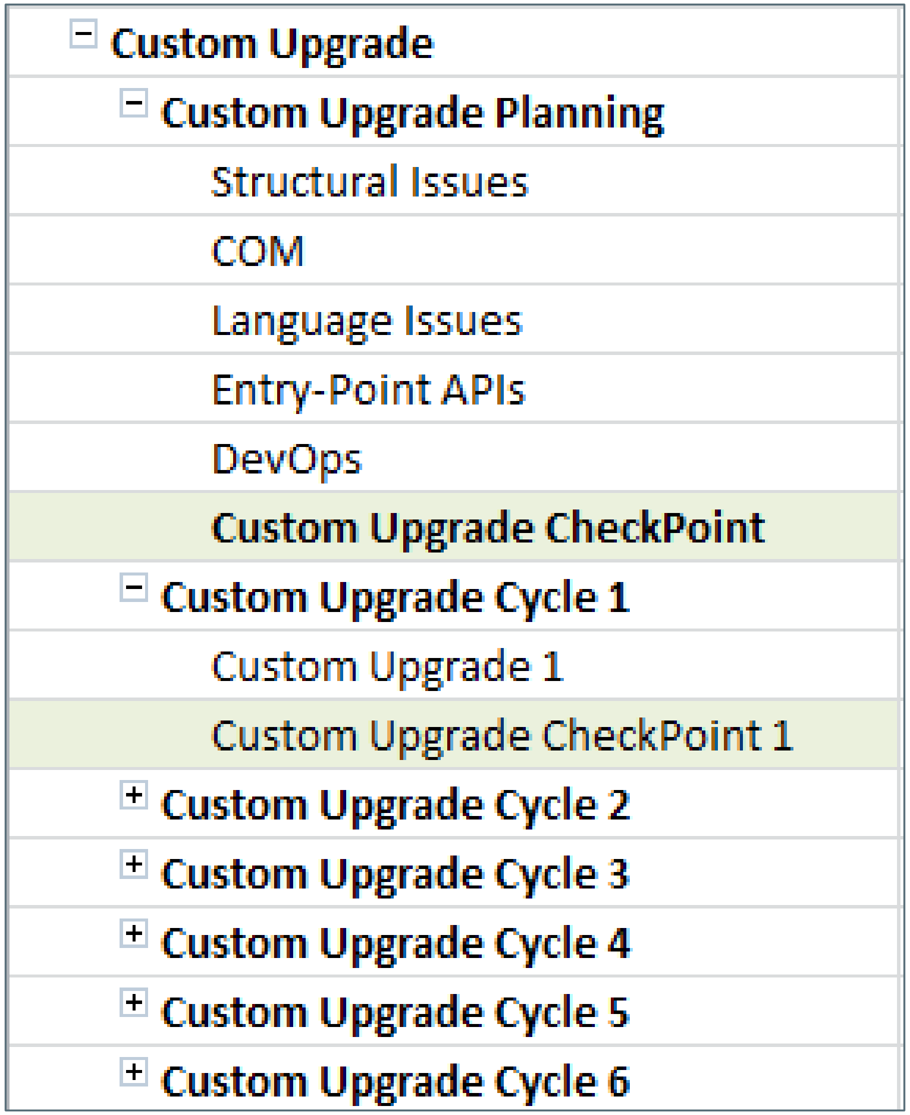
- 0: Source Complete, Ready to Translate
- 1: Translate Complete, Ready to Assess
- 2: Build Complete, Ready for Reengineering
- 3: Reengineering Complete, Ready for Testing
- 4: Verification Complete, Ready for Cut-Over

Methodology: Project Plan



- Legacy Upgrade
  - Assessment
  - Standard Upgrade
  - Custom Upgrade
    - Custom Upgrade Planning
    - Custom Upgrade Cycle 1
    - Custom Upgrade Cycle 2
    - Custom Upgrade Cycle 3
    - Custom Upgrade Cycle 4
    - Custom Upgrade Cycle 5
    - Custom Upgrade Cycle 6
  - Verification
    - Define Transition Plan
    - Execute Transition Plan

Proposed Phases



- Custom Upgrade
  - Custom Upgrade Planning
    - Structural Issues
    - COM
    - Language Issues
    - Entry-Point APIs
    - DevOps
    - Custom Upgrade CheckPoint
  - Custom Upgrade Cycle 1
    - Custom Upgrade 1
    - Custom Upgrade CheckPoint 1
  - Custom Upgrade Cycle 2
  - Custom Upgrade Cycle 3
  - Custom Upgrade Cycle 4
  - Custom Upgrade Cycle 5
  - Custom Upgrade Cycle 6

Custom Upgrade Cycle

1. Analyze requirements in the application
2. Select effective and economical strategy
3. Design solution feature
4. Develop Migration Unit Test (MUT)
5. Implement and Verify solution in MUT
6. Integrate solution feature with the application upgrade
7. Integrate feature results with new application
8. Select scope of work for next cycle

*Repeat until all **required** upgrade features are integrated into the new application*

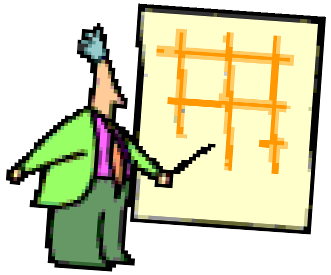
Methodology: Balance

Convert or Rewrite?

The Great Migrations Methodology balances automated translation and other techniques to create s custom upgrade solution that delivers high quality results with less risk and less effort.

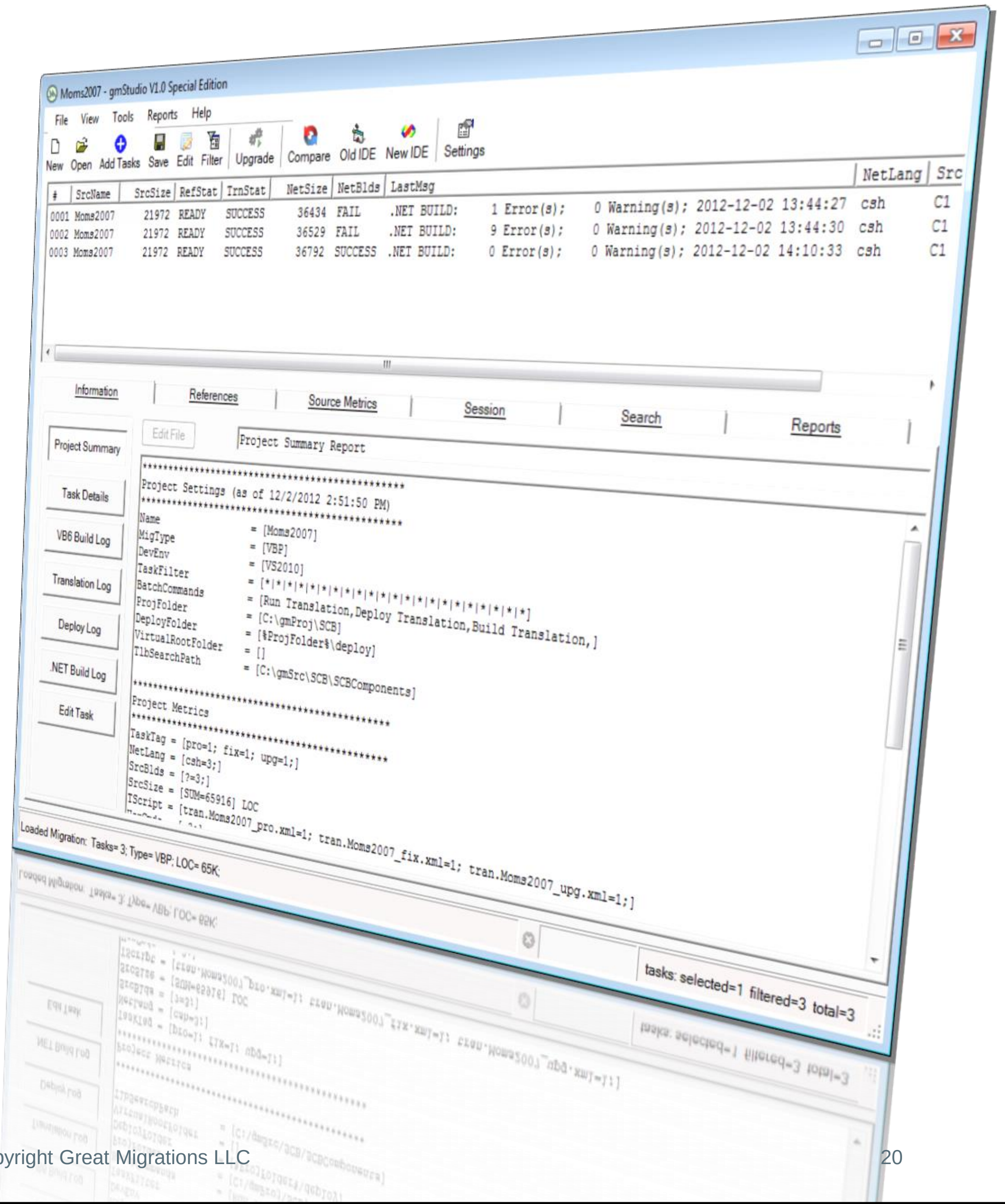
Methodology: Parts

- **Inputs**
  - VB6 Code – VBPs and source code files
  - COM Components (3PCs)
  - .NET Coding Standards, Design Standards, SCM Standards
  - Replacement Components (.NET Framework, 3PCs, In-House Components IHCs)
- **Tools**
  - **Attitude!** Check your preconceptions about the limits of automated reengineering.
  - **Brain!** Must Be Detail-Oriented! Vague objectives cannot be met.
  - gmStudio.exe / gmBasic.exe / gmDeploy.exe
  - .NET tools, MSBuild.exe / VBC.exe / CSC.EXE / ASPNET_Compiler.exe / VS2010
  - VB6, Programmer's Editor, File Comparison Tool
  - Excel (for analysis of reports), SQL Server
- **Steps**
  - Preparation: gather/refine inputs
  - Translation: run translations
  - Verification: code review, build tests, functional tests, technical tests
- **Outputs**
  - Repeatable high-performance VB6/ASP/COM to .NET upgrade solution
  - .NET codes of increasing quality



Technology: Agenda

- Overview
  - gmStudio
  - gmBasic
- Demonstrations
  - Preparation
  - Translation
  - Deployment
  - Verification
  - Refinement


Moms2007 - gmStudio V1.0 Special Edition
File View Tools Reports Help
New Open Add Tasks Save Edit Filter Upgrade Compare Old IDE New IDE Settings
Information References Source Metrics Session Search Reports
Project Summary
Task Details
VB6 Build Log
Translation Log
Deploy Log
.NET Build Log
Edit Task
Edit File

```
#    SrcName   SrcSize RefStat TrnStat  NetSize NetBlds LastMsg                                                           NetLang Src
0001 Moms2007  21972   READY   SUCCESS  36434   FAIL    .NET BUILD:  1 Error(s);  0 Warning(s); 2012-12-02 13:44:27  csh     C1
0002 Moms2007  21972   READY   SUCCESS  36529   FAIL    .NET BUILD:  9 Error(s);  0 Warning(s); 2012-12-02 13:44:30  csh     C1
0003 Moms2007  21972   READY   SUCCESS  36792   SUCCESS .NET BUILD:  0 Error(s);  0 Warning(s); 2012-12-02 14:10:33  csh     C1
```

```
Project Summary Report
*******************************************
Project Settings (as of 12/2/2012 2:51:50 PM)
*******************************************
Name                = [Moms2007]
MigType             = [VBP]
DevEnv              = [VS2010]
TaskFilter          = [*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*|*]
BatchCommands       = [Run Translation,Deploy Translation,Build Translation,]
ProjFolder          = [C:\gmProj\SCB]
DeployFolder        = [%ProjFolder%\deploy]
VirtualRootFolder   = []
TlbSearchPath       = [C:\gmSrc\SCB\SCBComponents]
*******************************************
Project Metrics
*******************************************
TaskTag = [pro=1; fix=1; upg=1;]
NetLang = [csh=3;]
SrcBlds = [?=3;]
SrcSize = [SUM=65916] LOC
TScript = [tran.Moms2007_pro.xml=1; tran.Moms2007_fix.xml=1; tran.Moms2007_upg.xml=1;]
```

```
Loaded Migration: Tasks= 3; Type= VBP; LOC= 65K;
tasks: selected=1 filtered=3 total=3
```

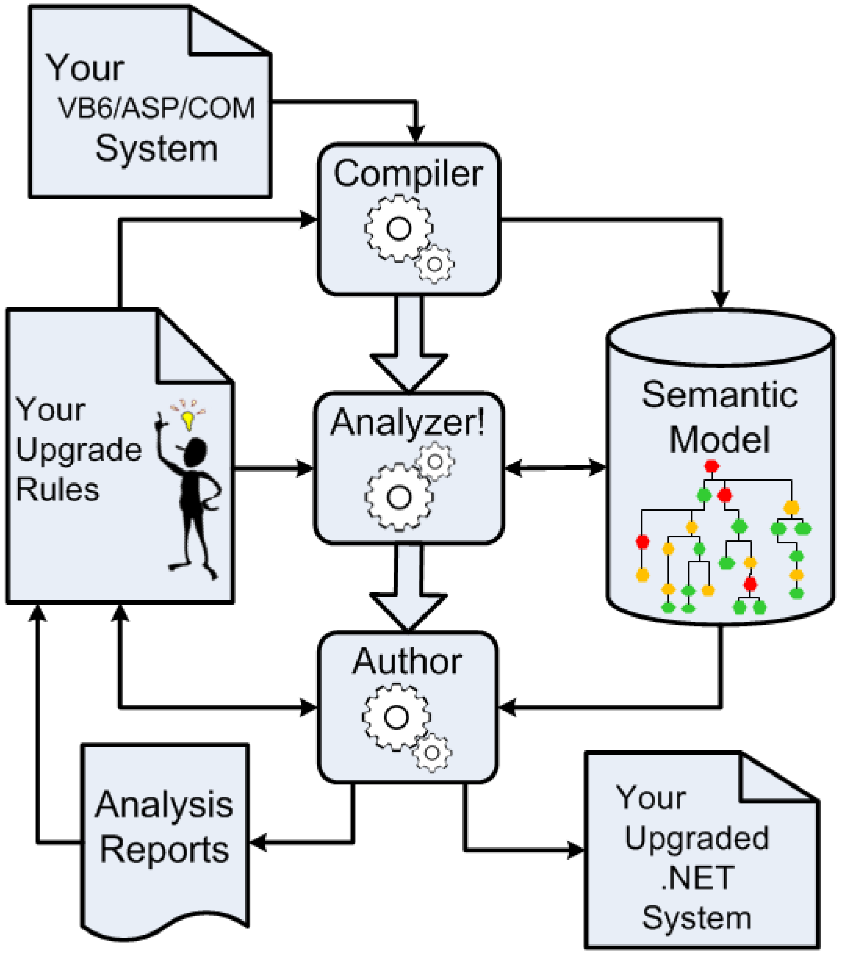
Technology: gmBasic Overview

What is gmBasic?

A highly configurable, robust VB6/ASP/COM compiler that produces source codes instead of binaries.

How does it work?

- **Compiler:** Builds a comprehensive semantic model of the codebase implementation.
- **Analyzer:** evaluates and restructures the model to fit the desired architecture patterns.
- **Author**: processes the optimized model to generate clean, correct code that meets custom standards and conventions.



VB6 to .NET, ASP to .NET, IDL to XML, XML to .NET, Scripting, Reporting…

Technology: gmStudio Overview

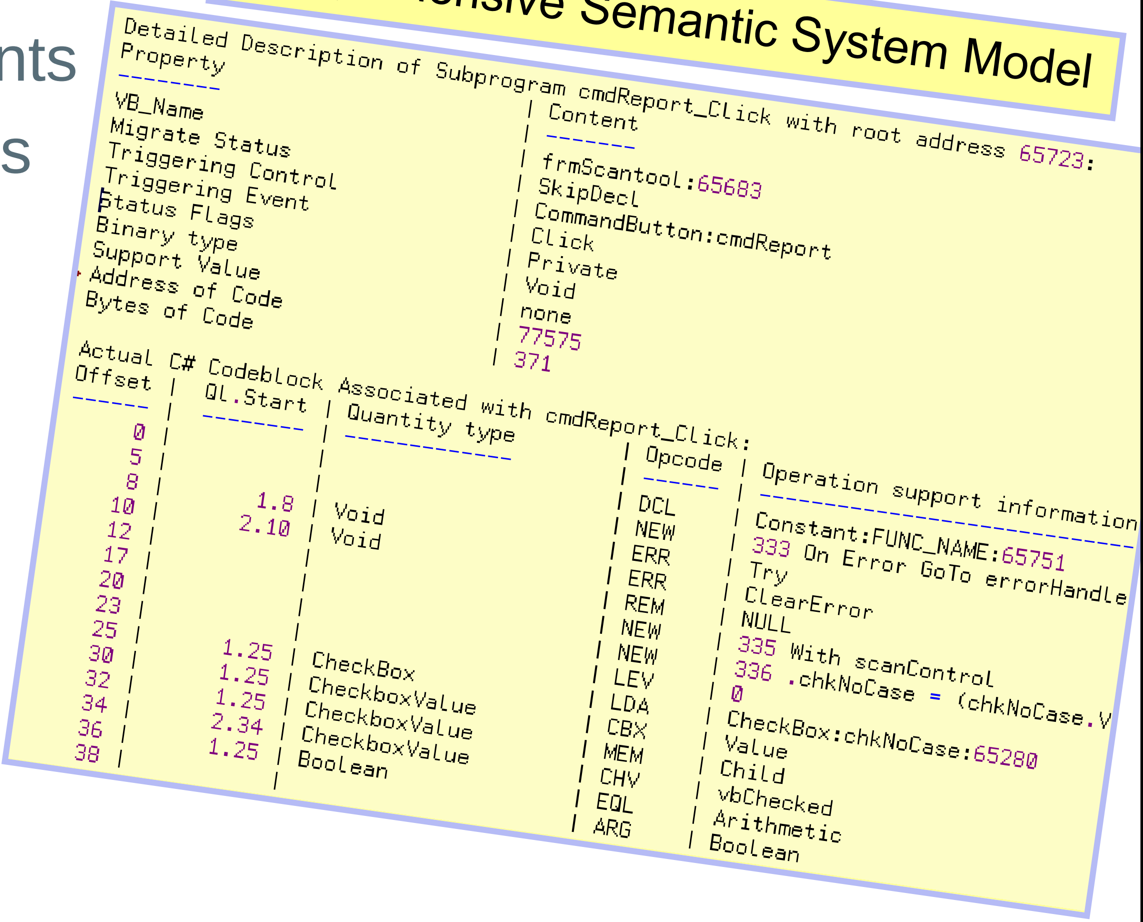
- Upgrade Development Environment
  - Project Setup – Code and COM Assessment
  - Process Orchestration
  - Solution Development and Experimentation
  - Search, Analysis, and Reporting
- “Integrated” Tools
  - Transformation: gmBasic.exe
  - Deployment: gmDeploy.exe
  - Code Editing: e.g. Notepad++
  - File/Folder Comparison: e.g. BeyondCompare
  - Other: VB6, VisualStudio, MSBuild, Excel, Custom

Technology: gmBasic Internals

Semantic Model

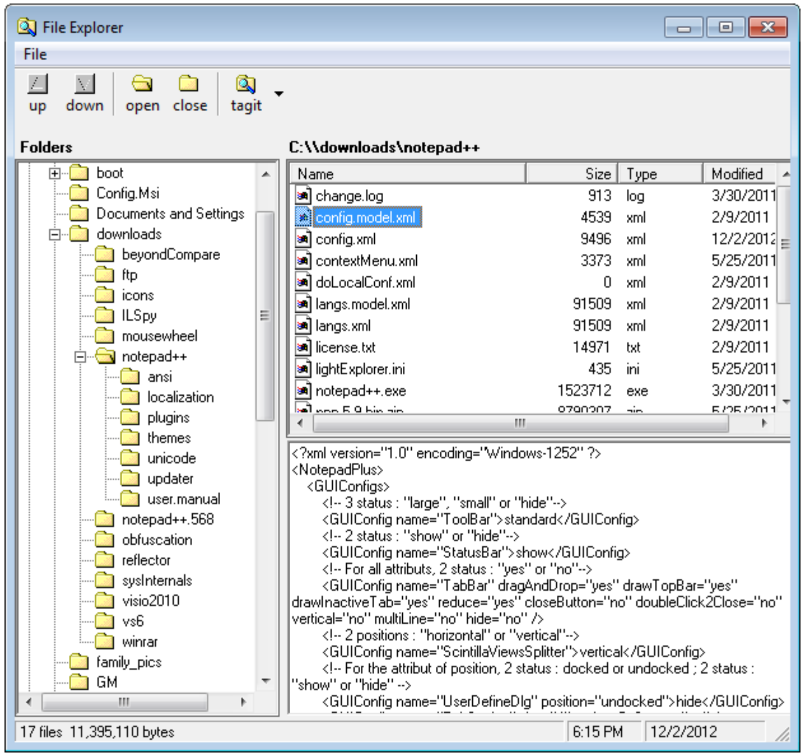
- Symbol Trees
  - External Components
  - Language Elements
  - Source Structures
- P-Code Tables
  - Operations
  - Expressions
  - Source Mappings

Comprehensive Semantic System Model



```
Detailed Description of Subprogram cmdReport_Click with root address 65723:
Property                                 | Content
---------                                | --------
VB_Name                                  | frmScantool:65683
Migrate Status                           | SkipDecl
Triggering Control                       | CommandButton:cmdReport
Triggering Event                         | Click
Status Flags                             | Private
Binary type                              | Void
Support Value                            | none
Address of Code                          | 77575
Bytes of Code                            | 371

Actual C# Codeblock Associated with cmdReport_Click:
Offset |  QL.Start | Quantity type         | Opcode | Operation support information
------ |  -------- | --------------        | ------ | ----------------------------------
     0 |           |                       | DCL    | Constant:FUNC_NAME:65751
     5 |           |                       | NEW    | 333 On Error GoTo errorHandle
     8 |       1.8 | Void                  | ERR    | Try
    10 |      2.10 | Void                  | ERR    | ClearError
    12 |           |                       | REM    | NULL
    17 |           |                       | NEW    | 335 With scanControl
    20 |           |                       | NEW    | 336 .chkNoCase = (chkNoCase.V
    23 |           |                       | LEV    | 0
    25 |      1.25 | CheckBox              | LDA    | CheckBox:chkNoCase:65280
    30 |      1.25 | CheckboxValue         | CBX    | Value
    32 |      1.25 | CheckboxValue         | MEM    | Child
    34 |      2.34 | CheckboxValue         | CHV    | vbChecked
    36 |      1.25 | Boolean               | EQL    | Arithmetic
    38 |           |                       | ARG    | Boolean
```

Demo: FileExplorer



Folders

```
boot
Config.Msi
Documents and Settings
downloads
    beyondCompare
    ftp
    icons
    ILSpy
    mousewheel
    notepad++
        ansi
        localization
        plugins
        themes
        unicode
        updater
        user.manual
    notepad++.568
    obfuscation
    reflector
    sysInternals
    visio2010
    vs6
    winrar
family_pics
GM
```

C:\\downloads\notepad++

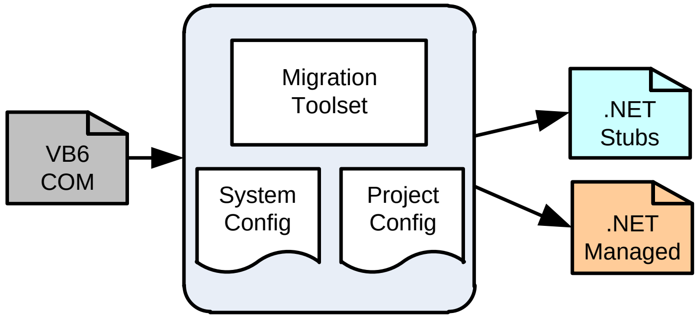
Name	Size	Type	Modified
change.log	913	log	3/30/2011
config.model.xml	4539	xml	2/9/2011
config.xml	9496	xml	12/2/2012
contextMenu.xml	3373	xml	5/25/2011
doLocalConf.xml	0	xml	2/9/2011
langs.model.xml	91509	xml	2/9/2011
langs.xml	91509	xml	2/9/2011
license.txt	14971	txt	2/9/2011
lightExplorer.ini	435	ini	5/25/2011
notepad++.exe	1523712	exe	3/30/2011

```
<?xml version="1.0" encoding="Windows-1252" ?>
<NotepadPlus>
   <GUIConfigs>
      <!-- 3 status : "large", "small" or "hide"-->
      <GUIConfig name="ToolBar">standard</GUIConfig>
      <!-- 2 status : "show" or "hide"-->
      <GUIConfig name="StatusBar">show</GUIConfig>
      <!-- For all attributs, 2 status : "yes" or "no"-->
      <GUIConfig name="TabBar" dragAndDrop="yes" drawTopBar="yes"
drawInactiveTab="yes" reduce="yes" closeButton="no" doubleClick2Close="no"
vertical="no" multiLine="no" hide="no" />
      <!-- 2 positions : "horizontal" or "vertical"-->
      <GUIConfig name="ScintillaViewsSplitter">vertical</GUIConfig>
      <!-- For the attribut of position, 2 status : docked or undocked ; 2 status :
"show" or "hide" -->
      <GUIConfig name="UserDefineDlg" position="undocked">hide</GUIConfig>
```

17 files 11,395,110 bytes | 6:15 PM | 12/2/2012

- Standalone EXE
  - ListView
  - TreeView
  - ImageList
  - StatusBar
  - ToolBar
  - RichTextBox
  - Scripting

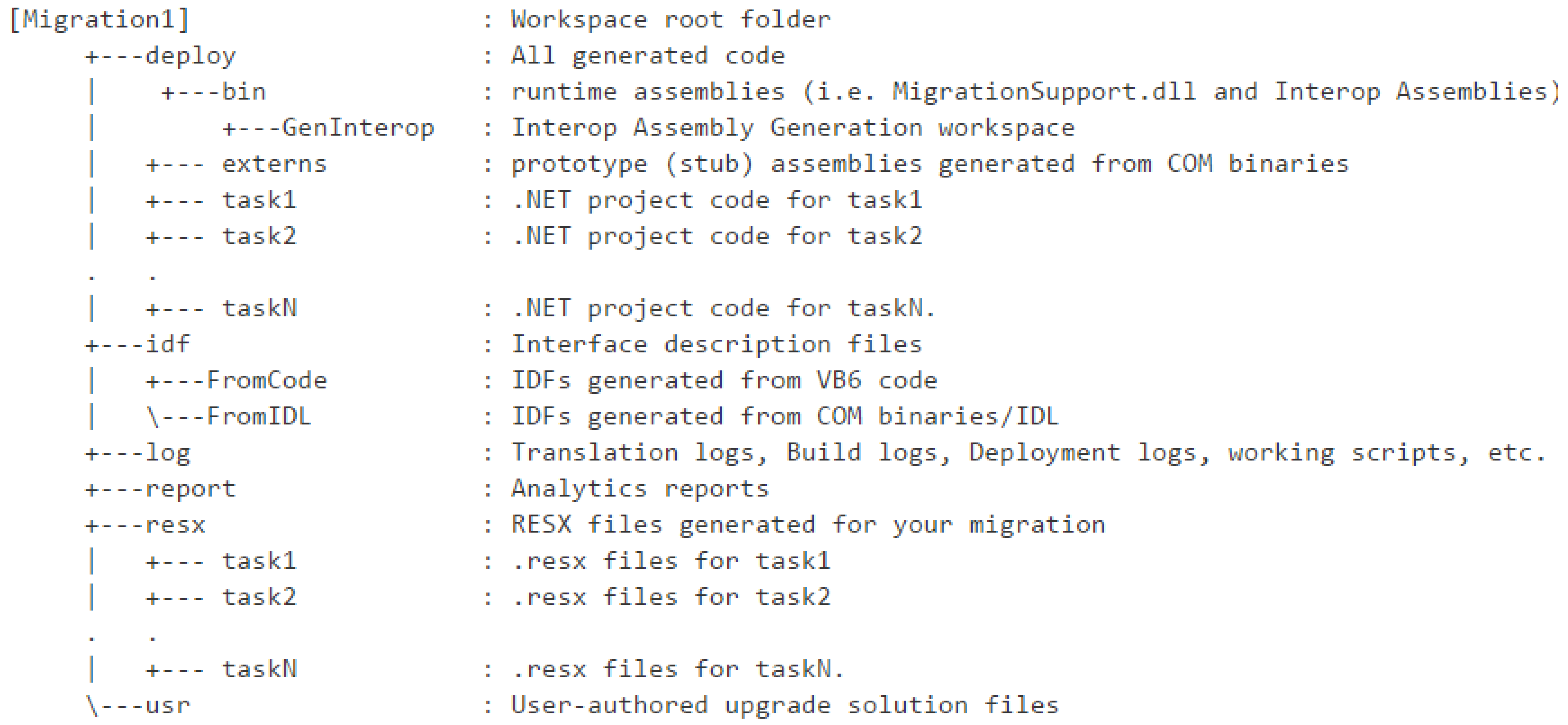
- VB6 to .Net (externals stubbed)
- VB6 to .Net (externals upgraded)
- Look at Code, Build, Report, Run



Technology: Upgrade Solution Files

- gmStudio Project Files
  - Project Settings
  - Project Tasks: VBPs, ASP Pages, Special Scripts
- Translation Script Templates
- COM Interface Description Files (IDFs)
- Project Metalanguage Files
  - Startup File
  - System Language Description
  - Source Language Description
  - Language Translation Rules
  - Code Authoring Rules and Templates
- User Batch Command Script Templates
- Other files: custom scripts, unit tests

Technology: Upgrade Workspace

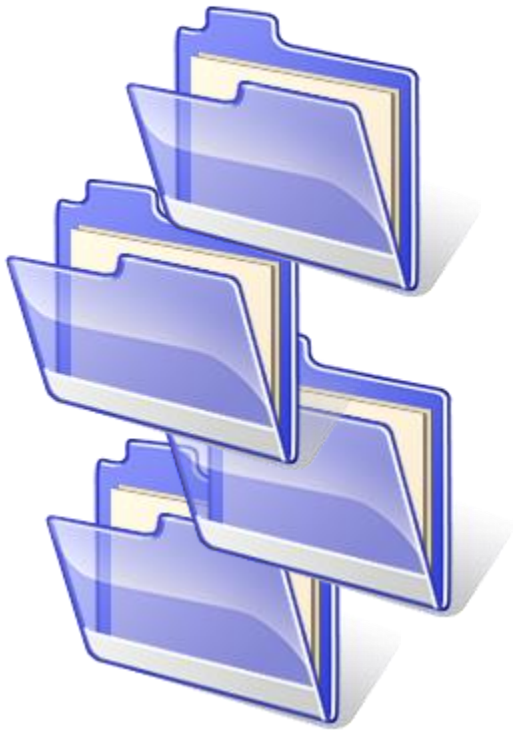


```
[Migration1]                   : Workspace root folder
     +---deploy                : All generated code
     |    +---bin              : runtime assemblies (i.e. MigrationSupport.dll and Interop Assemblies)
     |        +---GenInterop   : Interop Assembly Generation workspace
     |   +--- externs          : prototype (stub) assemblies generated from COM binaries
     |   +--- task1            : .NET project code for task1
     |   +--- task2            : .NET project code for task2
     .   .
     |   +--- taskN            : .NET project code for taskN.
     +---idf                   : Interface description files
     |   +---FromCode          : IDFs generated from VB6 code
     |   \---FromIDL           : IDFs generated from COM binaries/IDL
     +---log                   : Translation logs, Build logs, Deployment logs, working scripts, etc.
     +---report                : Analytics reports
     +---resx                  : RESX files generated for your migration
     |   +--- task1            : .resx files for task1
     |   +--- task2            : .resx files for task2
     .   .
     |   +--- taskN            : .resx files for taskN.
     \---usr                   : User-authored upgrade solution files
```

Technology: Generated Files

- Metadata, Artifacts, Logs
  - Actual Translation Scripts
  - Content Bundles (.BND)
  - Information Files (.VBI)
  - Interface Description Files (.XML)
  - Translation Logs
  - Deployment Logs
  - Build Logs
  - Reports (.TAB, .TXT)

- .NET Code and Binaries
  - Generated .NET Application Codes (Deploy Folder)
  - Generated Stub Classes (instead of interop)
  - Generated Stub Assemblies (instead of interop)
  - Interop Assemblies (rarely used)



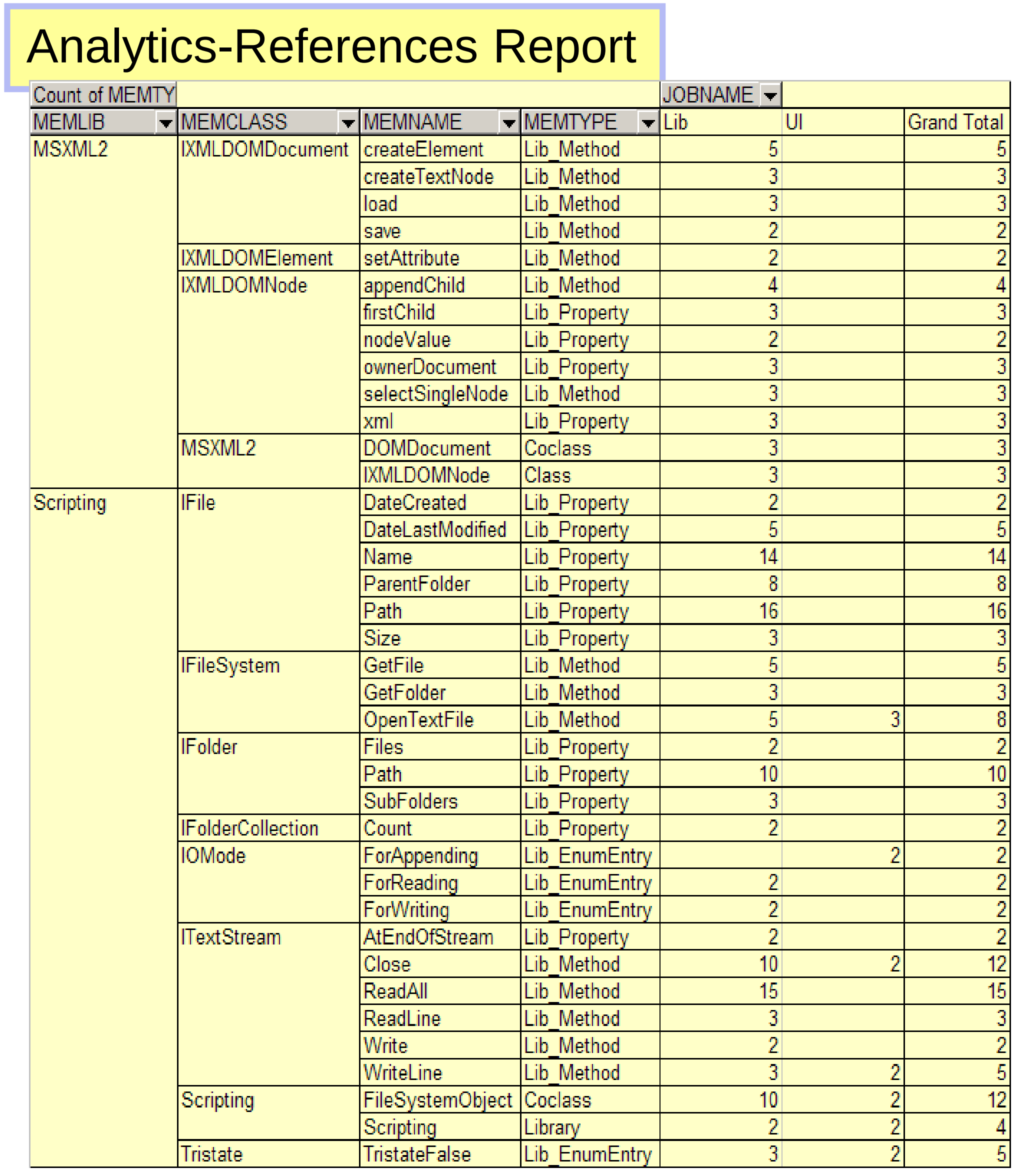
Technology: COM Upgrade, Analysis

- **What really matters:**
  - How you *actually* use COM/VB6 APIs

  AND

  - How you intend to replace them on the new platform.

- All API replacements are **NOT** created equal!



Analytics-References Report

Count of MEMTY				JOBNAME		
MEMLIB	MEMCLASS	MEMNAME	MEMTYPE	Lib	UI	Grand Total
MSXML2	IXMLDOMDocument	createElement	Lib_Method	5		5
		createTextNode	Lib_Method	3		3
		load	Lib_Method	3		3
		save	Lib_Method	2		2
	IXMLDOMElement	setAttribute	Lib_Method	2		2
	IXMLDOMNode	appendChild	Lib_Method	4		4
		firstChild	Lib_Property	3		3
		nodeValue	Lib_Property	2		2
		ownerDocument	Lib_Property	3		3
		selectSingleNode	Lib_Method	3		3
		xml	Lib_Property	3		3
	MSXML2	DOMDocument	Coclass	3		3
		IXMLDOMNode	Class	3		3
Scripting	IFile	DateCreated	Lib_Property	2		2
		DateLastModified	Lib_Property	5		5
		Name	Lib_Property	14		14
		ParentFolder	Lib_Property	8		8
		Path	Lib_Property	16		16
		Size	Lib_Property	3		3
	IFileSystem	GetFile	Lib_Method	5		5
		GetFolder	Lib_Method	3		3
		OpenTextFile	Lib_Method	5	3	8
	IFolder	Files	Lib_Property	2		2
		Path	Lib_Property	10		10
		SubFolders	Lib_Property	3		3
	IFolderCollection	Count	Lib_Property	2		2
	IOMode	ForAppending	Lib_EnumEntry		2	2
		ForReading	Lib_EnumEntry	2		2
		ForWriting	Lib_EnumEntry	2		2
	ITextStream	AtEndOfStream	Lib_Property	2		2
		Close	Lib_Method	10	2	12
		ReadAll	Lib_Method	15		15
		ReadLine	Lib_Method	3		3
		Write	Lib_Method	2		2
		WriteLine	Lib_Method	3	2	5
	Scripting	FileSystemObject	Coclass	10	2	12
		Scripting	Library	2	2	4
	Tristate	TristateFalse	Lib_EnumEntry	3	2	5

Technology: COM Upgrade, Implementation

Declarative Rules
  - Assembly References
  - Namespaces
  - Classes, Structs
  - Enumerations
  - Enum Entries
  - Members, Properties
  - Control Initialization (Designer)
  - Event Handlers
- Dynamic Rules
  - gmSL Scripts
  - Migration DLLs

Baseline COM IDF

+

Hand Customization

=

Custom IDF

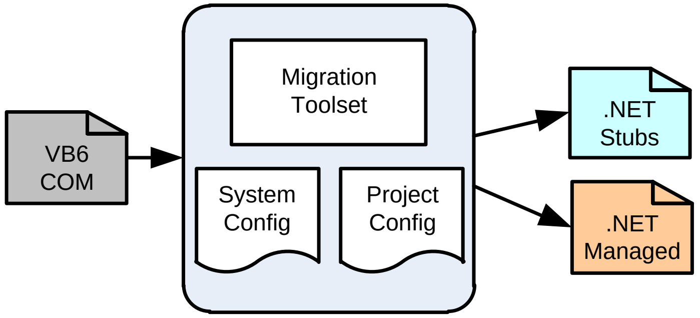
giving

Automated COM Replacement

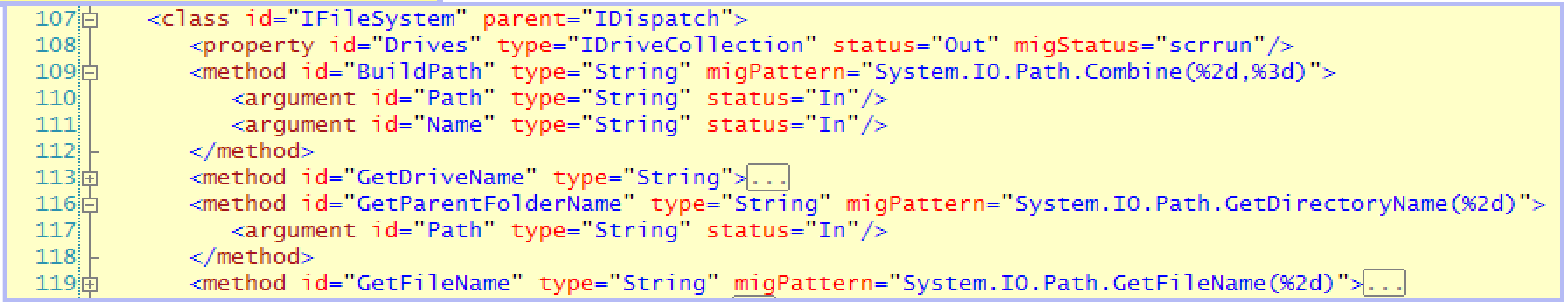
Technology: COM Replacement, Sample

Managed Interface Descriptions

- Template Generated from COM
- Map source API to target API

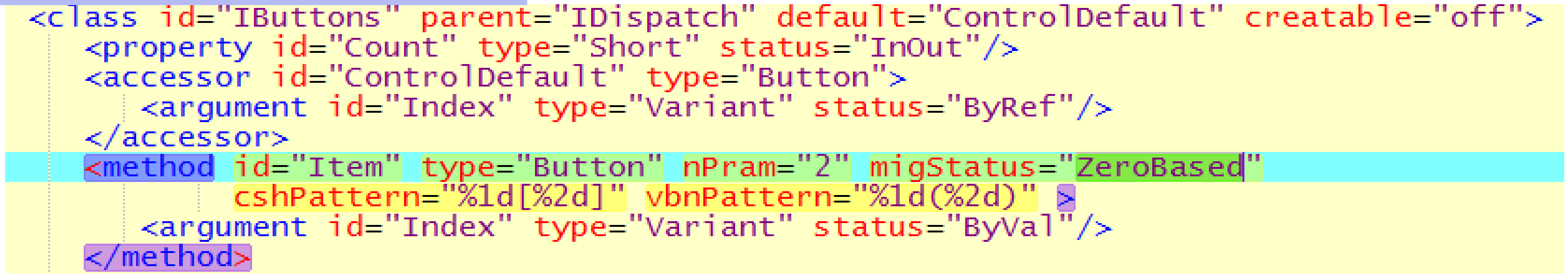


GM.Scrrun.dll.xml



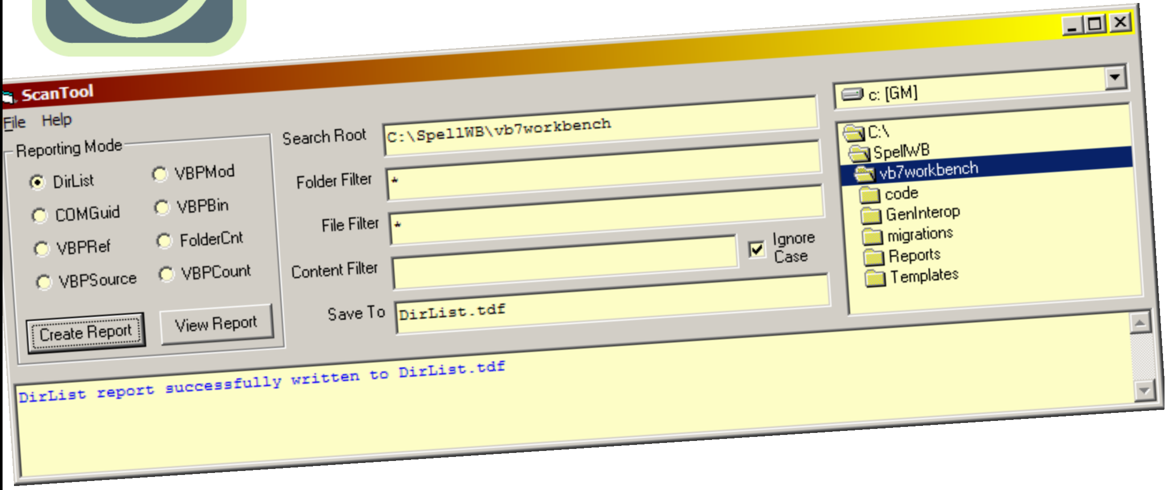
```
<class id="IFileSystem" parent="IDispatch">
   <property id="Drives" type="IDriveCollection" status="Out" migStatus="scrrun"/>
   <method id="BuildPath" type="String" migPattern="System.IO.Path.Combine(%2d,%3d)">
      <argument id="Path" type="String" status="In"/>
      <argument id="Name" type="String" status="In"/>
   </method>
   <method id="GetDriveName" type="String">...
   <method id="GetParentFolderName" type="String" migPattern="System.IO.Path.GetDirectoryName(%2d)">
      <argument id="Path" type="String" status="In"/>
   </method>
   <method id="GetFileName" type="String" migPattern="System.IO.Path.GetFileName(%2d)">...
```

GM.MsComCtLib.dll.xml



```
<class id="IButtons" parent="IDispatch" default="ControlDefault" creatable="off">
   <property id="Count" type="Short" status="InOut"/>
   <accessor id="ControlDefault" type="Button">
      <argument id="Index" type="Variant" status="ByRef"/>
   </accessor>
   <method id="Item" type="Button" nPram="2" migStatus="ZeroBased"
           cshPattern="%1d[%2d]" vbnPattern="%1d(%2d)" >
      <argument id="Index" type="Variant" status="ByVal"/>
   </method>
```

Demo: ScanTool



Search Root: C:\SpellWB\vb7workbench
Folder Filter: *
File Filter: *
Content Filter:
Ignore Case: ✓
Save To: DirList.tdf

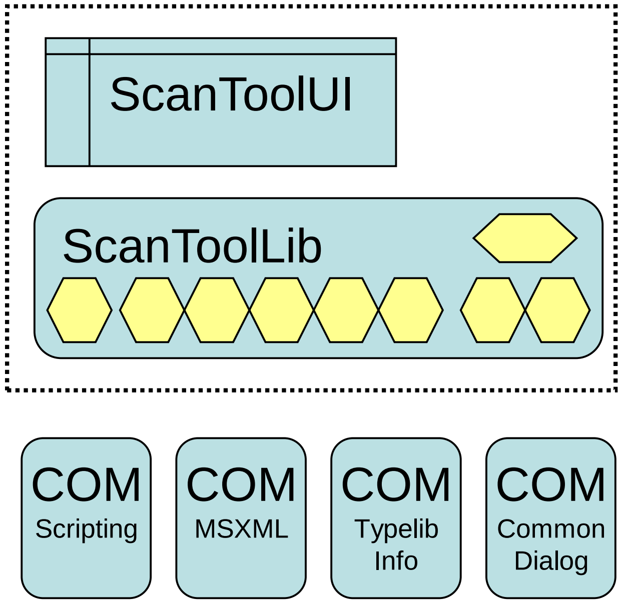
Reporting Mode: [● DirList] ○ COMGuid ○ VBPRef ○ VBPSource ○ VBPMod ○ VBPBin ○ FolderCnt ○ VBPCount

c: [GM]
C:\
SpellWB
vb7workbench
code
GenInterop
migrations
Reports
Templates

```
DirList report successfully written to DirList.tdf
```

- Two VBPs: UI.exe and LIB.dll
- Lib talks back to UI via events
- 4 External COM APIs
- Win32 APIs
- Many VB Intrinsics
- Object Polymorphism
- Error Handling
- Over 2000 LOC

- VB6 to .Net (externals stubbed)
  - Local Stubs
- VB6 to .Net (externals upgraded)
  - COM replacements
  - Custom Runtime
- Look at Code, Build, Report, Run

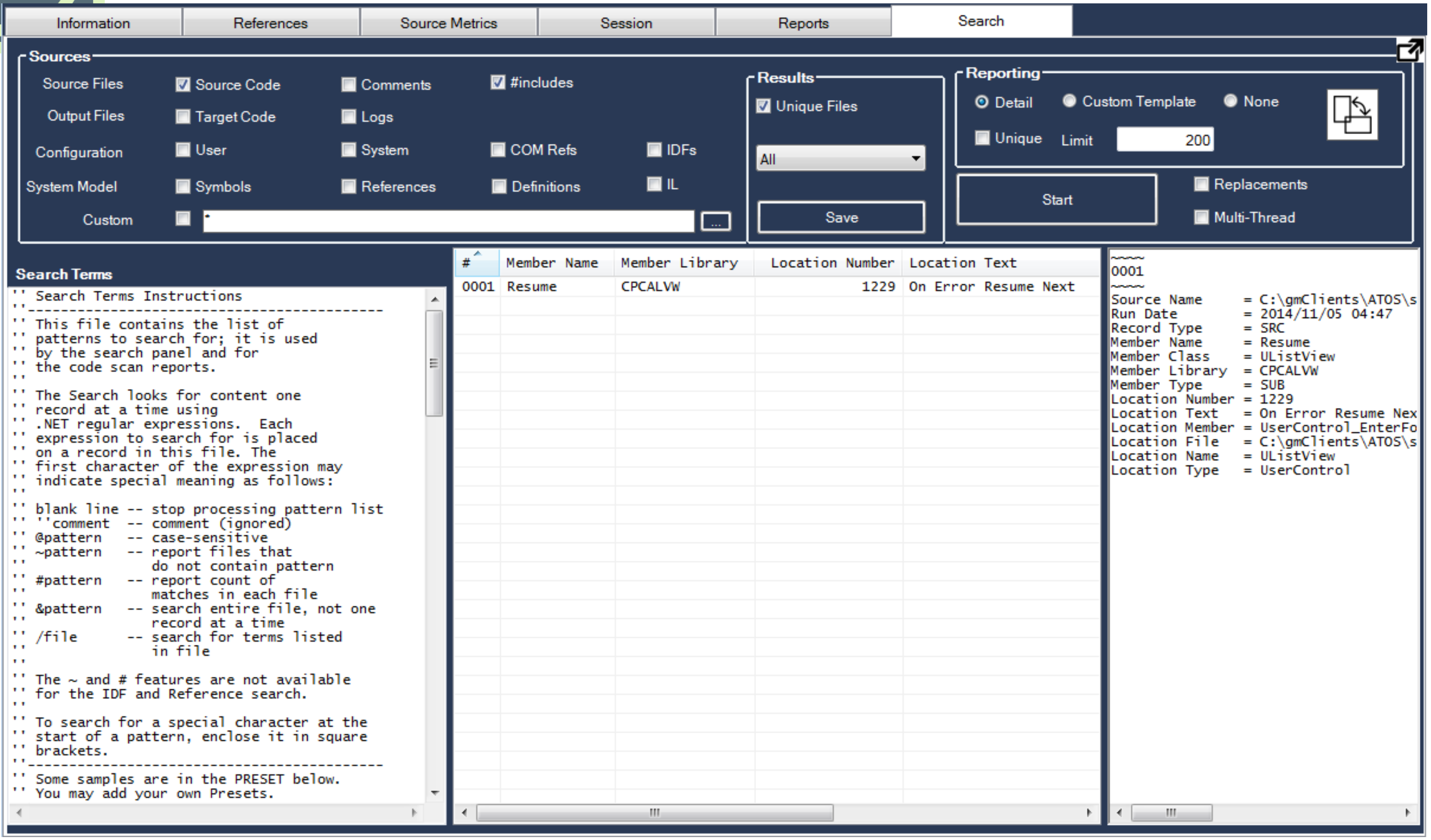


Technology: Reporting

- Run
  - Report menu (open after run)
  - Report Panel (batch runs)
- Types
  - Code Scans
  - Project Reports
  - Model-based
  - Utilities
- Formats
  - Tab-delimted
  - Unformatteed
- Locations
  - Workspace\log
  - Workspace\report

Title	Output File
Code Scan Reports	
Source Structure	[MigName]-SrcStruct.tab
Source References	[MigName]-SrcRef.tab
Source Members	[MigName]-SrcMember.tab
Source GUI Scan	[MigName]-SrcGUI.tab
Source Code Scan	[MigName]-SrcScan.tab
Iceberg	[MigName]-Iceberg.tab
Project Reports	
Project Summary	[MigName]-MigStat.txt
Metrics Summary	[MigName]-Metrics.htm
Migration Set	[MigName]-MigSet.tab
Code Bundles	[MigName]-Bundle.tab
.NET Build Logs	[MigName]-BldLog.tab
Translation Logs	[MigName]-TranLog.tab
All Logs	[MigName]-AllLog.txt
Interface File Headers	[MigName]-LibHeaders.tab
Interface File ProgIds	[MigName]-LibProgIds.tab
Semantic Model Reports	
Semantic References	[MigName]-AnaRef.tab
Semantic Definitions	[MigName]-AnaDef.tab
Semantic Symbols	[MigName]-AnaSym.tab
Semantic Audit	[MigName]-Audit.txt
Utility Reports	
Migration Project List	gmProjects.tab
Multi-Unit Script	tran.[MigName]_MultiJob.xml
Target Code Scan	[MigName]-BndScan.tab
Target Code Changes	[MigName]-BndChanges.txt

Technology: Searching



Sources

Source Files	[✓] Source Code	[ ] Comments	[✓] #includes	
Output Files	[ ] Target Code	[ ] Logs		
Configuration	[ ] User	[ ] System	[ ] COM Refs	[ ] IDFs
System Model	[ ] Symbols	[ ] References	[ ] Definitions	[ ] IL
Custom	[ ] *			

Results

[✓] Unique Files

All

Save

Reporting

[● Detail] ○ Custom Template ○ None

[ ] Unique Limit 200

Start

[ ] Replacements

[ ] Multi-Thread

Search Terms

```
'' Search Terms Instructions
''----------------------------------------------
'' This file contains the list of
'' patterns to search for; it is used
'' by the search panel and for
'' the code scan reports.
''
'' The Search looks for content one
'' record at a time using
'' .NET regular expressions.  Each
'' expression to search for is placed
'' on a record in this file. The
'' first character of the expression may
'' indicate special meaning as follows:
''
'' blank line -- stop processing pattern list
'' ''comment  -- comment (ignored)
'' @pattern   -- case-sensitive
'' ~pattern   -- report files that
''               do not contain pattern
'' #pattern   -- report count of
''               matches in each file
'' &pattern   -- search entire file, not one
''               record at a time
'' /file      -- search for terms listed
''               in file
''
'' The ~ and # features are not available
'' for the IDF and Reference search.
''
'' To search for a special character at the
'' start of a pattern, enclose it in square
'' brackets.
''----------------------------------------------
'' Some samples are in the PRESET below.
'' You may add your own Presets.
```

#	Member Name	Member Library	Location Number	Location Text
0001	Resume	CPCALVW	1229	On Error Resume Next

```
~~~~
0001
~~~~
Source Name     = C:\gmClients\ATOS\s
Run Date        = 2014/11/05 04:47
Record Type     = SRC
Member Name     = Resume
Member Class    = UListView
Member Library  = CPCALVW
Member Type     = SUB
Location Number = 1229
Location Text   = On Error Resume Nex
Location Member = UserControl_EnterFo
Location File   = C:\gmClients\ATOS\s
Location Name   = UListView
Location Type   = UserControl
```

Search, Drill Down, Report

Technology: SQL Analysis

- Bulk Insert to SQL
- Various Procs and Queries
- Load into Excel

Title	Output File
Code Scan Reports	
Source Structure	[MigName]-SrcStruct.tab
Source References	[MigName]-SrcRef.tab
Source Members	[MigName]-SrcMember.tab
Source GUI Scan	[MigName]-SrcGUI.tab
Source Code Scan	[MigName]-SrcScan.tab
Project Reports	
Project Summary	[MigName]-MigStat.txt
Semantic Model Reports	
Semantic References	[MigName]-AnaRef.tab
Semantic Definitions	[MigName]-AnaDef.tab

Technology: More Reengineering

Basic Transformations

- Replace COM/Win32 APIs with .NET replacements
- Reauthor, Remove, or Stubout a member, class, file or entire component
- Control target file names, folder names, etc.
- Control target Visual Studio project files (resx, assemblyinfo, *proj)
- Control formatting – blank lines, comments, indenting, boilerplate code
- Specify settings that control internal translator operation

Advanced Transformations

- Generate a complete skeleton of all application and external components
- Consolidate of Shared files into a new or existing host assembly
- Convert of COM classes to WCF web services
- Convert shared module state to thread-isolated state
- Break build cycles: convert circular references to interface references
- Define rules to map ASP/VB6 language elements to .NET coding patterns
- Shared Files Consolidation
- Custom

Technology: Select Command

- Select Identifier Attributes
- Select Value Attributes
- Select Enumerated Attributes
- Select Search String Attributes
- Select Location String Attributes
- Select ComputeConditional String Attribute
- Select Author Flag Attributes
- Select Compiler Flag Attributes
- Select Analyser Flag Attributes
- Select Process Flag Attributes

Technology: Registry Command

- Dependency: Specify a possibly omitted include file dependency
- EditFile: Supply a set of Fix statements for a specified file
- FixType: Fix the type of a source component
- FixStatus: Specify an ASP page status
- Guid: Define the value of a GUID
- IdfStatus: Specify the Interface Description File status of an external
- Include: Specify the path to an include file
- LibName: Specify a library name or file name
- OverLoadArgument: Specify types for arguments to be overloaded
- ProgId: Resolve a ProgId
- RefactorFile: Supply a set of Refactor statements for a specified file
- SharedFile: Specifies that a file is shared by multiple VBPs
- UsesInterfaces: Specifies that a project file uses certain interfaces

Technology: Fix/Replace

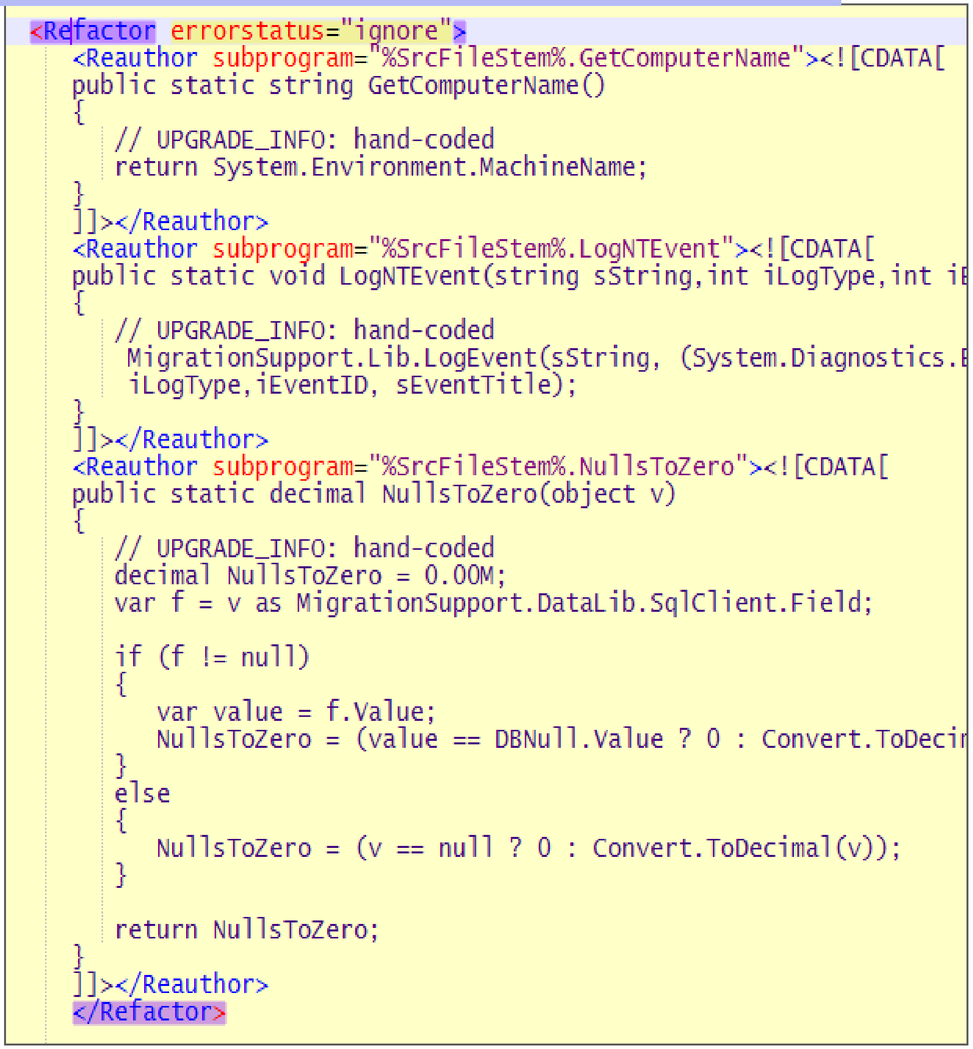
- When to use
  - Correcting source VB6 coding errors
  - Correcting rare exceptions
  - Work arounds
- Types of Fixes
  - Source Code Fix (Compile/Fix/Replace)
  - Target Code Fix (Author/Fix/Replace@lang="csh")
  - Target Project Fix (Author/Fix/Replace@lang="csproj")
  - Whole File (Author/Fix/ReplaceFile)
  - Target Stub Class (Author/Fix@FileFilter="[lib.dll]"/Replace)
  - Any File Fix (Fix@FileFilter="path"/Replace)
  - Regex Fix (bundle) (Author/Fix/Replace@status="regex")

Technology: Refactor Command

CallByName	Changes symbol-related code events that yield CallByName late binding calls into direct boxed calls.
Extend	Extends the content of a class by adding new components.
FixType	Changes the binary type of a component or group of components
Implements	Specifies that a VB6 class implements another class or interface.
MigClass	Introduces a new class that contains related refactoring information used for complex migration operations, especially as related to designer code.
Migrate	Specifies migration of a specific symbol introduced via an external library description.
Reauthor	Replaces the content of a subprogram with a completely rewritten block of code
Remove	Prevents a component from being authored
Rename	Changes the authored name of components
Replace	Replaces either the members of an external class or the patterns of opcodes via replacement declarations.

Technology : Refactor/Reauthor

Translation Script – Refactor/Reauthor



```
<Refactor errorstatus="ignore">
   <Reauthor subprogram="%SrcFileStem%.GetComputerName"><![CDATA[
   public static string GetComputerName()
   {
      // UPGRADE_INFO: hand-coded
      return System.Environment.MachineName;
   }
   ]]></Reauthor>
   <Reauthor subprogram="%SrcFileStem%.LogNTEvent"><![CDATA[
   public static void LogNTEvent(string sString,int iLogType,int iE
   {
      // UPGRADE_INFO: hand-coded
       MigrationSupport.Lib.LogEvent(sString, (System.Diagnostics.E
       iLogType,iEventID, sEventTitle);
   }
   ]]></Reauthor>
   <Reauthor subprogram="%SrcFileStem%.NullsToZero"><![CDATA[
   public static decimal NullsToZero(object v)
   {
      // UPGRADE_INFO: hand-coded
      decimal NullsToZero = 0.00M;
      var f = v as MigrationSupport.DataLib.SqlClient.Field;

      if (f != null)
      {
         var value = f.Value;
         NullsToZero = (value == DBNull.Value ? 0 : Convert.ToDeci
      }
      else
      {
         NullsToZero = (v == null ? 0 : Convert.ToDecimal(v));
      }

      return NullsToZero;
   }
   ]]></Reauthor>
   </Refactor>
```

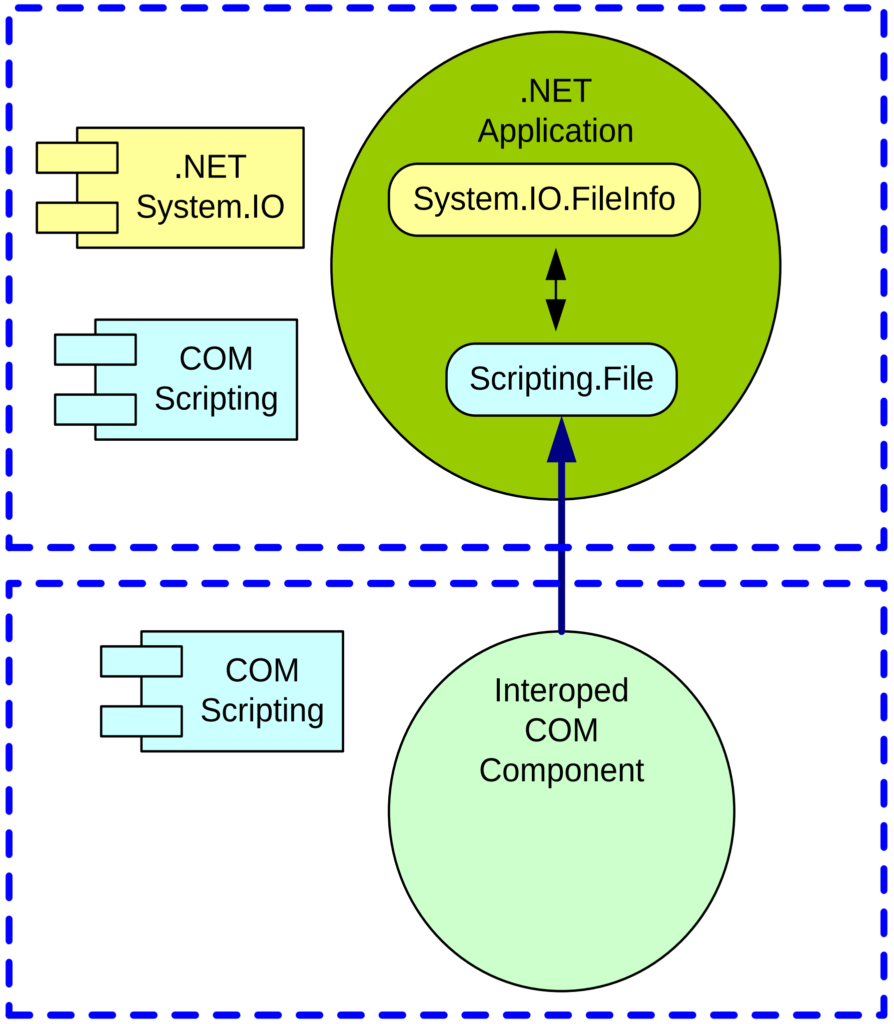
BEFORE

```
public static string GetComputerName()
{
   string GetComputerName = "";

   //  Set or retrieve the name of the computer.
   string strBuffer = "";
   int lngLen = 0;

   strBuffer = VBNET.Strings.Space(255 + 1);
   lngLen = VBNET.Strings.Len(strBuffer);
   if  (Convert.ToBoolean(GetComputerNameAPI(
         strBuffer,out lngLen)))
   {
      GetComputerName =
    VBNET.Strings.Left(strBuffer,lngLen);
   }
   else
   {
      GetComputerName = "";
   }
   return GetComputerName;
}
```

AFTER

```
public static string GetComputerName()
{
   // UPGRADE_INFO: hand-coded
   return System.Environment.MachineName;
}
```

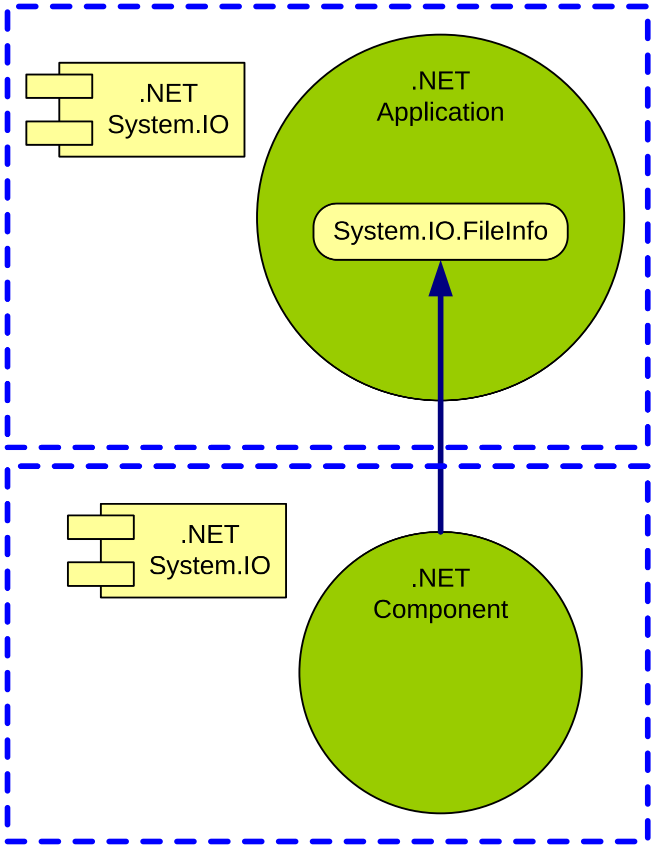
Sidebar: API Crossing



Be wary of Top-Down Migrations

- COM interfaces have COM types as member parameters and return types.
- If you interop COM components, your .NET clients will end up straddling the fence between COM and .NET and this will require more interop code which runs counter to the premise of adopting .NET in first place.

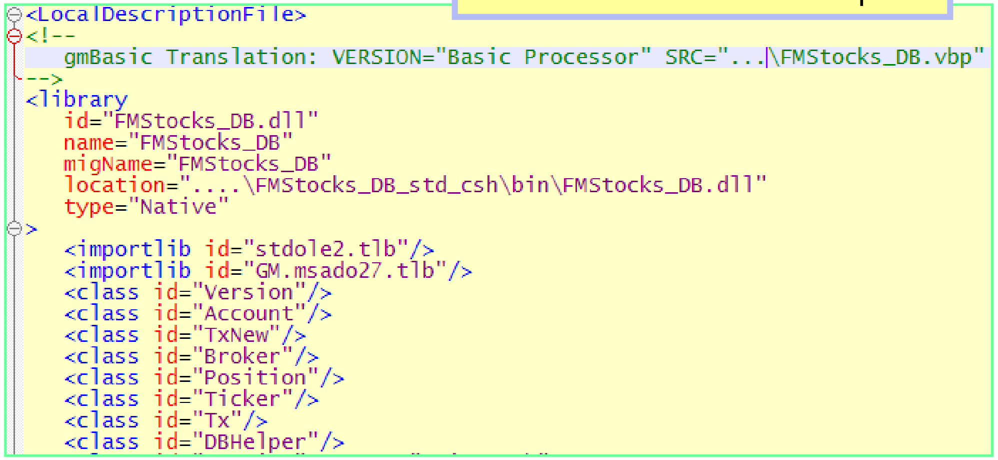
Sidebar: API Crossing



Bottom-Up, SmartPort Migrations

- **gmBasic remembers what it has translated and *knows* which components are going to .NET**
- **gmBasic knows how interfaces are changing to use new types**
- **gmBasic uses this information to generate clean, native code in client applications.**

Generated Interface Description



```
<LocalDescriptionFile>
<!--
   gmBasic Translation: VERSION="Basic Processor" SRC="...\FMStocks_DB.vbp"
-->
<library
   id="FMStocks_DB.dll"
   name="FMStocks_DB"
   migName="FMStocks_DB"
   location="....\FMStocks_DB_std_csh\bin\FMStocks_DB.dll"
   type="Native"
>
   <importlib id="stdole2.tlb"/>
   <importlib id="GM.msado27.tlb"/>
   <class id="Version"/>
   <class id="Account"/>
   <class id="TxNew"/>
   <class id="Broker"/>
   <class id="Position"/>
   <class id="Ticker"/>
   <class id="Tx"/>
   <class id="DBHelper"/>
```