Extreme Windows Desktop Integration

"A modern Windows application needs more than just blue toolbars, resizing or a treeview…"

Consultingwerk
software architecture and development

Be there to win!

All visitors of our booth and attendees of our presentations or workshops that eave a business card or fill out a short form will enter a drawing for an Apple iPod Touch.

The lucky winner will be announced at the end of the conference.

- **Getting started with Embedded Windows,** A practical introduction into WinKitLE (practical hands-on workshop), Mike Fechner & Marko Rüterbories, Sunday, June 5th, 1:30 – 4:30
- **SmartComponent Library: GUI for .NET and OERA the productive way!** (Commercial presentation), Mike Fechner & Marko Rüterbories, Monday, June 6th, 4:00 – 5:00
- **Extending the OpenEdge Architect Visual Designer,** Mike Fechner, Tuesday, June 7th, 4:00 – 5:00
- **Extreme Windows Desktop Integration,** Mike Fechner, Wednesday, June 8th, 11:15 – 12:15

www.consultingwerk.de

Visit us at booth 11

Mike Fechner, Director, Consultingwerk Ltd.

mike.fechner@consultingwerk.de

PUG Challenge Americas

Tuesday, June 7th, 2011

Mike Fechner, Consultingwerk Ltd.

- Independent IT consulting organization
- Focusing on **OpenEdge** and **.NET**
- Located in Cologne, Germany
- Vendor of tools and consulting programs
- 21 years of Progress experience (V5 … V10)
- GUI for .NET early adaptor (since 10/2006)
- Just started with iPhone/iPad app development

Mike Fechner, Consultingwerk Ltd.

- Customers in Germany, Europe, USA
- Working with small to large Progress Partners and direct end users
- Supporting some of the largest Progress Partners in Germany, Belgium, The Netherlands, Austria and UK with application modernization and user interface technologies
- Network of partnering consultants, like web4biz for Web UI’s, Whitestar Software, DBAppraise

Solutions for the OpenEdge GUI for .NET

Integrate existing applications into GUI for .NET™
using WinKit

Implement OERA & achieve true productivity with GUI for .NET™
using SmartComponent Library

Instant migration of Dynamics™ applications
with Dynamics4.NET

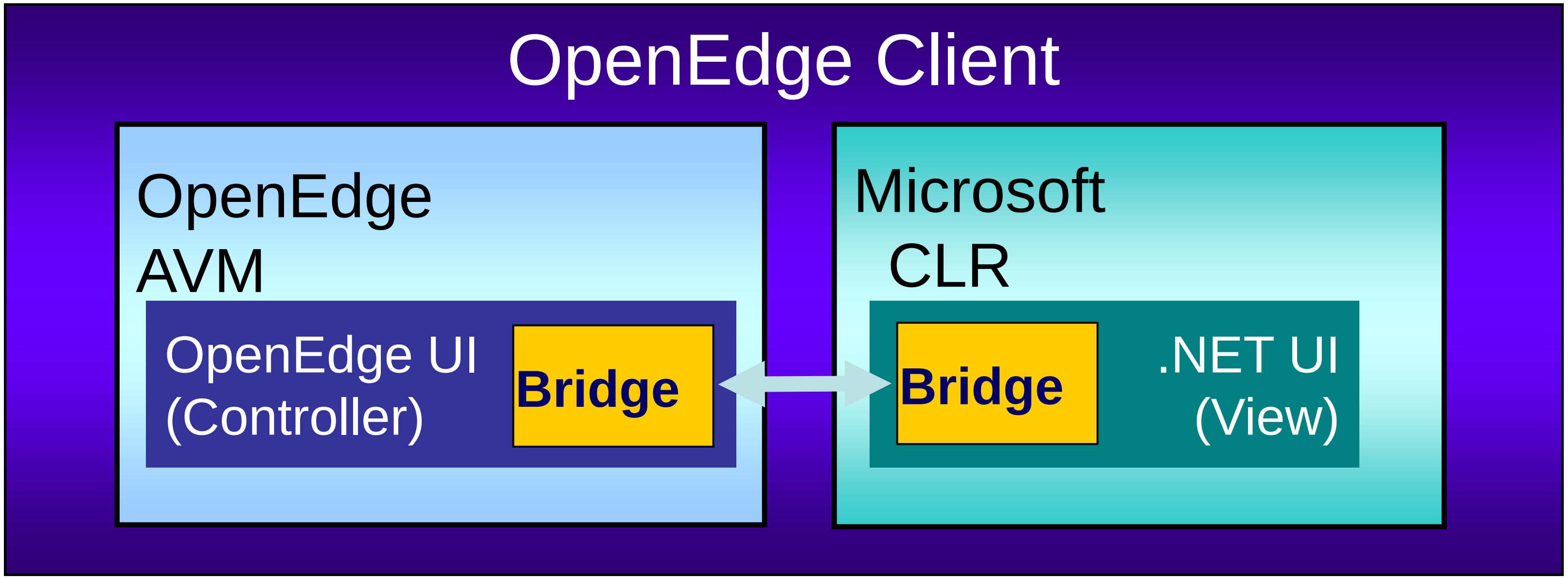
Access to sample code download

- http://blog.consultingwerk.de

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

AVM and .NET CLR in the same process



- GUI components
- Data access
- I/O blocking
- Event handlers

- GUI Visualization
- User interaction and events

OpenEdge AVM access to .NET classes

- Full access to all .NET classes, not just WinForms „GUI“ classes, restrictions only in multi-threading
- License restrictions removed in 10.2B02
  http://blog.consultingwerk.de/consultingwerkblog/2010/09/progress-relaxes-usage-restrictions-of-gui-for-net-bridge-with-10-2b-service-pack/
- Currently .NET access in GUI clients, potentially TTY and AppServer on Windows in OE11
  http://communities.progress.com/pcom/thread/38970?tstart=0
- **.NET classes can be used with the classical GUI**
- Most examples in this presentation do not require a GUI for .NET user interface

Access to COM objects

- COM is pre .NET technology
- COM-HANDLE data type
- Still supported in 10.2A/10.2B
- Active X Controls in Embedded ABL Windows are well supported
- Access to Active X objects from the ABL side of the AVM

COM-Interop

- Supported since 10.2B
- Active X objects (automation server) wrapped into .NET classes
- Microsoft Office, Enterprise Architect, …
- Advantages
  - type safeness (compile time validation)
  - faster than COM-HANDLE
  - automated garbage collection (no RELEASE OBJECT)
- Disadvantages
  - **no event handler, no System.EventHandler delegates!**
  - unable to mix & match COM-Handle with COM-Interop

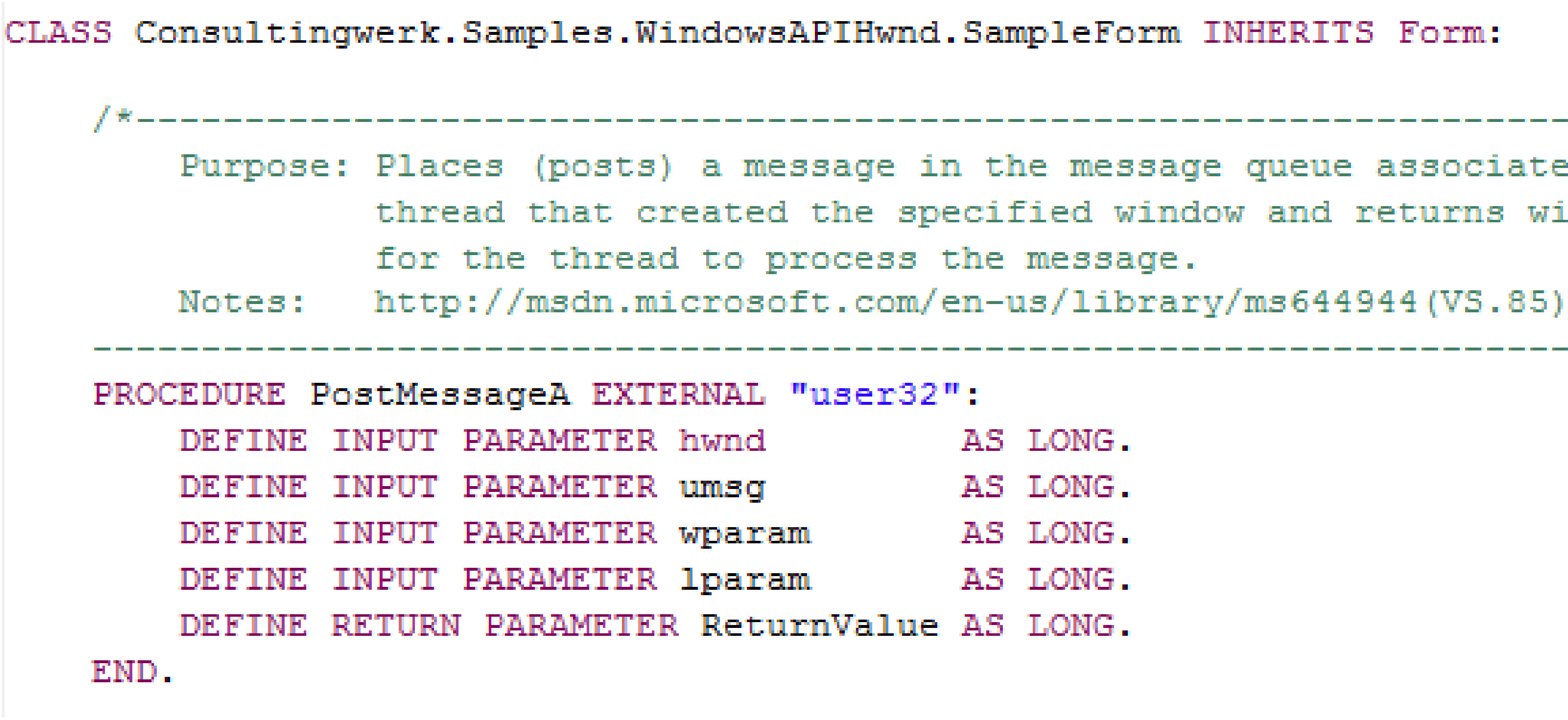
Windows DLL API Access

- External procedure definition
- Access to Windows API, 3rd party API, Progress/OpenEdge version independent
- External procedures also in classes… ***scary!***

- For .NET Controls: *Handle* property is the equivalent to *HWND* (Windows API Handle)

```
button2:Handle:ToInt32() -> HWND
```

Sample External DLL function in ABL class



```
CLASS Consultingwerk.Samples.WindowsAPIHwnd.SampleForm INHERITS Form:

    /*------------------------------------------------------------------------
        Purpose: Places (posts) a message in the message queue associate
                 thread that created the specified window and returns wi
                 for the thread to process the message.
        Notes:   http://msdn.microsoft.com/en-us/library/ms644944(VS.85)
    ------------------------------------------------------------------------
    PROCEDURE PostMessageA EXTERNAL "user32":
        DEFINE INPUT PARAMETER hwnd        AS LONG.
        DEFINE INPUT PARAMETER umsg        AS LONG.
        DEFINE INPUT PARAMETER wparam      AS LONG.
        DEFINE INPUT PARAMETER lparam      AS LONG.
        DEFINE RETURN PARAMETER ReturnValue AS LONG.
    END.
```

P/Invoke, P-Invoke

- .NET equivalent of external procedures (DLL functions)
- Used in many .NET API samples (Google, Codeproject)
- DllImport annotation in C#
- **Not compatible with GUI for .NET bridge**
- **Use ABL’s PROCEDURE EXTERNAL instead – you’ll just have to translate syntax**

```
[DllImport("User32 l", SetL E or=true)]

static extern Boole n M sageBe p (UInt32 beepType);
```

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

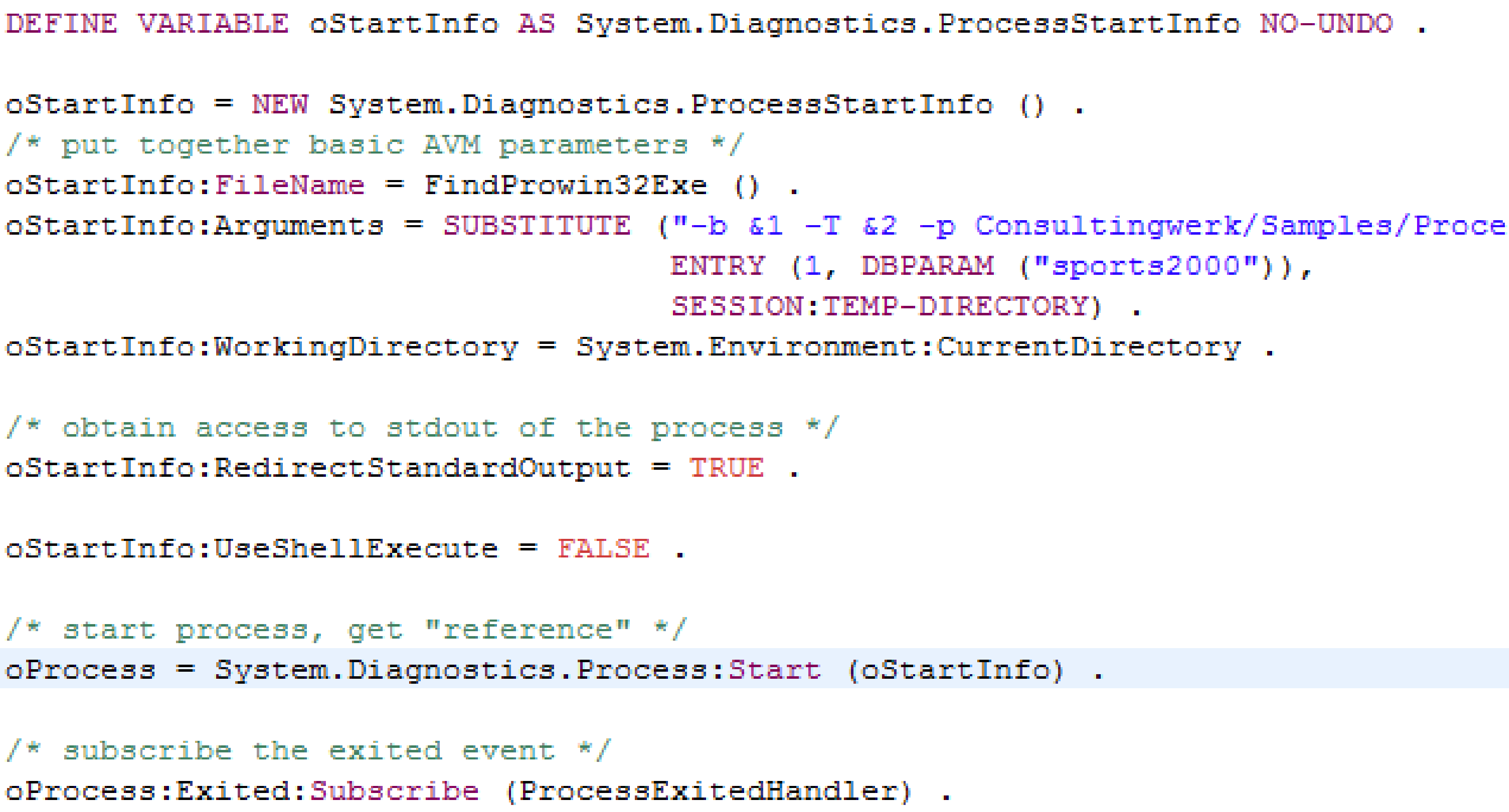
External Processes

- ***System.Diagnostics.Process*** class, ***System.Diagnostics.ProcessStartInfo*** class
- Start/Query/Stop external processes
- System.Diagnostics.Process:GetCurrentProcess() method
- Useful for monitoring Progress batch clients (work around multi-threading) or any other process you need to spawn from your application
- Allows GUI client to remain active but still “wait” for completion of the client

Getting command line of current process

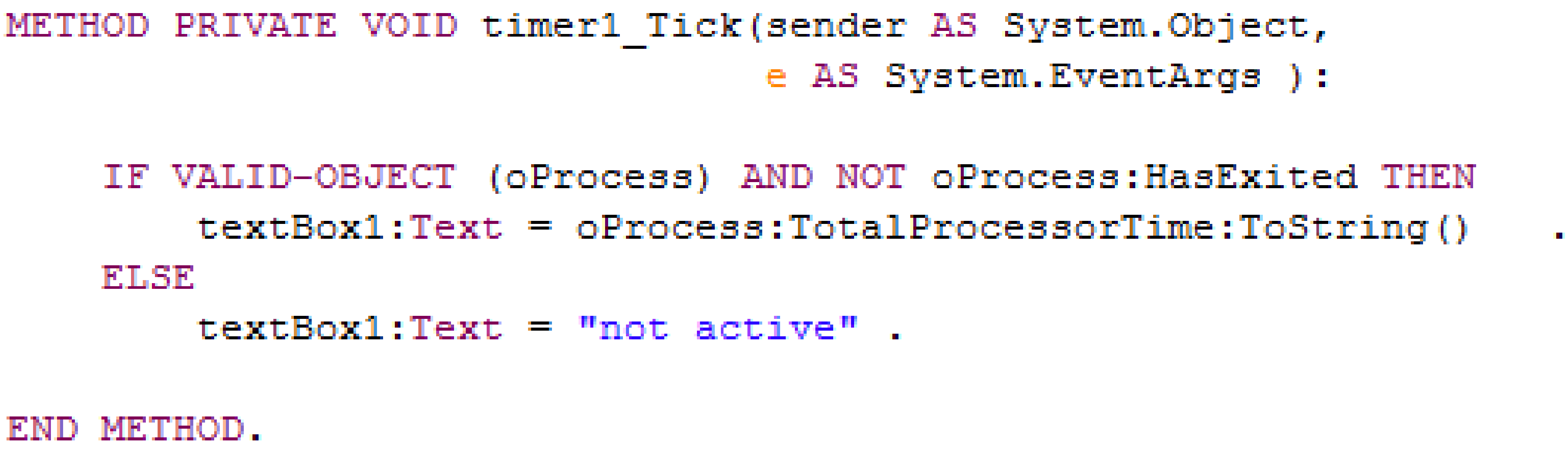
- ***System.Environment:CommandLine*** (CHARACTER) returns path to prowin32.exe plus all parameters
- Use ABL SESSION-Handle or DBPARAM etc. for achieving zero configuration background clients
- ***System.Environment:CurrentDirectory*** for working directory of current prowin32.exe

Launching a batch client



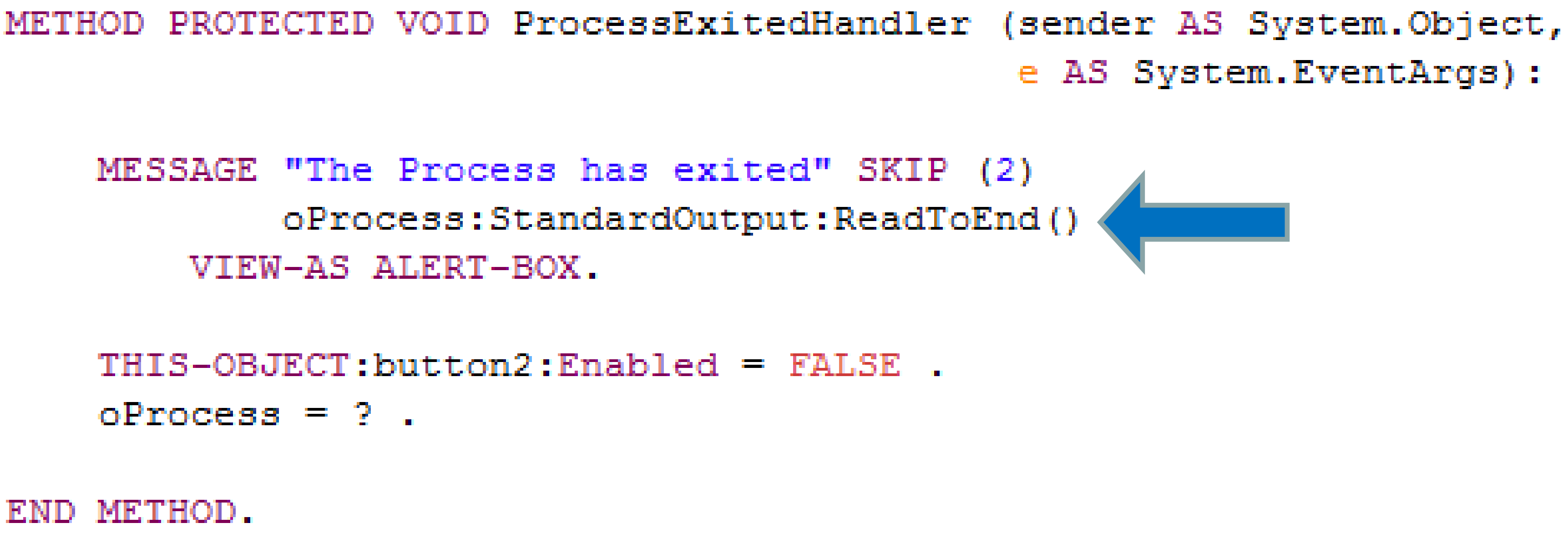
```
DEFINE VARIABLE oStartInfo AS System.Diagnostics.ProcessStartInfo NO-UNDO .

oStartInfo = NEW System.Diagnostics.ProcessStartInfo () .
/* put together basic AVM parameters */
oStartInfo:FileName = FindProwin32Exe () .
oStartInfo:Arguments = SUBSTITUTE ("-b &1 -T &2 -p Consultingwerk/Samples/Proce
                                   ENTRY (1, DBPARAM ("sports2000")),
                                   SESSION:TEMP-DIRECTORY) .
oStartInfo:WorkingDirectory = System.Environment:CurrentDirectory .

/* obtain access to stdout of the process */
oStartInfo:RedirectStandardOutput = TRUE .

oStartInfo:UseShellExecute = FALSE .

/* start process, get "reference" */
oProcess = System.Diagnostics.Process:Start (oStartInfo) .

/* subscribe the exited event */
oProcess:Exited:Subscribe (ProcessExitedHandler) .
```

Query if process is still active (Timer event)



```
METHOD PRIVATE VOID timer1_Tick(sender AS System.Object,
                                e AS System.EventArgs ):

    IF VALID-OBJECT (oProcess) AND NOT oProcess:HasExited THEN
        textBox1:Text = oProcess:TotalProcessorTime:ToString()    .
    ELSE
        textBox1:Text = "not active" .

END METHOD.
```

Reading stdout in Exited Event handle



```
METHOD PROTECTED VOID ProcessExitedHandler (sender AS System.Object,
                                            e AS System.EventArgs):

    MESSAGE "The Process has exited" SKIP (2)
            oProcess:StandardOutput:ReadToEnd()
        VIEW-AS ALERT-BOX.

    THIS-OBJECT:button2:Enabled = FALSE .
    oProcess = ? .

END METHOD.
```

- CHARACTER value
- May read it while the process is still active, but that would block your session when there is no data

Demo

- Launch Batch client and get result via stdout

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

File System Operation

- System.IO.Directory, DirectoryInfo classes
- System.IO.File, FileInfo classes

- Typically more options than OS… statements in ABL
- Exception (errors) integrated into structured error handling

- System.IO.FileSystemWatcher Component

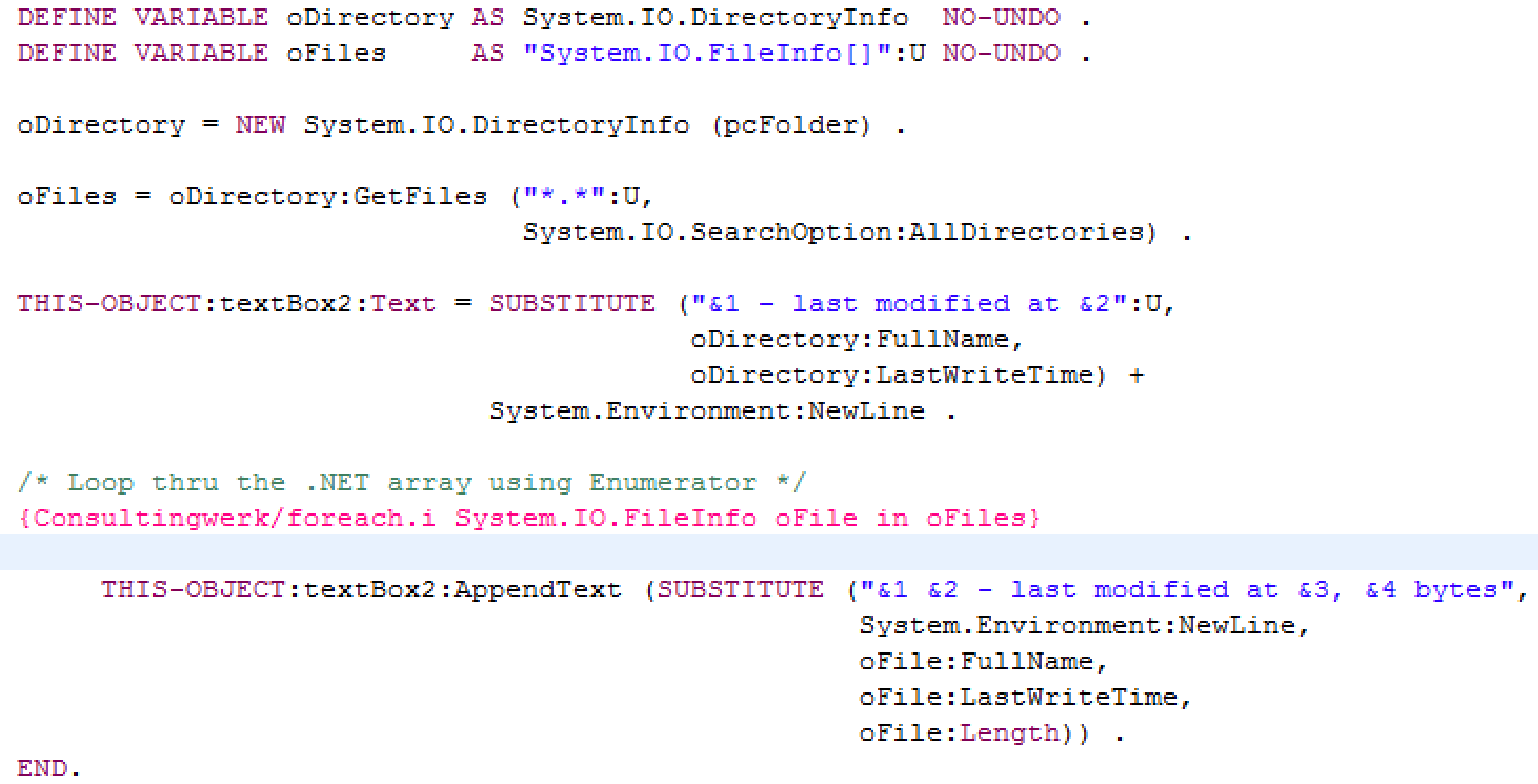
System.IO.Directory

- System.IO.Directory: Static methods
- System.IO.DirectoryInfo: Instance methods, properties, useful when accessing multiple members
- Query and Manipulate Directory properties (rename, create, delete, query/modify access date, **access control**)
- Query all contained directories/files including subfolders (better than INPUT FROM OS-DIR)
- …

System.IO.File

- System.IO.File: Static methods (more like FILE-INFO Handle)
- System.IO.FileInfo: Instance methods, properties, useful when accessing multiple members
- Query and Manipulate File properties (rename, create, copy, delete, query/modify access date, **access control**)
- better than FILE-INFO Handle (no session wide dependencies)
- …

Query all files in a folder

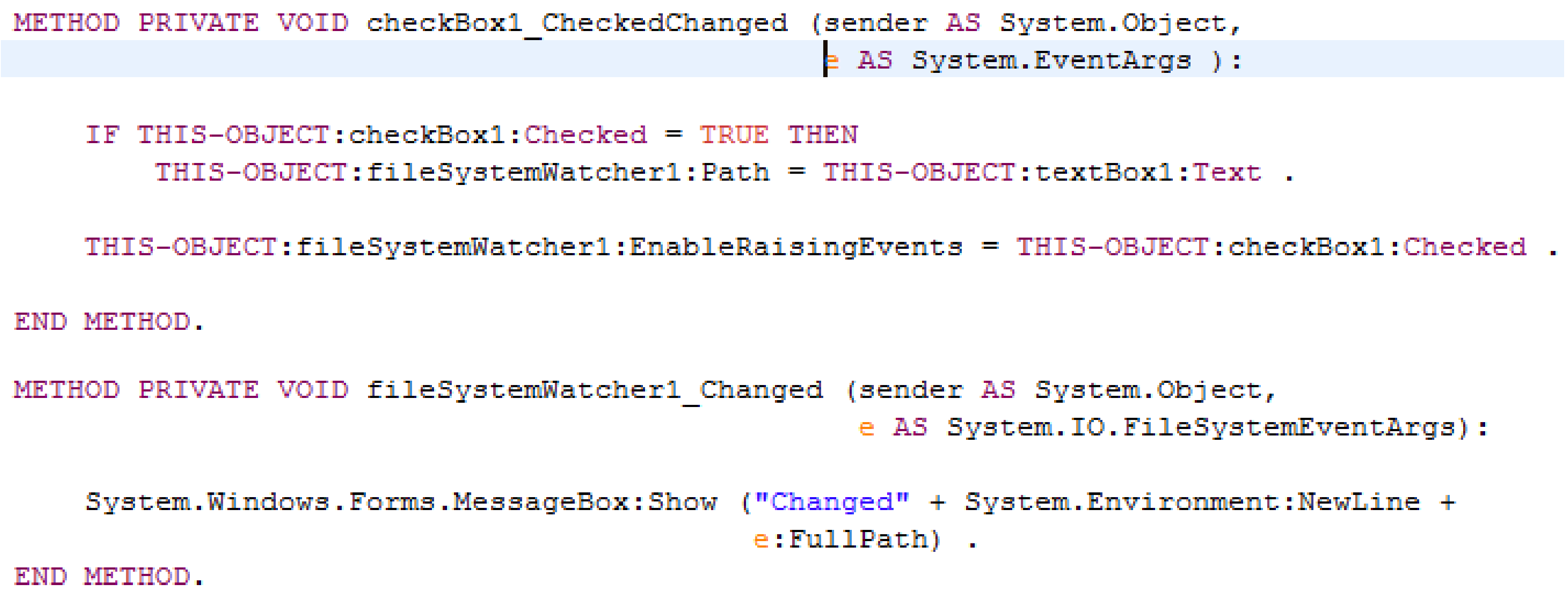


```
DEFINE VARIABLE oDirectory AS System.IO.DirectoryInfo  NO-UNDO .
DEFINE VARIABLE oFiles     AS "System.IO.FileInfo[]":U NO-UNDO .

oDirectory = NEW System.IO.DirectoryInfo (pcFolder) .

oFiles = oDirectory:GetFiles ("*.*":U,
                              System.IO.SearchOption:AllDirectories) .

THIS-OBJECT:textBox2:Text = SUBSTITUTE ("&1 - last modified at &2":U,
                                        oDirectory:FullName,
                                        oDirectory:LastWriteTime) +
                            System.Environment:NewLine .

/* Loop thru the .NET array using Enumerator */
{Consultingwerk/foreach.i System.IO.FileInfo oFile in oFiles}

     THIS-OBJECT:textBox2:AppendText (SUBSTITUTE ("&1 &2 - last modified at &3, &4 bytes",
                                                  System.Environment:NewLine,
                                                  oFile:FullName,
                                                  oFile:LastWriteTime,
                                                  oFile:Length)) .
END.
```

System.IO.FileSystemWatcher

- Component (can be placed in Visual Designer, but not a Control), can be created on the fly (also from non GUI for .NET)
- Can raise events when files in a folder (incl. sub folders) are created, modified or deleted
- Notified by OS (no need to poll for changes)
- Event buffer (64 k) in case the process is busy (see MSDN for details)

FileSystemWatcher example



```
METHOD PRIVATE VOID checkBox1_CheckedChanged (sender AS System.Object,
                                                    e AS System.EventArgs ):

    IF THIS-OBJECT:checkBox1:Checked = TRUE THEN
        THIS-OBJECT:fileSystemWatcher1:Path = THIS-OBJECT:textBox1:Text .

    THIS-OBJECT:fileSystemWatcher1:EnableRaisingEvents = THIS-OBJECT:checkBox1:Checked .

END METHOD.

METHOD PRIVATE VOID fileSystemWatcher1_Changed (sender AS System.Object,
                                                      e AS System.IO.FileSystemEventArgs):

    System.Windows.Forms.MessageBox:Show ("Changed" + System.Environment:NewLine +
                                              e:FullPath) .
END METHOD.
```

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

Drag & Drop

- Within your application
- To your application
- From your application

- Classic ABL GUI does only support FILE-DROP (receiving file-names from Windows Explorer etc.)
- Embedded Windows (WinKit) are a way to support Drag & Drop in classic GUI

Drag & Drop elements

- System.Windows.Forms.***Control:DoDragDrop*** (data AS System.Object, allowedEffects AS DragDropEffects) initiates Drag & Drop operation
- Effects: ***Copy, Link, Move, Scroll*** (like Explorer)
- ***Control:AllowDrop*** property to activate events on D&D
- ***DragEnter, DragOver, DragLeave, DragDrop*** events
- ***IDataObject*** interface & implementations

Control:DoDragDrop method

- Supported for each .NET Control
- ***data*** parameter:
  - .NET object instance within your application
  - String, Bitmap or MetaFile for external app
  - ISerializable or IDataObject for external app
- ***allowedEffects*** parameter: combination (one OR many Enum values of DragDropEffects)
  - Progresss.Util.EnumHelper:Or
- Need to be executed with a mouse button pushed

Drag & Drop events (recipient)

- ***DragEnter:*** D&D operation moves into a control. Controls can check ***“data”*** and accept/ignore by returning one of the allowed effects (sender)
- ***DragOver:*** D&D is moved within Control, if a control has multiple target areas, change cursor
- ***DragLeave:*** D&D is moved outside of Control, i.e. when BG color of Control changed during D&D
- ***DragDrop:*** User raises mouse button on a Control. Control should perform appropriate action with ***“data”***

IDataObject interface, DataObject class

- Wraps local object in D&D events
- Need to create an instance of DataObject for advanced context (more than a single format, data representation)
- Methods:
  - GetDataPresent (format AS character)
  - GetDataPresent (type AS System.Type)
  - GetFormats()
  - GetData()
  - System.Windows.Forms.DataFormat (Strings): Bitmap, ***FileDrop***, MetaFile, Rtf, Text, UnicodeText,

Drag & Drop traps

- **Don‘t miss to turn on AllowDrop on a receiptient Control!**
- Some Controls dislike it if you start Drag & Drop on mouse down, i.e. UltraGrid won’t raise DoubleClick event
- Alternatives
  - Use modifier key (CTRL, ALT, SHIFT)
  - Use right mouse button
  - Use Timer (i.e. .75 seconds) to start Drag & Drop
  - Use alternative section on UI (i.e. Row-Selector)
- sender receives no event about recipient's action (and if any at all)

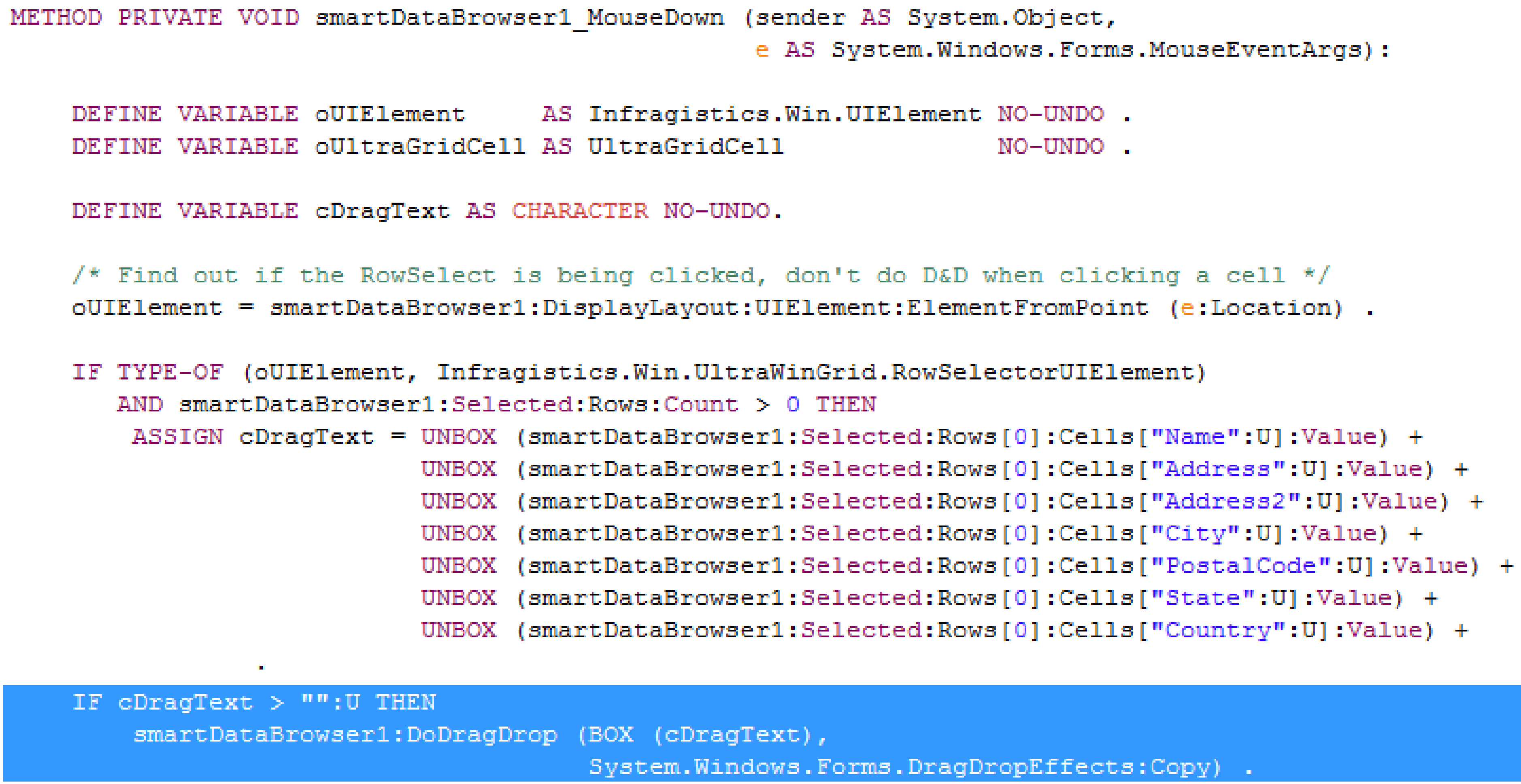
File Drag & Drop

- Two kind of file Drag & Drop operations
  - file name(s): String[]
  - file contents with suggested name: MemoryStream
- For file names, the Data is a .NET String Array of file names, the format is System.Windows.Forms.DataFormats:FileDrop
- For file contents, drop two format in the IData object:
  - „FileGroupDescriptor“ and
  - „FileContents“
- The sender needs to write the data to a memory stream at the beginning of the Drag & Drop operation
- Consultingwerk.Util.DragAndDropHelper class

Drag & Drop samples

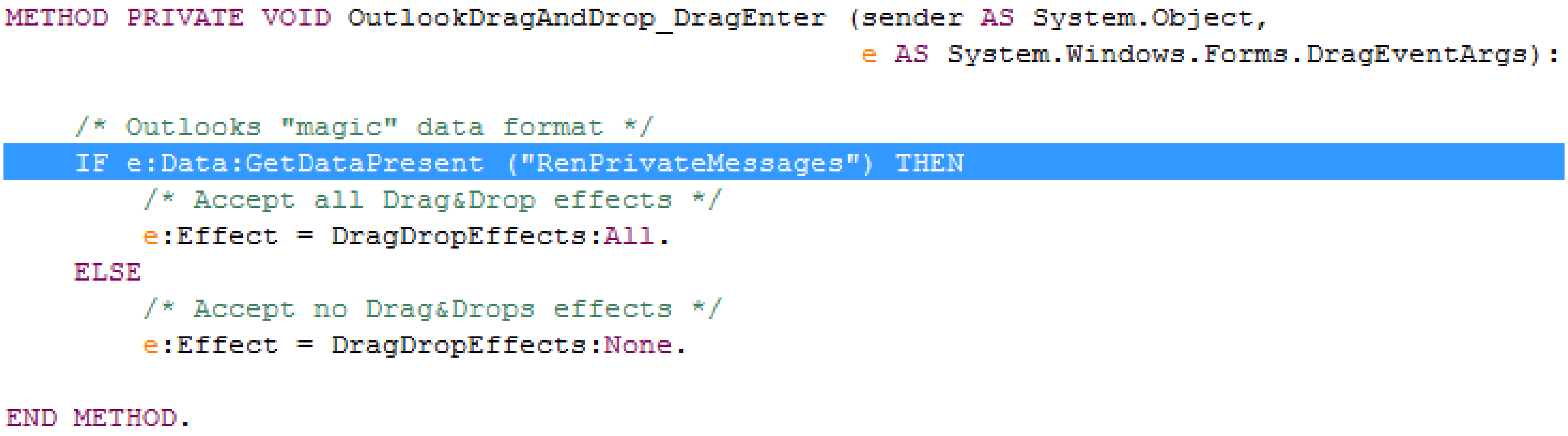
- Country, SalesRep object in CustomerExplorer
- System.String to Word, Outlook, …
- FileDrop in DragAndDropHelper (code review)
- Drop of File Contents (code review)
- Drag & Drop of Email, Contact, Appointment from Outlook

Drag & Drop System.String



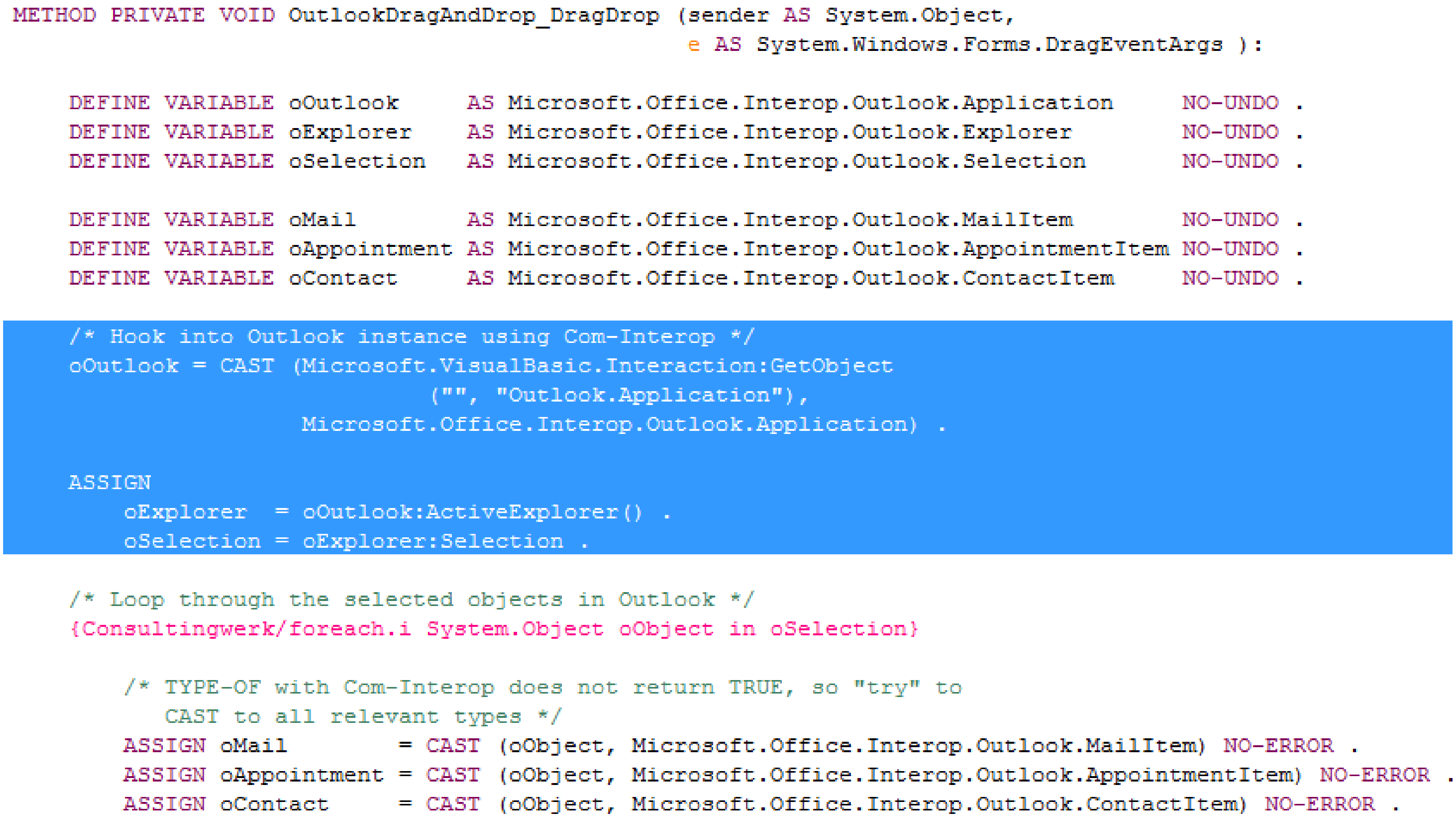
```
METHOD PRIVATE VOID smartDataBrowser1_MouseDown (sender AS System.Object,
                                                 e AS System.Windows.Forms.MouseEventArgs):

    DEFINE VARIABLE oUIElement     AS Infragistics.Win.UIElement NO-UNDO .
    DEFINE VARIABLE oUltraGridCell AS UltraGridCell              NO-UNDO .

    DEFINE VARIABLE cDragText AS CHARACTER NO-UNDO.

    /* Find out if the RowSelect is being clicked, don't do D&D when clicking a cell */
    oUIElement = smartDataBrowser1:DisplayLayout:UIElement:ElementFromPoint (e:Location) .

    IF TYPE-OF (oUIElement, Infragistics.Win.UltraWinGrid.RowSelectorUIElement)
       AND smartDataBrowser1:Selected:Rows:Count > 0 THEN
        ASSIGN cDragText = UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["Name":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["Address":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["Address2":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["City":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["PostalCode":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["State":U]:Value) +
                           UNBOX (smartDataBrowser1:Selected:Rows[0]:Cells["Country":U]:Value) +
                .
    IF cDragText > "":U THEN
        smartDataBrowser1:DoDragDrop (BOX (cDragText),
                                      System.Windows.Forms.DragDropEffects:Copy) .
```

Drag & Drop from Outlook



```
METHOD PRIVATE VOID OutlookDragAndDrop_DragEnter (sender AS System.Object,
                                                  e AS System.Windows.Forms.DragEventArgs):

    /* Outlooks "magic" data format */
    IF e:Data:GetDataPresent ("RenPrivateMessages") THEN
        /* Accept all Drag&Drop effects */
        e:Effect = DragDropEffects:All.
    ELSE
        /* Accept no Drag&Drops effects */
        e:Effect = DragDropEffects:None.

END METHOD.
```

Drag & Drop from Outlook



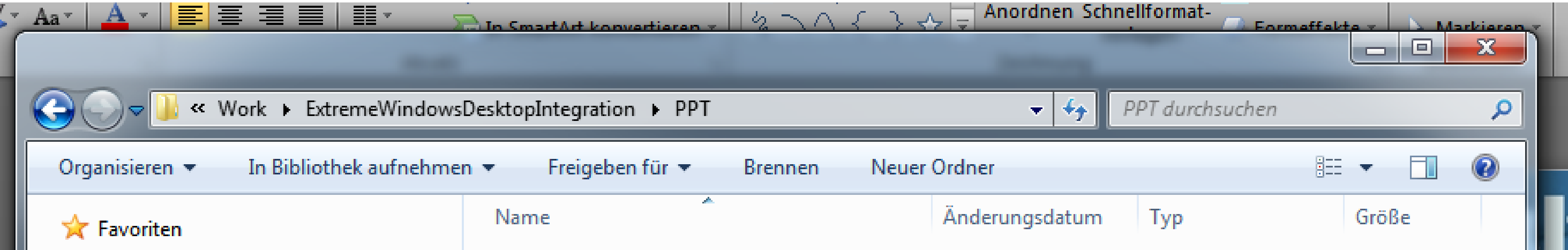
```
METHOD PRIVATE VOID OutlookDragAndDrop_DragDrop (sender AS System.Object,
                                                 e AS System.Windows.Forms.DragEventArgs ):

    DEFINE VARIABLE oOutlook     AS Microsoft.Office.Interop.Outlook.Application     NO-UNDO .
    DEFINE VARIABLE oExplorer    AS Microsoft.Office.Interop.Outlook.Explorer        NO-UNDO .
    DEFINE VARIABLE oSelection   AS Microsoft.Office.Interop.Outlook.Selection       NO-UNDO .

    DEFINE VARIABLE oMail        AS Microsoft.Office.Interop.Outlook.MailItem        NO-UNDO .
    DEFINE VARIABLE oAppointment AS Microsoft.Office.Interop.Outlook.AppointmentItem NO-UNDO .
    DEFINE VARIABLE oContact     AS Microsoft.Office.Interop.Outlook.ContactItem     NO-UNDO .

    /* Hook into Outlook instance using Com-Interop */
    oOutlook = CAST (Microsoft.VisualBasic.Interaction:GetObject
                              ("", "Outlook.Application"),
                     Microsoft.Office.Interop.Outlook.Application)  .

    ASSIGN
        oExplorer  = oOutlook:ActiveExplorer() .
        oSelection = oExplorer:Selection .

    /* Loop through the selected objects in Outlook */
    {Consultingwerk/foreach.i System.Object oObject in oSelection}

        /* TYPE-OF with Com-Interop does not return TRUE, so "try" to
           CAST to all relevant types */
        ASSIGN oMail        = CAST (oObject, Microsoft.Office.Interop.Outlook.MailItem) NO-ERROR .
        ASSIGN oAppointment = CAST (oObject, Microsoft.Office.Interop.Outlook.AppointmentItem) NO-ERROR .
        ASSIGN oContact     = CAST (oObject, Microsoft.Office.Interop.Outlook.ContactItem) NO-ERROR .
```

Agenda

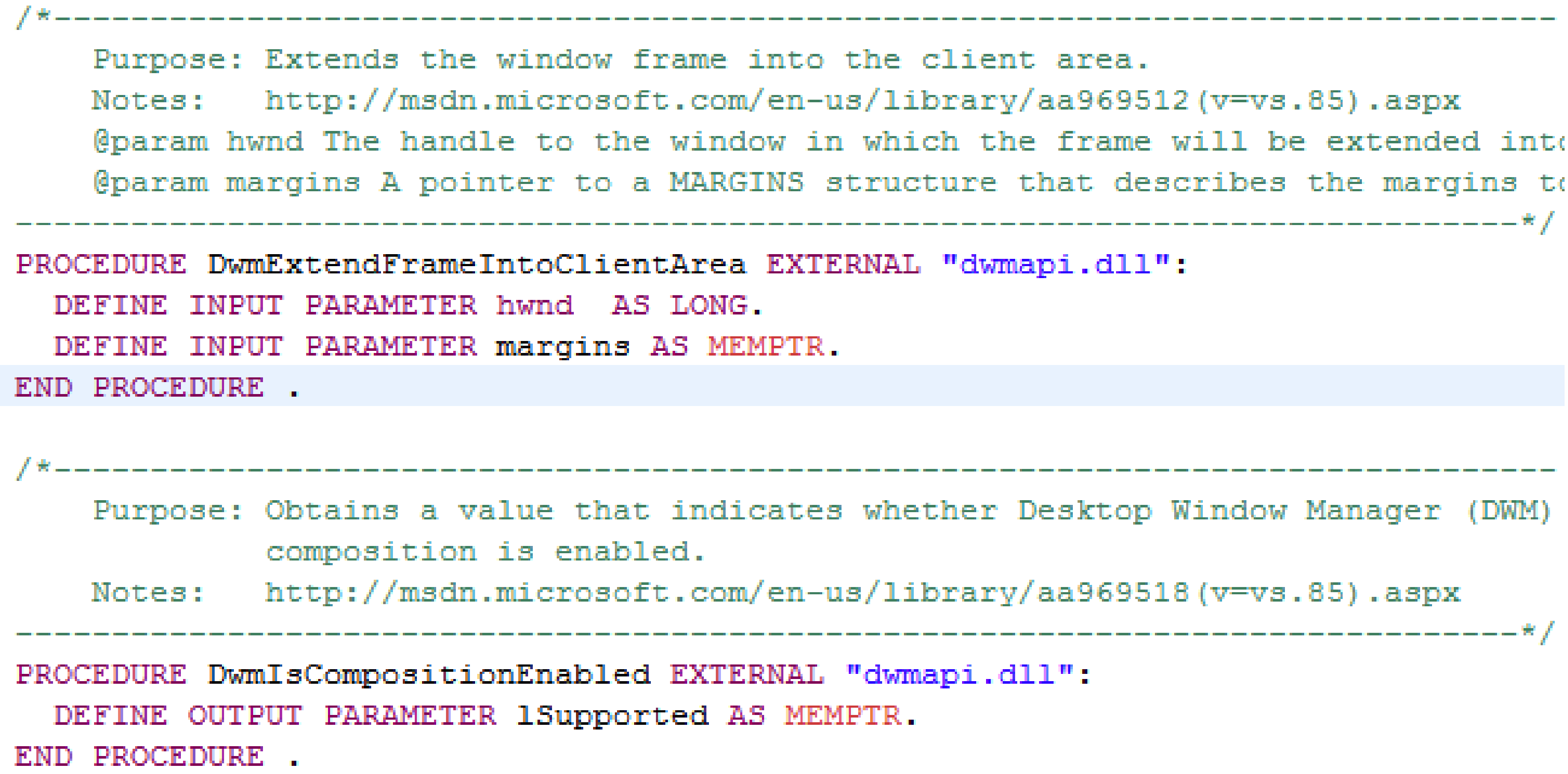
- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

Extended Transparency

- Windows 7 Explorer Windows extend the transparent region to more than the Window title
- „Fancy“ look for dialogs, message boxes, splash screens
- Not supported on pre Windows Vista
- Not supported on Terminal Server (maybe on the latest Windows 2008 R2 service pack)



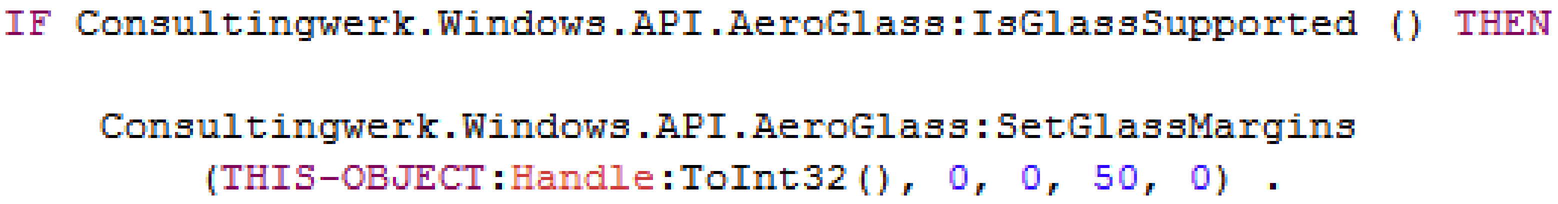
Extended Transparency



```
/*------------------------------------------------------------------------------
    Purpose: Extends the window frame into the client area.
    Notes:   http://msdn.microsoft.com/en-us/library/aa969512(v=vs.85).aspx
    @param hwnd The handle to the window in which the frame will be extended int
    @param margins A pointer to a MARGINS structure that describes the margins t
------------------------------------------------------------------------------*/
PROCEDURE DwmExtendFrameIntoClientArea EXTERNAL "dwmapi.dll":
  DEFINE INPUT PARAMETER hwnd  AS LONG.
  DEFINE INPUT PARAMETER margins AS MEMPTR.
END PROCEDURE .

/*------------------------------------------------------------------------------
    Purpose: Obtains a value that indicates whether Desktop Window Manager (DWM)
             composition is enabled.
    Notes:   http://msdn.microsoft.com/en-us/library/aa969518(v=vs.85).aspx
------------------------------------------------------------------------------*/
PROCEDURE DwmIsCompositionEnabled EXTERNAL "dwmapi.dll":
  DEFINE OUTPUT PARAMETER lSupported AS MEMPTR.
END PROCEDURE .
```

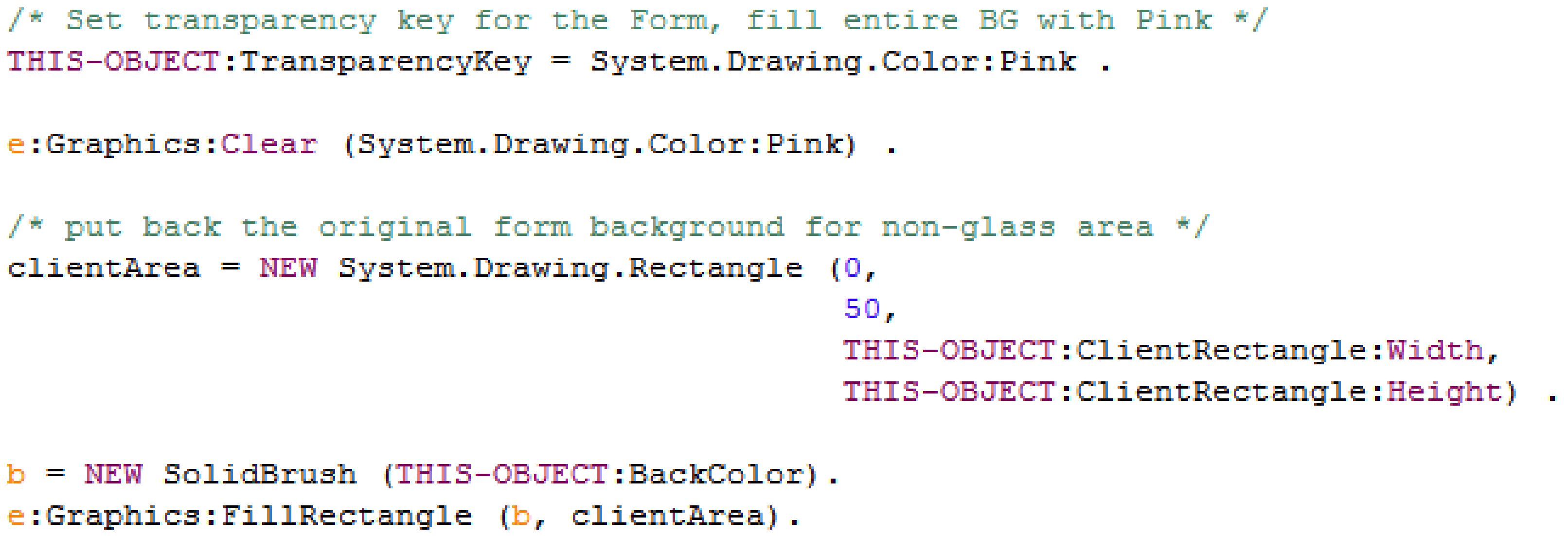
Extended Transparency

- API wrapper ***Consultingwerk.Windows.API.AeroGlass***
- During ***OnLoad*** (add 50 pixel from the top)



```
IF Consultingwerk.Windows.API.AeroGlass:IsGlassSupported () THEN

    Consultingwerk.Windows.API.AeroGlass:SetGlassMargins
        (THIS-OBJECT:Handle:ToInt32(), 0, 0, 50, 0) .
```

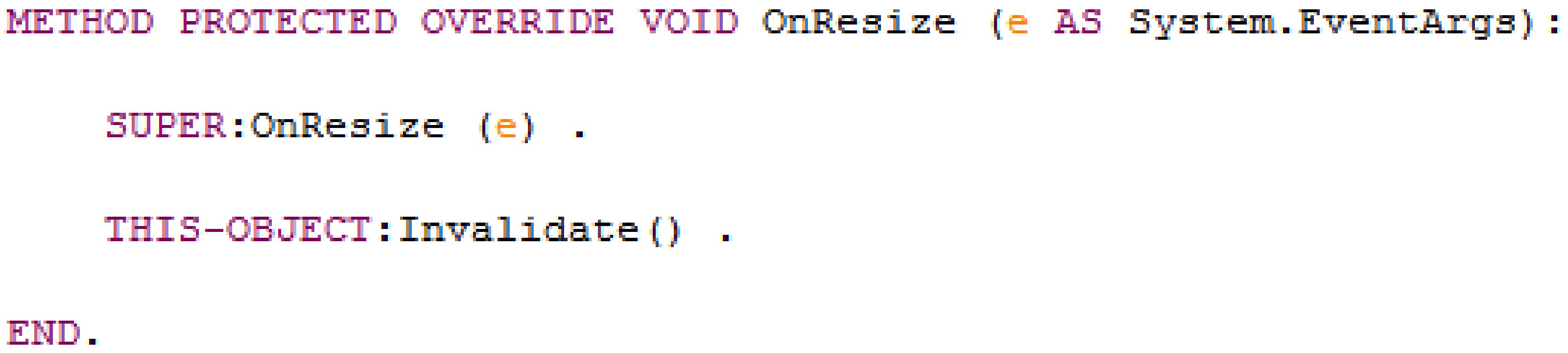
- In ***OnPaintBackground (e AS PaintEventArgs)*** override



```
/* Set transparency key for the Form, fill entire BG with Pink */
THIS-OBJECT:TransparencyKey = System.Drawing.Color:Pink .

e:Graphics:Clear (System.Drawing.Color:Pink) .

/* put back the original form background for non-glass area */
clientArea = NEW System.Drawing.Rectangle (0,
                                           50,
                                           THIS-OBJECT:ClientRectangle:Width,
                                           THIS-OBJECT:ClientRectangle:Height) .

b = NEW SolidBrush (THIS-OBJECT:BackColor).
e:Graphics:FillRectangle (b, clientArea).
```

Extended Transparency

- Invalidate Form (force repaint) during resize to avoid strange effects on screen



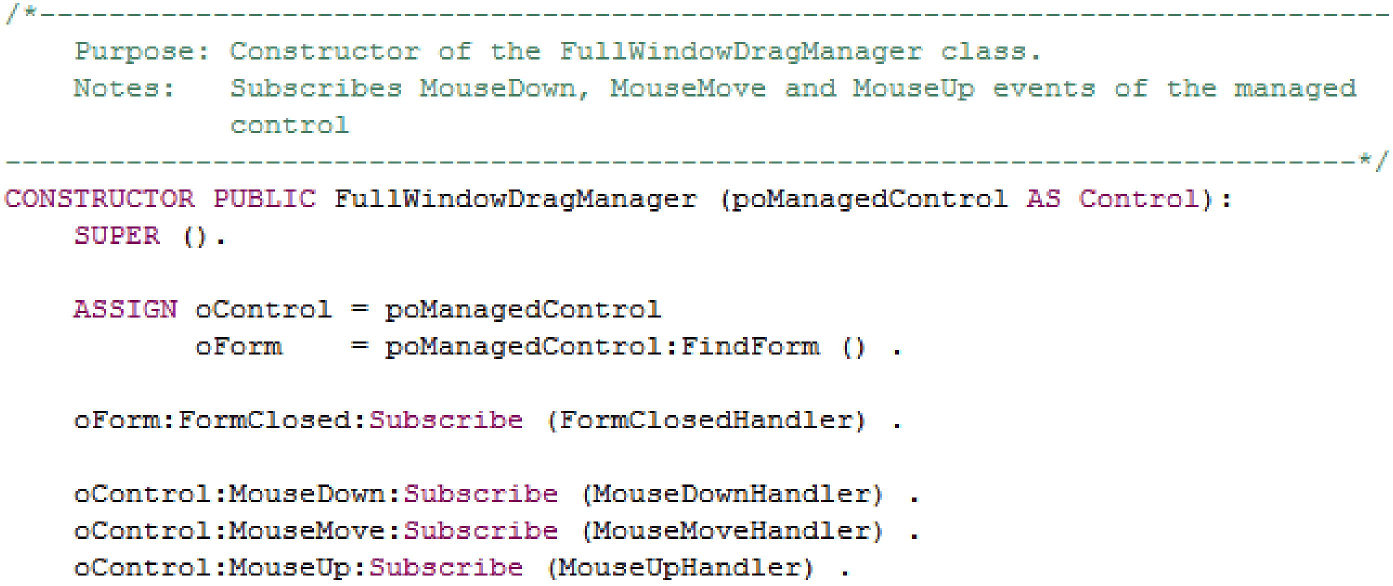
```
METHOD PROTECTED OVERRIDE VOID OnResize (e AS System.EventArgs):

    SUPER:OnResize (e) .

    THIS-OBJECT:Invalidate() .

END.
```

Full Window Drag

- Extended title does not allow distinction between window title and content
- User might get confused when trying to move a window on the screen
- Full Window Drag allows to extend the region where a Window can be moved around on the screen
- Requires handling Move Down, Up, Move events, translate Control coordinates to screen
- no standard Windows API, but common behavior in Windows Explorer, Ribbon, etc.

Full Window Drag

- Event handling of MouseMove etc. is wrapped in ***Consultingwerk.Windows.Util.Forms.FullWindowDragManager*** class
- Instantiate and pass Form or Control reference to constructor



```
/*------------------------------------------------------------------------------
    Purpose: Constructor of the FullWindowDragManager class.
    Notes:   Subscribes MouseDown, MouseMove and MouseUp events of the managed
             control
------------------------------------------------------------------------------*/
CONSTRUCTOR PUBLIC FullWindowDragManager (poManagedControl AS Control):
    SUPER ().

    ASSIGN oControl = poManagedControl
           oForm    = poManagedControl:FindForm () .

    oForm:FormClosed:Subscribe (FormClosedHandler) .

    oControl:MouseDown:Subscribe (MouseDownHandler) .
    oControl:MouseMove:Subscribe (MouseMoveHandler) .
    oControl:MouseUp:Subscribe (MouseUpHandler) .
```

Demo

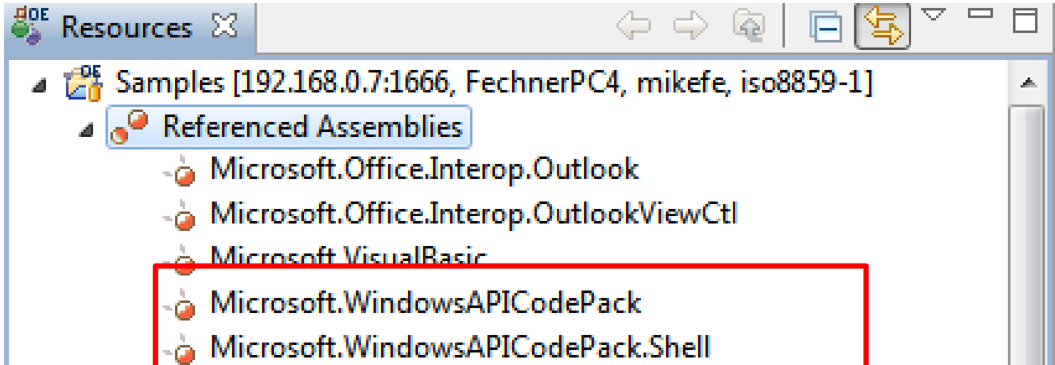
- Sample Form with extended transparency and full window drag

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

Windows 7 Taskbar Features

- Taskbar Icon Overlay
- Taskbar Progress Bar
- Jump Lists
- Windows® API Code Pack for Microsoft® .NET Framework: http://code.msdn.microsoft.com/WindowsAPICodePack
- .NET Assemblies, Current version 1.1

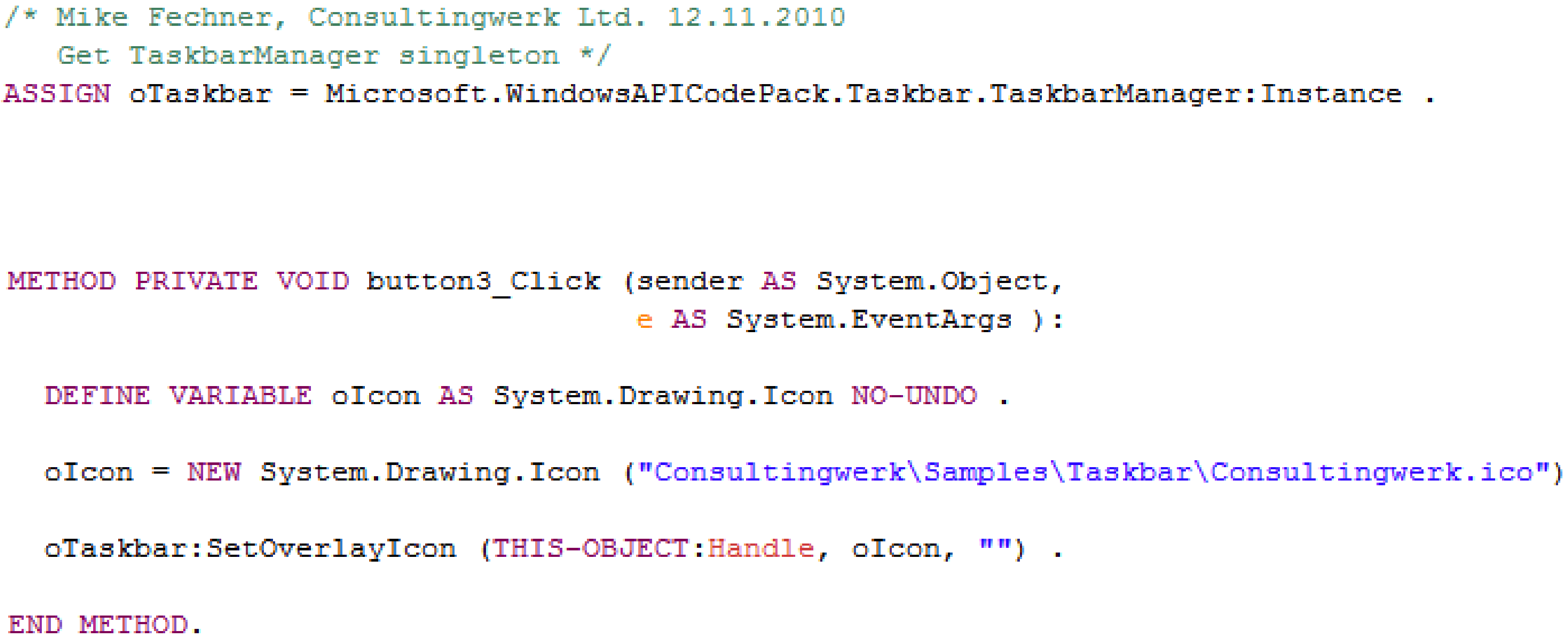


Samples [192.168.0.7:1666, FechnerPC4, mikefe, iso8859-1]
Referenced Assemblies
Microsoft.Office.Interop.Outlook
Microsoft.Office.Interop.OutlookViewCtl
Microsoft.VisualBasic
Microsoft.WindowsAPICodePack
Microsoft.WindowsAPICodePack.Shell

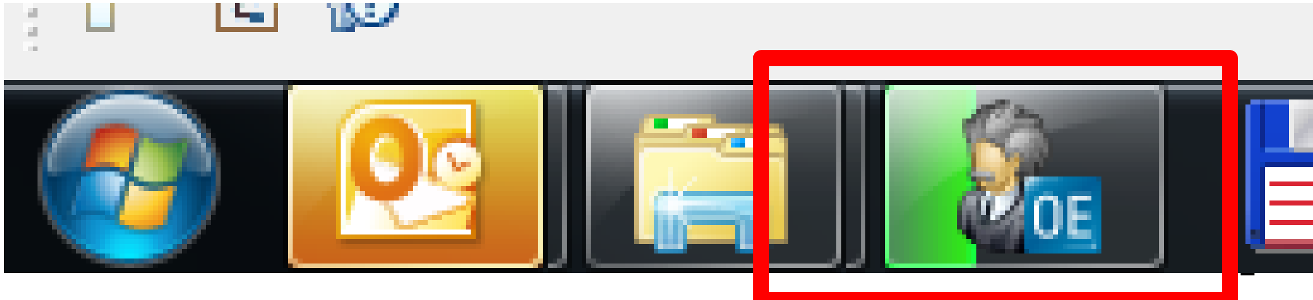
Windows 7 Taskbar Features

- Singleton class ***TaskbarManager***
- Overlay Icon
  - ***SetOverayIcon*** (Handle, Icon, Character)
- Progress Bar
  - ***SetProgressState*** (TaskbarProgressBarState, Handle)
  - ***SetProgressValue*** (Integer, Integer, Handle)
- ***TaskbarProgressBarState*** Enum
  - Error, Indeterminate, NoProgress, Normal, Paused
  - Indeterminate: Handled by Windows Explorer process, so movement also when prowin32.exe is busy

Taskbar Overlay Sample

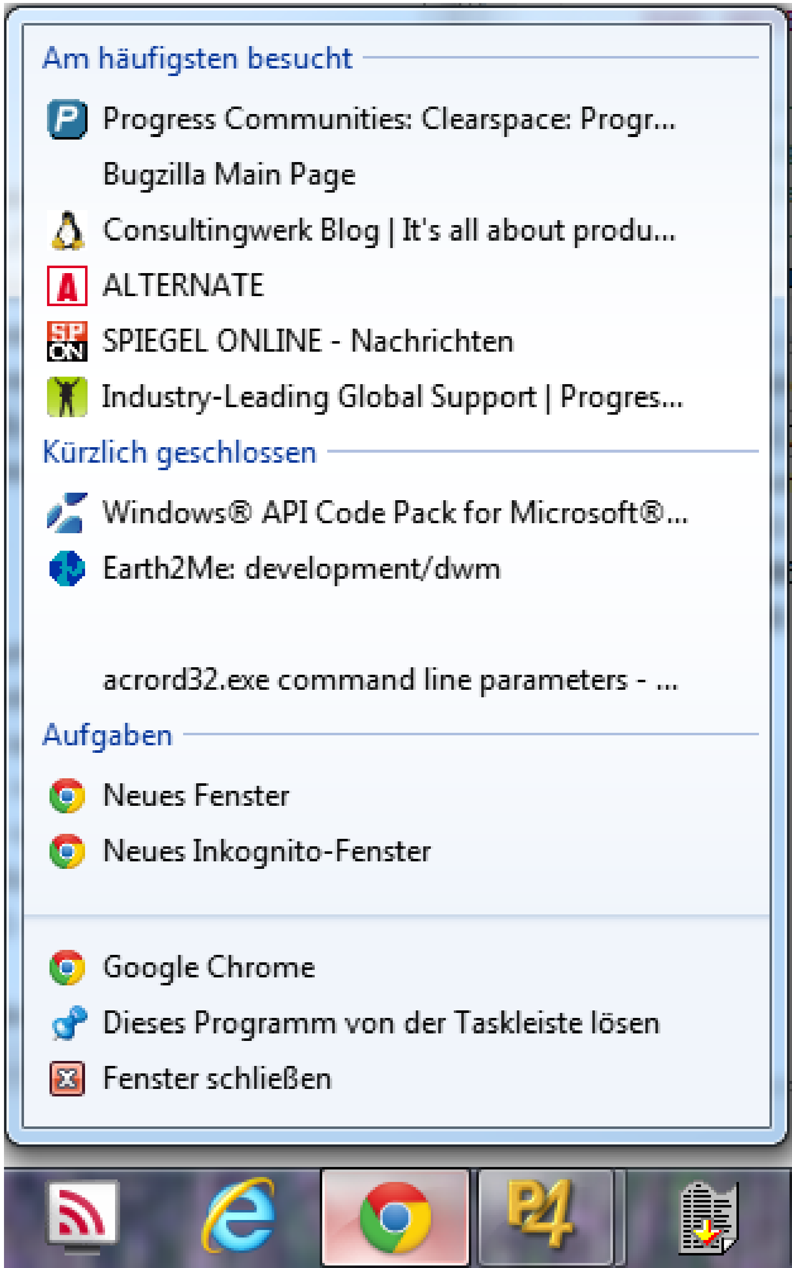


```
/* Mike Fechner, Consultingwerk Ltd. 12.11.2010
   Get TaskbarManager singleton */
ASSIGN oTaskbar = Microsoft.WindowsAPICodePack.Taskbar.TaskbarManager:Instance .



METHOD PRIVATE VOID button3_Click (sender AS System.Object,
                                   e AS System.EventArgs ):

  DEFINE VARIABLE oIcon AS System.Drawing.Icon NO-UNDO .

  oIcon = NEW System.Drawing.Icon ("Consultingwerk\Samples\Taskbar\Consultingwerk.ico")

  oTaskbar:SetOverlayIcon (THIS-OBJECT:Handle, oIcon, "") .

END METHOD.
```



Jumplists

- Customized context menu for the application
- Set by the application (first start is required)
- Will always launch new executable with defined parameters
- IPC required to talk into existing application
- Difficult (but certainly not impossible) with prowin32.exe (good JMS use case ☺)



Am häufigsten besucht

Progress Communities: Clearspace: Progr...

Bugzilla Main Page

Consultingwerk Blog | It's all about produ...

ALTERNATE

SPIEGEL ONLINE - Nachrichten

Industry-Leading Global Support | Progres...

Kürzlich geschlossen

Windows® API Code Pack for Microsoft®...

Earth2Me: development/dwm

acrord32.exe command line parameters - ...

Aufgaben

Neues Fenster

Neues Inkognito-Fenster

Google Chrome

Dieses Programm von der Taskleiste lösen

Fenster schließen

Demo

- Sample of using the Taskbar overlay feature

Agenda

- GUI for .NET Architecture and Windows API
- External Processes
- File System Operation
- Drag & Drop
- Extended Transparency, Full Window Drag
- Windows 7 Taskbar
- Notify Icons & Desktop Alerts

Notify Icons

- Icons in the System Tray (next to the clock)
- Can be used for non-modal, non-disturbing messages or notifications
- Supports Balloon Tooltip, Context Menu
- Visualize (background-)activity, i.e. synchronizing folder in Outlook
- Part of the Taskbar, managed by Windows Explorer / Desktop
- ***System.Windows.Forms.NotifyIcon*** Component
- Part of a Form Design or created via code

Notify Icons

- Properties
  - Icon (System.Drawing.Icon)
  - BalloonTipText, -Icon, -Title
  - ContextMenu (System.Windows.Forms.ContextMenu)
  - Visible

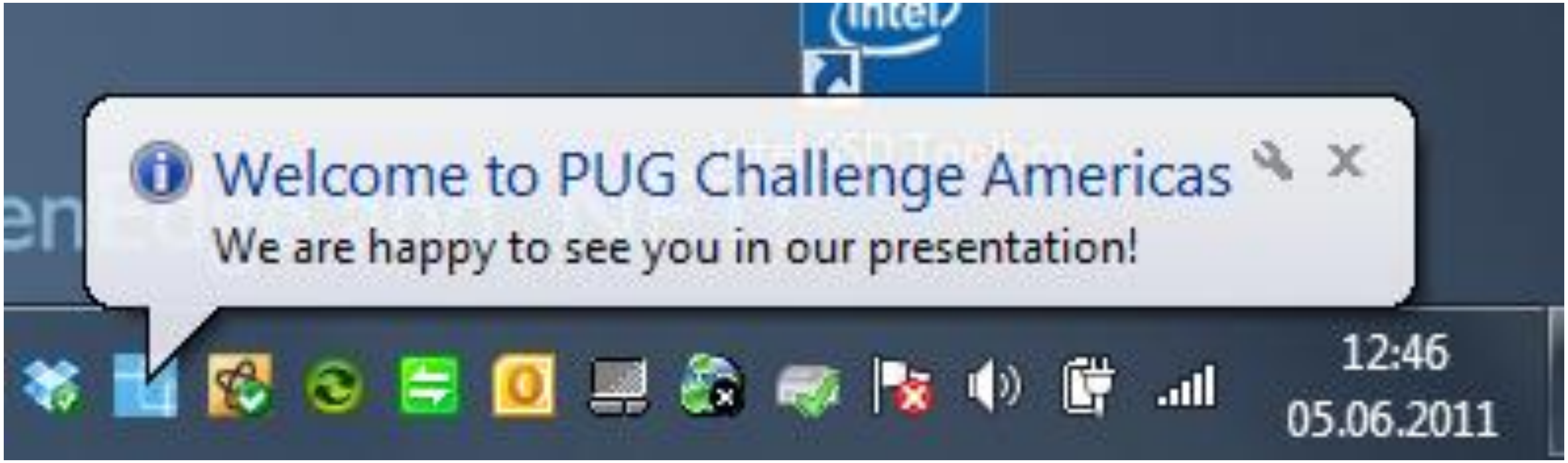
Notify Icons

- Methods
  - ShowBalloonTip()
  - Dispose()
- Events
  - Click, DoubleClick,
  - BallonTipClicked

Notify Icons

- Single Notify Icon instance can be used with alternating Icon’s (e.g. replace Icon to indicate application state)
- Notify Icons will need to be Dispose()‘d – otherwise they remain in the System tray even when the process has terminated (crashed or not)
- Effect known from other applications as well: Moving mouse over Icon will wipe it away

Notify Icon with Balloon Tip



Welcome to PUG Challenge Americas

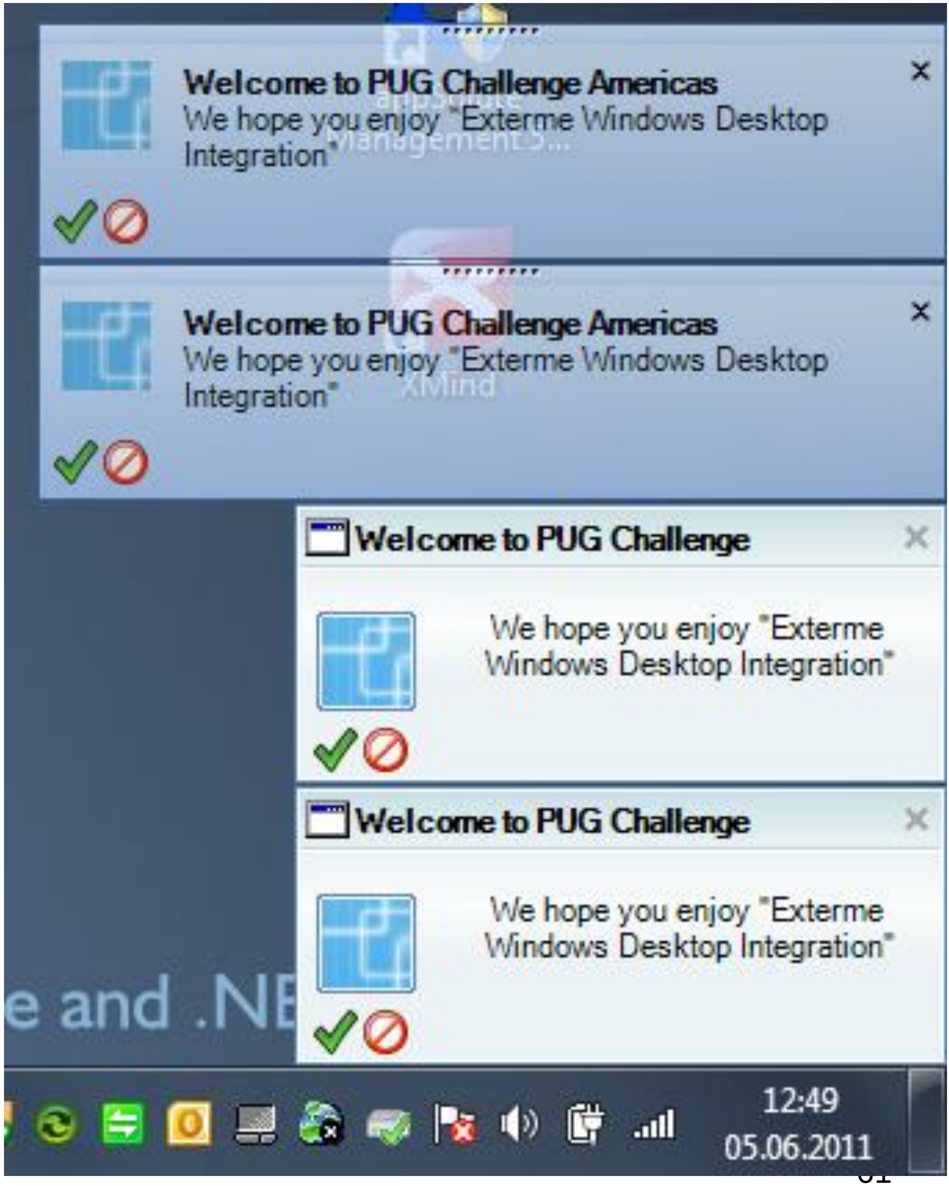
We are happy to see you in our presentation!

Desktop Alerts

- ***Similar to Outlook new Email notification***
- ***Infragistics.Win.Misc.UltraDesktopAlert***
- Can have buttons (special button class)
- Various user interactions
  - Clicking text, link
  - Clicking a button
  - Closing DesktopAlertWindow
- Useful for “Workflow” indications, new “order” cam in from a different system or user
- Sample requires OpenEdge UltraControls

UltraDesktopAlert

- DesktopAlerts‘s can be stacked



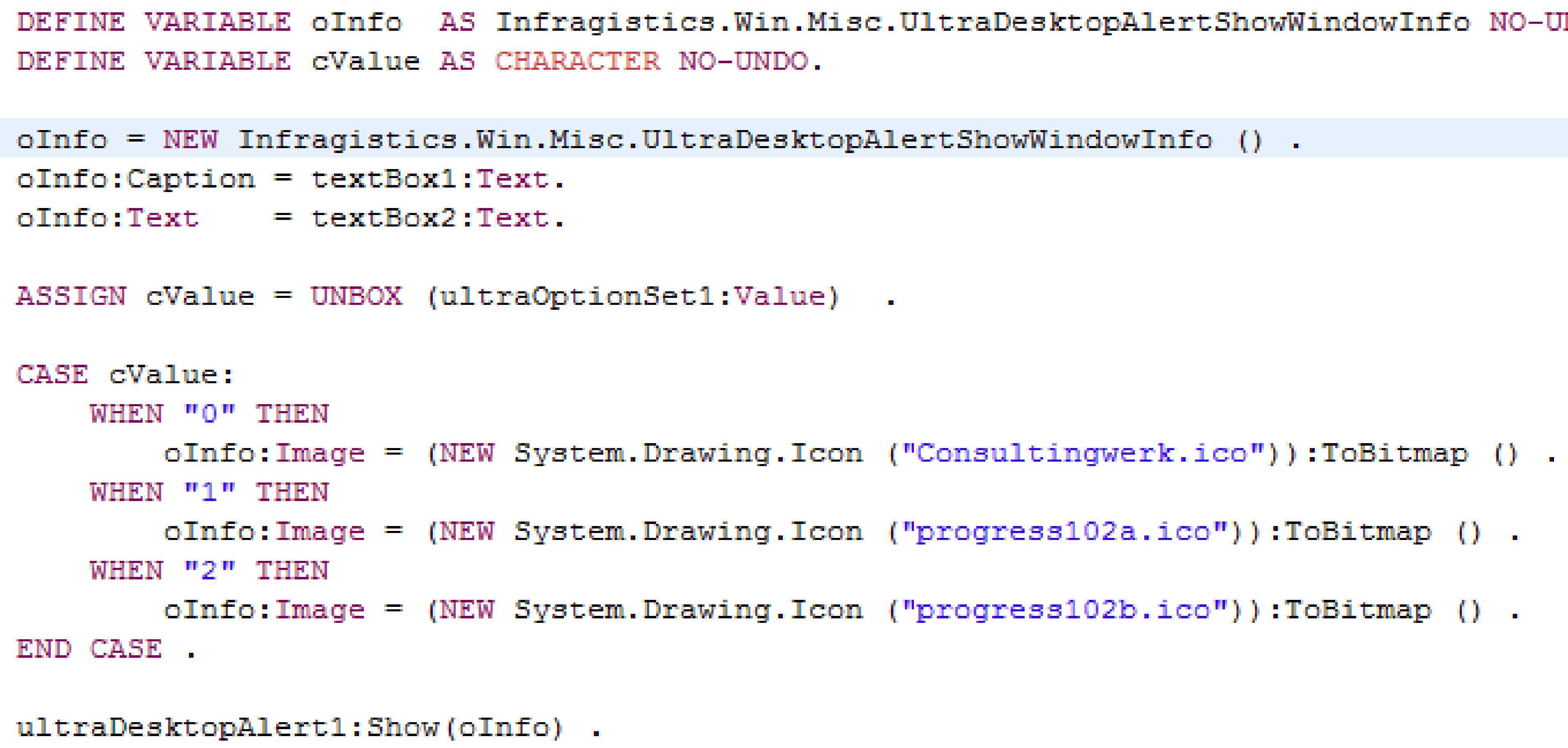
Welcome to PUG Challenge Americas
We hope you enjoy "Exterme Windows Desktop Integration"

Welcome to PUG Challenge Americas
We hope you enjoy "Exterme Windows Desktop Integration"

Welcome to PUG Challenge
We hope you enjoy "Exterme Windows Desktop Integration"

Welcome to PUG Challenge
We hope you enjoy "Exterme Windows Desktop Integration"

UltraDesktopAlert



```
DEFINE VARIABLE oInfo  AS Infragistics.Win.Misc.UltraDesktopAlertShowWindowInfo NO-U
DEFINE VARIABLE cValue AS CHARACTER NO-UNDO.

oInfo = NEW Infragistics.Win.Misc.UltraDesktopAlertShowWindowInfo () .
oInfo:Caption = textBox1:Text.
oInfo:Text    = textBox2:Text.

ASSIGN cValue = UNBOX (ultraOptionSet1:Value)  .

CASE cValue:
    WHEN "0" THEN
        oInfo:Image = (NEW System.Drawing.Icon ("Consultingwerk.ico")):ToBitmap () .
    WHEN "1" THEN
        oInfo:Image = (NEW System.Drawing.Icon ("progress102a.ico")):ToBitmap () .
    WHEN "2" THEN
        oInfo:Image = (NEW System.Drawing.Icon ("progress102b.ico")):ToBitmap () .
END CASE .

ultraDesktopAlert1:Show(oInfo) .
```

UltraDesktopAlert Properties

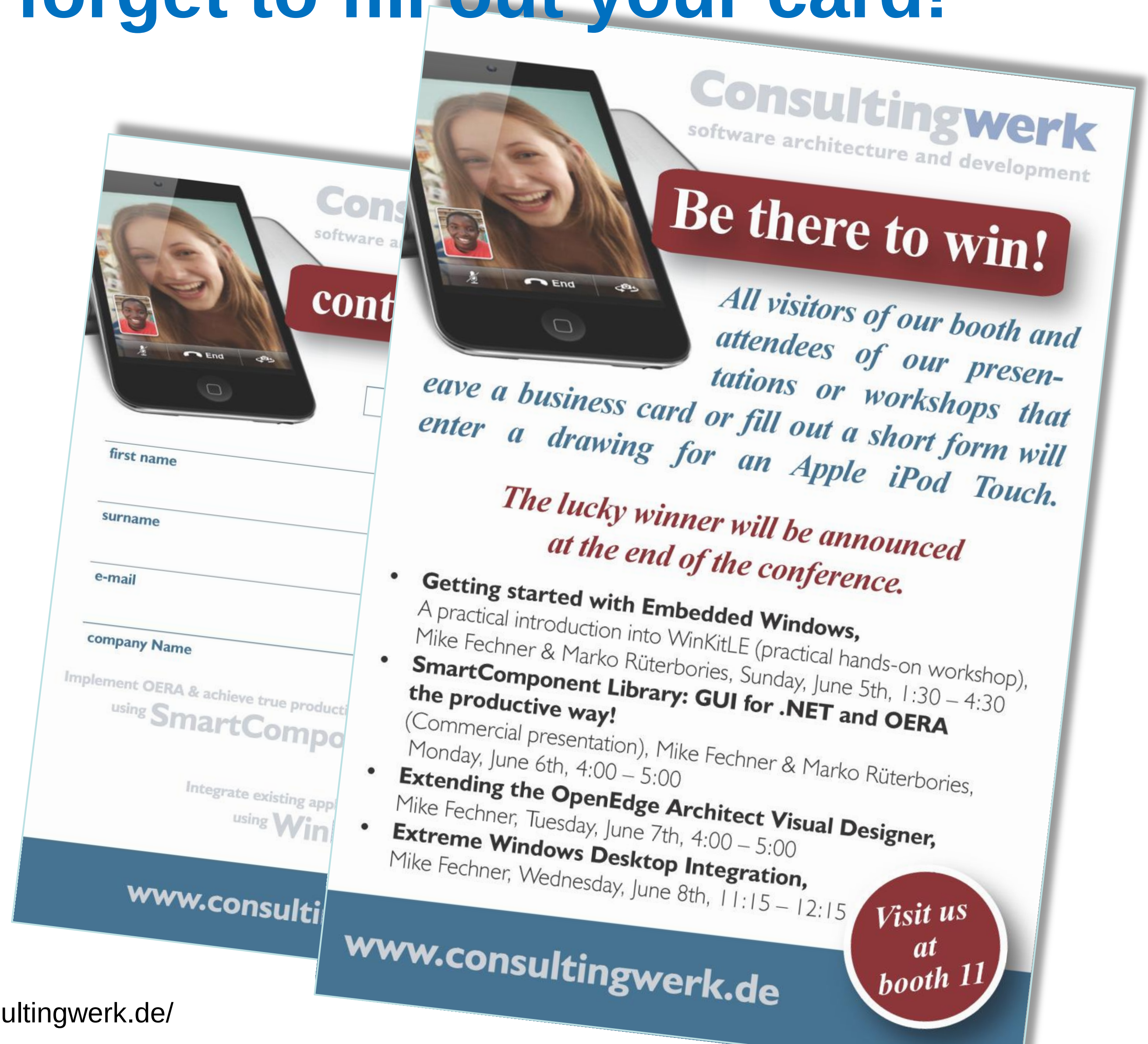
- ***AnimationSpeed:*** Slow, Medium, Fast
- ***AnimationStyleAutoClose:*** None, Fade, FadeAndScroll, Scroll
- ***AnimationStyleShow:*** None, Fade, FadeAndScroll, Scroll
- ***Style:*** Office2007, WindowsLiveMessenger

Questions

Access to sample code download

- http://blog.consultingwerk.de

Don‘t forget to fill out your card!



Cons
software a
cont

first name

surname

e-mail

company Name

Implement OERA & achieve true producti
using SmartCompo

Integrate existing app
using Win

www.consulti

Consultingwerk
software architecture and development

Be there to win!

All visitors of our booth and attendees of our presentations or workshops that eave a business card or fill out a short form will enter a drawing for an Apple iPod Touch.

The lucky winner will be announced at the end of the conference.

- **Getting started with Embedded Windows,** A practical introduction into WinKitLE (practical hands-on workshop), Mike Fechner & Marko Rüterbories, Sunday, June 5th, 1:30 – 4:30
- **SmartComponent Library: GUI for .NET and OERA the productive way!** (Commercial presentation), Mike Fechner & Marko Rüterbories, Monday, June 6th, 4:00 – 5:00
- **Extending the OpenEdge Architect Visual Designer,** Mike Fechner, Tuesday, June 7th, 4:00 – 5:00
- **Extreme Windows Desktop Integration,** Mike Fechner, Wednesday, June 8th, 11:15 – 12:15

Visit us at booth 11

www.consultingwerk.de