



# ArcPad Customization: Deep Dive

*Craig Greenwald (ESRI)*

*Stephen Quan (MapTel)*

# Session Evaluations Reminder

**Session Attendees:**

**Please turn in your session evaluations.**

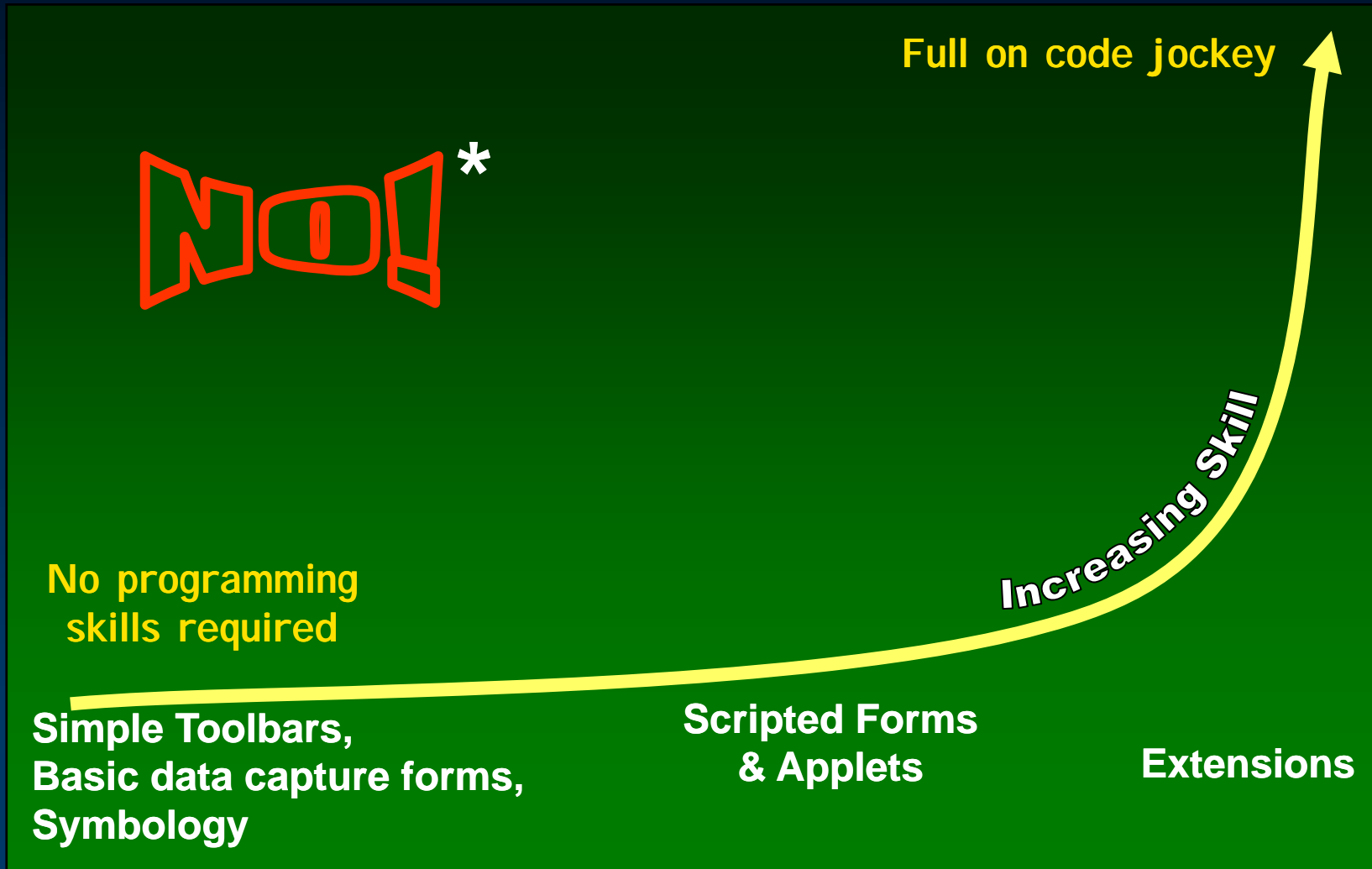
***... Thank you***

# Our assumptions

- **You are already customizing ArcPad 6.x or 7.x or are familiar with XML and the various ArcPad customization file types**
- **You know the difference between ArcPad and ArcPad Application Builder**
- **You are comfortable with basic forms and toolbar customizations and are ready to move to the next level**
- **You are comfortable programming in a scripting environment (VBScript / JScript)**
- **Some of you are C / C++ programmers or .NET (C# / VB.NET) programmers**

# Do I need to be a programmer?

\* But sometimes it  
Helps ☺



# Agenda

- **Customization Quick Overview**
- **Extensions Overview**
- **Deploying Projects**
- **Attribute Validation**
- **Related Tables**
- **Modifying Preferences**
- **Serial Device Communication**
- **Internet Communication**
- **External application communication**
- **Tips & Tricks / Best Practices**

# ArcPad 7.x Release History

- **7.0 released November 2005**
- **7.0 SP1 released March 2006**
- **7.0.1 released October 2006**
  - Cover mainly features of 7.0.1 today
- **7.1 Beta 1 March 2007 ([beta.esri.com](http://beta.esri.com))**
  - Planned release ... mid 2007
  - Related Table Support (ArcPad and Desktop Tools enhancements)
  - UI Enhancements
  - Will highlight 7.1 in this talk

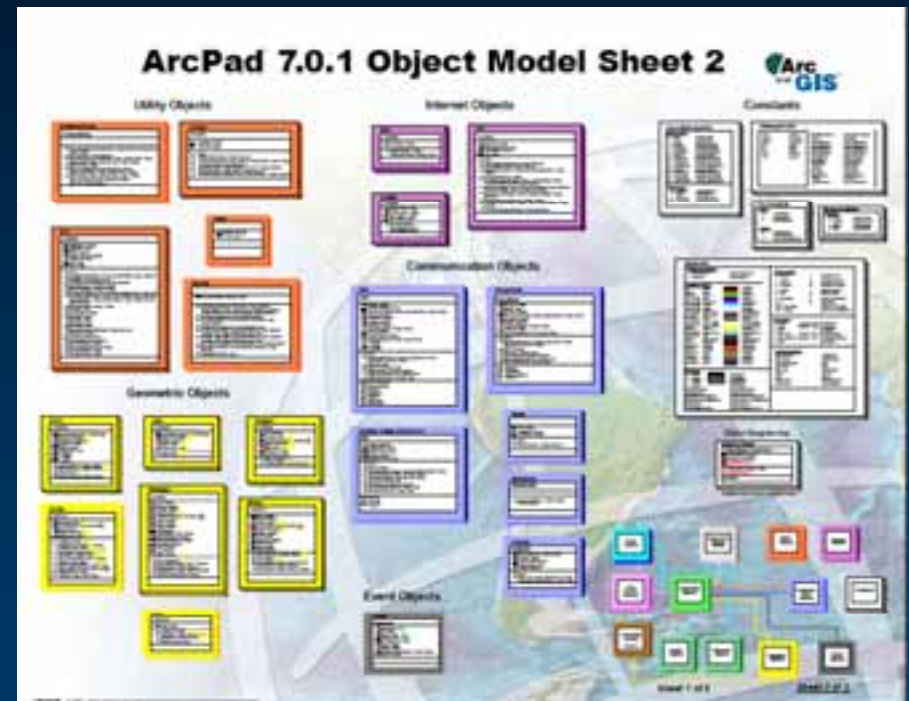
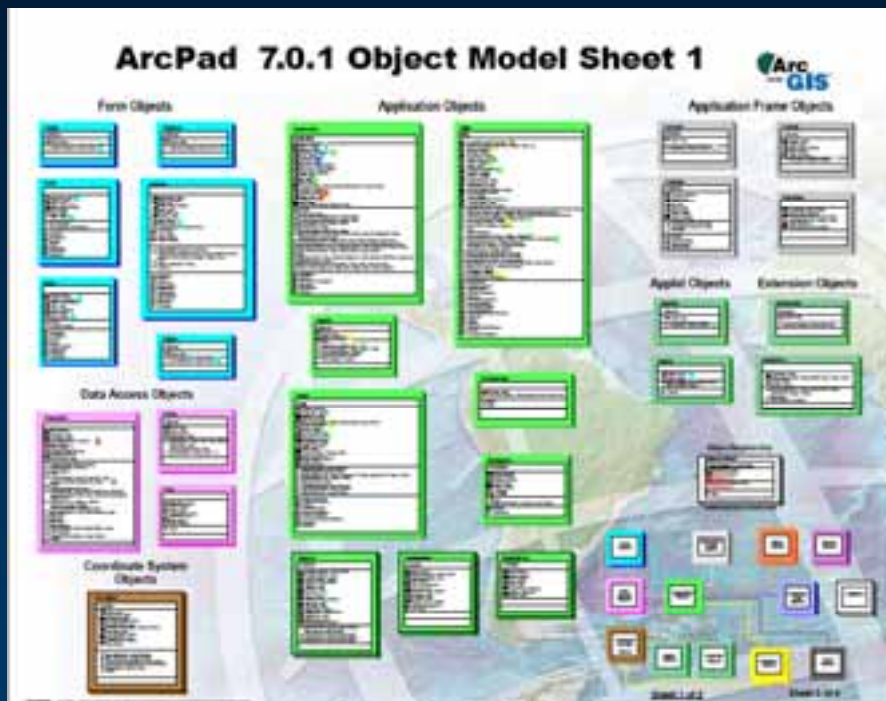
# What can I customize?

- **User interface (toolbars, forms)**
- **Create additional tools/functionality**
- **Enforce data integrity**
- **Create templates for data consistency**
- **Combine common tasks**
- **New data formats, protocols, projections, datum transformations, and devices**

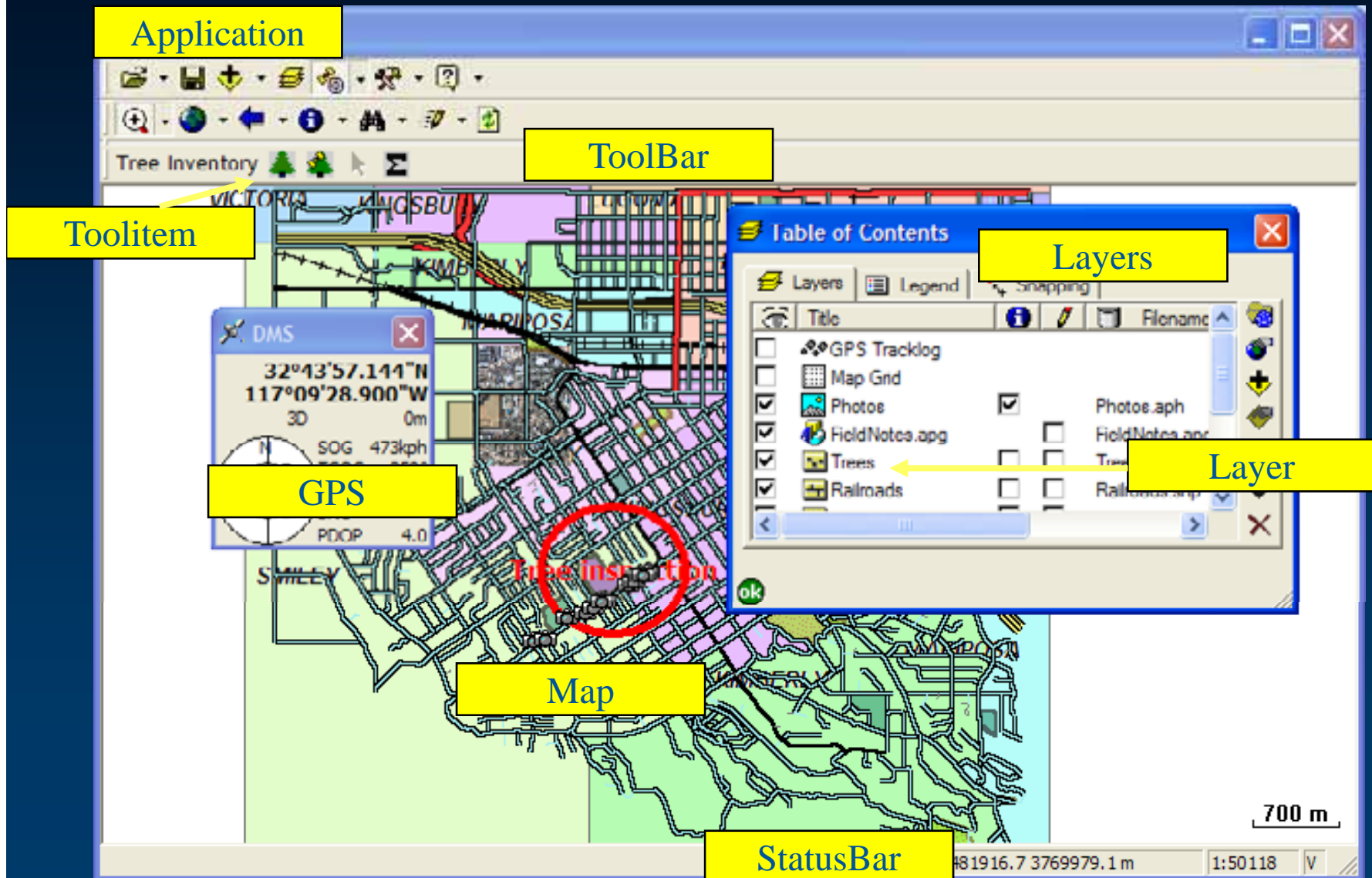
# Customization Options

- **Choosing between Script and Extension based customization**
  - Some solution deployment may require both scripting and extensions
- **Script Based**
  - Easy to learn and deploy
  - Interpreted
  - ArcPad Studio is the primary environment used for development
  - Used for :-
    - Custom toolbars and buttons
    - Data capture forms and data entry validation
    - Aggregating commands
  - Stored in ArcPad XML files and VBScript / JScript files
- **Extension Based**
  - Compiled C/C++ DLL
  - Harder to write
  - Platform support (Multiple CPU types)
  - Fast
  - Used for :-
    - Supporting additional file formats as layer types (vector and raster)
    - Supporting new instruments (GPS, rangefinders & cameras)
    - Extending the projection engine
    - Accessing low level APIs

# ArcPad Object Model Detail



# Common ArcPad Objects in the UI



# New Objects in ArcPad 7.x

- **FTP**
  - Open, close ftp connection – With password if required
  - Get files from server and put files onto server
- **Archive**
  - Compress files into .ZIP
  - Extract files from .ZIP
  - Password protected
- **Rangefinder**
  - Respond to rangefinder events
  - Access measurement data
- **Image**
  - Read/Write EXIF data in JPEG files
- **Multimedia**
  - Capture photos
- **Console**
  - Output window useful for application and debugging messages
- **[7.1] DataConnection**
  - Access to ArcPad AxF files and ability to execute SQL statements



## Routing Applet Demo

*How far can scripting take you?*

# Routing Applet Sample

- **Example of what can be done**
- **Written entirely in VBScript code using on ArcPad's object model**
- **Preparing your data for routing**
  - Polyline shapefile must have lines split at vertices (ie topology)
  - Easy way to do this in using 9.2's "Split Line At Vertices" tool
- **Running the applet**
- **How it works**
- **Available on ArcScripts (search for 'quickroute')**

# Extension Types

- **Layer**

- Support new raster or vector file formats not supported by ArcPad
- Formats can be implemented as read-only or full read-write with geometry and attribute editing
- Typical layer extension contains functions to open, close and draw layer data and retrieve or set layer, geometry or attribute data a values

- **Camera**

- Provides an interface to camera hardware to take a still photo
- Extension provides user interface specific to the requirements of the camera
- Photo is saved in JPEG format to specified filename
- ArcPad provides utility functions to simplify writing:
  - Read & write JPEG files
  - Read & write EXIF information (eg Camera parameters)



# Extensions

*A little C++ goes a long way...*

# Extension Types (continued)

- **GPS**

- Support GPS data formats not recognized by ArcPad
- Can be binary or ASCII
- Minimal requirement is to return latitude/longitude position
- Many other data values are supported. Eg Satellite positions, signal strengths, speed, heading etc.



- **Rangefinder**

- Support rangefinder data formats not recognized by ArcPad
- Two typical data formats
  - **NMEA style sentences**
    - \$XXX,value,value,value\*checksum
    - ArcPad preprocess NMEA sentences so all that is required is interpreting field values
  - **Proprietary text format**
- Minimum requirement is to measure distance
- Can also return bearing and inclination measurements if available

# Extension Types (continued)

- **Projection**

- Support additional projection algorithms eg. “Flat\_Polar\_Quartic”
- Integrates automatically with ArcPad’s projection engine and .PRJ file parser
- Two main functions required:
  - **Forward Project**
    - From latitude/longitude to projected coordinates
  - **Inverse Project**
    - From projected coordinates back to latitude/longitude

- **Datum Transformation**

- Support additional datum transformation methods eg. “NTv2”
- Integrates via Transforms.dbf configuration file
- Two main functions required to transform between to datums
  - **Forward Transform from Datum 1 → Datum 2**
  - **Inverse Transform from Datum 2 → Datum 1**



# Extension Types (continued)

- **Utility**

- Provide 'helper' functions
- Use whenever scripting is not enough
- Access via Extension object in scripts
  - **Escape method**
  - **ExProperties property**
- Can install tools on the tool bar

- **Notification**

- Used for GPS post processing
- Receives information whenever edits occur in ArcPad

# Back to the Future



- Demo
  - Venture into the past and the future!

# Back to the Future (The Applet)



```
Sub OnForward
  Dim E, T
  Set E = Extensions("TIME")
  T = E.Escape("gettime")
  T = T + 10950

  Call E.Escape("settime", T)
  Applet.Forms("frmMain").Pages("pagMain").Controls("lblTime").Value = Now
End Sub

Sub OnBack
  Dim E, T
  Set E = Extensions("TIME")
  T = E.Escape("gettime")
  T = T - 10950

  Call E.Escape("settime", T)
  Applet.Forms("frmMain").Pages("pagMain").Controls("lblTime").Value = Now
End Sub
```

# Back to the Future (C++ code)



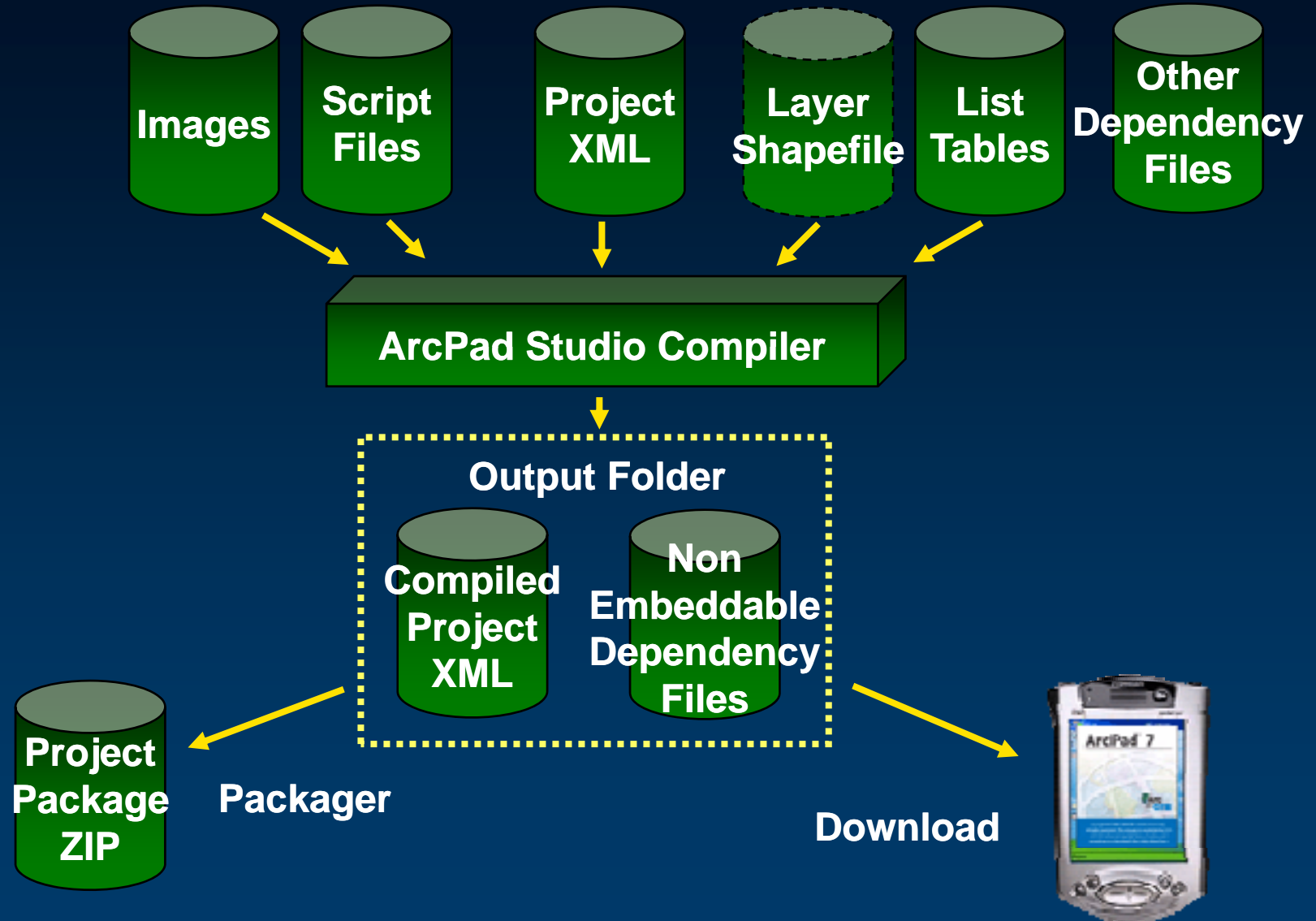
```
extern "C" __declspec(dllexport) HRESULT APEscape(HANDLE hObject, VARIANT
 *pParam1, VARIANT *pParam2, VARIANT *pParam3, VARIANT *pResult)
{
    // ...
    if (_tcsicmp(szParam1, TEXT("gettime")) == 0)
    {
        GetSystemTime(&SystemTime);
        VariantClear(pResult);
        pResult->vt = VT_DATE;
        SystemTimeToVariantTime(&SystemTime, &pResult->date);
    }
    else if (_tcsicmp(szParam1, TEXT("settime")) == 0)
    {
        VariantChangeType(&VariantTime, pParam2, 0, VT_DATE);
        VariantTimeToSystemTime(VariantTime.date, &SystemTime);
        SetSystemTime(&SystemTime);
    }
    // ...
    return S_OK;
}
```



# Deploying projects with ArcPad Studio

## *Compiling*

# ArcPad Studio Compile File Flow



# Compile Demo

*Deploying your customizations*

# ArcPad Studio Compile Tool

- Check script syntax for `<SCRIPT>` elements and `on...()` event handlers
- Check for duplicate form, page and control names
- Embed images
- Embed scripts
- Non embeddable images and tables are added as dependencies
- Process `<DEPENDENCY>` elements
- Encode
- Copy to output folder
- Create a package (ZIP file)
- Download to mobile device



# Attribute Validation

# Several ways to ensure valid attributes are entered

- **Control Properties in Forms**
  - Min / max values, combo box / list box with limit to list
  - Read only, required, default
- **Scripts**
  - OnValidate, OnLoad event handlers in forms
  - OnFeatureAdded, OnSelectionChanges, etc events
  - Only approach for complex rules
  - Only approach for automatic attributes
- **Database level management**
  - Introduced at 7.1 with AxF files
  - Subtypes, coded value domains, range domains
  - Read only, required, default

# Introducing AxF Files at ArcPad 7.1

- **A true relational database built on MS SQL Server Compact Edition**
- **Database level rule management instead of application level (forms & scripts)**
  - **Reliable referential constraints (foreign keys)**
- **Attribute rules presented automatically in Feature Properties dialog**
- **Object model enhanced to support AxF files**
  - **RecordSet object**
  - **New DataConnection object**
  - **Execute any SQL statements**
- **Generated from geodatabase by ArcPad Data Management extension for ArcGIS 9.2**
  - **Replaces existing ArcPad Tools in ArcMap**
  - **Includes desktop and geoprocessing tools**

# Key Features of ArcPad AxF Files

- Database level management of relationships
- Built in “on the fly” index management
- Automatic data entry fields based on database relationships
- “All in one file”
  - More robust of handling reboot, power down
- Password protected (optional)
- Encrypted (optional)
- Each feature table can have multiple symbology representations and input forms



# Attribute Validation Demo

*Form definition rules vs. Scripting*

# Complex validation script example

```
Dim dtEnteredDate
dtEnteredDate = ThisEvent.Object.Value

If (DateDiff("d",Now,dtEnteredDate) < -7) Then
    ThisEvent.Result = False
    ThisEvent.MessageText = "The date entered is too old."
    ThisEvent.MessageType = apCritical
ElseIf (dtEnteredDate > Now) Then
    ThisEvent.Result = False
    ThisEvent.MessageText = "The date entered is in the
future."
    ThisEvent.MessageType = apCritical
End If
```



# Related Tables

# Problem: Complex Attribute Structure

- Multiple attachments or inspections on a single structure
  - Sign post with multiple signs
  - Utility pole with multiple attachments
  - Sidewalk with annual inspections
- Single geographic feature with multiple associated records
  - No need to store geography multiple times
- Generally implemented as a feature class with one or more related tables



PoleID	Type	Built
123	Wood	8-Oct-99
124	Steel	5-Jul-87
125	Wood	7-Jun-99

PoleID	Insulator #	Type
123	1	200kv
123	2	400kv
123	3	200kv



PoleID	Insulator #	Inspector	Date
123	1	Jian	1-Jan-00
123	1	Elvin	5-Feb-04

# Related Table Support pre 7.1

- Rudimentary display-only related table supported provided by SUBTABLE control
- General Forms can be programmatically linked to external databases to create Edit Form behavior
- Use ArcPad's Data Access Objects with DBF tables
  - RecordSet, Fields, Field
- Use Microsoft's ADOCE for databases other than DBF
  - Creatable ActiveX objects
  - Similar to ArcPad's Data Access Objects
  - Needs to be installed on most mobile devices (free install)
  - \*\* No longer supported at CE 5 / WM 5
- <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/adoce31/html/adowlcm.asp>

# Populating the related table form: Programming Pattern

- **Form OnLoad event**
  - Open the table
  - Get the desired record
  - Write the record's field values or default values to form controls (edit, combobox, datetime, etc.)
- **Form OnOK event**
  - Write form control values to the record or create a new record
  - Close the table
- **OnValidate events**
  - Implement standard OnValidate event handlers at the page and/or control level to enforce data integrity rules

# Related Table Support in 7.1

- **New ArcPad AxF File Format**
- **Relationships managed at the database level**
  - Feature Properties dialog automatically presents related tables as additional tabs
- **Related tables exposed to Object Model**
  - RecordSet, Fields, Field, DataConnection

**Let's look at an example of editing related tables in ArcPad 7.1**



# Modifying Preferences

# Modifying ArcPad Preferences

- **Preferences object limited, but adequate to get the job done**
  - Path to ArcPadPrefs.apx and other key paths
  - Ability to re-load preferences from ArcPadPrefs.apx
- **ArcPadPrefs.apx structure well documented**
- **XMLDOM ActiveX object available on most systems (Microsoft.XMLDOM)**
- **A few lines of script do the trick**

# Set GPS averaging script example

```
function SetAveraging {
    var ARCPAD_APX_FILE = System.Properties("PersonalFolder") +
        "\\ArcPadPrefs.apx";

    Preferences.Write();

    var pArcPadXML = new ActiveXObject("Microsoft.XMLDOM");
    pArcPadXML.load(ARCPAD_APX_FILE);

    var pAVGElement = pArcPadXML.documentElement.
        selectSingleNode("PREFERENCES/GPS/AVERAGING");
    pAVGElement.setAttribute("enabled", "true");
    pAVGElement.setAttribute("point", "30");
    pAVGElement.setAttribute("vertex", "5");
    pArcPadXML.save(ARCPAD_APX_FILE);

    Preferences.Read();

    pArcPadXML = null;
    pAVGElement = null;
}
```



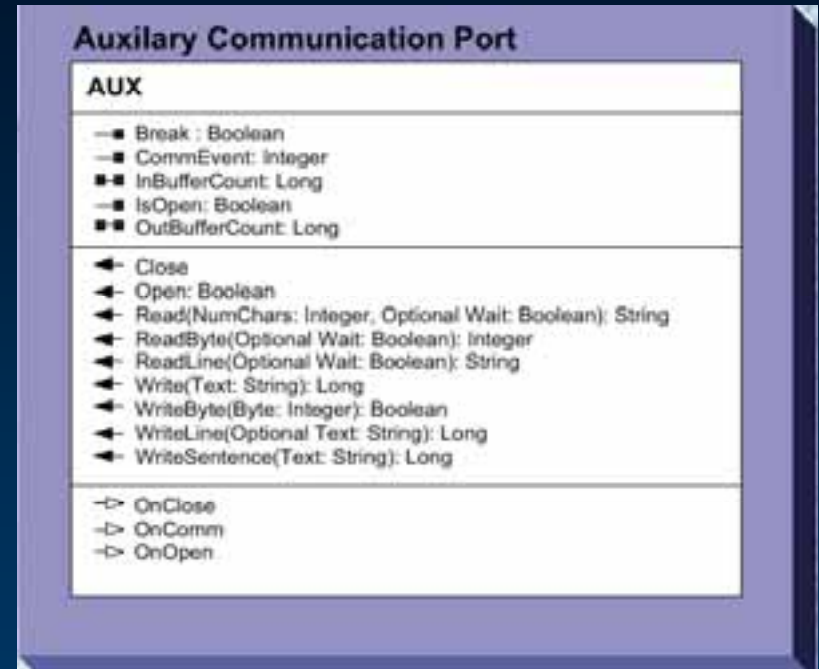
# Serial Device Communication

# Serial Device Communication

- **ArcPad supports serial communications**
- **Bluetooth devices are supported via the Serial Port Protocol (SPP)**
- **GPS, Rangefinder, and AUX communications are exposed via the corresponding objects**
- **Serial parameters are specified in ArcPadPrefs.apx**

# AUX Object

- Two way communication with attached serial device
- Supports ASCII and binary communication
- Uses include control or monitoring of other instruments or devices
  - Water quality probes
  - Soil quality probes
  - Temperature sensors



# Read and parse AUX port example

```
function ReadAUXForTemp()
{
// called in OnComm event to read data from AUX port

// read from the AUX port
var strRawData = AUX.ReadLine();

// parse the comma-delimited data
var arrParsedData = strRawData.split(",");

// if data is valid, return the temperature
if (arrParsedData[0] == "$PTEMP")
    return (parseFloat(arrParsedData[1]));
else
    return (NaN);
}
```



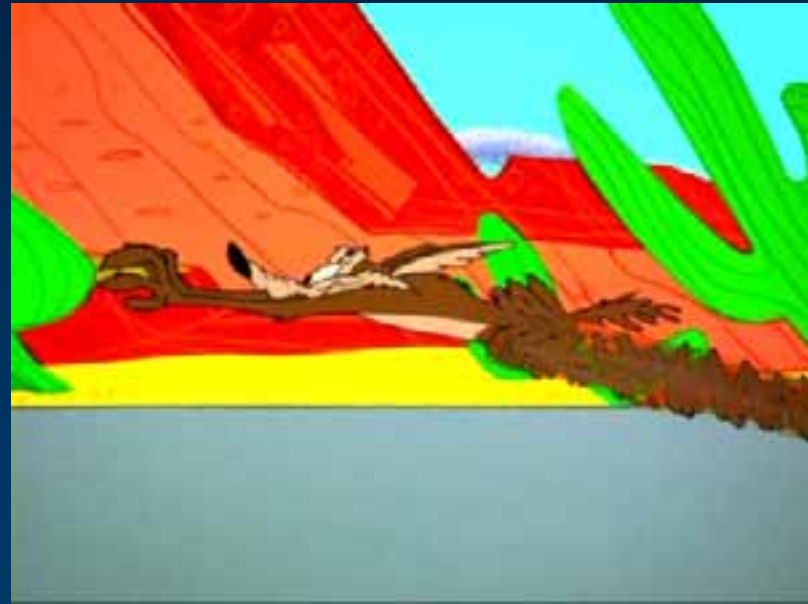
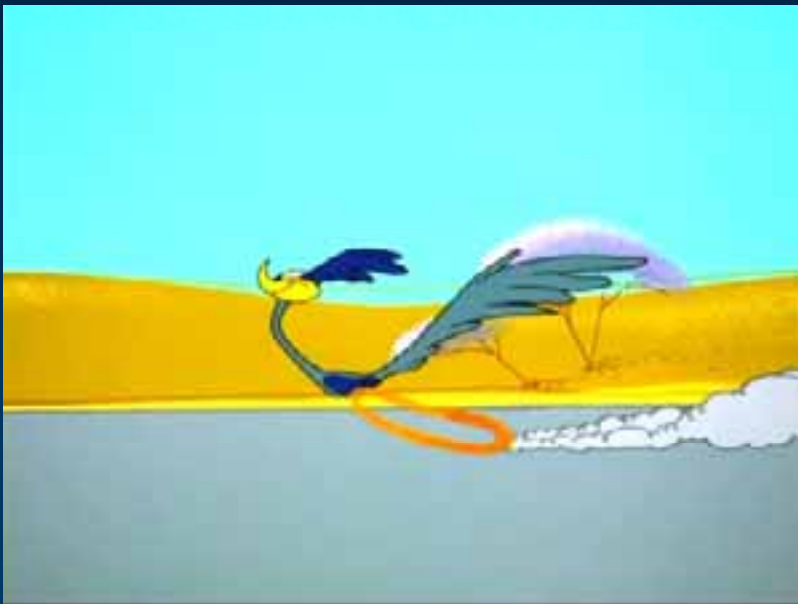
# Internet Communication

# Internet Communication

- **Move some application logic to the server side**
  - You must be connected to the internet (wired or wireless)
- **INET object provides connectivity with generic web servers (ASP, JSP, ASP .NET, ISAPI, etc.)**
- **FTP object provides access to FTP operations**
- **ArcIMS object provides connectivity with ArcIMS**
  - You must construct requests and parse responses
- **3rd party ActiveX components for consuming web services**
  - PocketSOAP

# Road Runner

- Demo
  - Help Road Runner out smart the Coyote!



# Road Runner (Overview)

- **Demoed features**
  - Demos the internet object: INET
  - Asset management with back-to-base reporting
  - First tier: ArcPad with an INET object
  - Middle tier: IIS ISAPI application that processes the INET data
  - Final tier: SQL Server

# Road Runner (ArcPad INET Code sample)

```
Sub OnUpload
  Dim L, R, C, INET, SQL, Result
  Set INET = Application.CreateAppObject("INET")
  Set L = Layers("RoadRunner.shp")
  Set R = L.Records
  Call R.MoveFirst
  C = 0
  While Not R.EOF
    If R.Fields("UPLOADED").Value <> "Y" Then
      SQL = "INSERT INTO ASSETS (ID, TXT, X, Y) VALUES ("
      SQL = SQL & R.Fields("ID").Value
      SQL = SQL & ",'" & R.Fields("TXT").Value & "'"
      SQL = SQL & "," & R.Fields.Shape.X
      SQL = SQL & "," & R.Fields.Shape.Y
      SQL = SQL & ")"
      Result = INET.OpenURL("http://nw8240/scripts/SQLExec.dll?sql=" & SQL)
      R.Fields("UPLOADED").Value = "Y"
      C = C + 1
    End If
    Call R.MoveNext
  Wend
  MsgBox C & " records uploaded"
End Sub
```

# Road Runner (C++ ISAPI Middle-Tier)

- Internet Server Application Programming Interface

```
#include <httpext.h>

BOOL WINAPI GetExtensionVersion(HSE_VERSION_INFO *pVer)
{
    pVer->dwExtensionVersion = HSE_VERSION;
    strncpy(pVer->lpszExtensionDesc,
            "My ISAPI Extension", HSE_MAX_EXT_DLL_NAME_LEN);
    return TRUE;
}

DWORD WINAPI HttpExtensionProc(EXTENSION_CONTROL_BLOCK *pECB)
{
    MyWrite(pECB, "<HTML><HEAD>...</HEAD><BODY>");
    // Extract a SQL statement from pECB->lpszQueryString and execute it.
    MyWrite(pECB, "</BODY></HTML>");
    return HSE_STATUS_SUCCESS;
}
```



# Integration With Other Applications aka “Inter-process Communications”

# Integration With Other Applications

- **Send scripts and events to ArcPad**
  - **SendMessage with WM\_COPYDATA**
  - **PostMessage with WM\_ARCPAD\_EXEVENT**
- **Data Exchange**
  - **Via Windows**
  - **Via Files**
  - **Via Shared Memory**
- **ArcPad's System Object exposes windowing WinAPIs**
  - **FindWindow, ShowWindow, GetClassName, GetWindowText, SetWindowText, SendMessage, PostMessage**
- **COM DLLs that support IDispatch**

# Send scripts to ArcPad from another app

- **Free examples with source code on ArcScripts**
  - Send script to ArcPad from external C/C++ program
  - Send script to ArcPad from external VB application
  - Send scripts to ArcPad from external .NET Compact Framework applications (C# / VB.NET)
  - Sending Scripts to Arcpad from external eVB-application

# Communicating with a .NET application

- **Sending data to controls in a .NET form**
- **Getting data from control in a .NET form**
- **Use System object**
  - FindWindow to locate the .NET application window
  - GetWindow to locate specific controls in the .NET window
  - SetWindowText to send data
  - GetWindowText to retrieve data
- **Example:**
  - .NET application contains business forms
  - Starts ArcPad
  - Digitize applet sends X,Y to .NET application



# Tips & Tricks Best Practices

# Tools of the Trade

- **ArcPad Studio**
- ***Spy++* and *Remote Spy++***
  - Used to find windows in other applications
  - Remote Spy++ comes with eVC and VS2005
  - Spy++ comes with Visual Studio
  - There are freeware equivalents if you don't have Visual Studio
    - Eg [www.dennisbabkin.com/winid](http://www.dennisbabkin.com/winid)
- **SOTI Pocket Controller**
  - Control your device via the desktop
- **Writing Extensions**
  - Visual Studio (VS2005 or VS2003 .NET)
  - Embedded Visual C++ (eVC4)

# Scripting Spaces

- **Several instances of scripting engines created**
  - Application → ArcPad.apx
  - Applets → Each loaded applet (.APA file)
  - Layers → Each loaded layer with an .APL file
- **Application.UserProperties can store variables that can be accessed from any scripting space**  
**Application.UserProperties("UserID") = lngID;**
- **The .Execute method can execute scripts that reside in a different scripting space**  
**Application.Applets("MyCoolApp").Execute("ProcessData");**

# Dealing with Directory Paths

- **Directory structures are very different between Windows CE and desktop Windows**
  - Avoid hard-coding paths wherever possible!
- **Layer.FilePath** returns the path of a layer
- **Application.Path** returns the path of ArcPad.exe
- **Preferences.Properties** returns
  - Default maps & data path (`Preferences.Properties("DataPath")`)
  - System files path (`Preferences.Properties("SystemPath")`)
  - Additional Applets file paths (`Preferences.Properties("AppletsPath")`)
  - Additional Extensions file paths  
(`Preferences.Properties("ExtensionPath")`)
- **Application.System.Properties** returns
  - My Documents path (`Application.System.Properties("PersonalFolder")`)
  - Windows temp folder path  
(`Application.System.Properties("TempFolder")`)

# Key Form Events

- **Initialization**
  - Use Form's OnLoad event
- **Closing**
  - Use Form's OnOK / OnCancel events
- **Validation**
  - Use each Control's OnValidate event
  - Use Page's OnValidate event
- **Control interaction**
  - ComboBox / ListBox → OnSelChange event
  - CheckBox → OnClick event
  - Button → OnClick event

# Form Control Naming Conventions

- Form → frm
- Page → pg
- Label → lbl
- Button → btn
- RadioButton → rdo
- ComboBox → cbo
- Slider → sld
- DateTime → dt
- TextBox → txt
- CheckBox → chk
- ImageBox → img
- ListBox → lbo
- UpDown → ud
- SubTable → stbl

# Debugging Tips

- Always use **Compile** tool to test projects
- Use **Console.Print** instead of **Application.MessageBox**
  - Does not stop script flow
- Put **Console** in debug mode to see events and performance
  - **Console.DebugFlags = 3**
- Use **Script** dialog to test code snippets
  - **Ctrl + Enter**
  - Cut and paste between Notepad and Script dialog
- Add error handling last
  - **On Error Resume Next (VBScript)**
  - **Try...Catch...Finally (JScript)**

# Reusing Code

- **Use relative references wherever possible**
  - OnLoad event of a form  
`Set pForm = ThisEvent.Object`  
`Set pForm = Applet.Forms("frmStats")`
  - OnClick event of a checkbox  
`Set pControls = ThisEvent.Object.Parent.Controls`  
`Set pControls = Applet.Forms.Pages("Page2").Controls`
  - OnOpen event of a layer  
`Set pRS = Layer.Records`  
`Set pRS = Map.Layers("Soils.shp").Records`

# Reusing Code

- **Works:**

```
<LAYER onload="Map.Layers(&quot;soils.shp&quot;).Editable = True">
```

- **Better:**

```
<LAYER onload="Layer.Editable = True">
```

- **Best:**

```
<LAYER onload="Call InitLayer">
```

# Resources

- <http://arcpadteam.blogspot.com>
- <http://support.esri.com>
- <http://arcscripts.esri.com>
- <http://msdn.microsoft.com/scripting/>
- <http://www.pocketpc.com/>
- <http://www.devbuzz.com/>
- <news://microsoft.public.scripting.vbscript>
- <news://microsoft.public.pocketpc.developer>

# Presentation materials

- PowerPoint presentation and code are posted on the conference web site
  - <http://www.esri.com/events/devsummit/index.html>
- EDN – downloads and videos

# Further questions?

- **TECH-TALK AREAS**

- **What:** Further opportunity to discuss questions and concerns with presenters and subject matter experts

- **Where:**

- **When:** during the next 30 minutes

- **ESRI Showcase**

- **Meet the teams**