Extreme Makeover: Moving from MapObjects to ArcGIS Engine

Jim Barry
Jianxia Song
Outline

- MapObjects vs. ArcGIS Engine
  - What are they? Why migrate?
- Functional Comparisons
  - How to migrate. Side by side.
- Deployment, Distribution Comparisons
  - Licensing

Tech Talk Room 3
Friday, March 17
11:30 a.m.–12:00 noon
Tech Talk—Extreme Makeover: Moving from MapObjects to ArcGIS Engine
Assumptions

• MapObjects-Windows Edition development

• ArcGIS Engine
  – Have used it
    • OR
  – May use it
MapObjects-Windows Edition

- Since 1997
- 50 object classes
  - Data Access
  - Map Display
  - Geocoding
  - Projections
  - Geometric Ops
- Map control
- Sample controls
  - Legend
  - Scalebar
MapObjects-Windows Edition

• **Pros:**
  - Simple map display, rendering, interaction
  - Shapefiles, ArcSDE, CAD, VPF, Imagery
  - Can mix coordinate systems (vector only)
  - Good set of basic renderers, customizable
  - Strong, customizable geocoders
  - Small API, small application footprint
MapObjects-Windows Edition

- **Abilities** (developer- or user-controlled):
  - Display map data
    - Interactivity: pan, zoom, thematic rendering
  - Queries:
    - spatial, attribute
  - Address matching
  - Very basic editing
  - Exposed Interfaces
    - ICustomRenderer
    - ICustomProjection
    - ICustomMarker, ~Line, ~Fill, ~Chart
MapObjects-Windows Edition

- **Cons** (compared to ArcGIS Engine):
  - Lack of coarse grained objects, controls
    - Rendering, Geoprocessing, Layout display
    - GUI tools
  - Single-threaded
  - Non-persistable: app state, locators, layer def.
  - Map output options limited
  - Weak symbolsets
  - Weak editing functionality, esp. multi-user
  - Weak support for latest data format types, versions
  - Weak support for labeling, annotation
MapObjects-Windows Edition

• **Cons** (compared to ArcGIS Engine):
  - Spatial reference functionality based on 8.x
  - Imagery not projectable
  - No raster GIS functionality, 3D display
  - No interoperability with ArcGIS
    • Personal Geodatabase
    • .mxd, .lyr, .style
  - No data conversion tools
  - No mechanism for controlling distribution
  - Lack of extendable framework
  - **Feature Complete**
ArcGIS Engine

- **Built on core ArcObjects components**
  - Can be extended by building custom COM components
- **1,000s object classes and interfaces**
  - Full GIS functionality
- **Controls**
  - Map, Table of Contents
  - PageLayout – for output operations
  - Toolbar – over 240 tools ready to use
  - Scene, Globe, Symbology
ArcGIS Engine

- **Why migrate?**
  - Easy to develop
    - MO: No Framework
    - AGIS: Solid Extendable Framework
  - Easy to deploy
    - MO: Difficult, manual, each app
    - AGIS: Easy, automatic, each machine
  - Advanced functionality
Migrating

• **Side-by-side comparisons**
  – MapControl and others
  – Toolbars
  – Layer symbology and rendering
  – Address Matching, Geocoding
  – Editing
  – Selecting, Querying
  – Save project, save state
Migrating - Basic Map Form

- Map, Legend, Toolbars

- MO
  - Connect to data locations, load individually

- ArcGIS Engine
  - Design-time: no code solution
  - Objects: load .mxd, .lyr, datasets through code
    - IWorkspaceFactory
  - Extension: load custom datasets

MapControl demo
Migrating – Data Loading

• MO
  – Design-time: Property page, not portable
  – Run-time:
    • Data access objects.
    • No save, restore

• ArcGIS Engine
  – Design-time: **Best:** create an .mxd
  – Objects:
    • AddShapefile, AddLayer, AddMxFile methods
    • WorkspaceFactory functionality for other data formats
  – Extendable: exposed interfaces
Migrating – Data Loading

Dim dc As New MapObjects2.DataConnection
dc.Database = "C:\data"
If Not dc.Connect Then
    gds = dc.FindGeoDataset("parcels")
    If Not gds Is Nothing Then
        mlyr.Geodataset = gds
        Map1.Layers.Add(mlyr)
    End If
End If

Vs.

AxMapControl1.AddShapefile("C:\data", "parcels")
Migrating – Map Navigation

• **MO**
  - Use generic toolbar control
  - Write all you own code behind each tool and button

• **ArcGIS Engine**
  - Design-time: Use the Map Navigation Toolbar, design-time
  - Objects: add tools to toolbar
  - Extendable: Create new commands and tools
Migrating – Layer Rendering

• MO
  – Code it all
    • 8 basic renderers
    • Numeric breaks? Generate your own statistics

• ArcGIS Engine
  – Design: Use ArcMap to create .lyr files
  – Objects: Coarse grained rendering styles
  – Extendable: Custom renderers, symbols
Migrating – Layer Rendering

- 5 Statistical Classes
  - Quantile
  - EqualInterval
  - NaturalBreaks
  - DefinedInterval
  - StandardDeviation

- Your own custom break definitions
Migrating – Geocoding

• MO
  – Code it all: matcher setup, single & batch matching

• ArcGIS Engine
  – Design-time: Use ArcCatalog to create Locators
  – Objects: Still can code it all, create locator on the fly
  – Extendable: Use GDK to create custom address styles, or extend provided interfaces

Geocoding demo
Migrating – Editing

- MO
  - Code it all: Shapefiles full ability
  - ArcSDE limited functionality

- ArcGIS Engine
  - Editing against the Geodatabase
  - Design: Use the Feature Editing tools
  - Objects: IWorkspaceEdit
  - Extendable:
    - Custom editing commands and tools
    - Write Geodatabase extension

Editing demo
Migrating – Selecting, Querying

• **MO**
  - Code it all
  - `MapLayer.Search` methods

• **ArcGIS Engine**
  - Design-time: Use the Feature Selection and Map Inquiry tools
  - Objects: Queries and selection
  - Extendable: Create new tools of your own
Migrating – Save Project

- MO
  - Code it all

- ArcGIS Engine
  - Design-time: MXDs
  - Objects: IMapDocument.Save

Save demo
Resources

- Product Documentation
- EDN website
  - Documentation Library, Downloads, Code Exchange
- ESRI Support Center
  - Forums, Whitepapers, ArcScripts
- Instructor-led Training
- Implementation Services
Deployment and License Distribution
MapObjects Runtime

- MapObjects Runtime – **MO23RT.EXE**
- Has Core and optional components
- Using command line arguments to specify optional components to install:
  - `mo23rt.exe /BIJOQR`
    - ArcSDE, Memtable, Imagery, ESRI fonts, Legend/Scalebar, .NET
- Includes assemblies for .NET deployment
MapObjects Distribution Licenses

- 50 deployment seats included with dev kit
- Add’l deployments available for purchase
  - 25 packs
  - Floating cost based on quantity
ArcGIS Engine licensing requirements

• **Components**
  - Must have ArcGIS Engine Runtime installed

• **Licensing**
  - Must have an ArcGIS license available
    - Desktop
      - ArcView, ArcEditor and ArcInfo
    - Engine
      - Standard Runtime
      - Geodatabase Extension
Installing the ArcGIS Engine Runtime

- ArcGIS Engine Developer Kit contains an ArcGIS Engine Runtime CD
- The runtime CD is freely distributable with your application
- CD contains a pre-built MSI that will install all of the required components for end users
- ArcGIS 9.2 Desktop will contain the runtime, no need to install on these systems…
How to check out a license with your Engine application

• Three ways:

1. Programmatically – AoInitialize() object

2. Add-ins

3. License control

* Can also check-out and check-in extensions!
How to license end-user machines

- Need to authorize each end-user machine to run ArcGIS Engine applications
  
  **Methods**
  
  - SoftwareAuthorization.exe wizard
    - Run manually by the end-user
    - Run automatically by your setup program
  
  - Or use an available ArcGIS Desktop license

  - Can be performed silently with ArcObjects code
    - Must be ESRI business partner
    - Must have special OEM license agreement with ESRI
For more information

- **EDN**

- “Strategies for Packaging and Deploying your ArcGIS Solutions” - Saturday 2:30-3:45 pm

- **Comments?**
  - Tech Talk from 11:30-12:00, Oasis 3
  - Or see Rob Elkins
Session Evaluations Reminder

Session Attendees:
Please turn in your session evaluations.

. . . Thank you

Jianxia Song – jsong@esri.com
Jim Barry – jbarry@esri.com
Q&A

Jianxia Song – jsong@esri.com
Jim Barry – jbarry@esri.com