



ArcGIS Server Geoprocessing (Deep Dive)

Sentha Sivabalan and Corey Tucker

Overview

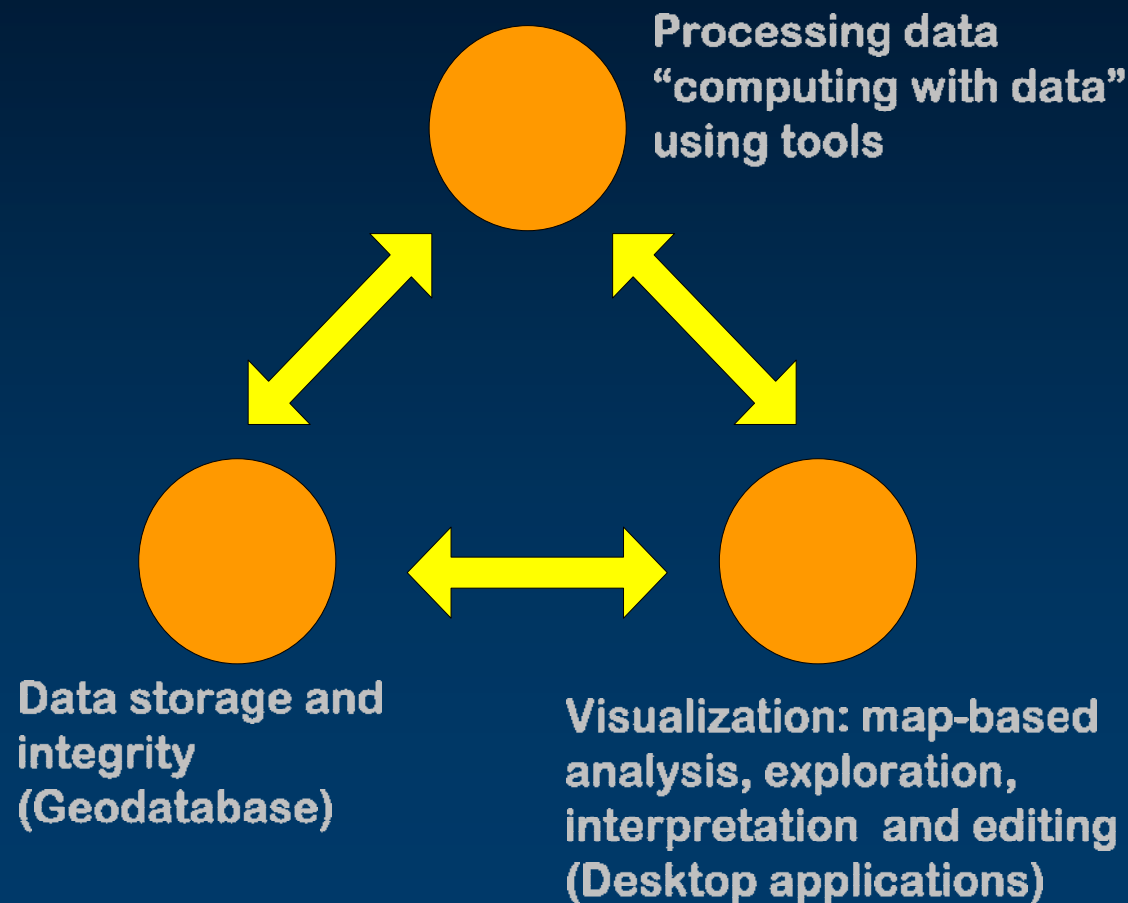
- **Review what is Geoprocessing**
- **Review what Geoprocessing with ArcGIS Server is all about**
 - How does it work
- **Review considerations when publishing models**
- **Using geoprocessing services**
 - Result Management, provide inputs
- **Using Geoprocessing services from the ADF's**

What is Geoprocessing?

- **Geoprocessing is the process of defining and executing a workflow**
- **A workflow defines the execution of a task**
 - Find a path from here to there
 - Find all the homes within a certain distance of here
 - Download data that meets certain criteria from a database
 - Etc
- **In ArcGIS a model is a workflow**
- **Tools are the building blocks of the ArcGIS geoprocessing framework**
- **Users can easily build new tools by combining existing ones using ModelBuilder**
 - Models are new tools with input and output parameters

Geoprocessing is...

...one of the 3 critical components of a GIS



Geoprocessing with ArcGIS Server

- **Geoprocessing Services were created so GIS professionals can publish tools to be used as tasks**
- **Tasks can be used in multiple client applications**
 - ArcGIS Explorer
 - Web ADF
 - ArcGIS Desktop
 - WSDL client

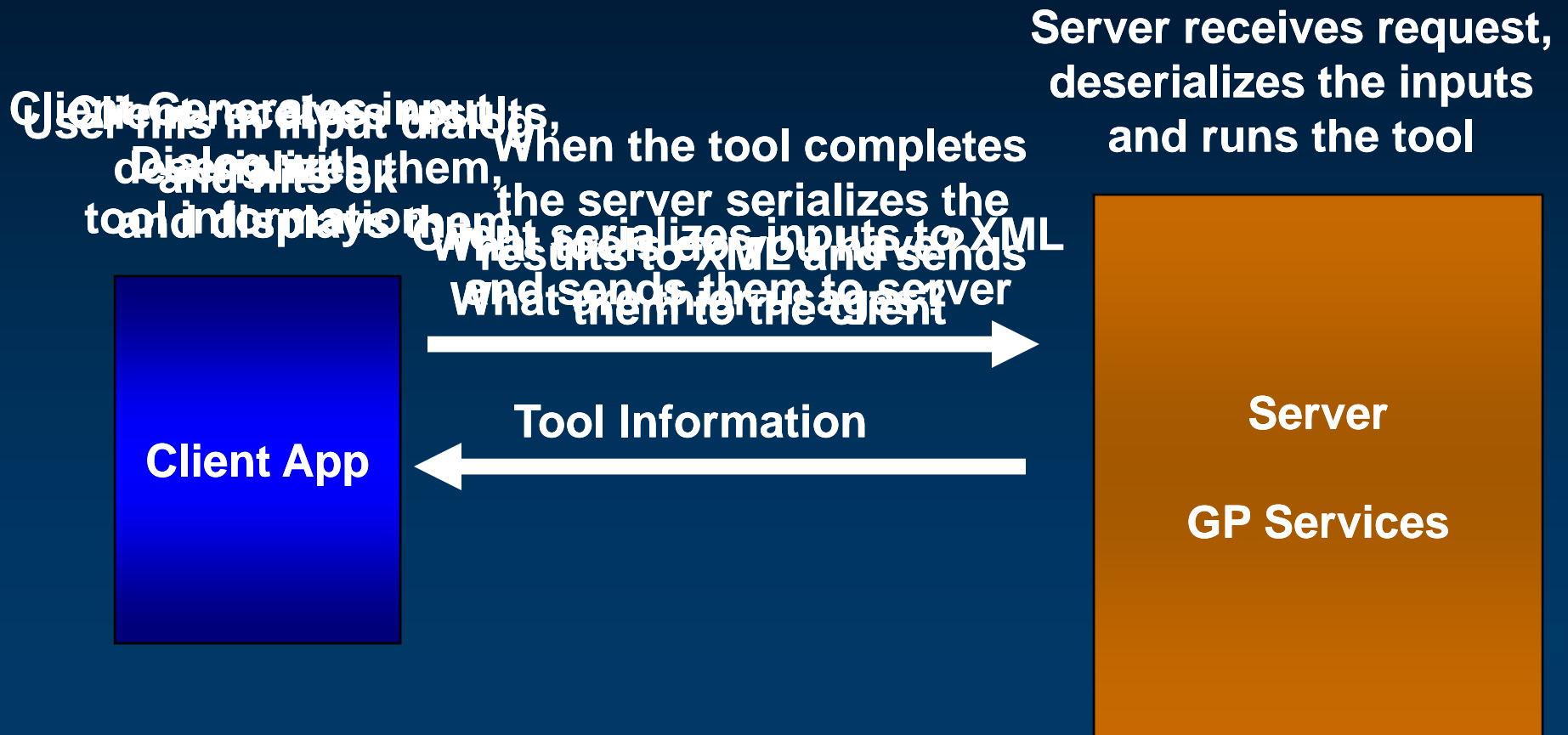
Geoprocessing with ArcGIS Server

- Any workflow built in ModelBuilder can be published to a server
- Changes will probably be required to make it happen, as designing a tool for a server takes some special considerations
- These include
 - What parameters are exposed to the end user?
 - Has all unnecessary processing been factored out of the workflow to improve performance
 - Are input datasets modified? If so, they probably should not be, as many instances of the model will use them.
 - And there are more....

Geoprocessing with ArcGIS Server

- **Tools must be published to be executed**
 - You cannot just execute any tool shipped with ArcGIS as a service
 - The tool must first be published as part of an ArcGIS Server Geoprocessing service
- **A geoprocessing service is the combination of a tool and its data**
 - Properties of the data must be defined prior to publishing a tool
 - **Data Type:** Raster, Feature, File, Scalar Value, Layer
 - **Feature Type:** Point, Line or Polygon
 - **Attributes:** What fields are editable?
 - **Symbology:** How are inputs and output drawn?

How a Geoprocessing Service works



How to publish a Geoprocessing Service

- Depending on how you want to provide the results of your service, you can choose from two publishing options

1. Publish a toolbox

- The tools in the toolbox become the available tasks
- This allows you to load balance a group of tools
- Data is always returned to the client
- Commonly called a **By Value** GP Service

2. Publish a map document

- Tool layers in the map document become the available tasks
- A map service displaying the result is returned to the client
- Commonly called a **By Reference** GP Service

How to publish a Geoprocessing Service

- **Publishing a map document creates a results map service and a geoprocessing service**
 - Useful for large datasets as the results stay on the server
 - Provides access to results that are not supported through the Web ADF's, such as rasters or TIN's
 - Both services always have the same name
- **This allows you to use layers in the map document as input to your tasks**
 - Improves performance as the data is opened for access
 - You can disassociate a map service from a geoprocessing service if you want to use map layers for input, but not have the results returned as a map service, but By Value

Publishing Guidelines – Data Types

- **Tool parameters must be a data type supported by ArcGIS Server**
 - **The model or script tool may use any data type internally, but the exposed parameters are limited to the following:**
 - **Feature Set**
 - **Record Set**
 - **Raster**
 - **String**
 - **File**
 - **Double**
 - **Long**
 - **Date**
 - **Boolean**
 - **Linear Unit**

Publishing Guidelines – Data Types

- **Model outputs are automatically converted to a supported data type whenever possible**
 - Table converts to record set
 - Feature class converts to feature set
 - All other unsupported types are published as a String
- **Output properties are defined by:**
 - The model variable - Symbology
 - The result of execution - Feature Type & Attributes

Publishing Guidelines – Data Access & Paths

- **Paths to data must be accessible to the geoprocessing service machine (SOC)**
 - Can use relative paths or UNC paths to make the toolbox or map document portable
 - Relative paths are relative to the toolbox or map document location
- **All output datasets must go to the Scratch Workspace**
 - The environment setting is dynamically defined for each job on the server
- **All intermediate data should be managed so it is created and deleted correctly**

Publishing Guidelines - Data Access & Paths

- **A file geodatabase is always created in the job directory for output that must go into a geodatabase (feature classes containing curves, long field names, etc)**
 - Always called Scratch.gdb
- **Outputs of a By Value service may be written to the in-memory workspace**
 - Only feature classes and tables are supported
 - Fast
 - Only appropriate for small datasets.

Publishing Guidelines – Server Types

- **There are two types of Geoprocessing Services**
 - Synchronous & Asynchronous
 - Chosen at time of publishing. Clients have no control over this.
- **Synchronous** services require the client to wait for the result when a task is executed
 - Good for fast tasks that take 1-3 seconds
 - Always provides a By Value result
- **Asynchronous** services immediately returns a result when a job is submitted to the server
 - The client can check the status and get a result at its leisure
 - Good for tasks that take time to complete or require a map service as the result
 - Can provide a By Value or By Reference result

Publishing Guidelines - Tips

- **Consolidate your tools into one toolbox if possible**
 - Creates a single service, which is a better use of resources
 - Can also have a map document with many tool layers
- **Use the structure for sharing tools as defined in the Desktop Help system**
 - Easy to understand and transport

http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=A_structure_for_sharing_tools

Publishing Guidelines - Tips

- **The File data type can be used to pass datasets of an unsupported type**
 - Using the a zip file, a task can return a file containing a workspace or any other file type
 - Great for data extraction services
 - See the ArcGIS Desktop help for more information

[http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Python scripts to zip and unzip data](http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?TopicName=Python%20scripts%20to%20zip%20and%20unzip%20data)

Result Management

- **A Jobs folder is created when ArcGIS Server is installed**
- **It will contain all of the results created by geoprocessing services**
 - Each new request will create a new folder with a unique name
 - This new folder will be the scratch workspace when the model you published as a task is executed by the server
- **Job folders are maintained by the service administrator**
- **Clients simply maintain a link to asynchronous results using the Job ID, the service name and the tool name**
- **If a result no longer exists on the server, it must be recreated**

Result Management

- **The size of a resultant feature set or record set may be limited by the publisher**
 - Prevents large results from being downloaded
 - Default is 500 records. Allowing large downloads will impact performance.
- **Raster results are not limited in size**
 - Use a By Reference result whenever the output raster could be large
 - Rasters that are downloaded are in the GeoTiff format

Using Geoprocessing Services

- Geoprocessing Services can be accessed programmatically using the Geoprocessing object or the GP Server object
- For Desktop and Engine developers, the IGeoprocessor provides the best access to the server

```
Private pGP As ESRI.ArcGIS.Geoprocessing.IGeoProcessor

' Add server toolbox
pGP.AddToolbox( "baccalieu;SanJuan_GPTool " )

' Execute the tool and get the ID of the Job created on
' the server
pResult = pGP.Execute("camp_colorado", pParamArray, Nothing)
m_JobID = pResult.ResultID
```

Accessing Results

```
' Create the result from the Job ID and tool name
```

```
Dim pResult As IGeoProcessorResult
```

```
pResult = pGP.CreateResult("Camp_Colorado", m_JobID)
```

```
Dim DataPath as String
```

```
Dim pGPUtills As IGPUtillities
```

```
pGPUtills = New GPUtillities
```

```
Dim pRasterDS As IRasterDataset
```

```
Dim pRasterLayer As IRasterLayer
```

```
' Check the job's status. If successful, add the result
```

```
' to the map
```

```
If pResult.Status = esriJobStatus.esriJobSucceeded Then
```

```
    DataPath = pResult.GetOutput(0).GetAsText
```

```
    pRasterDS = pGPUtills.OpenRasterDatasetFromString(DataPath)
```

```
    pRasterLayer = New RasterLayer
```

```
    pRasterLayer.CreateFromDataset(pRasterDS)
```

```
    Me.AxMapControl1.AddLayer(pRasterLayer)
```

```
End If
```

Using Geoprocessing Services – Feature Input

- **A feature class cannot be passed to a geoprocessing service**
 - **Instead you must create a feature record set and populate it with input features**
 - **Feature Recordsets support cursor access**
 - **Tools accept GPRcordset objects as valid parameter values**
 - **Create the object and then set its record set property using a generic feature record set**

Using Geoprocessing Services – Feature Input

```
Dim pRecordSetInit As IRecordSetInit
pRecordSetInit = New RecordSet
Dim Template as String
Template = "D:\Model\Data\exclusion_poly_template.shp"
Dim pGPUtills As IGPUtillities
pGPUtills = New GPUtillities

Dim pFeatureClass As IFeatureClass
pFeatureClass = pGPUtills.OpenFeatureClassFromString(Template)
Dim pFields As IFields
pFields = pFeatureClass.Fields
pRecordSetInit.CreateTable(pFields)
Dim pRecordSet As IRecordSet
pRecordSet = pRecordSetInit
```

Using Geoprocessing Services – Feature Input

```
Dim lField As Long
' Find the location of the geometry field
lField = pFields.FindField("SHAPE")

Dim pCursor As ICursor
pCursor = pRecordSetInit.Insert
' Open an insert cursor and set the geometry for the first row
Dim pRowBuffer As IRowBuffer
pRowBuffer = pRecordSetInit.CreateRowBuffer
pRowBuffer.Value(lField) = pGeom
pCursor.InsertRow(pRowBuffer)
' Create a GP Record set to be used as an input parameter
Dim pGPRecordSet As IGPRecordSet
pGPRecordSet = New GPFeatureRecordSetLayer
pGPRecordSet.RecordSet = pRecordSet
```

Using Geoprocessing Services

- **There are options for thin clients who have some data restrictions**
- **Densify Features**
 - **The Web ADF's do not support curves**
 - **Set DensifyFeatures to true to make sure curves are not returned**
- **Transport Type**
 - **Default is embedded**
 - **Possible to get url back for file or raster. May be useful for http access of file. The output or job directory must be an http directory for this to work.**



Demo – Authoring and Publishing a Geoprocessing Service

Geoprocessing with the ADF

- **The ADF Task Framework**
- **The Geoprocessing Task**
- **Working with a Geoprocessing Service Programmatically**

Technology used

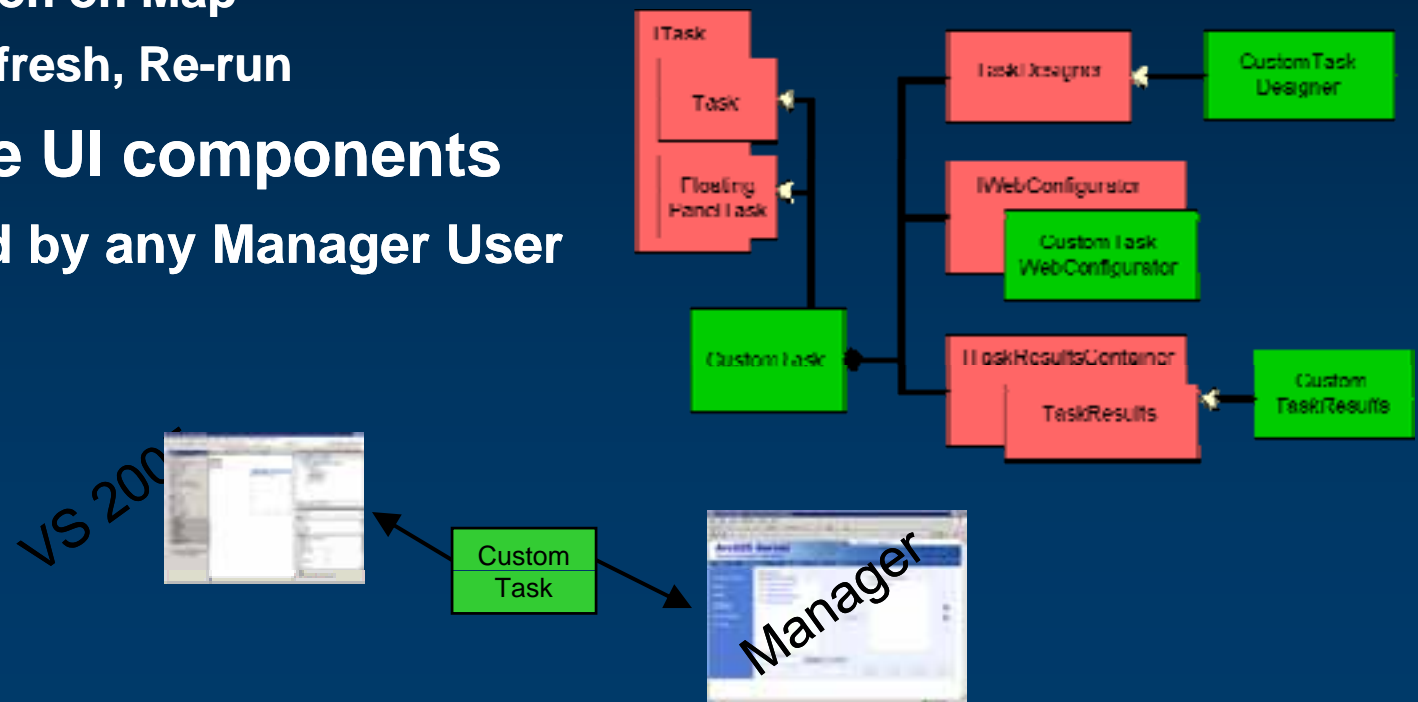
- **Examples use the .NET ADF**
 - Most information common to Java and .NET ADF
- **Code examples in C#**
 - Java equivalent explained

Web ADF Tasks

- **Encapsulate a workflow**
- **Generate a result**
- **Out of the box Tasks**
 - SearchAttributes
 - FindAddress
 - FindPlace
 - QueryAttributes
 - **Geoprocessing**
 - Editor

The ADF Task Framework

- Framework provides for:
 - Organization of tasks
 - Feedback when a task is executing
 - Management of results
 - Display
 - Visualization on Map
 - Delete, Refresh, Re-run
- Distributable UI components
 - Can be used by any Manager User
- Extensible



The .NET ADF Geoprocessing Task

- **Out-of-the-box UI for Input/Output parameters**
 - Feature set
 - Record set
 - File upload/download
 - Simple types
 - Date
 - Boolean
 - Long
 - Double
 - String
 - Linear Unit (e.g. 10 miles)

Demo: The ADF Geoprocessing Task

- **Job status**
- **Job messages**
- **Interacting with results**
- **Retrieving results of a previously executed job**

Geoprocessing Task: Properties

- **TaskDefinition**
 - Server, Service and Task
- **AsyncStatusCheckInterval**
 - How often job status is checked
- **VirtualDirectory**
 - Place on web server where file upload/download happens
- **MaxFileUploadSize**
 - A limit on file sizes for uploads
- **DrawResultsOnMapServer**
 - Whether results should be drawn on server, if available

Working with a GP Service Programmatically

- Generate proxies using WSDL
- Use the pre-generated proxies in the ADF
 - Can use with ArcGIS Server Local and Internet

```
//Internet
using (ESRI.ArcGIS.ADF.ArcGISServer.GPServerProxy gpServer =
    ESRI.ArcGIS.ADF.ArcGISServer.GPServerProxy.Create(
        "http://myserver/arcgis/services", "myservice", MyIdentity))
{ //call server methods }

//Local
using (ESRI.ArcGIS.ADF.ArcGISServer.GPServerProxy gpServer =
    ESRI.ArcGIS.ADF.ArcGISServer.GPServerProxy.Create("MyLocalMachine",
    "myservice", MyIdentity))
{
    //call server methods
    //Server context created/released within using block.
}
```

Working with a GP Service Programmatically

- **Java**

- **Connecting to the server:**

```
GPServerBindingStub stub = new GPServerBindingStub(new  
    URL("http://myServer/arcgis/services/Test/GPServer"),null);
```

```
//call GP server methods here
```

- **Server methods named the same**

Methods providing Information

- **GetExecutionType**
 - Synchronous, Asynchronous
- **GetTaskNames**
 - Names of all the tasks
- **GetTaskInfos**
 - Information about all tasks
- **GetToolInfo**
 - Information about a particular task
- **GetResultMapServerName**
 - Is a map service available to draw results on server?

Methods for Executing a Task

- **Synchronous Execution**
 - **Execute**

- **Asynchronous Execution**
 - **SubmitJob**
 - **GetJobMessages**
 - **GetJobStatus**
 - **GetJobResult**
 - **CancelJob**

Synchronous Execution

```
//Setup Inputs
GPValue[] inputs = new GPValue[2];
inputs[0] = gpLayer; //Record Set Layer
inputs[1] = gpUnit; //Linear Unit

//Setup result options
GPResultOptions options = new GPResultOptions();
options.DensifyFeatures = true;
options.DensifyFeaturesSpecified = true;

//Execute
GPResult result = gpServer.Execute(Task, inputs, options);
```

Asynchronous Execution

```
//Submit job
string gpJobID = gpServer.SubmitJob(Task, inputs);
//Check job status
esriJobStatus status = gpServer.GetJobStatus(gpJobID);
//Wait while job completes
//You can store job id and use it later to get results too.
while (status == esriJobStatus.esriJobNew || status ==
    esriJobStatus.esriJobSubmitted || status == esriJobStatus.esriJobWait
    ing || status == esriJobStatus.esriJobExecuting)
{
    System.Threading.Thread.Sleep(5000);
    status = gpServer.GetJobStatus(gpJobID);
}
//When job completes, get results
GPResult result = gpServer.GetJobResult(gpJobID, parameterNames,
    options);
```

Demo: A Custom Task using a GP Service

- **Calling a GP Service synchronously/asynchronously**
- **Using the ADF Task Framework to return results**

Presentation materials

- PowerPoint presentation and code will be posted on the conference web site

– <http://www.esri.com/events/devsummit/index.html>

Microsoft Keynote

- *Eddie Amos*
 - *Senior Director, Developer & Platform Evangelism*
- **6:00pm Today**
- **Oasis 2**

Further questions?

- **TECH-TALK AREAS**

- **What:** Further opportunity to discuss questions and concerns with presenters and subject matter experts
- **Where:** Tech Talk Area 3
- **When:** during the next 30 minutes

- **ESRI Showcase**

- **Please fill out your session evaluation**

- **Thank-you!**