



# Building and Optimizing Geoprocessing Services for ArcGIS Server

*Nathan Warmerdam*

*Scott Murray*

# Technical Workshop Outline

- **Documentation, Design and Configuration**
- **Example 1: Clip and Ship**
- **Example 2: Suitability Overlay**
- **Performance Tuning**
- **Performance Analysis**
- **Troubleshooting**

# Geoprocessing Service Documentation

- Geoprocessing Resource Center
  - <http://resources.esri.com/Geoprocessing/>
- Desktop Help → Geoprocessing → Geoprocessing with ArcGIS Server
  - [http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=An overview of geoprocessing with ArcGIS Server](http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=An%20overview%20of%20geoprocessing%20with%20ArcGIS%20Server)
- Examples
  - A dozen examples
  - All examples can be downloaded from the resource center
  - <http://resources.esri.com/Geoprocessing/index.cfm?fa=codeGallery>

- **A Tour of the Resource Center and Documentation**

# Geoprocessing Services

- The geoprocessing service allows you to publish custom tools to be used via ArcGIS Server.
- Geoprocessing services can be used by many different client applications
  - ArcGIS Desktop
  - ArcGIS Engine
  - ArcGIS Explorer
  - Web ADF
  - WSDL
  - Rest
  - JavaScript
  - FLEX
  - Silverlight

# Geoprocessing Services

- **Model tools or Script tools** contain the geoprocessing functionality run by geoprocessing services
  - We do not recommend that you publish system tools directly.
- A geoprocessing service is a tool plus its associated data
- Endless array of tasks can be created
  - Spatial analysis (vector, raster, network...)
  - Data Management (geodatabase, file based data)
  - Conversion (ETL and data loading )
  - etc
- You need to be knowledgeable about using geoprocessing tools to create a good geoprocessing service.

# Geoprocessing Service Behavior

- **Geoprocessing Services are very flexible and allow many different behaviors and optimizations**
- **Before Authoring and Publishing, identify what you want your service to do and how you want it to behave with clients.**

# Geoprocessing Services Configurations

- Geoprocessing Service from a toolbox
  - Each tool becomes a task
  - Client draws the results
- Geoprocessing Service from a map document
  - Each **Tool Layer** becomes a task
  - Tasks can access layers in the map document
  - Drawing Results – two options
    - Map Service draws the results
    - Client draws results

Execution Type:  Synchronous  Asynchronous

The tools exposed by the Geoprocessing Service are stored in:

A toolbox  
Toolbox: [ ]

A map  
Map Document: C:\Demos\UC2008\ServerTW\toolayer\Hurricane1 [ ]

Data Frame: Layers [Change...]

**Result Map Service: HurricaneBufferService**

Jobs Directory: c:\arcgisserver\arcgisjobs

Virtual Jobs Directory: http://valima/arcgisjobs

Execution Type:  Synchronous  Asynchronous

The tools exposed by the Geoprocessing Service are stored in:

A toolbox  
Toolbox: [ ]

A map  
Map Document: C:\Demos\UC2008\ServerTW\toolayer\Hurricane1 [ ]

Data Frame: Layers [Change...]

**Result Map Service: None**

Jobs Directory: c:\arcgisserver\arcgisjobs

Virtual Jobs Directory: http://valima/arcgisjobs



## Use a Map Document if:

- **Want to use a map service to draw the results**
  - Better cartographic rendering of results
  - Render large results
  - Output data too large to download
  - Web clients cannot draw raster
- **Using a layer improves the performance of the tool**
  - Network Analyst Layer
  - Select by Location or Select by Attribute
- **Want a choice list of layers in the map document**
  - Clip and ship with the ability to select the layer to clip

# Publishing Considerations – Execution Mode

- **Asynchronous (Submit Job)**
  - Results are saved on the server
  - Results can be drawn on the server
  - Results can also be downloaded if desired
  - Clients free to do other tasks
    - e.g. in ArcMap you can pan/zoom, run other tools while the job is running
  - Appropriate for longer processing jobs.
- **Synchronous (Execute)**
  - Client always receives and draws data.
  - Desktop Client waits until job is completed and results are returned
  - Appropriate for faster processing jobs. (<10 seconds)

# Demo – Creating a Tool Layer and Publishing a Map Document

- **Examples:**

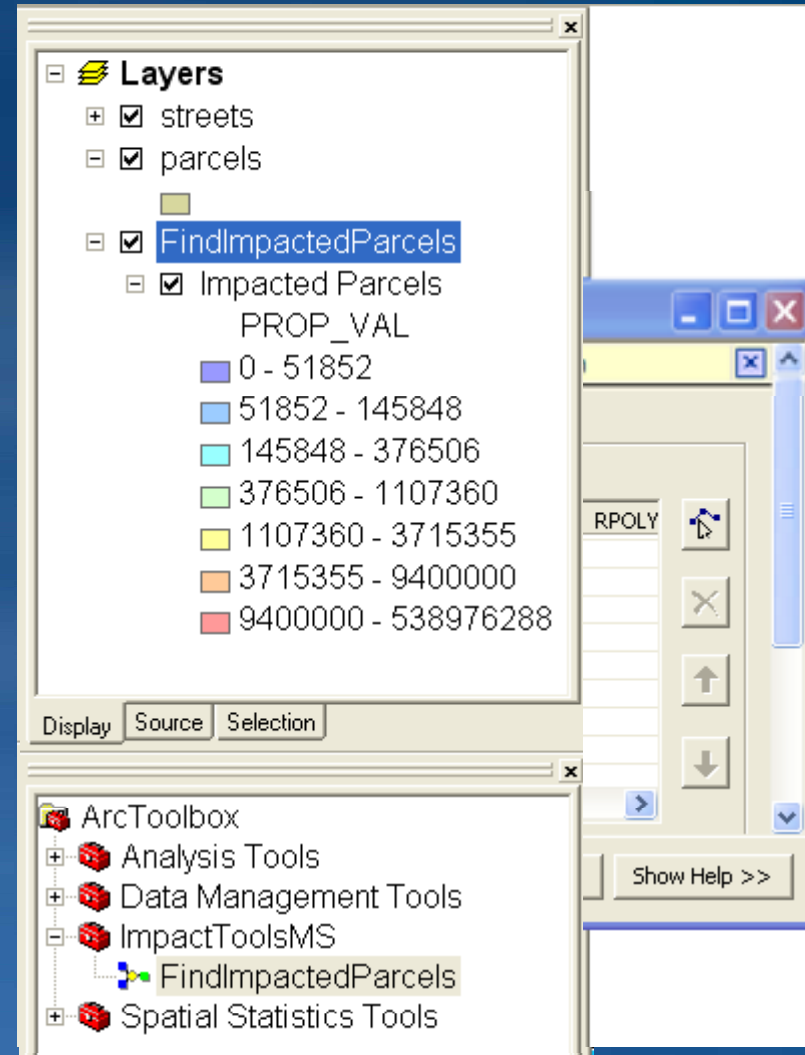
- **Clip and Ship:** enter area to clip a layer on the server and return a zip file
- **Extract Features:** clip features based on a zip code

- **Highlights:**

- Using a tool layer for a geoprocessing service
- Using input features
- Having a choice list of layers on the server
- Doing selection on a layer in the server
- Returning a zip file

# Tool Layer

- **What is a tool layer?**
  - A special group layer containing outputs of a tool
  - Defines parameter symbology
- **How to create?**
  1. Drag and drop a tool into a map
    - Tool outputs are added as sub-layers
  2. Open tool dialog and run











# Authoring a Model suitable for publishing

- Models and scripts that are run by services need to have certain characteristics to run correctly.
- Changes will likely be required to allow existing models and scripts to be published.
- Things to consider when creating a model for use as a Geoprocessing Service
  - Data Management (Source, Intermediate, Output)
  - Data Type of Parameters
  - Symbology
  - Optimization

# Authoring a Model suitable for publishing

- The tool must be portable
  - A new job workspace is created on the server each time the tool is executed.
  - The tool needs to be constructed so that it will run in the job workspace created by the server.

- [-]  arcgisserver
  -  arcgiscache
  - [-]  arcgisjobs
    - [-]  HydrologicModelingTbx\_GPServer
      - [+]  J9711F162B991423BA38DE2628EFB7A5E
      - [+]  JA2BBE914F7144214817CDD905E095397
      - [+]  JCACFAEF428004CD99731BE344CC9657D
      - [+]  JCF00CA5933E74647A23CDCF3D89C9793

## Data Management – Intermediate and Output

- When the server runs a model, the output and intermediate data should be written to the job directory
- Use the %ScratchWorkspace% inline variable in your paths.
  - %scratchworkspace%\outputfc.shp
- A file geodatabase named “scratch.gdb” is guaranteed to be in the jobs folder created for each job submission
  - %scratchworkspace%\scratch.gdb\outputfc
- You can use in-memory workspace for feature classes and tables
  - In\_memory\outputfc
  - Do not use for output that is drawn by the result map server.

## Data Management – Source Data

- If the data is not large or centralized, it can be useful to package everything in a folder and use relative paths
  - [http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Methods\\_for\\_distributing\\_tools](http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Methods_for_distributing_tools)
- If the data is distributed around the network, use UNC paths when building the model/script.
- Copy SDE Connection files relative to tbx.
  - Will not find “database connections” node.



# Authoring a Model suitable for publishing

- A subset of desktops data types are supported as valid parameters for a geoprocessing service

– Subset determined by light weight clients:

- ArcGIS Explorer
- Web Mapping Applications

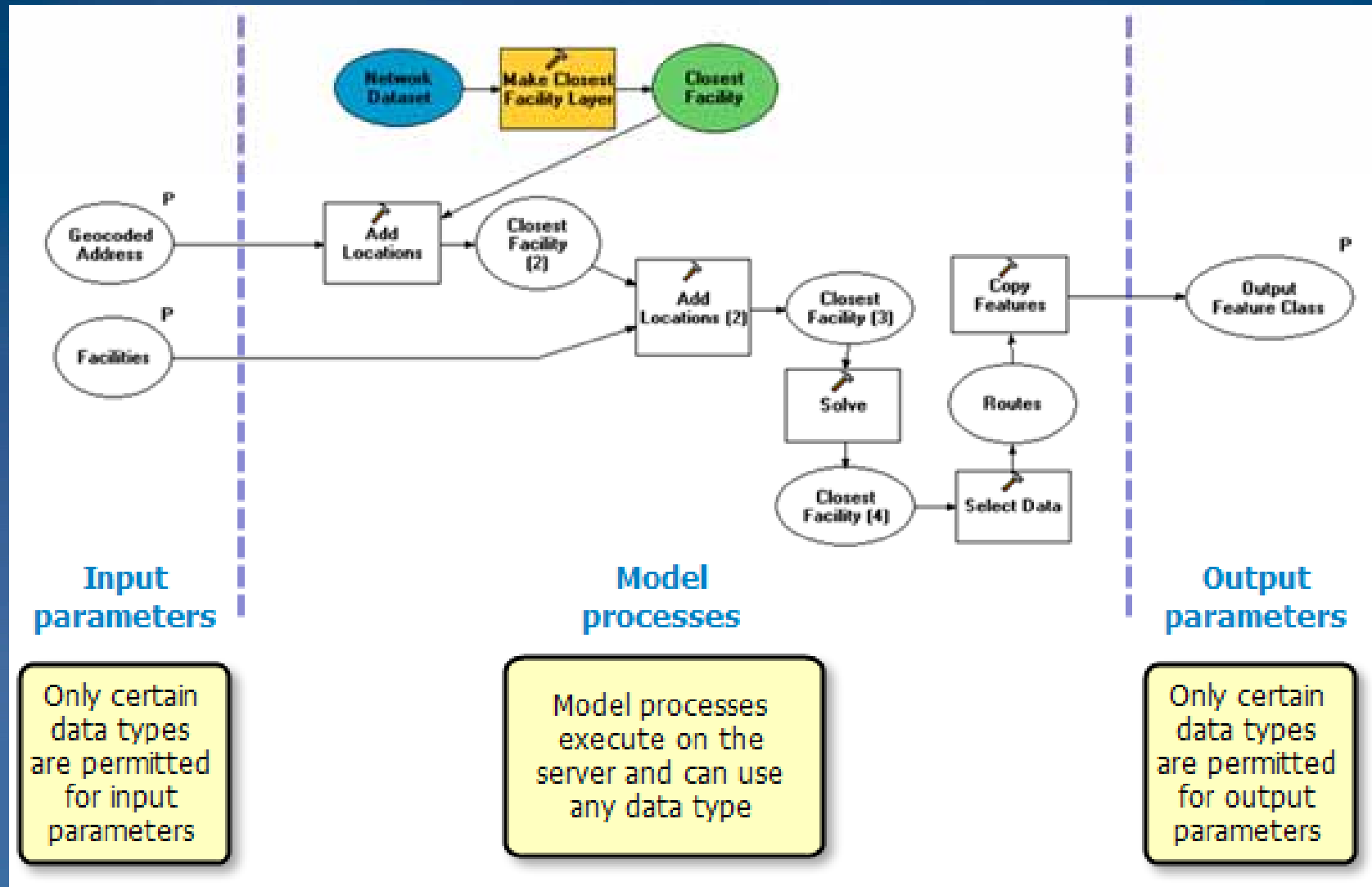
- Publishable tools need to be built accordingly.

- Feature Set
- Record Set
- Raster Dataset
- Feature Layer
- Raster Layer
- Table View
- Layer
- File
- String
- Long
- Double
- Boolean
- Date
- Linear Unit

- Feature Class\*
- Table\*

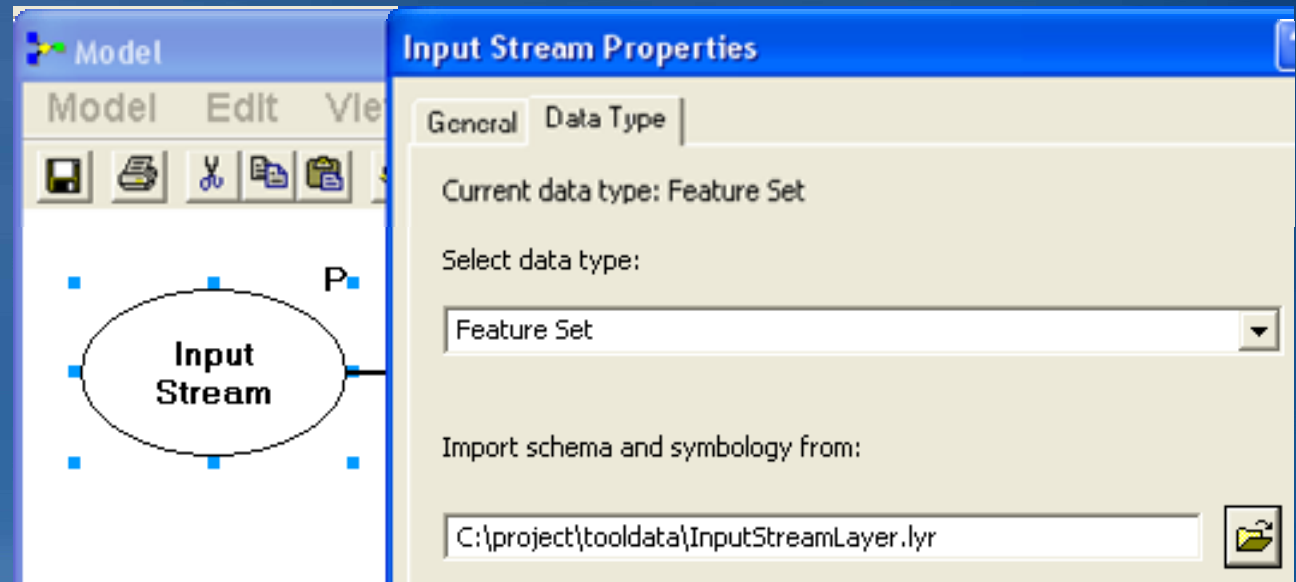
\*output only

# Parameter Types



# Parameter Types – Feature Sets and Record Sets

- Feature Class and Table variables
  - Publish only as output parameters.
- Feature Set and Record Set variables
  - Use Feature/Record Set for interactive input of features or rows
  - Schema defined in properties from existing layer, feature class, or table
    - Fields
    - Field domains
    - Feature type
    - Symbology

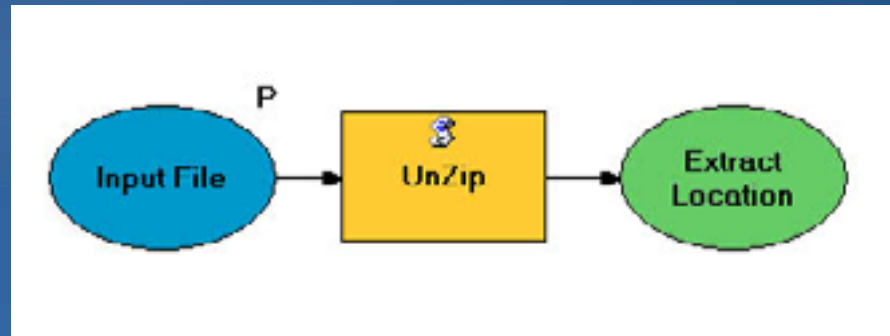


## Parameter Types – Layers

- **Layer parameter type allows clients to select from layers in a map on the server.**
  - Enables the use of datasets on the server.
  - Gives ability to work with “non-publishable” data types.
- **Models that use input layers must be published as Tool Layers in Map Documents**

## Parameter Types - File

- Publishes as input or output parameter
- Can be used to upload zip files to a server.
  - Samples in the help:  
[http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=907&pid=899&topicname=Python\\_scripts\\_to\\_zip\\_and\\_unzip\\_data](http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=907&pid=899&topicname=Python_scripts_to_zip_and_unzip_data)
  - Can send anything up to the server in a zip file and unzip server side.



# Demo: Using a geoprocessing service with a result map service draw

- **Example: Suitability Overlay**
- **Highlights**
  - Use a script for a geoprocessing service
  - Use a map service to draw raster results

# Publishing Considerations

- **Maximum Number of Records**

- This property limits the number of features returned from the server. The default is 500.
- Prevents large amounts of data from being transported across the internet.
- **Can be a gotcha. If your results don't display because of this reason, there should be a message in the tool messages**

- **Number of Instances**

- How many concurrent requests can run

- **Timeout**

- How long before the service automatically kills itself

- **Show Messages**

- Usually only used for debugging initial development

# Tuning: UNC Paths

- Reading and writing data to UNC paths is slower
- If using one server machine avoid UNC where possible
  - Use local path for jobs directory.
  - Use local path to source data if possible.
- If using a distributed server (many machines)
  - Jobs directory must be a UNC path.
  - Can use the **in\_memory** workspace for feature classes and tables.
  - Make a copy of the input data on each SOC machine.
  - **Use LocalJobsDirectory setting!**



## Local Jobs Directory

- **Local Jobs Directory reduces the use of UNC paths.**
- **When specified all intermediate and output data are written to a local job directory**
- **If the service is Asynchronous the local job directory is copied to the main server jobs directory**
- **Only relevant if server is distributed or jobs directory is a UNC location.**

## Setting the Local Jobs Directory (Pre 9.3.1)

- Stop the ArcGIS Server Object Manager (SOM) service
- Manually edit the geoprocessing service's .cfg file and add the <LocalJobsDirectory> tag.
  - .cfg for each service found in <install location>\Server\cfg
- Restart the SOM
- Restart the geoprocessing service.

```
<Properties>
  <Toolbox>C:\project\HydroTools.tbx</Toolbox>
  <MaximumRecords>500</MaximumRecords>
  <LocalJobsDirectory>C:\localjobs</LocalJobsDirectory>
  <JobsDirectory>\\savaii\arcgisserver\arcgisjobs</JobsDirectory>
  <JobsVirtualDirectory>http://vailima/arcgisjobs</JobsVirtualDirectory>
  <ExecutionType>Asynchronous</ExecutionType>
  <OutputDir>\\savaii\arcgisserver\arcgisoutput</OutputDir>
  <VirtualOutputDir>http://vailima/arcgisoutput</VirtualOutputDir>
  <ShowMessages>FALSE</ShowMessages>
</Properties>
```

# Setting the Local Jobs Directory (9.3.1)

ArcGIS Server - Geoprocessing Service Properties

General | Parameters | Capabilities | Pooling | Processes

Execution Type:  Synchronous  Asynchronous

The tools exposed by the Geoprocessing Service are stored in:

A toolbox

Toolbox:

A map

Map Document:

Data Frame:

Result Map Service:

Jobs Directory:

Virtual Jobs Directory:

Output Directory:

Virtual Output Directory:

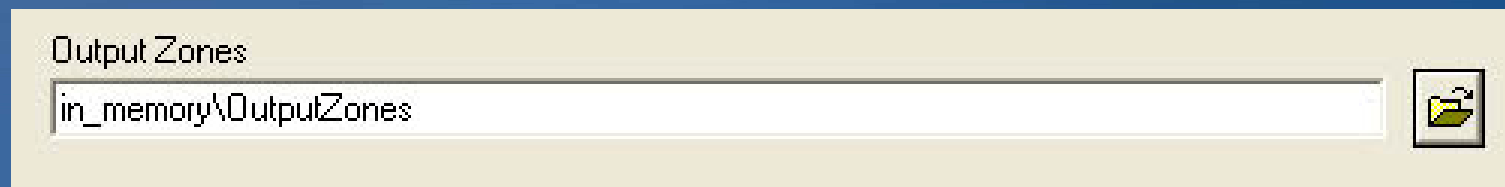
Maximum Number of Records Returned by Server:

Show Messages

Local Jobs Directory

## Tool Optimization – In Memory

- Data can be written out to the “in\_memory” workspace.
  - Only appropriate when overhead of writing to disk is significant portion of the total time it takes to run the model
  - If output is “in\_memory” the client must draw the result
- Use the “in\_memory” keyword to indicate that a dataset will be stored in memory.



# Tool Optimization – Pre-Processing

- **Pre-process any geoprocessing operations that you can and remove them from your model.**
  - **EX: A suitability model may use slope and aspect as criteria. It is not necessary to run slope every time the model is executed. Pre-process slope and aspect.**

# Server Tuning: Large Input Datasets

- Large input record set or raster
  - Increase the web server maximum size setting
  - **C:\inetpub\wwwroot\ArcGIS\Services\web.config** file

```
<?xml version="1.0" encoding="utf-8"?>  
<configuration  
  xmlns="http://schemas.microsoft.com/.NetConfiguration/v2.0">  
<system.web>  
  <httpRuntime maxRequestLength="20000"></httpRuntime>  
</system.web>
```

- [http://webhelp.esri.com/arcgisserver/9.3/dotNet/index.htm#common\\_problems.htm](http://webhelp.esri.com/arcgisserver/9.3/dotNet/index.htm#common_problems.htm)
- <http://support.esri.com/index.cfm?fa=knowledgebase.techarticles.articleShow&d=35971>

# Server Tuning: Timeouts

There are 3 Timeouts to be aware of:

- Client Wait Timeout

C:\inetpub\wwwroot\ArcGIS

```
<system.web>
```

```
  <httpRuntime executionTimeout="600" />
```

```
</system.web>
```

- ArcGIS Server Wait Timeout
- ArcGIS Server Execution Timeout
- If you are using synchronous execution with lots of users, you may need to increase the wait timeouts.

# Troubleshooting

- **Publishing fails if these errors are detected**
  - Check path to the toolbox or map document
  - Check parameter data types
    - Need to rework the tool to limit the parameter types
  - Check of invalid sub-layers in the tool layer
    - Usually means the output data used to create the tool layer has been deleted or not visible to the map server.
- **Most common problems**
  - Tool data paths not to the scratch workspace
  - Tool layer and sub-layer (parameter name) are the same.



# Troubleshooting

- The tool will be run by the ArcGISSOC account which may be different than the login you used to author the tool with.
- Make sure the ArcGISSOC account has access to all the input data.
- Make sure the ArcGISSOC account has access to other software used by the service
  - e.g. When using a custom .Net tool make sure the dll is usable by the ArcGISSOC account
- Debug by logging in as ArcGISSOC
  - Run in ArcMap if ArcGIS Desktop is installed on the server machine
  - Run with a python script if ArcGIS Desktop is not installed

# Troubleshooting Debugging

- Check the log files if you are getting errors
  - Via manager or on disk (**ArcGIS\server\user\log**)
- Debugging
  - Change to Asynchronous
  - Turn Messages On
  - Run and check Job Directory
    - See “Messages.xml”
    - See intermediate and output data

# Performance Logging

- Info:Standard (Info2): Record elapsed time for all service methods.
- Info:Detailed (Info3): Record elapsed time for key sub processes.

2008-02-08T13:40:45	INFO2	<u>Population.GPServer</u>	<u>GPServer.Execute</u>	Binary request received. Request size is 39434 bytes.	mb8	2240		100003
2008-02-08T13:40:45	INFO3	<u>Population.GPServerSync</u>	<u>GPServerSync.Load.summarizePopulation</u>	Load job j5c26f564c33640ffb6adae2df3cba23e message type Binary message size 39435	mb8	2324	0.10153	20023
2008-02-08T13:40:46	INFO3	<u>Population.GPServerSync</u>	<u>GPServerSync.Execute.summarizePopulation</u>	Execute job j5c26f564c33640ffb6adae2df3cba23e	mb8	2324	0.93816	20022
2008-02-08T13:40:46	INFO3	<u>Population.GPServerSync</u>	<u>GPServerSync.Save.summarizePopulation</u>	Save job j5c26f564c33640ffb6adae2df3cba23e message type Binary message size 1964	mb8	2324	0.00549	20024
2008-02-08T13:40:46	INFO2	<u>Population.GPServer</u>	<u>GPServer.Execute</u>	Binary request successfully processed. Response size is 1963 bytes.	mb8	2240	1.06580	100004

## Load and Scaling

- **Rule of Thumb: One Instance per CPU/Core**
  - More CPUs means you can have more instances
  - More instances means more throughput
  - More throughput means more concurrent users
- Without added resources (CPUs), added instances do not necessarily add throughput.
- Average response time
- =  $\left( \frac{(\text{users/socs}) + 1}{2} \right) * \text{execution time}$

# Load and Scaling – Graphs for increasing User Load

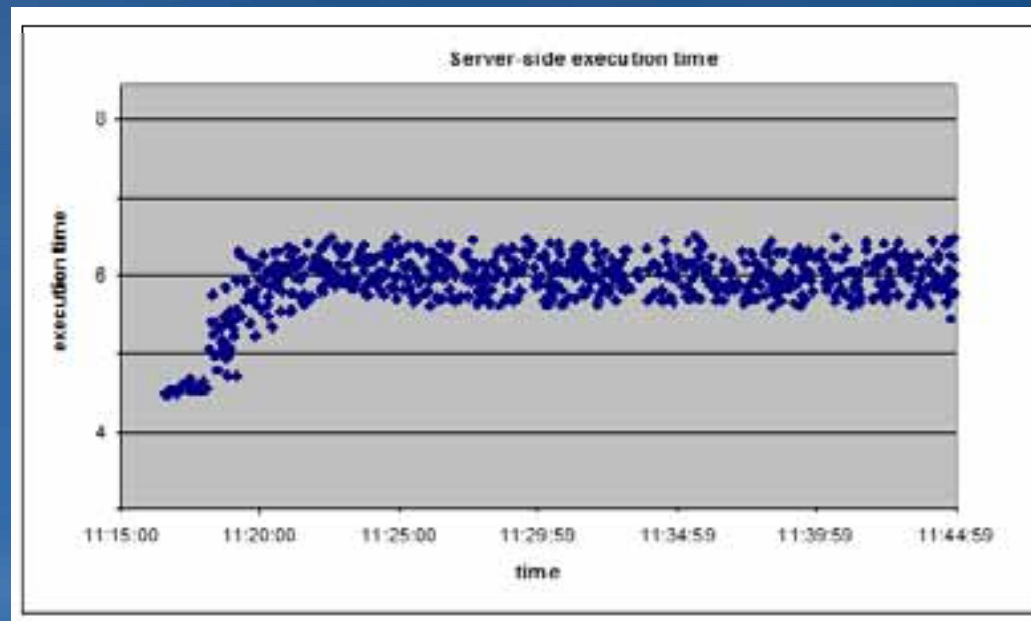


Transactions / sec

Average Transaction Time

User Load

Server-side Execution time



## API Result Options

- **Format**
  - **ESRI**
    - **Record set for features and tables**
    - **Tif for raster**
  - **KML**
    - **KML for features and rasters**
- **Embedded data or URL**
  - **Rasters and files can be returned as URL**
- **Spatial Reference**
  - **If the web client is drawing the results on a map control, need to request the results in the spatial reference of the map control.**

**Questions?**