



High Quality Printing from Web Apps

Julie Powell & Joel McCune

A large, stylized graphic of a blue ribbon with white stars, resembling the American flag, is draped across the bottom of the slide. The ribbon is positioned over a background image of the Washington Monument and the Jefferson Memorial in Washington, DC, with cherry blossoms in the foreground.

Esri Developer Summit
Washington, DC

Overview

- **OOTB**
- **AGO**
- **Custom Layout**
- **Advanced Printing**
- **Use PrintService in Web**
- **Integrate Advanced Print Service in WebApp**

High Quality Printing

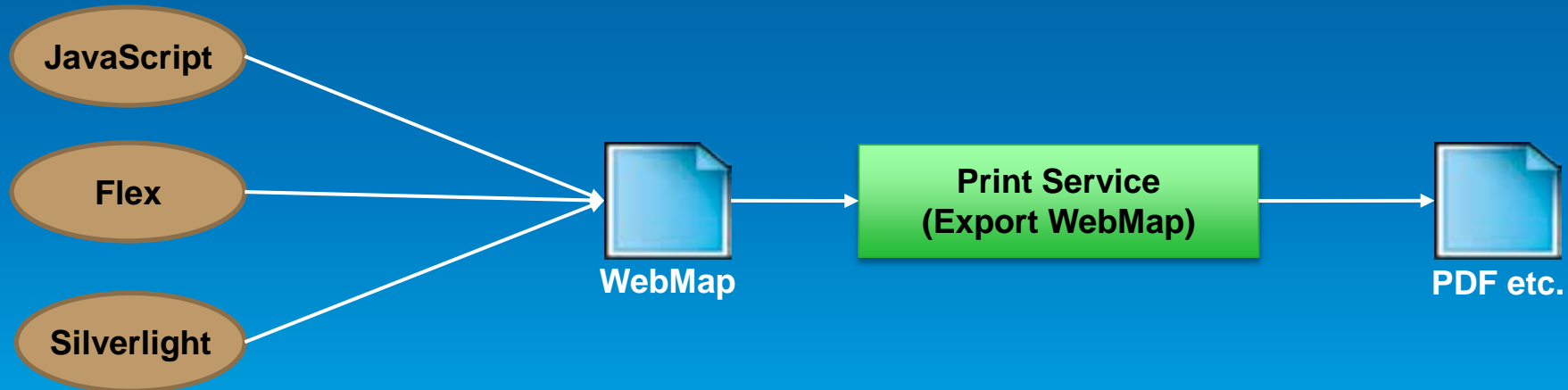
- Top request on ideas.arcgis.com
- Goals:
 - Create a printable document (**PDF** preferably)
 - Have that document printed to '**exact**' scale
 - Leverage **ArcMap layouts**
 - Support **large format plots**
 - Display ALL contents of my Web map
 - **Services, graphics, feature layers, etc.**
 - Be able to modify the map content during the process
 - High DPI
- We created a 3 tiered approach to meet these needs

High Quality Printing

- *Simple printing*
 - Out of the box Geoprocessing Service with ArcGIS Server
 - Small format
 - Synchronous
 - Token based secure services
- *Custom configuration*
 - Ability to supply your own templates
 - Asynchronous if desired
 - Handle HTTP secure services
 - Large format
- *Advanced cases*
 - Supported via ArcPy.Mapping
 - Map/Layer addition, replacement, etc.

Printing service overview

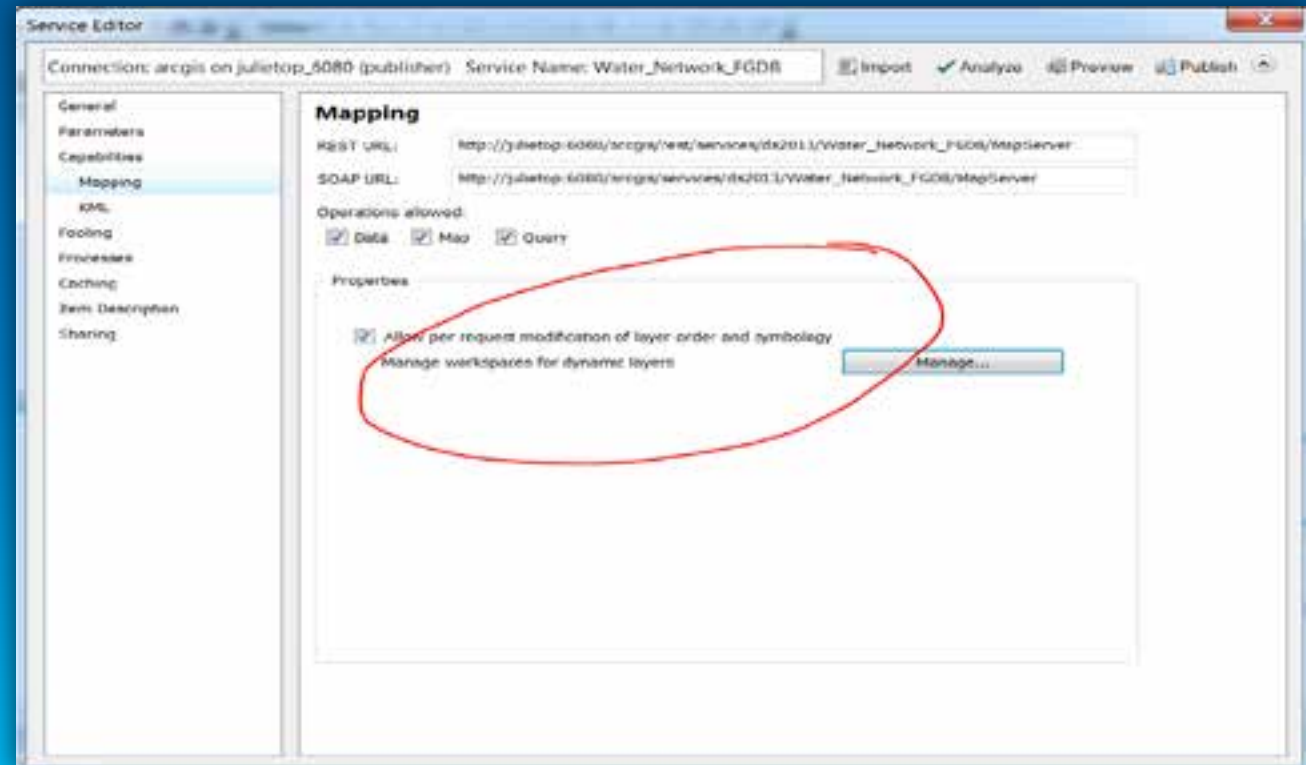
- Print services are powered by WebMaps
 - Consistent format produced by all the WebAPIs
 - JSON based spec for sharing services, graphics, symbols etc.
- Note: The services in the WebMap must be accessible to the Print Service
 - HTTP security is a special case



Cached Services & Printing Large Format

New at 10.1: Dynamic Layers

- Tiles are usually 96 DPI
- When you have the source data:
 - Enable Dynamic Layers
 - Server can generate higher res image



ArcGIS Online Organization & Portal

- Register your own print service
- Print on-premise services behind a firewall

The image shows a screenshot of the ArcGIS Online interface. The top part displays a map titled "Water Network" with a blue line representing a water network. A red circle highlights the "Print" button in the top toolbar. Below the map, there is a configuration page for "Utility Services". The page has a navigation menu on the left with options: General, Home Page, Gallery, Map, Item Details, Groups, Utility Services (selected), and Security. The main content area is titled "Utility Services" and includes a "Printing" section with a printer icon and a text field containing the URL: `http://juliectop:6080/arcgis/rest/services/LAUC/PrintUtility`. The top navigation bar includes "HOME", "GALLERY", "MAP", "GROUPS", and "MY CONTENT", along with a search bar and user information for "Julie Powell".

Demo

Out of Box Print Service

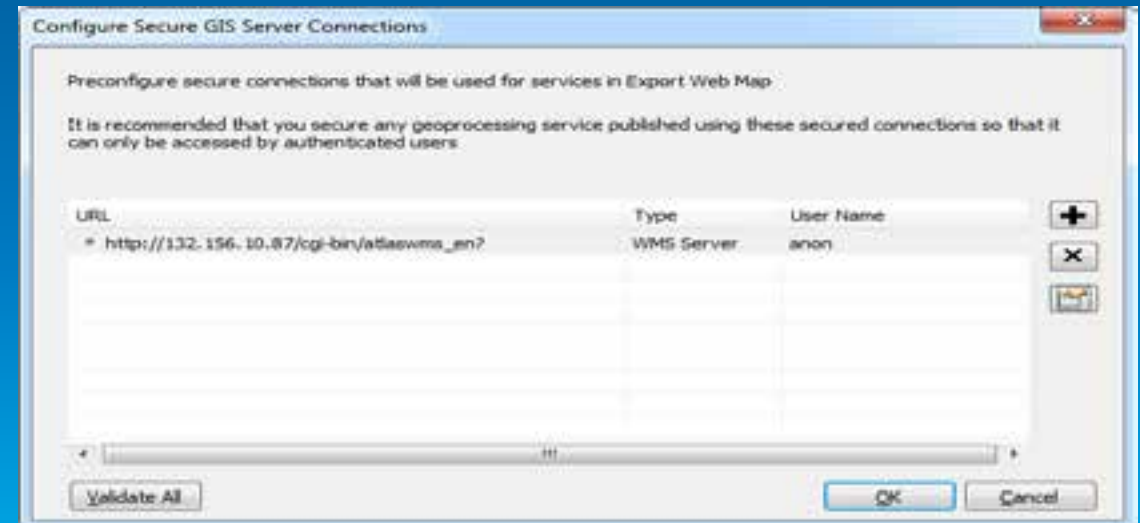
Custom configuration

- **Publish the Export Web Map tool as your own service**
 - Use your own layout templates
 - Choose to make the service asynchronous if desired
 - This choice depends on the size of your prints and the length of time they'll take
 - **Configure service security**
 - Token security handled automatically
 - HTTP security must be configured



Configuring secured services

- HTTP security requires configuration
 - Add service connections for each secure service before publishing
 - “Edit” the tool to edit security configurations
 - **Secure the print service itself if you configure security for services**

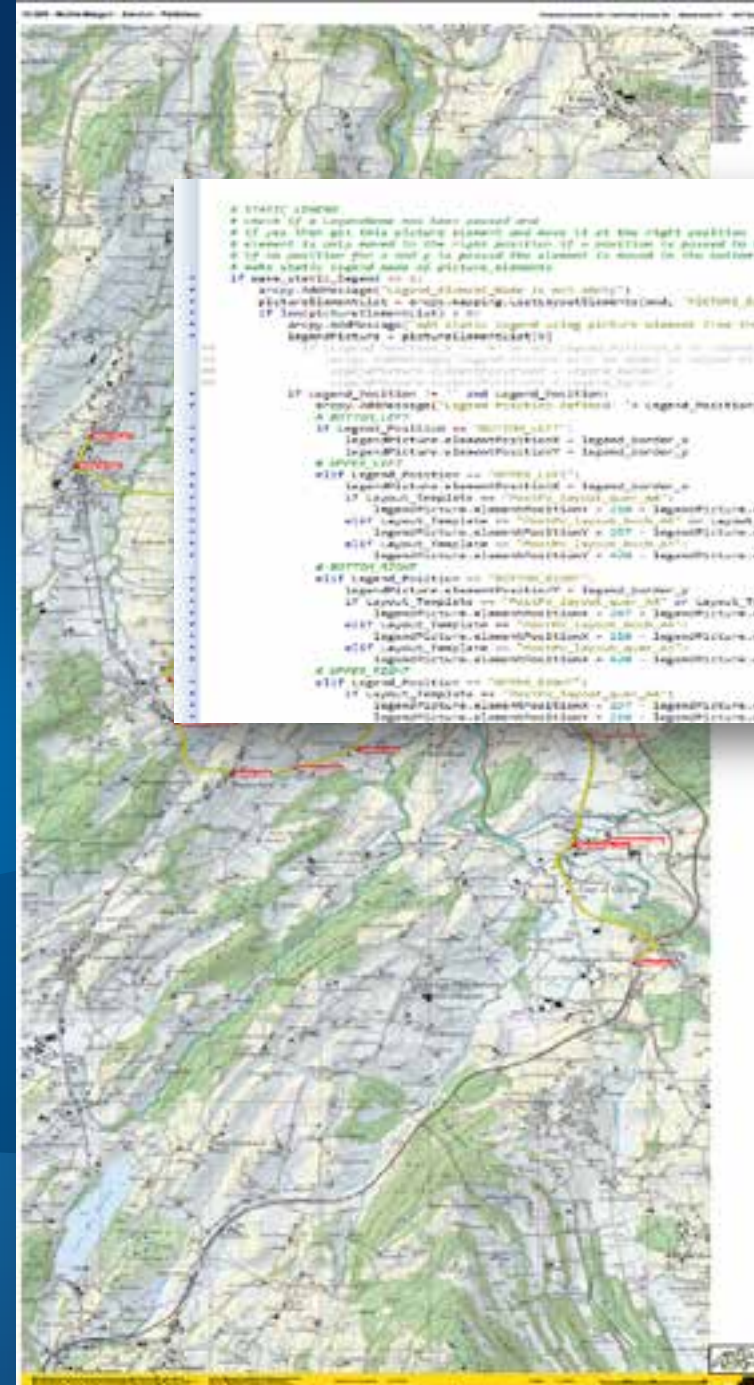


Demo

Printing with custom templates

Advanced Printing

Joel McCune



300 dpi



<http://resources.arcgis.com/en/help/main/10.1/index.html#//0154000005z3000000>

Considerations for Advanced Printing in Web Apps

- **Layout Templates (*.mxd) are the base**
- **Use arcpy.mapping**
- **Use Dynamic Map Services to get high dpi**
- **Leverage advantages of Data Driven Pages**
- **Think out of the box...**

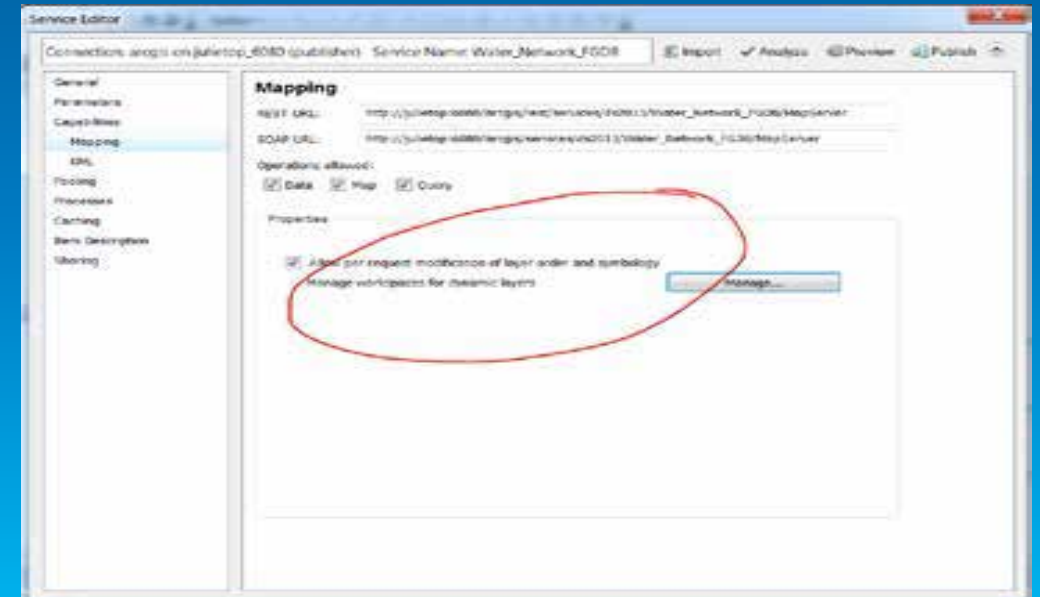
What's to do to get a high resolution output?

- **Screenshots & MapCaches range from 72 to 96 dpi**
- **Exchange basemap layer (cached map service) by using a dynamic map service**
 - **Can be done on the client or serverside**
 - **Use the capabilities of the APIs (client)**
 - **arcpy.mapping (server)**

Cached Services & Printing Large Format

New from 10.1

- **Cached Services & Printing Large Format**
- **New at 10.1: Dynamic Layers**
- **Tiles are usually 96 DPI**
 - **When you have the source data: Enable Dynamic Layers**
 - **Server can generate higher resolution image**



Work with the visibility of layers (client side)

```
if(_printParameters.exportOptions.dpi > 96){  
    _basemapChooserController.switchToPrintQualityBasemaps();  
}
```

```
public function switchToPrintQualityBasemaps():void{  
    for each (var baseMap:BaseMap in _basemapChooserModel.baseMaps){  
        if(baseMap.printLayer != null && baseMap.url != "" && baseMap.layer.visible){  
            baseMap.layer.visible = false;  
            baseMap.printLayer.visible = true;  
            baseMap.printLayer.alpha = baseMap.layer.alpha;  
        }  
    }  
}}
```

layer à TiledMapService

printLayer à DynMapService

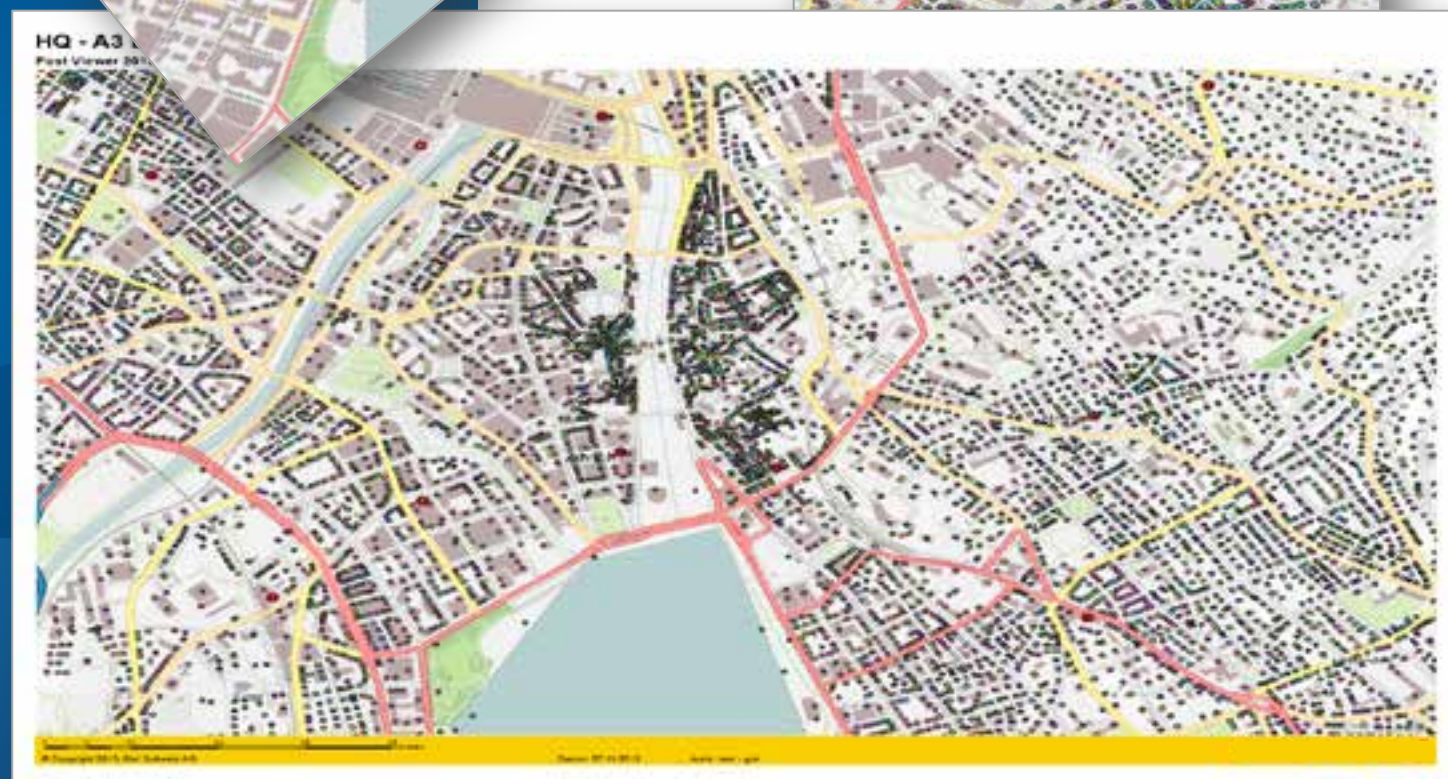
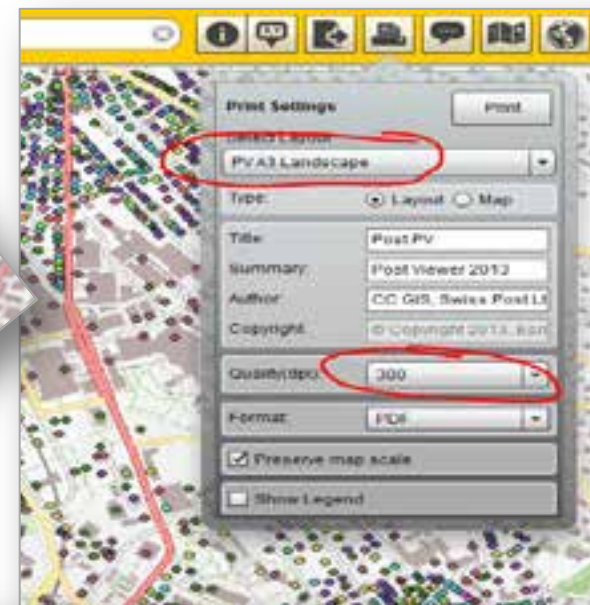
Modify the layers with arcpy.mapping

- **Get web map JSON**
- **Get template map document (mxd)**
- **Create new map document based on web map**
- **Remove cached layers**
- **Export to pdf**
- **Set output file of job**

Demo

Demo

High Resolution Printing



Custom print workflows: arcPy.Mapping

- arcpy.mapping method for converting Web Maps to MapDocuments
- **ConvertWebMapToMapDocument**
(webmap_json, {template_mxd}, {notes_gdb}, {extra_conversion_options})
 - Secure services handled via the *extra conversion options* parameter
 - WMS must be handled using *extra conversion options*
- Once WebMap converted > full capabilities of arcpy.mapping
- Return a printer-friendly output file (PDF, PNG, JPEG, ...)
- See the arcPy.Mapping help for examples, tutorials, code...

... the cookbook for your own print service

- Prepare your Layouts (this is the base, don't forget J)
- Define your custom parameters
- Create a new ArcGIS Toolbox and add you script as Tool
 - Add all parameters to the tool!
- Run the tool in ArcGIS for Desktop
- Publish it as Geoprocessing Service to your server

Demo

Advanced PrintService

```

# STATIC LEGEND
# Check if a LegendName has been passed and
# if yes then get this picture element and move it to the right position
# element is only moved to the right position if a position is passed to the service
# if no position for x and y is passed the element is moved to the bottom left corner
# use static legend made of picture elements
if make_static_legend == 1:
    arcpy.AddMessage("Legend element name is not empty")
    pictureElementList = arcpy.mapping.ListLayoutElements(mxd, "PICTURE_ELEMENT", LegendElementName)
    if len(pictureElementList) == 0:
        arcpy.AddMessage("No static legend using picture element from the layout will be used!")
        legendPicture = pictureElementList[0]
    # Legend Position: "L" for left Legend Position: "R" for right Legend Position: "B" for bottom Legend Position: "T" for top
    arcpy.AddMessage("Legend element will be added to bottom of print service. X = PageLayout_Bottom_Y = 0")
    LegendPictureElementList[0].LegendBorder_x = LegendBorder_x
    LegendPictureElementList[0].LegendBorder_y = LegendBorder_y
    if LegendPosition == "L" and LegendPosition_y == LegendPosition_y:
        arcpy.AddMessage("Legend Position Defined!")

```



```

LegendBorder_y
mxd.Layout_Getter_Alt()
LegendBorder_y
LegendBorder_y
mxd.Layout_Getter_Alt()
LegendBorder_x
LegendBorder_x
LegendBorder_x
LegendBorder_x

```



Demo

Use it!

Building Print Functionality into Your App

```
baseeapChooserController.switchToPrintQualityBaseeaps();
}
if(!exportMap){
    printParameters.layoutTemplate = selectedLayoutTemplate.id;
    printParameters.layoutOptions = new com.esri.ags.tasks.supportClasses.LayoutOptions();
    if(title != null){
        printParameters.layoutOptions.title = title.value;
    }
    if(author != null){
        printParameters.layoutOptions.author = author.value;
    }
    if(copyright != null){
        printParameters.layoutOptions.copyright = copyright.value;
    }
    if(subtitle != null){
        var customTextElements:Object = { "subtitle" : subtitle.value }
        printParameters.layoutOptions.customTextElements = customTextElements;
    }
    var legendOptions:LegendOptions = new LegendOptions();
    printParameters.layoutOptions.legendOptions = legendOptions;
    legendOptions.legendLayers = [];
    if(showLegend){
        //add customParameters to display a static Legend image on the layout
        var customParameters:Object = new Object();
        for(var key:Object in printParametersModel.customParameters){
            if(key != "config"){
                customParameters[key] = printParametersModel.customParameters[key];
            }
        }
        printParameters.customParameters = customParameters;
        //process map.layers and push layers to the legendOptions.legendLayers[]
        for each (var layer:Layer in map.layers)
        {
            var legendLayer:LegendLayer;
            if(layersExcludedInLegend == null || !layersExcludedInLegend.contains(layer))
                // TODO: Now we don't print WMSLayer in the Legend!
                if(!(layer is ArcGISTiledMapServiceLayer) &&
```

execute(printParameters, callback, errback)

Sends a request to the print service resource to create a print page using the information specified in the [printParameters](#) object. Upon completion, the onComplete event is fired and the optional callback function is invoked.

Return Value: [dojo.Deferred](#)

Input Parameters:

Parameter	Optional	Description
<PrintParameters> printParameters	Optional	A PrintParameters object containing printing options.
<Function> callback	Optional	The function to call when the task is completed. The argument passed to the function is the same as the onComplete event.
<Function> errback	Optional	An error object is returned during task execution.

Code Snippet:

```
var params = new esri.tasks.PrintParameters();
params.map = map;

printTask.execute(params, printResult);
```

Use the custom print service

- `esri.tasks.PrintParameters`
 - ...
 - var **extParams** =
 { "Legend_Element_Name": "Legend_Img01",
 "LEGEND_POSITION" : "BOTTOM_RIGHT" }
 - var `printParams` = new `PrintParameters()`;
 - `printParams.map` = `myMap`;
 - `printParams.template` = `myTemplate`;
 - `printParams.extraParameters` = **extParams**;
 - `printTask.execute(printParams, printResultCallback)`;
 - ...

Class hierarchy

esri.tasks.PrintParameters

Constructors

Constructor

















new esri.tasks.PrintParameters()

Properties

Property	Type
extraParameters	Object

customParameters (in Flex)

- A *PrintParameters* object at runtime

  "_printParameters"	com.esri.ags.tasks.supportClasses.PrintParameters (@159cbe41)
 _allLegendLayers	null
 customParameters	Object (@15a5fb81)
 Legend_Element_Name	"01_LG_Land_Karten_3pt_DE"
 Legend_Position	"BOTTOM_LEFT"
 Legend_Position_X	0
 Legend_Position_Y	0
 exportOptions	com.esri.ags.tasks.supportClasses.ExportOptions (@15a78df9)
 format	"PDF"
 layoutOptions	com.esri.ags.tasks.supportClasses.LayoutOptions (@15a78dd1)
 layoutTemplate	"PostPV_layout_quer_A4"
 map	com.esri.ags.Map (@d0b7479)
 outScale	NaN
 outSpatialReference	null
 preserveScale	false

ArcPy Samples

http://resources.arcgis.com/en/help/main/10.2/#/Tutorial_Basic_high_quality_web_map_printing_exporting_using_arcpy_mapping/0154000005z3000000/

http://resources.arcgis.com/en/help/main/10.2/#/Tutorial_Advanced_high_quality_web_map_printing_exporting_using_arcpy_mapping/0154000005z2000000/



Understanding our world.