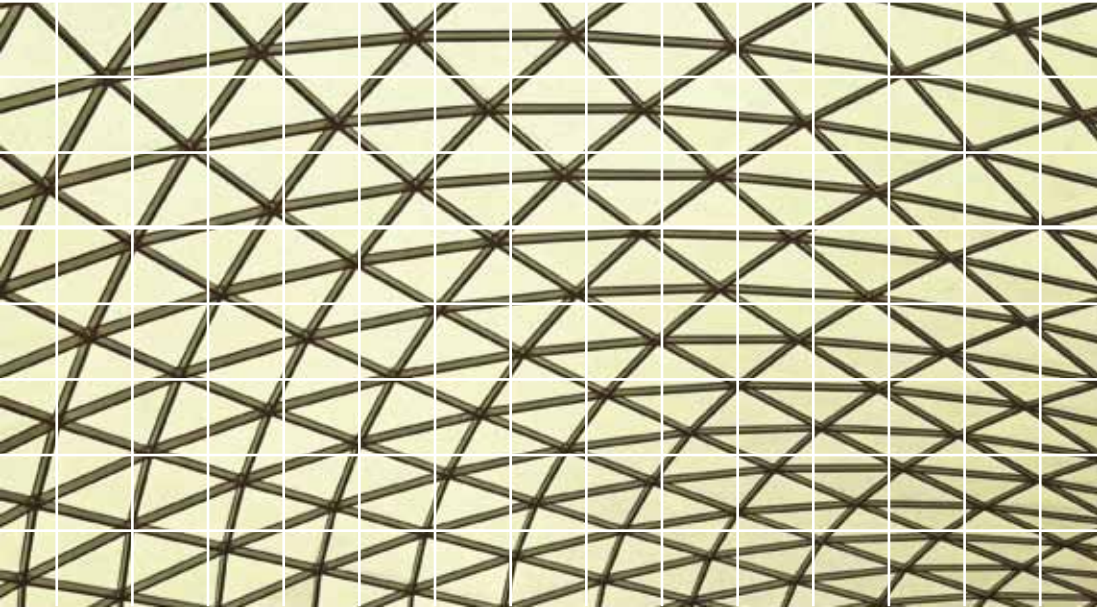


Web Client APIs for

# ArcGIS Server 9.3.1



August 2009



Innovation. Information. Interaction.

# Web Client APIs for ArcGIS Server 9.3.1

The ArcGIS platform enables you to focus on solving business problems with APIs and tools designed to deliver GIS functionality to your applications. Software developers who use the ArcGIS Server platform can: work with a variety of programming languages, standards, and tools; deploy GIS applications on a variety of platforms; access GIS Web services using REST or SOAP; and take advantage of the resources that ESRI provides to help software developers design and build useful GIS applications.

## Web ADF (.Net / Java)

### Technology Overview

ADF applications are web browser based applications where the functionality is delivered from software deployed within an application server, either ASP.NET or a Java based equivalent.

The Web ADFs can access ArcGIS Server either directly using local connections or via web services. Local connections allow access to the most complex functionality such as printing and editing, whereas web services provide more limited functionality that is more scalable.

### Development Complexity

Development using the Web ADFs is generally based on the use of components supplied by ESRI. This provides quick access to complex functionality within the browser with little development effort and is supported by the integration within Integrated Development Environments (IDEs), which allow for the whole development lifecycle to be speeded up.

Development is either in Java or .NET and a good understanding of the specific language is required, more complex development will require an understanding of platform AJAX technologies. Where possible the underlying framework should be leveraged rather than writing any new similar code

The Web ADF can also be deployed through the Web Mapping Application (WMA), which provides a ready-to-go application with out-of-the-box functionality delivered via a map-centric GIS client. The Web Mapping Application can also be deployed through the use of a wizard without the need for any development. Much application functionality can be accessed in this manner through the task framework.

### Application Functionality

The Web ADF can access all of ArcGIS Server down to the use of fine grained ArcObjects. This allows for the full range of ArcGIS functionality to be available to developers.

Lots of tasks and tools are available either from ESRI or developed in-house. These are discussed in the subsequent sections.

### Application Scalability

Due to their complicated nature Web ADF applications can be some of the most challenging applications to scale, especially if they make a large use of local connections to ArcGIS Server. Any applications that use this set of technologies should be thoroughly tested for performance and scalability.

### Decision Points

- More heavyweight application which requires deployment to an application server
- Doesn't require a browser plug-in
- Provides largest range of out-of-the-box functionality
- Only web API that can access fine grained ArcObjects
- Needs .NET or Java skills, JavaScript knowledge and AJAX understanding
- IDE Integrations. .NET version requires Visual Studio. Java version requires Eclipse IDE or equivalent
- Most suited to an intranet



# JavaScript API

## Technology Overview

The JavaScript API is a lightweight client technology fully hosted within a web browser. It uses the open source DOJO framework to provide connectivity to ArcGIS Server via the REST API using AJAX. Client applications are generally deployed as static HTML and JavaScript files and accessed via a web server.

## Development Complexity

The use of JavaScript and the DOJO framework requires a developer to have a good understanding of both technologies as well as the ArcGIS Server REST API. Client based functionality is limited to what can be done with JavaScript and HTML, but this means that users will have maximum access to the widest range of browsers.

The Dojo framework and the JavaScript API use Dijits to provide the building blocks for functions. These should be utilised wherever possible when building applications as they provide the greatest level of reuse. The Dojo framework itself should be leveraged wherever possible in order to minimise the issues with cross browser support and accessibility.

Development with the JavaScript API can be performed in simple tools such as notepad, but also using IDEs such as Eclipse and Visual Studio.

## Application Functionality

Out of the box the JavaScript API provides a limited amount of functionality, but out-of-the-box functions are supplemented by numerous samples available in the API resources. Dijits are provided for the main map and navigation controls, which can be used to access a variety of data types.

The REST API itself is a subset of the types of functions that can be supported by ArcObjects. Extending applications is possible through the creation of new services in a restful manner, using WCF or similar technologies. These can be integrated into the standard application through the creation of new Dijits, which can then be reused throughout the organisation.

## Application Scalability

Standard JavaScript API applications scale well as they make specific use of the REST API. Considerations should be made on the complexity of the map services that are used and any extra functionality provided outside of the core dijits, especially if it makes use of the geoprocessing framework or fine grained ArcObjects.

## Decision Points

- Lightweight application dependent on REST API
- Doesn't require a browser plug-in
- Provides small range of out of the box functionality, backed up numerous samples
- Quick to start development
- Requires knowledge of JavaScript and DOJO

ESRI is continually enhancing its Web APIs and each new release focuses on making improvements that meet the evolving needs of the web developer. To ensure that you have the most current version of this document please refer to the ESRI (UK) website.

## FLEX API

### Technology Overview

The Flex API enables the creation of rich application functionality based on the ADOBE Flex language. Applications can be hosted within a web browser using the standard ADOBE Flash plug-in or on the desktop through the ADOBE AIR deployment mechanism.

### Development Complexity

Developers should understand the ADOBE Flex development, including MXML, as well as web client development and interactions with ArcGIS Server via the REST API. FLEX also uses ActionScript to provide programmatic functionality, which is a version of JavaScript.

FLEX uses Widgets to provide functionality as components and, as with the JavaScript API, the number of standard map based components is limited, but there are a large number of standard FLEX components that can be reused to build applications.

### Application Functionality

Applications are generally lightweight in terms of the functionality they can provide, although the rich application nature of FLEX can deliver an improved user experience. This means greater care needs to be paid to the design. The out-of-the-box tools may be limited, but are backed up by an abundance of samples to allow development of new tools using the API.

As it is also a client based application, the FLEX API can allow for applications to store more features within the browser than a pure HTML/JavaScript application can, which enables more intuitive editing and dynamic analysis applications to be developed.

### Application Scalability

The FLEX API is based on the REST API and is therefore constrained by what that API can do. Generally, for properly designed map documents, the performance of a FLEX application should scale greater than a Web ADF application. For more complicated queries or the introduction of more complicated services external to the REST API, performance and scalability should be tested.

### Decision Points

- Lightweight application dependent on REST API
- Requires the Flash browser plug-in, which is widely adopted
- Requires FLEX and ActionScript knowledge
- IDE Integration with Flex Builder 3
- Provides out of the box functionality, backed up numerous samples

# Silverlight/ WPF API

## Technology Overview

The ArcGIS API for Microsoft Silverlight™/WPF™ enables the creation of rich internet and desktop applications. The API is built on the Microsoft Silverlight/WPF platform. The Microsoft Silverlight platform includes a lightweight version of the .NET Framework CLR (CoreCLR) and the Silverlight runtime - all hosted via a browser plug-in.

## Development Complexity

Developers will need to understand: XAML (eXtensible Application Markup Language), which is the foundation of the Silverlight presentation capability; .NET for developing code behind; and how to interact with the ArcGIS Server REST API. Although XAML is likely to be a new skill for developers to learn, the access to .NET will facilitate development with this API for existing .NET developers.

The Silverlight API utilises User Controls (often termed widgets) to provide functionality as components. There are slightly more controls available for the Silverlight API than in the Flex or JavaScript APIs, but not significantly more. However, there are a large number of standard .NET controls available (which developers may already be familiar with) and also a moderate amount of Silverlight specific controls.

## Application Functionality

Silverlight provides the foundation for Rich Internet Applications so can provide a great user experience when implemented thoughtfully. Applications are typically quite small in terms of functionality, which can contribute to a better user experience through ease of use. Although light on out-of-the-box tools, there are numerous samples available in the API resources to support developers.

Silverlight is a client based application and due to it being an internet browser plug-in (like FLEX), it allows applications to store more features within a browser than a typical HTML/JavaScript application. This facilitates the development of more on-the-fly analytical functions and map manipulation tasks.

## Application Scalability

Ultimately the Silverlight API is dependent on the REST API so will have the same scalability and also the same constraints that the Flex and JavaScript APIs have. Care should always be taken when designing web applications, especially when extending functionality to consume external custom data sources as there is then a reliance on the servers hosting these services. As with the other client APIs map services, any Geoprocessing functionality used should be optimised for use on the web.

## Decision Points

- Lightweight application dependent on REST API
- Requires the Silverlight browser plug-in
- Requires XAML and .NET knowledge
- IDE Integration with Visual Studio, easier to understand by .NET developers
- Optionally requires Microsoft Expression Blend for design purposes
- Provides out of the box functionality, backed up numerous samples

## Out of the Box Tools

|                        | JavaScript | Flex | Silverlight | Web ADF      | WMA   |
|------------------------|------------|------|-------------|--------------|-------|
| Identify               | X(2)       | X(2) | X(2)        | X            | X     |
| Navigation             | X          | X    | X           | X            | X     |
| Zoom Level             | X          | X    | X           | X            | X     |
| Back Extent            | X          | X(2) | X           | X            | X     |
| Forward Extent         | X          | X(2) | X           | X            | X     |
| Scale bar              |            | X    | X           | X            | X     |
| Overview Map           |            |      | X           | X            | X     |
| Magnifier Window       |            |      | X           | X            | X     |
| Map Copyright Text     |            |      |             | X            | X     |
| Table of Contents      |            |      |             | X            | X     |
| Query Attributes Task  | X(2)       | X(2) | X(2)        | X            | X     |
| Search Attributes Task | X(2)       | X(2) | X(2)        | X            | X     |
| Find Address Task      | X(2)       | X(2) | X(2)        | X            | X     |
| Simple Print Task      |            | X(2) |             | X            | X     |
| Geoprocessing Task     | X(2)       | X(2) | X(2)        | X            | X     |
| Find Place Task        | X(2)       | X(2) | X(2)        | X            | X     |
| Editor Task            |            |      |             | X            | X     |
| Measure tool           | X(2)       | X(2) | X(2)        | X            | X     |
| Geodatabase Editing    |            |      |             | X            | X     |
| Select by Geometry     | X(2)       | X(2) | X(2)        | X (1) / X(2) | X (1) |
| Buffer                 | X(2)       | X(2) | X(2)        | X(2)         |       |
| Draw Graphics          | X(2)       | X(2) | X(2)        | X(2)         |       |

(1) Through geo-processing. (2) Sample provided.

## Developer Functionality

|  | JavaScript | Flex | Silverlight | Web ADF | WMA |
|--|------------|------|-------------|---------|-----|
| Access Fine Grained ArcObjects                   |            |      |             | X       |     |
| Change Layer Definitions on map service layers   | X          | X    | X           | X       |     |
| MapTips (point, polyline, polygon)               | X          | X    | X           | X       |     |
| IDE Integration and controls                     |            | X    | X           | X       |     |
| Custom Data Sources                              | X          | X    | X           | X       |     |
| Page Layout Control (Prints ArcMap-like layouts) |            |      |             | X       |     |
| Add/Remove/Reorder map service layers            |            |      |             | X       |     |
| Change layer symbology on a per request basis    |            |      |             | X       |     |

## Supported Data Sources

|                | JavaScript | FLEX | Silverlight | WebADF | WMA |
|----------------|------------|------|-------------|--------|-----|
| Map Service    | X(1)       | X(1) | X(1)        | X      | X   |
| ArcIMS         |            | X    |             | X      | X   |
| WMS            | X          | X    | X           | X      | X   |
| Image Service  | X          | X    | X           | X      | X   |
| WFS            |            |      |             |        |     |
| WCS            |            |      |             |        |     |
| Graphics layer | X          | X    | X           | X      |     |
| Google Maps    | X          |      |             |        |     |
| Virtual Earth  | X          | X    | X           | X      | X   |

(1) Does not support custom projections