

ArcWebSM for Developers

Spatial Data and GIS Services Delivered Over the Internet

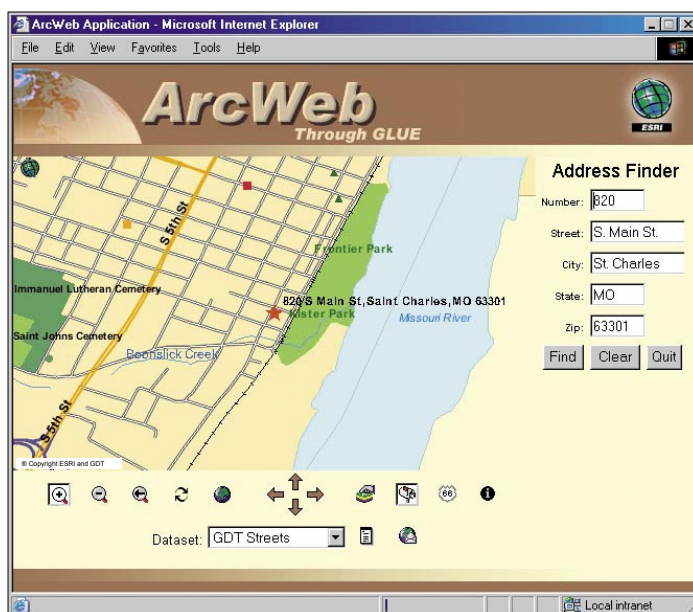
ArcWebSM for Developers

Spatial Data and GIS Services Delivered Over the Internet

Wouldn't your Web site be more effective if you were able to provide interactive maps showing driving directions, points of interest (POI), and other geographic-based content?

Perhaps you have thought about buying a geographic information system (GIS), but decided you do not have the time or the resources to implement one right now. Or you may have found that the task of maintaining the data you need is too expensive, time-consuming, or requires too much disk space.

From simple mapping to more complex tasks such as multipoint driving directions, ESRI's ArcWebSM for Developers makes developing lightweight, Web-enabled applications simple, fast, and cost-effective.



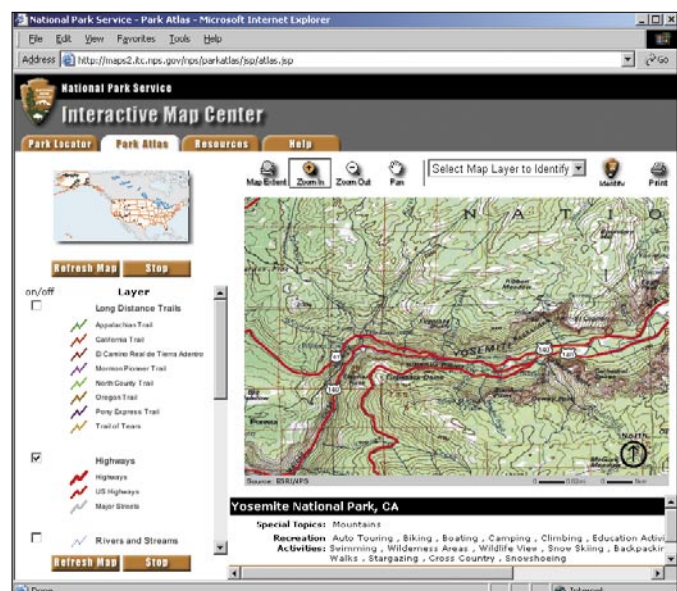
ArcWeb for Developers consists of the following services:

- **Map Image**—Generate a dynamic map of a specific location.
- **Place Finder**—Determine the location of a place (e.g., city, state) in the world.
- **Address Finder**—Determine the location of a street address.
- **Route Finder**—Generate a route with driving directions for multiple stops along the way.
- **Query**—Determine the characteristics (such as population or median income) of a location.
- **POI Manager**—Upload or input custom points of interest (such as store locations) for use with the ArcWeb for Developers services.
- **Proximity**—Find points of interest that are within a specified distance of a location.

What Is ArcWeb for Developers?

ArcWeb for Developers gives you a way to include geographic content and capabilities in your Web applications without having to host the data or do any of the GIS application development yourself. With ArcWeb for Developers, you can save significant development time, expense, and computer resources.

ArcWeb for Developers is part of ESRI's family of hosted GIS Web services, known as ArcWeb Services. It provides an easy and cost-effective way to access up-to-date spatial data and services. Developers can integrate these services into any application, leveraging their own data with access to continually updated information. Because ArcWeb for Developers uses Simple Object Access Protocol (SOAP) to communicate, it can be used independent of device, programming language, or operating system.



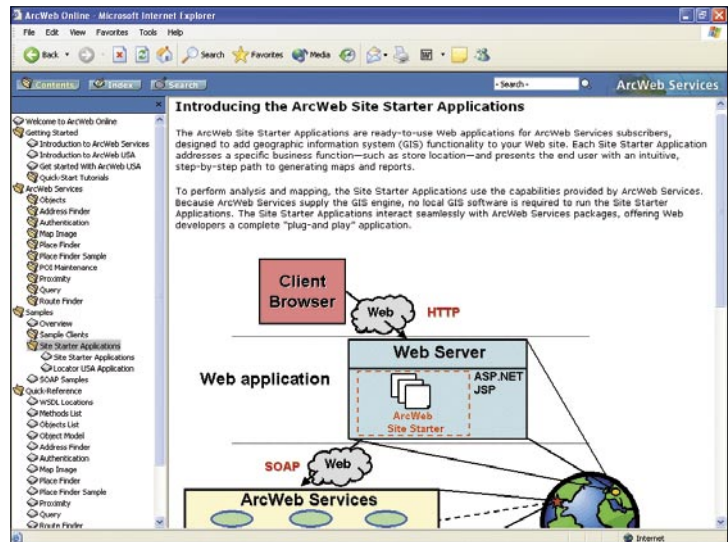
Easy Access to Current Data

ArcWeb for Developers gives you access to a rich collection of data that is always up-to-date including street, census population, topographic, and flood data. Data is currently offered from leading data providers such as Geographic Data Technology (GDT) and Tele Atlas, with new data sources being added all the time.

Service and Support

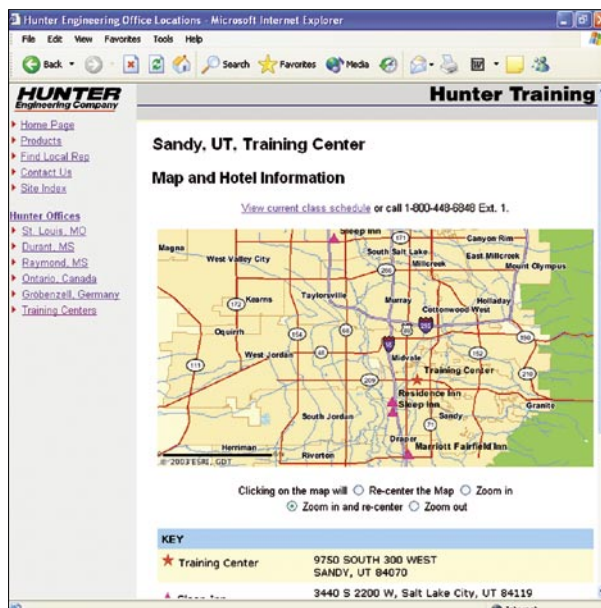
A key advantage of subscribing to ArcWeb for Developers is the service and support that ESRI provides. For instance, if you want to incorporate your points of interest (such as store locations) into your Web application, all you need to do is upload these locations to ESRI. We will then maintain and geocode them for you. Our hosting services are reliable and available 24/7. To ensure consistent, dependable uptime, ESRI maintains a redundant site at a remote location.

Also, ESRI offers technical support from our Developer Support Group to subscribers of ArcWeb for Developers services. Additional assistance is available through in-depth, online help including samples, tutorials, discussion forums, and much more.



Standards-Based for Ease of Use

ArcWeb for Developers is deployed through standard Web protocols including hypertext transfer protocol (HTTP) and XML. Also, ArcWeb for Developers uses XML-based SOAP to communicate, making it compatible with the majority of Web service frameworks available today.



"As a result of using ArcWeb for Developers, the training center area on our Web site is almost maintenance free. When training centers are added or removed, new maps are generated automatically. I don't have to do anything!"

Sheryl Daugherty, Webmaster
Hunter Engineering Company

Learn More

To learn more about ArcWeb for Developers
or to sign up for a free evaluation, visit

www.esri.com/arcwebdev



For more than 30 years ESRI has been helping people manage and analyze geographic information. ESRI offers a framework for implementing GIS technology in any organization with a seamless link from personal GIS on the desktop to enterprisewide GIS client/server and data management systems. ESRI GIS solutions are flexible and can be customized to meet the needs of our users. ESRI is a full-service GIS company, ready to help you begin, grow, and build success with GIS.

Corporate

ESRI
380 New York Street
Redlands, California
92373-8100, USA
Telephone: 909-793-2853
Fax: 909-793-5953

For more information
on ESRI, call

1-800-447-9778

(1-800-GIS-XPRT)

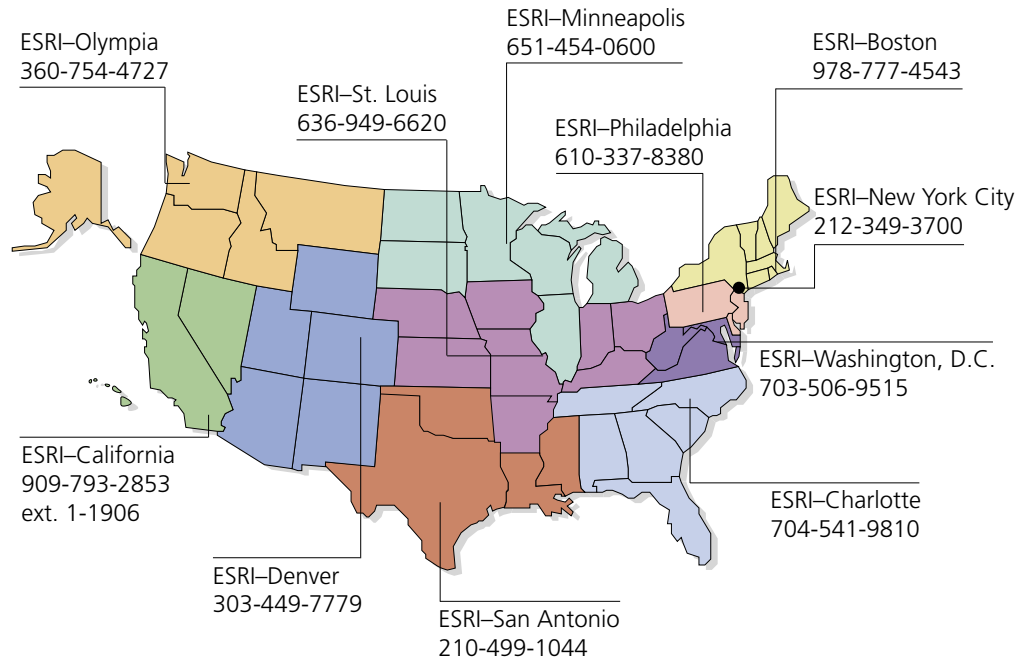
or contact an ESRI reseller near you.

Send e-mail inquiries to
info@esri.com

Visit ESRI's Web page at
www.esri.com

Outside the United States,
contact your local ESRI distributor.
For the number of your distributor,
call ESRI at 909-793-2853,
ext. 1235,
or visit our Web site at
www.esri.com/international

Regional Offices



International Offices

Australia 613-9867-0447	France 33-1-46-23-6060	Japan 81-3-3794-6681	Singapore 65-6742-8622
Belgium/Luxembourg 32-2-460-7480	Germany/Switzerland 49-8166-677-0 41-1-360-2460	Korea 82-2-571-3161	Spain 34-91-559-4375
Bulgaria 359-2-964-0850	Hungary 361-428-8040	Netherlands 31-10-217-0700	Sweden 46-23-755-400
Canada 416-441-6035	India 911-11-2620-3800	Poland 48-22-326-7300	Thailand 66-2-678-0707
China (Beijing) 86-10-6554-1618	Indonesia/Malaysia 62-21-570-7685 603-7874-9930	Portugal 351-2-1-781-6640	United Kingdom 44-1296-745-500
China (Hong Kong) 852-2730-6883	Italy 39-06-406-96-1	Romania 40-21-231-13-81	Venezuela 58-212-285-9394



Place ESRI business partner or distributor address here.

Copyright © 2003 ESRI. All rights reserved. ESRI, the ESRI globe logo, ArcWeb, @esri.com, and www.esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners. The Ordnance Survey data for Great Britain contained in this product is Crown Copyright material and Ordnance Survey is the copyright owner of Ordnance Survey Data.



No. GS-35F-5086H

ArcWebSM for Developers 2 Object Model

Key

Service Name
Interface key
◀ Request Property
■ Response Property
◀ Request/Response Property
◀ Method

Find Services

Place Finder Sample Service
PlaceFinderSample
◀ findPlace(placeName: string, placeFinderOptions: PlaceFinderOptions): LocationInfo
◀ getCountryCodes: Key[Value]
◀ getTypes: Key[Value]
◀ getVersion: string

Place Finder Service
PlaceFinder
◀ findPlace(placeName: string, placeFinderOptions: PlaceFinderOptions, token: string): LocationInfo
◀ getCountryCodes(dataSource: string, token: string): Key[Value]
◀ getTypes(dataSource: string, token: string): Key[Value]
◀ getVersion: string

Address Finder Service
AddressFinder
◀ findAddress(address: Address, addressFinderOptions: AddressFinderOptions, token: string): LocationInfo
◀ getAddress(point: Point, token: string): Address
◀ getCounties(dataSource: string, token: string): string
◀ getVersion: string

Proximity Service
Proximity
◀ findNearest(line: Geometry, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findNearest(point: Point, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findWithin(line: Geometry, radius: double, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findWithin(point: Point, radius: double, proximityOptions: ProximityOptions, token: string): ResultSet
◀ getAvailableFieldNames(proximityOptions: ProximityOptions): FieldDesc[]
◀ getVersion: string

Common
Location
■ description1: string
■ description2: string
■ locationExtent: Envelope
■ matchType: string
■ point: Point
■ score: double
■ type: string

LocationInfo
■ candidates: Location[]
■ errorCode: string
■ hasMore: boolean
■ matchType: string

PlaceFinderOptions
■ count: integer
■ dataSource: string
■ filterCountry: string
■ filterEnvelope: Envelope
■ filterType: string
■ searchType: string
■ startIndex: integer

Query Service
Query
◀ envelopeQuery(envelope: Envelope, queryOptions: QueryOptions, token: string): ResultSet
◀ geometryQuery(geometry: Geometry, queryOptions: QueryOptions, token: string): ResultSet
◀ getAvailableFieldNames(queryOptions: QueryOptions): FieldDesc[]
◀ getVersion: string
◀ pointQuery(point: Point, queryOptions: QueryOptions, token: string): ResultSet

QueryOptions
■ dataSource: string
■ geographicalLevel: string
■ returnFieldsList: string[]

POI Manager Service
PoiManager
◀ addLocation(poi: Poi, geocode: boolean, keepFormatting: boolean, poiManagerOptions: PoiManagerOptions, token: string): integer
◀ deleteAll(token: string): integer
◀ deleteLocations(ids: string[], token: string): integer
◀ findLocations(queryParams: QueryParams, token: string): PoiResultSet
◀ getPoiTypes(token: string): string[]
◀ getVersion: string
◀ updateLocation(id: string, poi: Poi, regeocode: boolean, poiManagerOptions: PoiManagerOptions, token: string): integer

Poi
■ city: string
■ country: string
■ desc: string[]
■ id: string
■ matchType: string
■ placeName: string
■ point: Point
■ state: string
■ street1: string
■ street2: string
■ type: string
■ zip: string

PoiManagerOptions
■ dataSource: string

PoiResultSet
■ pois: Poi[]
■ totalCount: long

QueryParams
■ city: string
■ count: integer
■ keyword: string[]
■ placeName: string
■ startIndex: string
■ state: string
■ status: string
■ type: string
■ zip: string

Common
FieldDesc
■ fieldLength: integer
■ fieldName: string
■ fieldPrecision: integer
■ fieldShortDesc: string
■ fieldType: string

ResultSet
■ fields: FieldDesc[]
■ rows: RowData[]
■ totalCount: integer

RowData
■ fieldValues: string[]

Map Services

MapImage Service
MapImage
◀ convertMapCoordToPixelCoord(viewExtent: Envelope, mapImageSize: MapImageSize, mapCoord: Point): PixelCoord
◀ convertPixelCoordToMapCoord(viewExtent: Envelope, mapImageSize: MapImageSize, mapClickPoint: PixelCoord): Point
◀ getBestMap(mapImageOptions: MapImageOptions, token: string): MapImageInfo
◀ getBestMapEnvelope(mapImageOptions: MapImageOptions): Envelope
◀ getLayerInfo(dataSource: string): Layer[]
◀ getLayerInfoForExtent(dataSource: string, mapImageSize: MapImageSize, mapExtent: Envelope): Layer[]
◀ getMap(mapExtent: Envelope, mapImageOptions: MapImageOptions, token: string): MapImageInfo
◀ getMarkerNames(iconDataSource: string): string[]
◀ getMarkerNamesForExtent(iconDataSource: string, token: string): string[]
◀ getThematicFields(thematicDataSource: string): string[]
◀ getThematicGeographiesForExtent(thematicDataSource: string, mapImageSize: MapImageSize, mapExtent: Envelope): string[]
◀ getThematicMap(mapExtent: Envelope, mapImageOptions: MapImageOptions, thematicField: string, thematicOptions: ThematicOptions, token: string): MapImageInfo
◀ getThematicMapForExtent(envelope: Envelope, mapImageOptions: MapImageOptions, thematicOptions: ThematicOptions, token: string): MapImageInfo
◀ getVersion: string

Common
CircleDescription
■ boundaryColor: string
■ center: Point
■ radius: double
■ radiusUnits: string

LabelDescription
■ antiAliasing: string
■ font: string
■ fontColor: string
■ fontSize: integer
■ fontStyle: string
■ halignment: string
■ interval: integer
■ outlineColor: string
■ transparency: double
■ valignment: string

Layer
■ layerName: string
■ visibility: string

LineDescription
■ color: string
■ geometry: Geometry
■ thickness: integer
■ transparency: double

MapImageInfo
■ mapExtent: Envelope
■ legendUrl: string
■ mapUrl: string

MapImageOptions
■ backgroundColor: string
■ circles: CircleDescription[]
■ dataSource: string
■ displayLayers: Layer[]
■ drawScaleBar: boolean
■ lines: LineDescription[]
■ mapImageFormat: string
■ mapImageSize: MapImageSize
■ markers: MarkerDescription[]
■ outputCoordSys: CoordinateSystem
■ polygons: PolygonDescription[]
■ returnLegend: boolean
■ scaleBarPixelLocation: PixelCoord

MapImageSize
■ height: integer
■ width: integer

MarkerDescription
■ color: string
■ iconDataSource: string
■ label: string
■ labelDescription: LabelDescription
■ location: Point
■ name: string

PixelCoord
■ x: integer
■ y: integer

PolygonDescription
■ boundaryColor: string
■ fillColor: string
■ fillType: string
■ geometry: Geometry
■ transparency: double

Search Services

Proximity Service
Proximity
◀ findNearest(line: Geometry, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findNearest(point: Point, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findWithin(line: Geometry, radius: double, proximityOptions: ProximityOptions, token: string): ResultSet
◀ findWithin(point: Point, radius: double, proximityOptions: ProximityOptions, token: string): ResultSet
◀ getAvailableFieldNames(proximityOptions: ProximityOptions): FieldDesc[]
◀ getVersion: string

ProximityOptions
■ dataSource: string
■ resultRange: ResultSetRange
■ returnFieldsList: string[]
■ units: string
■ whereClause: string

ResultSetRange
■ count: integer
■ startIndex: integer

Query Service
Query
◀ envelopeQuery(envelope: Envelope, queryOptions: QueryOptions, token: string): ResultSet
◀ geometryQuery(geometry: Geometry, queryOptions: QueryOptions, token: string): ResultSet
◀ getAvailableFieldNames(queryOptions: QueryOptions): FieldDesc[]
◀ getVersion: string
◀ pointQuery(point: Point, queryOptions: QueryOptions, token: string): ResultSet

QueryOptions
■ dataSource: string
■ geographicalLevel: string
■ returnFieldsList: string[]

ResultSet
■ fields: FieldDesc[]
■ rows: RowData[]
■ totalCount: integer

RowData
■ fieldValues: string[]

Meta Services

Account Info Service
AccountInfo
◀ getAvailableDataSources(group: string, service: string, token: string): DataSourcesInfo
◀ getDataSourceInfo(dataSource: string, token: string): DataSourceInfo
◀ getDataSourcesInfo(group: string, token: string): DataSourcesInfo
◀ getGroupInfo(group: string, token: string): GroupInfo
◀ getGroupInfoForExtent(group: string, token: string): GroupInfo
◀ getServiceInfo(dataSource: string, token: string): ServiceInfo
◀ getServiceInfoForExtent(dataSource: string, token: string): ServiceInfo
◀ getServiceInfoForGroup(group: string, token: string): ServiceInfo
◀ getServiceInfoForGroupAndExtent(group: string, token: string): ServiceInfo
◀ getServiceInfoForGroupAndExtentAndExtent(group: string, token: string): ServiceInfo
◀ getUserInfo(token: string): UserInfo
◀ getVersion: string

Data Source
■ desc: string
■ name: string

Data Source Info
■ dataSource: string
■ stats: UsageStats

Group
■ desc: string
■ name: string
■ services: Service[]

Group Info
■ group: string
■ stats: UsageStats

Service
■ accessUrl: string
■ desc: string
■ name: string
■ type: string

Service Data Source Info
■ dataSource: string
■ service: string
■ stats: UsageStats

Service Info
■ service: string
■ stats: UsageStats

Usage Stats
■ dailyAvg: long
■ pastDay: long
■ pastMonth: long
■ pastWeek: long
■ pastYear: long
■ percent: long
■ usage: long

User Info
■ dataSources: DataSourcesInfo
■ expectedEnd: date
■ expiration: date
■ groups: Group[]
■ individualServices: Service[]
■ poiAvail: integer

Common
CoordinateSystem
■ datumTransformation: string
■ projection: string

Envelope
■ coordinateSystem: CoordinateSystem
■ max: double
■ may: double
■ minx: double
■ miny: double

Geometry
■ coordinateSystem: CoordinateSystem
■ parts: integer[]
■ points: SimplePoint[]
■ type: integer

Key Value
■ key: string
■ value: string

Point
■ coordinateSystem: CoordinateSystem
■ x: double
■ y: double

SimplePoint
■ x: double
■ y: double

Authentication Service
Authentication
◀ getToken(username: string, password: string): string
◀ getToken(username: string, password: string, expiration: integer): string
◀ getVersion: string
◀ validateToken(token: string): integer

Utility Service
Utility
◀ getProjectedEnvelope(geometry: Geometry, toCoordSys: CoordinateSystem, token: string): Envelope
◀ getVersion: string
◀ projectEnvelope(envelope: Envelope, toCoordSys: CoordinateSystem, token: string): Envelope
◀ projectGeometry(geometry: Geometry, toCoordinateSystem, token: string): Geometry
◀ projectPoint(point: Point, toCoordSys: CoordinateSystem, token: string): Point

