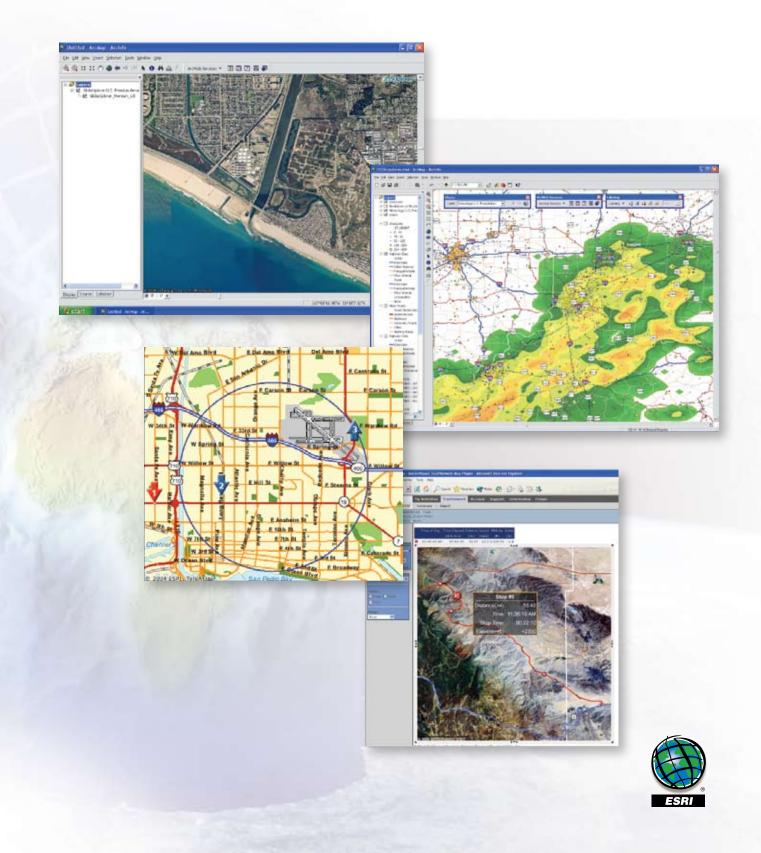
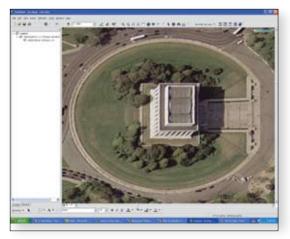
ArcWeb[™] Services

On-Demand Mapping and GIS via the Web



ArcWebsm Services

On-Demand Mapping and GIS via the Web



GlobeXplorer* image of the Lincoln Memorial, Washington, D.C., accessed via ArcGIS Desktop

"Nothing beats a visual representation. Show a client a map, and they see exactly the targeting technique used. It is very powerful."

Ken Gilford GIS Project Manager COX Target Media

What Are ArcWeb Services?

ESRI® ArcWeb⁵⁴⁴ Services are geographic information system (GIS) Web services hosted by ESRI. They give you on-demand access to a wide range of mapping and GIS capabilities and eliminate the overhead of purchasing and maintaining large datasets.

With ArcWeb Services, data storage, maintenance, and updates are handled by ESRI. You can access dynamic, up-to-date content and capabilities directly using ArcGIS®, or you can use ArcWeb Services to quickly build and deploy unique, Web-based applications. ArcWeb Services data and capabilities can also be combined with your data to create highly customized maps and applications.



Sportsim used ArcWeb Services to create this athletic performance tracking application.

Secretary Secret

Binspy used ArcWeb Services to build Fleetspy, a fleet tracking application.

Why Use ArcWeb Services?

ArcWeb Services allow you to quickly and easily add mapping and spatial functionality to your applications in a service-oriented architecture (SOA). Examples of how you can use ArcWeb Services include

- Data Visualization—View your data on a map to help spot patterns and trends that you may not otherwise have noticed.
- **Locators**—Help visitors to your Web site find store or office locations with clear maps and directions.
- Geofencing and Tracking—Locate and visualize any GPSenabled device as well as certain cell phones. For example, you can track your delivery vehicles, sales force, or maintenance crews, and you can determine if vehicles have entered or exited a specific area.

 Business Intelligence—Perform sales and market analyses and site location using the most up-to-date, detailed demographic data.

Who Uses ArcWeb Services?

ArcWeb Services are used by

- Commercial sector businesses that need locationbased services such as facility locators, mobile work force management, and asset tracking as well as location analytics such as business demographic and market area analysis
- Information technology departments using SOA in need of standards-compliant Web services
- GIS professionals needing to augment and share data in a standards-based environment
- Local, state, and federal governments wanting to better serve the public by providing Web-based
 311 applications, Megan's Law applications, office locations, and countless other Web-based services

"I love hard-dollar savings and short returns on investment. This is how technology is supposed to work in a business."

> Bob Bramski Vice President and CIO DS Waters



Gambino Real Estate's Web site uses ArcWeb Services to help home buyers locate available properties.



The United States Postal Service uses this intranet application to determine the locations of its facilities.



DS Waters uses Aligo's WorkTrack application to view the location of field employees. WorkTrack uses ArcWeb Services for its mapping capabilities.

On-Demand Spatial Data

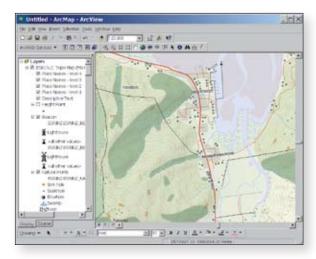
ArcWeb Services give you access to a wealth of data. In addition to street maps, you can access live weather maps, digital orthophoto quarter quadrangles, topographic maps, live traffic information, shaded relief imagery, flood data, and extensive demographic data. ESRI is continually adding to a lengthy list of data providers that includes Tele Atlas, NAVTEQ, Meteorlogix, GlobeXplorer, Pixxures, National Geographic, ESRI Community, and many more. ArcWeb Services allow you to reduce the overhead of purchasing and maintaining large datasets.

You can combine ArcWeb Services data with your own point, line, and polygon data to create maps and Web services that fit your specific needs.

Access ArcWeb Services as a Developer or through ArcGIS

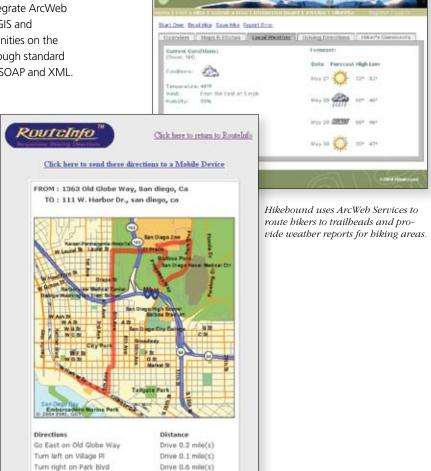
If you are a developer, you can easily integrate ArcWeb Services with any application and serve GIS and mapping capabilities to targeted communities on the Web. ArcWeb Services are deployed through standard Web protocols and languages including SOAP and XML.

If you are an ArcGIS Desktop (ArcView®, ArcEditor™, or ArcInfo®) user, you can use ArcWeb Services to enrich your work with capabilities, such as routing and geocoding, as well as dozens of datasets from the world's leading commercial data providers.



Topographic map of Havelock, New Zealand, accessed via ArcGIS Desktop

Hikehound



Traftools created its RouteInfo application using ArcWeb Services

Available Services

The following services are available with ArcWeb Services:

- Find Services—Find the location of an address, city, place, phone number, or Internet Protocol number. Use these services to get the street address closest to a particular location and perform batch geocoding on large volumes of data.
- Map and Route Services—Get a map of an area or generate
 multipoint driving directions. You can choose from a wide
 variety of map types, or you create a custom map service using
 a combination of your data and ArcWeb Services data. You
 can also create optimized driving directions or use the available
 traffic information to find routes around traffic.
- Spatial Query Services—Perform query and proximity searches such as "find nearest," "find within a distance," and "tell me about this area."
- Data Management Services—Upload and manage your own point, line, and polygon data as well as associated attribute information. You can then combine this data with ArcWeb Services data to create your own custom map services or use the data in conjunction with the Spatial Query Services.
- Wireless Services—Determine the current location of a cellular phone and send a text message to the phone or find out if the phone is switched on or off.
- **Report Services**—Create a wide variety of demographic reports, site reports, and thematic maps for a specific area.
- Utility Services—Give you everything you need to manage your ArcWeb Services account. Utility services also provide a set of useful geometric functions that you can use throughout your application—everything from map projection conversion to geometry creation to distance calculations.



Example of using the ArcWeb Services place-finder service in conjunction with ArcGIS Desktop



ArcWeb Services softwarebased application for helping citizens locate the nearest polling place

Lamar Advertising

Lamar Advertising Company is one of the largest and most experienced outdoor advertising companies in the United States. It currently operates 152 outdoor advertising companies in 43 states and is a leader in the highway logo sign business. Lamar provides clients with targeted placements for their outdoor advertising. Finding vacant billboards in the best areas for various products requires managing vast amounts of data including the actual location of billboards and demographics of areas where billboards are located. Using ArcWeb Services, sales staff can enter an address location requested by a client and perform a search radius in miles to find all available billboard panels within that radius, displaying them on a map. A sales staff of 1,200 requests an average of 132,000 map proposals each day.

"ESRI was the only vendor that could accommodate the requests of Lamar's sales staff and clients. ArcWeb Services were also easy for our Management Information Systems Department to implement."

Tom McNamee, Chief Information Officer, Lamar Outdoor Advertising

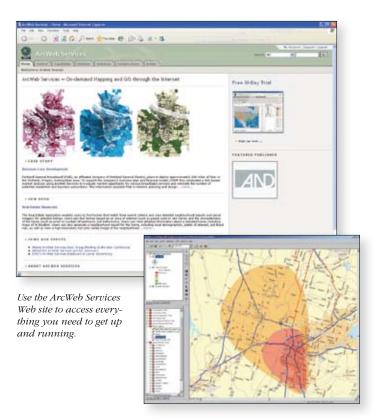


Easily Create, Customize, and Administer Services

When you subscribe to ArcWeb Services, you can take advantage of a Web site that gives you everything you need to get started. Through this Web site, you can easily search the library of available services and data; use Web-based wizards for uploading, geocoding, and managing your own data; and access easy-to-use tools for creating, managing, customizing, and publishing your own services. This Web site also includes all the resources you need for managing ArcWeb Services users within your organization and monitoring your account usage.

Adherence to Standards

ArcWeb Services communicate using SOAP, an international standard, making them compatible with any programming language, Web services toolkit, or Internet-enabled device. You do not have to learn new programming languages or environments to use ArcWeb Services, so you can rapidly deploy your applications.



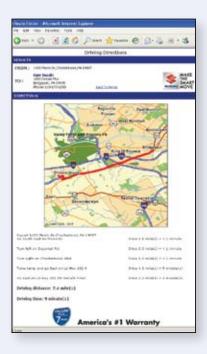
This plume modeling map combines local data with ArcWeb Services.

American Suzuki Motors Corporation

American Suzuki Motors Corporation (ASMC) has implemented a dealer locator on its Web site that is powered by ArcWeb Services. ASMC knows the automotive business well but did not want to have to become an expert in GIS technology. ASMC provided its store locations in digital format to ESRI, which then geocoded these locations. Dealer locations can be viewed on a street map at www.suzuki.com. When a visitor to the Suzuki Web site types in a ZIP Code, an ArcWeb Services query returns a list of Suzuki automobile dealers within a 50-mile radius. Along with a list of dealers, the query also returns hyperlinks to dynamic map displays and driving directions.

"When we compared the cost of hosting and maintaining the geographic data ourselves, it became clear that ArcWeb Services were the superior solution. Subscribing to ArcWeb Services is a great way to help our customers find and route themselves to our stores for a minimum cost."

David Harris, Internet Development Manager, ASMC



Reliable Support

A key advantage of subscribing to ArcWeb Services is the support that ESRI provides. Our hosting services are reliable and available 24/7. To ensure consistent and dependable uptime, ESRI maintains a redundant site at a remote location. Also, ESRI offers technical support to ArcWeb Services subscribers via phone, fax, and e-mail. Additional assistance is available through in-depth, online help including samples, tutorials, discussion forums, and much more.



ArcWeb Services come with extensive online help.

Learn More

To learn more about ArcWeb Services or sign up for a free 30-day evaluation, visit www.esri.com/arcwebservices.



For more than 35 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 enterprise-wide 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

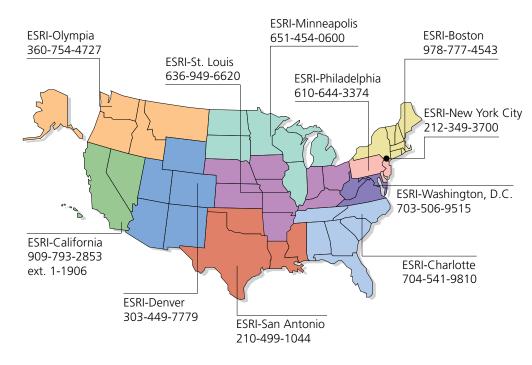
info@esri.com

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

www.esriaustralia.com.a
Belgium/Luxembourg www.esribelux.com
Bulgaria www.esribulgaria.com
Canada www.esricanada.com
China (Beijing) www.esrichina-bj.cn
China (Hong Kong) www.esrichina-hk.com

Australia

110	IIICE
www.es	rifrance.fr
www.esri-	Switzerland germany.de i-suisse.ch
	ngary srihu.hu
	dia iindia.com
62-21-5	a/Malaysia 527-1023 74-9930
	aly sriitalia.it

France

Japan	Singapore
www.esrij.com	www.esrisa.com
Korea	Spain
www.esrikr.co.kr	www.esri-es.com
Netherlands	Sweden
www.esrinl.com	www.esri-sweden.com
Poland	Thailand
www.esripolska.com.pl	www.esrith.com
Portugal	United Kingdom
www.esri-portugal.pt	www.esriuk.com
Romania	Venezuela
www.esriro.ro	www.esriven.com

Place ESRI business partner or distributor address here.

Copyright © 2005 ESRI. All rights reserved. ESRI, the ESRI globe logo, ArcWeb, ArcGIS, ArcView, ArcEditor, ArcInfo, ArcMap, @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.



ArcWeb Services 2006 SOAP Object Model

Data Management

taManager	DataFileInfo
- addFeatures(—■ dataFile: DataFile
dataFileName: string,	—■ dateCreated:long
features: Feature[],	- extent: Envelope
rollbackOnError: boolean,	—■ featureCount: integer
token:string): FeaturelDInfo[]	— geographicExtent: Envelope
createDataFile(- lastModified: long
dataFile: DataFile,	—■ maxScale: long
token: string): void deleteDataFile(minScale: long size: long
dataFileName: string,	- size; long
token:string): void	Feature
deleteFeatures(■— fieldValues: KeyValue[]
dataFileName: string,	- geometry: Geometry
featureIDs: string[],	The state of the s
token:string): void	FeatureIDInfo
findFeatures(—■ errorMessage: string
dataFileName: string, featureSearchOptions:	—■ id: string
FeatureSearchOptions,	
token:string): FeatureResultSet	FeatureInfo
geocodeFeatures(—■ dateCreated:long
dataFileName: string,	—■ feature: Feature
featureIDs: string[],	—■ geocodeDataSource: string
geocodeOptions: GeocodeOptions,	—■ geocodeStatus: string
token:string): GeocodeInfo() getAvailableDataFileAttributes (→ id: string
token: string): AttributeDesc[]	- lastModified: long
getDataFilesInfo(FeatureResultSet
dataFileNames: string[],	—■ features: FeatureInfo[]
token:string): DataFileInfo[]	- totalCount: integer
getUniqueValues(Total Country and
dataFileName: string,	FeatureSearchOptions
fieldName: string,	■ intersectGeometry: Geometry
resultSetRange: ResultSetRange, token:string): StringSet	- returnGeometry; boolean
getVersion; string	 searchOptions: SearchOptions
renameField(spatialBuffer: SpatialBuffer
dataFileName: string,	gen and finding the second and the second
oldFieldName: string,	FeatureUpdateOptions
newFieldName: string,	■— fieldValues: KeyValue[]
token:string): void	■— geometry: Geometry
updateDataFile(Figure 30-24
dataFileName: string, dataFile: DataFile,	FieldInfo
token:string): void	■■ isIndexed: boolean
updateFeatures(■■ label: string
dataFileName: string,	■■ name: string
featureUpdateOptions:	■■ type: integer
FeatureUpdateOptions,	GeocodeCandidate
featureIDs: string[],	
token:string): void	—■ address: Address
dressFields .	desc1: string desc2: string
	extent; Envelope
city: string country: string	→ matchType: string
houseNumber: string	- point: Point
intersection: string	—■ score: double
postalCode: string	—■ type: string
stateProvince: string	
street: string	GeocodeInfo
Tanto	candidates: GeocodeCandidate[]
aFile	errorMessage: string
addressFields: AddressFields	→ hasMore: boolean → matchType: string
attributes: KeyValue[]	- matchtype: string - totalCount: integer
coordSys: CoordSys fieldInfo: FieldInfo[]	- totalcount integer
geometryType: integer	GeocodeOptions
name: string	- dataSource: string
	extendedPostalCode: boolean
	■ formatAddressFields: boolean
	■— partialAddress: boolean

ommon	
AttributeDesc	KeyValue
■ length: integer	► key: string
name: string	■-■ value: string
 precision: integer 	Partie Novon Zonille W
— required: string	MapImageSize
—■ shortDesc: string	■— height: integer
■ type: string	■— width: integer
ircle	OrderBy
 center: Point 	■— descending: boolean
radius: double	■ field: string
 radiusUnits: string 	new sung
trueCircle: boolean	Point
CoordSys	■ coordSys: CoordSys
	■-■ x: double
→ datumTransformation: string → projection: string	■ ■ y: double
- projection: string	II.
DateTimeRange	ResultSet
— end: long	→ fields: FieldDesc[]
→ start:long	─■ rows: RowData[]
- State long	—■ totalCount: integer
nvelope	ResultSetRange
coordSys: CoordSys	The state of the s
	count: integer
→ maxY: double	startIndex: integer
minX: double	RowData
minY: double	7.5 M.C. ALL
ieldDesc	─■ fieldValues: string[]
■ length: integer	SearchOptions
→ name: string	■ orderByList: OrderBy[]
■ precision: integer	■ resultSetRange: ResultSetRange
—■ shortDesc: string	■ returnFields: string[]
■ type: string	■— whereClause: string
Geometry	SimplePoint
	■■ x:double
coordSys: CoordSys parts: integer[]	■-■ y: double
parts: integer[] points: SimplePoint[]	
type:integer	SpatialBuffer
THE RESERVE TO SERVE THE PROPERTY OF THE PROPE	■ bufferDistance: double
	■— bufferUnits: string

- totalCount: integer

Find

Common	Content Finder Service
Address	ContentFinder
 city: string country: string houseNumber: string intersection: string postalCode: string stateProvince: string street: string 	
AddressFinderInfo countryCodes: KeyValue[] dataSource: string extendedPostalCode: boolean partialAddress: boolean reverseGeocode: boolean snapTypes: string[]	token: string): ContentFinderR description of the string
AddressFinderOptions dataSource: string extendedPostalCode: boolean partialAddress: boolean resultSetRange: ResultSetRange snapType: string	token: string: ContentFinderR token: string: contentFinderR contentType: string, attributeName: string, token: string): string [] getVersion: string searchRecords(contentFinderSearchOptions:
GeocodeCandidate —■ address: Address	ContentFinderSearchOption: token: string): ContentFinderR
 desc2: string extent: Envelope matchType: string point: Point score: double type: string 	ContentFinderRecord attributes: KeyValue[] contentLastUpdated: long contentProperties: KeyValue[] extent: Envelope id: ContentFinderRecordID
GeocodeInfo	ContentFinderRecordID
 candidates: GeocodeCandidate[] errorMessage: string hasMore: boolean matchType: string 	■■ contentType: string ■■ name: string ■■ service: string
→ ■ totalCount: integer PlaceFinderInfo	ContentFinderResultSet
— countryCodes: KeyValue[] — dataSource: string	─■ totalCount: integer ContentFinderSearchOptions
dataSource: string types: KeyValue[]	ContentFinderSearchOptions
PlaceFinderOptions dataSource: string filterCountry: string filterExtent: Envelope filterType: string resultSetRange: ResultSetRange searchType: string	 dataType: integer extent: Envelope orderBy: string orderDescending: boolean ownerType: string resultSetRange: ResultSetRange returnAttributes: string[] returnContentProperties: boole
Address Finder Service	■— service: string ■— spatialOperator: string ■— whereClause: string
AddressFinder	The same as a sing
findAddressByPoint(point: Point,	Place Finder Service
addressFinderOptions: AddressFinderOptions, token: string): Address — findLocationByAddress(PlaceFinder → findPlace(placeName: string,

	addressFinderOptions:
	AddressFinderOptions,
	token: string): Address
-	findLocationByAddress(
	address: Address,
	addressFinderOptions:
	AddressFinderOptions,
	token: string): GeocodeInfo
•	findLocationByIP(
	ipAddress: string,
	addressFinderOptions:
	AddressFinderOptions,
	token: string): GeocodeInfo
-	findLocationByPhoneNumber(
	phoneNumber: string,
	address Finder Options:
	AddressFinderOptions,
	token: string): GeocodeInfo
-	getInfo(
	dataSources: string[],

lac	dress Manager Service
Add	ressManager
•	findAddressesByPoint(
	points: Point[],
	addressFinderOptions:
	AddressFinderOptions,
	token: string): AddressInfo[]
•	findLocationsByAddress(
	addresses: Address[],
	addressFinderOptions:
	AddressFinderOptions,
	token: string): GeocodeInfo[]
•	findLocationsByIP(
	ipAddresses: string[],
	addressFinderOptions:
	AddressFinderOptions,
	token: string): GeocodeInfo[]
4-	findLocationsByPhoneNumber(
	phoneNumbers: string[],
	addressFinderOptions:
	AddressFinderOptions,
	token: string): GeocodeInfo[]
•	getInfo(
	dataSources: string[1],

token: string): AddressFinderInfo[]

getVersion: string

- errorMessage: string

AddressInfo

Content Finder Service ContentFinder — getAvailableAttributes(

	The state of the s
	contentType: string,
	token: string): AttributeDesc[]
_	getAvailableServices(
	token: string): string[]
-	getDependentContent(
	contentFinderRecordID:
	ContentFinderRecordID,
	token: string): ContentFinderRecord[]
_	getRecord(
	contentFinderRecordID:
	ContentFinderRecordID,
	token: string): ContentFinderRecord
-	getSubContent(
	contentFinderRecordiD:
	ContentFinderRecordID,
	token: string): ContentFinderRecord[]
-	getUniqueValues(
	contentType: string,
	attributeName: string,
	token: string): string []
-	getVersion: string
-	searchRecords(

token: string):	string []
→ getVersion: string	1
← searchRecords(
contentFinder	SearchOptions:
ContentFinde	erSearchOptions,
token: string):	ContentFinderResultSet
ContentFinderRecor	d
—■ attributes: KeyVa	lue[]
—■ contentLastUpda	ited: long
- U.S.	44 44 4 4 4

- attitudatestities value ()
—■ contentLastUpdated: long
—■ contentProperties: KeyValu
- extent: Envelope
→ id: ContentFinderRecordID
ContentFinderRecordID
■ contentType: string
■■ name: string
■■ service: string

Con	tentFinderResultSet
-	records: ContentFinderRecord[
-	totalCount: integer

—■ totalCount: integer			
ContentFinderSearchOptions			
-	contentType: string		
	dataScale: long		
-	dataType: integer		
-	extent: Envelope		
	orderBy: string		
	orderDescending: boolean		
-	ownerType: string		
_	racultSatPanga-PacultSatPanga		

Plac	ce Finder Service
Plac	eFinder
+	findPlace(
	placeName: string,
	placeFinderOptions: PlaceFinderO
	token: string): Geocodelofo

	placeName: string,
	placeFinderOptions: PlaceFinderOption
	token: string): GeocodeInfo
-	getInfo(
	dataSources: string[],
	token: string): PlaceFinderInfo[]
-	getVersion: string

FinderSamp	e			
indPlace(
placeName	string.			
placeFinde	Options	Place	FinderOp	tions)
Geocodeln	fo			
etinfo(): Plac	eFinderl	nfo		
6	indPlace(placeName placeFinde GeocodeIn	findPlace(placeName: string, placeFinderOptions GeocodeInfo	findPlace(placeName: string, placeFinderOptions: Place	indPlace(placeName: string, placeFinderOptions: PlaceFinderOp GeocodeInfo

getVersion: string

Report Service				
Report	CustomReportVariables			
← createPDFDocument(■— variableDataSource: string			
templateName: string,	■ variableNames: string[]			
data: KeyValue[],				
token: string): string	CustomVariableInfo			
 getAvailableTemplateNames(—■ database: string			
token: string): string[]	—■ fieldDesc: FieldDesc			
→ getCustomSiteReport(k			
site: Site,	GeographylDOptions			
reportOptions: ReportOptions[],	- geographylDFilter: string			
variables: CustomReportVariables[],	 geographyLevelFilter: string 			
token: string): ReportInfo ← getCustomStandardGeographyReport(- keyword: string			
standardGeographies: StandardGeography(),	 outputGeographyLevel: string 			
reportOptions: ReportOptions, variables: CustomReportVariables[],	- resultSetRange: ResultSetRange			
token: string): ReportInfo	Reportinfo			
→ getCustomVariablesi	—■ dataSource: string			
variableDataSource: string,	—■ reportFormat: string			
token: string): CustomVariableInfo[]	—■ reportURL: string			
→ getDatabases(
token: string): string[]	ReportOptions			
→ getGeographyIDs(■— dataSource: string			
geographylDOptions: GeographylDOptions,	- reportFormat: string			
token: string): ResultSet	- reportHeader: KeyValue[]			
→ getReportHeaderKeys(The state of the s			
methodName: string,	Site			
token: string): KeyValue[] petSiteReports(- calcType: string			
site: Site.	polygons: Geometry[]			
reportOptions: ReportOptions[],	- ringsRadii: double[]			
combineReports: boolean,	- ringUnits: string			
token: string): ReportInfo[]	- siteLocation: Point			
→ getStandardGeographyReport(- siteName: string			
standardGeographies: StandardGeography[],				
reportOptions: ReportOptions,	StandardGeography			
token: string): ReportInfo	- geographyIDs: string[]			
→ getTables(geographyLevel: string 			
database: string,				
token: string): string[]	ThematicReportOptions			
 getThematicMapReport(sites: Site[]. 	- classificationMethod: string			
reportOptions: ReportOptions,	- colorPalette: string			
thematicReportOptions: ThematicReportOptions,	JAN ANALOWAYANANANANA			
thematicVariable: string,	■— numClasses; integer			
token: string): Reportinfo	2			
→ getThematicVariables(ThematicVariableInfo			
database: string,	—■ database: string			
table: string,	—■ shortDesc string			
token: string): ThematicVariableInfo[]	— table: string			
→ getVersion: string	—■ variable; string			

Report

Map & Route

Common	Map Image Serv	
CircleDesc	MapImageOptions	MapImage
 antialiasing: string boundaryColor: string boundaryTransparency: double boundaryType: string circle: Circle fillColor: string fillTransparency: double overlapLabels: string thickness: integer 	■ backgroundColor: string ■ circles: CircleDesc[] ■ dataSource: string ■ displayLayers: Layer[] ■ lines: LineDesc[] ■ mapImageFormat: string ■ mapImageSize: MapImageSize ■ mapLegend: MapLegend ■ markers: MarkerDesc[] ■ outputCoordSys: CoordSys	← convertMapCoords mapArea: MapAi mapSize: MapIm mapCoords: Poir token:string): Pix ← convertPixelCoords mapArea: MapAi mapSize: MapIm mapClickPoints: token:string): Po
LabelDesc	polygons: PolygonDesc[]	dokenstring); Fo → getBestMap(
■— angle: double	■— returnLayers: boolean	maplmageOptio
■— antialiasing: string	 scaleBars: ScaleBarDesc[] styleSheet: string 	bufferPercent: d token: string): M
 blockoutColor: string caps: string 	- stylestreet string	→ getBestMapArea(
font: string	MapLegend	maplmageOptio
■— fontColor: string	■— antialiasing: string	bufferPercent: de
 ■— fontSize: integer ■— fontStyle: string 	 autoextend: boolean backgroundColor: string 	token:string): Ma
- glowing: boolean	- columns: integer	mapArea: MapA
■— hAlignment: string	■— font: string	mapimageOptio
 interval: integer outlineColor: string 	 ■ height: integer ■ layerFontSize: integer 	thematicData:ThematicOption
 outlinecolor: string overlapLabels: string 	splitText: string	thematicColors:
 transparency: double 	 swatchHeight: integer 	token: string): M
■— vAlignment: string	 swatchWidth: integer title: string 	→ getESRiThematicM mapArea: MapAr
Layer	titleFontSize: integer	maplmageOptio
name: string	■— valueFontSize: integer	thematicField: st
- visibility: string	■— width: integer	thematicOptions thematicColors:
LayerInfo	MarkerDesc	token: string): M
—■ legends: LayerLegend[]	■— color: string	→ getFusedMapImag mapFusionOptic
—■ name: string —■ visibility: string	■ iconDataSource: string ■ label: string	outputImageFor
- Visibility: string	■ labelDesc: LabelDesc	token: string): str
LayerLegend	location: Point	→ getMaps(mapAreas: Map/
layerClassName: string	name: string size: integer	mapimageOptio
—■ symbolURL: string		token: string): M
LineDesc	PixelCoord	→ getMarkerNames(iconDataSource:
■— antialiasing: string	■ x: integer ■ y: Integer	token: string): str
- color: string	- Janeger	→ getSavedMap(savedMapID: stri
 endType: string geometry: Geometry 	PolygonDesc	token: string): M
■— jointType: string	■— antialiasing: string	→ getThematicFields
 lineType: string 	 boundary: string boundaryColor: string 	thematicDataSo
 overlapLabels: string thickness: integer 	 boundaryJointType: string 	token: string); str ← getValueMap(
- transparency: double	 boundaryTransparency: double 	mapArea: MapAr
	 boundaryType: string boundaryWidth: integer 	maplmageOptio
MapArea ■ center: Point	fil/Color: string	thematicData:Th codeColorValues
■■ extent: Envelope	■— fillinterval: integer	thematicOptions
■—■ scale: long	filTransparency: double	token: string): M
Mapimageinfo	■— fillType: string ■— geometry: Geometry	
- a layers: LayerInfo[]	■— overlapLabels: string	mapArea: MapA
→ legendURL: string	ScaleBarDesc	maplmageOptio
—■ mapArea: MapArea	■ barColor: string	token: string): str
—■ mapURL: string —■ outputCoordSys: CoordSys	■ barColor: string ■ barLength: integer	MapFusionOptions
- outputcouldays, coordays	■ barPixelLocation: PixelCoord	■— mapURL: string
	■— barUnits: string	- transparency doub

barWidth: integer

■— fontColor: string

fontSize: intege

Route Finder Service

dsToMapCoords(

RouteDisplayOptions

thickness: intege

RouteFinderOptions

■— avoidTraffic: boolean

■ returnMap: boolean

returnGeometry: boolean

returnTurnByTurnMaps: boolean

routeOptions: RouteOptions

optimizedStopOrder: integer[]

segmentDescs: SegmentDesc[

—■ turnByTurnMaps: MapImageInfo[]

routeMap: MapImageInfo

totalDesc: RouteSummary

routePath: Geometry

trafficDataSource: string

trafficSeverity: integer

■ units: string

RouteOptions

RouteStop

■ desc: string

- point: Point

hwyPref: Integer

■ precision: double

routeType: string

- distanceUnits: string

numericDistance: double

— numericTime: double

routeExtent: Envelope

descriptiveDirections: string

descriptiveDistance: string

- numericDistance: double

totalDistance: string

—■ totalTime: string

extent: Envelope

- numericTime: double

returnExtents: boolean

routeBarriers: Point[]

routeDisplayOptions: RouteDisplayOptions

turnByTurnMapOptions: MapImageOption

routeMapOptions: MapImageOptions

■ dataSource: string

language: string

transparency: double

■ color: string

ons: MaplmageOptic MaplmageInfo

ions: MaplmageOptio ThematicData. ns: ThematicOptions :ThematicColors, MapimageInfo

ions: MaplmageOp ns: ThematicOptions :ThematicColors, MaplmageInfo

tions: MapFusionOpt ormat: string,

MapImageInfo[]

ions: MaplmageOptio es: KeyValue[],

ns: ThematicOption:

 classColors: string[] colorPalette: string

ThematicData - data: KeyValue[precision: integer ■— sign: string

ThematicOptions - classificationMethod: string ■— classLabels: string[] ■— numClasses: integer - thematicOnlyLegend: boolean

Spatial Query

Spatial Query Service ← findFeaturesByExtent extent: Envelope, spatialQueryOptions: SpatialQueryOptions token: string): ResultSet findFeaturesByGeometry geometry: Geometry spatialQueryOptions: SpatialQueryOption token: string): ResultSet - findFeaturesByPoint(token: string): ResultSet dataSources: string[], token: string): SpatialQueryInfo[]

> - availableFields: FieldDesc[— dataSource: string **SpatialQueryOptions** dataSource: string

> > returnGeometry: boolean

spatialBuffer: SpatialBuffer

searchOptions: SearchOptions

SpatialQueryInfo

 distanceUnits: string returnDistance: boolean

Utility

Account Info Service **Utility Service** getServices(convertCircleToGeometry circle: Circle, token: string): ServiceSummary() numPoints: integer getUsage(token: string): Geometry convertCircularArcToGeometry(dateTimeRange: DateTimeRange arc: CircularArc, token: string): AccountSummary numPoints: integer, getVersion: string token: string): Geometry AccountinfoOption ellipse: Ellipse, adminOptions: AdminOption returnDataSources: string[] token: string): Geometry returnDetailedInfo: boolean getBestProjection[- returnServices: string[] token: string): Envelope getDataCollectionDate(- groupUsage: AccountUsage dataSource: string,

extent: Envelope, mapSize: mapImageSize token: string): string getDistance(location1: Point, location2: Point, units: string, numPoints: integer returnGeometry: boolean

getDriveTime(dataSource: string point: Point, interval: integer,

type: string,

—■ supportsArcXML: boolean —■ supportsWMS: boolean

- name: string —■ title: string —■ usage: UsageInfo

- averageDailyCreditUsage: double

creditsRemaining: double

dailyStorageCost: double

diskStorageQuota: long

— diskStorageUsed: long

—■ estimatedEndDate: long

—■ totalStorageCost: double

returnGroupUsage: boolean

—■ expiration: long

returnUsers: string[]

—■ summary: string

—■ title: string

AdminOptions

— creditsUsed: double

dataSourcesInfo: DataSourceInfo[] —■ desc: string —■ name: string —■ wsdl: string

ServiceUsage dataSourcesUsage: DataSourceUsage[- name: string usage: UsageInf

UsageInfo creditsUsed: double

numRequests: long - emailAddress: string —■ firstName: string

─■ isActive: boolean - isAdmin: boolean ─■ lastName: string username: string

UserUsage - accountUsage: AccountUsage servicesUsage: ServiceUsage[- user: User

token: string): DistanceInfo

units: string, token: string): Geometry qetSupportedProjectionsi

token: string): KeyValuel getVersion: string projectExtent(extent: Envelope, projectTo: CoordSys.

token; string); Envelope projectGeometry(geometry: Geometry projectTo: CoordSys, token: string): Geometry

 projectPoint(point: Point, projectTo: CoordSys,

CircularArc - center: Point - endAngle: double - innerRadius; double

 outerRadius: double radiusUnits: string startAngle: double trueCircle: boolean

DistanceInfo distance: double — geometry: Geometry

 axisUnits: string center: Point ■ majorAxis: double

■ minorAxis: double ■— rotation: double trueEllipse: boolean

Authentication Service Authentication

password: string, expiration: integer): string getToken(username: string, password: string): string getVersion: string validateToken(token: string): intege

Wireless

Wireless Location Service devices; MobileDevice[], deviceLocationOptions: DeviceLocationOptions token: string): DeviceLocationInfo[] getDeviceStatus(devices: MobileDevicel 1. token: string): DeviceStatus[] qetDeviceStops(deviceStopOptions: DeviceStopOptions token: string): DeviceStopInfo[] → getTripInfo(deviceTripOptions: DeviceTripOptions token: string): DeviceTripInfo getVersion: string sendSMS(devices: MobileDevicel L message: string, token: string): string[] DeviceLocationInfo —■ accuracy: string - errorMessage: string - point: Point —■ radius: double - timeStamp: long

—status: string DeviceStopInfo dateTimeRange: DateTimeRange - point: Point **DeviceStopOptions** - dataFileName: string

DeviceLocationOptions

errorMessage: string

- radiusUnits; string

DeviceStatus

 dateTimeRange: DateTimeRange ■ maxRadius; double - minDuration: integer ■ radiusUnits: string —■ angle: double distance: double

—■ path: Geometry - speed: integer DeviceTripOptions dataFileName: string dateTimeRange: DateTimeRange ■— returnAngle: boolean

 returnDistance: boolean ■ returnPath: boolean returnSpeed: boolean - units: string

■ dataSource: string

- phoneNumber: string

MobileDevice

Service Name - Request Property

www.esri.com/arcwebservices 800-447-9778

—■ Response Property ■ Request/Response Property

Method

