IHS > Developer Support Tool

IHS Meridian Energy Map Services v 1.1 Developer's Guide

December 19, 2014

Energy Map Services Developer’s Guide v 1.1

December 19, 2014

TRADEMARKS

Contents

Introduction

This document provides information for developers wanting to utilize the IHS Energy Map Services (EMS) in their mapping applications. EMS Map Services can be accessed over SSL via low-level APIs for SOAP and REST. In addition, ESRI has developed a variety of Web APIs on top of REST.

Basic examples showing how to establish a connection to the services are provided for a variety of programming languages.

For information on how to interactively connect to the Services in client applications such as ESRI ArcMap, ESRI's ArcGIS.com, and ESRI's ArcGIS Explorer Online, refer to the *IHS Energy Map Services Users Guide*.

Map Services Catalog

It is highly recommended that developers familiarize themselves with the IHS Energy Map Services catalog.

Folders and services can be browsed at:

Intl and US content:
https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/httpauth/arcgis/rest/services

Global content:
https://meridianmapservices.ihsenergy.com/wss/service/ags-relay/EMS_2_00/httpauth/arcgis/rest/services

1

Connecting to Map Services

Service-Level Security vs. Feature-Level Security

Standard ArcGIS Server security is implemented at the service level only. Based on your credentials, you either have access to a particular service folder or you do not.

EMS implements security at the feature level. Because of this, the process of connecting to EMS services is slightly different than for standard ArcGIS security. Feature-level security allows different users to access the same Map Service but see different spatial areas based on their user credentials. User A subscribes to well surface points for only the Rocky Mountain spatial region, while User B subscribes to only Texas. The result is that User A and User B can both add and use the same Map Service in their applications, but they will see entirely different regions. Support of this security requires slightly different methods of authentication.

EMS provides two types of user authentication: HTTP Basic and Token. The endpoint for the services depends on which type of authentication your application will be using. End point URLs are provided below

HTTP Basic Authentication

The endpoint for EMS Services with HTTP Basic authentication is:

Intl and US content:
https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/httpauth

Global content:
https://meridianmapservices.ihsenergy.com/wss/service/ags-relay/EMS_2_00/httpauth

You must use this endpoint for basic authentication. Tokens will not work with any service accessed via the /httpauth/ based URL.

Token Authentication

The endpoints that consume Energy Map Services with Token authentication are:

Intl and US content:
https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/token/arcgis/rest/services?token=

Global content:
https://meridianmapservices.ihsenergy.com/wss/service/ags-relay/EMS_2_00/token/arcgis/rest/services?token=

Example: https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/token/arcgis/rest/services?token=eyJkYXRhIjp7InNiaiI6eyJ1aWQiOi......

Token-based authentication is supported by this endpoint only. Attempting to use generated tokens with the Basic authentication endpoint will result in errors. It is also important to note that tokens must be requested from the token service and used when making a request against a token secured endpoint.

In order to receive an authenticated token use the endpoint below:

Intl and US content:
https://mapservices.ihs.com/administration/token

Global content:
https://meridianmapservices.ihsenergy.com/administration/token

Please see the section titled *JSON Token Service* for more information on using the token service endpoint.

2

Selecting Basic Authentication or Token-Based Authentication

There are two scenarios for working with secure IHS EMS services: End users with individual credentials and users without individual credentials.

End-User Holds an Individual Set of Credentials for Accessing IHS EMS

In this scenario, a user of a Web application is able to leverage their IHS user name and password to access IHS EMS.

When developing Web applications for such users, two additional scenarios can be identified:

1. Web application presents other content besides IHS EMS and access to the content is secured.
   In this scenario, the Web application will challenge users for credentials other than those used to access IHS EMS. In order to avoid a situation where the user is challenged twice (once for a general application authentication and again for IHS EMS), a server-side code should be used to set and identify supporting IHS EMS authentications.
   The server-side component should be responsible for mapping the user’s application credentials to the user’s credentials for IHS EMS. The server side component would then handle secure access to IHS EMS.
   The recommended technical solution in this case is **proxy and HTTP Basic Authentication.**
2. Web application presents only data from IHS EMS.
   In this scenario a user can be challenged directly in the Web browser to provide credentials for accessing IHS EMS. HTTP Basic Authentication over SSL should be used in this scenario.

End-User Does Not Hold an Individual Set of Credentials for Accessing IHS EMS

In this scenario a single set of credentials for IHS EMS is used to grant access to multiple end users of Web applications. Server-side code is needed to handle secure access to IHS EMS.

The recommended technical solution in this case is **proxy and token authentication**.

3

Technical Solutions for Developing Web Applications with Secure IHS EMS

Proxy and HTTP Basic Authentication

All communication with IHS EMS services is exchanged through a proxy handler. ArcGIS APIs used as Web services do not leverage any kind of authentication for accessing EMS services directly; security is handled by the proxy. Application-level authentication should be used to prevent unauthorized access to EMS services, and all communication between the proxy and the client-side application must be secured and restricted.

For example:

Given a Map Service:

https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/httpauth/US_BASE/Production/MapServer and a proxy address of mapsproxy.company.com.

The application requests all map tiles, queries and other information through the proxy instead of through the Map Service directly. Depending on the application, this can be done in multiple methodologies.

Many ESRI Web APIs support proxies using the “proxy,” or similar property on the layer object. Setting this property causes the API to automatically route requests to the proxy instead of directly to the Map Service. The proxy then must handle the request.

One caveat with this methodology using ESRI APIs is that the proxy must parse the original Map Service URL from the request. Manually appending additional parameters to the Map Service URL may cause problems if using the Silverlight API, as the Silverlight API will sometimes create invalid Map Service URLs. If using Silverlight, any additional parameters such as user names and passwords should not be passed to the proxy in the Map Service request.

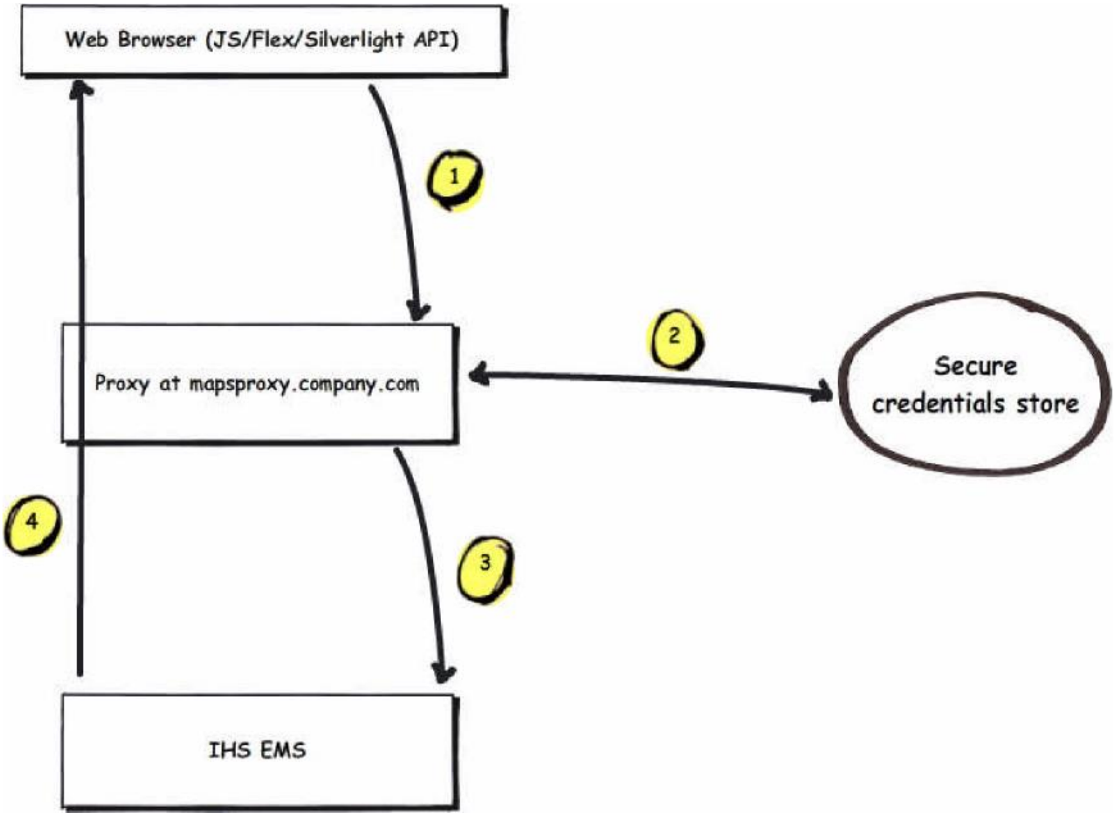
Non-ESRI API applications may need to directly request data against the proxy, and in such cases, building a proxy that can interpret the URL through URL handling may be beneficial. In such cases it may be possible to pass requests to the proxy in this manner:

http(s)://mapsproxy.company.com/US_BASE/Production/MapServer

The proxy will need to authenticate the user without prompting the user for a user name and password. This can be accomplished in several ways, depending on the server technology being used. In .NET, it is possible to pre-authenticate against the endpoint on the first use of the proxy.

In other technologies, adding the username and password to the request’s header will be more appropriate. Examples are available from IHS for several technologies and client platforms including .NET.

The diagram below describes the communication between user’s browser and IHS EMS:


Web Browser (JS/Flex/Silverlight API)
1
2
Proxy at mapsproxy.company.com
Secure credentials store
4
3
IHS EMS

For example:

1. A request is made from the Web browser application for map data using the local proxy.
   http(s)://mapsproxy.company.com
   The request hits the proxy. The application is secured with application-level authentication. For example: via Forms Authentication.
2. Since access to the proxy is allowed only for users authenticated at the application level,

the proxy already knows the identity of the caller. Based on the identity of the caller, the proxy is able to retrieve proper IHS EMS credentials for the user.

3. The URL of the incoming request is rewritten to match IHS EMS scheme: https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/httpauth/US_BASE/Production/MapServer, and the HTTP Authorization header with IHS EMS credentials is attached to the request. The request is sent to IHS EMS.
4. The response from IHS EMS is passed back through the proxy to a Web browser.

Proxy and Token-Based Authentication

In this scenario, IHS EMS URLs with token authentication are described.

A token is requested from the EMS token service using IHS EMS credentials. Once a token has been retrieved from the Service it is stored in a client's cookie, user session, or other unique storage. The Map Service is then added to the map and the token is appended either via client or via a proxy to each Map Service request.

The use of a proxy is optional, depending on client technology. However, in some cases it makes sense, even if a proxy is not required, since the token can be appended to the map request on the server side, and any other server-related needs can be resolved.

For example, assuming the use of a proxy:

Given proxy address is mapsproxy.company.com, instead of the IHS EMS URL to the Production service:

https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/httpauth/US_BASE/Production/Map Server

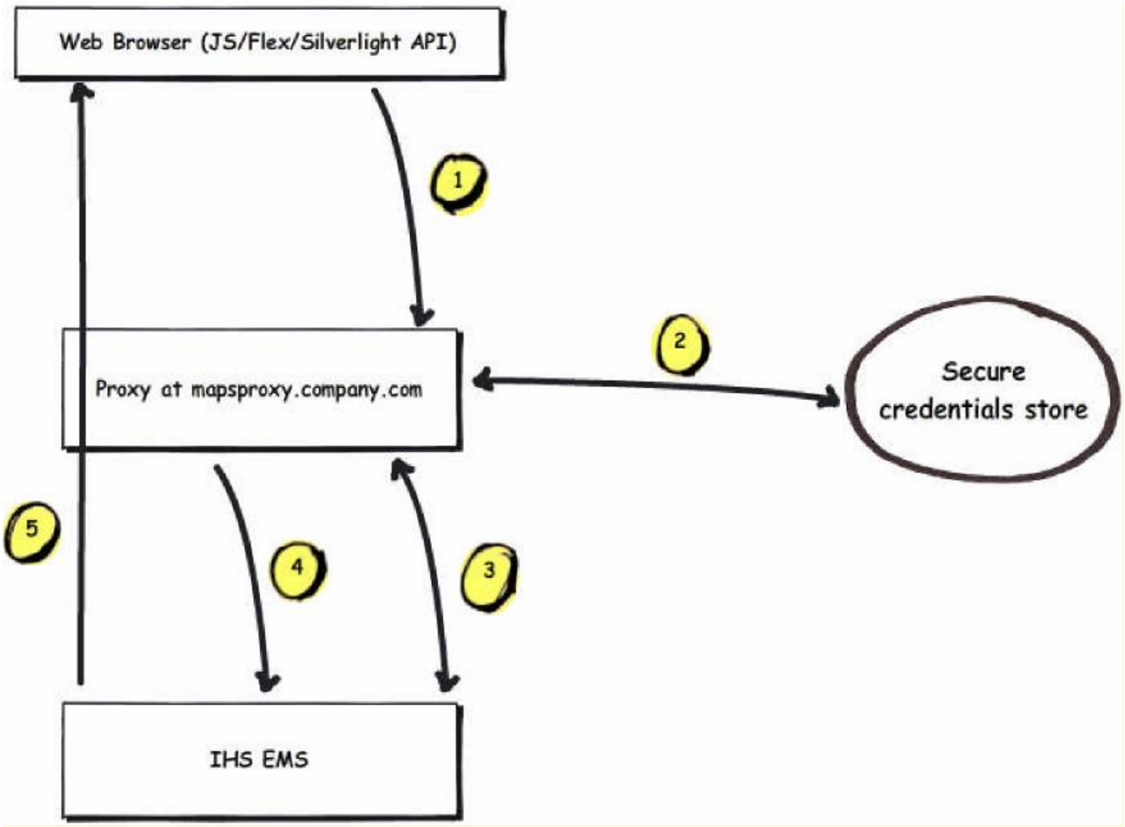
The following URL should be used in the Web layer's proxy property (or otherwise forwarded to this endpoint):

http(s)://mapsproxy.company.com

In the scenario above, IHS requires all IHS EMS data transported over the wire to be secured with SSL.

In this situation, the token can be passed from the client through the proxy, or the token appended to the request in the proxy itself. It is up you to choose the most appropriate method, given your situation.

Requests without the token cookie (first request or cookie expired) would be as follows:

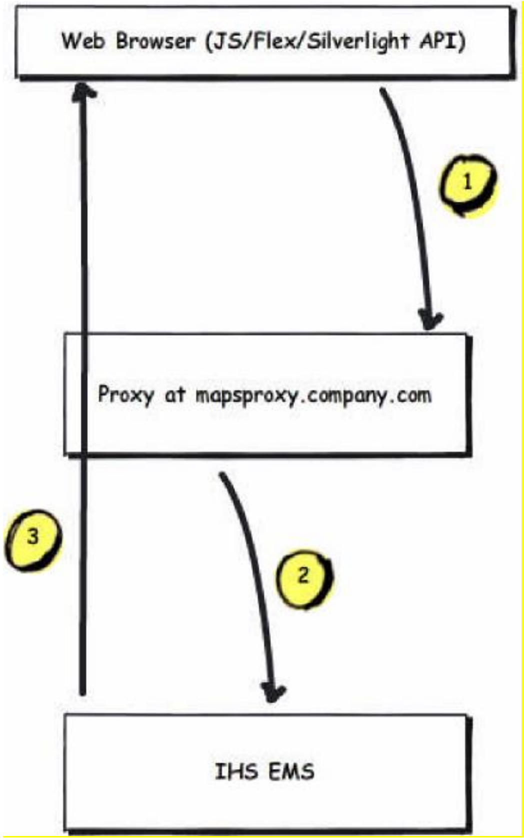

Web Browser (JS/Flex/Silverlight API)
1
Proxy at mapsproxy.company.com
2
Secure credentials store
5
4
3
IHS EMS

1. A request is made from the Web browser. For example: http(s)://mapsproxy.company.com/US_BASE/Production/MapServer.
2. The request hits the proxy. The application is secured with application level authentication. For example: via Forms Authentication.
3. The proxy detects there is no token cookie attached to the request.
4. Proxy retrieves generic IHS EMS credentials from secure store.
5. IHS EMS is called to create the token using following request: https://mapservices.ihs.com/administration/token?username=USERNAME&password=PASSWORD&type=uidwhere USERNAME/PASSWORD matches the generic credentials retrieved in step 2.
   The incoming HTTP request is rewritten to match IHS EMS token authentication scheme and token query string parameter is attached: https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/token/arcgis/rest/services/US_BASE/Production/MapServer?token=TOKEN VALUE.
   The request is sent to IHS EMS.

A response from EMS is passed back to the web application. A cookie containing the token is created and attached to the response.

Cookies should have an expiration time of less than an hour, and the domain that matches the proxy domain. This way all subsequent requests to the proxy will have the cookie automatically attached by the browser, and the cookie will expire before the IHS EMS token expires.

Requests with the token cookie proceed as follows:


Web Browser (JS/Flex/Silverlight API)
1
Proxy at mapsproxy.company.com
3
2
IHS EMS

1. A request is made from the web browser directly against the proxy: http(s)://mapsproxy.company.com
   The request hits the proxy. The application is secured with application level authentication. For example: via Forms Authentication. The token cookie is attached with the request.
2. The proxy detects that there is a token cookie attached to the request. The token is extracted from the cookie.
   An incoming HTTP request is rewritten to match IHS EMS token authentication scheme and token query string parameter is attached:
   https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/token/arcgis/rest/services/US_BASE/Production/MapServer?token=TOKEN VALUE.

A request is sent to IHS EMS.

⚠ **Warning!!!** It is possible IHS EMS will respond with the following XML message indicating token expiration:

```
    <?xml version="1.0" encoding="UTF-8"?>
<ServiceExceptionReport version="1.1.0">
    <ServiceException code="ServiceError">
    <![CDATA[Token expired]]>
    </ServiceException>
</ServiceExceptionReport>
```

In such a case, a new token needs to be retrieved as described in **Requests without the token cookie** section.

3. A response from IHS EMS is passed back to the web application. If a new token had to be created then a token cookie is created and attached to the response.

JSON Token Service

The JSON Token Service is used to generate a unique, secured EMS.

Note: This token is not the same as or compatible with ESRI's tokens or Token Service.

General Workflow

1. Request a token using EMS credentials; i.e. user name and password.
2. The Token Service returns a base64-encoded JSON containing the available user data and the token hash, signed by the Token Service.
3. The client (or proxy) appends the token to the token endpoint URL of a protected service.

Technical Information

Token Service URL

The JSON Token Service is part of the administration of the Web application and can be requested using the URL.

Intl and US content:
https://mapservices.ihs.com/administration/token

Global content:
https://meridianmapservices.ihsenergy.com/administration/token

Supported Authentication Methods

The Token Service provider generally can enable any authentication method available. The following methods are enabled by default:

username/password as form parameters, e.g. `username=example &password=example`

- HTTP Basic Authentication
- SAML Token, passed as form parameter, e.g ?ticket=[samltoken]

Token Format

The token itself is a base64-encoded representation of JSON. Here is an example of a broken-down token:

```
{
      "data":
      {
            "sbj":
            {
                  "uid" : "Administrator",
                  "roles": ["department1", "sMAdministrator",""],
                  "mail": "jdr@localhost.local"
            },
            "exp" : "2011-05-04T17 :40:25.139+0200",
            "isr" : "ct-security"
      },
      "sig" : "L8dATujH8dUlsdhS8OnH/dgC62CMr6H7zhZ8FiXpCjPM7OQ3SG33lSQTvda
arrhKjbVZj1cbRnv6xNg56Oc+pIUl/7CYUR/iVOs7vjn18xm+ir/R3waQ3zbX0b5J4oAA681BwBO3
13JzDFPmTl5e6/n+XiTcbp5t66LMhi6+1Bw="
}
```

Token Lifetime

The token is valid for 60 minutes. All requests issued with an expired token will result in a “token expired” response. You will have to request and use a new token before the 60-minute period expires.

Brief Tokens

Tokens can be large at times, especially if a user object possesses a large number of attributes. To avoid clients and servers having to deal with kilobyte- or megabytes-sized tokens, clients may request "brief" tokens, comprising only the fundamental user attributes. To request a "brief" token, client needs to add the type parameter to the request with the value uid.

IHS recommends using a “brief” token in most cases; the smaller token will be easier to use and is similar in size to the ESRI-type token. Using the “brief token” will significantly lower the amount of data being sent between the client, proxy, and map servers.

Requesting a brief token:

https://mapservices.ihs.com/administration/token?type=uid&username=me&password=secret

Requesting Protected Services

Generally the token can be used to authenticate at any resource. Nevertheless, the main purpose is to authenticate a user when loading a protected service.

To request a protected service, authenticating with the token, simply add the base64- encoded token as an additional form query parameter named "token", or add it as an HTTP header field.

Example Using Query Parameter

https://mapservices.ihs.com/wss/service/ags-relay/EMS_1_00/token/arcgis/rest/services/US_BASE/Production/MapServer?token=eyJkYXRhIjp7InNiaiI6eyJ1aWQiOiJNQVBURVNURVIx...

Example Using HTTP Header

```
Content-Length: ...
Token: eyJkYXRhIjp7InNiaiI6eyJ1...
Host: ...
```

Tokens with JavaScript (Dojo)

The following section shows how to use tokens with JavaScript. These examples specifically use Dojo, but can be easily translated to other JavaScript toolkits.

Requesting a Token

```
function requestJSONToken() {
    var tokenUsername   =   "Administrator";
    var tokenPassword = "Administrator";
    var jsonToken; dojo.xhrPost({
        url: "https://mapservices.ihs.com/administration/token",
        content: {
            username: tokenUsername,
            password: tokenPassword
        },
        load: function (response, ioargs) {
            if (!response) {
                //Authentication
                failed return;
            }
            //Authenticated!
            jsonToken = response;
        },
        error: function(error) { /*Authentication failed.*/ }
    });
}
```

Adding the Token to a Protected ArcGIS Map Service URL, Using ArcGIS JavaScript API

The following code snippet shows an example of adding a token to a protected ArcGIS Map Service URL.

```
function addServiceJSONToken() {
    if (!jsonToken) {
        //Please authenticate at Token Service! return;
    }
    var serviceUrl = "https://mapservices.ihs.com/wss/service/ags-
relay/EMS_1_00/token/arcgis/rest/services/US_BASE/Production/MapServer";
    var serviceUrlWithToken = serviceUrl + "?token=" + escape(jsonToken);
    var dynamicMapServiceLayer = new
    esri.layers.ArcGISDynamicMapServiceLayer(serviceUrlWit hToken);
    map.addLayer(dynamicMapServiceLayer);
}
```

4

Limitations

HTTP GET with Form-Based Authentication

In order to hide password information from browser address lines or server-side access log files, form-based authentication is disabled for HTTP GET requests. The Token Service rejects the requests with an error code.

Cross-Site Scripting

Most browsers have limitations with cross-domain scripting. This can cause issues when attempting to use the Services and handlers that are not on the same domain. In cases where this occurs, it is recommended to use a proxy to route requests to and from a remote resource.

5

Using EMS

Using EMS with ArcObjects

The GISClient Library

Documentation for the ArcObjects GISClient Library can be found at:

http://resources.arcgis.com/en/help/arcobjects-net/conceptualhelp/index.html#/ArcObjects_Help_for_NET_developers

Using EMS with REST and SOAP SDKs

SOAP

Documentation for the SOAP SDK can be found at:
http://help.arcgis.com/en/arcgisserver/10.0/apis/soap/index.htm

REST

Documentation for the REST API can be found at:
http://resources.arcgis.com/en/help/rest/apiref/

Code Example – Accessing REST Service Endpoint with Python (HTTP Basic Authentication)

At times, it may be necessary to access a Map Service through Python code. The following code snippet shows and example of how to access the REST Service Endpoint using Python.

```
import urllib2
from urllib2
import HTTPError

password_mgr = urllib2.HTTPPasswordMgrWithDefaultRealm() user_name =
r"demouser"
password = r"demoPassword"

EMS_url = r"https://mapservices.ihs.com/wss/service/ags-
relay/EMS_1_00/httpauth"
password_mgr.add_password(None, EMS_url, user_name, password)

handler = urllib2.HTTPBasicAuthHandler(password_mgr)
opener = urllib2.build_opener(handler)

opener.open(r"https://mapservices.ihs.com/wss/service/ags-
relay/EMS_1_00/httpauth?f=pjson")
urllib2.install_opener(opener)

try:
    response = urllib2.urlopen(r"https://mapservices.ihs.com/wss/service/ags-
relay/EMS_1_00/httpauth?f=pjson")

except HTTPError, e:
    print e.code

    # Process the response
print response.read()
```

Using EMS via the ESRI Web Map APIs

ESRI provides several Web-mapping APIs that you can use to display IHS Map Services.

Use of the ESRI Web APIs may require the use of proxies in some situations, as previously discussed. If you are using basic authentication and no proxy is used, the browser will challenge the user for credentials. This may be a satisfactory method of connecting for some applications.

If a proxy is chosen, implementation (for either token or basic authentication) will vary.

Note: It is important to remember that communication between the client mapping application and the server-side proxy *must* be secured so that communication can be passed only through the proxy from the client ESRI Web API map.

EMS Map Services are supported by all ESRI Web APIs

JavaScript

The documentation for the ArcGIS API for JavaScript can be found at: https://developers.arcgis.com/javascript/

Silverlight/WPF

The documentation for the ArcGIS API for Silverlight can be found at: https://developers.arcgis.com/silverlight/

Flex

The documentation for the ArcGIS API for Flex can be found at: https://developers.arcgis.com/flex/

6

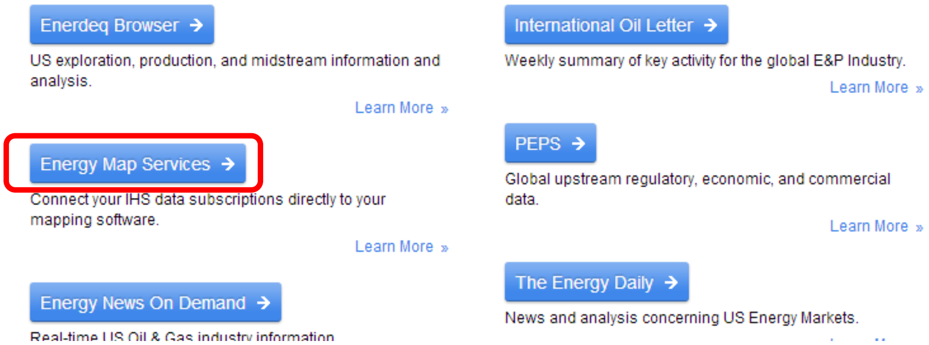
EMS Samples

IHS provides several samples to help you use the Map Services. These samples illustrate the use of both basic authentication and token authentication with simple Map Services. The samples illustrate the use of proxies to pre-authenticate against Map Services, or generate tokens. However, it is important to remember that these examples do not show all possible methods for using IHS Map Services.

Examples are available for the ESRI JavaScript API in .NET, the ESRI Silverlight API in .NET, and the Flex API using Flex/.NET. All .NET examples use either .NET 3.5 or .NET 4.0, and C#. The Silverlight example uses Silverlight 4, while the Flex examples use Adobe FlashBuilder 4.6.

To download the IHS Map Service examples, visit https://my.ihsenergy.com/MenuPage.aspx

Click the **Energy Map Services** link under **MyProducts/Services**.



US exploration, production, and midstream information and analysis.

Connect your IHS data subscriptions directly to your mapping software.

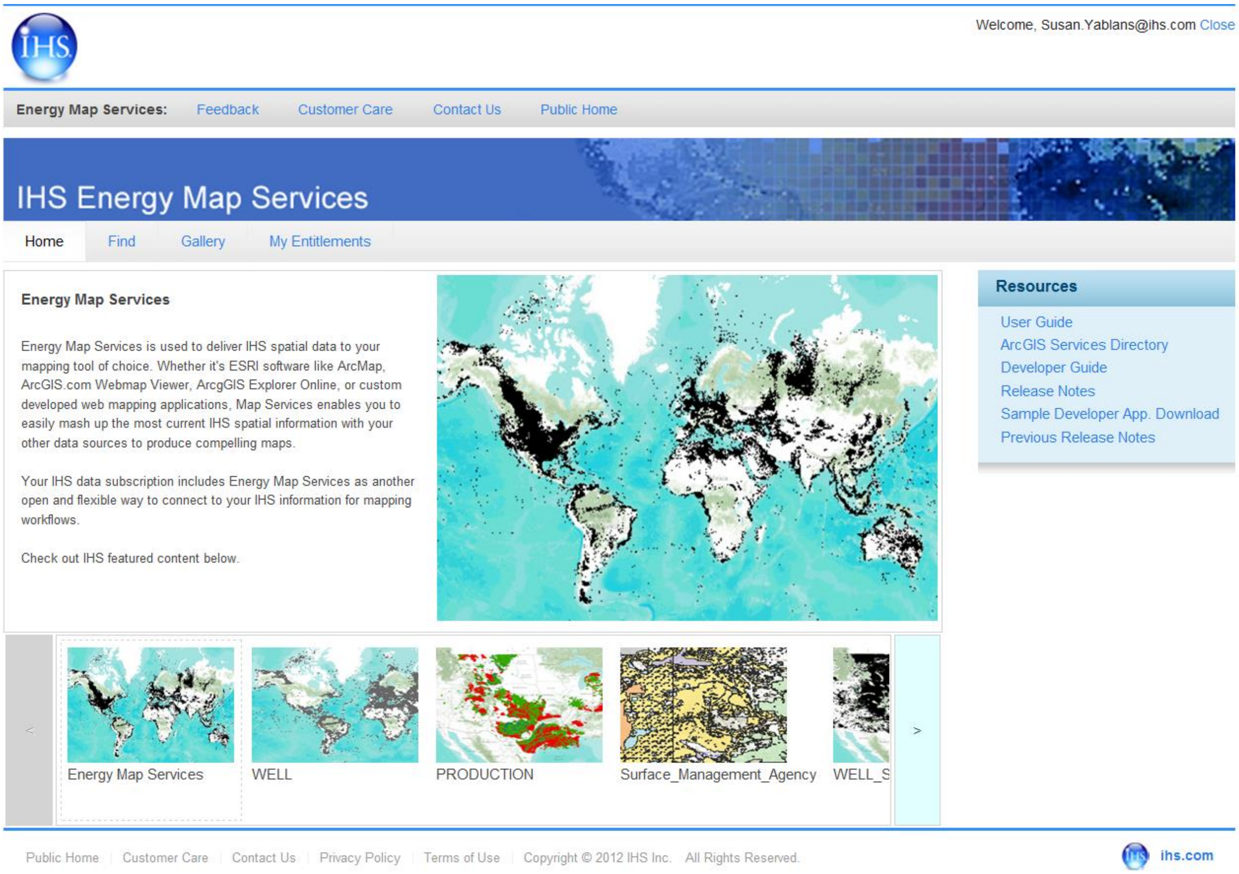
Real-time US Oil & Gas industry information

Weekly summary of key activity for the global E&P Industry.

Global upstream regulatory, economic, and commercial data.

News and analysis concerning US Energy Markets.

Clicking this link will take you to the Energy Map Services home page.



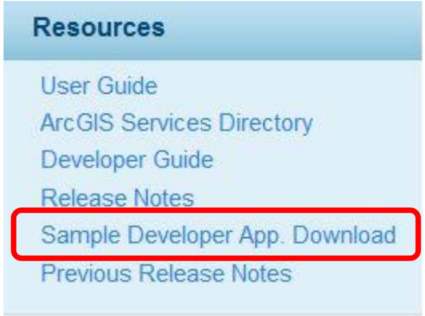
Energy Map Services

Energy Map Services is used to deliver IHS spatial data to your mapping tool of choice. Whether it's ESRI software like ArcMap, ArcGIS.com Webmap Viewer, ArcgGIS Explorer Online, or custom developed web mapping applications, Map Services enables you to easily mash up the most current IHS spatial information with your other data sources to produce compelling maps.

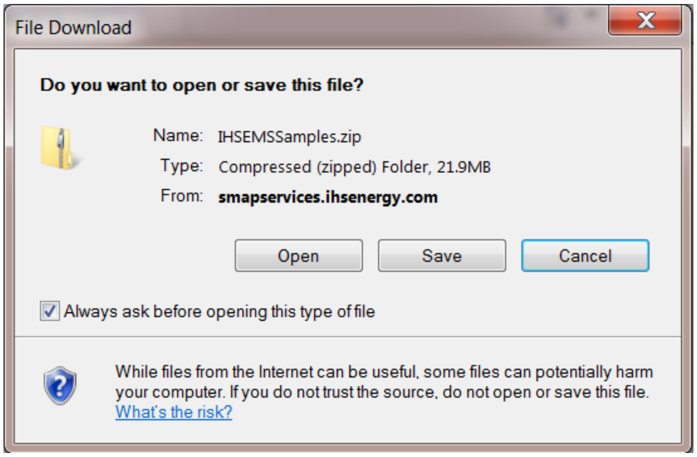
Your IHS data subscription includes Energy Map Services as another open and flexible way to connect to your IHS information for mapping workflows.

Check out IHS featured content below.

On the Home page, under Resources, click **Sample Developers App. Download**.



You will be prompted to Open or Save the download.



File Download

Do you want to open or save this file?

Name: IHSEMSSamples.zip

Type: Compressed (zipped) Folder, 21.9MB

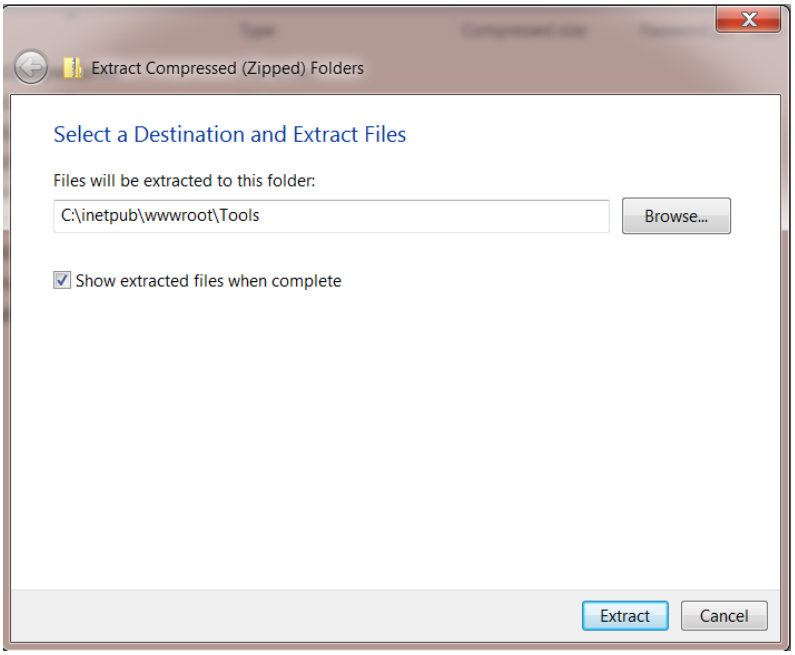
From: **smapservices.ihsenergy.com**

☑ Always ask before opening this type of file

While files from the Internet can be useful, some files can potentially harm your computer. If you do not trust the source, do not open or save this file. What's the risk?

Click **Save,** and then select a location to download the .zip file.

Double click the .zip file and select **Extract All Files**. Set the destination to a location of your choice.



Extract Compressed (Zipped) Folders

Select a Destination and Extract Files

Files will be extracted to this folder:

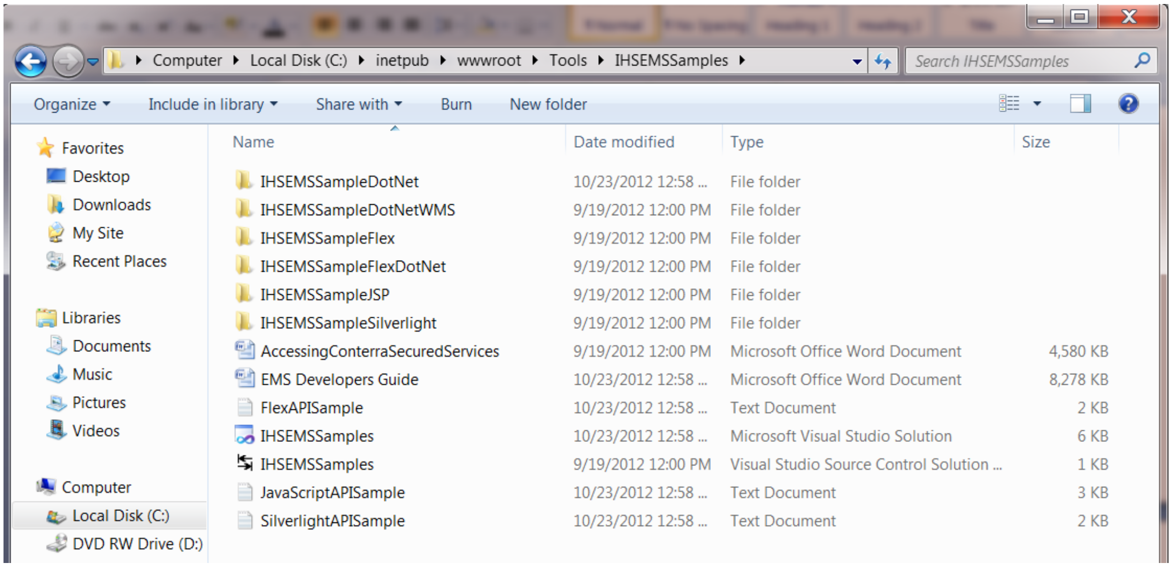
C:\inetpub\wwwroot\Tools

☑ Show extracted files when complete

For developers, set IIS to point to the correct directory.

For other users, consider putting the .zip file into the inetpub\wwwroot directory.

The .zip file contains the following set of folders and files:



Computer ▸ Local Disk (C:) ▸ inetpub ▸ wwwroot ▸ Tools ▸ IHSEMSSamples ▸

Name	Date modified	Type	Size
IHSEMSSampleDotNet	10/23/2012 12:58 ...	File folder	
IHSEMSSampleDotNetWMS	9/19/2012 12:00 PM	File folder	
IHSEMSSampleFlex	9/19/2012 12:00 PM	File folder	
IHSEMSSampleFlexDotNet	9/19/2012 12:00 PM	File folder	
IHSEMSSampleJSP	9/19/2012 12:00 PM	File folder	
IHSEMSSampleSilverlight	9/19/2012 12:00 PM	File folder	
AccessingConterraSecuredServices	9/19/2012 12:00 PM	Microsoft Office Word Document	4,580 KB
EMS Developers Guide	10/23/2012 12:58 ...	Microsoft Office Word Document	8,278 KB
FlexAPISample	10/23/2012 12:58 ...	Text Document	2 KB
IHSEMSSamples	10/23/2012 12:58 ...	Microsoft Visual Studio Solution	6 KB
IHSEMSSamples	9/19/2012 12:00 PM	Visual Studio Source Control Solution ...	1 KB
JavaScriptAPISample	10/23/2012 12:58 ...	Text Document	3 KB
SilverlightAPISample	10/23/2012 12:58 ...	Text Document	2 KB

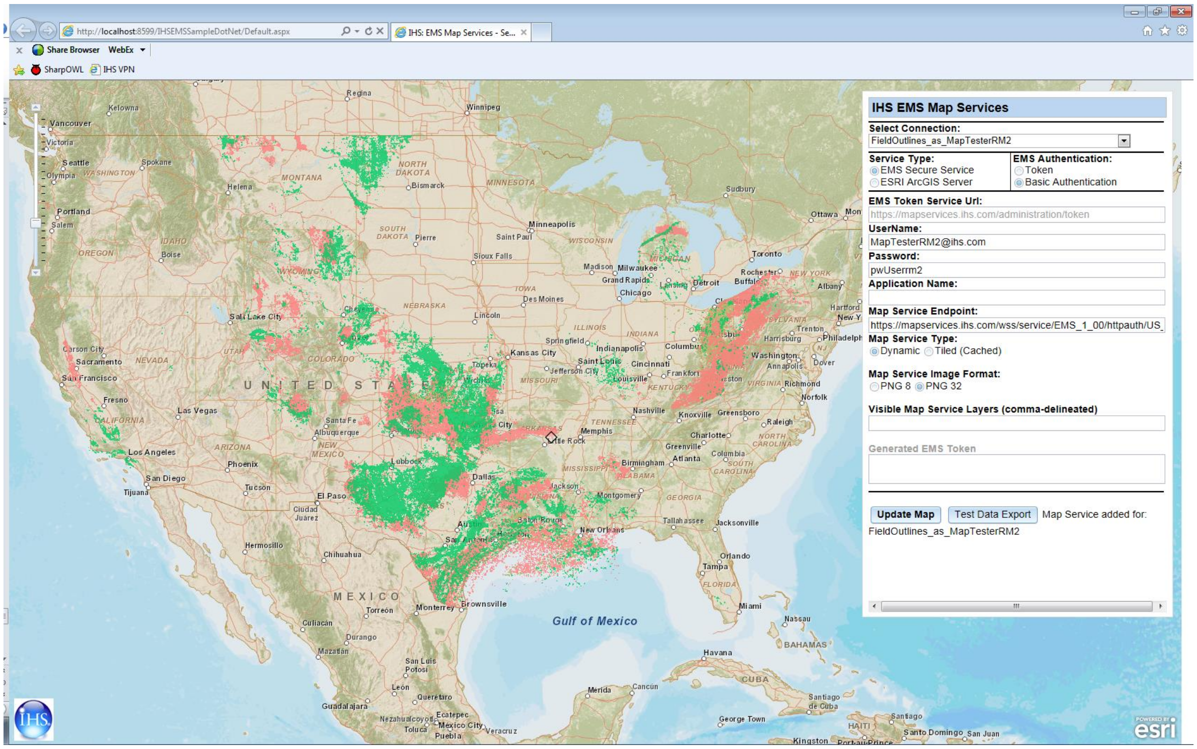
The EMS Developer’s Guide is located in this folder. It is strongly recommended you read this document before getting started with the samples.

The examples are meant for developers to quickly test, use and demonstrate connecting to Map Services. They are not complete applications meant for further customization but are meant to you help understand some of the methods that can be used to connect to and consume IHS Map Services.

JavaScript API/.NET Example

The following sample demonstrates both Basic Authentication and Token-based security when using the EMS Map Services. It uses a server-side proxy that is adapted from the freely available ESRI proxy. This sample uses configurations stored in the web.config file that the user can then update through a form using the map interface. The application supports changing multiple parameters in map tile requests, thus allowing developers to quickly test token, basic authentication, image formats, and visible layers.

Also included in this sample is a basic WMS sample that quickly and easily demonstrates how to generate a stand-alone map tile from the WMS services.


http://localhost:8599/IHSEMSSampleDotNet/Default.aspx
IHS: EMS Map Services - Se...
Share Browser
WebEx
SharpOWL
IHS VPN
UNITED STATES
MEXICO
Gulf of Mexico
esri

IHS EMS Map Services

Select Connection:
FieldOutlines_as_MapTesterRM2

Service Type:
● EMS Secure Service
○ ESRI ArcGIS Server

EMS Authentication:
○ Token
● Basic Authentication

EMS Token Service Url:
https://mapservices.ihs.com/administration/token

UserName:
MapTesterRM2@ihs.com

Password:
pwUserrm2

Application Name:

Map Service Endpoint:
https://mapservices.ihs.com/wss/service/EMS_1_00/httpauth/US_

Map Service Type:
● Dynamic ○ Tiled (Cached)

Map Service Image Format:
○ PNG 8 ● PNG 32

Visible Map Service Layers (comma-delineated)

Generated EMS Token

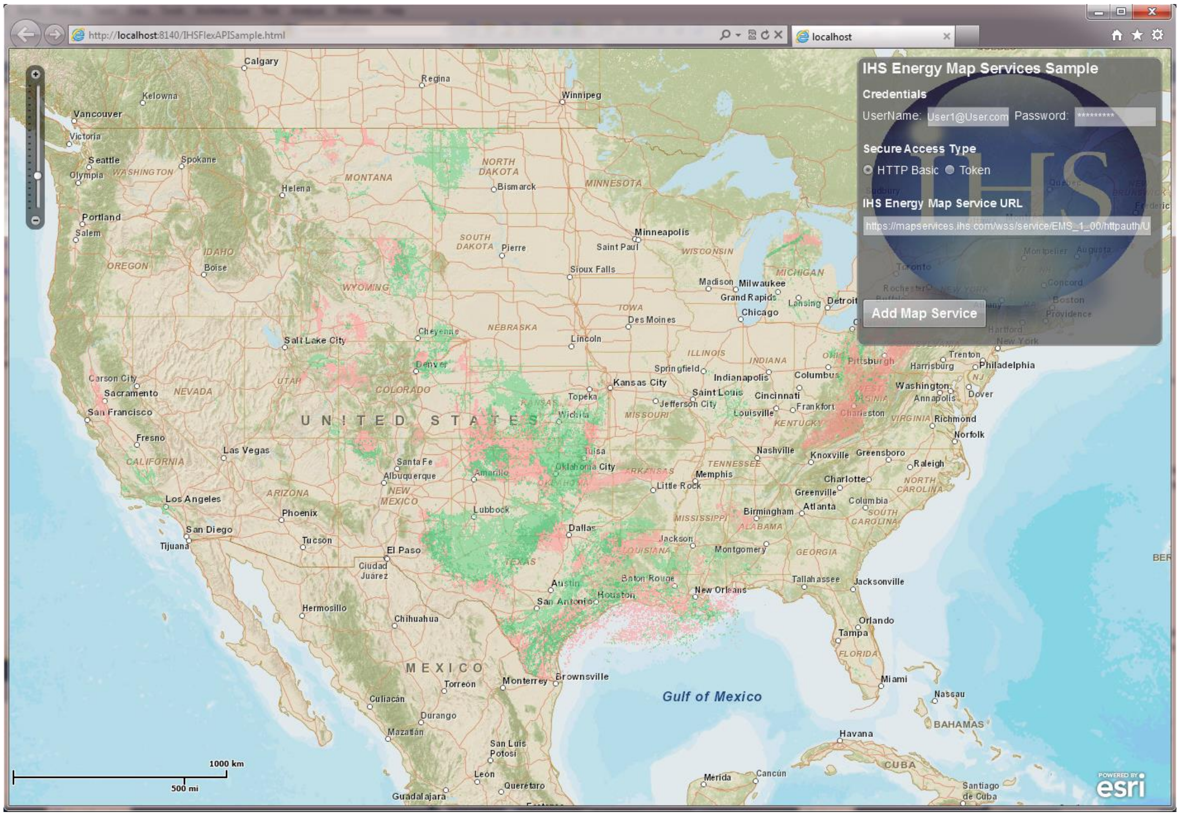
Update Map | Test Data Export | Map Service added for: FieldOutlines_as_MapTesterRM2

Figure 1 - JavaScript/.NET Example showing U.S. Fields added using Basic Authentication

Flex API/.NET Example

This sample demonstrates both Basic Authentication and Token based security when using the EMS Map Services. It includes a basic Flex API map, allowing users to add an EMS Map Service using either Token or Basic authentication.

It includes both a proxy and a token handler (generator) based on the ESRI sample proxies. The proxies and the Flex application container are .NET-based, but any server-side language can be used to support a Flex application.



http://localhost:8140/IHSFlexAPISample.html

IHS Energy Map Services Sample

Credentials

UserName: User1@User.com Password: *********

Secure Access Type

● HTTP Basic ○ Token

IHS Energy Map Service URL

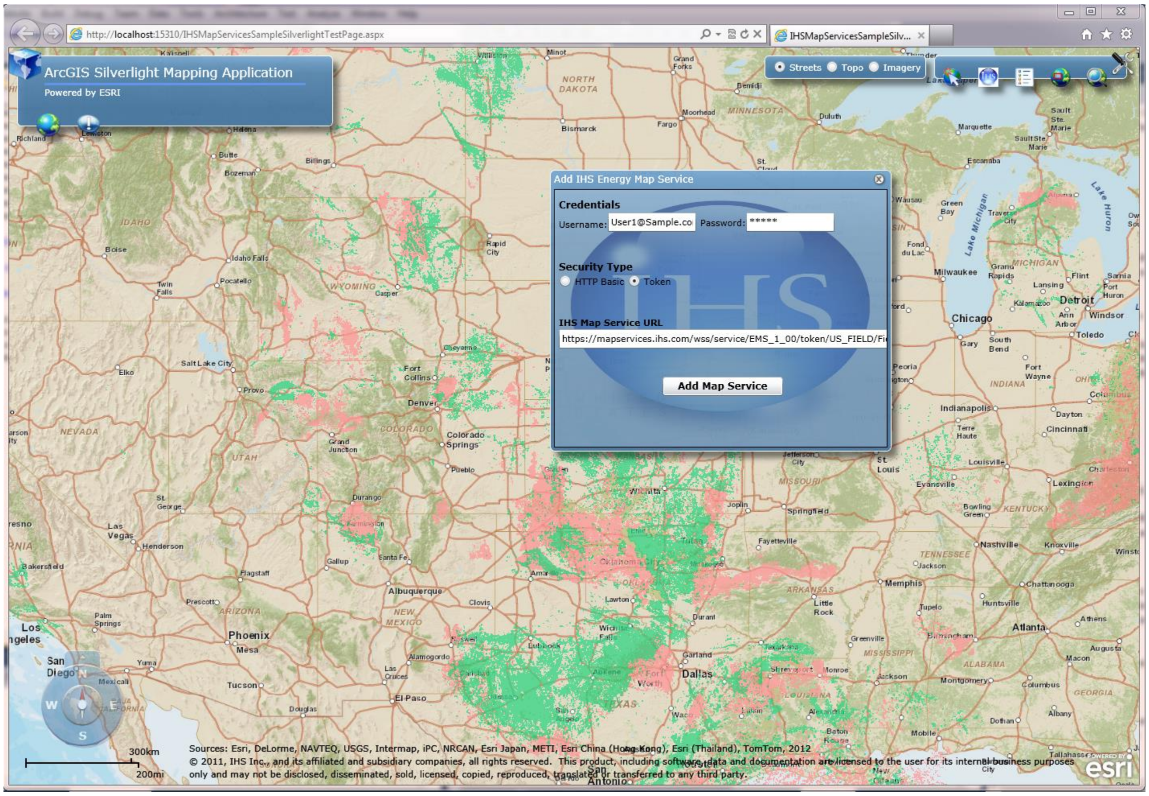
https://mapservices.ihs.com/wss/service/EMS_1_00/httpauth/U

Figure 2 – Flex/.NET example showing US Fields added using Basic Authentication

Silverlight/.NET Example

This sample demonstrates both Basic Authentication and Token-based security when using the EMS Map Services. It is based on the ESRI Silverlight template application, and allows users to add services using either Basic Authentication or Tokens.

Tokens are generated using a token handler, while communication with the server is transported through a proxy based on the sample ESRI proxy.



ArcGIS Silverlight Mapping Application
Powered by ESRI
● Streets ○ Topo ○ Imagery
IHSMapServicesSampleSilv...
300km
200mi
esri

http://localhost:15310/IHSMapServicesSampleSilverlightTestPage.aspx

Add IHS Energy Map Service

Credentials

Username: User1@Sample.co Password: *****

Security Type

○ HTTP Basic ● Token

IHS Map Service URL

https://mapservices.ihs.com/wss/service/EMS_1_00/token/US_FIELD/Fi

Add Map Service

Sources: Esri, DeLorme, NAVTEQ, USGS, Intermap, iPC, NRCAN, Esri Japan, METI, Esri China (Hong Kong), Esri (Thailand), TomTom, 2012

© 2011, IHS Inc., and its affiliated and subsidiary companies, all rights reserved. This product, including software, data and documentation are licensed to the user for its internal business purposes only and may not be disclosed, disseminated, sold, licensed, copied, reproduced, translated or transferred to any third party.

Figure 3 - Silverlight Example showing US Fields added with token based authentication