

The **RedSpider** Line



**Output Formats
Support**

Output Formats - Version: 2.0 - August 07

Edited on: 03/09/2007

IONIC Software
Rue de Wallonie 18
4460 Grâce-Hollogne (Liège)
Belgium
Tel : +32.4.364.03.64
Fax : +32.4.253.47.37
info@ionicsoft.com
www.ionicsoft.com

Sommaire

Sommaire	3
1.1 Output Formats	4
1.2 Overview	4
1.3 Image Outputs	4
1.3.1 Graphic Interlaced Format (GIF)	4
1.3.2 Joint Photographic Experts Group (JPEG)	5
1.3.3 Keyhole Markup Language (KML)	5
1.3.4 Scalable Vector Graphics (SVG)	6
1.3.5 GeoTIFF	7
1.3.6 Portable Network Graphic (PNG)	8
1.3.7 X-BMP	9
1.3.8 WBMP	9
1.4 Text Outputs	10
1.4.1 Plain Text Output	10
1.4.2 HTML	10
1.4.3 GeoRSS	11
1.5 Data Outputs	11
1.5.1 Shapefiles	11
1.5.2 GML 2/3	12
1.5.3 GeoTIFF	12
1.5.4 JPEG2000, ECW, NITF, DTED	12



1.1 Output Formats

In this section, we will discuss various types of:

- **Image** Outputs
- **Textual** Outputs
- **Data** Outputs

1.2 Overview

RedSpider Web™ and **RedSpider ImageArchive** support the output of data into various file formats for further use or for data sharing purposes. The product allows output in a variety of formats including images, text and HTML. In order to output data, the output format must be included in the HTTP request for a map, feature or coverage.

1.3 Image Outputs

The **RedSpider Web** and **RedSpider ImageArchive** products support a variety of different image outputs. All image outputs can be initiated from a GetMap request on either a raster, vector or coverage WMS, or a GetView request on a WTS.

1.3.1 Graphic Interlaced Format (GIF)

GIF is the most common format used on the Internet and is best for simple graphics, i.e., line art and simple images with large blocks with a few colors. GIF files are good for representing graphics, as opposed to JPEG or other image format types, because the file size is small and of a better quality. A GIF file can handle only 256 colors which makes it inappropriate for photo images. GIFs work well for images like company logos or screen shots. These images should be reduced to 16 colors, if possible, and saved as a GIF.

In order to use the GIF option for output format, a valid GIF license is required. IONIC is not liable for the use of GIF format by its customers without a valid GIF license. IONIC **RedSpider Web** and **RedSpider ImageArchive** only support the version 87a of GIF. This means that transparency and simple animations are not supported.

Copy and paste the example provided below in the Service Tester for a GetMap request in GIF format.

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
```



```
FORMAT=image/gif
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
```

1.3.2 Joint Photographic Experts Group (JPEG)

The Joint Photographic Experts Group (JPEG) is an organization that sets standards for graphic file formats. JPEG is a compressed format, with some loss of quality due to compression. JPEGs are best for photos because the file size is small and there is no limit to the number of colors used. Other file extensions used are .jpg, .jpeg, and .jpe.

RedSpider Web and **RedSpider** Image Archive support a QUALITY parameter in the GetMap request that sets the compression ratio between 0 and 100, 0 being the maximum.

Copy and paste the example provided below in the Service Tester for a GetMap request in JPEG format.

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/jpeg
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
QUALITY=70
```

1.3.3 Keyhole Markup Language (KML)

KML is a file format used to display geographic data in an Earth browser, such as Google Earth, Google Maps, and Google Maps for mobile. KML uses a tag-based structure with nested elements and attributes and is based on the XML standard.

KMZ files are KML files, sometimes along with raster images, the whole being compressed using ZIP compression technology.

IONIC **RedSpider** Web and **RedSpider** Image Archive provide several ways to produce KML and KMZ documents, depending on the nature of the data you want to output.

- For raster and coverage data (the Map and Coverage servlets), you can request KML to obtain a light document containing the URL to a GetMap request with a raster output format. You can also request KMZ, in which case the zip contains a light KML document and an image which is the output of a GetMap in PNG.
- For vector data sets (the WFS servlet), you can only request KML, KMZ being dedicated to raster data sets. As soon as the portrayal styles have been created in order to enable the WMS interfaces out of your servlet, a GetMap request will convert the data into graphics, like for the raster formats, and then convert it to KML. This means that few data attributes will be available, due to the late stage of this conversion.



To benefit from the whole power of the KML output, a specific Java rule can be written, compiled and uploaded on the server. Such a rule can be easily written thanks to a light Java API and a set of helper classes - see IONIC **RedSpider** Enterprise for more detail on this API.

Starting with **RedSpider** Web 3.4.2 and **RedSpider** ImageArchive 3.1, the KML and KMZ formats are published in the Capabilities document for the WMS interface, either as their format names "KML" or "KMZ" or as their mime-types "application/vnd.google-earth.kml+xml" or "application/vnd.google-earth.kmz", depending on the version of the WMS specification.

1.3.4 Scalable Vector Graphics (SVG)

SVG is an XML grammar used for modeling graphics. It differs from the GIF and JPEG in that it uses graphic objects rather than individual points. SVG is also a scalable format. This means that a graphic can be rendered at differing resolutions. Refer to the "**RedSpider** Product Line Concepts Guide" for additional information.

IONIC **RedSpider** Web supports three implementations of SVG.

- The default method of implementing SVG output is to issue a format=image/svg+xml (or format = SVG in older versions of the WMS spec) and the application will return an XML document that can be read by any W3C compliant viewer.

Warning

This option may produce outputs not readable in an Adobe SVG reader. The Adobe SVG Reader 3.0 is not completely compliant with the W3C specification and does not support base64 encoded content using the "data" protocol for SVG images. This basically means that Adobe will not support a data URL for embedded SVG image files, but will support embedded raster symbols.

If the output contains SVG embedded symbols or pattern fills, it cannot be viewed in the 3.0 version of Adobe. Convert the embedded symbols into pure raster format, GIF or PNG, since Adobe will support embedded symbols in this format or use other IONIC implementations that support non-compliant SVG viewers.

- The work-around to non-compliant SVG viewers is SVG output without embedded symbols. Simply edit the feature mapping file and add the "DontEmbedSVG" option. The procedure is outlined below:
 1. Locate the feature mapping file. The default location is the providers.fac file directory WEB-INF/classes/com/ionicsoft/wfs/server/resource. Open it in a text editor.
 2. Locate the "Option" section normally found before the beginning of the <Mapping> elements.
 3. Add a new Option listing as follows:
 4. <Option>
 5. <DontEmbedSVG>true</DontEmbedSVG>
 - </Option>
 6. Save the file and if necessary and re-start the servlet.



This option will now use HTTP references for linked or embedded symbols instead of the data URL. This means that all embedded symbols will now appear as HTTP references that the client must download to bring into the desired output.

Warning

Ensure that all embedded symbol files are downloaded and stored in the same directory or subdirectory as the main SVG document.

Warning

The Java rule developer must insure that the rule created relies on the Web servlet container that has this option in the providers.fac file.

- IONIC has also created a new mime type format for SVG that returns a zipped document that contains the main SVG document and all embedded or linked files. This option supports local SVG output. To use this option, use the parameter `format=image/svg+zip` or `format = SVG_ZIP` in older versions of WMS and the application will return a zipped file. Unzip the file and place the main SVG document and it's corresponding files into the same directory. Any W3C compliant SVG viewer can read the SVG file.

Note

If the document contains embedded SVG symbols, the output will not work in Adobe 3.0. Either convert the embedded symbols to pure raster format or ensure that the SVG document contains relative HTTP reference links. Links to embedded symbols using a data URL will not work in Adobe.

Due to Adobe SVG Viewer limitations, text rendered with a Halo will not display a complete image in Adobe SVG Viewer. Also, any style or rule producing one of the SVG codes Adobe mentions as non-supported will produce an unreadable file. See Adobe limitations at <http://www.adobe.com/svg/indepth/releasenotes.html>.

1.3.5 GeoTIFF

"GeoTIFF" refers to TIFF files which have geographic or cartographic data embedded as tags within the TIFF file. The geographic data, mainly the SRS and the extent in the header file, can be used to position the image in the correct location and geometry on the screen of a geographic information display.

IONIC Software offers full support of the GeoTIFF image format. Several data providers are committed to delivering imagery in GeoTIFF format including SPOT Image Corp, Trifid (representing Landsat data), Space Imaging, US Geological Survey, and the New York Department of Transportation. In addition, the United Kingdom Military Survey has announced it is testing the format for their products.

The following request shows how to return a GeoTIFF image from an IONIC WMS:

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
```



```

BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/tiff
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml

```

1.3.6 Portable Network Graphic (PNG)

PNG is a file format for image compression that, in time, is expected to replace the Graphics Interchange Format (GIF) that is widely used on the Internet. The PNG format was expressly developed to be patent-free. A PNG file is compressed in "lossless" fashion meaning all image information is restored when the file is decompressed during viewing. PNG includes the following upgrades from the GIF format:

- Degree of opacity (transparency)
- Interlacing
- Gamma correction
- True color or GIF palettes

Currently, the PNG format is not widely used, although major Internet browser providers have pledged to support the format in future versions.

The following request will return output in PNG format:

```

http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/png
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
QUALITY=30

```

RedSpider Web and **RedSpider** ImageArchive support PNG output in 8 or 24-bits. To do so, add the "QUALITY" parameter to a request, with a value between 0 and 50 for 8-bits PNG and between 51 and 100 for 24-bits PNG.

Configure any provider to produce PNG of a specific quality.

- For a WMS provider, raster or proxy-WMS, add a PARAMBLOCK tag named "quality" that holds a "PNG" parameter:
 - `<PARAMBLOCK NAME="quality">`
 - `<PARAM NAME="PNG" VALUE="30" />`
 - `</PARAMBLOCK>`
- For a WFS provider, the configuration is in the mapping file under the `<Option>` tag.



Create a <Generate8BitsPNG> tag containing "true" or "false" as in the following example:

- <Option>
- ...
- <Generate8BitsPNG>true</Generate8BitsPNG>
- ...
- </Option>

1.3.7 X-BMP

X-BMP is the default Windows BMP format. The following example is a GetMap request that returns an X-BMP.

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/x-bmp
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
```

1.3.8 WBMP

A Wireless Bitmap (WBMP) is a graphic image format for sending Web content to handheld wireless devices. The format is defined in the Wireless Application Protocol (WAP), Wireless Application Environment (WAE) Specification. For Web content that is directed to handheld phones or personal digital assistants (PDA) that have Web access, use the Wireless Markup Language (WML) to encode the page and its text. An image converted from a GIF, TIFF, or other graphic format can be included in the form of a WBMP file. The initial WAP WAE specification supports only WBMP type 0 that is a compression image in monochrome. As the bandwidth for wireless transmission increases, richer images will be supported.

The following request returns a WBMP image format:

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetMap
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/vnd.wap.wbmp
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
```



1.4 Text Outputs

RedSpider Web and **RedSpider** ImageArchive support the following types of text output.

1.4.1 Plain Text Output

To produce plain text output from an IONIC WMS or WFS, add the INFO_FORMAT parameter to the GetFeatureInfo request. This will return either plain text or Comma Delimited Tabs (CSV). The content depends on the connector type. The following is a GetFeatureInfo request that returns text output.

```
http://localhost:8080/ionicweb/map/BOSTON_LI?
VERSION=1.1.1
REQUEST=GetFeatureInfo
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=BOSTON_LI
STYLES=default
FORMAT=image/gif
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
QUERY_LAYERS=BOSTON_LI
INFO_FORMAT=text/plain
X=200
Y=220
```

1.4.2 HTML

The INFO_FORMAT parameter of the GetFeatureInfo request will also return output in HTML format. For vector data, the output will be a simple but smart HTML page that contains a header, body and footer text, a logo and title. The output will also contain a custom style sheet (CSS) that allows flexible configuration.

HTML output is accessible from the StyleEditor™. StyleEditor™ creates styles for HTML output that can be used in the Portrayal Engine. Refer to the "StyleEditor User Guide" for additional information on how to access this functionality.

Following is a GetFeatureInfo request that returns HTML output.

```
http://localhost:8080/ionicweb/wfs/BOSTON_SHAPE?
VERSION=1.1.1
REQUEST=GetFeatureInfo
WIDTH=400
HEIGHT=400
SRS=EPSG:26986
BBOX=225000,886000,237000,902000
LAYERS=protectedareas
STYLES=default
FORMAT=image/gif
BGCOLOR=0xFFFFFFFF
TRANSPARENT=FALSE
EXCEPTIONS=application/vnd.ogc.se_xml
QUERY_LAYERS=protectedareas
```



```
INFO_FORMAT=text/html
X=200
Y=220
```

1.4.3 GeoRSS

RSS (Rich Site Summary) is an XML format for delivering regularly changing web content. Many news-related sites, weblogs and other online publishers syndicate their content as an RSS Feed to whoever wants it.

GeoRSS is an emerging standard for encoding location as part of a RSS feed (see <http://www.georss.org> for in-progress work on GeoRSS).

IONIC **RedSpider** Web™ supports GeoRSS as output of GetFeature requests on vector data sets (the WFS servlet). It produces RSS 2.0 documents, with both GeoRSS Simple and GeoRSS GML outputs for the geometries. The sample output below shows a GeoRSS output by IONIC **RedSpider** Web 3.4.2.

```
<?xml version='1.0' encoding='utf-8' ?>
<rss version="2.0" xmlns:georss="http://www.georss.org/georss">
  <channel>
    <title>LocalName</title>
    <link>http://www.ionicsoft.com</link>
    <description>no description</description>
    <item>
      <title>CAMBRIDGE</title>
      <georss:where>
        <Point xmlns="http://www.opengis.net/gml" srsName="EPSG:26986">
          <pos>232226.47 901710.31</pos>
        </Point>
      </georss:where>
      <georss:point>42.36522907219097 -71.10877119738284</georss:point>
    </item>
  </channel>
</rss>
```

1.5 Data Outputs

1.5.1 Shapefiles

Shapefile is the most commonly used format for exchanging GIS data. **RedSpider** Web™ supports shapefile output in zip format. To obtain shapefile output, append the "outputFormat=SHAPE" parameter to a WFS GetFeature request. Following is a GetFeature request that returns Shape output:

```
<?xml version="1.0" encoding="UTF-8" ?>
<ogcwf:GetFeature maxFeatures="20"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:ogcwf="http://www.opengis.net/wfs"
  version="1.0.0"
  service="WFS"
  outputFormat="SHAPE" >
  <ogcwf:Query typeName="place_names">
    </ogcwf:Query>
</ogcwf:GetFeature>
```



1.5.2 GML 2/3

GML is an open, non-proprietary language used to create geo-spatial objects for the purpose of data sharing. GML serves as a data transport for geo-spatial objects as well as providing a means for describing geo-spatial Web services. GML is constantly evolving and has quickly become the standard geo-spatial information (GI) format for all products that are based on international GI standards.

GML2 is the default output format of the IONIC WFS 1.0.0 GetFeature request but GML2 output can also be explicitly requested by appending the "outputFormat=GML2" parameter to a WFS GetFeature request.

GML3 format is also supported by IONIC WFS. Request a GML3 output by appending the "outputFormat=GML3" parameter onto a WFS 1.0.0 GetFeature request. Starting with **RedSpider** Web 3.3, the OGC WFS 1.1.0 specification is supported. If a WFS 1.1.0 GetFeature request is sent, the default output format is GML3. This is only possible if the WFS provider is configured as a GML3 one. The feature types schema must include the GML 3.1.1 feature.xsd schema and the schema must validate against the GML 3.1.1 schemas. More information on setting up a GML3-compliant WFS is given in Appendix E "WFS Schema and Mapping Configuration", Section 4 "Moving to GML3".

Note that default outputFormat behavior can be overridden by defining the `<GMLOutputFollowModel>true</GMLOutputFollowModel>` in the `<Option>` section of the mapping file. Setting it to true will lead to output being driven by the feature model used to set up the WFS.

For more information about GML, please refer to the "**RedSpider** Product Line Concepts Guide".

1.5.3 GeoTIFF

When requesting coverages from a Web Coverage Service (WCS) or from an Image Archive Service (IAS), the coverages can be output in GeoTIFF format. This output is not a simple image, that is restricted to 3 or 4 bands of 8-bit integer values. Coverage data can be a 8-, 16- or 32-bit integer, signed or unsigned, and a 16-, 32- or 64-bit floating point.

Coverage data values do not represent pixel luminescence (red, green, blue, cyan, magenta, yellow and black), but a physically measured value (32-bit elevation float data, or 16-bit temperature short data).

A GeoTIFF cannot be viewed using standard image applications.

Coverage GeoTIFFs can have one to n bands, but the WCS output is currently limited to one, three or four bands.

1.5.4 JPEG2000, ECW, NITF, DTED

In a limited set of situations, **RedSpider** Web and **RedSpider** Image Archive allow data to be output in several other formats, thanks to the GDAL library.

The conditions to be able to produce JPEG2000, ECW, NITF or DTED output are:



- The service must be an IONIC WCS or IAS servlet. It is recognized as the path in the URL contains "coverage" or "ias". Example:
http://localhost:8080/ionicweb/coverage/BOSTON_SC.
- The type of the provider must be SimpleProvider, MultiSimpleProvider, IndexProvider or HierarchicalProvider.
- The request must be a WCS GetCoverage.
- For some formats like ECW, the appropriate proprietary library has to be linked with GDAL in order to be served. However they are not available on all platforms. Please refer the Administrator's Guide "Provider Types" appendix, Table "GDAL-based Source Formats by Platform" for more details on formats served via GDAL.

To make sure the service can produce a coverage in one of those formats, run a DescribeCoverage command on that tile. At the bottom of the output document, there is a XML section similar to the example below:

```
<supportedFormats>  
  <formats>GeoTIFF</formats>  
  <formats>DTED</formats>  
  <formats>ECW</formats>  
  <formats>JPEG2000</formats>  
  <formats>NITF</formats>  
</supportedFormats>
```

