# USGIN Glossary

This page defines terms as they are used within the context of the United States Geoscience Information Network (USGIN). Instanced terms that are used only in a specialized context (such as a specific tutorial or a specific AASG Geothermal Data content model) are not defined here, but are instead defined within the specific instance. Likewise, geoscience vocabulary is not defined here.

| **Term** | **Definition** |
| --- | --- |
| [ArcGIS](http://usgin.org/glossary/arcgis) | Proprietary [geographic information system (GIS)](http://usgin.org/glossary#gis) software created by [ESRI](http://www.esri.com/). |
| [Attribute (GIS)](http://usgin.org/glossary/attribute-gis) | Within the context of [geographic information systems](http://usgin.org/glossary#gis), attributes describes [features](http://usgin.org/glossary#feature).  For example, attributes describing a fault feature might include:   * The latitude and longitude coordinates of the fault * The age of the fault * The fault's dip and slip   Attributes are often stored in [databases](http://usgin.org/glossary#database) and subdivided into [records](http://usgin.org/glossary#database_record) and [fields](http://usgin.org/glossary#database_field); consequently, attributes can also be expressed as [markup language](http://usgin.org/glossary#markup_language) [elements](http://usgin.org/glossary#element). |
| [Attribute (Markup Language)](http://usgin.org/glossary/attribute-markup-language) | In XML documents, attributes modify markup language [elements](http://usgin.org/glossary#element), like so:  <element attribute="value" />  For a more detailed discussion, see the [USGIN XML Tutorial](http://usgin.org/content/xml-tutorial/). |
| [Binding](http://usgin.org/glossary/binding) | An explicit logical association between any two things. For example, a binding can exist between two resources; a binding can also exist between a location and the [resource](http://usgin.org/glossary#resource) found there. |
| [Capabilities Document](http://usgin.org/glossary/capabilities-document) | A capabilities document is an [XML](http://usgin.org/glossary#xml) document that describes the capabilities of a [web service](http://usgin.org/glossary#web_service). |
| [Content Model](http://usgin.org/glossary/content-model) | Within the context of USGIN, the [National Geothermal Data System (NGDS)](http://www.geothermaldata.org/), and the [AASG Geothermal Data project](http://www.stategeothermaldata.org), content models are Excel workbooks which contain template spreadsheets.  Content models provide NGDS [schemas](http://usgin.org/glossary#schema) that are designed to facilitate [interoperability](http://usgin.org/glossary#interoperability). Any data submitted for the AASG Geothermal Data project by [Arizona Geological Survey](http://azgs.az.gov) [subrecipients](http://usgin.org/glossary#subrecipient) must be structured by an NGDS schema provided by an NGDS content model. |
| [Data](http://usgin.org/glossary/data) | Data describes something in the real world. In the USGIN framework, data often describes a [resource](http://usgin.org/glossary#resource).  Though the terms data and information are often used interchangeably, it should be noted that data technically indicates raw observations; information connotes interpreted observations. A discrete cluster of related data is known as a [dataset](http://usgin.org/glossary#dataset). |
| [Database](http://usgin.org/glossary/database) | A method of storing [data](#data). In a database, data is divided up into [database records](http://usgin.org/glossary#database_record); in turn, database records are divided up into [database fields](http://usgin.org/glossary#database_field). The advantage of a database is that it can be sorted and searched by field contents.  Though modern databases are usually digital, a physical example of a database is a card catalog in a public library. In a card catalog, data is divided up into individual cards, which are directly analagous to database records. Each card (record) in the catalog corresponds with and describes a book. The information about each book is divided up into fields: title; author; subject; publication date; etc.  Digital databases can be in tabular format (that is, a table) in which rows represent individual records and columns constitute fields; or they can be viewed record-by-record. |
| [Database Field](http://usgin.org/glossary/database-field) | A subdivision of a [database record](http://usgin.org/glossary#database_record) in which a specific type of [data](http://usgin.org/glossary#data) is entered.  Using the analogy of a card catalog in a public library: if the card catalog is directly analagous to a [database](http://usgin.org/glossary#database), and if the cards in the catalog are directly analagous to [database records](http://usgin.org/glossary#database_record), then the different subdivisions of information found in each card in the catalog (title, author, publishing date, etc.) are all database fields.  Database fields are functionally equivalent to [markup language](http://usgin.org/glossary#markup_language) [elements](http://usgin.org/glossary#element). |
| [Database Key](http://usgin.org/glossary/database-key) | A [database field](http://usgin.org/glossary#database_field) designed to contain values that are used to organize and maintain the uniqueness of records within a [database](http://usgin.org/glossary#database). Database keys [identify](http://usgin.org/glossary#identifier) [records](http://usgin.org/glossary#database_record) in such way that they can be referenced by other databases; consequently, database keys allow databases to refer to records in other databases. |
| [Database Record](http://usgin.org/glossary/database-record) | A subdivision of a [database](http://usgin.org/glossary#database). Using the analogy of a card catalog in a public library, each record in a database is analogous to an index card in the card catalog; each card (record) corresponds with and describes an individual book. Database records are further subdivided into [fields](http://usgin.org/glossary#database_field), which contain specific kinds of data. |
| [Database View](http://usgin.org/glossary/database-view) | In the context of a [database](http://usgin.org/glossary#database), a view is a selection of [fields](http://usgin.org/glossary#database_field). For example: given a database with twelve fields, an arbitrary grouping of four such fields would constitute a view of the database.  A more concrete example is a card catalog in a public library: if one chose to look only at the **Title** field of each card in the catalog, the act of doing so would constitute a discrete view of the catalog. |
| [Dereference](http://usgin.org/glossary/dereference) | Verb. To display that which is referenced.  For example: academic papers and articles often cite other papers or articles as sources; the act of following one of these references and displaying the source document is the act of dereferencing.  In the context of the World Wide Web, dereferencing usually involves the act of displaying the document to which a given hyperlink refers. |
| [Element](http://usgin.org/glossary/element) | Elements are logical document components found in [markup languages](http://usgin.org/glossary#markup_language) such as [XML](http://usgin.org/glossary#xml) and [HTML](http://usgin.org/glossary#html). Elements simultaneously define the structure and content of a document. An element is demarcated by markup language tag*s*. For example:  <tag>content</tag>  The first tag opens the element; the second tag closes it; everything between the opening tag and the closing tag constitutes the content of the element.  In a more concrete example, HTML uses the <em> element to demarcate text that should be emphasized. So, an emphasized element of an HTML document would appear as follows:  <em>Emphasized content</em>  In a web browser, this element would produce the following result:  Emphasized content  Markup language elements do not always need an opening tag and a closing tag because some elements can close themselves by including a **space** followed by a **forward slash (/)** within the element. For example, the following element is self-closing:  <tag />  Markup language elements are functionally similar to [database fields](http://usgin.org/glossary#database_field), in that both serve to subdivide the content of a document.  For a much more detailed overview of elements, see the [USGIN XML Tutorial](http://usgin.org/content/xml-tutorial/). |
| [Feature](http://usgin.org/glossary/feature) | The word feature may be used to indicate a geologic feature, such as a fault, formation, or dike. Alternatively, a feature can be a [GIS](http://usgin.org/glossary#gis) representation of a real-world object on a map.  GIS features do not always correspond with geologic features because GIS software can be used to represent anthropogenic objects such as buildings, roads, or canals. The definition of the term feature therefore depends on its context: it can be used to mean either a geologic feature or an GIS feature.  GIS features are often described by [attributes](http://usgin.org/glossary#attribute). |
| [Feature Class](http://usgin.org/glossary/feature-class) | A feature class can be either a method of storing [GIS](http://usgin.org/glossary#gis) [features](http://usgin.org/glossary#feature) of the same geometry (point, line, or polygon); or it can be a discretionary or subjective grouping of homogenous GIS features. For example, "highways, primary roads, and secondary roads can be grouped into a line feature class named "roads." |
| [Geographic Information Systems (GIS)](http://usgin.org/glossary/geographic-information-systems-gis) | A system designed to capture, store, manipulate, analyze, manage, and present all types of geographically referenced [data](http://usgin.org/glossary#data). |
| [HTML](http://usgin.org/glossary/html) | HTML stands for Hypertext [Markup Language](http://usgin.org/glossary#markup_language). HTML is the predominant language in which web pages are written. |
| [HTTP](http://usgin.org/glossary/http) | HTTP stands for Hypertext Transfer [*Protocol*](http://usgin.org/glossary#protocol). HTTP is the networking protocol that is used to transfer information over the World Wide Web.  HTTP defines four basic operations (requests) made by clients to [servers](http://usgin.org/glossary#server):   * Get * Put * Post * Delete   These HTTP requests correspond to standard [database](http://usgin.org/glossary#database) CRUD operations:   * Create (corresponds with Put) * Retrieve (corresponds with Get) * Update (corresponds with Post) * Delete (corresponds with Delete)   HTTP also defines a variety of header parameters that may be included with requests; these header parameters specify language, desired media type for response, character encoding, time stamps for resources, etc.  In addition, HTTP defines a collection of codes automatically used in response to HTTP requests; these codes indicate various success, error, or redirect conditions. A particularly infamous HTTP code is **404: Not Found**.  HTTP is defined by an Internet Engineering Task Force (IETF) Request for Comment (RFC) document: <http://tools.ietf.org/html/rfc2616>. |
| [Identifier](http://usgin.org/glossary/identifier) | An identifier is a label that is used to systematically distinguish one thing from another. To identify something is to give it a systematic label that distinguishes it from other things.  Functionally, identifiers can be compared to names. We give people, places, and things names to distinguish them from one another. What differentiates names from identifiers is the fact that identifiers are systematic: identifiers are coined according to a sytem of logical rules.  A [URI](http://usgin.org/glossary#uri) is a specific kind of identifier. |
| [Interchange Format](http://usgin.org/glossary/interchange-format) | Interchange formats are file formats and [schemas](http://usgin.org/glossary#schema) that can be used to exchange data between hardware platforms and software applications, regardless of platform or application configuration. A useful example can be found in modern printers: files sent to the printer are exported in a format that all printers can read; this format constitutes an interchange format.  Interchange formats facilitate [interoperability](http://usgin.org/glossary#interoperability) because they allow developers to design hardware and software to interact with the interchange format, as opposed to interacting directly with other hardware and software platforms. This cuts down on the need for developers to future-proof their products and allows data available in an interchange format to remain live and viable on older hardware and software platforms.  From a technical perspective, there is some overlap between an interchange format and a schema: it can be said that a given interchange format is, in fact, a schema, whereas the file itself (the container format the data, such as an XML document) is an implementation of the interchange format.  As [web services](http://usgin.org/glossary#web_service) are used for data exchange, web-accessible data that is formatted and structured for deployment as a web service can be said to be an implementation of an interchange format. Likewise, data that has been conformed to an application-neutral schema or file format constitutes an interchange format. |
| [Interoperability](http://usgin.org/glossary/interoperability) | The capability to communicate, execute programs, or transfer [data](http://usgin.org/glossary#data) among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units. Often, interopability is facilitated by structuring data in such a way that it is consistently machine-processable without user interpretation or input.  For more information on interoperability as a goal of USGIN, see the [USGIN Objectives](http://usgin.org/content/usgin-objectives#interoperability) page. |
| [Mapping (Data)](http://usgin.org/glossary/map-data) | In the context of data, mapping is the process of interpreting and restructuring data. Often, data mapping takes place from one [schema](http://usgin.org/glossary#schema) to another, a process referred to as schema mapping.  Schema mapping is typically accomplished by conforming data to fit the structure of a given document. A simple example of schema mapping is the conversion of dates in a given document from the **MM-DD-YYYY** format to the **YYYY-MM-DD** format. Another example would be the conversion of units of measure from inches to meters, or converting unit notation from millimeters to mm.  Often, schema mapping is slightly more complex than the above examples would indicate. Sometimes, data must be mapped from a single field into multiple fields; here, an example would be the act of mapping dates from a single **Date** field into three separate fields corresponding with **Day**, **Month**, and **Year**. Likewise, schema mapping sometimes involves combining data from multiple fields into one field.  The word mapping can also be used as a noun. A mapping is an instance in which data has been mapped from one schema to another. |
| [Markup Language](http://usgin.org/glossary/markup-language) | Markup languages, such as [HTML](http://usgin.org/glossary#html) and [XML](http://usgin.org/glossary#xml), use [elements](http://usgin.org/glossary#element) to structure documents in such a way that the structure of the document is visible and readily distinguishable from the content of the document. Consequently, markup language documents can be used to store data, because the visible structure of the document permits users to subdivide data into elements in much the same way that data in a [database](http://usgin.org/glossary#database) is subdivided into [records](http://usgin.org/glossary#database_record) and [fields](http://usgin.org/glossary#database_field). |
| [Metadata](http://usgin.org/glossary/metadata) | Literally, "beyond [data](http://usgin.org/glossary#data)," metadata is often conceived as "data about data."  Any data used to organize, categorize, locate, or discover something is metadata. Because metadata is merely data that is used to find something, metadata can be (and often is) stored in [databases](http://usgin.org/glossary#database).  For more information about metadata, see the [USGIN Metadata Tutorial](http://usgin.org/content/usgin-metadata-tutorial). |
| [Open-Source](http://usgin.org/glossary/open-source) | Software can be considered open-source when it complies with the crieteria of the [Open Source Initiative](http://www.opensource.org). Briefly summarized: to be considered open-source, the software or license must...   1. ...be distributable and redistributable free of charge 2. ...be distributed with uncompiled source code 3. ...allow modifications and derived works 4. ...restrict modifications and derived works only in the event of further development by the original author 5. ...not discriminate against any person or group 6. ...not discriminate against any field, profession, or endeavor 7. ...be usable without acquisition of an additional license 8. ...not be restricted to use as part of a larger software package 9. ...not restrict the use of other software 10. ...not be predicated on any individual technology or style of interface   A more detailed list of these conditions may be found [here](http://www.opensource.org/docs/osd). |
| [Profile](http://usgin.org/glossary/profile) | A profile is a limited, specific implementation of a standard. Standards often permit a wide array of possible implementations; profiles implement a specific configuration of values selected from the range of values provided by the standard.  For example: MPEG-4 is an international audio-video encoding standard (ISO/IEC CD 14496); MPEG-4 Part 2 deals specifically with encoding video. MPEG-4 Part 2 standards specify that media should be encoded between 64 and 8000 kilobits of visual data per second (kbit/s), a range of data rates that accommodates anything from the audio stream of a digital telephone to the video stream of a DVD video (at 40,000 kbit/s, Blu-Ray video is well beyond the scope of MPEG-4 Part 2 and instead conforms to Part 10 of the MPEG-4 standard).  To simplify things, the MPEG-4 Part 2 standard lists several specific implementations, or profiles, of the MPEG-4 Part 2 standard, each of which specifies a maximum bitrate (in concert with other variables, such as frame rate, that are not included here):   * Simple Profile: 64-384 kbit/s, maximum   + Level 0: 64 kbit/s, maximum   + Level 0b: 128 kbit/s, maximum   + Level 1: 64 kbit/s, maximum   + Level 2: 128 kbit/s, maximum   + Level 3: 384 kbit/s, maximum * Advanced Simple Profile: 128-8000 kbit/s, maximum   + Level 0: 128 kbit/s, maximum   + Level 1: 128 kbit/s, maximum   + Level 2: 384 kbit/s, maximum   + Level 3: 768 kbit/s, maximum   + Level 3b: 1500 kbit/s, maximum   + Level 4: 3000 kbit/s, maximum   + Level 5: 8000 kbit/s, maximum   So, digital video encoded at 1200 kbit/s would conform to Level 3b of the Advanced Simple Profile of the MPEG-4 Part 2 standard. |
| [Protocol](http://usgin.org/glossary/protocol) | In the context of computing, a protocol is a special set of rules that enable communication between two computers.  A useful physical analogy is a traffic light: on green lights, cars pass through intersections; on red lights, cars are not permitted to pass through an intersection. The rules represented by traffic lights can be considered traffic protocols; the rules for computer-related tasks such as data transfer are computing protocols.  [HTTP](http://usgin.org/glossary#http) is one example of a computing protocol. |
| [Raster](http://usgin.org/glossary/raster) | Rasters use a data model in which data, usually images or continuous datasets, can be stored and represented visually as values within a grid of cells. Raster grid cells are assigned values that represent specific properties; these grid cell values usually can be decoded as colors. Consequently, a raster dataset is rather like a sheet of graph paper in which each cell contains a color that corresponds with the data represented by the cell. The resolution of a raster dataset is the number of cells on the X and Y axes of the raster grid.  Raster datasets with which most users will be familiar are digital images, including JPEG, TIFF, or GIF images. The individual raster grid cells of these images are referred to as pixels. If the resolution of such images is large enough, individual pixels will be difficult to discern with the naked eye (depending on the scale at which the image is viewed).  The amount of color data that can be stored in a given pixel depends on the format of the raster image. Common raster formats are JPEG, TIFF, GIF, PNG, and BMP.  Raster images are optimal primarily for the storage and display of continuous data sets, which model phenomena without distinct boundaries such as temperature gradients over a given area. Continuous data sets are difficult to display as [vectors](http://usgin.org/content/glossary#vector). A disadvantage of rasters is that raster image files can be very large depending on the resolution, color depth, and compression of the image.  Compare: [vectors](#vector). |
| [Resource](http://usgin.org/glossary/resource) | An [identifiable](http://usgin.org/glossary#identifier) thing that fulfills a requirement. Almost anything can be a resource, as long as it is identifiable and fulfils a requirement. |
| [Schema](http://usgin.org/glossary/schema) | From a practical perspective, schemas structure documents.  For example, a database is one form of document; a database schema describes:   * The specific properties of each field and table in the database * The nature of the data that should be entered into each table or field * The manner in which this data should be formatted.   As a practical example, a database schema might determine whether or not dates should be entered in a given field according to the **MM-DD-YYYY** or **YYYY-MM-DD** formats.  Database schemas thereby dictate where and how data should be entered into a database.  Schema validation is the process of checking data in a database against a schema. A database record containing data that has not been entered in accordance with the appropriate schema, such as data that has not been entered into the appropriate column or formatted in the appropriate way, is invalid. |
| [Server](http://usgin.org/glossary/server) | A server is a computing platform capable of listening for external requests (from clients) and serving responses to those requests; this interaction is referred to as the client-server relationship.  A server can be:   * A computer * A virtual computer (a computer created via software) * A peripheral, such as a printer   The ability to serve requests is usually conferred by installing server software on a computing platform and then configuring the computing platform appropriately (though individual applications are also capable of functioning as servers on their own). Server software designed to create a web server (a server capable of serving [HTTP](http://usgin.org/glossary#http) requests) is called [web server software](http://usgin.org/glossary#web_server_software). |
| [String](http://usgin.org/glossary/string) | In computing, a sequence of alphanumeric characters. |
| [Subrecipient](http://usgin.org/glossary/subrecipient) | Subrecipients are the state geological surveys (or equivalent state agencies) that are subcontracted by the [Arizona Geological Survey (AZGS)](http://azgs.az.gov/) under the [Department of Energy (DOE)](http://energy.gov/) contract No. DE-EE0002850 for the [National Geothermal Data System (NGDS)](http://www.geothermaldata.org) project to perform the subcontract requirement to make "at risk" geothermal data available online to promote geothermal development throughout the Unites States. |
| [Syntax](http://usgin.org/glossary/syntax) | A ruleset that governs the construction of phrases in a human- or machine-readable language. |
| [Token](http://usgin.org/glossary/token) | A token is a discrete, logical, non-elementary component of an information stream (here, non-elementary means that a token is not irreducible and can be reduced to smaller components). For example, a sentance is an information string; words are non-elementary components of a sentence and therefore tokens.  Computing information streams, such as [URIs](http://usgin.org/glossary#uri) can also be broken down into tokens. For example, USGIN URIs conform to the following syntax:  http://host/uri-gin/authority/resource-type/resource-specific/  ...wherein the following are tokens:   * http:// * host/ * uri-gin/ * authority/ * resource-type/ * resource-specific/   For more information about USGIN URIs, see the [USGIN URI tutorial](http://usgin.org/content/usgin-uri-tutorial/). |
| [Uniform Resource Identifier (URI)](http://usgin.org/glossary/uniform-resource-identifier-uri) | Uniquely [identifies](http://usgin.org/glossary#identifier) a [resource](http://usgin.org/glossary#resource); in the context of USGIN, URIs typically identify [database records](http://usgin.org/glossary#database_record), [features](http://usgin.org/glossary#feature), and vocabulary terms. USGIN URIs are also designed to [dereference](http://usgin.org/glossary#dereference) to representations of the resources they identify. See the [USGIN URI Tutorial](http://usgin.org/content/usgin-uri-tutorial) for more information about URIs. |
| [Vector](http://usgin.org/glossary/vector) | Vectors use a data model in which data, usually categorical or discrete data, is stored and represented visually as coordinate points, or vertices. A single vertex is a point; multiple vertices strung together can form lines (referred to as arcs in ESRI products), polygons, or even three-dimensional objects in virtual space; these vector-based objects can be [rasterized](http://usgin.org/glossary#raster) with relative ease.  [GIS](http://usgin.org/glossary#gis) software takes advantage of the characteristics of vector data to generate maps in which the [features](http://usgin.org/glossary#feature) on the map are defined by vertices. Each vertex is georeferenced, often taking advantage of Global Positioning System (GPS) satellites; each feature on the map is then described by [attributes](http://usgin.org/glossary#attribute), which provide the user with information that can be used to locate features and perform geospatial analysis.  Vector datasets are flexible, and their file size is usually small: they grow in size only as more data is added to the dataset. The primary disadvantage of vector datasets is their inability to represent phenomena without distinct boundaries.  Compare: [rasters](http://usgin.org/glossary#raster). |
| [Web Server Software](http://usgin.org/glossary/web-server-software) | Web server software is a software package that allows a computer to listen for, and respond to, incoming [HTTP](http://usgin.org/glossary#http) requests. An appropriately configured computer on which web server software has been installed can act as a web [server](http://usgin.org/glossary#server). |
| [Web Service](http://usgin.org/glossary/web-service) | A web service is a [server](http://usgin.org/glossary#server)-side [protocol](http://usgin.org/glossary#protocol) that responds to specific requests from client software applications. In response to these requests, web services access appropriately configured web-accessible [resources](http://usgin.org/glossary#resource) on the server. As long as a request for web-accessible resources uses appropriate [syntax](http://usgin.org/glossary#syntax), web services can respond. Web service syntax defines required input parameters, operation output, and the results of any operations performed.  Web services act as an intermediary between the client and the server, allowing the two systems to develop independently. This means that regardless of client or server make and model, regardless of changes to content on the server, a web service will be able to respond to requests as long as those requests are made using correct syntax.  **Etymological Note**: “hosting data as a service” is considered synonymous with “hosting data in accordance with a web service protocol.” Likewise, data hosted in accordance with a web service protocol is often referred to as a web service.  The [Open Geospatial Consortium](http://www.opengeospatial.org/) has produced several different flavors of web service that are relevant to [geographic information systems](http://usgin.org/glossary#gis), USGIN, the [National Geothermal Data System (NGDS)](http://www.geothermaldata.org/), and the [AASG Geothermal Data](http://www.stategothermaldata.org) project. These include:   * **Catalog Service – Web (CSW)**: Catalog services are designed to query databases containing [metadata](http://usgin.org/glossary#metadata) about other services, thereby allowing users to discover and access services more easily * **Vocabulary Service**: Vocabulary services are designed to systematically provide web-accessible definitions for specific vocabulary terms * **Web Coverage Service (WCS)**: Web coverage services are designed to publish continuous data sets. Continuous data sets are so named because the data they display does not have discrete boundaries. Consequently, continuous data sets are unsuited to [vector](http://usgin.org/glossary#vector) images and are instead stored as [raster](http://usgin.org/glossary#raster) images.  For example, data about oceanic temperature is a continuous data set, since it is difficult to establish clear boundaries in a body of water. Consequently, data about oceanic temperature would be stored as a raster image and published as a web coverage service. * **Web Feature Service (WFS)**: Web feature services provide georeferenced [features](http://usgin.org/glossary#feature) described by [attributes](http://usgin.org/glossary#attribute).  For example a feature service containing data about river systems might provide features with linear geometry representing river segments; each feature might be described by attributes such as average flow rate, width, and depth for the segment.  Feature services are useful for geospatial analysis. For example, a client application used to calculate the most efficient route between points in a city utilizing real-time traffic information would require a representation of the streets as features described by attributes.  Owing to the large amount of data associated with WFS, these web services tend to require fast Internet connections. * **Web Map Service (WMS)**: Web map services provide a georeferenced map image within a geographic bounding box; they typically provide georeferenced rasters of vector-based features. Web map services are most useful for visual exploration of geographic relationships.  Web map services respond to **getMap** requests by returning an image file (typically \*.tif, \*.jpg, \*.bmp or \*.png) for the requested area. **GetMap** requests specify the geographic area of the bounding box and the map that is desired; a variety of other parameters allow control over image size, map projection, and other details.  For example, a **getMap** request could be used to request a georeferenced satellite photo of Arizona; if the satellite photo exists on the server as a web-accessible image file, the web service will respond to the **getMap** request by providing the client with the desired image.  Web map service image files are often based on a source shapefile or feature class; consequently, web map services support **FeatureInfo** requests at point locations. **FeatureInfo** request returns are not standardized and depend on server configuration. |
| [XML](http://usgin.org/glossary/xml) | XML stands for Extensible [*Markup Language*](http://usgin.org/glossary#markup_language). XML acts as the basis for more specialized markup languages such as GML, GeoSciML, and KML.  Because XML [elements](http://usgin.org/glossary#element) are functionally similar to [database](http://usgin.org/glossary#database) [fields](http://usgin.org/glossary#database_field), USGIN specifies the usage of XML documents as an [interchange format](http://usgin.org/glossary#interchange_format) for [database records](http://usgin.org/glossary#database_record). To use XML documents as an interchange format, USGIN defines XML [schemas](http://usgin.org/glossary#schema) in which each XML document corresponds with a specific database record and each element in a given XML document corresponds with data entered in a database field. In these documents, elements define the database fields.  For example, a **Date** field containing the date 12/7/1941 would appear as follows in an XML document:  <date>12/7/1941</date>  For a much more detailed overview of XML, see the [USGIN XML Tutorial](http://usgin.org/content/xml-tutorial/). |

Funding for the USGIN initiative provided by the [National Science Foundation](http://nsf.gov) under NSF award number EAR-0753154, as part of the [INTEROP initiative](http://www.nsf.gov/pubs/2007/nsf07565/nsf07565.pdf). The USGIN initiative is managed by the [Arizona Geological Survey (AZGS)](http://azgs.az.gov) on behalf of the [Association of American State Geologists (AASG)](http://www.stategeologists.org/). Some USGIN documents and products were developed for use by the [National Geothermal Data System](http://www.geothermaldata.org) (NGDS), which is powered by USGIN; these materials are supported by the [Department of Energy](http://www.energy.gov) under grants EE-0002850 and EE-0001120 awarded to the AASG and [Boise State University](http://www.boisestate.edu/), respectively.

[quiet](http://www.guyzingear.com/gngsvr1.php?topicid=58361)

[quiet](http://www.guyzingear.com/gngsvr1.php?topicid=58361)

[randomness-usgin](http://www.guyzingear.com/gngsvr1.php?topicid=58361)