# Preserving State Government Digital Information
# Minnesota Historical Society

## Options for Improving Access to Legislative Records: A White Paper

---

***Abstract***

*Access to information has been greatly enhanced by the ubiquity of the web. But there are limitations that impede even greater access. Information stored in proprietary formats or dynamically generated content may be unreadable by web search engines, thus making certain web content invisible to searchers. Persons using assistive technologies may have difficulty interpreting certain content elements. Approaches to making records accessible are compared and relative strengths and weaknesses of selected tools and file formats discussed. Ways to facilitate aggregation and analysis which can improve access while leveraging user-created value are also considered.*

*Any comments, corrections, or recommendations may be sent to the project team, care of:*

Nancy Hoffman
Project Analyst
Minnesota Historical Society
nancy.hoffman@mnhs.org / 651.259.3367

---

### Overview of Access Considerations

Legislative records that are only available in paper form clearly limits access to those citizens who can get their hands on a copy of the printed document. Now that the internet "has become the de-facto platform of interoperability and search engines have emerged as primary information portals," [1] information made available on the web offers the promise of making these same documents and data easier to obtain and use.

Electronic records may still pose serious barriers to access and use. They will have many of the same limitations as paper records if they are only held on internal network systems and not made available on the web. Even exposing records on the internet will not automatically make them easy to find and use. Citizens will have difficulty finding electronic records hidden behind database search interfaces. Dynamically generated database content, accessible only through search forms is invisible to the machine-agents, also called crawlers, that find and index pages for the major search engines. As a result, dynamically-generated content will not be included in search engine results[2]. Dynamic content also presents a problem for disabled people using assistive technology to access the internet because this technology cannot interpret it.

[1] http://www.openarchives.org/ore/documents/CompoundObjects-200705.html [accessed 6/22/2009]
[2] http://radar.oreilly.com/2009/03/transforming-the-relationship.html [accessed 6/22/2009]

The ability to find and view information is just the first step in improving access. Aggregation and analysis can promote and provide even greater access by using data to address questions or solve problems that were never imagined when the information was created. Rather than attempting to present one access interface that serves perceived public needs, researchers have suggested that government entities make the underlying data available so that it can be used in a wide variety of ways.[3] Proprietary software formats and other restrictions to access limit the ability to collect and reuse information without resorting to "screen scraping," a technique in which a computer program gets information from the output of a program intended for human rather than machine use and therefore does not provide all of the context and meaning of the original data set. In short, making information fully accessible via the web means ensuring that it must be as easily read and used by machines as it is by people.

## RESOURCES

*The following is a discussion of the strengths and weaknesses of selected technical tools and file formats that can be employed to improve electronic access to legislative information.*

### Data Formats

Open source software is a key component of any effort to improve access to information on the internet. Software used to organize and manipulate information may be written in code that is open—available to be read and used by others—or it may be proprietary and unavailable for use or modification by anyone but the owner of the software. Information in open software formats makes it possible for anyone who is interested to access it directly. Information held in proprietary systems does not. Some pertinent common open electronic data formats are listed below.

### XML (Extensible Mark-up Language)

Extensible Markup Language (XML) is a simple text-based system of flexible user-created tags that can structure, store, and transport information independent of the hardware or software used. It has a suite of associated tools used to format, query, link, and point to XML tagged information.

XML was originally conceived as a way to facilitate large-scale electronic publishing, but it has also become increasingly important in the exchange of many kinds of information on the web and elsewhere.[4] In fact, XML can be used to exchange data between systems that were never designed to do so. For example, "With XML, your data can be available to all kinds of "reading machines" [such as] handheld computers, voice machines, news feeds, etc."[5] XML-aware applications can interpret XML tags, but the meaning will be contingent upon the context of the tags in an

[3] *Government Data and the Invisible Hand*, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1138083 [accessed 6/22/2009]; *Hack, Mash & Peer: Crowdsourcing Government Transparency* http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1023485 [accessed 6/22/2009].
[4] http://www.w3.org/XML/ [accessed 6/22/2009]
[5] http://www.w3schools.com/Xml/xml_usedfor.asp [accessed 6/22/2009]

application. Documents and other kinds of data can be combined and reused because XML syntax includes a system, called namespaces, for keeping the meanings of tags clear.

Widespread adoption of XML has lead to the creation of many tools for converting a variety of formats into XML. XML handles narrative, semi-structured, and hierarchical information particularly well. XML can become cumbersome when used to represent tabular or relational database structures, but some companies producing relational database software systems have added XML storage.[6]

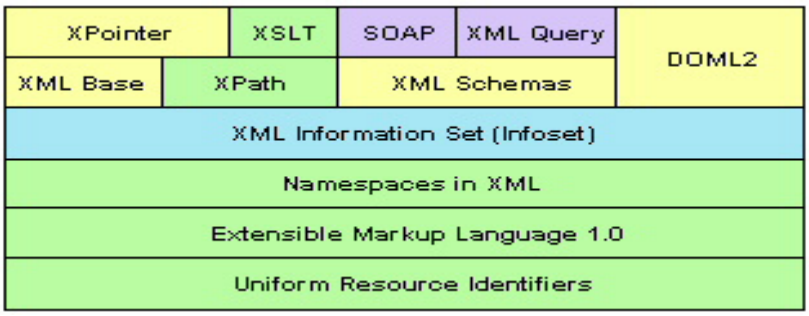
XML has become the basis of numerous specialized data description standards such as XBRL (Extensible Business Reporting Language, a format the Securities and Exchange Commission[7] now requires large firms to use[8]) and KML (Keyhole Markup Language, used by Google Maps), to express geographical information. RSS (Really Simple Syndication) is written in XML and allows information to be published once and viewed by many different programs. Ajax is a set of related tools that incorporates XML and JavaScript in order to allow creation of interactive web applications (see APIs below). One form of expressing semantic data also uses XML notation (see RDF below).

The non-proprietary, human-readable format lessens the chances that information will become completely unreadable in the way some unsupported, proprietary software has, thus increasing the likelihood of long-term accessibility and preservation of documents in XML. XML has been a W3C standard since 1998.[9]

---

Excerpted from: *developmentor XML Tutorial* [10]

"When people refer to XML today they are typically referring to an entire family of layered specifications… [this] figure shows how the different XML specifications are layered in terms of specification dependencies.




XML Specification Dependencies.
Green indicates a Recommendation, yellow a Candidate/Proposed Recommendation, blue a Working Draft, and purple a Note.

---

[6] http://www.oracle.com/technology/tech/xml/xmldb/index.html [accessed 6/22/2009]
[7] http://www.sec.gov/rules/final/2009/33-9002fr.pdf[ accessed 6/22/2009]
[8] http://www.informationweek.com/news/global-cio/compliance/showArticle.jhtml?articleID=207800147 [accessed 6/22/2009]
[9] http://www.w3.org/TR/REC-xml/ [accessed 6/22/2009]
[10] http://www.theserverside.net/tt/articles/showarticle.tss?id=DM_XML [accessed 6/22/2009]

## JSON

Since 2005, JSON has increasingly been used as an alternative to XML. JSON stands for JavaScript Object Notation. Despite its name, JSON is based upon features common to many programming languages and is not restricted to JavaScript. Like XML, JSON has a human-readable text-based notations system. It is built on simple name/value pairs and an ordered list of pairs (also call an array or sequence). JSON was designed to address the somewhat cumbersome aspects of XML such as the DOM, or tree data model that requires large numbers of tags and can slow performance.

JSON can greatly improve the speed and ease of data exchange because of its simple structure, but it does not use many features of XML, such as metadata (attributes), comments, processing instructions, a schema language, and namespaces. These features may be crucial to representing some types of data. Google and Yahoo have adopted JSON data feeds as an alternate interchange format to the XML-based RSS and ATOM formats.[11] Google also has tools available for converting between XML and JSON as well as an XPath-like JSON tool called JSONPath. [12]

## Semantic Applications

The concept behind the semantic web envisions the internet as one large database. As such, all resources on the web can be identified and interrelated in meaningful ways. Fully incorporating information into a semantic web depends upon data that is accessible and that contains meaningful metadata. The vast amount of text-based information that does not, and likely never will, have any useful metadata associated with it makes the possibility of a fully realized "Semantic web" unlikely in the near future. However, the semantic model of linked data or "graphs" has increasingly been employed to add a level of functionality to a variety of data sets that other approaches cannot easily offer. The semantic web is built upon the Resource Description Framework (RDF). The Wikipedia entry notes, "RDF's simple data model and ability to model disparate, abstract concepts has also led to its increasing use in knowledge management applications unrelated to Semantic web activity."[13]

Joshua Tauberer, who aggregates federal legislative information, observes, "The types of questions we can ask can easily grow in complexity and 'interestingness' using RDF. No XPath or XQuery query is going to be nearly so concise for those questions."[14] RDF has been a W3C standard since 1999.[15] Since that time, at least a relational database software manufacturer has added semantic functionality to its product.[16]

[11] http://code.google.com/apis/gdata/json.html [accessed 6/22/2009]
[12] http://code.google.com/p/jsonpath/ [accessed 6/22/2009]
[13] http://en.wikipedia.org/wiki/Resource_Description_Framework [accessed 6/22/2009]
[14] http://www.xml.com/pub/a/2006/02/08/govtrack-us-public-data-semantic-web.html?page=2 [accessed 6/22/2009]
[15] http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/ [accessed 6/22/2009]
[16] http://www.oracle.com/technology/tech/semantic_technologies/index.html [accessed 6/22/2009]

## RDF and triple stores

Resource Description Framework (RDF) statements are the basic tool for expressing semantic information on the web. RDF notations are usually written in the RDF/XML or N3 formats. RDF statements consist of three parts: a subject, a predicate, and an object (or a triple). Each of these pieces of the statement (and even the statement itself) then have unique identifiers or URIs (see URI below) assigned to them. Triple stores are an archive of RDF-encoded data. See www.govtrack.us/data/rdf/ for an example of legislative data in triple store format.

## RDFa

RDFa is a way of placing RDF notation directly into XHTML by using XML element attributes. It will be ignored by presentation-oriented systems but readable by RDF parsers. Adding metadata to existing XHTML pages to make it available for reuse addresses the lack of metadata in many text-based web documents mentioned above. This also provides an immediate benefit to anyone who publishes data on the web but wants to ensure that vital metadata does not become separated from the content it describes. RDFa-aware applications can make use of this data to do things such as manage content and share information more effectively.[17]

RDFa encoding can be generated automatically. One writer has noted, "Whenever you see HTML being generated automatically, you have an opportunity to create RDFa . . . [So] many web pages where we look up information are generated from a backend database. This is fertile ground for easy RDFa generation, which could make RDFa's ease of incorporating proper RDF triples into straightforward HTML one of the great milestones in the building of the semantic web."[18]

While RDFa may not currently offer any specific benefits to internet users with disabilities, the additional machine-readable metadata offers designers of assistive technology such as screen readers an opportunity to render content more clearly. RDFa has also become the most popular alternative to microformats (see below) that have proven to pose some accessibility-related problems.[19]

RDFa became an official W3C recommendation on October 16, 2008. [20]

## Microformats

Like RDFa, microformats[21] provide a way to add human- and machine-readable metadata in XHTML. Unlike RDFa, microformats specify the vocabulary used in each type of microformat and are therefore less flexible[22]. Garrett Dimon explains, "They use current XHTML tags such as address, cite, and blockquote and attributes such as rel, rev, and title to create semantically appropriate blocks of code. Microformats are great because they are both usable and elegant—and

[17] http://www.w3.org/2006/07/SWD/RDFa/scenarios/20070109/ [accessed 6/22/2009]
[18] http://www.xml.com/pub/a/2007/04/04/introducing-rdfa-part-two.html?page=2 [accessed 6/22/2009]
[19] http://www.standards-schmandards.com/2007/rdfa-and-accessibility/[accessed 6/22/2009]
[20] http://www.w3.org/TR/rdfa-syntax/ [accessed 6/22/2009]
[21] http://microformats.org/about/[accessed 6/22/2009]
[22] http://www.w3.org/TR/2008/CR-rdfa-syntax-20080620/ [accessed 6/22/2009]

all you need to do to get started with them is familiarize yourself with the best ways to apply the tags and attributes you already use."[23]

Some microformats are:

- hAtom - for marking up Atom feeds from within standard HTML
- hCalendar - for events
- hCard - for contact information
  - includes adr - for postal addresses,
  - geo - for geographical coordinates (latitude, longitude)
- hReview - for reviews
- hResume - for resumes or CVs
- rel-directory - for distributed directory creation and inclusion
- rel-nofollow, an attempt to discourage 3rd party content spam (e.g., spam in blogs)
- rel-tag - for decentralized tagging (Folksonomy)
- xFolk - for tagged links
- XHTML Friends Network (XFN) - for social relationships
- XOXO - for lists and outlines

Although microformats are easy to use and a simple way to make data available for reuse directly into programs like Microsoft Outlook, the hCalendar microformat in particular includes abbreviations that are not friendly to blind people using screen readers. The BBC[24] and the Massachusetts[25] state government have decided to discontinue use of microformats on their websites for this reason and are looking at RDFa as an alternative.

### Topic Maps

Topic Maps seek to enhance traditional finding aids such as site maps or indices by adding meaningful associations between the listed subjects. Topic Maps have three basic components: 1) a set of topics; 2) a set of associations or links with each topic involved in the association characterized by the role type represented; 3) the component that points to the occurrences of the topic in a resource set. Searches employing Topic Maps, as with other semantic technologies, can be much more accurate than simple full text searching.[26] "A topic map usually contains several overlapping hierarchies which are rich with semantic cross-links."[27]

Topic Maps are a family of ISO standard knowledge representation languages.[28] Similar to RDF, the Topic Maps languages use URIs (see below) as the principal subject identifiers.[29]

[23] http://www.digital-web.com/articles/microformats_primer/ [accessed 6/22/2009]
[24] http://times.usefulinc.com/2008/06/24-uf-rdfa [accessed 6/22/2009]
[25] http://www.snee.com/bobdc.blog/2008/03/accessibility-problems-with-mi.html [accessed 6/22/2009]
[26] http://www.ontopia.net/topicmaps/materials/tao.html#d0e1306 [accessed 6/22/2009]
[27] http://www.xml.com/pub/a/2002/09/11/topicmaps.html [accessed 6/22/2009]
[28] http://www.y12.doe.gov/sgml/sc34/document/0129.pdf [accessed 6/22/2009]
[29] http://www.mondeca.com/index.php/en/knowledge_center [accessed 6/22/2009]

## URIs - URNs and URLs

Uniform Resource Identifiers[30] (URIs) can be one of two types: Uniform Resources Names (URNs) or Uniform Resource Locators (URLs), or in other words, names and addresses. They are important because they allow a unique identifier to be assigned to a resource to distinguish it from every other resource that is represented on the web and a place to get it (or, more precisely, a representation of it).

Use and reuse of internet resources require stable names and addresses. Standards organizations have taken up this challenge and formed services to register name and address identifiers, including systems called resolvers that will maintain access to addresses even when a location changes.

Examples**:**

**PURLs**

"A PURL is a Persistent Uniform Resource Locator. Functionally, a PURL is a URL. However, instead of pointing directly to the location of an internet resource, a PURL points to an intermediate resolution service. The PURL resolution service associates the PURL with the actual URL and returns that URL to the client. The client can then complete the URL transaction in the normal fashion. In web parlance, this is a standard HTTP redirect.

"The On-line Computer Library Center (OCLC) PURL Service has been strongly influenced by the active participation of OCLC's Office of Research in the Internet Engineering Task Force Uniform Resource Identifier working groups. There is nothing incompatible between PURLs and the ongoing URN (Uniform Resource Name) work. PURLs satisfy many of the requirements of URNs using currently deployed technologies and can be transitioned smoothly into a URN architecture once it is deployed."[31]

**Handles.net**

"The Handle.net initiative by the Corporation for National Research Initiatives (CRNI) also seeks to ensure stable references by providing a resolution service for accurately directing users to resources regardless of changes to their location.

"There are many reasons you might want to use the Handle System, but one simple one is that you have information and other resources represented in digital form, sometimes called digital content, that you want users to access via the internet and you plan to keep that content available over long periods of time. If the content's location is likely to have to change during that time, then you need a resolution system like the Handle System. You would assign your digital content unique identifiers, not just identify the objects by their locations. A location—a given URL, for example—is not a persistent identifier if the content moves to another location. . . . The Global Handle Registry is responsible for returning data stored in handle records that are accessed by prefixes and/or by 'service handles' to clients that request it. Clients use that data to locate the service responsible for

[30] http://www.w3.org/Provider/Style/URI [accessed 6/22/2009]
[31] http://purl.oclc.org/ [accessed 6/22/2009]

resolving identifiers beginning with a given prefix. This system is designed so that clients and local resolvers may cache the referral information and not interact with the GHS very often."[32]

**INFO URI**

"The motivation behind developing the INFO URI scheme was to allow legacy identification systems to become part of the World Wide Web global information architecture so that the information assets they identify can be referenced by web-based description technologies such as XLink, RDF or Topic Maps. Note that we are concerned with 'information assets,' not 'digital assets' per se—the information assets may be variously digital, physical or conceptual.

"The INFO URI scheme does not compete with independent URI registrations but rather cooperates with independent URI registrations by providing a lightweight early URI registration mechanism to support referencing of public information assets ahead of any possible subsequent URI scheme or URN namespace application. Note that in the majority of cases no subsequent URI scheme or URN namespace application will be made by a Namespace Authority as the INFO resource identifier alone will be sufficient in providing an identification service. Only if additional services are required would a Namespace Authority seek to register a URI scheme independently.

"The INFO URI scheme was developed by members of the library and publishing communities working together under the auspices of ANSI/NISO."[33]

**The DOI System**

The Digital Object Indentifiers (DOI) System is associated with Handles.net (see above). "The DOI System provides a framework for persistent identification, managing intellectual content, managing metadata, linking customers with content suppliers, facilitating electronic commerce, and enabling automated management of media. DOI names can be used for any form of management of any data, whether commercial or non-commercial.

"The system is managed by the International DOI Foundation, an open membership consortium including both commercial and non-commercial partners, and has recently been accepted for standardisation within ISO … Unique identifiers are essential for the management of information in any digital environment. Identifiers assigned in one context may be encountered, and may be re-used, in another place (or time) without consulting the assigner, who cannot guarantee that his assumptions will be known to someone else. To enable such interoperability requires the design of identifiers to enable their use in services outside the direct control of the issuing assigner. The necessity of allowing interoperability adds the requirement of persistence to an identifier: it implies interoperability with the future. Further, since the services outside the direct control of the issuing assigner are by

[32] http://www.handle.net/ [accessed 6/22/2009]
[33] http://info-uri.info/registry/docs/misc/faq.html [accessed 6/22/2009]

definition arbitrary, interoperability implies the requirement of extensibility. Hence the DOI System is designed as a generic framework applicable to any digital object, providing a structured, extensible means of identification, description and resolution. The entity assigned a DOI® name can be a representation of any logical entity.

"The DOI System data model consists of a data dictionary and a framework for applying it. Together these provide tools for defining what a DOI name specifies (through use of a data dictionary), and how DOI names relate to each other, (through a grouping mechanism, Application Profiles, which associate DOI names with defined common properties). This provides semantic interoperability, enabling information that originates in one context to be used in another in ways that are as highly automated as possible.

"The DOI System uses an interoperable data dictionary built from an underlying ontology."[34]

## REST

Representational State Transfer (REST) is the design principle upon which the internet is built. The uniform resource identifiers discussed above are important because they define one of the essential elements of REST – the resources. In order to exchange information between computers via the internet's Hypertext Transfer Protocol (HTTP) only two things are needed: the identifier of a resource, and the type of action to be taken – and there are just four, 'Get', 'Post', 'Put', and 'Delete.' Usually, web browsers 'Get' a resource called a web page indentified by a URL.

Tim Bray, who launched one of the first public web search engines[35] explains, "…the input to GET is a URI, that simple, wonderful little string that can be used to address anything on the web, which is more or less everything in the world these days. That means you can email that URI around, or feed it to a search engine's spider, or bookmark it, or do a bunch of other things that probably haven't been invented yet. When you POST, you contact the server and send it a package of information, not just a URI, and get back a package of information. You should use POST if you want to do anything more than just retrieve information, for example buy or blow up something. So most times when you fill out a form on a web site and press Submit, it's using a POST."[36]

Many web interfaces do not conform to REST principles (are not RESTful). They are built on protocols that transmit messages, primarily Simple Object Access Protocol (SOAP)[37]. Many programmers accustomed to using SOAP methods find it easier to use than REST.[38] However, most SOAP methods are usually one-off solutions. They can easily break when part of the system changes. Software architect and author Benjamin Carlyle[39] suggests that,

"REST architecture will be the successful style in large information systems consisting of separately-upgradable parts. I take the existing Web as evidence of this. It is already the way

[34] http://www.doi.org/overview/sys_overview_021601.html [accessed 6/22/2009]
[35] http://www.infoq.com/interviews/tim-bray-future-of-web [accessed 6/22/2009]
[36] http://www.tbray.org/ongoing/When/200x/2003/05/12/SoapAgain [accessed 6/22/2009]
[37] http://en.wikipedia.org/wiki/SOAP [accessed 6/22/2009]
[38] http://netzooid.com/blog/2007/08/19/ease-of-development-rest-vs-rpc/ [accessed 6/22/2009]
[39] http://www.linkedin.com/in/benjamincarlyle [accessed 6/22/2009]

> large-scale software architecture works. REST accommodates the fact that different parts of this architecture are controlled by different people and agencies. It deals with very old software and very new software exist[ing] in the same architecture. It codifies a combination of human and technical factors that make large-scale machine cooperation possible."[40]

If the ultimate value of making government information available on the web is the ability of users to find and use that information, RESTful systems are the way to maximize that potential value. Although setting up a RESTful architecture may take more time initially for developers who are not familiar with it, REST allows information to be easily reused and shared, can improve system performance, and reduce maintenance costs over the long run.

## Sitemaps

Perhaps the most basic tool for fully exposing information on the web has been put forward by Google:

> "The Sitemap Protocol allows you to inform search engines about URLs on your websites that are available for crawling. In its simplest form, a Sitemap that uses the Sitemap Protocol is an XML file that lists URLs for a site. The protocol was written to be highly scalable so it can accommodate sites of any size. It also enables webmasters to include additional information about each URL (when it was last updated; how often it changes; how important it is in relation to other URLs in the site) so that search engines can more intelligently crawl the site.
>
> "Sitemaps are particularly beneficial when users can't reach all areas of a website through a browseable interface. (Generally, this is when users are unable to reach certain pages or regions of a site by following links.) For example, any site where certain pages are only accessible via a search form would benefit from creating a Sitemap and submitting it to search engines."[41]

Google has promoted Sitemaps; other major search engines such as Yahoo, Ask, and MSN also support them. Tools to automate the production of sitemaps take into account the fact that large databases may have thousands of record URLs that would be prohibitive to create individually[42]. Producing large files can still be time-consuming, but it realizes the intent of posting the records in the first place. Vanessa Fox points out some of the problems database records can create for searchers and search engines,

> "In addition to being buried behind JavaScript and containing little language people would actually search for, it's hidden in a popup with a URL like this:
>
> <http://factfinder.census.gov/servlet/IdentifyResultServlet?_mapX=281&_mapY=216&_latitude=&_longitude=&_pageX=442&_pageY=554&_dBy=100&_jsessionId=0001cv7n8rWxjslrmI9aRw5nr:134a7lbrs">http://factfinder.census.gov/servlet/IdentifyResultServlet?_mapX=281&_mapY=216&_latitude=&_longitude=&_pageX=442&_pageY=554&_dBy=100&_jsessionId=0001cv7n8rWxjslrmI9aRw5nr-V:134a7lbrs>.

[40] http://soundadvice.id.au/blog/2007/08/27/ [accessed 6/22/2009]
[41] https://www.google.com/webmasters/tools/docs/en/protocol.html
[42] http://googlewebmastercentral.blogspot.com/2008/12/sitemap-submission-made-simple.html [accessed 6/22/2009]

> … If I share that URL on a social media site, email, or in this blog post, anyone who tries to visit it just gets a "session as expired" message. It goes without saying that this kind of URL can't be indexed by search engines no matter how sophisticated they become."[43]

The bill to reauthorize the E-government Act of 2002 (S. 2321) specifies that government information available on the web must be exposed to search engines in a manner such as this.[44] While this bill did not pass out of committee, the intent is bound to resurface in other guidelines and regulations.

**APIs (Application Program Interfaces)**

The web search forms that limit access to information by search engine crawlers (see Sitemaps above) represent one implementation of an API, or Application Program Interface. APIs are a set of declarations of functions or procedures to support requests made on software programs. As such they can allow information sharing across different applications and platforms.

Calls for information from a program are usually written in scripting languages. Common scripts used to create search forms are CGI (Common Gateway Interface), PHP (Hypertext Preprocessor scripting language), and ASP (Active Server Pages). Many scripts are language- and program-neutral, working with a variety of software applications.

Dan Cohen, a digital humanities scholar at George Mason University, explains that a simple API can be created "by repackaging [an] existing search tool using simple web services protocols such as REST where a URL sent to a server returns an XML document rather than an HTML results page. Users of the API can then parse the XML document to extract the information they need or would like to combine with (or pass on to) other services."[45]

Making APIs allows searching, sorting, combining, and analyzing public records in multiple ways rather than restricting interaction with the information to a single predetermined interface. It does not however usually offer access to the raw data or the entire dataset.

> Programmer Jeni Tennison writes, "Working with public-sector information on the web, one of the things that I take an interest in is making government data freely available for anyone to re-present, mash-up, analyse and generally do whatever they want to do... people who control data don't realise that the smallest changes can be beneficial: they don't need to do everything right now, just something. There are three fundamental things that you need to do: identify the data that one controls, represent that data in a way that people can use, and expose the data to the wider world, but you can choose the degree to which you do each of these things."[46]

[43] http://radar.oreilly.com/2009/03/transforming-the-relationship.html [accessed 6/22/2009]
[44] "Experts: Government Data Unseen Online," *Information Management Journal*, 42:2, p. 20
[45] http://www.dlib.org/dlib/march06/cohen/03cohen.html [accessed 6/22/2009]
[46] http://www.jenitennison.com/blog/node/100 [accessed 6/22/2009]

## Further Reading

*Further information on the technologies listed above*

### W3C eGovernment Interest Group,

Improving Access to Government through Better Use of the Web
W3C Interest Group Note
http://www.w3.org/TR/egov-improving/

Subsection: How Can Open Government Data Be Achieved?
http://www.w3.org/TR/egov-improving/#OGD.how

### XML

XML.gov
http://xml.gov/

> Home page of an organization that promotes use of XML to address problems of data interoperability in government

Legislative Documents in XML at the United States House of Representatives
http://xml.house.gov/

About the Organization for the Advancement of Structured Information Standards
http://www.oasis-open.org/who/

> "OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society. The consortium produces more web services standards than any other organization along with standards for security, e-business, and standardization efforts in the public sector and for application-specific markets. The Consortium hosts two of the most widely respected information portals on XML and web services standards, Cover Pages and XML.org."

### JSON

Introducing JSON
http://json.org/

> Home page for programming information

JSON: The Fat-Free Alternative to XML
http://www.json.org/xml.html

> JSON.org comparison of JSON and XML

### RDFa

W3C Working Group Note, RDFa Primer
http://www.w3.org/TR/xhtml-rdfa-primer/

> RDFa definition and specifications from the World Wide Web Consortium

RDFa Wiki

http://rdfa.info/wiki/Introduction
Guidance and resources on RDFa use, including a video covering the basics of RDFa mark-up by Manu Sporny

### Microformats

*Discussions of the relative pros and cons of Microformats and RDFa*

The BBC, accessibility, the hCalendar microformat and RDFa. Mia Ridge. June 23, 2008
http://openobjects.blogspot.com/2008/06/bbc-accessibility-hcalendar-microformat.html

BBC, Microformats and RDFa. Griffin Caprio. June 23, 2008
http://www.oreillynet.com/xml/blog/2008/06/bbc_microformats_and_rdfa.html

hAccessibility. James Craig. April 27, 2007
http://www.webstandards.org/2007/04/27/haccessibility/

W3C Questions & Answers blog
http://www.w3.org/QA/2008/06/war-of-the-worlds.html

Microformats - Part 0: Introduction
http://blog.mozilla.com/faaborg/2006/12/11/microformats-part-0-introduction/

What Are Microformats. Micah Dubinko. March 23, 2005
http://www.xml.com/pub/a/2005/03/23/deviant.html

### Topic Maps

XML Topic Maps (XTM) 1.0, TopicMaps.Org Specification
http://www.topicmaps.org/xtm/1.0/
XTM is an XML version of Topic Maps

OASIS Cover Pages Technology Report, (XML) Topic Maps
http://xml.coverpages.org/topicMaps.html
Overview of XML Topic Maps

Mondeca - Kluwer Belgium
http://www.mondeca.com/index.php/en/success_stories/kluwer_belgium_groupe_wolters_kluwer
Project overview of Topic Maps implementation for a legal publishing company

### URIs

W3C Architecture domain, Naming and Addressing: URIs, URLs,…
http://www.w3.org/Addressing/

Uniform Resource Identifier (URI): Generic Syntax – RFC January 2005
http://gbiv.com/protocols/uri/rfc/rfc3986.html

What's a URI and why does it matter? Henry S. Thompson. 4 December 2008
http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/

## REST

Representational State Transfer
http://en.wikipedia.org/wiki/Representational_State_Transfer
Wikipedia has an excellent overview of REST

How I Explained REST to My Wife. Ryan Tomayko. December 12, 2004
http://tomayko.com/writings/rest-to-my-wife
Non-technical explanation of the how and why of REST

Addressing Doubts about REST. Stefan Tilkov. March 13, 2008
http://www.infoq.com/articles/tilkov-rest-doubts
Comparison of REST to SOAP

CFMX - SOAP vs REST benchmarks. Mark Lynch. March 13, 2008
http://www.lynchconsulting.com.au/blog/index.cfm/2008/3/13/CFMX--SOAP-vs-REST-benchmarks
Test comparison of SOAP and REST implementations on the same data set

Google's Gaffe. Paul Prescod. April 24, 2002
http://webservices.xml.com/pub/a/ws/2002/04/24/google.html
A 2002 comparison of the SOAP vs. REST versions of the Google API

XMLRPC vs REST vs SOAP vs CIM vs RMI vs Message Bus vs … Lots of RPC Options. Michael Dehaan. July 17, 2008
http://www.michaeldehaan.net/?p=665
REST vs. XML-RPC - a list of pluses and minuses for these techniques and others

RESTful Service Design, Cesare Pautasso and Erik Wilde
http://dret.net/netdret/docs/soa-rest-icwe2009/design.pdf
Slide tutorial describing how “Web services" can also use other technologies, such as RESTful implementations on top of HTTP Presentation.

## Sitemaps

Sitemaps.org home
http://www.sitemaps.org/index.php

Google Sitemaps Information Center
http://www.smart-it-consulting.com/google-sitemaps.htm

Webmasters/Site owners Help, Sitemaps page
http://www.google.com/support/webmasters/bin/answer.py?answer=40318

### APIs

Programmable Web, Government APIs and Mashups Dashboard
http://www.programmableweb.com/government

Sunlight Labs, Projects
http://www.sunlightlabs.com/projects/
Examples of some APIs posted on federal government information

Larimer County, Public Records Databases API
http://www.larimer.org/databases/api.htm
Examples of a Colorado County that has released an API for government records

Google Data APIs Overview
http://code.google.com/apis/gdata/overview.html
Home page for a set of Google Data APIs available in Atom or RSS formats