FERMILAB-CONF-11-871-CD



# A method of searching LDAP directories using XQuery

**Ted Hesselroth**

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510-5011

E-mail: tdh@fnal.gov

**Abstract**. A method by which an LDAP directory can be searched using XQuery is described. The strategy behind the tool consists of four steps. First the XQuery script is examined and relevant XPath expressions are extracted, determined to be sufficient to define all information needed to perform the query. Then the XPath expressions are converted into their equivalent LDAP search filters by use of the published LDAP schema of the service, and search requests are made to the LDAP host. The search results are then merged and converted to an XML document that conforms to the hierarchy of the LDAP schema. Finally, the XQuery script is executed on the working XML document by conventional means. Examples are given of application of the tool in the Open Science Grid, which for discovery purposes operates an LDAP server that contains Glue schema-based information on site configuration and authorization policies. The XQuery scripts compactly replace hundreds of lines of custom python code that relied on the unix ldapsearch utility. Installation of the tool is available through the Virtual Data Toolkit.

## 1. Information Services and the Grid

Grid computing is the practice of utilizing computing resources that are deployed across administrative domains[1]. In simpler terms, it is performing computations at multiple sites which are operated by different institutions. One premise of grid computing is that sites are configured at the discretion of their operators; therefore, sufficient information to use the resource is not available a priori. Even the presence of resources may change over time.

Grid computation requires interaction within this dynamic and heterogeneous environment. Having accurate and timely information on the deployed infrastructure is an essential component of effective grid utilization, not only for the discovery of services but for uncovering their configuration, without which it may be impossible to actually use them.

Due to the dynamic nature of independently-operated sites, a centralized information service was chosen by the Worldwide LHC Computing Grid (WLCG) [2] to support grid computing for the Large Hadron Collider. The service data store is populated by WLCG sites, which transmit information on installed services and their configuration to the central server. The information thus becomes available to clients which query the server.

### 1.1. The Lightweight Directory Access Protocol

There are a variety of ways of serving information on the internet, besides the ubiquitous HTML format delivered using the http protocol. The method chosen by WLCG is to use LDAP, the Lightweight Directory Access Protocol[3]. Besides the actual protocol for exchanging directory information, LDAP provides the ability to define an information model, or schema, customized to the information being presented. This allows the delivery of information about anything for which a schema can be devised. Another part of LDAP is LDIF, the LDAP Interchange Format. The format

Published under licence by IOP Publishing Ltd


Operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy.

specifies a consistent way of presenting of LDAP entries and their attributes as text headers and name/value pairs, allowing for serialization and the exchange of information between LDAP clients and servers.

The actual software used as the information server is the Berkeley Database Information Index[4]. BDII is an LDAP server with the additional capability of aggregating information from multiple sources. In practice, sites deliver information to the BDII server in the form of LDIFs, which BDII then merges and uses to update the database. The LDIF information from the sites is gathered by Generic Information Providers[5], which either read information from manually-maintained files or obtain it from resource-specific probes installed alongside their respective service. The LDIF-encoded information is delivered to BDII via the CEMon[6] utility. BDII servers are run by both the WLCG in Europe and by the Open Science Grid[7] in the United States. The WLCG BDII service is periodically updated by all of the information in its OSG counterpart.

### 1.2. The Glue Schema

Though the software described here can be used with any schema and any LDAP server, the specific tool that is based on the method described here uses the Glue Schema[8], from the Grid Laboratory Uniform Environment. The version supported by the software described in this paper is 1.3. The Glue Schema serves as an abstract model of a site environment. It is informed by the features of software commonly found on grid sites, with sufficient coverage of services and their parameters such that the resources that a site chooses to make available may be advertised.

### 1.3. XQuery vs LDAP filters

One drawback of the use of LDAP in the WLCG is the very complexity of the Glue Schema. While in principle the LDAP service itself can support quite complex schemas, in practice the schemas have been relatively simple. Consequently, clients have not needed the capacity to perform complex searches. LDAP is searched using “filters”, defined in RFC2254[9], which do provide for limited selection of the search object and attribute names along with basic comparison and Boolean operators. However, no exploitation of any structured data model represented in the schema is supported by LDAP filters. Furthermore, in common search tools such as ldapsearch, results are returned as LDIF, requiring custom parsing tools for extraction of the values of interest. In practice, complex LDAP searches must be composed of multiple simple filter searches, with intermediate parsing of results and reformulation of filter parameters. With reasonable expertise, this can be achieved with a scripting language such as shell script or Python, but results in search tools that are quite specialized in their purpose and would require learning of a particular data handling motif to modify or extend.

Fortunately, the Glue Schema, though expressible as LDAP schema, is not dependent on LDAP. Since it is a hierarchical arrangement of elements and attributes, the Glue Schema may also be expressed as an XML schema. Any informational item that exists in the LDAP directory thus has a corresponding XPath expression associated with it. This suggests the possibility of performing searches based not on LDAP filters, but on XPath[10] expressions, and even further, composing more complex searches using XQuery[11]. One advantage of such an approach would be that it would not be necessary to learn the particulars of some scripting framework based on LDAP search filters (if one existed) in order to write complex searches. Furthermore, these languages are designed precisely for the extraction and manipulation of hierarchical data, and therefore the expressions in case of deep schemas such as the Glue Schema, are more natural to work with.

### 1.4. XML and Directory Services

XML, the Extensible Markup Language[12], became a W3C standard in 1998. Various derivative technologies have been developed based on the XML standard. XPath[10] is a W3C specification from 1999 and 2007 for describing navigation within an XML document, canonically by the construction of the path to an item of information within the document tree, made by the composition of the names of its ancestor elements.

Other technologies that are, in turn, based on XPath are XSLT and XQuery. XSLT, or Extensible Stylesheet Language Transformations[13,14], is a language for transforming XML documents. The transformations are themselves written as XML documents according to the schema as part of its W3C 1999 standard. XQuery, the Extensible Query Language[11], which became a W3C standard in 2007, is similar to XSLT in capability, but with a style more similar to procedural programming. As its name suggests, its constructs are suitable for the application of performing queries on a document. Both languages rely on XPath, but the XQuery language more explicitly so.

By defining a schema in XML, a markup language for a specific purpose may be created. A markup language which enables the use of XQuery to search LDAP directories is the Directory Services Markup Language[15] (DSML).

The Directory Services Markup Language is an XML schema that allows LDAP data to be expressed in XML format. This provides an alternative to the LDIF representation described above. DSML became an OASIS standard in 2001. The standard also includes schema for performing other operations on LDAP servers, such as updates and deletes. DSML Tools[16] provided and continues to maintain Java-based tools for performing LDAP searches and outputting the results as XML.

## 2. An XQuery Strategy for searching LDAP

Based on the ability of expressing Glue Schema as XML schema, and the availability of a tool to acquire LDAP search results as XML, the following strategy for searching an LDAP directory using XQuery presents itself:

1. Write the XQuery script, based on the directory schema.
2. Examine the XQuery script and extract the XPaths from it.
3. Convert the XPath expressions to LDAP filters.
4. Perform searches on the ldap host using the ldap filters, and merge results as XML file.
5. Run the XQuery on the resulting XML file.

The following diagram illustrates the step of the process, and the associated software components.

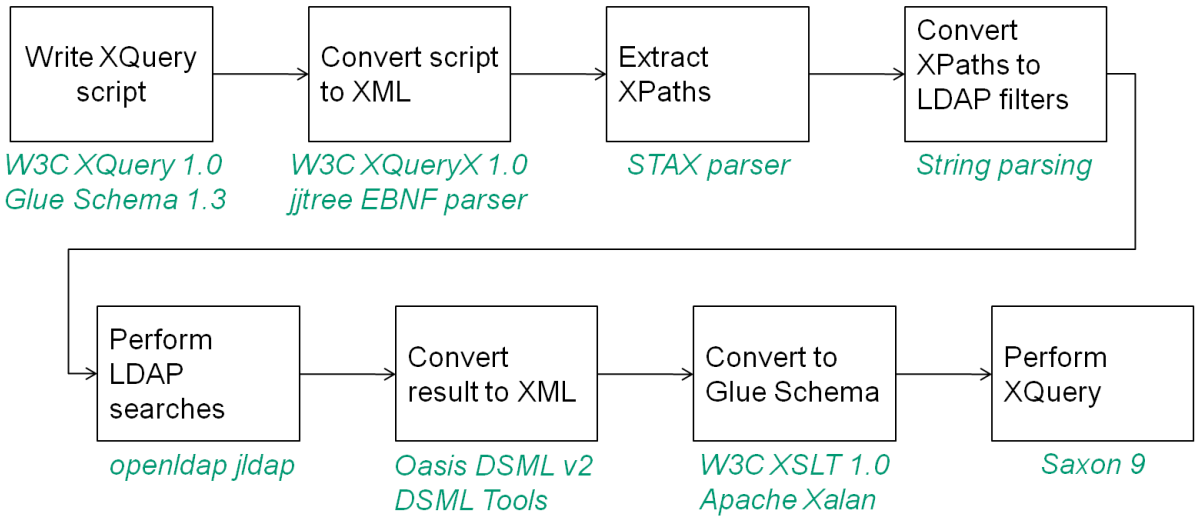


**Figure 1.** Diagram of the processing steps.

Details related to these steps are as follows.

2.1. The XQuery scripts.

Any XQuery script to be evaluated must be written with the knowledge of the schema of the LDAP directory. It is most convenient if the LDAP server can report its own schema, in which case a query using DSML Tools[16] may obtain it in DSML format. Besides informing the construction of the XQuery scripts, the resulting schema file is placed in the Java classpath at runtime. This assists the software in converting the XPaths contained in the XQuery to their LDAP filter equivalents, see section 2.3, below.

One of the goals of the software is to make it easier to write queries for the directory service. In addition, several XQuery scripts thought to be of particular value to grid users have been written and bundled together with the software to provide a tool. The so-named OSG Discovery Tool[17] provides search capabilities runnable through wrapper scripts by which the user can provide search parameters. In the way, customized results can be obtained without the need for any programming. The tool is an Open Science Grid[8]software product and is provided through the Virtual Data Toolkit[18] package. The provided XQueries are designed to search LDAP directories of the Open Science Grid and the Worldwide Large Hadron Collider Computing Grid [2]. Therefore, the Glue Schema 1.3 DSML file is bundled with the Discovery Tool, though the tool is capable of searching any LDAP directory for which the schema is available.

2.2. Parsing the XQuery.

Since the XQuery is itself an XML document, XPaths are extracted from it by means of a parsing operation. The XQuery is preprocessed into XQueryX[19] form (a W3C standard) in order to facilitate recognition of the location axes and step expressions. Additional elements such as comparison and Boolean operators are also detected. A STAX[20] processor is used to create lists of such elements, which are then used to construct the XPath expressions.

2.3. The LDAP filters.

LDAP filters are constructed from the XPath expressions by parsing them as strings. Since both XPath expressions and LDAP filters can include comparison and Boolean operations, the XPath expressions are constructed so as to include them if present, and the operations carried forward to the LDAP filters. This allows a more precise search on the LDAP directory, reducing the volume of XML data to be later queried. In addition, the schema of the LDAP directory in DSML format is used at this point, to disambiguate the construction of LDAP filters from the XPaths. LDAP allows the type of an element to be either “auxiliary” or “structural”; the latter is what must be used in the LDAP filters. If an XPath expression contains an element of type “auxiliary”, its nearest ancestor of type “structural” is determined by the software through an examination of the schema.

2.4. Performing the LDAP search.

The openldap[21] jldap software is used to perform the search, and the result transformed to XML using DSML Tools. The result, while in XML, is expressed in generic DSML schema rather than the schema used by the LDAP server. Therefore, this step includes an additional process of converting the result to the more specific schema by means of an XSL transform. In the case of the XQueries bundled with the Discovery Tool, a transform to convert to the Glue Schema is used.

2.5. Running the XQuery.

The Saxon[22]software is used to perform the XQuery on the resulting XML. In order to avoid the invocation of steps 2.2 to 2.4 every time a command is run, a cache file is employed. Upon the first invocation all bundled XQueries are parsed and corresponding LDAP filters obtained. A search as per step 2.4 and the results saved in an XML file. The one XML file is then sufficient for evaluation of all the provided XQueries. The file is kept current; when a user performs a search, if the file has expired it is refreshed.

The above steps are all performed on the client side. The software also contains a server which can perform the above steps on behalf of a client, though at this time it can only process XPath expressions, using the ReST model. It would be a minor alteration to allow posting of XQueries or implementation of URL paths representing them by this proxy service.

## 3. The OSG Discovery Tool

In the OSG Discovery Tool package, the processing steps are executed by shell scripts, started off by a wrapper shell script named for the particular XQuery to be executed. For example an XQuery, which finds the Storage Elements supporting a particular Virtual Organization and constructs a site URL for each one, has a wrapper named get_surl. An example invocation and output are shown below.

```
[discovery]$ get_surl --vo jdem

srm://bsrm-1.t2.ucsd.edu:8443/srm/v2/server?SFN=/hadoop/jdem/TESTFILE
srm://fndca1.fnal.gov:8443/srm/managerv2?SFN=/pnfs/fnal.gov/usr/fermigrid/volatile/jdem/TESTFILE
srm://osg.crc.nd.edu:8443/srm/v2/server?SFN=/dscratch/osg/bestman/TESTFILE
srm://osg-se.sprace.org.br:8443/srm/managerv2?SFN=/pnfs/sprace.org.br/data/TESTFILE
```

**Figure 2.** A get_surl example.

Formatting is also supported. In the get_storage_area command, an XQuery is called which returns several pieces of information on advertised Storage Areas. It is convenient to display this output in columns, as shown in the following example. An HTML table format is also available.

```
[discovery]$ get_storage_area

STORAGE ELEMENT ID       STORAGE AREA NAME                TOTAL     FREE      VOs
atlas.bu.edu             Default Storage Area             122984    47379     atlas,cdf,mis,ops,osg
atlfs02.phy.duke.edu     Default Storage Area             0         0         atlas
bonner-grid.rice.edu     Default Storage Area             21474     6596      atlas,cms,star
bsrm-1.t2.ucsd.edu       Default Storage Area             1219733   103967    accelerator,astro,atlas,…
cit-se.ultralight.org    Default Storage Area             577173    137168    DOSAR,LIGO,alice,atlas,cdf,…
cms-0.mps.ohio-state.edu Default Storage Area             41015     9669      cms,mis,ops
cmssrm.fnal.gov          LFSOnlyPools:replica:online      531711    486045    cms,mis
cmssrm.fnal.gov          ResilientPools:replica:online    428079    86887     cms,mis
cmssrm.fnal.gov          stagePools:custodial:nearline    161695    18349     cms,mis
…
```

**Figure 3.** A get_storage_area example, showing columnar formatting.

A table of bundled queries supported at the present time is shown below.

| OSG Discovery Tool Command | Purpose |
|---|---|
| get_gridftp_storage_element_id | Returns a list of IDs for gridftp Storage Elements on which the specified VO is authorized. |
| get_gridftp_url | Returns a list of example site URLS for Storage Elements on which the specified VO is authorized. |
| get_mount_path | Returns a Compute Element mount point for a Storage Element which authorizes the specified VO, followed by the associated job manager. |
| get_os_versions | Finds operating system versions for clusters and sites. |
| get_runtime_versions | Finds runtime software versions for clusters and sites. |
| get_site_name | Returns a list of sites that authorize Storage Element access for the specified VO. |
| get_site_name_for_seid | Returns the site name for a storage element with the given id. |
| get_srm_copy_cmd | Returns a list of example srm commands for Storage Elements |

| | |
|---|---|
| get_srmcp_cmd | on which the specified VO is authorized. |
| get_srm_reserve_space_cmd<br>get_srm_sp_reserve_cmd | Returns a list of example srm commands for Storage Elements on which the specified VO is authorized. |
| get_srm_storage_element_id | Returns a list of IDs for SRM Storage Elements. |
| get_storage_area | Displays space and authorization info for storage areas. |
| get_storage_element_id | Returns a list of IDs for gridftp and SRM Storage Elements on which the specified VO is authorized. |
| get_surl | Returns a list of example site URLS for Storage Elements on which the specified VO is authorized. |

**Table 1.** Queries currently bundled with the OSG Discovery Tool.

## 4. Acknowledgments

This work was supported by the Office of Science, U.S. Department of Energy, SciDAC program under Contract DE-FC02-06ER41436 and the National Science Foundation Cooperative Agreement, PHY-0621704. Fermilab is operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy. Fermilab is partially funded by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Dept. of Energy, under Contract DE-AC02-06CH11357.

**References**

[1] Foster, Ian; Carl Kesselman 1999 The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers. ISBN 1-55860-475-8.
[2] The Worldwide LHC Computing Grid (WLCG), http://lcg.web.cern.ch/LCG.
[3] Howes, Timothy A. 1997 The Lightweight Directory Access Protocol : X.500 Lite. CITI Technical Report 95–8. http://www.openldap.org/pub/umich/ldap.pdf.
[4] BDII web site. https://twiki.cern.ch/twiki/bin/view/EGEE/BDII.
[5] Czajkowski, K., Fitzgerald, S., Foster, I. And Kesselman, C. 2001 "Grid information services for distributed resource sharing," 10th IEEE International Symposium on High Performance Distributed Computing. Proceedings. pp.181-194.
[6] CEMON home page. http://grid.pd.infn.it/cemon/field.php.
[7] Open Science Grid web site. http://www.opensciencegrid.org.
[8] Glue Schema web site. http://glueschema.forge.cnaf.infn.it.
[9] Howes, T. 1997 The String Representation of LDAP Search Filters. RFC 2254.
[10] W3C XPath web site. http://www.w3.org/TR/xpath.
[11] W3C XQuery web site. http://www.w3.org/XML/Query.
[12] W3C XML web site. http://www.w3.org/XML.
[13] P. Kilpel and D. Wood 2001 SGML and XML document grammars and exceptions. Inf. Comput. 169, 2.
[14] W3C XSLT web site. http://www.w3.org/Style/XSL.
[15] OASIS DSMl web site. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dsml.
[16] DSML Tools web site. http://www.dsmltools.org.
[17] OSG Discovery Tool web site. https://twiki.grid.iu.edu/bin/view/ReleaseDocumentation/OSGStorageDiscoveryTool.
[18] Virtual Data Toolkit web site. http://vdt.cs.wisc.edu.
[19] W3C XQueryX web site. http://www.w3.org/TR/xqueryx.
[20] STAX web site. http://stax.codehaus.org.
[21] Openldap web site. http://www.openldap.org.
[22] Saxon web site. http://saxon.sourceforge.net.