Aleph SRU Gate

Version 20 and later

CONFIDENTIAL INFORMATION

DISCLAIMER

TRADEMARKS

Document released: August 2010

Web address: http://www.exlibrisgroup.com

Table of Contents

Overview

SRU (Search/Retrieve via URL) is a standard search protocol for Internet search queries, utilizing CQL (Common Query Language) and standard query syntax for representing queries (http://www.loc.gov/standards/sru/).

An SRU request is a HTTP URL. It consists of a base URL and a search part, separated by a question mark. The search part consists of parameters separated by an ampersand, each with the structure `key=value`. For example:

http://localhost:5661/usm01?version=1.1&operation=searchRetrieve&query=dinosaur

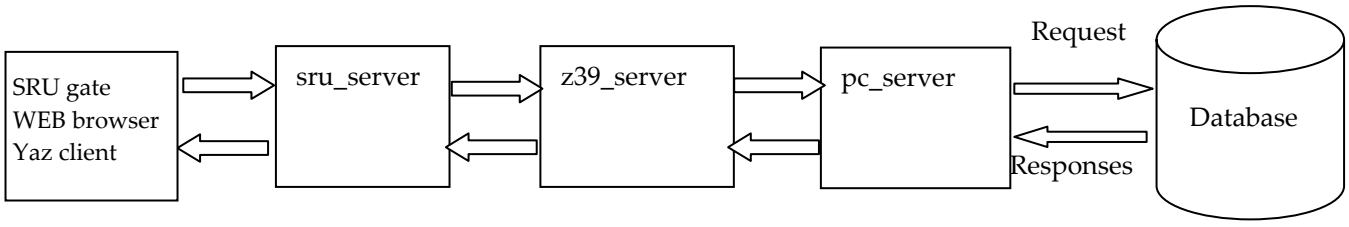
The base URL is http://localhost:5661/usm01 and the search part is version=1.1& operation=searchRetrieve&query=dinosaur.

The response to an SRU request is an XML document, for example:

```
<record>
   <recordSchema>
info:srw/schema/1/dc-v1.1
   </recordSchema>
   <recordPacking>xml</recordPacking>
   <recordData>
         <srw_dc:dc>
               <dc:title>This is a Sample Record</dc:title>
</srw_dc:dc>
   </recordData>
   <recordPosition>1</recordPosition>
   <extraRecordData>
       <rel:rank>0.965</rel:rank>
   </extraRecordData>
</record>
```

Architecture

In Aleph, a SRU gate enables searching in remote SRU databases.



Aleph's SRU gate supports only the SearchRetrieve SRU service.

Supported Actions

SearchRetrieve Request

Request Parameters

The SRU SearchRetrieve request has the following format:

http://<host>:<port>/<base>?<request parameters>

- <host> is the sru_server host
- <port> is the sru_server port
- <base> is the Aleph base code
- <request parameters> are parameters

For example:

http://il-aleph07:5663/usm01?version=1.1&operation=searchRetrieve&query=dc.title=(gold)&maximumRecords=1

The <request parameters> are specified in the following table:

Parameter Name	Mandatory /Optional	Supported	Description
version	Mandatory	Supported	SRU protocol version
query	Mandatory	Supported	Contains a query expressed in CQL to be processed by the server. See CQL Query on page 9 for more information on CQL queries supported by the Aleph SRU client.
startRecord	Optional	Supported	The position within the sequence of matched records of the first record to be returned. The first position in the sequence is 1. The value supplied must be greater than 0. The default value, if none is supplied, is 1.
maximumRecords	Optional	Supported	The number of records requested to be returned. The value must be 0 or greater.

Parameter Name	Mandatory /Optional	Supported	Description
recordPacking	Optional	Supported	A string that determines how the record should be escaped in the response. Defined values are `string` and `xml`. The default is `xml`.
recordSchema	Optional	Supported	The schema requested for the records to be returned. Supported value: `marcxml`.
recordXPath	Optional	Not Supported	An XPath expression, to be applied to the records before returning them.
resultSetTTL	Optional	Not Supported	The number of seconds that the client requests that the created result set should be maintained.
sortKeys	Optional	Not Supported	This parameter contains a sequence of sort keys to be applied to the results.
stylesheet	Optional	Supported	A URL for an XML style sheet. The SRU server adds a reference to this style sheet to the response, enabling the client application to see the response rendered according to rules of the supplied style sheet.
extraRequestData	Optional	Not Supported	This parameter provides additional profile specific information.
operation	Mandatory	Supported	The string `searchRequest`

Request Processing

- The `sru_server` translates the search query from CQL to Z39.50 query language and sends it to the `z39_server`.
- The `z39_server` processes the request and sends the response to the `sru_server`.
- The `sru_server` packs the response into a SRU format and returns it to the SRU client.
- The SRU client parses the XML and creates BIB records in VIR01.

Response Parameters

Parameter name	Mandatory \ Optional	Supported	Description
Version	Mandatory	Supported	The SRU protocol version.
numberOfRecords	Mandatory	Supported	The number of records matched by the query. If the query fails this number is 0.
resultSetId	Optional	Not Supported	The identifier for a result set that is created by the execution of the query.
resultSetIdleTime	Optional	Not Supported	The number of seconds after which the created result set is deleted.
Records	Optional	Supported	A sequence of records matched by the query or surrogate diagnostics. Aleph sru_client is configured to receive marxml records according to: http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd
nextRecordPosition	Optional	Not Supported	The next position within the result set following the final returned record.
diagnostics	Optional	Supported	A sequence of non-surrogate diagnostics generated during execution.
extraResponseData	Optional	Not Supported	Additional information.
echoedSearch RetrieveRequest	Optional	Not Supported	The request parameters echoed back to the client in a simple XML form.

Example of SearchRetrieve Response

```
<?xml version="1.0" ?>
- <zs:searchRetrieveResponse xmlns:zs="http://www.loc.gov/zing/srw/">
  <zs:version>1.1</zs:version>
  <zs:numberOfRecords>1000</zs:numberOfRecords>
- <zs:records>
- <zs:record>
  <zs:recordSchema>info:srw/schema/1/marcxml-v1.1</zs:recordSchema>
  <zs:recordPacking>xml</zs:recordPacking>
- <zs:recordData>
- <record xmlns="http://www.loc.gov/MARC21/slim">
   <leader>00692nam a22002051 4500</leader>
```

```
<controlfield tag="001">000000002-7</controlfield>
<controlfield tag="005">20020418155342.1</controlfield>
<controlfield tag="008">700716s1964 mauach b 000 0 eng</controlfield>
- <datafield tag="010" ind1="" ind2="">
<subfield code="a">68001304</subfield>
</datafield>
- <datafield tag="035" ind1="0" ind2="">
<subfield code="a">(OCoLC)00430285</subfield>
</datafield>
- <datafield tag="040" ind1="" ind2="">
<subfield code="a">DLC</subfield>
<subfield code="c">DLC</subfield>
<subfield code="d">m.c.</subfield>
<subfield code="d">FLL</subfield>
</datafield>
- <datafield tag="050" ind1="0" ind2="">
<subfield code="a">LD2151</subfield>
<subfield code="b">.S7 1964</subfield>
</datafield>
- <datafield tag="245" ind1="0" ind2="4">
<subfield code="a">The Story of Harvard, a short history.</subfield>
</datafield>
- <datafield tag="260" ind1="" ind2="">
<subfield code="a">Cambridge,</subfield>
<subfield code="b">University Information Center</subfield>
<subfield code="c">[1964]</subfield>
</datafield>
- <datafield tag="300" ind1="" ind2="">
<subfield code="a">33 p.</subfield>
<subfield code="b">illus., facsim., ports.</subfield>
<subfield code="c">17 cm.</subfield>
</datafield>
- <datafield tag="504" ind1="" ind2="">
<subfield code="a">"Some books of Harvard history": p. 33.</subfield>
</datafield>
- <datafield tag="610" ind1="2" ind2="0">
<subfield code="a">Harvard University</subfield>
<subfield code="x">History.</subfield>
</datafield>
- <datafield tag="710" ind1="2" ind2="">
<subfield code="a">Harvard University.</subfield>
<subfield code="b">Information Center.</subfield>
</datafield>
- <datafield tag="956" ind1="" ind2="">
<subfield code="a">CONV</subfield>
<subfield code="b">00</subfield>
<subfield code="c">20040307</subfield>
<subfield code="l">USM01</subfield>
<subfield code="h">1200</subfield>
</datafield>
- <datafield tag="956" ind1="" ind2="">
<subfield code="a">CONV</subfield>
<subfield code="b">00</subfield>
<subfield code="c">20040307</subfield>
<subfield code="l">USM01</subfield>
<subfield code="h">1402</subfield>
</datafield>
```

```
</record>
</zs:recordData>
<zs:recordPosition>1</zs:recordPosition>
</zs:record>
</zs:records>
</zs:searchRetrieveResponse>
```

CQL Query

Common Query Language (CQL) is a formal language for representing queries to information retrieval systems such as Web indexes, bibliographic catalogs, and museum collection information. The design objective is that queries be readable and writable by people and that the language be intuitive while maintaining the expressiveness of more complex languages. An SRU search statement is expressed in CQL syntax.

A CQL query is made up of a single search clause or multiple search clauses connected by Boolean operators. Each search clause consists of the following components:

- Index
- Relation
- Search Term

For example: `title = cat`

In this example, the index is `title`, the relation is = and the search term is `cat`.

It is also possible to include only the search term, for example: `cat`

In this example, the search term is `cat`, and no index or relation is expressed in the query.

The index name may include a prefix that defines the context set. For example:

`dc.title = cat`

In this example, the index is `title`. The index belongs to the context set `dc`, the relation is = and the search term is `cat`.

The same index name (for example, `title`) belonging to different context sets or not belonging to any context set may be treated differently by the SRU\SRW server.

A CQL query may also be made up of multiple search clauses connected by boolean operators, for example: `dc.title = history` or `dc.title = art`

A CQL query may contain special characters, for example:

- A truncation sign , for example: `title=histo*`
- An anchor sign (^), for example: `title=^history` – find all titles starting with `history`

The Aleph SRU client conforms to CQL conformance level 1.

The indexes that the SRU client supports are:

- cql.serverChoice – corresponds to the WRD aleph index
- dc.creator – corresponds to the WAU aleph index
- dc.title – corresponds to the WTI aleph index
- dc.subject – corresponds to the WSU aleph index
- dc.publisher – corresponds to the WPU aleph index
- dc.date – corresponds to the WYR aleph index
- dc.language – corresponds to the WLN aleph index
- dc.format – corresponds to the WFM aleph index
- rec.id – corresponds to the DIRECT and SYS aleph index

SRU Search Configuration

Target Configuration File

For each remote base, the `sru_client` configuration is stored in the `./alephe_tab/sru_gate` directory. This directory consists of `.conf` files, one for each SRU target.

The configuration file contains different settings concerning work with this target.

File name: sru_gate_`<target>.conf`

Example of file name: sru_gate_loc_sru.conf (where loc_sru is the SRU target code)

Note:
File name - `<target>` is in lowercase.
The file contains a list of settings. Each setting must appear in a separate line.
Lines starting with # or `!` are comments.

The file must contain the following settings:

- The target's general parameters
- A translation of the CCL search/scan codes into SRU attributes

These settings are explained in the following sections.

Target General Parameters

The following is an example of the general parameters of `./alephe_tab/sru_gate/sru_gate_loc_sru.conf`

```
target   LOC_SRU
hostname z3950.loc.gov:7090
database voyager
recordtype USMARC
version 1.1
```

Target <target name>

`<target name>` is the logical name of the target which is used by `sru_gate`. Values must be in uppercase. The target configuration in `sru_gate_<target>.conf` must start from this parameter.

`<target name>` is the logical name of the target throughout the system. Therefore, the same target code must be present in various setups, such as:

- Aleph configuration: `tab_base.lng`
- Universal Gateway configuration: in file name `<target>.conf`
- Universal Gateway configuration: in `<target>.conf; Z58-BASE-REMOTE <base>` (`base` = `target` and is in uppercase)

hostname <host : port>

<host : port> is the host of the target. The port (if needed) is the port of the target.

database <database name>

`<database name>` is the name of remote database to be searched.

recordtype <record type>

`<record type>` is the desired format of the retrieved record. If the line is not supplied, the format of retrieved record is defined by the target.

version < version >

`<version>` is the SRU version that is used. The default value is `1`.

Translation of CCL Search/Scan Codes into Z39 Attributes

When sending a CCL search query to `sru_gate` from SearchRequest, the CCL search codes are translated to SRU attributes.

This section defines the translation of the CCL qualifier to SRU attributes

Sample of translate parameters of `./alephe_tab/sru_gate/sru_gate_loc_sru.conf`

```
find WRD  cql.serverChoice
find WAU  dc.creator
find WTI  dc.title
find WSU  dc.subject
find WPU  dc.publisher
find WYR  dc.date
find WLN  dc.language
find WFM  dc.format
find DIRECT rec.id
find SYS    rec.id
```

Universal Gateway Configuration File

For each remote library accessed using the Universal Gateway, a configuration file is created in the `$alephe_gate` directory.

File name: `<target>.conf`

Example of file name: `loc_sru.conf` (where `loc_sru` is the SRU target code)

Note:
In file name - `<target>` is in lowercase.
The file contains a list of settings. Each setting must appear in a separate line.
Lines starting with `#` or `!` are comments.

The file must contain the following settings:

- General Settings
- Find Request Settings
- Response Record Settings

These settings are explained in the following sections.

General Settings

General Settings define the general aspects of the target (access-method, character conversion, etc.

The following is an example of the General Settings section of `./alephe/gate/loc_sru.conf`

```
############################################################
#General Settings
############################################################
Z58-BASE-NAME                LOC_SRU
Z58-ACCESS-METHOD            SRU
Z58-BASE-REMOTE              LOC_SRU
Z58-HOST-NAME                z3950.loc.gov:7090
```

The following is an explanation of the parameters:

Z58-BASE-NAME <name>

<name> is the library name displayed in the Web OPAC.

This must be defined if a SID tag or Item Remote link is displayed in the Full Record screen.

Z58-ACCESS-METHOD <access-method>

<access-method> is the method by which remote library is accessed. <access-method> must be SRU. This setting is mandatory.

Z58-BASE-REMOTE <name>

<name> is the logical name of target used in the SRU gate configuration (./alephe_tab/sru_gate/sru_gate_<target>.conf). This setting is mandatory

Find Request Settings

The following is an example of the Find Requests section of ./alephe/gate/loc_sru.conf

```
############################################################
#Find Request Settings
############################################################
Z58-OUT-FIND-CHAR-CONV
```

Z58-OUT-FIND-CHAR-CONV <char-conv>

This parameter defines the character conversion routine applied in a search query. Aleph formulates queries in UTF-8 which must be converted to the character set required by the target database.

The following are possible values of <char-conv>:

- UTF_TO_ANSEL
- UTF_TO_MAB
- UTF_TO_8859_1 and so on.

The full list of possible values is defined in:

$alephe_unicode/tab_character_conversion_line

This setting must be used if the encoding used by the target database for search queries differs from UTF-8.

Response Record Settings

The following is an example of the Response Record section of ./alephe/gate/loc_sru.conf:

```
############################################################
#Present Response Settings
############################################################
Z58-IN-RECORD-TYPE          USMARC
Z58-IN-RECORD-CREATE        vir_z00_sru_marcxml
```

Z58-IN-RECORD-TYPE <response-record-type>

<response-record-type> is the type of records in the remote library.

The following values are possible:

- USMARC
- UNIMARC
- DANMARC
- MAB

This setting is mandatory.

Z58-IN-RECORD-CREATE <program>

<program> is used for converting the incoming records into standard Aleph format.

Possible value is: vir_z00_sru_marcxml (used when Z58-IN-RECORD-TYPE is USMARC).

This setting is mandatory.

Z58-IN-RECORD-CHAR-CONV <char-conv>

<char-conv> defines the character conversion routine applied when converting the incoming records into standard Aleph format. Aleph formulates queries in UTF-8 which must be converted to the character set required by the target database.

The following are possible values of <char-conv>:

- UTF_TO_ANSEL
- UTF_TO_MAB
- UTF_TO_8859_1 and so on.

The full list of possible values is defined in:

$alephe_unicode/tab_character_conversion_line

This setting must be used if the encoding used by the target database for search queries differs from UTF-8.

Z58-IN-RECORD-FIX <fix routine>

<fix routine> defines the fix routine that calls for the incoming records. <fix routine> must be defined in tab_fix of the EXT library to which the target is connected (for example, EXT01).

This setting is optional.

Z58-IN-DIRECT-KEY <key>

<key> defines how subfield $c of SID tag of the record is filled.

Possible value: TAG.

This setting is optional.

Z58-IN-DIRECT-KEY-PARAM <param>

<param> - parameter for Z58-IN-DIRECT-KEY.

Possible values are: 001, 035, 092, 016, and 544.

Defines from which tag of the incoming record the value for subfield $c of SID tag is taken. This setting must be defined if Z58-IN-DIRECT-KEY is defined.

Setup and Configuration Example

This section shows an example setup of an SRU search for base code (target): LOC_SRU (Library of Congress via SRU).

Servers Activation

To enable an SRU search, the following servers must be running:

- SRU server – util w/3/7/8
- Z39.50 server – util w/3/4
- PC server – util w/3/3

./alephe/tab/update tab_base.lng

Set LOC_SRU base in tab_base.lng:

```
!         1          2           3            4     5     6     7   8     9
!!!!!!!!!!!!!!!!!!!!-!-!!!!!!!!!!!!!!!!!!!!-!!!!!-!!!!!-!!!!!-!!!!!-!-!!!!!!!!!>
LOC_SRU                LOC SRU              EXT01            EXT01
```

./alephe/tab/sru_gate/sru_gate_loc_sru.conf

Define the conf file for loc_sru base:

```
##############################################################
# Library of Congress via SRU
##############################################################
target   LOC_SRU
hostname z3950.loc.gov:7090
database voyager
recordtype USMARC
version 1.1

find WRD  cql.serverChoice
```

```
find WAU  dc.creator
find WTI  dc.title
find WSU  dc.subject
find WPU  dc.publisher
find WYR  dc.date
find WLN  dc.language
find WFM  dc.format
find DIRECT rec.id
find SYS    rec.id
```

./alephe/gate/loc_sru.conf

Define the gate for `loc_sru` base:

```
############################################################
## LOC_SRU
############################################################

############################################################
#General Settings
############################################################
Z58-BASE-NAME                LOC_SRU
Z58-ACCESS-METHOD            SRU
Z58-BASE-REMOTE              LOC_SRU
Z58-HOST-NAME                z3950.loc.gov:7090

############################################################
#Find Request Settings
############################################################
Z58-OUT-FIND-CHAR-CONV

############################################################
#Present Response Settings
############################################################
Z58-IN-RECORD-TYPE           USMARC
Z58-IN-RECORD-CREATE         vir_z00_sru_marcxml
Z58-IN-RECORD-CHAR-CONV

############################################################
#Scan Request Settings
############################################################
Z58-IN-SCAN-CHAR-CONV
Z58-OUT-SCAN-CHAR-CONV

############################################################
#Sort Request Settings
############################################################
Z58-SORT                     N

############################################################
#Holdings Info Settings
############################################################
Z58-HOLDING-METHOD           OPAC
```

GUI Configuration .\alephcom\tab\SearBase.dat

Set LOC_SRU base in SearBase.dat (list of databases for GUI search)

```
!!!!!!!!!!!!!!!!!!!!!!!!!!+!!!!!!!!!!!!!!!!!!!!+!!!!!+!!
Library of Congress sru    LOC_SRU              EXT01 0
```

GUI Configuration .\alephcom\tab\Locate.dat

Set LOC_SRU base in Locate.dat (list of databases for locate activities)

```
LOC_SRU                     LOC_SRU              EXT01
```

OPAC Configuration ./alephe/www_f_lng/base-list-include-<xxx01>

Add LOC_SRU to base-list-include-<xxx01> (for example, base-list-include-usm01 - list of databases to search)

```
   <option value=LOC_SRU $$FXX-S"LOC_SRU ">LOC_SRU</option>
```

OPAC Configuration ./alephe/www_f_lng/base-list

Add LOC_SRU to base-list (list of databases)

```
  <td class=text3>
    <img src="&icon_path/f-small-dot.gif" border="0" alt="">
    <a href="&server_f?func=find-b-0&local_base=LOC_SRU">
       Library of Congress  SRU</a>
```

OPAC Configuration ./alephe/www_f_lng/locate-list

Add LOC_SRU to locate-list (list of databases for locate activities)

```
<tr>
  <td class=td2>
     <input type=checkbox name="lib5" value="LOC_SRU"> Library of Congress
via SRU
  </td>
 </tr>
```