

GRX

The Gold Rush XML Web Service

Colorado Alliance of Research Libraries

3801 E. Florida, Ste. 515 • Denver, CO 80210 • (303) 759-3399 • FAX: (303) 759-3363
© Copyright Colorado Alliance 2008

Table of Contents

Introduction to GRX.....	3
Purpose of GRX.....	3
GRX System Components.....	3
SOAP.....	4
SOAP Toolkits.....	4
Requirements for using GRX.....	5
GRX URLs.....	5
GRX and Gold Rush IDs.....	5
GRX FAQ.....	6
What is GRX?.....	6
What can I do with GRX?.....	6
Do I have to use SOAP?.....	6
Do I have to use PHP?.....	6
What do I need to create a GRX client?.....	6
Is there a website where I can test GRX functions?.....	7
GRX Functions.....	8
Browse Functions.....	8
browseJrnIFT.....	8
browseJrnI.....	11
browseJrnISubjects.....	13
browseDB.....	15
browseDBCategory.....	17
Search Functions.....	19
searchByISSN.....	19
searchByDBTitle.....	25
searchByJrnITitle.....	28
searchByKeyword.....	32
searchBySubject.....	36
A to Z List Functions.....	41
journalListA2Z.....	41
databaseListA2Z.....	43
Full Record Functions.....	45
fullJTRec.....	45
fullDBRec.....	50
getDBTitlelist.....	53
Miscellaneous Functions.....	55
grIResolveArticle.....	55
holdingsExport.....	63
How To.....	65
How to install nuSOAP.....	65
How to install Perl SOAP modules.....	65
How to create a GRX client in PHP with NuSOAP.....	67
How to display GRX results on a web page.....	68
How to Increase the Timeout on Your GRX Client.....	69
Sample PHP Client.....	70

Introduction to GRX

Purpose of GRX

As libraries move into greater integration in the presentation of their different Web-based offerings, there is an increasing call for greater flexibility in the way Gold Rush can be queried and result-sets may be displayed and integrated into local Web sites. The traditional Gold Rush public interface has some customization options but a site must work within the confines of its table-driven result sets. The Gold Rush XML Gateway (GRX) is a service that will take Gold Rush “out-of-the-box” so that interaction with it is completely customizable by a library.

With GRX, Gold Rush is now a Web service, accepting XML requests and returning XML responses. The SOAP standard handles the transfer of the XML data. With this functionality, you will be able to make Gold Rush look like almost anything desired since each local site can create and control their own templates and style sheets. The original Gold Rush Public interface will continue to be fully supported for those sites that do not want to use the GRX Web service.

At the present, a library must create its own client application, style sheets and templates for using GRX. But as different organizations develop creative applications, we will be happy to share different approaches to using and integrating this technology.

Since XML result sets can range from an individual record to the extraction of your entire Gold Rush set of holdings, the possibilities for its use are endless. Examples of some applications might include:

- Create an entirely new starting interface for Gold Rush that is tailored to your exact needs
- Create mash-ups or integrate Gold Rush more fully into other library applications or services
- Present the Gold Rush result sets with different data than are currently displayed. For example, you may want to have the initial results of a browse show elements that do not currently appear to limit the need to go to a full record.
- Extract an XML representation of your Gold Rush data to import into a next generation library interface or share with another vendors.

GRX System Components

GRX runs on a SUSE Linux system, running Apache and PHP. At time of release, the Apache server is version 1.3.x and PHP is version 5.3.4. In addition to PHP, GRX uses the nusoap PHP toolkit, version 0.7.2.

GRX uses the SOAP protocol for receiving XML calls and returning XML results.

SOAP

SOAP is a framework for exchanging XML-based messages between applications residing on different computers. It defines a SOAP envelope, which includes a SOAP header and a SOAP body. The XML data is all encapsulated in the SOAP body. The SOAP header includes information about the application or the data in the SOAP body.

While SOAP provides a simple framework for passing XML data between systems, it also creates a lot of overhead, increasing the size of the returned data considerably. It can also be slower than other methods of passing XML data. However, it also provides a method for defining complex data structures, which can be very complicated in some of the more efficient methods.

Because Gold Rush utilizes some rather complicated data structures, SOAP was chosen for GRX. Even with the extra overhead of SOAP, most GRX functions return a fairly small amount of data, and it still performs at an acceptable speed.

Most modern programming languages provide some means of creating a SOAP message. This is particularly true for languages commonly used for web applications, like PHP or Perl.

As of PHP 5, you can use a native PHP SOAP extension. To use it, you need to configure your compile of PHP using the "--enable-soap" switch.

For PHP users, there is also a third-party toolkit called nuSOAP. The nuSOAP toolkit can be downloaded at <http://sourceforge.net/projects/nusoap/>. The nuSOAP toolkit is a simpler solution than the native support in PHP5, and also works well with PHP4. It can simply be placed in an accessible directory on your webserver and called from your PHP scripts. It provides fairly simple methods for creating SOAP servers and clients, as well as for parsing the XML returned in a SOAP package.

To get started with nuSOAP, you might want to look at Scott Nichol's introduction to NuSOAP at <http://www.scottnichol.com/nusoapintro.htm>.

If you want to learn more about the SOAP standard, you can start with the official W3C SOAP page (<http://www.w3.org/TR/soap/>). You may also want to view the W3C SOAP tutorial (<http://www.w3schools.com/soap/default.asp>).

SOAP Toolkits

PHP5: native PHP SOAP functions
 nuSOAP

PHP4: nuSOAP

Perl: SOAP::Lite
 SOAP::Lite::Utility

Note:

The perl SOAP modules are dependent on other perl modules. You will most likely need the following modules as well:

XML::Parser

MIME::Base64

URI

libwww-perl

Other modules are optional, but may be required by your environment. For example, if you use SSL on your website, you will also need the module IO::Socket::SSL.

Requirements for using GRX

To make use of GRX, you will need a webserver with support for a scripting language like PHP or Perl. You will also need the expertise to write scripts in the chosen language that interact with web services (like GRX) and your web server. You will also need some experience working with arrays in the chosen language. Of course, some familiarity with Gold Rush is also required.

If you are using PHP, we suggest you use the nuSOAP toolkit.

GRX URLs

GRX Web Service URL: <http://grx.coalliance.org/grx.php>

GRX Test Site: <http://grx.coalliance.org/grxtest/>

GRX Documentation: (This document)

http://grx.coalliance.org/grxtest/dsp/grx_documentation.pdf (PDF)

http://grx.coalliance.org/grxtest/dsp/grx_documentation.odt (OpenDocument)

GRX and Gold Rush IDs

When GRX is interacting with the Gold Rush database, you will need a Journal Title ID or a Database ID in order to request a full record for a Journal or a Database. In most cases, you will have made one call to GRX using either a browse function or a search function. That call will return a list of items with their ids. Then, you will use an id from the first results to request a full record for the selected item.

This two-step process is exactly what happens in Gold Rush public. In that interface, you perform a search or browse, and view the results in a web page. When you click on one of the titles in your result set, it uses the item's id to find resources in your holdings table for that item. Then, it displays the full record for that item, with all of the resources you have access to.

GRX FAQ

What is GRX?

GRX is a web service providing access to Gold Rush data through functions that mimic the workings of the Gold Rush public interface. Any search or browse that can be performed in Gold Rush public can also be performed through GRX.

You can call GRX functions from a SOAP client that is written in PHP, Perl, C++, Java, or any programming language that provides support for the SOAP protocol.

GRX returns results in XML wrapped in a SOAP envelope. Your SOAP client can parse the results and display them in whatever way you choose.

What can I do with GRX?

GRX has functions that mimic the functions in Gold Rush Public. You can perform any browse or search function that you can perform in Gold Rush Public. You can retrieve full journal or database records. You can retrieve a journal title list for a specific database. You can request an article resolution by Gold Rush Linker, as you would through ArticleFinder. You can even export all of your holdings from Gold Rush to an XML file for your own analysis or to send to another vendor who needs your online holdings.

Do I have to use SOAP?

Yes. GRX relies on SOAP, and it will not understand your requests if you don't use SOAP.

Do I have to use PHP?

No. You can use any programming or scripting language that supports the SOAP protocol. In addition to PHP, this includes Perl, C++, Java, ASP, and probably others as well. Choose the language you are most comfortable with.

What do I need to create a GRX client?

Because GRX is a web service, you will need a client program to interact with it. Since you will also probably want to display your results on the web, you need a web server that is capable of running the program and displaying the results in HTML. Common environments include a Linux system with apache and PHP or Perl, or an MS Windows system with IIS and PHP or Perl. If you already have a webserver, the chances are that you will be able to create a GRX client on that system.

Is there a website where I can test GRX functions?

Yes. <http://grx.coalliance.org/grxtest/> is website that will allow you to review information about each function, and to test each function against the Gold Rush test database.

Much of the information about each function is repeated in this document, but once you have become familiar with using GRX, the website should serve as a handy reference for finding the input parameters or returned array for any given GRX function.

For those of you who are more familiar with XML, you can point your browser to <http://grx.coalliance.org/grx.php>. There you can view the WSDL for GRX. It will tell you what input parameters are accepted by each function, and it will also tell you about the structure of returned arrays.

GRX Functions

All GRX functions return an array of results. In most cases, the returned array is actually an array of arrays. The sub-arrays are used to separate parts of the results into recognizable sections. Most of the functions include a sub-array called "grx_error". If this sub-array is populated, there was an error executing the function. If the "grx_error" sub-array is not populated, the function completed without errors.

Note: It is possible, and even likely, that there will be some sub-arrays that are not populated. When a sub-array is not populated, it just means that there was no data for that sub-array. It does not indicate an error.

Browse Functions

The GRX browse functions allow you to perform the same browses that are available in the Gold Rush Public interface.

browseJrnlFT

Gold Rush Public Equivalent: Browse by Journal Title, limiting to Full Text

Description:

browseJrnlFT accepts an institution code and a letter, word, or phrase and returns a list of all full text Journal titles that begin with the given string in the given institution's holdings. The word or phrase you enter is called a browse key. Given the browse key "A", this function will return all of the full text journals with a title that begins with the letter "A". If your browse key is "american journal", this function will return a list of full text journals with titles that begin "American Journal".

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'BROWSE_KEY'  
)
```

Example Input:

```
array (  
  '001_AUR',  
  'C'  
)
```

The above example would request a search of the holdings of institution 001_AUR (Auraria) for all full text journal titles beginning with the letter "C".

Return Array

```
Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 27250
      [jtitle] => C Magazine
      [issn] => 1480-5472
    )
    [1] => Array
    (
      [id] => 48009
      [jtitle] => C Magazine (Canada)
      [issn] => 1193-8625
    )
    [2] => Array
    (
      [id] => 61833
      [jtitle] => C O R S journal.
      [issn] => 0574-9638
    )
    [3] => Array
    (
      [id] => 68636
      [jtitle] => C R O M O H S
      [issn] => 1123-7023
    )
    [...]
  ),
  [grx_error] =>
)
```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the browse. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```
$inst_code = '001_AUR';
$browse_key = 'C';
$browse_list = $grx_client->call(
    'browseJrnlFT',
    array($inst_code,$browse_key)
);
```

Checking for success...

```
if (is_array($browse_list['grx_error'])) {
    $error_string = $browse_list['grx_error']['grx_error_string'];
    $error_function = $browse_list['grx_error']['grx_error_function'];
    $error_detail = $browse_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

browseJrnl

Gold Rush Public Equivalent: Browse by Journal Title

Description:

browseJrnl accepts an institution code and a letter, word, or phrase and returns a list of all Journal titles that begin with the given string in the given institution's holdings. The word or phrase you enter is called a browse key. Given the browse key "A", this function will return all of the journals with a title that begins with the letter "A". If your browse key is "american journal", this function will return a list of journals with titles that begin "American Journal".

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'BROWSE_KEY'  
)
```

Example Input:

```
array (  
  '001_FCP',  
  'ceramic'  
)
```

The above example would request a search of the holdings of institution 001_FCP (Fort Collins Public) for all journal titles beginning with the word "ceramic".

Return Array

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [id] => 10043  
      [jtitle] => Ceramic Industry  
      [issn] => 0009-0220  
    )  
    [1] => Array  
    (  
      [id] => 16818  
      [jtitle] => Ceramica.  
      [issn] => 0366-6913  
    )  
    [2] => Array  
    (  
      [id] => 12897
```

```

        [jtitle] => Ceramics (Silikaty)
        [issn] => 0862-5468
    )
    [3] => Array
    (
        [id] => 8307
        [jtitle] => Ceramics International
        [issn] => 1385-8947
    )
    [...]
),
[grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the browse. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '001_FCP';
$browse_key = 'ceramic';
$browse_list = $grx_client->call(
    'browseJrnl',
    array($inst_code,$browse_key)
);

```

Checking for success...

```

if (is_array($browse_list['grx_error'])) {
    $error_string = $browse_list['grx_error']['grx_error_string'];
    $error_function = $browse_list['grx_error']['grx_error_function'];
    $error_detail = $browse_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}

```

browseJrnlSubjects

Gold Rush Public Equivalent: Browse by Journal Subject

Description:

browseJrnlSubjects accepts a letter, word, or phrase and returns a list of all journal subjects that begin with the given string. The word or phrase you enter is called a browse key. Given the browse key "A", this function will return all of the journal subjects that begin with the letter "A". If your browse key is "Transportation -- France ", this function will return a list of journals with subjects that begin with "Transportation -- France ".

Input Parameters

```
array (  
  'BROWSE_KEY'  
)
```

Example Input:

```
array (  
  'C'  
)
```

The above example would request a list of journal subjects beginning with the letter "C".

Return Array

```
Array  
(  
  [subjects] => Array  
  (  
    [0] => C (Computer program language)  
    [1] => C++ (Computer program language)  
    [2] => Cabinet officers -- Registers.  
    [3] => Cabinetmaking  
    [4] => CabINETwork  
    [5] => Cable television -- United States  
    [6] => Cactus  
    [7] => CAD/CAM systems  
    [8] => Calcification  
    [9] => Calcification, physiologic  
    [10] => Calcium  
    [...]  
  ),  
  [grx_error] =>  
)
```

In the return array, there is a sub-array labeled "subjects". This sub-array will hold the subjects returned by the browse. The sub-array labeled "grx_error" should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```
$browse_key = 'C';
$browse_list = $grx_client->call(
    'browseJrnlSubjects',
    array($browse_key)
);
```

Checking for success...

```
if (is_array($browse_list['grx_error'])) {
    $error_string = $browse_list['grx_error']['grx_error_string'];
    $error_function = $browse_list['grx_error']['grx_error_function'];
    $error_detail = $browse_list['grx_error']['grx_error_detail'];
    print ">h1<Error: $error_string>/h1<";
    print ">p<Function where error occurred: $error_function>/p<";
    print ">p<Error Detail: $error_detail>/p<";
}
else {
    // process the returned array
}
```

browseDB

Gold Rush Public Equivalent: Browse Databases by Name

Description:

browseDB accepts an institution code and a letter, word, or phrase and returns a list of all databases that begin with the given string in the given institution's holdings. The word or phrase you enter is called a browse key. Given the browse key "A", this function will return all of the databases with a name that begins with the letter "A". If your browse key is "academic", this function will return a list of databases with names that begin "Academic".

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'BROWSE_KEY'  
)
```

Example Input:

```
array (  
  '106_UNM',  
  'A'  
)
```

The above example would request a search of the holdings of institution 106_UNM (Univ. of New Mexico) for all database names beginning with the letter "A".

Return Array

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [dbid] => 2255  
      [dbtitle] => AAAS Journals  
    )  
    [1] => Array  
    (  
      [dbid] => 2958  
      [dbtitle] => AALAS  
    )  
    [2] => Array  
    (  
      [dbid] => 2604  
      [dbtitle] => ABC-CLIO Journals  
    )  
    [3] => Array
```

```

        (
            [dbid] => 2406
            [dbtitle] => ACCP Journals
        )
        [...]
    ),
    [grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the browse. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '106_UNM';
$browse_key = 'A';
$browse_list = $grx_client->call(
    'browseDB',
    array($inst_code,$browse_key)
);

```

Checking for success...

```

if (is_array($browse_list['grx_error'])) {
    $error_string = $browse_list['grx_error']['grx_error_string'];
    $error_function = $browse_list['grx_error']['grx_error_function'];
    $error_detail = $browse_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}

```


browseDBCategory

Gold Rush Public Equivalent: Browse Databases by Category

Description:

browseDBCategory accepts an institution code and a letter, word, or phrase and returns a list of all database categories that begin with the given string, along with any databases in the given institution's holdings that have been assigned that category. The word or phrase you enter is called a browse key. Given the browse key "A", this function will return all of the database categories with a title that begins with the letter "A", and a list of databases in your holdings for each of those categories. If your browse key is "business", this function will return a list of database categories that begin "Business" and a list of databases from your holdings that matches each category.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'BROWSE_KEY'  
)
```

Example Input:

```
array (  
  '001_ARO',  
  'B'  
)
```

The above example would request a search of the holdings of institution 001_ARO (Aurora Public Library) for all database categories beginning with the letter "B".

Return Array

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [subject] => Bibliography  
      [itemlist] => Array  
      (  
        )  
      )  
    )  
    [1] => Array  
    (  
      [subject] => Biography  
      [itemlist] => Array  
      (  
        [0] => Array  
        (  
          [dbid] => 2134  
          [dbtitle] => Biographies Plus  
        )  
      )  
    )  
  )  
)
```

```

    )
    [1] => Array
    (
      [dbid] => 1286
      [dbtitle] => Biography Resource Center
    )
  )
)
[2] => Array
(
  [subject] => Biology
  [itemlist] => Array
  (
    [0] => Array
    (
      [dbid] => 83
      [dbtitle] => Agricola
    )
    [1] => Array
    (
      [dbid] => 2439
      [dbtitle] => Science Resource Center
    )
  )
)
)
[...],
[grx_error] =>
)

```

In the returned array, the sub-array "brief_recs" will hold a numbered array of database categories. The sub-array "itemlist" under the "brief_recs" sub-array will hold a numbered list of databases in your holdings that are assigned the current category. Note that the "itemlist" array may be empty if you do not have any databases with a particular category. In the example above, the "itemlist" array under the "subject" Bibliography is an empty array because the institution does not have any databases that have been assigned the category Bibliography.

Client Code

Calling the function...

```

$inst_code = '001_ARO';
$browse_key = 'B';
$browse_list = $grx_client->call(
    'browseDBCcategory',
    array($inst_code,$browse_key)
);

```

Checking for success...

```

if (is_array($browse_list['grx_error'])) {
    $error_string = $browse_list['grx_error']['grx_error_string'];
    $error_function = $browse_list['grx_error']['grx_error_function'];
    $error_detail = $browse_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}

```

Search Functions

The search functions in GRX allow you to perform the same searches you can perform in Gold Rush Public. All of the GRX search functions accept essentially the same parameters, `inst_code`, `searchterm`, `limit_ft`, and `limit_free`. The `limit_ft` and `limit_free` parameters are optional. If you do not include them, the search will not be limited. If you set either one to the value "Y", and pass that to the function, that limit will be applied.

All of the GRX searches can return either a list of brief records, or a single brief record and a full record. If the search finds multiple hits, the list of brief records is returned. If the search finds a single hit, there will only be one brief record, and the embedded full record will include a full record for the single item that was found.

searchByISSN

Gold Rush Public Equivalent: Search by ISSN

Description:

`searchByISSN` accepts an institution code, an ISSN, and optional limiters for full text or free resources and returns a list of all Journal titles that have the given ISSN in the given institution's holdings. If only one record is found with the given ISSN, GRX will return the full record for the found resource, embedded in the result array. If more than one record is found with the given ISSN, grx returns a list of journal titles and ids with the given ISSN.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'ISSN',  
  'LIMIT_FT',  
  'LIMIT_FREE'  
)
```

Example Input:

```
array (  
  '001_FCP',  
  '0040-781X',  
  'Y',  
  false  
)
```

The above example would request a search of the holdings of institution 001_FCP (Fort Collins Public) for the issn "0040-781X", limiting to full text resources.

Return Array

If multiple records found...

```
Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 2344
      [jtitle] => Management Accounting
      [issn] => 0025-1690
    )
    [1] => Array
    (
      [id] => 58224
      [jtitle] => Strategic Finance (Montvale)
      [issn] => 1524-833X
    )
  )
  [found_resource] =>
  [grx_error] =>
)
```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the search. The sub-array labeled “found_resource” will be empty because it is only used if a single record is found. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

If one record found...

```
Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 3283
      [jtitle] => Scientific American
      [issn] => 0036-8733
    )
  )
  [found_resource] => Array
  (
    [full_rec] => Array
    (
      [title_id] => 3283
      [jtitle] => Scientific American
      [issn] => 0036-8733
      [eissn] =>
      [alt_titles] => Array
      (
        [0] => Science Scope
        [1] => Scientific American microform
        [2] => Scientific American (Abstracts)
        [3] => Scientific American (International Edition)
      )
    )
  )
)
```

```

[subjects] => Array
(
  [0] => Science
  [1] => Technology
  [2] => Science
)

[ft] => Array
(
  [0] => Array
  (
    [dbname] => Academic Search Premier
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=aph&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1995-01-01
    [end] => current
    [grit_msg] =>
  )

  [1] => Array
  (
    [dbname] => Business Source Premier
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=buh&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1995-01-01
    [end] => current
    [grit_msg] =>
  )

  [2] => Array
  (
    [dbname] => Health Source Consumer Edition
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=hph&db=hxh&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1984-01-01
    [end] => current
    [grit_msg] =>
  )

  [3] => Array
  (
    [dbname] => Health Source Nursing/Academic Edition
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=hch&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1995-01-01
    [end] => current
    [grit_msg] =>
  )

  [4] => Array
  (
    [dbname] => MAS Ultra: School Edition
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=ulh&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1995-01-01
    [end] => current
    [grit_msg] =>
  )
)

```

```

)
[lf] => Array
(
  [0] => Array
  (
    [dbname] => Fort Collins Public Library Local Journals
    [url] => http://198.59.47.15/search/i?0036-8733
    [provider] => FORT COLLINS PUBLIC LIBRARY
    [start] =>
    [end] =>
    [grit_msg] =>
  )
)
[cit] => Array
(
  [0] => Array
  (
    [dbname] => Academic Search Premier
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=aph&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1984-01-01
    [end] => current
    [grit_msg] =>
  )
  [1] => Array
  (
    [dbname] => Agricola
    [url] => http://0-search.epnet.com.dalva.fcgov.com/login.asp?
    profile=web
    [provider] => EBSCO INDUSTRIES
    [start] =>
    [end] =>
    [grit_msg] =>
  )
  [2] => Array
  (
    [dbname] => Business Source Premier
    [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
    db=buh&jid=%22SIA%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 1984-01-01
    [end] => current
    [grit_msg] =>
  )
  [3] => Array
  (
    [dbname] => CINAHL Plus
    [url] => http://0-search.ebscohost.com.dalva.fcgov.com/login.asp?
    profile=nursing
    [provider] => EBSCO INDUSTRIES
    [start] =>
    [end] =>
    [grit_msg] =>
  )
  [4] => Array
  (
    [dbname] => ERIC
    [url] => http://0-search.epnet.com.dalva.fcgov.com/login.asp?

```

```

        profile=general
        [provider] => EBSCO INDUSTRIES
        [start] =>
        [end] =>
        [grit_msg] =>
    )

    [5] => Array
    (
        [dbname] => Health Source Consumer Edition
        [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
        db=hph&db=hxh&jid=%22SIA%22&scope=site
        [provider] => EBSCO INDUSTRIES
        [start] => 1995-01-01
        [end] => current
        [grit_msg] =>
    )

    [6] => Array
    (
        [dbname] => Health Source Nursing/Academic Edition
        [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
        db=hch&jid=%22SIA%22&scope=site
        [provider] => EBSCO INDUSTRIES
        [start] => 1984-01-01
        [end] => current
        [grit_msg] =>
    )

    [7] => Array
    (
        [dbname] => MAS Ultra: School Edition
        [url] => http://0-search.epnet.com.dalva.fcgov.com/direct.asp?
        db=ulh&jid=%22SIA%22&scope=site
        [provider] => EBSCO INDUSTRIES
        [start] => 1985-01-01
        [end] => current
        [grit_msg] =>
    )

    [8] => Array
    (
        [dbname] => TOPICsearch
        [url] => http://0-search.epnet.com.dalva.fcgov.com/login.asp?
        profile=tsp
        [provider] => EBSCO INDUSTRIES
        [start] =>
        [end] =>
        [grit_msg] =>
    )
    )
    [grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_rec”. This sub-array will have a single numbered record holding the brief record for the single item that was found by the search. The sub-array labeled “found_resource” will be populated with a full journal title record. For a full explanation of the full record embedded in the “found_resource” sub-array, see documentation for the *fullJTRec* function below. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```
$inst_code = '001_FCP';
$issn = '0040-781X';
$limit_ft = false;
$limit_free = false;
$issn_results = $grx_client->call(
    'searchByISSN',
    array($inst_code,$issn,$limit_ft,$limit_free)
);
```

Checking for success...

```
if (is_array($issn_results['grx_error'])) {
    $error_string = $issn_results['grx_error']['grx_error_string'];
    $error_function = $issn_results['grx_error']['grx_error_function'];
    $error_detail = $issn_results['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```


searchByDBTitle

Gold Rush Public Equivalent: Search by Database Title

Description:

searchByDBTitle accepts an institution code, a searchterm, and optional limiters for full text or free resources and returns a list of all database names that have the given searchterm in the given institution's holdings. If only one record is found with the given searchterm, GRX will return the full record for the found database. If more than one record is found with the given searchterm, grx returns a list of database names and ids with the given searchterm.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'SEARCHTERM',  
  'LIMIT_FT',  
  'LIMIT_FREE'  
)
```

Example Input:

```
array (  
  '001_FCP',  
  'eric',  
  false,  
  false  
)
```

The above example would request a search of the holdings of institution 001_FCP (Fort Collins Public) for databases with "business" in their names.

Return Array

If multiple records found...

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [dbid] => 41  
      [dbtitle] => Business Source Premier  
    )  
    [1] => Array  
    (  
      [dbid] => 2052  
      [dbtitle] => Regional Business News  
    )  
  )  
  [found_resource] =>  
  [grx_error] =>  
)
```

In the return array, there is a sub-array labeled "brief_rec". This sub-array will hold the brief records returned by the search. The sub-array labeled "found_resource" will be empty because it is only used if a single record is found. The sub-array labeled "grx_error" should be empty, indicating that the function completed without errors.

If one record found...

```
Array
(
  [brief_rec] => Array
  (
    [0] => Array
    (
      [dbid] => 7
      [dbtitle] => ERIC
    )
  )
  [found_resource] => Array
  (
    [full_rec] => Array
    (
      [dbname] => ERIC
      [dbdescription] => The ERIC (Educational Resources Information Center)
      database is sponsored by the U.S. Department of Education to provide
      extensive access to educational-related literature. The ERIC database
      corresponds to two printed journals:Resources in Education (RIE) and
      Current Index to Journals in Education (CIJE) . Both journals
      provide access to some 14,000 documents and over
      20,000 journal articles per year.
```

In addition, ERIC provides coverage of conferences, meetings, government documents, theses, dissertations, reports, audiovisual media, bibliographies, directories, books and monographs.

Major areas of coverage include:
Adult, Career, and Vocational Education
Counseling and Personnel Services
Elementary and Early Childhood Education
Education Management
Handicapped and Gifted Children
Higher Education
Information Resources
Junior Colleges
Languages and Linguistics
Reading and Communication Skills
Rural Education and Small Schools
Science, Mathematics, and Environmental Education
Social Studies and Social Science Education
Teacher Education
Tests, Measurement, and Evaluation
Teacher Education
Test, Measurement, and Evaluation
Urban Education
[titlecount] => 1277
[subjectlist] => Array
(
 [0] => Education
)
[provlist] => Array
(
 [0] => Array
(
 [provname] => EBSCO INDUSTRIES

```

        [dburl] =>
            http://0-search.epnet.com.dalva.fcgov.com/login.asp?profile=general
        [grit_msg] =>
    )
    )
    [provcount] => 1
    [dbtype] => Citation
    )
    [grx_error] =>
    )
    [grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will have a single numbered record holding the brief record for the single item that was found by the search. The sub-array labeled “found_resource” will be populated with a full database title record. For a full explanation of the full record embedded in the “found_resource” sub-array, see documentation for the *fullDBRec* function below. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '001_FCP';
$searchterm = 'eric';
$limit_ft = false;
$limit_free = false;
$db_title_list = $grx_client->call(
    'searchByDBTitle',
    array($inst_code,$searchterm,$limit_ft,$limit_free)
);

```

Checking for success...

```

if (is_array($db_title_list['grx_error'])) {
    $error_string = $db_title_list['grx_error']['grx_error_string'];
    $error_function = $db_title_list['grx_error']['grx_error_function'];
    $error_detail = $db_title_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}

```

searchByJrnlTitle

Gold Rush Public Equivalent: Search by Journal Title

Description:

searchByJrnlTitle accepts an institution code, a searchterm, and optional limiters for full text or free resources and returns a list of all Journal titles that have the given searchterm in the given institution's holdings. If only one record is found with the given searchterm, GRX will return the full record for the found resource. If more than one record is found with the given searchterm, grx returns a list of journal titles and ids with the given searchterm.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'SEARCHTERM',  
  'LIMIT_FT',  
  'LIMIT_FREE'  
)
```

Example Input:

```
array (  
  '001_FCP',  
  'ceramic',  
  'Y',  
  false  
)
```

The above example would request a search of the holdings of institution 001_FCP (Fort Collins Public) for the journals with "ceramic" in their titles, limiting to full text resources.

Return Array

If multiple records found...

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [id] => 47561  
      [jtitle] => Advanced Ceramics Report  
      [issn] => 0268-9847  
    )  
    [1] => Array  
    (  
      [id] => 90924  
      [jtitle] => Advances in Applied Ceramics  
      [issn] => 1743-6753  
    )  
    [2] => Array
```

```

    (
      [id] => 267200
      [jtitle] => Advances in Applied Ceramics
      [issn] => 1743-6761
    )
    [...]
  )
  [found_resource] =>
  [grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the search. The sub-array labeled “found_resource” will be empty because it is only used if a single record is found. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

If one record found...

```

Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 16818
      [jtitle] => Ceramica.
      [issn] => 0366-6913
    )
  )

  [found_resource] => Array
  (
    [full_rec] => Array
    (
      [title_id] => 16818
      [jtitle] => Ceramica.
      [issn] => 0366-6913
      [eissn] =>
      [alt_titles] => Array
      (
        [0] => Cer amica electronic resource
        [1] => Cer mica
        [2] => Cera??mica
        [3] => Ceramica
        [4] => Ceramica (Sao Paulo)
        [5] => CERAMICA -SAO PAULO-
        [6] => CERAMICA ASSOCIACAO BRASILEIRO DE CERAMICA ONLINE
        [7] => Ceramica.
      )
      [subjects] =>
      [ft] => Array
      (
        [0] => Array
        (
          [dbname] => Directory of Open Access Journals
          [url] => http://www.scielo.br/ce
          [provider] => LUND UNIVERSITY
          [start] => 1997-01-01
          [end] => current
          [grit_msg] =>
        )
        [1] => Array
        (

```

```

        [dbname] => SCIELO Brazil
        [url] => http://www.scielo.br/scielo.php/script_sci_serial/
        pid_0366-6913/lng_en/nrm_iso
        [provider] => SCIENTIFIC ELECTRONIC LIBRARY ONLINE
        [start] => 1997-05-01
        [end] => current
        [grit_msg] =>
    )
)
[1ft] => Array
(
)
[cit] => Array
(
    [0] => Array
    (
        [dbname] => ArticleFirst
        [url] => http://0-firstsearch.oclc.org.skyline.cudenver.edu/
        dbname=ArticleFirst;autho=xxx-xxx-xxx;FSIP
        [provider] => OCLC
        [start] =>
        [end] =>
        [grit_msg] =>
    )
    [1] => Array
    (
        [dbname] => GeoRef
        [url] => http://0-search.epnet.com.skyline.cudenver.edu/login.asp?
        profile=web&defaultdb=geh
        [provider] => EBSCO INDUSTRIES
        [start] =>
        [end] =>
        [grit_msg] =>
    )
)
)
[grx_error] =>
)
[grx_error] =>
)
)

```

In the return array, there is a sub-array labeled “brief_rec”. This sub-array will have a single numbered record holding the brief record for the single item that was found by the search. The sub-array labeled “found_resource” will be populated with a full journal title record. For a full explanation of the full record embedded in the “found_resource” sub-array, see documentation for the *fullJTRec* function below. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '001_FCP';
$issn = '0040-781X';
$limit_ft = false;
$limit_free = false;
$journal_list = $grx_client->call(
    'searchByJrnlTitle',
    array($inst_code,$issn,$limit_ft,$limit_free)
);

```

Checking for success...

```
if (is_array($journal_list['grx_error'])) {
    $error_string = $journal_list['grx_error']['grx_error_string'];
    $error_function = $journal_list['grx_error']['grx_error_function'];
    $error_detail = $journal_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

searchByKeyword

Gold Rush Public Equivalent: Search by Keyword

Description:

searchByKeyword accepts an institution code, a searchterm, and optional limiters for full text or free resources and returns a list of all journals that have the given searchterm in the given institution's holdings. If only one record is found with the given searchterm, GRX will return the full record for the found resource. If more than one record is found with the given searchterm, grx returns a list of journal titles and ids with the given searchterm.

Keyword searches in Gold Rush search the journal's title, alternate titles, and subjects.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'SEARCHTERM',  
  'LIMIT_FT',  
  'LIMIT_FREE'  
)
```

Example Input:

```
array (  
  '001_AUR',  
  'ceramic',  
  'Y',  
  false  
)
```

The above example would request a search of the holdings of institution 001_AUR (Auraria Library) for journals with "ceramic" in their keyword list, limiting to full text resources.

Return Array

If multiple records found...

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [id] => 47561  
      [jtitle] => Advanced Ceramics Report  
      [issn] => 0268-9847  
    )  
  )  
  [1] => Array
```



```

    (
      [id] => 90924
      [jtitle] => Advances in Applied Ceramics
      [issn] => 1743-6753
    )
  [2] => Array
  (
    [id] => 205
    [jtitle] => American Ceramic Society Bulletin
    [issn] => 0002-7812
  )
  [...]
)
[found_resource] =>
[grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the search. The sub-array labeled “found_resource” will be empty because it is only used if a single record is found. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

If one record found...

```

Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 90924
      [jtitle] => Advances in Applied Ceramics
      [issn] => 1743-6753
    )
  )
  [found_resource] => Array
  (
    [full_rec] => Array
    (
      [title_id] => 90924
      [jtitle] => Advances in Applied Ceramics
      [issn] => 1743-6753
      [eissn] =>
      [alt_titles] => Array
      (
        [0] => Advances in applied ceramics : structural, functional
        and bioceramics
        [1] => Advances in applied ceramics : structural, functional
        and bioceramics.
        [2] => Advances in Applied Ceramics: Structural, Functional
        & Bioceramics
      )
      [subjects] =>
      [ft] => Array
      (
        [0] => Array
        (
          [dbname] => Academic Search Elite
          [url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?
          db=afh&jid=%221E2Q%22&scope=site
          [provider] => EBSCO INDUSTRIES
          [start] => 2005-01-01
        )
      )
    )
  )
)

```

```

    [end] => current
    [grit_msg] =>
  )
  [1] => Array
  (
    [dbname] => Academic Search Premier
    [url] => http://0-search.ebscohost.com.skyline.cudenver.edu/direct.asp?db=aph&jid=%221E2Q%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 2005-02-01
    [end] => current
    [grit_msg] =>
  )
)
[lft] => Array
(
)
[cit] => Array
(
  [0] => Array
  (
    [dbname] => Academic Search Elite
    [url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?db=afh&jid=%221E2Q%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 2005-01-01
    [end] => current
    [grit_msg] =>
  )
  [1] => Array
  (
    [dbname] => Academic Search Premier
    [url] => http://0-search.ebscohost.com.skyline.cudenver.edu/direct.asp?db=aph&jid=%221E2Q%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 2005-02-01
    [end] => current
    [grit_msg] =>
  )
  [2] => Array
  (
    [dbname] => Compendex
    [url] => http://0-www.engineeringvillage2.org.skyline.cudenver.edu/controller/servlet/Controller?CID=quickSearch&database=combined
    [provider] => ELSEVIER
    [start] =>
    [end] =>
    [grit_msg] =>
  )
  [3] => Array
  (
    [dbname] => Web of Science
    [url] => http://0-portal.isiknowledge.com.skyline.cudenver.edu/portal.cgi?DestApp=WOS&Func=Frame
    [provider] => INSTITUTE FOR SCIENTIFIC INFORMATION - ISI
    [start] =>
    [end] =>
    [grit_msg] =>
  )
)
)
)
[grx_error] =>
)
[grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will have a single numbered record holding the brief record for the single item that was found by the search. The sub-array labeled “found_resource” will be populated with a full journal title record. For a full explanation of the full record embedded in the “found_resource” sub-array, see documentation for the *fullJTRec* function below. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```
$inst_code = '001_FCP';
$searchterm = 'ceramic';
$limit_ft = false;
$limit_free = false;
$search_results = $grx_client->call(
    'searchByKeyword',
    array($inst_code,$searchterm,$limit_ft,$limit_free)
);
```

Checking for success...

```
if (is_array($search_results['grx_error'])) {
    $error_string = $search_results['grx_error']['grx_error_string'];
    $error_function = $search_results['grx_error']['grx_error_function'];
    $error_detail = $search_results['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

searchBySubject

Gold Rush Public Equivalent: Search by Subject

Description:

searchBySubject accepts an institution code, a subject, and optional limiters for full text or free resources and returns a list of all journals that have the given subject in the given institution's holdings. If only one record is found with the given subject, GRX will return the full record for the found resource. If more than one record is found with the given subject, grx returns a list of journal titles and ids with the given subject.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'SEARCHTERM',  
  'LIMIT_FT',  
  'LIMIT_FREE'  
)
```

Example Input:

```
array (  
  '001_AUR',  
  'ceramic',  
  'Y',  
  false  
)
```

The above example would request a search of the holdings of institution 001_AUR (Auraria Library) for journals with "ceramic" in their subject list, limiting to full text resources.

Return Array

If multiple records found...

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [id] => 205  
      [jtitle] => American Ceramic Society Bulletin  
      [issn] => 0002-7812  
    )  
    [1] => Array  
    (  
      [id] => 14553  
      [jtitle] => American ceramics.  
      [issn] => 0278-9507  
    )  
  )  
)
```

```

)
[2] => Array
(
  [id] => 12821
  [jtitle] => British Ceramic Transactions
  [issn] => 0967-9782
)
[...]
)
[found_resource] =>
[grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the search. The sub-array labeled “found_resource” will be empty because it is only used if a single record is found. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

If one record found...

```

Array
(
  [brief_recs] => Array
  (
    [0] => Array
    (
      [id] => 9334
      [jtitle] => Fungal Genetics & Biology
      [issn] => 1087-1845
    )
  )
  [found_resource] => Array
  (
    [full_rec] => Array
    (
      [title_id] => 9334
      [jtitle] => Fungal Genetics & Biology
      [issn] => 1087-1845
      [eissn] => 1096-0937
      [alt_titles] => Array
      (
        [0] => Experimental Mycology
        [1] => Fungal genet. biol.
        [2] => Fungal Genetics & Biology
        [3] => FUNGAL GENETICS AND BIOLOGY
        [4] => Fungal genetics and biology electronic resource : FG & B
        [5] => Fungal genetics and biology (Online)
        [6] => Fungal genetics and biology (Online).
        [7] => Fungal genetics and biology (Print)
        [8] => Fungal genetics and biology : FG & B.
        [9] => Fungal genetics and biology : FG & B
        [10] => Fungal genetics and biology : FG & B.
        [11] => FUNGAL GENETICS AND BIOLOGY FG B
      )
    )
    [subjects] => Array
    (
      [0] => Biology
      [1] => Fungi -- Genetics
      [2] => Fungi
    )
  )
)

```

```

    [3] => Mycology
    [4] => Botany
  )
  [ft] => Array
  (
    [0] => Array
    (
      [dbname] => ScienceDirect Freedom Collection (Colorado Alliance)
      [url] => http://0-www.sciencedirect.com.skyline.cudenver.edu/
      science/journal/10871845
      [provider] => ELSEVIER
      [start] => 1993-01-01
      [end] => current
      [grit_msg] =>
    )
  )
  [lft] => Array
  (
    [0] => Array
    (
      [dbname] => Auraria Local Library Journals
      [url] => http://132.194.32.30/search/i?1096-0937
      [provider] => AURARIA LIBRARY
      [start] =>
      [end] =>
      [grit_msg] =>
    )
  )
  [cit] => Array
  (
    [0] => Array
    (
      [dbname] => Academic Search Premier
      [url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
      direct.asp?db=aph&jid=%222TM%22&scope=site
      [provider] => EBSCO INDUSTRIES
      [start] => 2002-02-01
      [end] => current
      [grit_msg] =>
    )
    [1] => Array
    (
      [dbname] => Agricola
      [url] => http://0-search.epnet.com.skyline.cudenver.edu/login.asp?
      profile=web&defaultdb=agr
      [provider] => EBSCO INDUSTRIES
      [start] =>
      [end] =>
      [grit_msg] =>
    )
    [2] => Array
    (
      [dbname] => ArticleFirst
      [url] => http://0-firstsearch.oclc.org.skyline.cudenver.edu/
      dbname=ArticleFirst;autho=xxx-xxx-xxx;FSIP
      [provider] => OCLC
      [start] =>
      [end] =>
      [grit_msg] =>
    )
    [3] => Array
    (
      [dbname] => Biosis
      [url] => http://0-search.epnet.com.skyline.cudenver.edu/login.asp?
      profile=bioabs

```

```

    [provider] => EBSCO INDUSTRIES
    [start] =>
    [end] =>
    [grit_msg] =>
  )
[4] => Array
(
  [dbname] => Medline (PUBMED)
  [url] => http://www.ncbi.nlm.nih.gov/entrez/query.fcgi
  [provider] => NATIONAL LIBRARY OF MEDICINE
  [start] =>
  [end] =>
  [grit_msg] =>
)
[5] => Array
(
  [dbname] => Web of Science
  [url] => http://0-portal.isiknowledge.com.skyline.cudenver.edu/portal.cgi?DestApp=WOS&Func=Frame
  [provider] => INSTITUTE FOR SCIENTIFIC INFORMATION - ISI
  [start] =>
  [end] =>
  [grit_msg] =>
)
)
)
)
[grx_error] =>
)
[grx_error] =>
)
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will have a single numbered record holding the brief record for the single item that was found by the search. The sub-array labeled “found_resource” will be populated with a full journal title record. For a full explanation of the full record embedded in the “found_resource” sub-array, see documentation for the *fullJTRec* function below. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '001_AUR';
$issn = 'Fungi -- Genetics';
$limit_ft = false;
$limit_free = false;
$search_results = $grx_client->call(
    'searchBySubject',
    array($inst_code,$searchterm,$limit_ft,$limit_free)
);

```

Checking for success...

```
if (is_array($search_results['grx_error'])) {
    $error_string = $search_results['grx_error']['grx_error_string'];
    $error_function = $search_results['grx_error']['grx_error_function'];
    $error_detail = $search_results['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```


A to Z List Functions

journalListA2Z

Description:

journalListA2Z accepts an institution code and returns a list of all journals in the given institution's holdings.

This function can take several minutes to return your results. Because of this, you will probably need to increase the timeout for your client before calling this function.

If you have 20,000 journals, this function should take about 2 to 3 minutes. If you have 50,000 journals, it should take about 6 to 7 minutes.

Input Parameters

```
array (  
  'GR_INST_CODE'  
)
```

Example Input:

```
array (  
  '001_FCP'  
)
```

The above example would request a journal list for 001_FCP (Fort Collins Public).

Return Array

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [id] => 14444  
      [jtitle] => 'Teen.'  
      [issn] => 0040-2001  
    )  
    [1] => Array  
    (  
      [id] => 166128  
      [jtitle] => (vwa)  
      [issn] => 0259-6512  
    )  
    [2] => Array  
    (  
      [id] => 16659  
      [jtitle] => '.EXE : The Software Developers' Magazine  
    )  
  )  
)
```

```

        [issn] => 0268-6872
    )
    [...]
)
[grx_error] =>
)

```

In the return array, there is a sub-array labeled “brief_recs”. This sub-array will hold the brief records returned by the search. The sub-array labeled “grx_error” should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```

$inst_code = '001_FCP';
$journal_list = $grx_client->call(
    'journalListA2Z',
    array($inst_code)
);

```

Checking for success...

```

if (is_array($journal_list['grx_error'])) {
    $error_string = $journal_list['grx_error']['grx_error_string'];
    $error_function = $journal_list['grx_error']['grx_error_function'];
    $error_detail = $journal_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}

```

databaseListA2Z

Description:

databaseListA2Z accepts an institution code and returns a list of all databases in the given institution's holdings.

Input Parameters

```
array (  
  'GR_INST_CODE'  
)
```

Example Input:

```
array (  
  '001_MSC'  
)
```

The above example would request a database list for 001_MSC (Mesa State College).

Return Array

```
Array  
(  
  [brief_recs] => Array  
  (  
    [0] => Array  
    (  
      [dbid] => 2255  
      [dbtitle] => AAAS Journals  
    )  
    [1] => Array  
    (  
      [dbid] => 777  
      [dbtitle] => ACM Digital Library  
    )  
    [2] => Array  
    (  
      [dbid] => 1270  
      [dbtitle] => ACS Web Editions  
    )  
    [...]  
  )  
  [grx_error] =>  
)
```

In the return array, there is a sub-array labeled "brief_recs". This sub-array will hold the brief records returned by the search. The sub-array labeled "grx_error" should be empty, indicating that the function completed without errors.

Client Code

Calling the function...

```
$inst_code = '001_FCP';
$database_list = $grx_client->call(
    'databaseListA2Z',
    array($inst_code)
);
```

Checking for success...

```
if (is_array($database_list['grx_error'])) {
    $error_string = $database_list['grx_error']['grx_error_string'];
    $error_function = $database_list['grx_error']['grx_error_function'];
    $error_detail = $database_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

Full Record Functions

fullJTRec

Gold Rush Public Equivalent: Full Journal Record

Description:

fullJTRec accepts an institution code and a journal id, and returns a full record for the specified journal from the institution's holdings.

You can obtain journal ids from the results of the journal search or browse functions, the journalListA2Z function, or the getDBTitles function.

Input Parameters

```
array (
  'GR_INST_CODE',
  'JOURNAL_ID'
)
```

Example Input:

```
array (
  '001_AUR',
  '3528'
)
```

The above example would request a full record for the journal with id 3528 (Time) from the 001_AUR (Auraria Library) holdings.

Return Array

```
Array
(
  [full_rec] => Array
  (
    [title_id] => 22226
    [jtitle] => Ethical Theory and Moral Practice
    [issn] => 1386-2820
    [eissn] =>
    [alt_titles] => Array
    (
      [0] => Ethical Theory & Moral Practice
      [1] => Ethical theory & moral practice electronic resource
      [2] => Ethical Theory and Moral Practice
      [3] => Ethical theory and moral practice (Online)
      [4] => Ethical theory and moral practice (Online).
      [5] => Ethical Theory and Moral Practice: An International Forum
    )
  )
)
```

```

)
[subjects] => Array
(
  [0] => Ethics
  [1] => Social Usages
  [2] => Etiquette
)
[ft] => Array
(
  [0] => Array
  (
    [dbname] => Academic Search Elite
    [url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?
db=afh&jid=%22OEY%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 2003-01-01
    [end] => current
    [grit_msg] =>
  )
  [1] => Array
  (
    [dbname] => Academic Search Premier
    [url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
direct.asp?db=aph&jid=%22OEY%22&scope=site
    [provider] => EBSCO INDUSTRIES
    [start] => 2003-03-01
    [end] => current
    [grit_msg] =>
  )
  [2] => Array
  (
    [dbname] => Advanced Placement Source
    [url] => http://0-search.epnet.com.skyline.cudenver.edu/
    [provider] => EBSCO INDUSTRIES
    [start] => 2003-01-01
    [end] => current
    [grit_msg] =>
  )
  [3] => Array
  (
    [dbname] => Expanded Academic Index ASAP
    [url] => http://0-find.galegroup.com.skyline.cudenver.edu/itw/
infomark/1/1/1/purl=rc18%5fEAIM%5F0%5F%5Fjn+%22Ethical
+Theory+%22Moral+Practice
%22?sw\_aep=xxxxxxx
    [provider] => GALE GROUP
    [start] => 1998-03-01
    [end] => current
    [grit_msg] =>
  )
  [4] => Array
  (
    [dbname] => Springer LINK (Springer Online Journals)
    [url] => http://0-www.springerlink.com.skyline.cudenver.edu/
openurl.asp?genre=journal&issn=1386-2820
    [provider] => SPRINGER
    [start] => 1998-01-01
    [end] => current
    [grit_msg] =>
  )
)
[lft] => Array
(
  [0] => Array
  (

```

```

[dbname] => Auraria Local Library Journals
[url] => http://132.194.32.30/search/i?1386-2820
[provider] => AURARIA LIBRARY
[start] =>
[end] =>
[grit_msg] =>
)
)
[cit] => Array
(
[0] => Array
(
[dbname] => Academic Search Elite
[url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?
db=afh&jid=%22OEY%22&scope=site
[provider] => EBSCO INDUSTRIES
[start] => 1998-03-01
[end] => current
[grit_msg] =>
)
[1] => Array
(
[dbname] => Academic Search Premier
[url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
direct.asp?db=aph&jid=%22OEY%22&scope=site
[provider] => EBSCO INDUSTRIES
[start] => 2003-03-01
[end] => current
[grit_msg] =>
)
[2] => Array
(
[dbname] => Advanced Placement Source
[url] => http://0-search.epnet.com.skyline.cudenver.edu/
[provider] => EBSCO INDUSTRIES
[start] => 1998-03-01
[end] => current
[grit_msg] =>
)
[3] => Array
(
[dbname] => ArticleFirst
[url] => http://0-firstsearch.oclc.org.skyline.cudenver.edu/
dbname=ArticleFirst;autho=xxx-xxx-xxx;FSIP
[provider] => OCLC
[start] =>
[end] =>
[grit_msg] =>
)
[4] => Array
(
[dbname] => Expanded Academic Index ASAP
[url] => http://0-find.galegroup.com.skyline.cudenver.edu/itw/
infomark/1/1/1/purl=rc18%5fEAIM%5F0%5F%5Fjn+%22Ethical+
Theory+&+Moral+Practice
%22?sw\_aep=auraria\_main
[provider] => GALE GROUP
[start] => 1998-03-01
[end] => current
[grit_msg] =>
)
[5] => Array
(
[dbname] => Humanities Index with Full Text
[url] => http://0-vnweb.hwwilsonweb.com.skyline.cudenver.edu/

```

```

    hww/Journals/searchAction.jhtml?sid=HWW:HUM&issn=1386-2820
    [provider] => WILSON
    [start] => 2003-03-01
    [end] => current
    [grit_msg] =>
  )
  [6] => Array
  (
    [dbname] => Philosopher's Index
    [url] => http://0-web5s.silverplatter.com.skyline.cudenver.edu/
    webspirs/start.ws?customer=xxxxxxx&databases=PHIL
    [provider] => SILVER PLATTER
    [start] =>
    [end] =>
    [grit_msg] =>
  )
)
)
[grx_error] =>
)

```

The returned array for a full Journal record starts with a sub-array labeled “full_rec”. This sub-array will hold all of the information about the journal. Under the “full_rec” array, will be a mix of elements and sub-arrays. The simple elements include the journals id (title_id), the title (jtitle), the ISSN (issn), and the EISSN (eissn).

There are sub-arrays for alternate titles (alt_titles), subjects (subjects), full text resources (ft), local full text resources (lft), and citation only resources (cit). The “ft”, “lft”, and “cit” sub-arrays hold information about the resources for each of those resource types. Under each, there is a numbered array of resources, each with data about the resource.

The information about the resource includes the database name (dbname), URL (url), Provider (provider), Start date (start), End date (end), and a Gold Rush Incident Tracker message (grit_msg). In the “ft” sub-array, the start and end dates refer to full text start and end dates in your holdings. In the “cit” sub-array, the start and end date refer to the citation start date and citation end date in your holdings.

As usual, we want the grx_error sub-array to be empty, indicating that no errors occurred during execution of the function.

Client Code

Calling the function...

```

$inst_code = '001_AUR';
$jtid = '3528';
$journal_rec = $grx_client->call(
    'fullJTRec',
    array($inst_code,$jtid)
);

```


Checking for success...

```
if (is_array($journal_rec['grx_error'])) {
    $error_string = $journal_rec['grx_error']['grx_error_string'];
    $error_function = $journal_rec['grx_error']['grx_error_function'];
    $error_detail = $journal_rec['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

fullDBRec

Gold Rush Public Equivalent: Full Database Record

Description:

fullDBRec accepts an institution code and a database id, and returns a full record for that DB.

You can obtain database ids from the results of the database search and browse functions or the databaseListA2Z function.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'DB_ID'  
)
```

Example Input:

```
array (  
  '001_AUR',  
  '83'  
)
```

The above example would request a full record for the database with id "83" (Agricola) in the 001_AUR (Auraria Library) holdings.

Return Array

```
Array  
(  
  [full_rec] => Array  
  (  
    [dbname] => Agricola  
    [dbdescription] => The AGRICOLA (Agricultural Online Access) database  
    contains bibliographic records of materials acquired by the National  
    Agricultural Library (NAL) and cooperating institutions in the  
    agricultural and related sciences. Ninety percent of the records  
    describe journal articles and book chapters, and the remaining ten  
    percent describe monographs, series, microforms, audiovisuals, maps,  
    and other types of materials. Together they provide worldwide coverage  
    of the agricultural literature.
```

In addition, AGRICOLA contains subfiles of related bibliographic citations that have been prepared by sources other than the National Agricultural Library, for example, the Food and Nutrition Information Center (FNIC) and the American Agricultural Economics Documentation Center (AAEDC). These and other information centers and cooperators contribute subfiles to AGRICOLA covering special subjects.

Pre-1979 AGRICOLA records are derived from the CAIN format; from 1979 on, MARC-formatted records are used to create the database.

```

    [titlecount] => 4239
    [subjectlist] => Array
    (
      [0] => Agriculture
    )
    [provlist] => Array
    (
      [0] => Array
      (
        [provname] => EBSCO INDUSTRIES
        [dburl] => http://0-search.epnet.com.skyline.cudenver.edu/login.asp?
profile=web&defaultdb=agr
        [grit_msg] =>
      )
      [provcount] => 1
      [dbtype] => Citation
    )
  )
  [grx_error] =>
)

```

The returned array for a full Database record starts with a sub-array labeled “full_rec”. This sub-array will hold all of the information about the database. Under the “full_rec” array, will be a mix of elements and sub-arrays. The simple elements include the database name (dbname), the description (dbdescription), the number of journal titles in the database (titlecount), the number of providers (provcount), and the type of database (dbtype).

There are sub-arrays for database categories/subjects (subjectlist), and providers (provlist). The “subjectlist” sub-array holds a numbered list of categories assigned to this database. The “provlist” sub-array holds a numbered array of the providers of that database in your holdings. For each provider, there are elements holding the provider name (provname), the database URL (dburl), and any Gold Rush Incident Tracker message (grit_msg).

As usual, we want the grx_error sub-array to be empty, indicating that no errors occurred during execution of the function.

Client Code

Calling the function...

```

$inst_code = '001_AUR';
$dbid = '83';
$database_rec = $grx_client->call(
    'fullDBRec',
    array($inst_code,$dbid)
);

```

Checking for success...

```
if (is_array($database_rec['grx_error'])) {
    $error_string = $database_rec['grx_error']['grx_error_string'];
    $error_function = $database_rec['grx_error']['grx_error_function'];
    $error_detail = $database_rec['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

getDBTitlelist

Gold Rush Public Equivalent: View Title List (on Full DB Record)

Description:

getDBTitlelist accepts an institution code and a database id, and returns a title list for that DB.

Input Parameters

```
array (  
  'GR_ONST_CODE',  
  'DB_ID'  
)
```

Example Input:

```
array (  
  '001_FCP',  
  '83'  
)
```

The above example would request a title list for the database with id "83" (Agricola) from the holdings of institution 001_FCP (Fort Collins Public).

Return Array

```
Array  
(  
  [title_list] => Array  
  (  
    [0] => Array  
    (  
      [id] => 2807  
      [jtitle] => Aaas Annual Meeting & Science Innovation Exposition  
      [issn] => 0031-403X  
    )  
    [1] => Array  
    (  
      [id] => 12620  
      [jtitle] => Abhandlungen des Naturwissenschaftlichen Vereins in Hamburg  
      [issn] => 0173-7481  
    )  
    [2] => Array  
    (  
      [id] => 12622  
      [jtitle] => Abhandlungen und Verhandlungen des Naturwissenschaftlichen  
      Vereins in Hamburg  
      [issn] => 0301-2697  
    )  
    [...]  
  )  
  [grx_error] =>  
)
```

The returned array has a sub-array labeled “title_list”. The title_list sub-array holds a numbered list of journals that are in the current database. For each database, there are elements for the journal id (id), the journal title (jtitle), and the ISSN (issn).

The “grx_error” sub-array should be empty unless an error occurred during processing the function.
Client Code

Calling the function...

```
$inst_code = '001_FCP';
$dbid = '83';
$db_title_list = $grx_client->call(
    'getDBTitlelist',
    array($inst_code,$dbid)
);
```

Checking for success...

```
if (is_array($db_title_list['grx_error'])) {
    $error_string = $db_title_list['grx_error']['grx_error_string'];
    $error_function = $db_title_list['grx_error']['grx_error_function'];
    $error_detail = $db_title_list['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

Miscellaneous Functions

grlResolveArticle

Gold Rush Public Equivalent: Article Finder

Description:

grlResolveArticle accepts an institution code and an array of metadata about an article. It returns an array with the same results you would get from the standard Gold Rush Linker web interface.

Input Parameters

```
array (
  'GR_INST_CODE',
  array (
    'atitle' => ARTICLE_TITLE,
    'title' => JOURNAL_TITLE,
    'issn' => ISSN,
    'volume' => VOLUME,
    'issue' => ISSUE,
    'spage' => SPAGE,
    'date' => DATE,
    'aulast' => AULAST,
    'aufirst' => AUFIRST
  )
)
```

Example Input:

```
array (
  '001_AUR',
  array (
    'atitle' => 'I Want your job, Lady!',
    'title' => 'Time',
    'issn' => '0040-781X',
    'volume' => '161',
    'issue' => '19',
    'spage' => '52',
    'date' => '2003',
    'aulast' => 'Cullen',
    'aufirst' => 'Lisa'
  )
)
```

The above example would request an article resolution for the article described in the input array from the 001_AUR (Auraria Library) holdings.

Return Array

```
Array
(
  [citation] => Array
  (
    [atitle] => I want your job, Lady!
    [title] => Time
    [issn] => 0040-781X
    [volume] => 161
    [issue] => 19
    [pages] => 52
    [date] => 2003
    [aulast] => Cullen
    [aufirst] => Lisa
  )
  [article_access] => Array
  (
    [0] => Array
    (
      [link_url] => http://0-vnweb.hwwilsonweb.com.skyline.cudenver.edu/hww/jumpstart.jhtml?&sid=HWW:BRD&genre=article&issn=0040-781X&volume=161&issue=19&spage=52&pages=52&date=2003&atitle=I%20want%20your%20job%2c%20Lady%26#33%3b&title=Time&aulast=Cullen&aufirst=Lisa
      [link_label] => WilsonWeb
    )
    [1] => Array
    (
      [link_url] => http://0-openurl.ebscohost.com.skyline.cudenver.edu/linksvlc/linking.aspx?sid=GoldRush:Linker&genre=article&issn=0040-781X&volume=161&issue=19&spage=52&pages=52&date=2003&atitle=I%20want%20your%20job%2c%20Lady%26#33%3b&title=Time&aulast=Cullen&aufirst=Lisa
      [link_label] => EBSCOhost
    )
    [2] => Array
    (
      [link_url] => http://0-infotrac.galegroup.com.skyline.cudenver.edu/itw/infomark/0/1/1/purl=rc3_ITOF_0_ke_SN+0040-781X+AND+VO+161+AND+IU+19+AND+SP+52?sw_aep=xxxxxxxxx
      [link_label] => Gale
    )
  )
  [journal_access] => Array
  (
    [0] => Array
    (
      [resource_url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?db=auh&jid=%22TIM%22&scope=site
      [resource_db] => Academic%20Abstracts%20FullTEXT%20Ultra
      [resource_provider] => EBSCO%20INDUSTRIES
      [resource_startdate] => 1990-05-07
      [resource_enddate] => current
      [resource_datnote] =>
    )
    [1] => Array
    (
      [resource_url] => http://0-search.epnet.com.skyline.cudenver.edu/direct.asp?db=afh&jid=%22TIM%22&scope=site
      [resource_db] => Academic%20Search%20Elite
      [resource_provider] => EBSCO%20INDUSTRIES
    )
  )
)
```



```

[resource_startdate] => 1990-05-07
[resource_enddate] => current
[resource_datenote] =>
)
[2] => Array
(
[resource_url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
direct.asp?db=aph&jid=%22TIM%22&scope=site
[resource_db] => Academic%20Search%20Premier
[resource_provider] => EBSCO%20INDUSTRIES
[resource_startdate] => 1990-05-01
[resource_enddate] => current
[resource_datenote] =>
)
[3] => Array
(
[resource_url] => http://0-search.epnet.com.skyline.cudenver.edu/
[resource_db] => Advanced%20Placement%20Source
[resource_provider] => EBSCO%20INDUSTRIES
[resource_startdate] => 1990-05-07
[resource_enddate] => current
[resource_datenote] =>
)
[4] => Array
(
[resource_url] => http://0-vnweb.hwwilsonweb.com.skyline.cudenver.edu/hww/Journals/searchAction.jhtml?sid=HWW:BRD&issn=0040-781X
[resource_db] => Book%20Review%20Digest%20Plus
[resource_provider] => WILSON
[resource_startdate] => 1995-01-01
[resource_enddate] => current
[resource_datenote] =>
)
[5] => Array
(
[resource_url] => http://infotrac.galegroup.com/itweb/auraria_main?
db=BCRC
[resource_db] => Business%20&%20Company%20Resource%20Center
[resource_provider] => GALE%20GROUP
[resource_startdate] => 1983-01-01
[resource_enddate] => current
[resource_datenote] =>
)
[6] => Array
(
[resource_url] => http://0-
infotrac.galegroup.com.skyline.cudenver.edu/
itw/infomark/1/1/1/purl=rc18%5fBIM%5F0%5F%5Fjn+%22Time%22?
sw_aep=xxxxxxxx
[resource_db] => Business%20ASAP
[resource_provider] => GALE%20GROUP
[resource_startdate] => 1983-01-03
[resource_enddate] => current
[resource_datenote] =>
)
[7] => Array
(
[resource_url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
direct.asp?db=buh&jid=%22TIM%22&scope=site
[resource_db] => Business%20Source%20Premier
[resource_provider] => EBSCO%20INDUSTRIES
[resource_startdate] => 1990-05-01
[resource_enddate] => current
[resource_datenote] =>
)

```



```

)
[14] => Array
(
  [resource_url] => http://0-search.ebscohost.com.skyline.cudenver.edu/
  direct.asp?db=mih&jid=%22TIM%22&scope=site
  [resource_db] => Middle%20Search%20Plus
  [resource_provider] => EBSCO%20INDUSTRIES
  [resource_startdate] => 1990-05-01
  [resource_enddate] => current
  [resource_daternote] =>
)
[15] => Array
(
  [resource_url] => http://0-vnweb.hwwilsonweb.com.skyline.cudenver.edu/hww/Journals/searchAction.jhtml?sid=HWW:RDG&issn=0040-781X
  [resource_db] => Readers'%20Guide%20Full%20Text%20Select%20-%20Full%20Text%20Only
  [resource_provider] => WILSON
  [resource_startdate] => 1995-01-01
  [resource_enddate] => current
  [resource_daternote] =>
)
)
[local_access] => Array
(
  [0] => Array
  (
    [resource_url] => http://132.194.32.30/search/i?0040-781X
    [resource_db] => Auraria%20Local%20Library%20Journals
    [resource_provider] => AURARIA%20LIBRARY
    [resource_startdate] =>
    [resource_enddate] =>
    [resource_daternote] =>
  )
)
[indexing_access] => Array
(
  [0] => Array
  (
    [resource_db] => Medline
    [resource_provider] => OCLC
    [resource_url] => http://0-firstsearch.oclc.org.skyline.cudenver.edu/dbname=MEDLINE;autho=xxx-xxx-xxx;FSIP
    [resource_startdate] => 1973-01-01
    [resource_enddate] => current
    [resource_daternote] =>
  )
)
[custom_sections] => Array
(
  [0] => Array
  (
    [section_id] => 1
    [section_name] => Other
    [target] => Array
    (
      [0] => Array
      (
        [link_url] => http://www.google.com/search?q=0040-781X
        [link_label] => Google
      )
      [1] => Array
      (
        [link_url] => http://search.yahoo.com/search?p=0040-781X
        [link_label] => Yahoo
      )
    )
  )
)

```

```

    )
    [2] => Array
    (
      [link_url] => http://search.yahoo.com/search?p=Lisa%20Cullen
      [link_label] => Yahoo
    )
    [3] => Array
    (
      [link_url] => http://www.amazon.com/exec/obidos/ASIN/0040-781X
      [link_label] => Amazon.com
    )
  )
)
[1] => Array
(
  [section_id] => 57
  [section_name] => ILL
)
[2] => Array
(
  [section_id] => 56
  [section_name] => Catalogs
  [target] => Array
  (
    [0] => Array
    (
      [link_url] => http://132.194.32.30/search/i?SEARCH=0040-781X
      [link_label] => Skyline
    )
    [1] => Array
    (
      [link_url] => http://132.194.32.30/search/a?SEARCH=Cullen,%20Lisa
      [link_label] => Skyline
    )
    [2] => Array
    (
      [link_url] => http://132.194.32.30/
      [link_label] => Skyline
    )
    [3] => Array
    (
      [link_url] => http://prospector.coalliance.org/search/i?SEARCH=0040-781X
      [link_label] => Prospector
    )
    [4] => Array
    (
      [link_url] => http://prospector.coalliance.org/search/a?SEARCH=Cullen,%20Lisa
      [link_label] => Prospector
    )
    [5] => Array
    (
      [link_url] => http://prospector.coalliance.org/
      [link_label] => Prospector
    )
    [6] => Array
    (
      [link_url] => http://scholar.google.com/scholar?q=0040-781X&ie=UTF-8&oe=UTF-8&hl=en&btnG=Search
      [link_label] => Google Scholar
    )
    [7] => Array
  )
)

```

```

        (
          [link_url] => http://scholar.google.com/scholar?
          q=Cullen,%20Lisa&ie=UTF-8&oe=UTF-8&hl=en&btnG=Search
          [link_label] => Google Scholar
        )
      [8] => Array
      (
        [link_url] => http://scholar.google.com
        [link_label] => Google Scholar
      )
    )
  )
)
[proctime] => 1 second(s)
[grx_error] =>
)

```

The array returned by `grlResolveArticle` holds a number of sub-arrays representing the different sections of a normal GR Linker results screen. The “citation” sub-array holds the article's citation that would normally be displayed at the top of GR Linker results. It includes elements for standard OpenURL citations.

The “article_access” sub-array holds a numbered list of article links. For each link, it has elements for the `link_url` and the `link_label`. If you have configured Gold Rush to wrap URLs with your proxy server information, all URLs will reflect that setting and include the proxied URL.

The “journal_access” sub-array holds a numbered list of links to full text resources. The record for each resource includes elements for the `resource_url`, `resource_db`, `resource_provider`, `resource_startdate`, `resource_enddate`, and `resource_datanote`. Even if you have GR Linker configured to hide some of these elements in its normal display, they will all be returned through GRX. You can decide which elements you want to display after retrieving the results.

The “local_access” sub-array holds a numbered list of links to your local full text holdings. It includes the same elements as the “journal_access” sub-array.

The “indexing_access” sub-array holds a numbered list of links to resources that do not have any full text, but do have indexing for the article being resolved. It includes the same elements as the “journal_access” sub-array.

The “custom_sections” sub-array holds a numbered list of the custom sections that you have defined in the Gold Rush Staff Toolbox -> Settings -> Manage Gold Rush Linker -> Display Settings. For each of your custom sections, there will be a sub-array holding the `section_id`, `section_name`, and yet another sub-array named “target” with a numbered list of targets that you have in that custom section. If you have defined any sections that do not have any targets associated with them, the “target” sub-array will not be present for that section.

Finally, there is one element in the top-level array named “proctime”. This will tell you approximately how long it took GR Linker to complete the article resolution.

And, once again, we hope the “grx_error” sub-array is empty, indicating a successful article resolution.

Client Code

Calling the function...

```
$inst_code = '001_AUR';
$metadata = array();
$metadata['atitle'] = 'I want your job, lady!';
$metadata['title'] = 'Time';
$metadata['issn'] = '0040-781X';
$metadata['volume'] = '161';
$metadata['issue'] = '19';
$metadata['spage'] = '52';
$metadata['date'] = '2003';
$metadata['aulast'] = 'Cullen';
$metadata['aufirst'] = 'Lisa';

$grl_results = $grx_client->call(
    'grlResolveArticle',
    array($inst_code,$metadata)
);
```

Checking for success...

```
if (is_array($grl_results['grx_error'])) {
    $error_string = $grl_results['grx_error']['grx_error_string'];
    $error_function = $grl_results['grx_error']['grx_error_function'];
    $error_detail = $grl_results['grx_error']['grx_error_detail'];
    print "<h1>Error: $error_string</h1>";
    print "<p>Function where error occurred: $error_function</p>";
    print "<p>Error Detail: $error_detail</p>";
}
else {
    // process the returned array
}
```

holdingsExport

Description:

holdingsExport accepts an institution code and a Gold Rush Staff username and password. If the user passes authentication, it will export your holdings to an XML file. It compresses the file and returns an array with information about the export, including the url you can use to retrieve the file.

This function can take over a minute to return your results. Because of this, you may need to increase the timeout for your client before calling this function.

If you have 50,000 holdings records, this function should take about 30 seconds. If you have 150,000 holdings records, it should take about 1 minute. If you have 250,000 holdings records, it should take about 1.5 minutes.

Input Parameters

```
array (  
  'GR_INST_CODE',  
  'USERNAME',  
  'PASSWORD'  
)
```

Example Input:

```
array (  
  '001_ARO',  
  'scott',  
  'scottspass'  
)
```

The above example would request an export of holdings for 001_ARO (Aurora Public Library).

Return Array

```
Array  
(  
  [status] => Export Complete  
  [record_count] => 55419 of 55419  
  [url] => http://grx.coalliance.org/grx_fetch_export.php?  
  f=001_ARO_holdings_2008-02-28.xml.gz  
)
```

The array returned by `holdingsExport` is a simple, one-level array, with elements for `status`, `record_count`, and `url`. The “status” element will tell you if there was an error. The values for status can be:

“Export Complete” indicating a successful export.

“Incorrect input parameters” indicating that the parameters passed to the function were not formatted correctly or were incomplete.

“User is not authorized to export” indicating that the user passed to the function is not authorized to export holdings through GRX.

“Could not query database for holdings” indicating that an internal error occurred which kept GRX from getting the data from your holdings table.

“Couldn't create holdings file” indicating that an internal error occurred which kept GRX from creating the export file.

The “record_count” element tells you how many records were found in your holdings and how many were exported. If the two numbers do not match, it usually indicates that you have some bad data in your holdings table.

The “url” element holds the URL you can use to download the completed export file. The file will be an XML file that has been gzipped. Your file will be available for pickup for seven days. After that, it will be automatically deleted, and you will have to run your export again.

Client Code

Calling the function...

```
$user_info = array();
$user_info['inst_code'] = '001_ARO';
$user_info['gr_username'] = 'scott';
$user_info['gr_password'] = 'scottspass';
$export_info = $grx_client->call(
    'holdingsExport',
    array($user_info)
);
```

Checking for success...

```
if (array_key_exists('status',$export_info)) {
    if ($export_info['status'] == 'Export Complete') {
        // No Errors
    }
    else {
        // There was an error
        $error = $export_info['status'];
        print "$error";
    }
}
```


How To...

How to install nuSOAP

The NuSOAP project is hosted by SourceForge. You can download the current version of NuSOAP from <http://sourceforge.net/projects/nusoap/>.

Once you have downloaded a copy of nusoap.php, you just have to unzip it into a directory that is accessible to your webserver. Depending on your PHP configuration, you may already have a directory that you use for included files. You can also put the NuSOAP directory in the same directory where your GRX client will reside.

Just be sure that you can include files from the NuSOAP directory. If you do not put NuSOAP in the same directory as your client, you may need to edit your PHP ini file and add the NuSOAP directory to the include_path variable.

You should also make sure that the ownership and permissions of the NuSOAP directory match the ownership and permissions of other PHP files in your webspace.

How to install Perl SOAP modules

You can download all perl modules from cpan at <http://www.cpan.org/>.

There are two ways to install perl modules. Using either method, you will want to be the root user while installing perl modules. So, begin by su'ing to root.

If your perl installation has support for the CPAN module, you can simply type “cpan modulename” at a system prompt.

If your perl installation does not have support for the CPAN module, you can manually download the files from <http://www.cpan.org/>.

After you have downloaded a perl module, unzip and untar it into an out-of-the-way directory. This directory should NOT be in your webspace. It will only be used during the installation process and can be deleted afterward.

Change into the directory for the perl module, type “perl Makefile.PL” and hit return. When that command has completed, enter “make” and hit return. When that command completes, type “make test” and perl will test the module to make sure that it is behaving properly. Once you have passed all appropriate tests, enter “make install” and perl will complete the installation of the module.

Here's a sample session, installing the perl module SOAP::Lite.

```
# tar xzvf SOAP-Lite-0.69.tar.gz
# cd SOAP-Lite-0.6.9
# perl Makefile.PL
# make
# make test
# make install
```

During the “perl Makefile.PL” step, if perl tells you that you are missing some required modules, make a note of the missing modules, and install them before returning to this directory and completing the installation.

The perl modules you may want to consider are:

Prerequisites for SOAP::Lite

- XML::Parser
- MIME::Base64
- URI
- libwww-perl

Optional prerequisites (these modules support additional functionality in SOAP::Lite)

- Crypt::SSLeay
- IO::Socket::SSL
- Apache
- Net::POP3
- MIME::Lite
- IO::File
- Net::FTP
- MQSeries
- Net::Jabber
- FCGI
- MIME::Tools
- DIME::Tools
- Compress::Zlib

The SOAP Lite modules

- SOAP::Lite
- SOAP::Lite::Utility

How to create a GRX client in PHP with NuSOAP

Creating a client for GRX in PHP with NuSOAP is a six-step process.

1. Include the nusoap.php file.
2. Create a SOAP client object.
3. Set parameters for the function you want to call.
4. Call GRX.
5. Check for errors.
6. Process the results

Once you have created the SOAP client object in a PHP script, you can use the script to call one specific function from GRX, or you can use the script to call any of the GRX functions based on input from the user.

In the following example, we'll create a GRX client that calls the GRX function “searchByJrnlTitle” with a search term of “ceramic” without limiting to Full Text or Free resources. It will perform the search against the holdings of institution 001_AUR (Auraria Library).

```
1. <?php
2.
3. // Include the nusoap.php file
4. // (this path assumes NuSOAP directory is in same directory as script)
5. require "nusoap.php"
6.
7. // Create the SOAP client object pointing to grx.
8. $grx_client = new soapclient('http://grx.coalliance.org/grx.php');
9.
10.// Set up parameters for the call
11.$inst_code = '001_AUR';
12.$searchterm = 'ceramic';
13.
14.// Make the call to GRX
15.$search_results = $grx_client->call('searchByJrnlTitle',
    array($inst_code,$searchterm));
16.
17.// Check for errors
18.if (is_array($search_results['grx_error'])) {
19.// There is an error. Set some variables and display error.
20. $error_string = $search_results['grx_error']['grx_error_string'];
21. $error_function = $search_results['grx_error']['grx_error_function'];
22. $error_detail = $search_results['grx_error']['grx_error_detail'];
23. print "<h1>Error: $error_string</h1>";
24. print "<p>Function where error occurred: $error_function</p>";
25. print "<p>Error Detail: $error_detail</p>";
26.}
27.else {
28.// There are no errors, process the returned array and output in some way.
29.print_r($search_results);
30.}
31.
32.??>
```

If we want to alter this script so that it limits the search to full text, we would make the following changes to the parameters and the call.

```
// Set up parameters for the call
$inst_code = '001_AUR';
$searchterm = 'ceramic';
$limit_ft = 'Y';

// Make the call to GRX
$search_results = $grx_client->call('searchByJrnlTitle',
array($inst_code,$searchterm,$limit_ft));
```

We just added a line to set a variable for our full text limit. Then, in the call to GRX, we add the `$limit_ft` parameter.

How to display GRX results on a web page

In the above example, we didn't really do any processing on the returned array. We just printed it out using the “`print_r`” PHP function. It is more likely that you will want to display the results in a browser.

GRX results are always returned in an array, so you will have to loop over the array if you want to display the results as HTML. The GRX search functions complicate things a little bit because they might return a list of journals, or a full record for a single journal.

So, for search results, you will want to check to see if there is a single hit or a list of hits. Either way, we'll be drilling down through a multi-level array to get at the data we want.

Building on the previous example, we'll be replacing line 29, “`print_r($search_results)`” with several lines that loop over the array and format the results for an HTML table.

Here's the new code...

```
// Check for an embedded full record
if (is_array($search_results['found_resource'])) {
// There is a full record
// Start a table to hold the results
    print "<table border=\"1\">\n";
    print "<tr><th>Title ID</th><th>Title</th><th>ISSN</th></tr>\n";
// Loop over the results, drilling down to the first level
    foreach ($search_results as $rec => $data) {
        if ($rec == 'found_resource') {
// Now drill down to the next level, where our data is.
            foreach ($data as $element => $element_data) {
                $found_rec_id = $element_data['title_id'];
                $found_rec_title = $element_data['jtitle'];
                $found_rec_issn = $element_data['issn'];
// Print the table row with our data.
                print
"<tr><td>$found_rec_id</td><td>$found_rec_title</td><td>$found_rec_issn</td></tr>\n";
            } // end of foreach ($data as $element => $element_data)
        } // end of if ($rec == 'found_resource')
```

```

        } // end of foreach ($search_results as $rec => $data)
        print "</table>\n";
    } // end of if (is_array($search_results['found_resource']))
    else {
// There is no full record, just a list of brief records
// Start a table
        print "<table border='1'\n";
        print "<tr><th>Record
Number</th><th>ID</th><th>Title</th><th>ISSN</th></tr>\n";
// Loop over the first level of the array.
        foreach ($search_results as $rec => $data) {
            if ($rec == 'brief_recs') {
// Loop over the second level array in $search_results['brief_recs']
                foreach ($data as $brfrec_num => $brfrec_data) {
                    $brf_rec_id = $brfrec_data['id'];
                    $brf_rec_title = $brfrec_data['jtitle'];
                    $brf_rec_issn = $brfrec_data['issn'];
// Print the table row with our data.
                    print
"<tr><td>$brfrec_num</td><td>$brf_rec_id</td><td>$brf_rec_title</td><td>$brf_re
c_issn</td></tr>\n";
                } // end of foreach ($data as $brfrec_num => $brfrec_data)
            } // end of if ($rec == 'brief_recs')
        } // end of foreach ($search_results as $rec => $data)
        print "</table>\n";
    } // end of else
} // end of else

```

How to Increase the Timeout on Your GRX Client

A few of the GRX functions take some time to complete – holdingsExport, journalListA2Z, and some browses. For these functions, you may need to increase the timeout on your GRX client.

Using our example client above, we would need to add a couple of lines after creating the client on line 8.

```
$grx_client = new soapclient('http://grx.coalliance.org/grx.php');
```

Add these lines to extend the timeout.

```
$grx_client->timeout = 1200;
$grx_client->response_timeout = 1200;
```

The timeout values are in seconds, so 1200 seconds is 20 minutes. No GRX function should take that long to complete, so this is a pretty safe number to use in your client.

Sample PHP Client

```
<?php

// Include the nusoap.php file
// (this path assumes NuSOAP directory is in same directory as script)
require "nusoap.php"

// Create the SOAP client object pointing to grx.
$grx_client = new soapclient('http://grx.coalliance.org/grx.php');
$grx_client->timeout = 1200;
$grx_client->response_timeout = 1200;
// Set up parameters for the call
$inst_code = '001_AUR';
$searchterm = 'ceramic';
$limit_ft = 'Y';

// Make the call to GRX
$search_results = $grx_client->call('searchByJrnlTitle',
array($inst_code,$searchterm,$limit_ft));

// Check for errors
if (is_array($search_results['grx_error'])) {
// There is an error. Set some variables and display error.
$error_string = $search_results['grx_error']['grx_error_string'];
$error_function = $search_results['grx_error']['grx_error_function'];
$error_detail = $search_results['grx_error']['grx_error_detail'];
print "<h1>Error: $error_string</h1>";
print "<p>Function where error occurred: $error_function</p>";
print "<p>Error Detail: $error_detail</p>";
}
else {
// There are no errors, process the returned array and output in some way.
// Check for an embedded full record
if (is_array($search_results['found_resource'])) {
// There is a full record
// Start a table to hold the results
print "<table border='1'\>\n";
print "<tr><th>Title ID</th><th>Title</th><th>ISSN</th></tr>\n";
// Loop over the results, drilling down to the first level
foreach ($search_results as $rec => $data) {
if ($rec == 'found_resource') {
// Now drill down to the next level, where our data is.
foreach ($data as $element => $element_data) {
$found_rec_id = $element_data['title_id'];
$found_rec_title = $element_data['jtitle'];
$found_rec_issn = $element_data['issn'];
// Print the table row with our data.
print
"<tr><td>$found_rec_id</td><td>$found_rec_title</td><td>$found_rec_issn</td></tr>\n";
} // end of foreach ($data as $element => $element_data)
} // end of if ($rec == 'found_resource')
} // end of foreach ($search_results as $rec => $data)
print "</table>\n";
} // end of if (is_array($search_results['found_resource']))
else {
// There is no full record, just a list of brief records
// Start a table
print "<table border='1'\>\n";
print "<tr><th>Record
Number</th><th>ID</th><th>Title</th><th>ISSN</th></tr>\n";
// Loop over the first level of the array.
```

```

    foreach ($search_results as $rec => $data) {
        if ($rec == 'brief_recs') {
// Loop over the second level array in $search_results['brief_recs']
            foreach ($data as $brfrec_num => $brfrec_data) {
                $brf_rec_id = $brfrec_data['id'];
                $brf_rec_title = $brfrec_data['title'];
                $brf_rec_issn = $brfrec_data['issn'];
// Print the table row with our data.
                print
                "<tr><td>$brfrec_num</td><td>$brf_rec_id</td><td>$brf_rec_title</td><td>$brf_rec_issn
                </td></tr>\n";
            } // end of foreach ($data as $brfrec_num => $brfrec_data)
        } // end of if ($rec == 'brief_recs')
    } // end of foreach ($search_results as $rec => $data)
    print "</table>\n";
} // end of else
} // end of else
}
?>

```