connectMLS : connectMLS RETS Developer's Start Guide

The Purpose of This Guide

This guide is intended as an accessible starting point for the first-time RETS (Real Estate Transaction Standard) developer. It provides a basic overview of what RETS is and what a RETS server does, then describes some useful techniques for beginning to work with the connectMLS RETS server.

More advanced RETS users, or those who prefer to begin with a more detailed technical approach, may want to proceed directly to the RETS website; detailed links are provided in Additional Resources, below.

What is RETS?

According to the overview at www.rets.org:

> The Real Estate Transaction Standard is a common language spoken by systems that handle real estate information, such as multiple listing services. A common language enables computers like the one on your desk to receive information from many different real estate systems or MLSs without being specially "trained" to understand the information from each.

connectMLS RETS Transaction Types

There are several transaction types that RETS servers may provide, some optional and some required.   The key transaction types for client programs that need to download MLS data using RETS are **Login**, **GetMetadata**, **Search**, **GetObject**, and **Logout**, all of which are provided by the connectMLS server. In this section we give brief descriptions of each of these transaction types. For examples of how to issue requests for each transaction type, see Getting Started.

Login

Before downloading MLS data, the client user or program must log into the RETS server, i.e., provide a valid user ID and password. This is done by means of a Login transaction.

GetMetadata

Although all MLS providers carry listing information, there is no specific standard for *how* it is carried. For example, the particular listing details that are available may vary from one provider to another, as may the field names under which it is stored. How then is a RETS client to determine what data is available to be queried, or how to construct the proper queries?

The answer is in the RETS *metadata*, or "data about the data". This data provides a structured description of the information available from a given RETS server so that the client can determine how to ask that particular server for the information it needs. For more specific information about the connectMLS metadata, see Basics of Using connectMLS Metadata, below.

The purpose of the GetMetadata transaction is, of course, to enable the client to retrieve this metadata from the RETS server.

Search

The Search transaction is the primary transaction type for retrieving MLS data from a RETS server. With a properly formulated Search transaction request, a client may specify a search type (e.g., Agent, Property, etc.), search criteria, data fields to be retrieved, the maximum number of results to return, etc.   A full description of the request options for a Search transaction can be found in the RETS specification document (see Additional Resources, below).

GetObject

The GetObject transaction provides a means of downloading objects associated with a given listing. A very common use for this transaction type is to retrieve listing photos.

Logout

Once a RETS session is complete, the client uses a Logout transaction to sign out of the server.

Getting Started

In this section, we show you how to get started by issuing RETS transaction requests directly from your favorite browser. This will allow you to experiment with the various transaction types and get a feel for how a RETS client interacts with a RETS server.

Sample Login Transaction

To issue a Login transaction request, type the following into your browser, substituting the URL of your connectMLS server for <connectMLS.rets.server.url>:

```
http://<connectMLS.rets.server.url>/rets/server/login
```

You will then be prompted for a login id and password. Once you have successfully logged in, you will see an XML response similar to the following:

```
<RETS ReplyCode="0" ReplyText="">
   <RETS-RESPONSE>
       MemberName=My Name
       User=mylogin,,,mylogin
       Broker=BKNAME
       MetadataVersion=1
       MinMetadataVersion=1
       TimeoutSeconds=1800
       Action=/rets/server/action
       GetObject=/rets/server/getobject
       Login=/rets/server/login
       Logout=/rets/server/logout
       Search=/rets/server/search
       GetMetadata=/rets/server/getmetadata
       ServerInformation=/rets/server/serverinformation
   </RETS-RESPONSE>
</RETS>
```

Notice that this response includes locator information for the other available actions (GetObject, Logout, Search, GetMetadata, ServerInformation) available on your RETS server.

Sample Metadata Transactions

 The following sample GetMetadata transaction will return a list of the Resources (search types) available from your RETS server.

```
http://<connectMLS.rets.server.url>/rets/server/getmetadata?Type=METADATA-
RESOURCE&Format=COMPACT&ID=0
```

If you are familiar with HTTP syntax, you will see that this transaction request passes a list of parameters and values as follows:

- Type=METADATA-RESOURCE
- Format=COMPACT
- ID=0

We now describe each of these parameters in greater detail. The parameters marked with a '*' are required for all GetMetadata transactions.

*Type: The type of metadata to be returned. A complete list of legal values for this parameter can be found in the RETS specification document, but the most useful values for most connectMLS RETS users will be:

- METADATA-SYSTEM -- General system information
- METADATA-RESOURCE -- A list of the Resources (search types) available on the server
- METADATA-CLASS -- A list of the Classes (search subtypes) available on the server, organized by Resource
- METADATA-TABLE -- A list of the data fields available on the server, organized by Resource and Class
- METADATA-OBJECT -- A list of the object types (e.g., photographs) available on the server, organized by Resource
- METADATA-LOOKUP -- A list of the lookup tables (used for drop-down lists) available on the server, organized by Resource.
- METADATA-LOOKUP_VALUES -- A list of the codes and values in each lookup table, organized by Resource and lookup table name.

Format: The XML format in which the data should be returned. There are three format types defined in the RETS specification: COMPACT, in which data are arranged in a simple tabular form; COMPACT-DECODED, which is similar except that encoded values are expanded to human-readable format using the appropriate lookup tables; and STANDARD-XML, a less concise format with a more detailed XML structure. The STANDARD-XML format is incompletely supported by connectMLS at present, but will be offered in the future.

*ID: A parameter to restrict the metadata returned, e.g., to return the available data fields for Agent search types only. For details on how to use this parameter, consult the RETS Specification document. A value of '0' means no restriction; all the available metadata of the specified type is returned.

In Basics of Using connectMLS Metadata below, we present some sample metadata and show you how to get started interpreting it. You are encouraged to experiment with variations on the query above to get a feel for what the different types of metadata look like.

Sample Search Transaction

The following sample Search transaction will return listing numbers and prices for residential properties whose price is greater than $300,000.   (Note that this query assumes that "ResidentialProperty" is one of the data classes available for the "Property" search type; if this is not true for your connectMLS RETS server, you can run this transaction by substituting a valid class. Available classes for each search type can be found in the metadata.)

```
http://<connectMLS.rets.server.url>/rets/server/search?SearchType=Property
&Class=ResidentialProperty&QueryType=DMQL2&Format=COMPACT&StandardNames=1
&Select=ListingID,ListPrice&Query=(ListPrice=300000%2B)&Count=1&Limit=10
```

(Note that in non-browser-based RETS requests, the above query parameter would be Query=(ListPrice=300000+). However, if you enter the query directly into a browser address, you need to use the ascii code "%2B" instead of "+".)

This transaction request passes the following list of parameters and values:

- SearchType=Property
- Class=ResidentialProperty
- QueryType=DMQL2
- Format=COMPACT
- StandardNames=1
- Select=ListingID,ListPrice
- Query=(ListPrice=300000+)
- Count=1
- Limit=10

We now describe each of these parameters in greater detail. The parameters marked with a '*' are required for all Search transactions.

*SearchType: The type of MLS data that the search is to return. There are nine standard search types (called "Resources") defined in the RETS specification; every RETS server is required to support at least one of these, and to report all supported resources in its metadata. The 'Property' resource, which encompasses the actual listing data, is provided by most if not all RETS servers.

*Class: The subtype of the data to be returned. Every resource has at least one class associated with it, while some may have several; e.g., the Property resource may include separate classes for single-family homes, condominiums, rental units, etc. Classes are not specified in the RETS standard, and may vary widely from one RETS provider to another. They are, however, required to be listed in the metadata.

*QueryType: The query language in which the **Query** parameter is expressed. This parameter exists as a placeholder in case multiple query languages become supported in later expansions of RETS; at present, however, its value is always 'DMQL2'.

Format: The XML format in which the data should be returned. There are three format types defined in the RETS specification: COMPACT, in which data are arranged in a simple tabular form; COMPACT-DECODED, which is similar except that encoded values are expanded to human-readable format using the appropriate lookup tables; and STANDARD-XML, a less concise format with a more detailed XML structure. The STANDARD-XML format is incompletely supported by connectMLS at present, but will be offered in the future.

StandardNames: This parameter may have a value of either 0 or 1. If a value of 1 is specified, then the RETS server will interpret the field selection list and query as being expressed in terms of *standard names*, which should be the same from one RETS server to the next. If the value is 0, or if the parameter is not specified, then the server assumes that local, server-specific system names are being used instead. (For more details on standard vs. system names, see Basics of Using connectMLS Metadata.)

Select: A comma-separated list of the fields whose values should be returned in the search results. If this parameter is not specified, all available fields for the specified resource and class are returned.

*Query: The actual search query for the transaction. A detailed specification of the DMQL2 query language is outside the scope of this document, but may be found in the RETS specification document (see Additional Resources).

Count: The count type for the search; this parameter may have a value of 0, 1 or 2. If a value of 1 is specified, then the total number of records in the database that match the search criteria is returned along with the results. If 2 is specified, then the count is returned instead of the list of results. If 0 is specified, or if the Count parameter is omitted, the results are returned without the count.

Limit: The maximum number of results to be returned for the query. Please note that, with a few exceptions, the connectMLS system imposes a hard limit of 2000 results for any given query, even if the Limit parameter is set higher.

Offset: The number of the matching record to start the return set with. E.g., if you specify an offset of 1001, a query with no specified limit would return the 1001st - 3000th matching records -- or the 1001st through last, if there are not 3000 matching records. If the Limit parameter were set to 10, the search

would return the 1001st through 1010th records. (For more information, see Question 3 in Frequently Asked Questions below.)

There are a number of other optional parameters to the Search transaction request; these are documented in detail in the RETS specification document.

The results of the sample query will look similar to the following:

```
<RETS ReplyCode="0" ReplyText="">
<COUNT Records="2117">
<DELIMITER value="09"/>
  <COLUMNS>    ListingID   ListPrice </COLUMNS>
  <DATA>    12345678    300000 </DATA>
  <DATA>    23456789    300000 </DATA>
  <DATA>    34567890    300000 </DATA>
  <DATA>    45678901    300000 </DATA>
  <DATA>    56789012    310000 </DATA>
  <DATA>    67890123    310000 </DATA>
  <DATA>    78901234    310000 </DATA>
  <DATA>    89012345    300000 </DATA>
  <DATA>    90123456    350000 </DATA>
  <DATA>    01234567    400000 </DATA>
</RETS>
```

Sample GetObject Transaction

The following sample GetObject transaction will download a photograph for a listing:

```
http://<connectMLS.rets.server.url>/rets/server/getObject?
Type=Photo&Resource=Property&Id=06541130:0
```

This transaction request passes the following list of parameters and values:

- Type=Photo
- Resource=Property
- Id=06541130:0

We now describe each of these parameters in greater detail. All three parameters are required for all GetObject transactions.

*Type: The type of object to be retrieved. The available types for your RETS server can be found in the metadata (query for the METADATA-OBJECT metadata type).

Resource: The Resource with which the object is associated, e.g., a property photo would be associated with the "Property" resource, while an agent headshot might be associated with the "Agent" resource. Available resources for your RETS server are listed in the metadata (query for METADATA-RESOURCE).

*Id: The identifier for the record with which the object is associated. For property listings, this would be the listing Id. The ":0" at the end of the sample parameter indicates that the "preferred" photograph should be returned if more than one photograph is available. Other options are:

- ":n" to return the nth photograph (e.g., ":2" would return the second)
- ":*" to return all photographs

Basics of Using connectMLS Metadata

In Getting Started above, we offered the following sample metadata query:

```
http://<connectMLS.rets.server.url>/rets/server/getmetadata?Type=METADATA-
RESOURCE&Format=COMPACT&ID=0
```

Suppose that you submitted this query and got the following XML results back:

```
<RETS ReplyCode="0" ReplyText="Operation Successful">
<METADATA-RESOURCE Version="1.0.0" Date="2007-12-04T16:24:37">

<COLUMNS>
    ResourceID    StandardName    VisibleName    Description  KeyField    ClassCount
  ClassVersion    ClassDate    ObjectVersion
ObjectDate    LookupVersion    LookupDate
</COLUMNS>

<DATA>
    Agent    Agent    Agent    All Agents    USERID    1    1.0.0  2008-01-18T12:00:00    1.0.0
  2008-01-18T12:00:00    1.0.0
 2008-01-18T12:00:00
</DATA>

<DATA>
    Office    Office    Office    List of Real Estate Offices  OFFICEID    1    1.0.0
  2008-01-18T12:00:00    1.0.0
 2008-01-18T12:00:00    1.0.0    2008-01-18T12:00:00
</DATA>

<DATA>
    Property    Property    Property    Property    LN    18    1.0.0  2008-01-18T12:00:00
  1.0.0    2008-01-18T12:00:00    1.0.0
 2008-01-18T12:00:00
</DATA>


</METADATA-RESOURCE>
</RETS>
```

The <COLUMNS> entry tells you how to interpret the <DATA> lines that follow. So, for example, the first
element of each <DATA> line is a Resource ID, the fourth is a description of the resource, and the sixth
is the number of classes belonging to the resource. This particular output would indicate that the server
supports three Resources (search types): "Agent", "Office", and "Property".

Now, consider the following (very truncated) sample results from a METADATA-TABLE query:

```
<RETS ReplyCode="0" ReplyText="Operation Successful">
<METADATA-TABLE Resource="Property" Class="ResidentialProperty" Version="1"
 Date="2008-01-18T12:00:00">

<COLUMNS>
    SystemName    StandardName    LongName    DBName    ShortName    MaximumLength    DataType
  Precision    Searchable
    Interpretation    Alignment    UseSeparator    EditMaskID    LookupName    MaxSelect    Units
  Index    Minimum    Maximum
    Default    Required    SearchHelpID    Unique
</COLUMNS>


<DATA>
        SP            Sale Price    SP    Sale Price    12    Int        1
 Currency
```

```
</DATA>

<DATA>
     ST        ListingStatus    Status  ST      Stat    4       Character              0
 Lookup                          ST

<DATA>
     STATE            State    STATE   State   2       Character              1       Lookup
                 STATE
</DATA>
.
.
.
</METADATA-TABLE>
</RETS>
```

At first glance, this example doesn't appear to follow the interpretation guidelines we have already given. After all, there are 23 column names, but none of the data lines seems to have more than 10 entries!

The reason for this is that some of the metadata values for any given field may be null. Because the <DATA> line elements are tab-separated, null values are difficult to detect visually, and column values will usually not line up under the column headers. It is up to each client RETS program to properly interpret the XML, recognizing tabs as separators and that two sequential tabs indicate a null value in the corresponding field.

Once properly interpreted, the metadata above gives information about three fields: "SP" (whose LongName tells you that it represents the sale price), "ST" (listing status), and "STATE" (state). You will also be able to tell that SP is of type Currency, and that ST and STATE are encoded values with associated lookup tables (also called "ST" and "STATE" in this case).

Note that "SP", "ST" and "STATE" are the *System Names* for these fields, and can therefore be used in search transactions where the **StandardNames** parameter is set to 0 or not set at all (see Sample Search Transaction, above). The *Standard Names* for these fields are null in the metadata, which means that these fields cannot be used in a search transaction where the StandardNames parameter is set to 1.

Frequently Asked Questions

In this section we answer some questions that are frequently posed by users new to connectMLS.

1. **It looks like there's something wrong with the dates. What's the problem?**
2. **My search only returns 2000 listings, and I know there are more. What's going on?**
3. **OK, but I really need to see more than 2000 results. Is there any way around the limit?**
4. **If I don't want to use the Offset parameter, is there another way to get a larger set of results?**
5. **I have only one RETS login, and when I try to run two processes, one of them keeps getting kicked out. What do I do?**
6. **How do I download the photograph(s) for a listing?**
7. **OK, but these are medium-resolution photos. What if I need larger ones, or thumbnails?**

1. It looks like there's something wrong with the dates. What's the problem?

In keeping with the RETS specification, all dates are provided in GMT (Greenwich Mean Time).

2. My search only returns 2000 listings, and I know there are more. What's going on?

Because of the size of the listing database, queries that are too general could return unmanageably large result sets if no limit were placed on the number of records. To avoid the resulting strain on shared resources, we have imposed a limit of 2000 results for any given query, which should be sufficient for most purposes.

We have made an exception to this limit for queries that include only fields from the following list:

- Listing ID
- Modification Timestamp
- Photo Count
- Photo Update Date
- User ID
- User Status
- Office ID
- Office Status

The corresponding field names can be found in your metadata. (See Basics of Using connectMLS Metadata, above.)

3. OK, but I really need to see more than 2000 results. Is there any way around the limit?

To download more than 2000 listings, use the Offset parameter described in Sample Search Transaction. Detailed instructions are as follows:

1. First, run a search on your desired query with the Count parameter set to 2, e.g.:

```
<search url>?Class=ResidentialProperty&SearchType=PropertyQueryType=DMQL2&
Format=COMPACT&StandardNames=1&Query=(<your query here>)&Count=2
```

This will return a count of the records that match your query without trying to return the actual record set. If the count is less than 2000, you do not need to use Offset, but can simply run your search.

[NOTE: More advanced users may wish to skip this step, instead looking for the <MAXROWS> tag at the end of a search to determine whether all rows have been sent. More details can be found in the RETS Specification document (see Additional Resources).]

2. Run the desired query as above, but eliminate the Count parameter, instead setting Limit to a value less than or equal to 2000 (we recommend 500, especially if more than a few fields are requested) and Offset to 1:

```
<search url>?Class=ResidentialProperty&SearchType=PropertyQueryType=DMQL2&
Format=COMPACT&StandardNames=1&Query=(<your query here>)&Limit=500&Offset=1
```

The seemingly redundant Offset of 1 is important here, as it signals the server to cache information that will assist it during the remaining steps below.

3.-?. Run the query again with an offset of Limit+1 (501 in the example above) for the next block of records, then (2*Limit)+1 for the next, and so on, until all matching records have been retrieved:

```
<search url>?Class=ResidentialProperty&SearchType=PropertyQueryType=DMQL2&
Format=COMPACT&StandardNames=1&Query=(<your query here>)&Limit=500&Offset=501

<search url>?Class=ResidentialProperty&SearchType=PropertyQueryType=DMQL2&
Format=COMPACT&StandardNames=1&Query=(<your query here>)&Limit=500&Offset=1001
```

IMPORTANT: In order to ensure that a complete set of records is returned without duplicates, the server caches vital information the first time you run the query, as mentioned above in step 2. In order to avoid resetting this cache, it is important to finish all iterations of the query in one session, without running any other queries between them. Furthermore, each iteration should be identical except for the offset value, as shown above. Changing the order of the parameters, for example, will reset the cache.

4. If I don't want to use the Offset parameter, is there another way to get a larger set of results?

The Offset parameter is the simplest and most efficient way to get around the download limit. Unlike some other RETS servers, the connectMLS RETS server is not prone to record duplication and/or incomplete results when this parameter is used as described above; this is because of its cache-based approach to offset queries.

However, if you prefer, you can implement a similar cache strategy yourself as follows: First,perform a query that selects only the listing ID and modification timestamp fields. As noted in FAQ #2 above, there is no restriction on the number of records that may be downloaded for a query of this type. After downloading the listing IDs, issue a second set of queries using these IDs, separated by commas, to retrieve the full listing data.

Detailed Instructions:

1. Issue a RETS query to the RETS server using "ModificationTimestamp >= [Date]" in the argument and specifying only two columns: LN (MLS Number) and ModificationTimestamp. In this case, [Date] is a point in time at which you last queried the RETS server for updates or a point in time from which you want to start over. The RETS server will return the value of these two columns for all records matching the search criteria.

Example query string:

```
<search
 url>?Class=ResidentialProperty&SearchType=Property&QueryType=DMQL2&Format=COMPACT&StandardNames=1&Select=Modif
&Query=(ModificationTimestamp=2008-01-01T00:00:00+)
```

2. After the query has returned, use the ModificationTimestamp to compare the search results to records in your local database to see what has changed. Build a list of records to download represented by the MLS Numbers (see next).

3. Issue a RETS Query to the RETS server using a comma delimited list of 100 LN's (MLS Numbers) at a time, requesting full search result data for those requested records. Repeat this process until all the records are downloaded. (Note: You can use any number up to 2000 but 100 is recommended for optimal performance).

Example query string:

```
<search
 url>?Class=ResidentialProperty&SearchType=Property&QueryType=DMQL2&Format=COMPACT&StandardNames=1&Query=(Listi
 seperated list numbers>)
```

5. I have only one RETS login. When I try to run two processes, one of them keeps getting kicked out.

Any given account can have at most one logged-in session at a time. Multiple processes may use a single account provided that they are not logged in simultaneously. If a second login transaction is attempted while a session is in progress, the existing session will be terminated.

To use the same login for multiple transactions and/or processes, note the following:

- A single logged-in session can issue up to two concurrent transactions.
- A single logged-in session can issue an unlimited number of non-concurrent (i.e., consecutive) transactions.
- Different processes may use different login sessions as long as they don't run at the same time.
- If you require multiple simultaneous access to the RETS Server, please contact your MLS Board for more accounts.

6. How do I download the photograph(s) for a listing?

See <u>Sample GetObject Transaction</u> above.

7. OK, but these are medium-resolution photos. What if I need larger ones, or thumbnails?

For a thumbnail version of a photo, use the query described above, but substitute "Type=ThNail" for "Type=Photo".

Likewise, for a higher-resolution version of the photo, substitute "Type=HrPhoto".

Additional Resources

The most comprehensive source of information on RETS is the organizational web page. Here are a few bookmarks to get you started:

- RETS organization main page: http://www.rets.org
- RETS developer resources: http://www.rets.org/cms/developer
- RETS discussion forums: http://forums.rets.org
- RETS specification documentation: http://www.rets.org/documentation
- RETS FAQ:http://www.rets.org/cms/faq
- Glossary of Terms: http://www.rets.org/glossary