RETS
Real Estate Transaction Standard

Version: 2

Meeting Package - August 2008 RETS Meeting

Information

There are four change proposals for the August meeting. Discussions on the changes will be discussed on the rets.org forum at http://forums.rets.org under the Discussion forum.

The RETS 1.7.1 document will also be voted on at this meeting and should be reviewed and discussed on the same forum.

There will be limited time during the meeting to discuss the changes and standards document, so please review the materials and make your comments before the meeting.

This document consists of the four change proposals. The standards document is provided as a separate document.

Paul Stusiak

RETS Change Proposal 66

Deprecate Lookup Types LookupBitmask and LookupBitstring

Information

Change Proposal Number: 66

Change Proposal Title: Deprecate Lookup Types LookupBitmask and LookupBitString

Originating Workgroup: RETS 1.7.1 Document

Date: June 9th, 2008

Version: 1

RETS Version: 1.8

Status: Accepted for Voting

Submitted Date: June 9th, 2008

Voting Date: August 7th, 2008

Contact Information

Author: Mark Lesswing

Organization: National Association of REALTORS(R)

Telephone:

E-mail: mlesswing@realtors.org

Author: Paul Stusiak

Organization: Falcon Technologies Corporation

Telephone:

E-mail: pstusiak@falcontechnologies.com

Synopsis

The lookup types LookupBitmask and LookupBitstring be removed from the standard and be marked deprecated because they represent implementation details and are not generally used, making the documentation of LookupType too complex.

Rationale

Confusion has existed around the LookupType metadata which has three different types of lookup representation. These three different types exist to provide some direct mapping between the standard and certain relational databases. In general use, most if not all implementations have settled on a single form, the direct mapping, where each lookup triplet (LongValue, ShortValue and Value) is represented by a single lookup Value. Having unused lookup formats does not improve the interoperability of the standard and requires additional code on the part of client vendors to support this limited feature, making a barrier to entry to the standard.

Proposal

The proposal will modify Section 11.4.3, Table 11-10, removing the definitions for LookupBitmask and LookupBitstring. The existing text,

> *The value to be sent to the server when performing a search. This field must be numeric for LookupBitmask and LookupBitstring types. For LookupBitmask fields, 2(value-1) is used to compute this component as part of the applicable choices. For LookupBitstring fields, this is the position with in the field, 1-based, at which the value contains a"1".*

will be replaced by

> *The value to be sent to the server when performing a search.*

Impact

This change will remove LookupBitstring and LookupBitmask from the standard and from the compliance tester. Existing systems that provide this feature may still have this feature, but it will be an extension to the standard. Client applications will not work with such systems without additional coding and documentation to describe how the form is used.

Given that the premise of the change proposal is that this is not a widely used form, the actual impact will be limited or no impact.

Compatibility

This change will be compatible with RETS 1.8 and higher.

Document History

Date	Version	Author	Description
June 9, 2008	1	Paul Stusiak	Initial Release
June 11, 2008	1	None	Change Board Review

RETS Change Proposal 67

Add RESO Schema as additional output formats to RETS1

Information

Change Proposal Number: 67

Change Proposal Title: Add RESO Schema as additional output formats to RETS 1

Originating Workgroup: RETS Schema, RETS Syndication

Date: June 9, 2008

Version: 1

RETS Version: 1.7.1

Status: Accepted for Voting

Submitted Date: June 9th, 2008

Voting Date: August 7th, 2008

Contact Information

Author: Paul Stusiak

Organization: Falcon Technologies Corporation

Telephone:

E-mail: pstusiak@falcontechnologies.com

Synopsis

Additional output formats are desirable for RETS 1.7 or RETS 1.8 to support the top level RESO Schema that have been adopted. Specifically the schemas Member, Office, Listing and Syndication are to be added to the list of output formats.

Rationale

The RESO Schema provide a standard way to represent many of the principal data elements of the listing process. Great interest has been expressed in having this included in the existing infrastructure built around RETS 1.7 or to incorporate this into RETS 1.8. To quickly get this

into use, providing the schemas as output formats will permit existing queries to retrieve the information in a different format, in addition to the existing formats.

Proposal

Section 7.4.2 of the RETS 1.7.1 document provides a description of the formats that can be returned. This list will be increased to allow the return of the Members, Offices, Listings and Syndication formats.

> *Format* ::= **COMPACT** | **COMPACT-DECODED** | **STANDARD-XML** | **STANDARD-XML:**dtd-version | **Members.xsd** | **Offices.xsd** | **Listings.xsd** | **Syndication.xsd**

Additional text will be provided to describe the meaning of the new formats. New text is indicated in red

> ... defined by the RETS Data XML DTD. Members.xsd means the current Members.xsd, Offices.xsd means the current Offices.xsd, Listings.xsd means the current Listings.xsd and Syndication.xsd means the current Syndication.xsd. Servers MUST ...

Servers are not expected to return schema formats that are not appropriate to the Resource. Specifically, severs may return error messages when the schema type does not match the resource type. Additional text will be added before section 7.4.3. Further, a valid response uses the appropriate schema as the response body and not the format of section 3.

> Servers MAY return an error message when the schema type does not match the resource. For example, if the requested resource was Agent and a Syndication format was requested, the server SHOULD return an error with code 20211. See Table 7-1 for the meaning of the error code.
>
> Servers SHOULD return an error when an invalid combination of Resource and Schema are requested using the Message format described in Section 3. Valid combinations of resources and schemas are described in the table below.

RESOURCE	SCHEMA
Property	Listings.xsd
Property	Syndication.xsd
Office	Offices.xsd
Agent	Members.xsd

> When a valid combination is requested and records match the request, the response format MUST use the schema definition and be well-formed XML. That means that the message format described in Section 3 is not used and the COUNT and SELECT arguments are ignored.

When a valid combination is requested and an error occurs before transmission, no records are transmitted and the message format described in Section 3 is used to transmit the error code. When a valid combination is requested and an error occurs during transmission, the client MUST manage the error state, the server is not expected to send an error code.

When a valid combination is requested and the response exceeds a business rule limit on the server, the SHOULD should return no records and an error code of 20208. Servers SHOULD provide a reply text that will assist the client to correct the condition. This reply text may include contact information, counts of returned records and the limit number, suggestions about how to reformulate the query or other descriptive text about how to resolve the problem.

Impact

This is new functionality and should not impact existing implementations. New implementations will see increased data transmission sizes based on the size of XML instance documents. Such implementations are encouraged to use HTTP compression to improve bandwidth utilization.

Compatibility

This change request is compatible with version 1.7.1 and higher.

Document History

Date	Version	Author	Description
June 9, 2008	1	Paul Stusiak	Initial Release
June 11, 2008	1	None	Change Board Review
June 15, 2008	2	Paul Stusiak	Modify schema version statement

RETS Change Proposal 69: LookupType Value

Author: Steve Clarke

Organization: MarketLinx Inc.

Telephone Number: (505) 286-4800

Address: 1400 Centerpoint Blvd. Suite 100, Knoxville, TN 37932

Email: sclarke@marketlinx.com

Status: Accepted for Voting

Date: June 6, 2008

Voting Date: August 7th, 2008

RETS Version: 1.7.1

Synopsis

Certain MarketLinx site configurations observed already do have occasional *LookupType* values defined that violate the current specification requirement that a *LookupType* value is ALPHANUM. In some cases these are in fact string fields that may have had a Lookup metadata added after the fact. Data in the database is already stored with non-ALPHANUM characters. Furthermore, in these situations, there are already RETS clients that are using this data in their feeds.

Under the current specification, we would have only two options for 100% compliancy in this area:

1) We can map these *LookupType* values to some arbitrary (100% alphanumeric) value and perform this mapping both when processing the search request as well as returning the data. This would not only require significant server development, but it would also add performance overhead and would impact all current implementations against these servers. Actually changing the data stored in the native databases is not an option because other MLS applications depend on this.

2) We could simply eliminate the *Lookup* from the metadata and treat these fields strictly as string fields. This would bring the servers into

compliance, but it would reduce the quality of the metadata and the value of the server.

This proposal intends to make *LookupType* values more flexible by allowing them to behave in a similar capacity as regular string fields, while still providing the value of the Lookup metadata.

Rationale

The RETS standard already supports non-alphanumeric data in both queries and results. The DMQL2 BNF already supports ***string-literal***, which is any TEXT except double-quote itself. DMQL parsers already can deal with this type of quoted string in search criteria against string fields.

With minor changes to the specification, a *Lookup* field can be treated exactly like a string field, except that it brings additional metadata that defines the set of possible values. This approach gives added flexibility to the systems without introducing additional overhead and without impacting existing client use cases.

Proposal

Specification Changes

11.4.3 Lookup Type

Table 11-20 Metadata Content: Lookup Type

Change content type for Value element from 1*32ALPHANUM to 1*128TEXT.

Note: this change was already discussed and voted on in December 2007 (Miami) RCP by Matt McGuire, but at the time it deferred spec review WG . Unfortunately, this was missed in the 1.7.1 work done recently.

7.7.1 Query language BNF

Change definition of lookup element to:

lookup ::= <any legal *string-literal* value as per metadata>|<any legal TEXT value for the field as per metadata>

Sample search criteria:

HEAT=%7CBSBRD,ELEC,GEOTH

HEAT=%7CBSBRD,ELEC,GEOTH,"HV/AC"

HEAT=%7C"BSBRD","ELEC","GEOTH","HV/AC"

HEAT=%7C"BSBRD","ELEC","GEOTH","HV/AC"

HEAT=%7E"BSBRD","ELEC","GEOTH","HV/AC"

HEAT=%2B"BSBRD","ELEC","GEOTH","HV/AC"

Compatibility

At certain sites, MarketLinx servers already expose non-alphanumeric *LookupType* values in RETS metadata. MarketLinx servers already support the capability of quoting a *LookupType* value in search criteria. A wide variety of client vendors are already successfully using this feature. A few vendors are unable to query using quoted *LookupType* values and are hesitant to implement because this is not presently described in the specification. This proposal will not change the way the affected servers operate. It merely formalizes the approach that is already supported. Vendors that do not support non-alphanumeric *LookupType* values and clients that choose not to support this type of query are not affected at all.

RETS Change Proposal

Search Has Key Index Support

Information

Change Proposal Number: 68

Change Proposal Title: Search Has Key Index Support

Originating Workgroup: RETS 1.7.1 Document

Date: July 8th, 2008

Version: 3

RETS Version: 1.7.1

Status: Accepted for Voting

Submitted Date: June 9th, 2008

Voting Date: August 7th, 2008

Contact Information

Author: Matthew McGuire

Organization: MarketLinx Inc.

Telephone Number: (865) 470-1500

Address: 1400 Centerpoint Blvd. Suite 100, Knoxville, TN 37932

Email: mmcguire@marketlinx.com

Synopsis

This proposal addresses a common concern among client applications designed to replicate the data available on a given RETS server. This proposal attempts to resolve these concerns by providing additional metadata and modifying the Search Transaction to retrieve the keys for a given Class.

Rationale

Presently if a client application wishes to replicate data they must either acquire all of the records using a single search request, multiple requests using OFFSET, or multiple requests using a local index of keys. The first approach is often limited to a maximum number of records. The second approach is based on a feature that may not be supported by all servers and may not guarantee all the records. The third approach depends on a previously acquired set of keys which may or may not be complete at the time of use. Additionally there is no way for a server vendor to advertise support for data replication. By providing a metadata specifically designed to advertise keys used for replication, the client can acquire a list of keys for retrieving the records accurately and the server can advertise support for replication by providing the metadata.

Proposal

Specification Changes

The following sections detail each area of the existing specification that needs to be changed or clarified and provides reasoning related to each change. Each area of the change will be listed according to the section of the specification using numbering in italics. For example changes to the METADATA-SYSTEM response format would look like the following: "***Section 11.2.1***".

This proposal extends the Search Transaction and affects server support for records limits in a specific way. Language to this effect is added to the following sections.

Section 7.4.3 Limit

The following sentence should be added to the end of the section.

Any request that sets a numeric Limit disables support for unlimited key index results as described in section ***7.4.5 Select***.

Section 7.4.5 Select

The following sentence should be added to the end of the section.

If the requested Class advertises HasKeyIndex as True in the Class Metadata and the client only selects fields advertised as InKeyIndex as True in the Table Metadata, the Server MUST return all the matching records unless the Client has declared a Limit other than NONE.

Section 11.3.1 Class

The following additional Metadata Field should be added to Table 11-7 Metadata Content: Resource Class.

HasKeyIndex – Boolean – This value declares that a Class supports the retrieval of key data for fields advertised in the Table Metadata as InKeyIndex.

Section 11.3.2 Table

The following additional Metadata Field should be added to Table 11-9 Metadata Content – Tables:

InKeyIndex – Boolean – This value declares that a field may be used in the Select argument to suppress normal Search Limit behavior following the rule outlined in Section 7.4.5.

Compatibility

The additional metadata and behavior should not impact systems before version 1.7.1. Given the common need for this behavior, including this in the 1.7.1 is desirable.

RETS Change Proposal
Search Has Key Index Support
Template Version: 1.0
Version: 3

Document History

Date	Version	Author	Description
June 9, 2008	1	Mathew McGuire	Initial Release
June 12, 2008	2	Mathew McGuire	Rework following conference call
July 8, 2008	3	Paul Stusiak	Remove certain ambiguities and reformatting