



# CARETS RETS

## User Reference Guide

**August 25, 2010**

(Revision 1.15)

**Presented by CARETS Member MLS Organizations:**

**Desert Area MLS (DAMLS)**

Combined LA/Westside MLS (CLAW)

CRIS Net MLS

i-Tech MLS

Multi Regional MLS (MRMLS)

Southern California MLS (SoCal)

Ventura County Regional Data Share (VCRDS)



# Table of Contents

<b>1. General Information .....</b>	<b>4</b>
Background .....	4
<b>2. Rules and Guidelines.....</b>	<b>4</b>
CARETS RETS/IDX User Responsibilities .....	4
Permission Types .....	5
IDX/Public Feed Restrictions .....	5
Property Record Restriction Field.....	6
<b>3. Disclaimer.....</b>	<b>6</b>
<b>4. CARETS RETS Metadata Summary .....</b>	<b>7</b>
CARETS Search Type Information .....	7
CARETS RETS Metadata Field Naming Conventions.....	7
Lookup Value Notation.....	8
Important Agent Resource Fields.....	8
Important Office Resource Fields .....	10
Important Property Resource Fields.....	11
Important History Resource Fields .....	12
Important Open House Resource Fields .....	13
Important Media Resource Fields and Notes .....	14
Unit Resource (applies to Residential Income).....	14
Deletion Resource .....	15
<b>5. CARETS Access and Security.....</b>	<b>16</b>
CARETS Login and Security Credentials.....	16
Accessing CARETS RETS Servers .....	16
<b>6. RETS Query Examples:.....</b>	<b>17</b>
Search Type: .....	17
Class:.....	17
Query: .....	17
Search syntax: .....	18
Search Options: (Optional) .....	19
Select: .....	19
Format:.....	19
Limit: .....	19
StandardNames: .....	19
Count: .....	19
Search Offsets .....	19
Examples of fully formed RETS queries: .....	20
Example of searching for a listing whose price has changed in the last week: .....	20
Example of retrieving Back on Market Date data: .....	20
Examples of fully formed RETS Media queries: .....	20
Example of Deletion query: .....	20
<b>7. Updated and Deleted Listings: .....</b>	<b>21</b>
<b>8. CARETS DOM/CDOM Field Names and Definitions.....</b>	<b>22</b>
<b>9. CARETS Technical Contact Information.....</b>	<b>23</b>



**Appendix A – IDX/Public Publication Guidelines ..... 25**

**Appendix B - Sample Login/Logout trace: ..... 28**

**Appendix C - CARETS® RETS Supported Features ..... 31**

**Appendix D - Deletions Resource FAQ..... 34**

**Appendix E – Timed Downloads FAQ..... 35**



## 1. General Information

### ***Background***

CARETS is an aggregated property listing data feed emanating from participating California MLS organizations. Data is exchanged between its users and MLS organizations using the Real Estate Transaction Standard (RETS) supported by the National Association of Realtors®. Information regarding RETS can be found at [www.rets.org](http://www.rets.org).

The CARETS RETS Service is hosted by the MRIS Cornerstone system (<http://www.mris.com>).

CARETS RETS users can subscribe through any one of the participating CARETS MLS organization described in Section 8 “CARETS Contact Information”. Users are required to follow the rules and guidelines outlined in this document and their MLS RETS user agreement.

All administrative and support questions should go directly to your primary CARETS MLS that supplied you with the CARETS security and logon credentials.

## 2. Rules and Guidelines

### ***CARETS RETS/IDX User Responsibilities***

All CARETS users are responsible to verify, on a daily basis, that their Broker and/or Agent customer(s) have an “active” status. The CARETS user must immediately turn off services for their customer if that customer does not have an active status. If there are any questions or issues regarding a specific agent and/or office then contact the MLS responsible for that agent/office.

Many CARETS fields have restrictions defined by the Permission Type (*see next section*) and the viewer (i.e. Agent or Consumer). By default, all fields will be searchable and viewable to the public unless explicitly noted otherwise. If a field is deemed restricted then that field should not be searched or viewed.

***All CARETS users must ensure that CARETS restricted field rules are adhered to by themselves and any other party (direct or indirect) that can view CARETS data.***

## Permission Types

Permission Types define the rules placed on the CARETS user and their viewers.

As an example, it is possible that a CARETS user, designated with a Full Feed permission type, could deliver information to the public or a user with IDX/Public Feed permission. The CARETS user with Full Feed permission would then need to abide by the IDX/Public Feed rules while delivering CARETS data.

Each CARETS user is assigned one Permission Type.

The possible permission types follow:

1. **MLS Feed:** Applies to CARETS users that will represent a CARETS MLS and will deliver MLS information to authorized Brokers and Agents.
2. **Full Feed:** Applies to CARETS users that are or represent a CARETS Broker with an active status.
  - **Restrictions:** Full Feeds will have access to all CARETS resources less PROPERTY resource fields Expiration Date and Pending Price information for listings they do not own.
3. **IDX/Public Feed:** A user that will be showing MLS information to the general public.
  - **Restrictions:** Expiration Date and Pending Price are not searchable or transmitted to users with an IDX/Public Feed permission type.
  - **Other Restrictions:** The IDX/Public Feed has the most restrictions. Detail information can be found in "**Appendix A - IDX Feed Restrictions**" as well as other sections throughout this document.

## IDX/Public Feed Restrictions

### IDX/Public Feed Resources

- All IDX/Public feeds will gain access to Property, Agent, Office, Media, Open House, Deleted and Zapped Class. Change/History will not be made available to IDX/Public feeds. Fields that cannot be displayed are outlined in "**Appendix A - IDX Feed Restrictions**".

### IDX Office/Agent Information

- IDX/Public feeds will have access to Office/Agent status fields as well as the Office/Agent contact fields in the Office/Agent resources. Address information is not considered to be a contact field.

### Listing Status

- IDX/Public feeds can only search and display those listings with a CARETSListingStatus of: Active, Back Up Offer, Pending, Closed or Leased.

### Pending/Sales Data

- All IDX/Public feeds will include 1 year of Sold data and Pending data. Cancellations and other Off Market listings are not to be included. Expiration Date and Pending Price can never be queried or displayed. Closing price (ClosePrice) is only searchable and visible with a Closed or Leased status.

## Property Record Restriction Field

### ListingSubscriptionClassList

The ListingSubscriptionClassList is a **multi-lookup** field in the Property Resource that describes how a specific listing can be viewed by the public. The possible lookup values and their meaning follow:

### Publish Lookup Values

- CIDX-DNP Do Not Publish Listing for Public Access
- CIDX-PUB Publish Listing for Public Access

### Publish Lookup Values with Address Restrictions

- CIDX-PNS Publish without Street Number information (*details in Appendix "A"*)
- CIDX-PNA Publish without Address information (*details in Appendix "A"*)

### Acting on Restrictions

If a record is restricted (CIDX-DNP) then the total record should be omitted from the feed. If a field is restricted then that field should not be searchable and omitted or null in the xml feed.

**Example:** IDX/Public Feed permission type user pulls a CIDX-PNA listing, the address fields are either null or removed completely from the record. If that same user pulls a batch of listings and one is a CIDX-DNP listing, then that entire listing is omitted from the feed.

ListingSubscriptionClassList is a restricted field and cannot be searched or viewed by a user with IDX/Public Feed Restrictions. The IDX/Public feed user will already have those removed from the downloaded dataset.

## 3. Disclaimer

The following disclaimer needs to be placed on the bottom of each displayable page that a CARETS listing appears.

*The information being provided by CARETS is for the visitor's personal, non-commercial use and may not be used for any purpose other than to identify prospective properties visitor may be interested in purchasing.*

*Any information relating to a property referenced on this web site comes from the Internet Data Exchange (IDX) program of the CARETS MLS. This web site may reference real estate listing(s) held by a brokerage firm other than the broker and/or agent who owns this web site.*

*Any information relating to a property, regardless of source, including but not limited to square footages and lot sizes, is deemed reliable but not guaranteed and should be personally verified through personal inspection by and/or with the appropriate professionals.*

*The data contained herein is copyrighted by CARETS® and is protected by all applicable copyright laws. Any dissemination of this information is in violation of copyright laws and is strictly prohibited.*

## 4. CARETS RETS Metadata Summary

### CARETS Search Type Information

- **Agent**
- **Office**
- **Property**
  - **CARETS ALL (ALL)** – Search properties regardless of property type
  - **CARETS Residential (RES)**
  - **CARETS Residential Lease (RNT)**
  - **CARETS Residential Income (RIN)**
  - **CARETS Mobile Home (MHE)**
  - **CARETS Land (LND)**
  - **Zap** – A collection of property listings that have been temporarily removed but may be reinstated.
- **Unit** – Unit information for Residential Income
- **OpenHouse** – Agent and public open house information
- **Media** – Photos and Virtual Tours
- **Deletions** – Properties permanently deleted from CARETS database
- **History** – A detail transaction log of every change for the other search types

### CARETS RETS Metadata Field Naming Conventions

SystemName	Resource	Example(s)	Description
<b>*ID</b>	All	UniqueID, ListingID, SaleOfficeID, AltOfficeID	Identification fields that are derived from MLS Systems.
<b>*Key</b>	All	ListingKey	Key fields are internal CARETS system fields providing connections between different resources or classes. These are numeric and guaranteed to be unique.
<b>*Timestamp</b>	All	ModificationTimestamp	A date-formatted field that contains the date/time a record (or its relations) was last modified in the CARETS System.
<b>ListAgent*</b>	Property	ListAgentFirstName	Fields related to the Listing Agent.
<b>AltListAgent*</b>	Property	AltListAgentFirstName	Fields related to the Alternate Listing or Co-Listing Agent.
<b>SaleAgent*</b>	Property	SaleAgentFirstName	Fields related to the Selling Agent.
<b>AltSaleAgent*</b>	Property	AltSaleAgentFirstName	Fields related to the Alternate Selling or Co-Selling Agent.

## Lookup Value Notation

CARETS maintains three different types of Lookup values. The type of lookup can be ascertained by looking for a single letter enclosed in a parentheses at the end of the Lookup LongValue.

Lookup Type	Description	Notation/Example
Common	A lookup value that is supported by ALL CARETS MLS.	A common lookup will NOT have (N) or (L) following the lookup LongValue
Non Common	A lookup value that <b>may not</b> be adopted by ALL CARETS MLS.	(N) / Example: Jogging Track (N)
Legacy	A lookup that is in the CARETS database but is no longer accepted for new listings going forward.	(L) / Example: Koreatown (KREA) (L)

## Important Agent Resource Fields

SystemName	Definition
<b>AgentContactOrder1 – AgentContactOrder6</b>	The preferred order to contact responsible agent. This is not the contact information, but the description of the contact type. For example: "Agent Cell Ph", "Agent Direct Ph", "Agent Email".
<b>AgentDesignations</b>	Designations held by Agent, ie., GRI, PH.D
<b>AgentEditor</b>	The last person (AgentID) that modified the agent record.
<b>AgentExternalSystemID</b>	A CARETS Agent Unique Identifier field composed of Agent ID information (as shown below) concatenated with "AGT" and SubSystemLocale identifiers ("SOCAL", "DAMLS", "CLAW", "CRISNet", "ITECH", "MRMLS", "VCRDS") * CLAW/CRIS Net/SoCal:AgentID * i-Tech/VCRDS/DAMLS: NRDS ID Number * MRMLS: Matrix ID Number
<b>AgentID</b>	Agent Identifier used by the local MLS. Not guaranteed to be unique
<b>AgentKey</b>	CARETS Unique Agent Identifier
<b>AgentLicenseNumber</b>	Agent DRE (Dep of Real Estate) License Number
<b>AgentModificationTimestamp</b>	This timestamp depicts the date and time an agent field in the Agent Resource was modified in the CARETS database. This timestamp also updates when data in the related Office record is changed since Agent is a dependent resource of Office.
<b>AgentPrimaryObjectModificationTs</b>	This timestamp depicts the date and time a field in the agent resource was modified in the CARETS database, either migrated from the source or by some other CURE internal process. Any update to a dependent resource is NOT included in this timestamp. Agent is a dependent resource of Office, therefore, Office updates that affect Agent fields are NOT reflected in this field.
<b>AgentOfficeID</b>	Office Identifier used by the local MLS. Not guaranteed to be unique
<b>AgentSourceCreationTimestamp</b>	MLS Date Date/Time this record was added to MLS



<b>AgentSourceModificationTimestamp</b>	MLS Date Date/Time this record was last updated
<b>AgentStatus</b>	<p>Describes the current status of the agent within in the system:</p> <ul style="list-style-type: none"> <li>* Active: Active status enables the member to login</li> <li>* Pending: Obsolete</li> <li>* Inactive: Member cannot access the system</li> <li>* Pending/Inactive: Obsolete</li> <li>* Terminated: Member cannot access the system</li> </ul>
<b>AgentSubSystemLocale</b>	<p>An internal field that identifies the sub-system where the Agent listing originated. A sub-system is equivalent to a member MLS systems participating in CARETS. Valid value for this field equals: "CARETS_CLAW", "CARETS_CRISNET", "CARETS_ITECH", "CARETS_MRMLS", and "CARETS_SOCAL", "CARETS_DAMLS", "CARETS_VCRDS".</p>
<b>AgentSystemLocale</b>	<p>An internal field that identifies the system destination system. The only valid value for this field equals "CARETS".</p>
<b>AgentUserName</b>	The CARETS Agent MLS User Name (not completed by every MLS)
<b>OfficeKey</b>	CARETS Unique Office Identifier pointing to the Office Resource

**\*\* Please note that a field whose meaning is not clear is included in these resource field sections. Fields whose meanings were obvious from the field name or previous descriptions have been excluded.**

## Important Office Resource Fields

SystemName	Definition
<b>AgentOfficeID (OfficeID)</b>	The MLS Office Identifier. Not guaranteed to be unique
<b>BrokerID</b>	The MLS Office Identifier, concatenated with "OFC" and SubSystemLocale identifiers ("SOCAL", "CLAW", "ITECH", "MRMLS"). Each MLS uses a different identifier: * CLAW: AgentOfficeID * CRISNet: AgentOfficeID * i-Tech/DAMLS/VCRDS: AgentOfficeID * MRMLS: Matrix ID Number * SoCal: AgentOfficeID
<b>OfficeBrokerID</b>	Broker Identification field associated with the Listing Agent's Broker
<b>OfficeCarrierRoute</b>	Contains the USPS Carrier Code.
<b>OfficeCompanyMainOfficeID</b>	(not used)
<b>OfficeEditor</b>	The last person (AgentID) that modified the agent record.
<b>OfficeKey</b>	CARETS Unique Office Identifier.
<b>OfficeModificationTimestamp</b>	CARETS Date/Time this record was last updated
<b>OfficeResponsibleMemberID</b>	(not used)
<b>OfficeSourceCreationTimestamp</b>	MLS Date Date/Time this record was added to MLS
<b>OfficeSourceModificationTimestamp</b>	MLS Date Date/Time this record was last updated
<b>OfficeStatus</b>	Describes the type of agent found in the associated listing: A, I, T, Purge (Active, Inactive, Terminated, Purge)
<b>OfficeSubSystemLocale</b>	An internal field that identifies the sub-system where the Office listing originated. A sub-system is equivalent to a member MLS systems participating in CARETS. Valid value for this field equals: "CARETS_CLAW", "CARETS_CRISNET", "CARETS_ITECH", "CARETS_MRMLS", and "CARETS_SOCAL", "CARETS_DAMLS", "CARETS_VCRDS".
<b>OfficeSystemLocale</b>	An internal field that identifies the system destination system. The only valid value for this field equals "CARETS".
<b>OfficeType</b>	* Appraiser Office: Identifies Appraiser Office; Office will be comprised of member type: Appraiser * Reciprocal Office: Identifies a Reciprocal Office; Used to provide reciprocal KEY Access * MLS only Office: Office contains Member Types: Designated Broker MLS Only and MLS Only * Affiliate Office: Office contains Member type: Affiliate Member * REALTOR Office: Office contains Member types: Designated Realtor, REALTOR, Secondary DR, Secondary REALTOR * Commercial Office: Not used * Secondary Realtor: Not used. * MLS Staff: Identifies Office that contains Member type: Staff

## Important Property Resource Fields

SystemName	Description
[AgentDesc]ID	Agent Identification field used by MLS. Not guaranteed to be unique.
[AgentDesc]Key	Agent Identifier guaranteed to be unique within CARETS
[AgentDesc]Languages	The language(s) that are spoken by the Agent.
[AgentDesc]OfficeKey	The MLS-specific field that identifies the Office for the Selling Agent in the Office Resource. Guaranteed to be unique within CARETS.
[OfficeDesc]ID	Identifier used by the local MLS. Not guaranteed to be unique
AreaOther	An area not already defined in the CARETS Area lookup list.
BathsLegacyTotal	String allowing for whatever total of bathrooms existed in legacy systems. Examples may be "three full and one half", 1.2, One and a half, 1.5. Keep only for historical/archive listings
CARETSListingStatus	The preferred Listing Status for the field in the CARETS system. <i>The CARETS Field ListingStatus is a generic RETS compatible status field. (see below)</i>
ContactOrder1 – ContactOrder6	This field indicates the first contact for the property listing. These are not the contact information, but the description of the contact type. For example: "Agent Cell Ph", "Agent Direct Ph", "Agent Email". Not all of these fields are used by each MLS.
DepositKey	The dollar amount paid by the rentor/lease to receive the keys to the property described in the listing.
LastMediaDate	This timestamp depicts the date and time a Media record was inserted, updated or deleted from the Media Resource in CARETS. This is for all Media types including Photos, Virtual Tour and Documents
LastModifiedBy	AgentID that last modified the property resource record..
LastPhotoDate	This timestamp depicts the date and time a photo was inserted, updated or deleted from the Media Resource in CARETS. This is for Media type of photo only
ListingID	The CARETS system representation of the local MLS Listing ID/ML number. Slight formatting changes may be made from the original MLS formatting. This is not guaranteed to be unique.
ListingKey	The CARETS guaranteed unique identifier for a listing
ListingModificationDate	CARETS Date/Time record was last updated by the source MLS
ListingStatus	ListingStatus is a required RETS field that maps the value from listing status field <b>CARETSListingStatus</b> to the RETS standard listing statuses. This is a derived RETS compatible field.
ListingSubscriptionClassList	"ListingSubscriptionClassList" can have one of the following values: <ul style="list-style-type: none"> <li>* CIDX-PUB (Means: Publish including all address fields.)</li> <li>* CIDX-PNS (Means: Publish with No Street Number fields.)</li> <li>* CIDX-PNA (Means: Publish with No Address fields of any kind.)</li> <li>* CIDX-DNP (Means: Do Not Publish.)</li> </ul>
ListingSubscriptionList	CARETS Internal field - Users can ignore
MatchCode	A lookup value which indicates a confidence level how well CARETS has geocoded the property listing.
ModificationTimestamp	CARETS Date/Time record was last updated <b>or</b> any related resource records like agent, office, media, etc. Example: If listing agent resource record is updated, all of that agent's listing records will have its ModificationTimestamp updated as well.
OpenOtherSpacesTotal	Parking spaces that are not considered Garage, Carport or Covered.

<b>PrimaryObjectModificationTimestamp</b>	CARETS Date/Time record was last updated <b>BUT NOT</b> related resource records like agent, office, media, etc. <i>note: This field is only populated after 8/4/2009.</i> Example: If listing agent resource record is updated, that agent's listing records will <b>NOT</b> have its ModificationTimestamp updated.
<b>RenewalCompensation</b>	RenewalCompensation indicates whether there is compensation offered when a lease on the property listing is renewed, or not.
<b>SubSystemLocale</b>	An internal field that identifies where the listing originated. A sub-system is equivalent to a member MLS systems participating in CARETS. Valid value for this field equals: "CARETS_CLAW", "CARETS_CRISNET", "CARETS_ITECH", "CARETS_MRMLS", and "CARETS_SOCAL", "CARETS_DAMLS", "CARETS_VCRDS".
<b>SystemLocale</b>	Ignore. The only valid value for this field equals "CARETS".
<b>UniqueID</b>	The CARETS system globally unique identifier with the local MLS listing number (ListingID) concatenated with an abbreviated MLS identifier. (i.e. "22113277:ITEC")

### Important History Resource Fields

SystemName	Definition
<b>PropHistChangeTimestamp</b>	This timestamp depicts the date and time a record was inserted into the CARETS History Resource whether it was a NEW listing, agent etc., or an update to any field in the resources
<b>PropHistChangeType</b>	Category Description of the nature of the change * DOMP: Days On Market Property * MEDIA: Change in the Media * NEW: New Listing * PARTY: Change in an Agent, Owner, or Occupant related field * PRICE: Change in Listing Price * STATUS: Change in Listing Status * OTHER: Any change that cannot be included in one of the previous category types
<b>PropHistColumnName</b>	Name of the Internal CARETS Database Column where the value is being changed. Can be ignored.
<b>PropHistItemNumber</b>	Internal sequence to handle MVOs
<b>PropHistKey</b>	The unique key for a history change.
<b>PropHistListingKey</b>	The CARETS unique identifier for a listing's Property History across all member-MLS systems.
<b>PropHistNewColumnValue</b>	New column value
<b>PropHistNewPickListValue</b>	New picklist value is the column is a lookup
<b>PropHistOriginalColumnValue</b>	Old column value
<b>PropHistOriginalPickListValue</b>	Old picklist value is the column is a lookup
<b>PropHistPartyKey</b>	Internal key of the user who made the change
<b>PropHistRecordKey</b>	Unique key the column value belongs to.
<b>PropHistSubSystemLocale</b>	Sub system the change belongs to (CLAW_SOCAL,etc).
<b>PropHistSystemLocale</b>	System the change belongs to. (CARETS)
<b>SystemName</b>	RETS Metadata SystemName. (Internal Use Only)
<b>FullStreetAddress</b>	Address of the listing being changed.
<b>ListingID</b>	MLS Number for the listing being changed.

## Important Open House Resource Fields

SystemName	Definition
<b>ListingID</b>	Local MLS Number (not guaranteed to be unique)
<b>OpenHouseBegins</b>	Time the open house starts
<b>OpenHouseClose</b>	Time the open house ends
<b>OpenHouseDate</b>	Open house date.
<b>OpenHouseExternalSystemID</b>	Open house unique key from the source system
<b>OpenHouseItemNumber</b>	Internal sequence to number open houses within the listing
<b>OpenHouseKey</b>	CARETS Unique Open House Identification field
<b>OpenHouseListingKey</b>	The ListingKey field value for the Property Listing associated with this Open House entry
<b>OpenHouseModificationTimestamp</b>	This timestamp depicts the date and time a field in the Open House Resource was modified in the CARETS database
<b>OpenHouseShowingAgentID</b>	AgentID of the agent showing the house.
<b>OpenHouseSubSystemLocale</b>	Sub system the change belongs to (CLAW_SOCAL,etc).
<b>OpenHouseSourceCreatedBy</b>	Person who entered the open house in the source system
<b>OpenHouseSourceCreationTimestamp</b>	Time and date the open house was entered in the source system.
<b>OpenHouseSourceModificationTimestamp</b>	Time and date the open house was last modified in the source system.
<b>OpenHouseType</b>	Type of open house (i.e. Broker)
<b>PropertyType</b>	Property type of the listing.
<b>UniqueID (ExternalSystemID)</b>	Listing Unique key from the source system (see <i>Property Resource above for detail</i> )

## Important Media Resource Fields and Notes

SystemName	Definition
PropMediaBytes	Size of the photo in bytes.
PropMediaCaption	Photo caption
PropMediaCreatedTimestamp	Time and date when the photo was created in the system
PropMediaDescription	Additional description for this photo.
PropMediaDisplayOrder	Display order of this photo.
PropMediaExternalKey	MLS key of this photo from the source MLS system.
PropMediaFileName	Photo path or URL where the photo is found. Used internally.
PropMediaKey	CARETS Unique key
<b>PropMediaModificationTimestamp</b>	<b>CARETS Date and time the photo was modified last. Note: Any media field updated will trigger an update to the PropMediaModificationTimestamp</b>
PropMediaSize	Photo size (Large, Medium, Thumb, etc).
PropMediaSubSystemLocale	Sub system the change belongs to (CLAW, SOCAL, etc).
PropMediaSystemLocale	System the change belongs to. (CARETS)
PropMediaType	Photo type
PropMediaURL	Photo URL
PropMediaVendorID	<i>(please ignore)</i>
PropMimeType	Photo Mime Type (Always JPG)
PropObjectKey	Listing internal key, it is the same ListingKey.
<b>UniqueID (ExternalSystemID)</b>	External key of the listing this photo belongs to with the addition of the source MLS abbreviation.
PropItemNumber	Internal sequence to number photos within a listing and photo type.
ListingID	ListingID from Property Resource (not guaranteed to be unique)

CARETS uses the RETS GetObject to retrieve media information as images change on each of the source MLS systems. Hence, the PropItemNumber will equal whatever is retrieved from the source MLS systems. For the most part re-numbering does occur if an agent deletes image #1 but this is not always the case.

CARETS has implemented its media resource per the RETS specification hence the sent Object-ID will be a sequential number starting from 1. If a user asks for Object-ID of 0, they will get Object-ID of 1 which is the first image which is how you ask for the main image in RETS GetObject. **The first image returned is considered to be the main photo.**

It is fully possible to have ItemNumbers in the Media Resource which do not match the Object-ID that comes out of the GetObject call.

## Unit Resource (applies to Residential Income)

SystemName	Definition
PropUnitItemNumber	CARETS Property Unit Identification Number. Each Property Listing could contain a PropUnitItemNumber of "1", "2", etc.
PropUnitKey	CARETS Unique Identification field for the property unit associated with the listed property
PropUnitListingKey	The ListingKey field value for the Property Listing associated with this Property Unit
PropUnitModificationTimestamp	This timestamp depicts the date and time a field in the Unit Resource was modified in the CARETS database.
PropUnitPartKey	CARETS Unique Identification Number for the specific property unit

## ***Deletion Resource***

<b>SystemName</b>	<b>Definition</b>
<b>DeletionTimestamp</b>	Date and time the listing/media/photo record was deleted
<b>SchemaShortName</b>	Table that stored that deleted record
<b>TableName</b>	Table that stored that deleted record
<b>UniversalKey</b>	Internal key of the deleted record. Cornerstone Unique Identification field of anything that has been deleted

## 5. CARETS Access and Security

### **CARETS Login and Security Credentials**

- User ID (*this will be provided to you*)
- Password (*this will be provided to you*)
- User-Agent (*this will be provided to you*)
- User-Agent Password (*RETS-UA-Authentication is implemented, but not required*) - If you would like to use this feature, you will need to provide us with a password that will be assigned to your assigned User-Agent.
- Login URL (*this will be provided to you*)
- RETS Version (*CARETS is compatible with RETS versions 1.01, 1.5 & 1.7*)

Please refer to sections 3 and 4 of the RETS specification for a complete list of arguments and their usage

### **Accessing CARETS RETS Servers**

Each RETS account has access to the CARETS Test Server and Production Server. The test server should be used for viewing latest changes in the RETS metadata. New and modified data elements will be loaded into the test server (sometimes coined as PTEST) prior to transmission to Production.

CARETS Production URL: <http://carets.retscore.com:6103/platinum/login>

CARETS Test (PTEST) URL: <http://ptest.mris.com:6103/ptest/login>

Security Credentials to the Test Server (PTEST) are as follows:

**UserID:** CARETSTEST

**Password:** PCARETSTEST

**User Agent:** MRIS Conduit/2.0



## 6. RETS Query Examples:

Login URL: <http://ptest.mris.com:6103/ptest/login> – CARETS QA/Test Only

Search URL: <http://ptest.mris.com:6103/ptest/search> - CARETS QA/Test Only

(\***Note:** Production URL will be delivered with your security credentials from the MLS)

### Search Options: (Required)

#### Search Type:

&SearchType=Agent  
 &SearchType=Office  
 &SearchType=Property  
 &SearchType=Media  
 &SearchType=History  
 &SearchType=OpenHouse  
 &SearchType=Deletions

#### Class:

&Class=Agent	Agent	(SearchType=Agent)
&Class=Office	Office	(SearchType=Office)
&Class=RES	CARETS Residential	(SearchType=Property)
&Class=RNT	CARETS Residential Lease	(SearchType=Property)
&Class=ALL	CARETS All Property Types	(SearchType=Property)
&Class=Zap	CARETS Zap	(SearchType=Property)
&Class=PROP_MEDIA	CARETS Property Media	(SearchType=Media)
&Class=PROP_HIST	Property History	(SearchType=History)
&Class=DELETIONS	Deleted Objects	(SearchType=Deletions)

#### Query:

&Query= In this section of the query string enter your search arguments. Syntax will be explained below.

The “&Query=” portion of the search string should be constructed as follows. All search arguments will be contained within a set of brackets (). Each individual search argument should also be contained within a separate set of brackets (). There should be no spaces and commas should separate search arguments.

Ex. “&Query=(ListPrice=100000-135000),(ListingStatus=|ACTIVE), (County=|ORANGE-CA)”

**Search syntax:**

**OR searches:**

The '|' character is used to signify 'OR' searches.

Ex. (LocaleListingStatus=|Active,Contract,Sold) will find any listing with a STATUS = Active or Contract or Sold.

**NOT searches:**

The '~' character is used to signify 'NOT' searches.

Ex. (BathroomFeatures=~WALKINCLOSET,CONVERTEDR%20BEDROOM) will find listings where the Bathroom Features do NOT have a Walk-in Closet or a Converted Bedroom.

**AND searches:**

The '%2B' character is used to signify 'AND' searches.

Ex. (CoolingType=%2BCENTRAL%20A/C,CEILING%20FAN) will find listings with central air-conditioning AND ceiling fans.

**Search for text CONTAINING:**

Using an '\*' character on both sides of a string will search for a value containing that string.

Ex. (Subdivision=\*Mont\*) will find listings with a subdivision name containing 'Mont'.

**Searching for text STARTING WITH:**

Using an '\*' character on the right side of a string will search for a value beginning with that string.

Ex. (Subdivision =Mont\*) will find listings with a subdivision name starting with 'Mont'.

**Searching for text with an UNKNOWN character:**

Using a '?' character in place of the unknown character will search for a string identical to the entered string with ANY character in place of the '?'.  
Ex. (Subdivision =Mont?) will find listings with a subdivision name starting with 'Mont'.

**Searching for values greater than:**

Using the '%2B' character at the end of a numeric value in a query will provide a 'greater than or equal' to search.

Ex. (ListPrice=100000%2B) will find listings with a ListPrice >= \$100,000.00

**Searching for values less than:**

Using the '-' character at the end of a numeric value in a query will provide a 'less than or equal' to search.

Ex. (ListPrice=100000-) will find listings with a ListPrice <= \$100,000.00

**Searching for a range of values:**

Using the '-' character between two values in a query will result in a range search.

Ex. (ListPrice=100000-135000) will find listings with a ListPrice between \$100,000.00 and \$135,000.00.

## Search Options: (Optional)

### Select:

&Select= In this section of the query string enter the columns to display, separated by commas, with no spaces.

Ex. &Select=ListingID,ModificationTimestamp,ListingStatus,ListPrice

### Format:

&Format= In this section of the query string, specify the format in which you would like the data returned. Available formats are currently COMPACT, COMPACT-DECODED, and STANDARD-XML. Leaving the format unspecified will result in data being returned in COMPACT. For further explanation of these options please see section 13 of the Real Estate Transaction Specification Version 1.0.

### Limit:

&Limit= Use this option to specify a limit for the number of results returned. Setting the limit equal to 'NONE' will return 5000 possible results. The default setting is 500.

### StandardNames:

&StandardNames= Use this option to retrieve RETS defined standard column names, or system column names.  
Values:  
0 = System Column Names (default)  
1 = RETS defined column names

### Count:

&Count= This option turns on and off the count tag.  
Values:  
0 = Records Only (default)  
1 = Record Count and Records  
2 = Count Only

### Search Offsets

There is a 500,000 record limit so grouping with the offset feature is sometimes helpful. To use Offset with MRIS Cornerstone, the first query MUST start with Offset=1 so that the system can setup for a consistent return of an Offset. Then the SAME logged in session must submit the EXACT SAME DMQL the only exception being that Offset needs to increment by whatever Limit you use as part of the Offset call. Now, with history, the first call might take some time as it has to grab and sort all the keys, but every subsequent call will be very quick.

All other optional search options specified in the RETS standard are not included in the current build for CARETS.

For further explanation of the query language used in RETS please refer to section 7.7 of the Real Estate Transaction Specification Version 1.0.

### **Examples of fully formed RETS queries:**

The following query will search for any Active listings in Orange County, CA, with a List Price of \$100,000.00 or less, with an In-Ground Pool. It will be output in XML format, returning a maximum of 50 records.

[http://ptest.mris.com:6103/ptest/search?limit=50&QueryType=DMQL2&StandardNames=0&Count=0&Format=STANDARD-XML&SearchType=Property&Class=ALL&Query=\(ListPrice=100000-\),\(County=|10000003263\),\(PoolConstruction=|50000897945\)\)](http://ptest.mris.com:6103/ptest/search?limit=50&QueryType=DMQL2&StandardNames=0&Count=0&Format=STANDARD-XML&SearchType=Property&Class=ALL&Query=(ListPrice=100000-),(County=|10000003263),(PoolConstruction=|50000897945)))

### **Example of searching for a listing whose price has changed in the last week:**

This DMQL will query the History Resource for a PRICE change within a date range and will be output in COMPACT-DECODED format, returning the original price, new price and listing ID.

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP\\_HIST&Query=\(PropHistChangeType=PRICE\),\(PropHistChangeTimestamp=2008-07-07T16:02:52-2008-07-11T12:54:03\),\(SystemName=ListPrice\)&Select=PropHistNewColumnValue,PropHistOriginalColumnValue,ListingID](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP_HIST&Query=(PropHistChangeType=PRICE),(PropHistChangeTimestamp=2008-07-07T16:02:52-2008-07-11T12:54:03),(SystemName=ListPrice)&Select=PropHistNewColumnValue,PropHistOriginalColumnValue,ListingID)

### **Example of retrieving Back on Market Date data:**

This DMQL will query the History Resource. You can use PropHistChangeTimestamp for your date range, and this will output the ListingID and ListingKey, as well as the timestamp. You can optionally drop the SystemName on output.

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP\\_HIST&Query=\(SystemName=BackToActiveDate\),\(PropHistChangeTimestamp=2008-07-01T18:57:19%2b\)&Select=ListingID,PropHistListingKey,PropHistChangeTimestamp,SystemName](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP_HIST&Query=(SystemName=BackToActiveDate),(PropHistChangeTimestamp=2008-07-01T18:57:19%2b)&Select=ListingID,PropHistListingKey,PropHistChangeTimestamp,SystemName)

### **Examples of fully formed RETS Media queries:**

By using the value from the ListingKey from the Property Resource and searching with that value on PropObjectKey from the Media Resource. The following query will search for all the media records from a single ListingKey and will be output in Compact-Decoded format.

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP\\_MEDIA&Query=\(PropObjectKey=50005497282\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP_MEDIA&Query=(PropObjectKey=50005497282))

The following query will search for all the media records with a modification timestamp of 06/18/2008 and greater. It will be output in Compact-Decoded format, return a maximum of 25 records.

[http://ptest.mris.com:6103/ptest/search?limit=25&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP\\_MEDIA&Query=\(PropMediaModificationTimestamp=2008-06-16T14:00:00%2b\)](http://ptest.mris.com:6103/ptest/search?limit=25&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP_MEDIA&Query=(PropMediaModificationTimestamp=2008-06-16T14:00:00%2b))

The following query will search for all virtual tours URL's using the PropMediaSize column with a modification timestamp of 06/18/2008 and greater. It will be output in Compact-Decoded format.

[http://ptest.mris.com:6103/ptest/search?&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP\\_MEDIA&Query=\(PropMediaSize=|1000000293\),\(PropMediaModificationTimestamp=2008-06-16T14:00:00%2b\)](http://ptest.mris.com:6103/ptest/search?&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Media&Class=PROP_MEDIA&Query=(PropMediaSize=|1000000293),(PropMediaModificationTimestamp=2008-06-16T14:00:00%2b))

### **Example of Deletion query:**

The following query will identify open house records deleted from the Open House resource.

[http://ptest.mris.com:6103/ptest/search?&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=\(SchemaShortName=LIS\),\(TableName=OPEN\\_HOUSE\),\(DeletionTimestamp=2009-10-20T16:40:03-2009-11-05T17:40:03](http://ptest.mris.com:6103/ptest/search?&QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=(SchemaShortName=LIS),(TableName=OPEN_HOUSE),(DeletionTimestamp=2009-10-20T16:40:03-2009-11-05T17:40:03)

## 7. Updated and Deleted Listings:

Due to the security profile that may have been assigned, you will only see the Active listings in your data pulls. To find listings that should be removed, please review the queries in **Example A** below.

The first query in **Example A** will let you know what listings have moved from an Active status to a non-displayable status of Cancelled, Expired and Withdrawn (Entering date/times needed). If you still see any orphaned records after running these updates, they may have been deleted.

You can find the orphaned listings with the query in **Example B** below (Entering date/times needed).

### **Example A:**

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL&StandardNames=0&Count=1&Format=COMPACT-DECODED&SearchType=History&Class=PROP\\_HIST&Query=\(PropHistOriginalColumnValue=50041200437\),\(PropHistChangeType=STATUS\),\(PropHistChangeTimestamp=2009-04-01T00:00:01-2009-07-06T00:00:01\),\(PropHistNewColumnValue=|50041200445,50041200441,50041200439\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL&StandardNames=0&Count=1&Format=COMPACT-DECODED&SearchType=History&Class=PROP_HIST&Query=(PropHistOriginalColumnValue=50041200437),(PropHistChangeType=STATUS),(PropHistChangeTimestamp=2009-04-01T00:00:01-2009-07-06T00:00:01),(PropHistNewColumnValue=|50041200445,50041200441,50041200439))

### **Example B:**

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP\\_HIST&Query=\(PropHistOriginalColumnValue=\\*\),\(SystemName=LocaleListingStatus\),\(PropHistChangeTimestamp=2009-07-11T16:23:59-2009-07-12T16:23:59\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=History&Class=PROP_HIST&Query=(PropHistOriginalColumnValue=*),(SystemName=LocaleListingStatus),(PropHistChangeTimestamp=2009-07-11T16:23:59-2009-07-12T16:23:59))

When maintaining a distributed database you will need to know when a listing has been deleted from the database.

Please use the following information:

- You must use the Deletions Resource
- You must specify the LISTINGS TableName
- You may use the DeletionTimestamp to specify a date range for deleted listings
- To check for individual listings, you may use UniversalKey (UniversalKey is equivalent to the ListingKey from the Property Resource)
- You should specify SchemaShortName=LIS for better performance when using a date range

### **Sample DMQL searching for a single listing:**

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=\(TableName=LISTINGS\),\(UniversalKey=10348285951\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=(TableName=LISTINGS),(UniversalKey=10348285951))

### **Sample DQML searching for all deleted listings within a date range:**

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=\(SchemaShortName=LIS\),\(TableName=LISTINGS\),\(DeletionTimestamp=2009-07-01T12:23:01%2b\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=(SchemaShortName=LIS),(TableName=LISTINGS),(DeletionTimestamp=2009-07-01T12:23:01%2b))

To search for deleted Media please use the following:

- You must use the Deletions Resource
- You must specify the MEDIA TableName
- You may use the DeletionTimestamp to specify a date range for deleted media
- To check for individual media records, you may use UniversalKey (UniversalKey is equivalent to the PropMediaKey from the Media Resource)

### **Sample DQML searching for all deleted media records within a date range:**

[http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=\(DeletionTimestamp=2008-06-23T17:34:02-2008-06-24T17:34:02\),\(TableName=MEDIA\)](http://ptest.mris.com:6103/ptest/search?QueryType=DMQL2&StandardNames=0&Count=0&Format=COMPACT-DECODED&SearchType=Deletions&Class=DELETIONS&Query=(DeletionTimestamp=2008-06-23T17:34:02-2008-06-24T17:34:02),(TableName=MEDIA))

## 8. CARETS DOM/CDOM Field Names and Definitions

CARETS Field Name	Definition
DOM	DOM is the time period between the original listing date and the date a contract to purchase is signed or goes off the market, less any time the listing was off market. Days On Market is often abbreviated as DOM.
CDOM	CDOM is defined as the total Days on Market (DOM) for ALL listings of a single property that are not closed within a 90 day period.
StatusChangeDate	The calculated DOM at the time of the status change (RawDom) is stored in the listings in addition to the date of the last status change which is the LastStatusChangeDate.
CDOMCalculatedDate	Each time the contents of a listing transaction change, the CDOM is recalculated. When the CDOM is calculated, the calculated CDOM at the time of the status change (RawCDOM) is stored in the listing in addition to the date of the last status change which is the CDOMCalculatedDate.
RawDOM	The calculated DOM at the time of the status change
RawCDOM	Each time the contents of a listing transaction change, the CDOM is recalculated. When the CDOM is calculated, the calculated CDOM at the time of the status change (RawCDOM) is stored in the listing in addition to the date of the last status change which is the CDOMCalculatedDate.
DOMUpdateYN	When this flag is TRUE (1), it is expected that the DOM displayed to a user is RawDom + (Current Date – LastStatusChangeDate). In all other cases, the display should be RawDom.
CDOMUpdateYN	When this flag is TRUE (1), it is expected that the DOM displayed to a user is RawDomProperty + (Current Date – LastStatusChangeDate). In all other cases, the display should be RawDomProperty.

## 9. CARETS Technical Contact Information

### Combined LA/Westside MLS (CLAW)

822 S. Robertson Blvd., Suite 202  
Los Angeles, CA 90035  
**Primary Contact:** Daniel Ortega  
**Bus Phone:** (310) 358-1100 xt 133  
**Email:** [danielortega@themls](mailto:danielortega@themls)  
**URL:** <http://www.themls.com>

### CRIS Net MLS

7232 Balboa Blvd  
Van Nuys, CA 91406  
**Primary Contact:** Heidi Harper  
**Bus Phone:** (818) 947-2278  
**Fax:** (818) 779-7080  
**Email:** [Heidih@srar.com](mailto:Heidih@srar.com)  
**URL:** <http://www.srar.com>

### Desert Area MLS (DAMLS)

44475 Monterey Ave  
Palm Desert, CA 92260  
**Primary Contact:** Paul Potter  
**Bus Phone:** (760) 346-5637  
**Email:** [paul@cdaronline.com](mailto:paul@cdaronline.com)  
**URL:** <http://www.caldesertrealtors.com>

### i-Tech MLS

1070 E. Green Street, Suite 100  
Pasadena, CA 91106  
**Primary Contact:** Rick Stever  
**Bus Phone:** (626) 447-5731  
**Email:** [Rick@TheStevens.com](mailto:Rick@TheStevens.com)  
**URL:** <http://www.pfar.org>

### Multi Regional MLS (MRMLS)

3201 Temple Ave., Suite 250  
Pomona, CA 91768  
**Primary Contact:** Rob Larson  
**Bus Phone:** (909) 859-2055  
**Email:** [rob@imrmls.com](mailto:rob@imrmls.com)  
**URL:** <http://imrmls.com>

### **Southern California MLS (SoCal MLS)**

101 East Lincoln Blvd., Suite 120  
Anaheim, CA 92805

**Primary Contact:** Zach Garcia

**Bus Phone:** (714) 517-9059

**Fax:** (714) 517-2604

**Email:** [Zach@socalmls.com](mailto:Zach@socalmls.com)

**URL:** <http://www.socalmls.com>

### **Ventura County Regional Data Share (VCRDS)**

463 Pennsfield Place, Suite 102  
Thousand Oaks, CA 91360

**Primary Contact:** Ginny Greer

**Bus Phone:** (805) 495-1331

**Fax:** (805) 379-3641

**Email:** [mls@cvar.com](mailto:mls@cvar.com)

**URL:** <http://www.cvar.com>



## Appendix A – IDX/Public Publication Guidelines

The grid below defines what fields must be omitted from an IDX/Public Feed.

CARETS Field SystemName	IDX/Public Restrictions	Comment
<b>Compensation/Service Information</b>		
SellingOfficeCompensation	Omit Always	
SellingOfficeCompensationType	Omit Always	
VariableRateCompensationYN	Omit Always	
ServiceLevel		Should not be displayed by IDX Vendors. For web site functionality only.
RenewalCompensation	Omit Always	
ListingAgreement		Should not be displayed by IDX Vendors. For web site functionality only.
<b>Showing Information</b>		
ShowInformationRemarks	Omit Always	
LockBoxLocation	Omit Always	
LockBoxType	Omit Always	
PhoneToShow	Omit Always	
ShowingOfficePhone	Omit Always	
ShowingAccess	Omit Always	
IsSignOnPropertyYN	Omit Always	
<b>Owner/Occupant</b>		
OccupantType	Omit Always	
OccupantName	Omit Always	
OccupantPhoneExt	Omit Always	
OtherPhoneDescription	Omit Always	
OtherPhoneNumber	Omit Always	
OtherPhoneExt	Omit Always	
<b>Remarks</b>		
MemberRemarks	Omit Always	
<b>Address</b>		
FullStreetAddress	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
StreetNumber	Include with CIDX-PUB; Omit with CIDX-PNA or CIDX-PNS	Refers to lookup field ListingSubscriptionClassList
StreetNumberModifier	Include with CIDX-PUB; Omit with CIDX-PNA or CIDX-PNS	Refers to lookup field ListingSubscriptionClassList
StreetDirPrefix	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
StreetName	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
StreetSuffix	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList

StreetDirSuffix	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
UnitNumber	Include with CIDX-PUB; Omit with CIDX-PNA or CIDX-PNS	Refers to lookup field ListingSubscriptionClassList
Longitude	Include with CIDX-PUB; Omit with CIDX-PNA or CIDX-PNS	Refers to lookup field ListingSubscriptionClassList
Latatude,	Include with CIDX-PUB; Omit with CIDX-PNA or CIDX-PNS	Refers to lookup field ListingSubscriptionClassList
ThomasGuideMapPage	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
ThomasMapXLetter	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
ThomasMapYNumber	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
ThomasGuideFullMapString	Include with CIDX-PUB; Omit with CIDX-PNA	Refers to lookup field ListingSubscriptionClassList
<b>Transaction /Status</b>		
ListingStatus	IDX/Public feeds can only display those listings with a CARETSListingStatus of: Active, Back Up Offer, Pending, Closed (e.g. Sale or Lease)	
ExpirationDate, ProjectedActivationDate, WithdrawnDate	Omit Always	
PendingPrice	Omit Always	
ClosePrice	Omit Always - Unless CARETSListinStatus is Closed / Leased	
ListingEntryDate	Omit Always	
<b>Agent Information in Property Resource</b>		
xxxxAgentPrimaryAssociation	Omit Always	xxxx = List, Alt
<b>Agent Resource</b>		
AgentCarrierRoute	Omit Always	
AgentCityName	Omit Always	
AgentEditor	Omit Always	
AgentExternalSystemID		Should not be displayed by IDX Vendors For web site functionality only.
AgentFullStreetAddress	Omit Always	
AgentLastLoginDate	Omit Always	
AgentLicenseNumber		Should not be displayed by IDX Vendors For web site functionality only.
AgentPassword	Omit Always	
AgentPostalCode	Omit Always	
AgentRecordComments	Omit Always	

AgentState	Omit Always	
AgentStreetAdditionalInfo	Omit Always	
AgentZip4	Omit Always	
NRDSMemberID	Omit Always	
<b>Office Resource</b>		
AgentOfficeID	Omit Always	-
NRDSCorporateLicenseID	Omit Always	
NRDSFranchiseID	Omit Always	
NRDSOfficeID	Omit Always	
NRDSParentCompanyID	Omit Always	
OfficeAddressAdditionalInfo	Omit Always	
OfficeCarrierRoute	Omit Always	
OfficeCity	Omit Always	
OfficeCityName	Omit Always	
OfficeCounty	Omit Always	
OfficeEditor	Omit Always	
OfficeFullStreetAddress	Omit Always	
OfficeKey	Omit Always	
OfficeName	Omit Always	-
OfficePostalCode	Omit Always	
OfficeRecordComments	Omit Always	
OfficeResponsibleMemberID	Omit Always	
OfficeSourceCreationTimestamp	Omit Always	
OfficeSourceModificationTimestamp	Omit Always	
OfficeState	Omit Always	
OfficeZip4	Omit Always	

## Appendix B - Sample Login/Logout trace:

### --Send--

GET /ptest/login HTTP/1.1  
Method: GET  
User-Agent: MRIS API Testing Tool/2.0  
Rets-Version: RETS/1.7  
Accept: \*/\*  
Host: ptest.mris.com:6103  
Connection: keep-alive

### --Receive--

HTTP/1.1 401 Unauthorized  
Cache-Control: private  
Date: Thu, 03 Jul 2008 15:35:52 GMT  
Content-Length: 0  
Content-Type: text/html  
WWW-Authenticate: Digest realm="users@mris.com", nonce="313231353039393335323834322059afc58aa05e5e474c5a3ac2d08a3203", opaque="6e6f742075736564"  
RETS-Version: RETS/1.0  
X-Powered-By: Servlet/2.4 JSP/2.0

### --Send--

GET /ptest/login HTTP/1.1  
Method: GET  
User-Agent: MRIS API Testing Tool/2.0  
Rets-Version: RETS/1.7  
Accept: \*/\*  
Authorization: Digest username="username", realm="users@mris.com", nonce="313231353039393335323834322059afc58aa05e5e474c5a3ac2d08a3203", uri="/ptest/login", response="7c57446a3444ca52918a5f9238732c5a", algorithm="MD5", opaque="6e6f742075736564"  
Host: ptest.mris.com:6103  
Connection: keep-alive



**--Receive--**

HTTP/1.1 200 OK  
Cache-Control: no-cache="set-cookie"  
Date: Thu, 03 Jul 2008 15:35:52 GMT  
Transfer-Encoding: chunked  
Content-Type: text/plain  
Set-Cookie: RETS-Session-ID=10009871014  
RETS-Version: RETS/1.0  
X-Powered-By: Servlet/2.4 JSP/2.0

<RETS ReplyCode="0" ReplyText="V2.3.1 545: Success">  
MemberName=Test User  
User=3000507, Agent:Association Member Primary:Realtor/Shareholder:Status Active:System-CARETS:Broker  
Office:Office:Residential:Update, 50  
Broker=CRTS,998  
MetadataVersion=1.3.0  
MinMetadataVersion=1.1.1  
OfficeList=CRTS,998  
TimeoutSeconds=1800  
Search=http://ptest.mris.com:6103/ptest/search  
GetObject=http://ptest.mris.com:6103/ptest/getobject  
PostObject=http://ptest.mris.com:6103/ptest/postobject  
Update=http://ptest.mris.com:6103/ptest/update  
Get=http://ptest.mris.com:6103/ptest/get  
Login=http://ptest.mris.com:6103/ptest/login  
Action=http://ptest.mris.com:6103/ptest/get?Command=Message  
GetMetadata=http://ptest.mris.com:6103/ptest/getmetadata  
ChangePassword=http://ptest.mris.com:6103/ptest/changepassword  
Logout=http://ptest.mris.com:6103/ptest/logout  
</RETS>

**--Send-**

GET /ptest/get?Command=Message HTTP/1.1  
Method: GET  
User-Agent: MRIS API Testing Tool/2.0  
Rets-Version: RETS/1.7  
Accept: \*/\*  
Cookie: \$Version=1; RETS-Session-ID=10009871014; \$Path=/  
Authorization: Digest username="3000507", realm="users@mris.com",  
nonce="313231353039393335323834322059afc58aa05e5e474c5a3ac2d08a3203", uri="/ptest/get",  
response="8b5b06a87a3aaf4d3e2fa3e743e30c96", algorithm="MD5", opaque="6e6f742075736564"  
Host: ptest.mris.com:6103  
Connection: keep-alive

**--Receive--**

HTTP/1.1 200 OK  
Cache-Control: no-cache="set-cookie"  
Date: Thu, 03 Jul 2008 15:35:52 GMT  
Transfer-Encoding: chunked  
Content-Type: text/html  
Set-Cookie: RETS-Session-ID=10009871014; Version=1; Path="/"  
Set-Cookie: RETS-Session-ID=10009871014; Version=1; Path="/"  
RETS-Version: RETS/1.0  
X-Powered-By: Servlet/2.4 JSP/2.0

<HTML>  
<HEAD>  
<TITLE>System Messages</TITLE>  
</HEAD>  
<BODY>  
No Messages Found  
</BODY>  
</HTML>

**--Send--**

GET /ptest/logout HTTP/1.1  
Method: GET  
User-Agent: MRIS API Testing Tool/2.0  
Rets-Version: RETS/1.7  
Accept: \*/\*  
Cookie: \$Version=1; RETS-Session-ID=10009871014; \$Path=/  
Authorization: Digest username="3000507", realm="users@mris.com",  
nonce="313231353039393335323834322059afc58aa05e5e474c5a3ac2d08a3203", uri="/ptest/logout",  
response="d751e1d2bcd69ad302bfe923f475e50d", algorithm="MD5", opaque="6e6f742075736564"  
Host: ptest.mris.com:6103  
Connection: keep-alive

**--Receive--**

HTTP/1.1 200 OK  
Cache-Control: no-cache="set-cookie"  
Date: Thu, 03 Jul 2008 15:35:52 GMT  
Transfer-Encoding: chunked  
Content-Type: text/plain  
Set-Cookie: RETS-Session-ID=10009871014; Version=1; Path="/"

RETS-Version: RETS/1.0  
X-Powered-By: Servlet/2.4 JSP/2.0

<RETS ReplyCode="0" ReplyText="V2.3.1 545: Success">  
Goodbye UserName.</RETS>

## Appendix C - CARETS® RETS Supported Features

The following chart contains the table of contents from the RETS 1.0 specification in the left column. The right column shows 'NA' if the item is for reference only, 'Yes' if this feature exists for the CARETS RETS Server, and 'No' if this feature does not exist.

Version 1.0 Real Estate Transaction Specification	CARETS System (managed by MRIS Cornerstone)
<b>1. Introduction</b>	
1.1 Purpose	NA
1.2 Scope	NA
1.3 Requirements	NA
1.4 Terminology	NA
<b>2. Notational Conventions</b>	
2.1 Augmented BNF	NA
2.2 Typographic Conventions	NA
2.3 Rules	NA
2.3 Atoms and Primitive Entities	NA
<b>3. Message Format</b>	Yes
3.1 General Message Format	Yes
3.2 Request Format	Yes
3.3 Header Field Format	Yes
3.4 Required Client Request Header Fields	Yes
3.5 Optional Client Request Header Fields	Yes
3.6 Response Format	Yes
3.7 Required Server Response Header Fields	Yes
3.8 Optional Server Response Header Fields	No
3.9 General Status Codes	Yes
<b>4. Login Transaction</b>	Note: Digest Authentication only supported.
4.1 Security	Digest Authentication
4.2 Digest Algorithm	Yes
4.3 WWW-Authentication Response Header	Yes
4.4 The Authorization Request Header	Yes
4.5 Authorization Example	NA
4.6 Required Request Arguments	None required
4.7 Optional Request Arguments	Yes
4.7.1 BrokerCode Tag	Yes
4.8 Optional Response Header Fields	Yes
4.9 Login Response Body Format	Yes
4.10 Required Response Arguments	Yes
4.10.1 Broker Tag	Yes
4.10.2 Member Name Tag	Yes
4.10.3 Metadata Version Tags	Yes
4.10.4 User information tag	Yes
4.10.5 Capability URL List	Yes
4.11 Optional Response Arguments	Yes
4.11.1 Accounting Tags	No
4.11.2 Access Control Tags	Yes
4.11.3 Office List Tag	Yes
4.12 Well-Known Names	NA
4.13 Capability URL List	Yes
4.14 Reply Codes	Yes
<b>5. GetObject Transaction</b>	Yes
5.1 Required Client Request Header Fields	Yes
5.2 Optional Client Request Header Fields	Yes
5.3 Required Request Arguments	Yes
5.4 Optional Request Arguments	Yes

5.4.1 Location	Yes
5.5 Required Server Response Header Fields	Yes
5.6 Optional Server Response Header Fields	Yes
5.6.1 Location	Yes
5.6.2 Description	Yes
5.7 Required Response Arguments	Yes
5.8 Optional Response Arguments	Yes
5.9 Metadata	Yes
5.10 Resources	Yes
5.11 Multipart Responses	Yes
5.12 Reply Codes	Yes
<b>6. Logout Transaction</b>	Yes
6.1 Required Request Arguments	None required
6.2 Optional Request Arguments	None required
6.3 Required Response Arguments	None required
6.4 Optional Response Arguments	Yes – billing not supported
<b>7. Search Transaction</b>	Yes
7.1 Search Types	Yes
Agent	By role
History	Yes
Media	Yes
Office	By role
OpenHouse	Yes
Property	Yes
Prospect	Future
Tax	Yes
Tour	Yes
7.2 Search Terminology	Yes
7.2.1 Field Delimiter	Yes
7.2.2 Field Name	Yes
7.2.3 Record Count	Yes
7.2.4 Other terms	Yes
7.3 Required Request Arguments	Yes
7.3.1 Search Type and Class	Yes
7.3.2 Query Specification	Yes
7.4 Optional Request Arguments	Yes
7.4.1 Count	Yes
7.4.2 Format	Yes
7.4.3 Limit	Yes
7.4.4 Offset	Yes
7.4.5 Select	Yes
7.4.6 Restricted Indicator	Future
7.4.7 StandardNames	Yes
7.5 Required Response Arguments	None required
7.6 Search Response Body Format	Yes – all
7.7 Query language	Yes
7.8 Reply Codes	Yes
<b>8. Get Transaction</b>	Yes - Note: Returned in Login URL for Action response.
8.1 Required Request Arguments	None required
8.2 Optional Request Arguments	None required
8.3 Required Response Arguments	None required
8.4 Optional Response Arguments	None required
8.5 Status Conditions	Yes
<b>9. Change Password Transaction</b>	Yes - Specification broken - Needs re-working.
9.1 Required Request Arguments	Yes
9.2 Optional Request Arguments	None required
9.3 Required Response Arguments	Yes
9.4 Optional Response Arguments	None required
9.5 Reply Codes	Yes
9.6 Encryption Key Construction	Yes
<b>10. Update Transaction</b>	Yes



<b>11. Metadata Format</b>	Yes
11.4.2 Lookup	Yes
11.4.3 LookupType	Yes
11 RETS-X LookupTypeByParent	Yes – RETS Extension
<b>12. GetMetadata Transaction</b>	Yes
12.1 Required Client Header Fields	Yes
12.2 Required Request Arguments	Yes
12.3 Optional Request Arguments	Yes
12.4 Required Server Response Header Fields	Yes
12.5 Optional Server Response Header Fields	Yes
12.6 Required Response Arguments	None required
12.7 Optional Response Arguments	None required
12.8 Metadata Response Body Format	Yes
12.9 Metadata	Yes
12.10 Reply Codes	Yes
<b>13. Compact Data Format</b>	Yes
13.1 Overall format	Yes
13.2 Decoded Format	Yes
13.3 Transmission standards	Yes
<b>14. Session Protocol</b>	Yes
14.1 Connection Establishment	Yes
14.2 Authorization	Digest Authentication supported
14.3 Session	Yes
14.4 Termination	Yes

## Appendix D - Deletions Resource FAQ

**Q: What are the most common TableName's to query:**

A: LISTING\_SUBSCRIPTIONS (roles)  
 MEDIA (media)  
 PARTIES (agent & office)  
 PARTY\_ROLES (roles)  
 PROFESSIONAL\_RELATIONSHIPS (roles)  
 PROPERTY\_RELATIONSHIPS (roles)  
 LISTINGS (Property)

Note: The role type tables are not relevant to maintaining a distributed database.

**Q: What does SchemaShortName mean?**

A: The Schema name could be one of the following, LIS (Listing), SYS (System) and PUB (Public Records) which is not currently implemented for your system. You're going to be interested only in **LIS** at this time.

**Q: What does UniversalKey mean?**

A: UniversalKey is the value for the corresponding TableName (see below):

Resource	TableName	SchemaShortName	UniversalKey
Agent	PARTIES	LIS	AgentKey
Property	LISTING	LIS	ListingKey
Media	MEDIA	LIS	PropMediaKey
Office	PARTIES	LIS	OfficeKey

## Appendix E – Timed Downloads FAQ

Safe Data Download Algorithm to ensure data is not missed:

- 1.) Recommended Data Pull Interval - 10 Minutes.
- 2.) Record time of initial full bulk load data pull.
- 3.) Start incremental data pulls from time of initial full bulk load less 1 hour to be safe.
- 4.) During first incremental pull, record MAX(ModificationTimestamp) from Cornerstone that was received in data pull.
- 5.) Subtract the run time of your data pull process and subtract an additional minute (the additional minute is overkill, but you need to subtract some small increment of time to account for the possibility of Oracle read-consistency putting in a timestamp just before the time you started your run). Let's call this NextRunTms. Remember that this comes back to you in PDT/PST.
- 6.) The next data pull would be  
Query=(ModificationTimestamp=NextRunTms+) - Remember to convert NextRunTms to GMT as per the RETS specification.
- 7.) Do same as step (4) onwards.

This may result in you pulling identical data for several 10 minute timeslices during very low update periods, but that is not really a problem as this will be, at most a few records.

During Switch to/from Daylight Savings Time - Recommendation: Turn off data loading process from 30 minutes before to one hour and 30 minutes after time switch in your timezone.