

RESO

Application Interoperability Best Practices Document

December 2, 2013

Section 1 - Introduction

This interoperability best-practice document recommends how real estate software should interoperate and integrate with MLS systems and other software. The standard includes two core specifications: (1) Links and Attributes for Passing Information between Applications; and (2) Application Metadata.

A 'best-practice' standard guide is a recommendation from RESO but it is not considered mandatory to be RESO compliant.

Section 2 - Links and Attributes for Passing Information Between Applications

This document outlines specific use cases for passing information between applications by hyperlink with attributes. The common use cases are passing listings, saved searches, and contact information from one application to another. Please add below other use cases and examples of which you are aware and would like to see part of the standard.

Any ID shown below should correspond to the ID for the listing as issued from that MLS's RETS server, whether using traditional RETS or a future REST API version of RETS. Listing IDs shown in examples below use the vendor-independent listing ID format.

Data to be passed	Parameter to append to LaunchUri	Example: http://myapp.com/ as an app's LaunchUri
Listing IDs	LaunchListingIds= followed by a comma separated list of listing IDs	http://myapp.com/?LaunchListingIds=ListingId [, ListingId(0-n) [, ListingSource]
Listing search	LaunchListingSearch= followed by one or more queries that use the syntax of the future RETS API query language.	http://myapp.com/?LaunchListingSearch=TotalBedrooms%20Gt%204

Saved Search IDs	LaunchSavedSearchIds = followed by a comma separated list of saved search IDs	http://myapp.com/?LaunchSavedSearchIds=20060412165917817933000000,20080917142739989238000000
Listing Cart IDs	LaunchListingCartIds= followed by a comma separated list of listing cart IDs	http://myapp.com/?LaunchListingCartIds=20060412165917817933000000,20080917142739989238000000
Contact IDs	LaunchContactIds= followed by a comma separated list of contact IDs	http://myapp.com/?LaunchContactId=20060412165917817933000000,20080917142739989238000000
Contact search	LaunchContactSearch= followed by one or more queries that use the syntax of the future RETS API query language.	http://myapp.com/?LaunchContactSearch=FirstName%20Eq%20'Bill%25'

If there are no items (listings, searches, etc.) with which to launch an app, the LaunchUri should be opened as-is. If there is at least one item with which to launch an app, the application being launched must accept either a GET or POST request. If the launch URI with parameters is longer than the browser can accept (typically around 2,000 characters for IE), then a POST request must be issued. If the launch URI with parameters is shorter than the browser can accept, then a GET request must be issued.

Section 3 - Application Metadata

The following JSON is a proposed format for describing a software application and its capabilities. This document will be published by the developer, an application store provider, or other service that offers the application to end users.

Section 4 - Application Data

- Product Name
- Company Name
- Version
- Description
- Support type and contact info (Support URI, Hours, Email, and Phone)
- App categories (CMA, CRM, Forms, Maps, Stats, etc.) A proposal on standardization of app categories is below.
- URIs to the application icon in multiple sizes (sizes TBD)
- Launch URI
- App launch categories - an array of zero or more records showing that the application may optionally be launched with any of the following:
 1. Launch categories - one of the following:
 - Listing IDs (using a proposed vendor-independent listing ID format described in a section below or a proprietary ID format)
 - Listing search (using the future RESO-standard API query language)
 - Saved search IDs (using a vendor-specific ID format)
 - Listing cart IDs (using a vendor-specific ID format)
 - Contact IDs (using a vendor-specific ID format)
 - Contact search (using the future RESO-standard API query language)
 2. Launch category quantity: an optional integer, indicating how many items an application may be launched with, with a blank value meaning unlimited. For example, a CMA app may be best launched with only a single listing, whereas a report printing app could be launched with multiple listings with good results. Other applications have maximum number of items they may be launched with.

Section 5 Document Details

Sample Response:

```
{
  "D": {
    "Success": true,
    "Results": [
      {
        "ProductName": "Cloud CMA",
        "CompanyName": "WR Studios",
        "Version": "3.7",
        "Description": "Cloud CMA is .....",
        "SupportedBy": "Developer",
        "SupportURI": "http://cloudcma.com/support",
        "SupportPhone": "123-456-7890 ext 123",
        "SupportEmail": "support@cloudcma.com",
        "AppCategories": ["CMA"],
        "LaunchUri": "http://cloudcma.com/spark",
        "LaunchCategories": [
          {
            "Category": "ListingId",
            "Quantity": 6,
          }
        ]
      }
    ],
    "LastModified": "2010-11-22T20:09:37Z"
  }
}
```

Section 6 App Categories

Establishing a standard list of application categories is a part of this proposal as well. The proposed category list is as follows:

- IDX
- Mobile
- CMA
- CRM
- Search
- Photos
- Statistics
- Flyers
- Prospecting
- Maps
- VOW or Customer Portal
- Brokerage Management
- Accounting
- Social Networking
- Showing Scheduling
- Public Records
- Virtual Tours
- Video
- Web Sites
- Forms
- Transaction Management
- Document Management
- Electronic Signatures
- Training
- Syndication
- Other

Section 7 - Standardization of API Listing IDs

To avoid incompatible vendor-specific listing IDs, this proposal also includes an approach for vendor-independent listing IDs.

For listings, the ID is a concatenation of an MLS abbreviation followed by a hyphen and the list number. For example, if a fictitious MLS called “ABC MLS” has an abbreviation of “ABC”, the ID for list number 5817542 would be ABC-5817542. This will require a single, comprehensive list of unique, standardized MLS abbreviations to be agreed upon and used by everyone involved. This approach creates IDs that are vendor-independent while facilitating MLS data sharing scenarios. For example, if someone wishes to create a CMA based on three listings with each listing in a different MLS, this approach to generation of IDs allows those three IDs to be passed to an application without requiring multiple fields to be passed to identify a listing or IDs that are proprietary to specific vendors.

Proprietary listing IDs may be used as well.