Merlin DAM
API Options Overview

Prepared by: Andrew Forber, CTO

File Name: APIs Overview (YYYYMMDD).docx

Revision Date

1.0 November 27, 2013 Initial Release

1.1 January 21, 2014 Revised – more complete description of SODA capabilities

1.2 November 11, 2016 Merlin Upload API added and other descriptions updated.

1.3 August 15, 2018 Merlin SDK

1.4 October 2, 2019 New Letterhead

1.5 October 3, 2019 RSS deprecated

1.6 October 4, 2019 AEF edits

Distribution:

Internal and external – product information for technically advanced clients and prospects

[Introduction 4](#_Toc523918285)

[1. Tier 1 APIs 4](#_Toc523918286)

[1.1 OpenSearch 4](#_Toc523918287)

[1.2 Simple Object Delivery API - SODA 4](#_Toc523918288)

[1.3 Merlin Upload API 5](#_Toc523918289)

[2. Tier 2 - The MerlinX API 5](#_Toc523918290)

[3. MerlinOne SDK 6](#_Toc523918291)

# Introduction

There are several methods of interfacing external programs to Merlin. Which method you choose will depend on a number of factors:

* How complex an interaction with Merlin you require. Are you just searching and returning results and files? Does your application need to update the database?
* What level of effort and expertise do you have available to build your interface? Some of the options are simpler than others to program.
* Is your client platform able to support rich applications?

Access to and use of any of these APIs is subject to licensing, and to non-disclosure agreements between the customer and MerlinOne Inc.

# Tier 1 APIs

The combination of the following components allows a developer to search, retrieve metadata as well as thumbnail/high resolution files pertaining to specific objects. The Upload API is allows asset files to be input to Merlin.

## OpenSearch

OpenSearch 1.0 is a recognized standard protocol for providing search capabilities to external systems. OpenSearch clients exist in many current commercial software packages. OpenSearch server interfaces are provided by most Internet search engines, including Google, Yahoo and Bing.

Merlin provides an optional OpenSearch server implementation as part of the Merlin web UI support software. It can be used to allow external programs which implement the client specification to search within Merlin and retrieve metadata.

OpenSearch can only search the main Merlin catalog. For example, it does not have an awareness of Collections or saved searches. It allows you to search by Asset ID number, date range, and a range of sophisticated full-text-search capabilities including field-level searches, case-sensitivity, word proximity, phrases, Boolean operators, and wildcards.

OpenSearch is also extensible by XML namespacing, and using WebMaster’s plug-in mechanism to customize search behavior. It can be configured to search anything which is searchable in the Merlin client, and return any metadata which is in the database. It delivers results in a fixed XML format which can be parsed using conventional XML parsing libraries RSS and ATOM formats are not currently supported.

OpenSearch is “RESTful” web API delivering results over HTTP. It is useful if you only want to search for objects within Merlin and return metadata about them, or in conjunction with the Simple Object Delivery API (SODA).

## Simple Object Delivery API - SODA

The Simple Object Delivery API provides a way to download assets discovered using OpenSearch. When the feature is enabled for a given profile, the OpenSearch API will deliver an encrypted URL which, when issued with a GET back to the server, will return the high-resolution file or the thumbnail files for the search result. Optionally, the API will deliver the object to a separate server target, which existing Merlin routing infrastructure can then route to other locations via FTP. If the object is an image file or some other type with embeddable metadata then the file is delivered with metadata updated from the database (i.e. edits within Merlin are reflected in the output). Cropping,resolution and file format modifications for image files are also available. The output of the item using this method is fully logged in the Merlin database just like any other output.

SODA can gives your application access to the following:

* Permanent links to large and small thumbnail files for company-internal applications. These are not recommended for embedding in emails or public-facing web pages.
* Temporary links to large and small thumbnails. Temporary links are encrypted (using strong AES encryption) and will stop working some interval, chosen by you, after the link is delivered to your application. The encrypted links to can be set to expire 1 or more days after the link is created, after which SODA returns a “Link expired” result. This is the preferred method for embedding public links from to Internet-accessible Merlin in web pages. For quasi-permanent links you can have Opensearch provide an expiration time of many years.
* Temporary links to high resolution files which will return the file in the HTTP reply, or for delivery to server targets on the Merlin system. The expiry time on these links can be set in 1 second increments. The URL can be formatted to deliver the file to the delivery folder either synchronously or asynchronously. Image files (usually JPEG and TIFF) are rotated upright, and your HTTP client may add parameters to the link to crop and/or resize the image. For most image formats you can request delivery in a different format, for example converting Camera RAW to JPEG. For downloads, you have the option of requesting an Attachment or Inline view of the incoming data, and can specify the destination filename.

Like OpenSearch, SODA is RESTful, and does not require clients to manage session data. There is no authentication required: all of the security is managed by OpenSearch when it creates the encrypted URL.

## Merlin Upload API

The Merlin Upload API is a simple RESTful web service provider which you can use to upload files to Merlin. It requires you to write a simple web service client app.

Upload API uses HTTPS as the transport layer. You can use the API to upload any file type to a selected MIO queue, with metadata. For large files (over 2GB) your web service client app can upload files in segments for reassembly when all segments have arrived on the server. Authentication is via an API key included as a parameter in the HTTPS POST data. Object metadata is accepted for Merlin database fields which are configured on the server side, to protect against accidental modification of other fields. The data is in HTTP POST *multipart/form-data* format. The response to the API request is a message confirming that the file has been delivered successfully to the server. The product includes a simple example client program written in C#.

# Tier 2 - The MerlinX API

This is the web service API on which the Merlin X client is based. With this API it is possible for a customer with programming expertise to perform all operations that the Merlin X full-featured web client performs, including searching, output, data editing and version control.

The Merlin X API is based on HTTPS POST operations, and exchanges data using JSON as the language describing requests and responses. Unlike OpenSearch, SODA, and the Upload API, it is a “stateful” interface which maintains user state on the server. Client programs to operate with that context by logging in and logging out of sessions. Client programs make parameterized requests to “controllers” on the web server which perform actions like outputs, metadata queries and metadata updates. Requests for things like searches against full text or relational metadata fields are interpreted and passed to the search engine by the server software, so complex operations have simple interfaces. High level functions like “check in” and “check out” for version control and advanced search methods are also abstracted on the server.

All access to the features of Merlin is abstracted by this API: direct interaction with the database (e.g. SQL-language access) is not supported.

Although it requires more work to program, the MerlinX API allows you to implement all aspects of a DAM client. Documentation for the API consists of an overview document and a live documentation and test user interface built into the API.

# MerlinOne CMS SDK

Using the documentation and example code provided by our SDK kit you will be able to embed a version of the MX user interface in another web-based “host” application. We currently have implementations of some CMS integrations using this method, including WordPress and Sitecore. The SDK gives users access to the most commonly used functions of the full MX DAM. They can:

* Perform searches.
* Select assets to work with.
* Interact with MerlinX collections (sometimes referred to as light boxes or projects).
* Push or pull files to transfer them from the MX system to the host system.

The MX Embedded application incorporates the following components:

* The MX Javascript framework, implemented by a few dozen class modules.
* A handful of open source Javascript libraries including jQuery, RequireJS, Moment (date utilities), jQote (template engine), Growl (notifications) and several others. The functionality any of these foundation libraries may be utilized directly by plug-ins or indirectly through MX framework classes.
* Font Awesome for icons. https://fontawesome.com/v4.7.0/

Use of this method of integration uses a separate Merlin API called MXChange, which you must call from the code running on your CMS (or similar system) server. This type of integration requires programming expertise in the CMS system on both the server and client sides. Based on our experience, it allows you to integrate a fully featured Merlin integration into your web-based application with one to two business weeks of development effort.