

How To: Use the XML Lookup Utility

The Ed-Fi Data Standard allows two alternative values for reference types: *lookup types* and *identity types*. The Ed-Fi Bulk Load API and the [Console Bulk Loader](#) require that identity type references always be used. The XML Lookup Utility is a command-line application which resolves lookup references to identity references in Ed-Fi XML documents. The utility is suitable for interactive or scripted use as part of a larger data-ingestion process.

The utility uses Ed-Fi Standard XSD files (with extensions) combined with a functional Ed-Fi ODS / API to add identity type information to reference types that only contain lookup types, thus allowing the Bulk Load API and the Console Bulk Loader to ingest the files. It does not need to be recompiled when the Ed-Fi standard or extensions are changed, as it dynamically maps the Ed-Fi Standard XSD elements to the Ed-Fi REST API endpoints documented in the Swagger metadata.

When the utility is run against one or more XML documents, copies of the the original XML documents with the resolved identity types are created. The original XML documents are left unchanged by the utility.

The utility source code can be found in the Ed-Fi-ODS repository in the \Utilities\DataLoading\EdFi.XmlLookup.Console directory. It is part of the LoadTools solution (\Utilities\DataLoading). The LoadTools solution can be compiled using Visual Studio 2013 or 2015 (Community Edition or better).

Command-Line and Configuration Parameters

The utility accepts command-line parameters as well as values in the EdFi.XmlLookup.Console.config file, the following command line parameters

	Parameter	.config Setting	Description	Default	Example Parameter Value
/?	Help		Displays help information		/?
/d	Data Folder	DataFolder	The location of the XML files	Working directory of the utility	/d "c:\data"
/a	API URL	ApiUrl	The Ed-Fi ODS API base URL		/a "http://server/api/v2.0"
/y	School Year	SchoolYear	The School year for the ODS API (Appended to API URL)		/y "2017"
/p	Profile		If provided, the profile to use when calling the API		/p read_only_profile
/o	OAuth URL	OAuthUrl	The OAuth URL for retrieving a session token		/o "http://server/oauth"
/k	OAuth Key	OAuthKey	The application key for the session		/k myKey
/s	OAuth Secret	OAuthSecret	The secret associated with the key for the session		/s mySecret
/m	OpenAPI Metadata URL	SwaggerUrl	The OpenAPI (Swagger) base URL		/m "http://server/swagger"
/f	Force Metadata Reload		If provided, it forces a reload of the metadata	Do not force reload	/f
/w	Working Folder	WorkingFolder	A read/write directory to store the metadata information	Working directory of the utility	/w "c:\temp\lookup"
/x	XSD Folder	XsdFolder	The directory containing the Ed-Fi Standard XSD files	the /Schema directory under the Working Folder	/x "c:\ed-fi\standard"

Operation

The XML Lookup Utility requires values for all the configuration values (except for /p and /f) to be present either in the .config file, or as command-line parameters.

Existing identity type values are never overwritten. Where identity types can be inferred directly from lookup types, that conversion happens first. Identical lookup type values within a document are resolved against the API call only once. Lookup type values may be reused several and possibly hundreds of times instead of being resolved from the API every time. Memory use is kept at a minimum by scanning XML documents rather than loading the entire document into memory. A session token is only retrieved once when the utility runs.

Processing Overview

When the XML Lookup Utility is run, the following actions occur:

1. The Ed-Fi Standard XML files are read and checked for errors.
2. The OpenAPI data is read from the cache, or a cache built from the provided OpenAPI URL.
3. Mappings are created for:
 - a. Lookup to Identity XML Types.
 - b. Incoming XML Lookup types and GetByExample query parameters.
 - c. GetByExample resources and Identity XML Types.
4. For each XML file in the data directory:
 - a. The file is scanned for reference types.
 - b. If the reference type contains an identity type, it is ignored.
 - c. If the lookup type contains all the fields required for the identity type, an identity type is created without calling the ODS / API.
 - d. If the lookup type has not been resolved, the appropriate GetByExample API call is invoked and an identity type is created.
5. The file is then read again and a copy of the file is written with identity types being injected.

Troubleshooting

Because the ODS Data Standard and the associated REST API is dynamic and extensible, and because security is highly configurable, there are no guarantees that the utility can successfully create every possible lookup-to-identity mapping. However, troubleshooting information is available after the utility runs to assist in resolving some of these issues.

The most typical issues are related to security configuration for the client application associated with the key/secret pair, missing resources, and incomplete or ambiguous lookup elements. A successful lookup returns exactly one resource. Missing or multiple resources are an error.

Security issues can be resolved by changing the settings for the client application in the security administration tools. Lookup failures may also be the result of ambiguous or non-existent data in the ODS / API platform.

Comments in Generated XML Documents

The generated XML documents contain XML comments identifying values that were generated. These comments show which identity values were created, and which lookup values were used to create them, which can be useful in tracing issues in the results.

Logging Messages

As configured by default, the Utility displays logging information on the screen and captures more detailed information in a log file for each run. Generally speaking, all ERROR- and CRITICAL-level issues found in the log file should be resolved before submitting the resolved XML documents for loading.

Logging configuration may be changed by adjusting the .config file appropriately. Refer to the excellent [log4net documentation](#) for detailed instructions on how to configure application logging to suit your exact needs.