Eduction

Software Version 12.3

Programming Guide

MICRO FOCUS

Document Release Date: June 2019
Software Release Date: June 2019

Legal notices

Copyright notice

Documentation updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

You can check for more recent versions of a document through the MySupport portal. Many areas of the portal, including the one for documentation, require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in.

Additionally, if you subscribe to the appropriate product support service, you will receive new or updated editions of documentation. Contact your Micro Focus sales representative for details.

Support

Visit the MySupport portal to access contact information and details about the products, services, and support that Micro Focus offers.

This portal also provides customer self-solve capabilities. It gives you a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the MySupport portal to:

- Search for knowledge documents of interest
- Access product documentation
- View software vulnerability alerts
- Enter into discussions with other software customers
- Download software patches
- Manage software licenses, downloads, and support contracts
- Submit and track service requests
- Contact customer support
- View information about all services that Support offers

Many areas of the portal require you to sign in with a Software Passport. If you need a Passport, you can create one when prompted to sign in. To learn about the different access levels the portal uses, see the Access Levels descriptions.

About this PDF version of online Help

This document is a PDF version of the online Help.

This PDF file is provided so you can easily print multiple topics or read the online Help.

Because this content was originally created to be viewed as online help in a web browser, some topics may not be formatted properly. Some interactive topics may not be present in this PDF version. Those topics can be successfully printed from within the online Help.

Contents

Chapter 1: Introduction

This section introduces Micro Focus Eduction.

Eduction

Eduction identifies and extracts *entities* from text. An entity is a word, phrase, or block of information, such as a person's name, an address, a date, or a telephone number.

Eduction includes a comprehensive set of predefined entities, for many languages and geographical locations, so that you can extract names, credit card numbers, addresses, and so on. You can also extend Eduction by defining your own entities.

You can use Eduction to:

- extract entities from documents, and add them to metadata fields, before the documents are added to your IDOL index. For example, you might extract company names from your document content and tag the documents with these names. In this scenario, the document content might be the body of an e-mail, the content of a PDF file, or text from a web page. Tagging the documents would allow a front-end application to present a list of companies, so that your users can filter their search results based on the companies that are mentioned.
- perform sentiment analysis. Sentiment analysis identifies positive and negative sentiment in text. You could, for example, extract positive and negative comments from product reviews.
- identify personally identifiable information (PII) in your data, so that you can manage this data and conform to regulation such as the General Data Protection Regulation (GDPR).
- redact sensitive information in text or IDOL documents, so that you can conform to data protection standards and use your records for multiple purposes.

There are several ways in which you can use Eduction:

- **CFS and NiFi Ingest**. Eduction is commonly used to enrich documents during the ingestion process, before they are added to the IDOL index. For example, you can extract entities and tag the documents so that is easier to find documents related to a specific person, place, or subject. To run Eduction as part of the ingestion process, use either Connector Framework Server (CFS) or IDOL NiFi Ingest.
- **Eduction Server**. You can use the Eduction Server to extract entities, redact information, and perform sentiment analysis on plain text.
- **Build a custom application using the Eduction SDK**. Micro Focus provides Eduction SDKs for C and Java, so that you can include Eduction in your own applications.

Eduction Concepts

This section introduces some of the Eduction concepts and terminology used throughout this guide.

Grammar Files

A *grammar file* defines one or more entities that you want to extract.

- **Standard grammars**. Eduction includes a collection of grammar files covering common entities such as names, social security numbers, postal addresses, telephone numbers, and so on. For a complete list of standard grammars, see Standard Grammars, on page 92.

  Standard grammar files are licensed by category and by language, so that it is possible to be licensed for any combination of category (for example, sentiment, place, or person) and language.

- **User grammars**. You can extend the capabilities of Eduction by writing your own grammar files, either from scratch or by referencing existing entities.

  To reference the standard grammars in your own grammar files, you must have an appropriate license.

Grammar files are created in XML format, and can be compiled into ECR format. Compiling a grammar file into the ECR format makes it much faster to load at run-time. Most of the standard grammar files are supplied only in ECR format.

Entities can be defined in several ways. You might define a dictionary of possible matches, for example to extract names of people or places. Alternatively, you might specify what a match looks like without having to list each possibility. The latter approach would be suitable for extracting dates and times, or telephone numbers, because these conform to a known pattern.

Entities can be defined recursively, and rules can refer to entities in other grammar files. This allows you to create more complicated entities that match data such as URLs or postal addresses.

Linguistic Sentiment Analysis

The sentiment analysis grammar files contain dictionaries of types of word (for example, positive adjective, negative noun, neutral adverb, and so on), and patterns that describe how to combine these dictionaries to form positive and negative phrases.

For example, you could run sentiment extraction using the English sentiment grammar file (`sentiment_eng.ecr`), with the following hotel review as the input file:

The room was nice enough, with a plug in radiator, tv with an English news channel, hot shower, comfy bed. The receptionist we first dealt with was miserable and rude, and just grunted at us and rolled her eyes because we were too early for check in having just got off the morning train from Khabarovsk. Fortunately, a younger receptionist with a nice smile appeared, spoke to us helpfully suggesting a few cafes nearby to pass some time, and we tried to forget about the other woman. Breakfast is terrible. Unidentifiable cordials, gloomy porridge, bread rolls filled with things you don't expect for breakfast, like potato, egg and dill. Don't come here for the breakfast, but for the cost of the room in a city like Vladivostok, the hotel is still decent value for money.

The following is a sample of the output that this produces:

```
<?xml version="1.0" encoding="UTF-8"?>
  <MATCHLIST>
    <DOCUMENT Type="IDOL IDX" ID="Unknown">
      <FIELD Name="DRECONTENT">
        <FIELD_INSTANCE Value="1">
          <MATCH EntityName="sentiment/positive/eng" Offset="7" OffsetLength="5"
          Score="1.05" NormalizedTextSize="17" NormalizedTextLength="17"
          OriginalTextSize="17" OriginalTextLength="17">
            <ORIGINAL_TEXT>The room was nice</ORIGINAL_TEXT>
            <NORMALIZED_TEXT>The room was nice</NORMALIZED_TEXT>
            <COMPONENTS>
              <COMPONENT Name="TOPIC" Text="The room" Offset="0"
              OffsetLength="0" TextSize="8" TextLength="8"/>
              <COMPONENT Name="SENTIMENT" Text="nice" Offset="13"
              OffsetLength="13" TextSize="4" TextLength="4"/>
            </COMPONENTS>
          </MATCH>
          <MATCH EntityName="sentiment/negative/eng" Offset="494"
          OffsetLength="492" Score="1.2" NormalizedTextSize="21"
          NormalizedTextLength="21" OriginalTextSize="21"
          OriginalTextLength="21">
            <ORIGINAL_TEXT>Breakfast is terrible</ORIGINAL_TEXT>
            <NORMALIZED_TEXT>Breakfast is terrible</ NORMALIZED_TEXT>
            <COMPONENTS>
              <COMPONENT Name="TOPIC" Text="Breakfast" Offset="0"
              OffsetLength="0" TextSize="9" TextLength="9"/>
              <COMPONENT Name="SENTIMENT" Text="terrible" Offset="13"
              OffsetLength="13" TextSize="8" TextLength="8"/>
            </COMPONENTS>
          </MATCH>
```

```
        </FIELD_INSTANCE>
      </FIELD>
    </DOCUMENT>
</MATCHLIST>
```

The following example configuration shows the recommended usage:

```
[Eduction]
ResourceFiles=grammars/sentiment_eng.ecr
// Note: replace sentiment_eng.ecr by sentiment_user_eng.ecr if using user
modification

// standard entities for all sentiment analysis in English:
Entity0=sentiment/positive/eng
Entity1=sentiment/negative/eng
EntityField0=POSITIVE_VIBE
EntityField1=NEGATIVE_VIBE
EntityComponentField0=TOPIC,SENTIMENT
EntityComponentField1=TOPIC,SENTIMENT

// some invalid matches are given very low scores so that we can filter them out:
MinScore=0.1

// for extraction of Twitter handles, hashtags and emoticons:
TangibleCharacters=@#:;

// for displaying metadata:
OutputScores=True
OutputSimpleMatchInfo=False
EnableComponents=True
```

For more information on the sentiment analysis grammar files, how to adjust the sentiment analysis by extending the grammars, and the features that the sentiment grammars support, refer to *IDOL Expert*.

Perform Sentiment Analysis on Short Comments

The standard sentiment analysis grammars are designed for high precision. For some sources of short comment data, such as YouTube comments, no positive or negative matches are found in some documents despite sentiment clearly being expressed.

If recall with the full `sentiment_eng.ecr` grammar file is too low, and your documents are generally short comments, use `sentiment_basic_eng.ecr` to extract additional matches.This grammar contains carefully-selected lists of positive and negative terms that help determine the sentiment of a document in which `sentiment_eng.ecr` found no matches.

`sentiment_basic_eng.ecr` contains terms in title case, but research shows that for most data these impair recall, so these are given a lower score. Micro Focus recommends that you set `EntityMinScore`*`N`* to `0.4` to filter out these terms unless you need them.

`sentiment_basic_eng.ecr` does not expose TOPIC or SENTIMENT components, and does not use scores to reflect strength or reliability of polarity. The following additional example configuration shows the recommended usage:

```
[Eduction]
ResourceFiles=grammars/sentiment_eng.ecr,grammars/sentiment_basic_eng.ecr
// optional further layer of analysis for very short documents:
Entity2=sentiment/basic_positive/eng
Entity3=sentiment/basic_negative/eng
EntityField2=BASIC_POSITIVE_VIBE
EntityField3=BASIC_NEGATIVE_VIBE
// remove this setting to include basic matches in titlecase - this is not
recommended because on most data it decreases precision:
EntityMinScore2=0.4
EntityMinScore3=0.4
```

Components

Some of the standard grammar files contain *components*, which enable you to extract attributes from matched phrases, such as topic, subject, and positive or negative sentiments. The attributes are called components because they are the components of a single match.

For example, if you used sentiment analysis to match the phrase *Their service is fantastic* as conveying positive sentiment, you can then use components to identify *service* as the subject matter, and *fantastic* as the adjective that describes the subject (note that the sentiment is not necessarily an adjective in all cases). You can also set up components when you write your own custom grammar files.

> **NOTE:**
> `sentiment_basic_eng.ecr` does not support the TOPIC or SENTIMENT components.

For more information on how to configure and define components in your grammar files, and when to use them, refer to *IDOL Expert*.

Related Topics

- EntityComponentFieldN, on page 67

Extraction

Eduction extracts entities from documents based on the rules you have created in your dictionaries and grammars. Eduction can output files in multiple formats using `edktool`.

For each nominated field in a document, Eduction identifies each instance of the requested entity. Eduction returns an XML list of matches, or adds the matches to the source document as new fields. Eduction can also identify components of an entity match, such as:

- the parts of a social security number or phone number.
- a confidence score for the accuracy of the match.

Results Relevance

Eduction returns entities based on the extraction rules from the grammars and dictionaries. Eduction provides a test mode to measure extraction relevance *precision* and *recall*. Precision and recall are

based on the comparison between human-marked results and engine-marked results. The following terms describe result relevance as used in Eduction.

- **True Positives** (TP). Human-marked results that are also marked by the engine. These results specify that an entity returned by the engine has also been marked as true by the person marking the document.
- **False Positives** (FP). Engine-marked results that are not marked by a human. These results specify that an entity returned by the engine has not been marked as true by the person marking the document.
- **True Negatives** (TN). Results that are not marked either by the person marking the document, or the engine.
- **False Negatives** (FN). Human-marked results that are not marked by the engine. These results specify that an entity not returned by the engine has been marked as true by the person marking the document.

From these relevance terms, you can determine precision and recall as follows:

- Recall is the percentage of true relevant entities that are extracted by an extraction rule, that is,

  ```
  TP / (TP + FN) * 100
  ```

- Precision is the percentage of extracted entities that are true entities, that is,

  ```
  TP / (TP + FP) * 100
  ```

Case Sensitive Matches

You can configure Eduction to match characters case sensitively or case insensitively. By default, it is case sensitive, which has better performance.

The simplest way to match case insensitively is to disable the `MatchCase` configuration parameter (set the parameter to `False` in the configuration file). Alternatively, if you are creating your own custom XML grammar files, you can configure individual grammars, entities, and entries individually to be case sensitive or insensitive. If you configure case sensitivity at a lower level, it overrides the higher level settings. Additionally, if you reference the entity in another entity, it maintains its own case sensitivity setting.

Most entities in the standard grammars do not have case sensitivity set explicitly, giving you the flexibility to use case sensitivity as required in your grammars.

> **NOTE:**
> If you design an entity for case-insensitive matching, it is important that entries in the entity have a consistent case style to ensure that all matches are extracted correctly. You should use all lower case, all upper case, or all initial capitals, but not a mixture.Eduction uses an optimization technique for case insensitive matching that might not extract every possible match if the entity is not defined consistently.

Case Insensitive Match Performance

Case sensitive matching generally has better performance than case insensitive matching. If you require case insensitive matching, you can use case normalization to give the same performance as

case-sensitive matching.

When you want to use case normalization:

- Do not set case sensitivity explicitly in grammars and entities.
- Set the `MatchCase` configuration parameter to **`True`**.
- Create all entries in your entities in either all lower case, or all upper case.
- Set `CaseNormalization` to:
- **`LOWER`** if all your entities are lower case
- **`UPPER`** if all your entities are upper case.

Eduction normalizes the input data accordingly before the (case sensitive) matching. This process means that both your input and grammars are all in the same case, so the matching is effectively case insensitive, with the performance benefits of case sensitive matching.

When to Configure Case Sensitivity

Micro Focus recommends that you always create and use Eduction grammars that allow you to do case sensitive matching, because it has better performance. Most of the standard grammars come with entities using common and appropriate case styles. Some also have different entities for different case styles. If your data uses a consistent case, it is unlikely that you need to use case insensitive matching.

Case Sensitivity and Configured Field Names

You can set the `CaseSensitiveFieldName` configuration parameter to **`1`** to preserve the case and case sensitivity of configured field names in your output. See CaseSensitiveFieldName, on page 62 for more information.

Related Topics

- CaseNormalization, on page 61
- MatchCase, on page 73

Eduction Architecture

Figure 1 shows the basic Eduction architectural flow for extracting entities from a document when Eduction is used with IDOL.

Figure 1: IDOL Eduction workflow

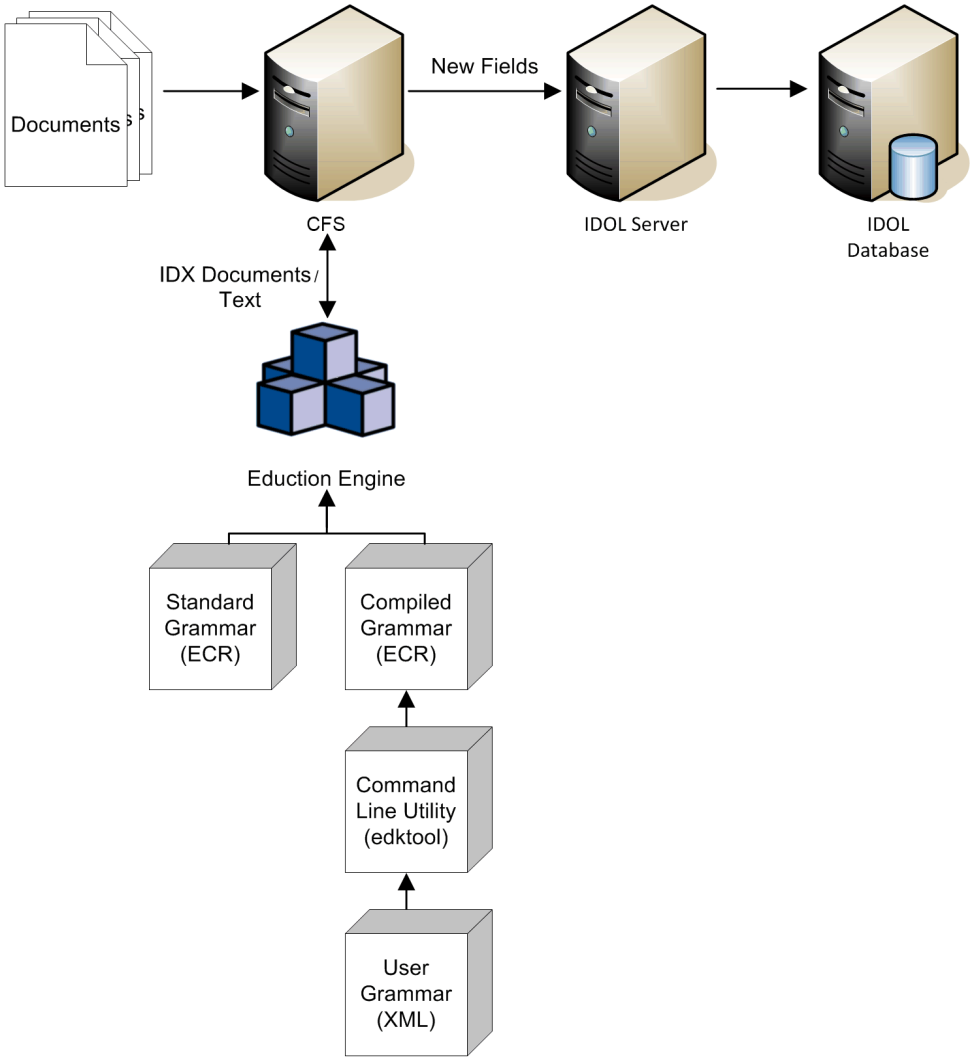


Figure 2 shows the Eduction components used when programming with the Eduction SDK.

Figure 2: Eduction SDK workflow

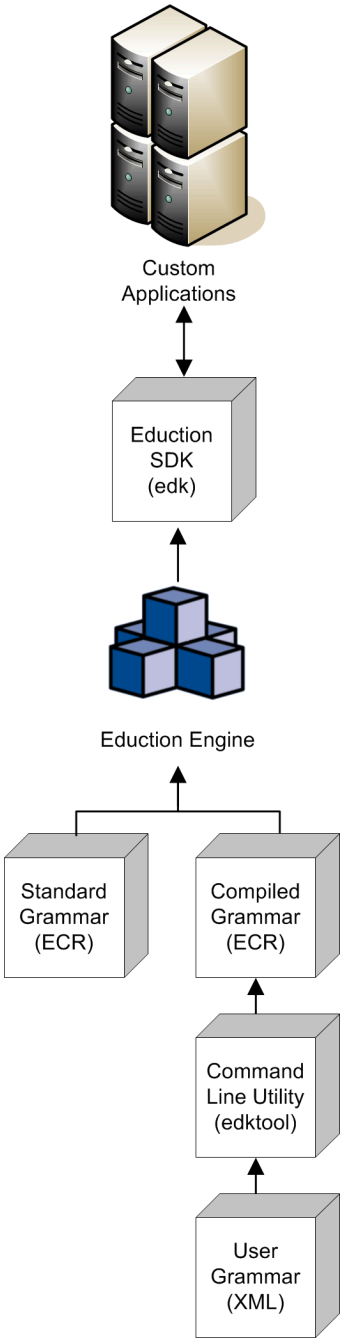


Figure 3 shows how you can perform extraction by using the Eduction grammar and the Eduction ACI server.

Figure 3: Extraction using the Eduction ACI Server

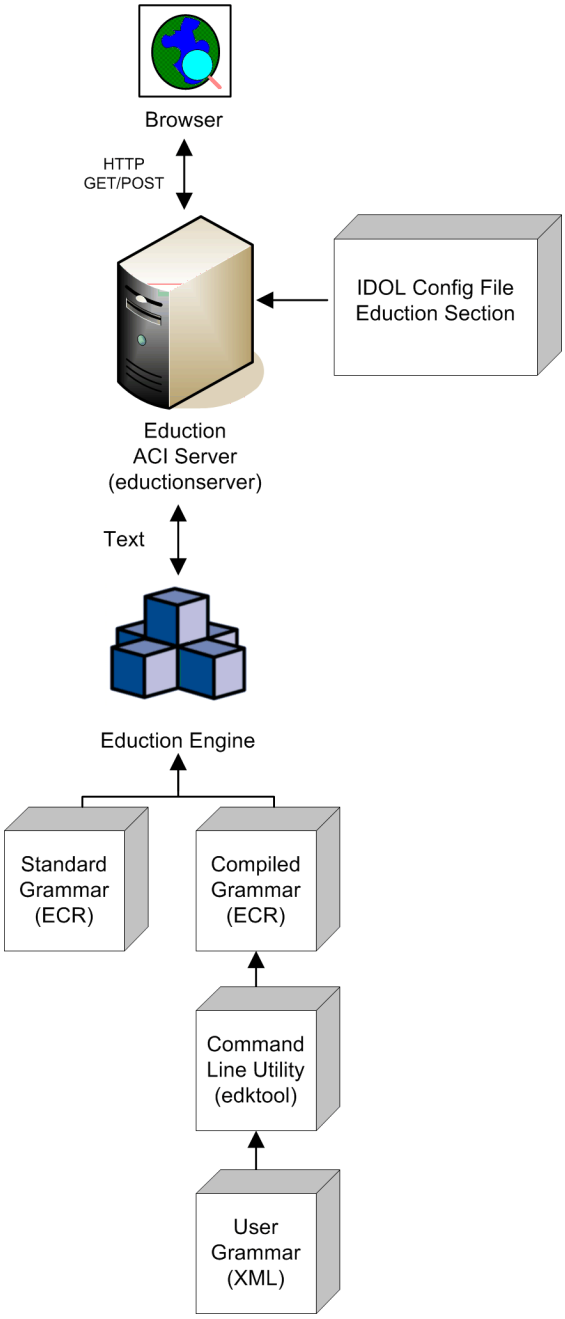
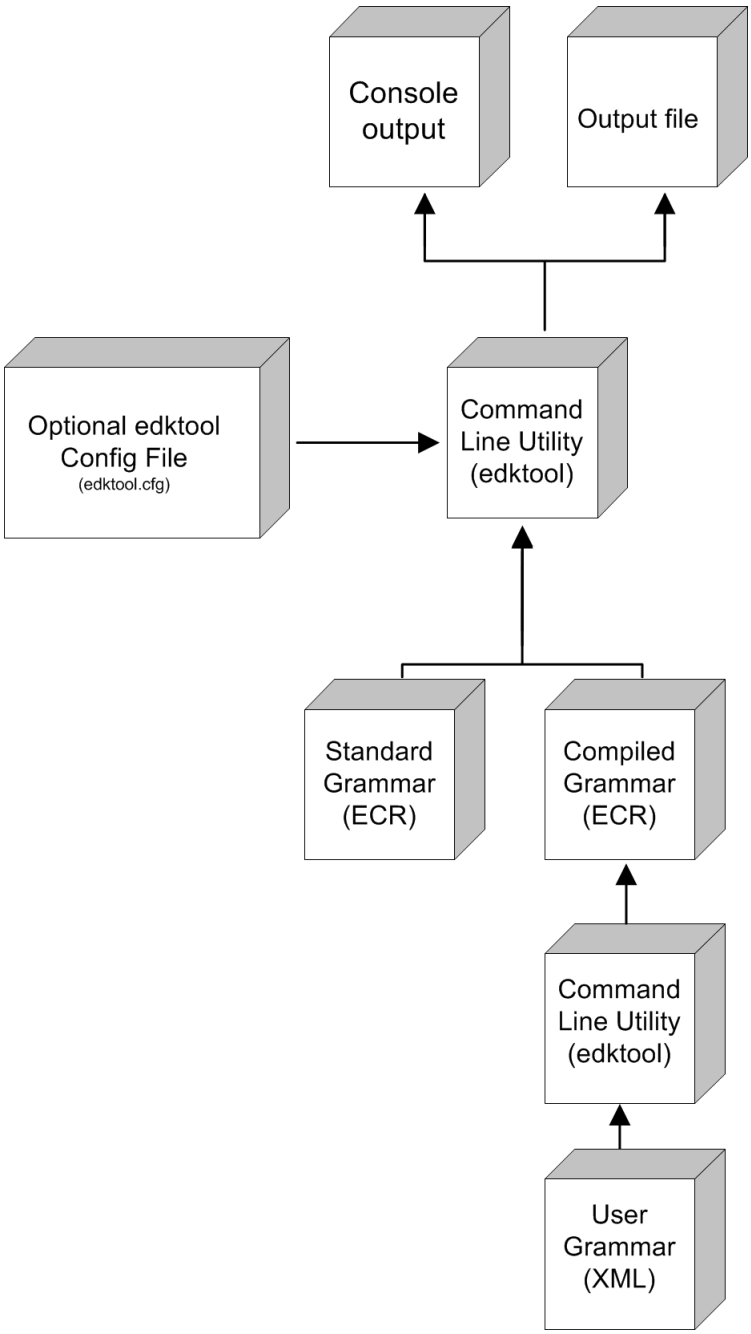


Figure 4 shows the workflow if you want to use the `edktool` command-line tool to compile and test grammar files, list entities, extract entitles, and so on. For more information see edktool Command-Line Tool, on page 41.

Figure 4: `edktool` workflow



Licenses

To use IDOL solutions, you must have a running License Server, and a valid license key file for the products that you want to use. Contact Micro Focus Big Data Support to request a license file for your installation.

License Server controls the IDOL licenses, and assigns them to running components. License Server must run on a machine with a static, known IP address, MAC address, or host name. The license key file is tied to the IP address and ACI port of your License Server and cannot be transferred between machines. For more information about installing License Server and managing licenses, see the *License Server Administration Guide*.

When you start Eduction, it requests a license from the configured License Server. You must configure the host and port of your License Server in the Eduction configuration file.

You can revoke the license from a product at any time, for example, if you want to change the client IP address or reallocate the license.

CAUTION:
Taking any of the following actions causes the licensed module to become inoperable.

You **must not**:

- Change the IP address of the machine on which a licensed module runs (if you use an IP address to lock your license).
- Change the service port of a module without first revoking the license.
- Replace the network card of a client without first revoking the license.
- Remove the contents of the `license` and `uid` directories.

All modules produce a `license.log` and a `service.log` file. If a product fails to start, check the contents of these files for common license errors. See Troubleshoot License Errors, on page 22.

Display License Information

You can verify which modules you have licensed either by using the IDOL Admin interface, or by sending the `LicenseInfo` action from a web browser.

To display license information in IDOL Admin

- In the **Control** menu of the IDOL Admin interface for your License Server, click **Licenses**.

  The **Summary** tab displays summary information for each licensed component, including:

  - The component name.
  - The number of seats that the component is using.
  - The total number of available seats for the component.
  - (Content component only) The number of documents that are currently used across all instances of the component.
  - (Content component only) The maximum number of documents that you can have across all instances of the component.

  The **Seats** tab displays details of individual licensed seats, and allows you to revoke licenses.

To display license information by sending the LicenseInfo action

- Send the following action from a web browser to the running License Server.

`http://LicenseServerHost:Port/action=LicenseInfo`

where:

LicenseServerHost	is the IP address of the machine where License Server resides.
Port	is the ACI port of License Server (specified by the `Port` parameter in the `[Server]` section of the License Server configuration file).

In response, License Server returns the requested license information. This example describes a license to run four instances of IDOL Server.

```
<?xml version="1.0" encoding="UTF-8" ?>
<autnresponse xmlns:autn="http://schemas.autonomy.com/aci/">
  <action>LICENSEINFO</action>
  <response>SUCCESS</response>
  <responsedata>
    <LicenseDiSH>
      <LICENSEINFO>
        <autn:Product>
          <autn:ProductType>IDOLSERVER</autn:ProductType>
          <autn:TotalSeats>4</autn:TotalSeats>
          <autn:SeatsInUse>0</autn:SeatsInUse>
        </autn:Product>
      </LICENSEINFO>
    </LicenseDiSH>
  </responsedata>
</autnresponse>
```

Configure the License Server Host and Port

Eduction is licensed through License Server. In the Eduction configuration file, specify the information required to connect to the License Server.

To specify the license server host and port

1. Open your configuration file in a text editor.
2. In the `[License]` section, modify the following parameters to point to your License Server.

`LicenseServerHost`	The host name or IP address of your License Server.
`LicenseServerACIPort`	The ACI port of your License Server.

   For example:

```
[License]
LicenseServerHost=licenses
LicenseServerACIPort=20000
```

3. Save and close the configuration file.

Revoke a Client License

After you set up licensing, you can revoke licenses at any time, for example, if you want to change the client configuration or reallocate the license. The following procedure revokes the license from a component.

To revoke a license

1. Stop the IDOL solution that uses the license.
2. At the command prompt, run the following command:

   InstallDir/ExecutableName[.exe] –revokelicense –configfile *cfgFilename*

This command returns the license to the License Server.

You can send the `LicenseInfo` action from a web browser to the running License Server to check for free licenses. In this sample output from the action, one IDOL Server license is available for allocation to a client.

```
<autn:Product>
   <autn:ProductType>IDOLSERVER</autn:ProductType>
   <autn:Client>
      <autn:IP>192.123.51.23</autn:IP>
      <autn:ServicePort>1823</autn:ServicePort>
      <autn:IssueDate>1063192283</autn:IssueDate>
      <autn:IssueDateText>10/09/2003 12:11:23</autn:IssueDateText>
   </autn:Client>
      <autn:TotalSeats>2</autn:TotalSeats>
      <autn:SeatsInUse>1</autn:SeatsInUse>
</autn:Product>
```

Troubleshoot License Errors

The table contains explanations for typical licensing-related error messages.

License-related error messages

Error message	Explanation
Error: Failed to update license from the license server. Your license cache details do not match the current service configuration. Shutting the service down.	The configuration of the service has been altered. Verify that the service port and IP address have not changed since the service started.

License-related error messages, continued

Error message	Explanation
Error: License for *ProductName* is invalid. Exiting.	The license returned from the License Server is invalid. Ensure that the license has not expired.
Error: Failed to connect to license server using cached licensed details.	Cannot communicate with the License Server. The product still runs for a limited period; however, you should verify whether your License Server is still available.
Error: Failed to connect to license server. Error code is SERVICE: *ErrorCode*	Failed to retrieve a license from the License Server or from the backup cache. Ensure that your License Server can be contacted.
Error: Failed to decrypt license keys. Please contact Autonomy support. Error code is SERVICE:*ErrorCode*	Provide Micro Focus Big Data Support with the exact error message and your license file.
Error: Failed to update the license from the license server. Shutting down	Failed to retrieve a license from the License Server or from the backup cache. Ensure that your License Server can be contacted.
Error: Your license keys are invalid. Please contact Autonomy support. Error code is SERVICE:*ErrorCode*	Your license keys appear to be out of sync. Provide Micro Focus Big Data Support with the exact error message and your license file.
Failed to revoke license: No license to revoke from server.	The License Server cannot find a license to revoke.
Failed to revoke license from server *LicenseServer Host*:*LicenseServerPort*. Error code is *ErrorCode*	Failed to revoke a license from the License Server. Provide Micro Focus Big Data Support with the exact error message.
Failed to revoke license from server. An instance of this application is already running. Please stop the other instance first.	You cannot revoke a license from a running service. Stop the service and try again.
Failed to revoke license. Error code is SERVICE:*ErrorCode*	Failed to revoke a license from the License Server. Provide Micro Focus Big Data Support with the exact error message.
Your license keys are invalid. Please contact Autonomy Support. Error code is ACISERVER:*ErrorCode*	Failed to retrieve a license from the License Server. Provide Micro Focus Big Data Support with the exact error message and your license file.
Your product ID does not match the generated ID.	Your installation appears to be out of sync. Forcibly revoke the license from the License Server and rename the `license` and `uid` directories.

License-related error messages, continued

Error message	Explanation
Your product ID does not match this configuration.	The service port for the module or the IP address for the machine appears to have changed. Check your configuration file.

Chapter 2: Deploy Eduction SDK

This chapter describes the files in the Eduction SDK and how to deploy the SDK. It contains the following sections:

Eduction SDK

The Eduction SDK includes the following components:

- Standard collection of grammar files covering a range of commonly used entities
- `edktool` command-line tool used for compiling Eduction XML source grammar files into compiled run-time ECR files
- C API
- Java API

The Eduction SDK includes reference documentation for the C and Java APIs. To view the documentation, open `c_api/help/index.html` or `java_api/help/index.html` in a web browser.

C API Component

The C API component of the Eduction SDK includes:

- The Eduction header file (`edk.h`).
- The Eduction SDK library linker file (`edk.lib`), Windows only.
- The Eduction SDK library on Windows (`edk.dll`) or shared object on UNIX (`libedk.so`).
- Several sample C programs that demonstrate various SDK features.

Solution files and project files are provided for Visual Studio 2010.

To use the Eduction SDK in C, include the `edk.h` header file from your C source code and link with the SDK library. A sample makefile is provided showing how to compile and link to the SDK.

> **NOTE:**
> On Windows, you must specify the Eduction library in the `PATH` environment variable. On UNIX, the shared object must be in the library search path.

Build the Sample Programs

You can use the sample programs to validate that the Eduction EDK compiles, links, and runs correctly.

UNIX

Use the following procedure to build the sample programs on UNIX.

To build the sample programs on UNIX

1. Navigate to the `c_api/test` directory.
2. Set the `PLATFORM` environment variable (refer to the makefile for more details).
3. Run `make dir=ext`.

The resulting binaries require that the shared object be in the library search path. By default, this is not the case, so a Perl script is provided to add the shared object to the search path by setting `LD_LIBRARY_PATH`. The script then runs the test binary. To run the script, use the following command:

```
perl runtestedk.pl ext
```

The Perl script expects a valid Eduction OEM license key with the name `licensekey.dat` to be located in the `test` directory. If the test works correctly, you see a line of text in the output beginning with the string `PASS:`, otherwise you see the string `FAIL:`.

Windows

To build the test executable binaries, a Visual Studio 2010 solution file, `test.sln`, is provided in the `test` folder.

The resulting binaries require that the `edk.dll` be in the library search path. By default, this is not the case, so a Perl script is provided to add the library to the search path and then run the test binary. To run the script, use the following command:

```
perl runtestedk.pl ext
```

The Perl script expects a valid Eduction OEM license key with the name `licensekey.dat` to be located in the `test` directory. If the test works correctly, you see a line of text in the output beginning with the characters `PASS:`, otherwise you see `FAIL:`.

Java API Component

The Java API component of the Eduction SDK includes:

- The Eduction library on Windows (`edkjni.dll`) or shared object on UNIX (`libedkjni.so`).
- The Java JAR file (`edk.jar`).
- Java code samples that you can compile and execute by using the Eduction SDK. These samples

are located in the `\src\com\autonomy\eduction\test\` directory of the Eduction SDK root directory, and are named `SampleN.java` (`Sample1.java`, `Sample2.java`, and so on). Each `SampleN.java` file illustrates aspects of SDK functionality, and contains detailed descriptions of the functionality in question. For information on how to compile and run `Sample1.java`, see Build and Run the Sample Programs, below.

> **NOTE:**
> Before you follow the instructions on how to compile and run `Sample1.java`, you must copy the `licensekey.dat` file into the `test` subfolder of your Eduction SDK installation. You must perform all command-line operations from the Eduction SDK installation directory.

> **NOTE:**
> On Windows, you must specify the Eduction library in the `PATH` environment variable. On UNIX, the shared object must be in the library search path.

Build and Run the Sample Programs

Use the following procedure to build and run the sample programs.

To build and run the sample programs

1. Navigate to the `java_api` directory.
2. Run `ant all`.

Chapter 3: API Reference

This chapter describes the C and Java APIs for Eduction SDK, and contains the following sections:

C API Concepts

This section describes concepts required to implement C language applications for Eduction SDK.

Include Files

The Eduction SDK C API uses the `edk.h` include file. This include file contains the core APIs for the Eduction SDK engine.

Naming Conventions

The types, functions, and macros specific to Eduction SDK are prefixed with the string `Edk`.

Concurrency Control

Concurrency in Eduction is handled using *sessions*. An instance of an `EDKEngine` is initialized with corresponding grammars for entity extraction. Each such engine can be associated with one or more sessions. All the sessions in the engine share the same grammars. The engine must be configured fully before any sessions are created.

After you create a session, an `EdkError` is thrown if you change the engine settings. However, each individual session can process many documents or streams. Each session maintains its state independent of others. Each engine must be associated with at least one session by default.

Standalone API Usage

The Eduction Software Development Kit (SDK) C API allows C developers to interact directly with the Eduction engine.

At the core of the API is an eduction engine. The first call of the API in an application should create the engine. You must then set a valid license for the engine.

After you create it, you can optionally configure an engine to determine its matching behavior. You must load one or more resource files that contain the eduction grammars, and you must add one or more entities, from the loaded resource files, to match against.

Data processing is performed in an eduction session. You can create multiple sessions for each eduction engine. All sessions use the same loaded grammars and entities. Each session maintains its own state so that the sessions can run concurrently in a multithreaded application.

You can use the session to process multiple documents. You can pull (stream) or push (add) data. You call a function to get the next available match. You can call this function repeatedly to cycle through all the matches. For each match, you can access the associated text and properties by using several function calls.

You can keep a session alive for as long as necessary. However, you must destroy it before you destroy the engine that is associated with it. The call to destroy the engine should be the last call of this API in an application.

This section describes the skeletal structure of a stand-alone application using the API. See the source code in C API Examples, on the next page. Typically, your application takes the following actions:

1. Include `edk.h`.
2. Instantiate the engine and obtain an engine handle.

   You can create the engine without a configuration, or you can call `EdkEngineCreateFromConfigFile` to create an engine from an appropriate configuration file. If you use `EdkEngineCreateFromConfigFile`, you can skip steps 3 and 4.
3. Set the license key.
4. Configure the engine to:
   - set optional parameters.
   - load the grammar files to use for matching.
   - add specific entities from grammars to use for matching.
5. Create a session associated with the engine, and obtain a session handle. You can create and run concurrent sessions in a multithreaded application. Each session uses the same grammars, but maintains its own state.
6. Feed UTF-8 encoded text to the session.
7. Call `EdkGetNextMatch` to obtain an entity match. You can call this method repeatedly to obtain all matches.

   > **NOTE:**
   > If you create your engine from a configuration file that includes post-processing tasks, the post-processing tasks automatically run as part of `EdkGetNextMatch` and you do not need to run them separately.
8. If required, call `EdkGetRedactedText` to produce redacted output.
9. For each match, get details and properties of the match.
10. To process multiple documents, repeat Step 6 to Step 9.
11. Release resources when done. You must destroy all session handles before destroying the engine handle.

C API Examples

This section describes the sample programs that are included in the SDK to demonstrate specific API functions. The sample programs are available in the `c_api/test` folder of the Eduction SDK.

basic.c

The `basic.c` program demonstrates the basic usage of the C API functions, as described in Standalone API Usage, on page 28. This program:

- creates an Eduction engine.
- configures an Eduction engine.
- associates a session with the Eduction engine.
- adds text to the session, and performs an entity extraction on the text.
- outputs the results.

To build and run the sample, navigate to the `test` directory in your Eduction installation, and open the text file `readme.txt`. This file provides specific directions on how to compile and run the sample. You must have an Eduction OEM license key to run the sample.

cjknormalization.c

The `cjknormalization.c` program builds upon `basic.c` and shows how you can perform normalization of Chinese, Japanese, and Korean input text before you process it in an eduction session.

postprocess.c

The `postprocess.c` program builds upon `basic.c` and shows the work flow needed to support post-processing in eduction. The program:

1. creates and configures an eduction engine.
2. creates post-processing tasks.
3. creates an eduction session to process the input.
4. adds input text to the session.
5. performs the extraction.
6. collects the matches from the extraction.
7. runs post-processing tasks on the matches.
8. prints the results.
9. cleans up the post-processing tasks.

redaction.c

The `redaction.c` program demonstrates the use of the `EdkGetRedactedText` function. It shows how to set up an eduction engine with the specified grammars and entities, and how you can use that engine to produce redacted output when given an input buffer.

multithread.c

The `multithread.c` example program is for use on the Linux platform.

This sample program demonstrates the use of multiple eduction sessions associated with a single eduction engine running in parallel.

This program takes arguments of the names of the text files to process. The program has the following stages:

1. The program creates and configures an eduction engine.
2. The program creates a worker thread for each text file.
3. The worker thread starts with the eduction engine and the text file. This worker thread:
   - creates a session associated with the eduction engine.
   - adds text from the text file to the session.
   - performs an entity extraction of the text.
4. The main program then waits for each thread to finish extraction, exit, and join.

Java API Concepts

Eduction SDK provides a Java API that enables your application to create an extraction engine and perform entity extractions.

This section describes the concepts used to write Java applications with the Eduction SDK.

The Java SDK consists of a JAR file, and a DLL (Windows) or shared object (UNIX/Linux).

- The JAR file contains the Eduction Java class library, and the interface to the Eduction Java Native Interface (JNI).
- The DLL or shared object implements the Eduction JNI library, and performs the Eduction functionality.
- Java developers can use either the Eduction JNI, the class library, or both. The JNI provides functionality almost identical to that of the Eduction C API. The class library encapsulates related JNI methods, implements exception handling, and provides return values from method calls that simplify application programming.

Naming Conventions

The main JNI class that provides access to native functionality is `EDKJNI`. Support classes for the JNI are prefixed with `EDKJNI`, for example `EDKJNIVersion`.

The Eduction class library classes are prefixed with `EDK`, for example `EDKEngine`.

Concurrency Control

Concurrency in Eduction is handled using *sessions*, represented by an `EDKSession` object. An instance of an `EDKEngine` object is initialized with corresponding grammars for entity extraction. Each such engine can be associated with one or more sessions. All the sessions in the engine share the same grammars. The engine must be configured fully before any sessions are created.

After a session is created, engine settings cannot be changed, or an exception is thrown. However, each individual session can process many documents or streams. Each session maintains its state independent of others. Each engine must be associated with at least one session by default.

Standalone API Usage

This section describes the basic structure of a stand-alone application using the API. See the source referenced in Java API Example, on the next page. Typically, your application takes the following actions:

1. Instantiate an `EDKEngine` object.

   > **TIP:**
   > Micro Focus recommends that you program against the `TextExtractionEngine` interface to instantiate `EDKEngine`.

   You can create the engine without a configuration, or you can create an engine from an appropriate configuration file. If you use a configuration file, you can skip steps 2 and 3.

2. Set the license key.
3. Configure the engine, load the grammars, and select entities.
4. Instantiate an `EDKSession` object and associate it with the engine.

   > **TIP:**
   > Micro Focus recommends that you program against the `TextExtractionSession` interface to instantiate `EDKSession`.

5. Feed UTF-8 encoded text to the session or pass the session a stream it can read.
6. Execute a `for each` loop to obtain an `EDKMatch` object for each match in a session.

   > **NOTE:**
   > If you create your `EDKEngine` from a configuration file that includes post-processing tasks, the post-processing tasks automatically run as part of `EDKMatch` and you do not need to run

> them separately.

7. Get the properties of the match.
8. Release resources when done.

Java API Example

The Eduction Java API includes sample programs that illustrate basic Eduction Java API usage. The samples are located in the `src/com/autonomy/eduction/test` folder. The samples are documented inline and are also described in the HTML documentation.

For information on how to build and run the sample programs, see Build and Run the Sample Programs, on page 27.

Chapter 4: Eduction ACI Server

This section describes the Eduction ACI Server.

Introduction to Eduction Server

The Eduction ACI Server is a stand-alone server based on the ACI server. The Eduction Server processes UTF-8 encoded text, matching upon entities defined in Eduction grammars. Results return as XML, with tags in the ACI hierarchy. You define the grammars to load and the entities to match on using a configuration file. Every time you send a query, the server creates a new engine with the grammars and entities that you specified.

You can make requests from a browser or ACI client. Browsers can make requests to process small amounts of text using an HTTP GET request, or larger amounts using an HTTP POST request.

You can specify configuration settings as query parameters to override the settings in the configuration file for individual queries. See Select Entities at Runtime, on page 39 for more information.

Command-Line Options

Usage: `eductionserver [options]`

Options	Description
`-version`	Displays the program version. This option must be the only argument. The program ends after the version information is displayed.
`directory`	Sets the working directory. The Eduction ACI server starts from this directory. All other arguments used in the command line are relative to this directory.
`-configfile` *`file`*	Sets the configuration file name. This option overrides the default configuration file name of `eductionserver.cfg`.
`-revokelicense`	Revokes a lock on the Eduction license from the License Server.
`-install`	Installs Eduction as a service (Windows only). Syntax: `-install [-start auto|manual|disabled] [-username username] [-`

Options	Description
	`password password]` The `-start` option allows you to specify the startup mode. By default, it is automatic. If you do not supply a user name and password, Eduction runs under a local account.
`-uninstall`	Uninstalls the Eduction service.

NOTE:
Options are case sensitive.

NOTE:
On Linux, the ACI server requires the C++ library, `libstdc++.so`. To ensure the server can locate the required library, set the Library Path:

```
setenv LD_LIBRARY_PATH bin:$LD_LIBRARY_PATH
```

Example Configuration File

The Eduction ACI Server configuration file contains settings required for the server to run and specifies settings to use when performing Eduction. The default configuration file is named `eductionserver.cfg`. You can start the Eduction Server using a different configuration file by using the `-configfile` command-line option (see Command-Line Options, on the previous page).

```
[License]
LicenseServerHost=127.0.0.1
LicenseServerACIPort=20000

[Server]
Port=7075
Threads=1

[Eduction]
ResourceFiles=person_name_engus.ecr
Entity0=person/femalefirstname/engus
Entity1=person/malefirstname/engus
Entity2=person/lastname/engus
//MinScore=0.5
//MaxEntityLength=12
//MatchCase=0
//CaseNormalization=Lower
//AllowOverlaps=1
//EnableComponents=1
//MatchWholeWord=0
//RedactionOutputString=[censored]

[Logging]
```

```
LogLevel=full
0=ApplicationLogStream

[ApplicationLogStream]
LogFile=application.log
```

Include an External Configuration File

You can share configuration sections or parameters between ACI server configuration files. The following sections describe different ways to include content from an external configuration file.

You can include a configuration file in its entirety, specified configuration sections, or a single parameter.

When you include content from an external configuration file, the `GetConfig` and `ValidateConfig` actions operate on the combined configuration, after any external content is merged in.

In the procedures in the following sections, you can specify external configuration file locations by using absolute paths, relative paths, and network locations. For example:

```
../sharedconfig.cfg
K:\sharedconfig\sharedsettings.cfg
\\example.com\shared\idol.cfg
file://example.com/shared/idol.cfg
```

Relative paths are relative to the primary configuration file.

> **NOTE:**
> You can use nested inclusions, for example, you can refer to a shared configuration file that references a third file. However, the external configuration files must not refer back to your original configuration file. These circular references result in an error, and Eduction does not start.
>
> Similarly, you cannot use any of these methods to refer to a different section in your primary configuration file.

Include the Whole External Configuration File

This method allows you to import the whole external configuration file at a specified point in your configuration file.

To include the whole external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the external configuration file.
3. On a new line, type a left angle bracket (`<`), followed by the path to and name of the external configuration file, in quotation marks (`""`). You can use relative paths and network locations. For example:

```
< "K:\sharedconfig\sharedsettings.cfg"
```

4. Save and close the configuration file.

Include Sections of an External Configuration File

This method allows you to import one or more configuration sections (including the section headings) from an external configuration file at a specified point in your configuration file. You can include a whole configuration section in this way, but the configuration section name in the external file must exactly match what you want to use in your file. If you want to use a configuration section from the external file with a different name, see Merge a Section from an External Configuration File, on the next page.

To include sections of an external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the external configuration file section.
3. On a new line, type a left angle bracket (<), followed by the path of the external configuration file, in quotation marks (""). You can use relative paths and network locations. After the configuration file path, add the configuration section name that you want to include. For example:

   ```
   < "K:\sharedconfig\extrasettings.cfg" [License]
   ```

   > **NOTE:**
   > You cannot include a section that already exists in your configuration file.

4. Save and close the configuration file.

Include Parameters from an External Configuration File

This method allows you to import one or more parameters from an external configuration file at a specified point in your configuration file. You can import a single parameter or use wildcards to specify multiple parameters. The parameter values in the external file must match what you want to use in your file. This method does not import the section heading, such as `[License]` in the following examples.

To include parameters from an external configuration file

1. Open your configuration file in a text editor.
2. Find the place in the configuration file where you want to add the parameters from the external configuration file.
3. On a new line, type a left angle bracket (<), followed by the path of the external configuration file, in quotation marks (""). You can use relative paths and network locations. After the configuration file path, add the name of the section that contains the parameter, followed by the parameter name. For example:

   ```
   < "license.cfg" [License] LicenseServerHost
   ```

To specify a default value for the parameter, in case it does not exist in the external configuration file, specify the configuration section, parameter name, and then an equals sign (=) followed by the default value. For example:

```
< "license.cfg" [License] LicenseServerHost=localhost
```

You can use wildcards to import multiple parameters, but this method does not support default values. The * wildcard matches zero or more characters. The ? wildcard matches any single character. Use the pipe character | as a separator between wildcard strings. For example:

```
< "license.cfg" [License] LicenseServer*
```

4. Save and close the configuration file.

Merge a Section from an External Configuration File

This method allows you to include a configuration section from an external configuration file as part of your Eduction configuration file. For example, you might want to specify a standard SSL configuration section in an external file and share it between several servers. You can use this method if the configuration section that you want to import has a different name to the one you want to use.

To merge a configuration section from an external configuration file

1. Open your configuration file in a text editor.
2. Find or create the configuration section that you want to include from an external file. For example:

   ```
   [SSLOptions1]
   ```

3. After the configuration section name, type a left angle bracket (<), followed by the path to and name of the external configuration file, in quotation marks (""). You can use relative paths and network locations. For example:

   ```
   [SSLOptions1] < "../sharedconfig/ssloptions.cfg"
   ```

   If the configuration section name in the external configuration file does not match the name that you want to use in your configuration file, specify the section to import after the configuration file name. For example:

   ```
   [SSLOptions1] < "../sharedconfig/ssloptions.cfg" [SharedSSLOptions]
   ```

   In this example, Eduction uses the values in the [SharedSSLOptions] section of the external configuration file as the values in the [SSLOptions1] section of the Eduction configuration file.

   > **NOTE:**
   > You can include additional configuration parameters in the section in your file. If these parameters also exist in the imported external configuration file, Eduction uses the values in the local configuration file. For example:
   >
   > ```
   > [SSLOptions1] < "ssloptions.cfg" [SharedSSLOptions]
   > SSLCACertificatesPath=C:\IDOL\HTTPConnector\CACERTS\
   > ```

4. Save and close the configuration file.

Server Actions

You can run actions on the Eduction ACI server using the HTTP request:

`http://Host:Port/?action=action[&Parameter=Value[&Parameter=Value...]]`

For example:

`http://localhost:13000/action=GetStatus`

The Eduction ACI server provides the following actions (case insensitive):

`GetStatus`	Returns the status of the Eduction Server, including version information and entities selected for matching.
`EduceFromFile`	Performs Eduction on text (read from a file) and returns the matches.
`EduceFromText`	Performs Eduction on text (provided in the request) and returns the matches.
`RedactFromFile`	Performs Eduction on text (read from a file) and redacts the matches to return the redacted text.
`RedactFromText`	Performs Eduction on text (provided in the request) and redacts the matches to return the redacted text.

For more information about these actions, and example responses, refer to the *Eduction Server Reference*.

Select Entities at Runtime

You can customize the behavior of extraction for individual actions by specifying the configuration settings as query parameters in the ACI request. For example:

`http://localhost:13000/?action=EduceFromFile&MatchCase=True&Grammars=place_albal.ecr`

If you specify a parameter as part of a query, it overrides the corresponding parameter value in the configuration file.

You can specify a comma-separated list of grammar files to load as the value of the `Grammars` parameter. For example:

`Grammars=GrammarFile[,GrammarFile2]`

This corresponds to the `ResourceFiles` setting in the configuration file.

You can specify a comma-separated list of entities to use as the value of the `Entities` parameter. If no entities are specified in the configuration file, all public entities from the grammar files that you configured are available.

Alternatively,if your query uses several grammar files or entities, you can use wildcard expressions in the `Grammars` or `Entities` parameters. You can use the * wildcard to match any number of characters, or the ? wildcard to match a single character. For example:

```
action=EduceFromText&Text=I thought it was a bad idea. Es ist nicht
gut.&Grammars=sentiment_*.ecr
```

This example uses all the available sentiment grammars for the extraction without you having to type a lengthy comma-separated list.

> **NOTE:**
> The grammar files and entities must already be specified in the configuration file. There must be no space before or after a comma.

For information about the actions and action parameters that are available with Eduction Server, refer to the *Eduction Server Reference*.

Chapter 5: edktool Command-Line Tool

This section describes `edktool` and the configuration files that specify Eduction settings that are used with it.

About edktool

`edktool` is a tool for Eduction that allows you to compile and test your grammars. `edktool` can perform the following functions:

- compile grammars
- list available entities in a grammar file
- extract entities from a file based on a grammar and select entities from the grammar for extraction
- test the accuracy of the extraction process

> **NOTE:**
> On Linux, `edktool` requires the C++ library, `libstdc++.so`. To ensure the tool can locate the required library, set the Library Path:
>
> ```
> setenv LD_LIBRARY_PATH bin:$LD_LIBRARY_PATH
> ```

Wildcard Expressions in edktool

The -e and -g parameters in the Generate , Compile, Assess, Extract and Benchmark options in edktool support wildcard expressions. For example, if you want to use all of the available sentiment analysis files in the `grammars` directory, you can type `-e "grammars/sentiment_*.ecr"` instead of typing a lengthy comma-separated list of multiple files.

You can use the `*` wildcard to match any number of characters, or the `?` wildcard to match a single character.

> **NOTE:**
> In some cases (for example, if you are running Linux), the command shell automatically expands wildcard expressions, which can produce unexpected results in Eduction. To avoid this, you should enclose your wildcard expression in quotation marks.

edktool Options

This section describes how to use `edktool` to list the entities that exist in a grammar file, test and benchmark extraction, compile grammar files, and complete other tasks.

To view a summary of the options you can use with `edktool`, run the following command:

```
edktool help
```

The tool also provides more detailed help for some features. For example, to view more information about the `compile` feature, run the following command:

```
edktool help compile
```

Compile

This option creates a compiled Eduction grammar file.

You can use wildcard expressions in the `-e` parameter; see Wildcard Expressions in edktool, on the previous page for more information.

`-l <licensefile>`	The file containing a valid license key for Eduction. If you do not specify a license key at the command line, `edktool` assumes that the location of the license file is `licensekey.dat`. If the license is kept in this location, you do not need to specify this parameter.
`-i <inputfile>`	The grammar file to process. The input file can be an uncompiled (Source) XML Eduction grammar file or a plaintext grammar file.
`-e <entity>`	A comma-separated list of entities to include in the output file. If you do not include any entities in the command line, Eduction includes all entities in the input file in the output file. If you include entities in the command line, Eduction includes only those entities specified in the output file.
`-o <outputfile>`	The output file name. If you do not specify the output file name, Eduction creates an output file using the XML grammar file name with `.ecr` appended.
`-p`	Set this parameter if you want to use a plaintext grammar file (containing one potential match on each line) rather than an XML grammar file as the input text to compile from.

When compiling, the XML file must follow the Eduction syntax rules for laying out grammar files. The ECR file is a proprietary format that is optimized for fast loading into the Eduction engine at run time. While the engine can load XML grammar files, as well as compiled ECR files, compiling a grammar file makes loading quicker.

Because compiled grammar files are binary files and cannot be read, the `List` option allows you to view the public entities in a compiled grammar file.

You can also specify the `-p` parameter at the command line to compile a grammar file in ECR format from a plaintext grammar file. The plaintext grammar file must be in the format described in Plaintext Grammar File Format , on page 50.

Examples

To compile `mygrammar.xml` into `mygrammar.ecr`:

```
edktool c mygrammar.xml
```

To compile all the entities in the `common` entity type in `mygrammar.xml` into `compiledgrammar.ecr`:

```
edktool c -i mygrammar.xml -e common/* -o compiledgrammar.ecr
```

List

This option lists the entities in an uncompiled (Source) XML Eduction grammar file or a compiled ECR grammar file. Listing the contents of an XML file lists all entities in the file, both private and public. Listing the contents of a compiled ECR file lists all public entities. Private entities not referenced by the public entities are removed from the compiled ECR file.

To enable this feature, type `edktool l <grammarfile>` at the command line.

You can also include the optional `-a` parameter when using the LIST option. As well as listing the components that the entity can return, this lists the licence requirements for a particular compiled grammar file. For example, the following output:

```
category: place              languages: English or French
```

indicates that the user must be licensed for either English or French in the place category. If multiple lines appear, then the license must satisfy the conditions in every line.

If you include the optional `-q` "Quiet Mode" parameter, `edktool` removes all descriptive messages from the output and shows the entity list only. The output includes components if you also set the `-a` parameter.

Example

To list all public entities in the compiled grammar file `mygrammar.ecr`:

```
edktool list mygrammar.ecr
```

Permissions

This option reads any specified directory and returns a list of all compiled grammar files inside it that you can access using the specified licence.

To enable this feature, type `edktool p -d <directory> -l <licencefile>` at the command line. You can also include the optional parameter `-a` to return a list of all compiled grammar files inside the directory that are **not** accessible under the specified licence.

> **NOTE:**
> If you do not specify a license key at the command line, `edktool` assumes that the location of

> the license file is `licensekey.dat`. If the license is kept in this location, you do not need to specify the `-l <licensefile>` parameter.

You can include the optional `-q` parameter to enable "Quiet Mode" and remove descriptive messages from the output. If you enable “Quiet Mode”, the output consists of a list of file names only, in the format `Valid: filename.ecr` or, if you also included the `-a` parameter, `Invalid: filename.ecr`.

Generate

This option generates an uncompiled XML source file from a plaintext grammar file.

To enable this feature, type `edktool g -i <inputfile>` at the command line. You can also specify the optional `-o` and `-e` parameters.

You can use wildcard expressions in the `-e` parameter; see Wildcard Expressions in edktool, on page 41 for more information.

`-i <inputfile>`	The plaintext grammar file to process. This file must contain one potential match on each line.
`-e <entity>`	The resulting XML grammar file contains a single entity; you can specify the name of this entity as the value of the -e parameter. If you do not specify a name, the entity will be given a default name based on the name of the input file.
`-o <outputfile>`	The output file name.

Related Topics

- Plaintext Grammar File Format , on page 50

Assess

This option enables you to assess the performance and accuracy of an Eduction grammar against a set of pre-tagged examples.

You must supply a text file with one phrase on each line; the `Assess` feature checks whether each line contains a match.

You must specify at least one input file, using the `-v` parameter or the `-w` parameter. If required, you can specify both of these parameters.

You can use wildcard expressions in the `-e` and `-g` parameters; see Wildcard Expressions in edktool, on page 41 for more information.

`-l <licensefile>`	The file containing a valid license key for Eduction. If you do not specify a license key at the command line, `edktool` assumes that the location of the license file is `licensekey.dat`. If the license is kept in this location, you do not need to specify this parameter.
`-c <configfile>`	A configuration file controlling the assessment. The configuration file can be

	either an IDOL Server style .CFG configuration file or an XML configuration file. See Configuration Files for Eduction Settings, on page 51. You can specify one or more grammar files and one or more entities in place of a configuration file. Specifying a configuration file overrides the grammar or entity parameters.
`-g <grammarfile>`	A grammar file to use when `-c` is not used. If you provide a grammar file but you do not specify any entities with `-e`, Eduction extracts all entities in the grammar file.
`-e <entity>`	The entities to extract when `-c` is not used. Separate multiple entities with a comma.
`-x`	(Optional) Modifies the behavior so that the `Assess` feature checks for exact matches.
`-m <matched entities>`	(Optional) This parameter does not change the extraction behavior, but enables you to check which entities are producing the matches.
`-v <valid_input>`	A file of phrases where a match would be valid.
`-w <invalid_input>`	A file of phrases where a match would be invalid.
`-a`	(Optional) The output includes explanations of each failure, and statistics such as recall, precision, and F1 (depending on the type of input file you provided). Include the `-a` parameter to display additional output, including the results for every phrase in your input files.
`-o <outputfile>`	(Optional) By default, Eduction sends output to the console. To send the output to a file, use the `-o` parameter. The output is a list of all phrases that failed. For valid input this would be a phrase that contained no match; for invalid input this would be a phrase that contained a match.
`-q`	(Optional) Sets “Quiet Mode” so that descriptive messages are removed, and the output consists only of a list of examples that failed, in the form `"FAIL: "text" is matched by "entity""` or similar, depending on the test specifications. If you also set the `-a` parameter, examples that pass are also included in the output.

For more information on how to use the Assess feature to check the effectiveness and performance of your grammar files, refer to *IDOL Expert*.

Example

```
edktool a -l <license> -c <configuration_file> [-a] [-o <output_file>]
```

Run several assessments from a single Eduction configuration file.

The configuration file must contain a numbered `[assessmentN]` section for each assessment you want to run. You must specify the input files, the entities to match, and whether matching should be exact. For example:

```
[assessment0]
valid=data.txt

[assessment1]
entities=entity1,entity2
valid=match.txt
invalid=should_not_match.txt
exact=true
```

You can specify multiple entities either by separating them with commas, or by using wildcard expressions. You can use the * wildcard to match any number of characters, or the ? wildcard to match a single character. For example, set `Entities` to `org/soccer/*` to use the entities `org/soccer/us`, `org/soccer/gb`, `org/soccer/de`, and so on without having to type a lengthy comma-separated list.

Extract

This option extracts entities from a document. It can print the output to a file, or to the console. You can use this option to test your grammars.

You can use wildcard expressions in the `-e` and `-g` parameters; see Wildcard Expressions in edktool, on page 41 for more information.

Redact Extraction Results

You can enable redaction on extracted matches in `edktool` either by setting `RedactedOutput` to **`True`** in the `edktool` configuration file, or by specifying a redaction file using the `-r` parameter at the command line. Note that `edktool` only performs redaction on fields that you have configured as IDOL search fields.

If you have specified an IDX file to perform extraction on, existing fields are preserved in their unredacted form, and a redacted copy of each search field is added to the IDX file, with `_REDACTED` appended to the original field name. For example:

```
#DREREFERENCE 1
#DREFIELD DRECONTENT_REDACTED="The driver ########## was questioned."
#DRECONTENT
The driver Joe Bloggs was questioned.
#DREENDDOC
```

If you have specified a plaintext file to perform extraction on, the entities identified as matches by `edktool` are redacted from the input text to form the redacted output. For example:

Input:

```
The driver Joe Bloggs was questioned.
```

Output:

```
The driver ########## was questioned.
```

Eduction sends redacted output to the file specified in the `-r` parameter. If you do not specify this argument but you have enabled redaction in the configuration file, Eduction displays redacted output in the console after the list of matches, unless you have specified the `-q` parameter at the command line to enable Quiet mode. In Quiet mode, redacted output does not display in the console.

`-l <licensefile>` — The file containing a valid license key for Eduction.

If you do not specify a license key at the command line, `edktool` assumes that the location of the license file is `licensekey.dat`. If the license is kept in this location, you do not need to specify this parameter.

`-i <inputfile>` — The file to perform entity extraction on. The input file can be either an IDOL IDX file, an IDOL XML file, or a plain text file. It must be UTF-8 encoded.

> **NOTE:**
> If the input file is an XML file, the configuration file (in either IDOL configuration file format or XML format) must contain entries for the `DocumentDelimiterCSVs` parameter. If this setting is not correct, Eduction might not find any documents in the XML file. For information on how to set this option, see DocumentDelimiterCSVs, on page 64.

`-c <configfile>` — A configuration file controlling the extraction. The configuration file can be either an IDOL Server style .CFG configuration file or an XML configuration file. See Configuration Files for Eduction Settings, on page 51.

You can specify one or more grammar files and one or more entities in place of a configuration file. Specifying a configuration file overrides the grammar or entity parameters.

`-g <grammarfile>` — A grammar file to use when `-c` is not used.

If you provide a grammar file but do not specify any entities with `-e`, Eduction extracts all entities in the grammar file.

`-e <entity>` — The entities to extract when `-c` is not used. Separate multiple entities with a comma.

`-o <outputfile>` — The file containing the results of the extraction. The content of the optional output file depends on the type of input file provided and whether the `-m` option is used.

If the input file type is an IDOL file and the `-m` option is *not* used, the output file is identical to the input file, except the matched entities are appended to each document as additional fields. This behavior is the same as Eduction running in IDOL.

If the input file is a plain text file or an IDOL file with the `-m` option, the output file is an XML file containing the matched entities.

If the input file is an IDOL file, the output file also contains document information.

`-m` — Produce match results for IDOL input files.

`-q`	(Optional) Sets “Quiet Mode” so that descriptive messages and redacted output are removed, and the output consists of the XML matchlist only (that is, an XML document with all the matches and any configured metadata).

`-r <redaction_file>`	A copy of the input file, with all matches redacted. For example, if you specified an IDX input file, the content is sent to the redaction file as follows, with the redactions made in place:
	`#DREREFERENCE 1`
	`#DRECONTENT`
	`The driver ########## was questioned.`
	`#DREENDDOC`

`-p`	Set this parameter if you want to use a plaintext grammar file rather than an XML grammar file. The plaintext grammar file must be in the format described in Plaintext Grammar File Format , on page 50.
`-b`	Set this parameter to read the input file in binary mode, rather than text mode. If you create a grammar file that only matches entities with Windows (CR LF) line endings and you run `edktool` on Windows, the input file must be read in binary mode for any matches to be found. Micro Focus recommends that you create grammar files capable of handling both Windows and Unix line endings.

The extract option requires an input file (either in IDOL IDX, IDOL XML, or plain text format) and either a configuration file or a grammar file. If you do not provide a configuration file, `edktool` searches the file for any specified entities in the specified grammar (or all entities, if none are specified). For example, in the simplest command line:

```
C:\>edktool e -i myData.txt -g grammar1.ecr,grammar2.ecr
```

`edktool` is invoked with no configuration file. It uses the command-line arguments to process the data file `myData.txt` with the grammar files `grammar1.ecr` and `grammar2.ecr`. Eduction identifies all the entities in the two grammar files, and matches on these. The output is sent to the console in XML format, identifying matches in the data file and using the entity names to generate field names for the matches that contain the matched data. Assuming `myData.txt` is a plain text file, the entire body of the file is matched.

Examples

```
edktool e -i myPlainTextFile.txt -g myGrammar.ecr
```

Extracts all entities in `myGrammar.ecr` from `myPlainTextFile.txt`, sending the output to the console in XML format, with the field names for the matching text automatically generated from the entity names found in `myGrammar.ecr`.

```
edktool e -i myIDOLfile.idx -c myIDOLConfigFile.cfg –o myoutputfile.idx
```

Using the configuration file `myIDOLConfigFile.cfg`, extract entities from the file `myIDOLfile.idx` and direct the output with additional Eduction fields to the file `myoutputfile.idx`.

```
edktool e -i myIDOLfile.idx -c myIDOLConfigFile.cfg –o myoutputfile.xml -m
```

The same as the previous example, except output the match results to an `edktool` XML file.

Benchmark

EDKTool supports a benchmarking mode. This allows a user to run multiple concurrent extraction sessions a number of times in order to test the performance of a grammar. The input document is read once and fed into each session. Timing information is produced once all runs have completed. Note that the input document must be plaintext; IDX files are not supported.

You can use wildcard expressions in the `-e` and `-g` parameters; see Wildcard Expressions in edktool, on page 41 for more information.

`-l <licensefile>`	The file containing a valid license key for Eduction. If you do not specify a license key at the command line, `edktool` assumes that the location of the license file is `licensekey.dat`. If the license is kept in this location, you do not need to specify this parameter.
`-i <inputfile>`	The file to perform entity extraction on. The input file must be plaintext.
`-c <configfile>`	A configuration file controlling the extraction. The configuration file can be either an IDOL Server style .CFG configuration file or an XML configuration file. See Configuration Files for Eduction Settings, on page 51. You can specify one or more grammar files and one or more entities in place of a configuration file. Specifying a configuration file overrides the grammar or entity parameters.
`-g <grammarfile>`	A grammar file to use when `-c` is not used. If you provide a grammar file but do not specify any entities with `-e`, Eduction extracts all entities in the grammar file. Note that the MinScore parameter can only be used if you are using `-c`. This means that, although you can use a grammar that supports scoring as the value for `-g`, no matches will be filtered out based on that.
`-e <entity>`	The entities to extract when `-c` is not used. Separate multiple entities with a comma.
`-d`	(Optional) Detail - causes the output to show the matching strings and indicate where they appear in the input.
`-s <sessions>`	Number of sessions to run concurrently during each iteration of the benchmarking test.
`-n <number>`	Number of iterations of the benchmarking test to run.
`-b`	Set this parameter to read the input file in binary mode, rather than text mode. If you create a grammar file that only matches entities with Windows (CR LF) line endings and you run `edktool` on Windows, the input file must be read in binary mode for any matches to be found. Micro Focus recommends that you create grammar files capable of handling both Windows and Unix line endings.

The benchmarking option runs the specified number of concurrent sessions and iterations and then displays the timing for each run, with a summary showing the total number of observations, maximum and minimum times, and the standard deviation.

Measure

This option measures precision and recall between extraction runs by comparing the *expected results* of entity extraction with the *actual results*.

Expected results are created once and remain as a base reference for ongoing tests. Actual results are generated as required each time a grammar is modified. The two results are compared to generate precision and recall information.

To generate expected results, run `edktool -extract`, and then revise the generated output file so that it contains the correct matches. From then on, `edktool -extract` is used only to create the actual results, and the two files are compared against each other to generate precision and recall information on an ongoing basis.

`-e <expectedfile>`	The expected results file from `edktool -extract`.
`-a <actualfile>`	The actual results file from subsequent extraction runs with modified grammar files.
`-o <resultsfile>`	The results, including precision, recall, and differences.
`-q`	(Optional) Sets “Quiet Mode” so that descriptive messages are removed, and the output consists of only an XML document containing the differences between the expected and actual output.

For more information on how to use the Measure feature to check the effectiveness and performance of your grammar files, refer to *IDOL Expert*.

Example

The following example compares `expected.xml` with `actual.xml` and puts the difference in `difference.xml`, including precision and recall. “Quiet Mode” is enabled, so all descriptive messages are removed from the output.

```
edktool m -e expected.xml -a actual.xml -o difference.xml -q
```

Help

This option lists the valid `edktool` options along with brief descriptions for each.

Plaintext Grammar File Format

Plaintext grammar files must have only a single entity, that consists entirely of headwords. Patterns, synonyms, scoring and so on are **not** supported.

Each line in the grammar file must consist of either a headword, a blank line, or a comment (a line beginning with // that is skipped when the file is read). Whitespace and blank lines are ignored when the file is read.

Configuration Files for Eduction Settings

The Extraction option of `edktool` can take its configuration settings from one for the following file types:

- .CFG file
- XML file

You can use the same Eduction configuration settings in each file format, including wildcard expressions where applicable.

Define Eduction Settings in the .CFG Configuration File

The Eduction configuration settings that can be defined in the IDOL Server format .CFG configuration file are described in Eduction Parameters, on page 55.

The .CFG configuration file consists of several sections that are identified by a phrase in square brackets. Each section contains parameters (name/value pairs). For example:

```
[Eduction]
ResourceFiles=C:\MyGrammar\gram1.ecr
```

To define Eduction settings in the .CFG configuration file

1. Open the .CFG configuration file in a text editor.
2. Set the Eduction parameters as required. The following parameters are available (see Example Configuration File, on page 35 and Eduction Parameters, on page 55 for more information):
3. Set the following parameters in the `[Server]` section of the configuration file. These settings are critical for the correct reading of documents.

   ```
   CantHaveFieldCSVs
   DocumentDelimiterCSVs
   ```
4. Save and close the configuration file.

Modify Configuration Parameter Values

The following section describes how to enter parameter values in the configuration file.

Enter Boolean Values

The following settings for Boolean parameters are interchangeable:

```
TRUE = true = True = ON = on = Y = y = 1

FALSE = false = False = OFF = off = N = n =0
```

Enter String Values

Some parameters require string values that contain quotation marks. Percent-encode each quotation mark by inserting a backslash before it.

For example:

```
FIELDSTART0="<font face=\"arial\"size=\"+1\"><b>"
```

Here, the beginning and end of the string are indicated by quotation marks, while all quotation marks that are contained in the string are percent-encoded.

If you want to enter a comma-separated list of strings for a parameter, and one of the strings contains a comma, you must indicate the start and the end of this string with quotation marks.

For example:

ParameterName=**cat,dog,bird,"wing,beak",turtle**

If any string in a comma-separated list contains quotation marks, you must put this string into quotation marks and percent-encode each quotation mark in the string by inserting a backslash before it.

For example:

ParameterName="**<font face=\"arial\"size=\"+1\"><b>",dog,bird,"wing,beak",turtle**

Sample Configuration File

The following shows the configuration for a sample Eduction task:

```
[Eduction]
ResourceFiles=C:\MyGrammar\gram1.ecr,C:\MyGrammar\gram2.ecr
ZoneStart0=<TEXT>
ZoneEnd0=</TEXT>
ZoneStart1=acknowledgements
ZoneEnd1=introduction
Entity0=common/aus_holidays
EntityField0=HOLIDAYS
EntityZone0=0
Entity1=common/us_holidays
EntityField1=HOLIDAYS
EntityZone1=0
Entity2=us/social_security_number
EntityField2=SS_NUMBER
EntityZone2=1
SearchFields=DRECONTENT
AllowDuplicates=HOLIDAYS

[Logging]
LogLevel=Full
```

This sample uses two grammar files. It searches for all Australian and U.S. holidays in the `DRECONTENT` field between the text *<Text>* and *</Text>*, adding the matches as additional fields `HOLIDAYS`. It also

searches for a single social security number in `DRECONTENT` between the text *acknowledgements* and *introduction* and adds the results as a new field `SS_NUMBER`.

Define Eduction Settings in the XML Configuration File

The Eduction configuration elements that you can define in the XML file are described in Eduction Parameters, on page 55.

The following XML configuration file example shows all the available XML elements:

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Sample Eduction XML configuration file for the edktool utility -->

<Eduction>
    <!-- Global Settings (Defaults shown) -->
    <MatchWholeWord>true</MatchWholeWord>
    <SuppressMatchLogging>false</SuppressMatchLogging>
    <MaxEntityLength>256</MaxEntityLength>
    <AllowOverlaps>false</AllowOverlaps>
    <EnableComponents>false</EnableComponents>
    <OutputSimpleMatchInfo>true</OutputSimpleMatchInfo>
    <MatchCase>true</MatchCase>
    <DocumentDelimiterCSVs>*/DOCUMENT</DocumentDelimiterCSVs>
    <CantHaveFields>
        <CantHaveField>*/DRESTORECONTENT</CantHaveField>
        <CantHaveField>*/CHECKSUM</CantHaveField>
        <CantHaveField>*/DREWORDCOUNT</CantHaveField>
        <CantHaveField>*/DRETYPE</CantHaveField>
        <CantHaveField>*/IMPORTBODYLEN</CantHaveField>
        <CantHaveField>*/IMPORTMETALEN</CantHaveField>
        <CantHaveField>*/IMPORTLINKLEN</CantHaveField>
        <CantHaveField>*/IMPORTTITLELEN</CantHaveField>
        <CantHaveField>*/IMPORTQUALITY</CantHaveField>
        <CantHaveField>*/DREPAGE</CantHaveField>
        <CantHaveField>*/DREFILENAME</CantHaveField>
        <CantHaveField>*/dredoctype</CantHaveField>
        </CantHaveFields>

    <!-- Eduction grammar (resource) files to load -->
    <ResourceFiles>
        <ResourceFile>phone.ecr</ResourceFile>
        <ResourceFile>jargon.ecr</ResourceFile>
    </ResourceFiles>

    <!-- IDOL databases to search. Applies only to IDOL IDX or IDOL XML input
documents -->
    <Databases>
        <Database>Contact</Database>
        <Database>Customer</Database>
```

```
    </Databases>

    <!-- Document fields to search. ignored for plain text input documents
(DRECONTENT is the default) -->
    <SearchFields>
        <SearchField>DREREFERENCE</SearchField>
        <SearchField>DRETITLE</SearchField>
        <SearchField>DRECONTENT</SearchField>
    </SearchFields>

    <!-- Definitions of search zones within a document -->
    <Zones>
        <Zone>
            <Name>Summary</Name>
            <StartPattern>Executive Summary</StartPattern>
            <EndPattern>Introduction</EndPattern>
        </Zone>
        <Zone>
            <Name>Body</Name>
            <StartPattern>Introduction</StartPattern>
        </Zone>
    </Zones>

    <!-- Fields generated from a match. Always required, but applies only to IDOL
IDX or IDOL XML input documents where the output is also a modified IDOL document -
->
    <TargetFields>
        <TargetField>
            <Name>PHONE</Name>
            <AllowDuplicates>false</AllowDuplicates>
            </TargetField>
    </TargetFields>

    <!-- Eduction grammar entities used for searching -->
    <Entities>
        <Entity>
            <Name>phone/all</Name>
            <TargetField>PHONE</TargetField>
            <MatchRange>1,2-4</MatchRange>
            <MinScore>0.5</MinScore>
            <Zone>Summary</Zone>
            <Zone>Body</Zone>
        </Entity>
    </Entities>

</Eduction>
```

If Eduction reads an IDOL XML data file, you must configure `DocumentDelimiterCSVs`, and also at least one entry for the `CantHaveFields` setting. If this is not present, Eduction defaults to `DOCUMENT` and `EDUCTION_DUMMY_FIELD` respectively.

Eduction Parameters

The following parameters can be used in the .CFG configuration file or the XML configuration file.

AllowDuplicates

A list of document fields in which Eduction can write multiple results from a single entity. If you allow multiple results from a single entity by setting AllowMultipleResults, on the next page to `TRUE`, and the input text contains more than one match to an entity, Eduction writes the results to multiple fields with the same name.

This parameter is used only when generating output in IDOL IDX format. It has no effect on XML.

You can specify multiple fields by separating them with commas.

Type:	String
Default:	
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`AllowDuplicates=ANIONIC_SURFACTANTS,PERSON`
See Also:	AllowMultipleResults, on the next page EntityN, on page 70 EntityFieldN, on page 68

AllowMultipleResults

This parameter specifies how many results to return, when Eduction finds multiple matches at the same offset (starting position) in the input text. Eduction returns only one result by default, but you can choose to return all of the matches or up to one per entity.

Set this parameter to one of the following values:

- **`All`** or **`True`**. Eduction returns all results.
- **`OnePerEntity`**. Eduction returns up to one result per entity at each offset.
- **`No`** or **`False`**. Eduction does not return multiple results at the same offset.

> **NOTE:**
> If you are writing the results of Eduction to document fields in IDX format, and you have allowed multiple results for a single entity by setting this parameter to `All` or `True`, you should include the relevant fields in the parameter AllowDuplicates, on the previous page.

This parameter can be useful when the same text has multiple interpretations. For example, if the input text contains the word `Georgia`, this could refer to a person's name, the U.S. state, or the country. By default, Eduction returns only one match. This is appropriate if it is not important to you that `Georgia` has multiple interpretations. Set `AllowMultipleResults=All` to return all three matches. Set `AllowMultipleResults=OnePerEntity` to return one match from each entity.

Example

The following table shows how the results from Eduction change when you set the parameters `AllowMultipleResults` and `AllowOverlaps`.

In this example, the input is "The President of the United States of America is in London today to meet the British Prime Minister", and three entities have been defined:

- entity1 matches political offices, for example "President of the United States".
- entity2 matches corporate titles including "President".
- entity3 matches places including "United States" and "United States of America".

Parameters	AllowOverlaps=False	AllowOverlaps=True
AllowMultipleResults =False	Eduction returns the match "President of the United States" (entity1). The match "President" (entity2) is ignored because it shares the same starting point as "President of the United States" and `AllowMultipleResults=FALSE`. The matches "United States" and "United States of America"	Eduction returns the match "President of the United States" (entity1). The match "President" (entity2) is ignored because it shares the same starting point as "President of the United States" and `AllowMultipleResults=FALSE`. Overlapping matches are allowed, so Eduction returns a match

	(entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	"United States of America" (entity3). The match "United States" (entity3) is ignored because it shares the same starting point as "United States of America" and `AllowMultipleResults=FALSE`.
AllowMultipleResults =OnePerEntity	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2). Although it shares the same starting point as "President of the United States" it is matched by a different entity and `AllowMultipleResults` is set to `OnePerEntity`. The matches "United States" and "United States of America" (entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2). Although it shares the same starting point as "President of the United States" it is matched by a different entity and `AllowMultipleResults` is set to `OnePerEntity`. Overlapping matches are allowed, so Eduction returns a match "United States of America" (entity3). The match "United States" (entity3) is ignored because it shares the same starting point as "United States of America" and `AllowMultipleResults` is set to `OnePerEntity`.
AllowMultipleResults =True	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2) because `AllowMultipleResults=True`. The matches "United States" and "United States of America" (entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	Eduction returns all of the matches. These are "President of the United States" (entity1), "President" (entity2), "United States" (entity3), and "United States of America" (entity3).

Type:	String
Default:	No

Required:	No
Configuration Section:	Any section that you have defined for Eduction settings
Example:	`AllowMultipleResults=All`
See Also:	AllowDuplicates, on page 56 AllowOverlaps, below EntityN, on page 70 EntityFieldN, on page 68 NonGreedyMatch, on page 76

AllowOverlaps

A Boolean value that specifies whether to return more than one match, when Eduction finds overlapping matches that start at different characters (offsets). To return overlapping matches set this parameter to `True`.

> **NOTE:**
> To specify whether to return overlapping matches that have the same offset, use the configuration parameter AllowMultipleResults, on page 57.

Example

The following table shows how the results from Eduction change when you set the parameters `AllowMultipleResults` and `AllowOverlaps`.

In this example, the input is "The President of the United States of America is in London today to meet the British Prime Minister", and three entities have been defined:

- entity1 matches political offices, for example "President of the United States".
- entity2 matches corporate titles including "President".
- entity3 matches places including "United States" and "United States of America".

Parameters	AllowOverlaps=False	AllowOverlaps=True
AllowMultipleResults =False	Eduction returns the match "President of the United States" (entity1). The match "President" (entity2) is ignored because it shares the same starting point as "President of the United States" and `AllowMultipleResults=FALSE`. The matches "United States" and	Eduction returns the match "President of the United States" (entity1). The match "President" (entity2) is ignored because it shares the same starting point as "President of the United States" and `AllowMultipleResults=FALSE`.

	"United States of America" (entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	Overlapping matches are allowed, so Eduction returns a match "United States of America" (entity3). The match "United States" (entity3) is ignored because it shares the same starting point as "United States of America" and `AllowMultipleResults=FALSE`.
AllowMultipleResults =OnePerEntity	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2). Although it shares the same starting point as "President of the United States" it is matched by a different entity and `AllowMultipleResults` is set to `OnePerEntity`. The matches "United States" and "United States of America" (entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2). Although it shares the same starting point as "President of the United States" it is matched by a different entity and `AllowMultipleResults` is set to `OnePerEntity`. Overlapping matches are allowed, so Eduction returns a match "United States of America" (entity3). The match "United States" (entity3) is ignored because it shares the same starting point as "United States of America" and `AllowMultipleResults` is set to `OnePerEntity`.
AllowMultipleResults =True	Eduction returns the match "President of the United States" (entity1). Eduction returns the match "President" (entity2) because `AllowMultipleResults=True`. The matches "United States" and "United States of America" (entity3) are ignored because they overlap with "President of the United States" and `AllowOverlaps=FALSE`.	Eduction returns all of the matches. These are "President of the United States" (entity1), "President" (entity2), "United States" (entity3), and "United States of America" (entity3).

Type:	Boolean

Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`AllowOverlaps=True`
See Also:	AllowMultipleResults, on page 57 NonGreedyMatch, on page 76

CantHaveFieldCSVs

Names of fields Eduction ignores when reading an XML file. Allows you to specify the fields in documents that are discarded before the documents are stored.

To specify multiple fields, separate them with commas (there must be no space before or after a comma). You can use wildcards.

Type:	String
Default:	
Required:	No
Configuration Section:	**Server**
Example:	`CantHaveFieldCSVs=*/STANDARD_HEADER` In this example, any `STANDARD_HEADER` fields that a document contains are discarded before the document is stored in IDOL server.
See Also:	

CaseNormalization

The case conversion to use for all incoming text. To improve performance, use this parameter to convert all text to lowercase or uppercase before attempting to match text.

This parameter takes one of the following values:

- **None**. No case conversion.
- **Lower**. All incoming text is converted to lowercase.
- **Upper**. All incoming text is converted to uppercase.

If your grammar file consists of only lowercase or only uppercase characters but your text is mixed case, you can improve performance by setting `CaseNormalization` to **`Lower`** or **`Upper`** respectively. This provides a greater performance improvement than setting `MatchCase` to **`False`**.

If you set this parameter to `Lower` or `Upper`, set `MatchCase` to **`True`**.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`CaseNormalization=lower`
See Also:	CaseNormalizationBehavior, below CaseSensitiveFieldName, below MatchCase, on page 73

CaseNormalizationBehavior

Specifies the algorithm to use for case normalization. This parameter accepts one of the following values:

- **Default**. The default behavior.
- **Turkic**. Use this option with Turkic languages to ensure that case normalization performs correctly with the dotted and dotless "i" characters.

Type:	String
Default:	Default
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`CaseNormalizationBehavior=Turkic`
See Also:	CaseNormalization, on the previous page

CaseSensitiveFieldName

A Boolean that specifies whether to preserve the case of configured field names. By default, the Eduction module converts all field names to uppercase when it produces matches. To preserve the case of the field names, set this parameter to **`True`**. This option makes field names case sensitive.

Type:	Boolean
Default:	False

Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`CaseSensitiveFieldName=True`
See Also:	

CJKNormalization

This parameter allows you to specify how to normalize Chinese, Japanese, and Korean data before extraction, in all Eduction components.

You can specify the value of `CJKNormalization` as follows:

- `Kana`. Half width kana to full width kana.
- `OldNew`. Old kanji to new kanji.
- `Number`. Chinese or kanji number characters to ASCII number characters.
- `HWNum`. Full width number characters to ASCII number characters.
- `HWAlpha`. Full width alphabet characters to ASCII alphabet characters.
- `SimpChi`. Traditional Chinese to simplified Chinese.
- `FWJamo`. Half width jamo to full width jamo.

Separate multiple options with a comma.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`CJKNormalization=SimpChi,Kana`
See Also:	

Databases

The names of the databases to which a document belongs. Eduction runs only on documents that belong to the comma-separated list of databases. If you do not list databases, Eduction is run on documents from all databases.

> **NOTE:**
> If an IDX does not have a `DREDBNAME` entry for a document, matching is not done on that document. However, if all databases are selected, matching is done.

Type:	String
Default:	
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`Databases=DB1,DB2,DB3`
See Also:	EntityN, on page 70 EntityFieldN, on page 68

DocumentDelimiterCSVs

Specifies the fields in an XML file that mark the start and end of an IDOL document. You must have only one document level for each XML schema.

When identifying fields use the formats:

- `FieldName` to match root-level fields.
- `*/FieldName` to match all fields except root-level.
- `Path/FieldName` to match fields that the specified path points to.

Type:	String
Default:	*/DOCUMENT
Required:	No
Configuration Section:	**Server**
Example:	`DocumentDelimiterCSVs=*/DOCUMENT,*/SPEECH` In this example, the beginning and end of individual documents in a file is marked by opening and closing `DOCUMENT` and `SPEECH` tags.
See Also:	

EnableComponents

Set this parameter to **`False`** to return only the entity. Set it to **`True`** to return the entity and all the components of the entity.

This parameter requires `OutputSimpleMatchInfo` to be set to **`False`**.

Type:	Boolean

Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`EnableComponents=True`
See Also:	OutputSimpleMatchInfo, on page 77

EnableUniqueMatches

A Boolean that specifies whether to return only unique matches in each document. To return a single occurrence of a particular value, set this parameter to **`True`**. When `EnableUniqueMatches=True`, two `Entity`*`N`* definitions cannot return the same value, even if they use different patterns. If the same value occurs more than once, only the first instance is returned, even if the matches occur for different entities.

Duplicates display by default unless you set `EnableUniqueMatches` to **`True`** to explicitly remove them.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`EnableUniqueMatches=True`
See Also:	

Entities

A list of entities that you want to modify using the post processing script. If you do not set this parameter, you can use the script to modify the matches for every entity.

You can separate multiple entities with a comma, or, you can use wildcard expressions. You can use the * wildcard to match any number of characters, or the ? wildcard to match a single character. For example, set `Entities` to `phone/*` to apply the script to the `phone/landline/gb`, `phone/mobile/gb` entities and so on.

Type:	String
Default:	None
Required:	No
Configuration	Any section that you have defined for an Eduction post processing task.

Section:	

Example:	`Task0=EductionLuaPostProcessing`
	
	`[EductionLuaPostProcessing]`
	`Script=scripts/eduction_post_process.lua`
	`Entities=phone/landline/gb,phone/mobile/gb`

See Also:	Script, on page 81 ProcessEnMasse, on page 78 TaskN, on page 83

EntityAdvancedField*N*

A comma-separated list of advanced fields to return.

To use this option you must:

- set `OutputSimpleMatch` to **`False`** for edktool.
- set `EnableComponents` to **`True`** for edktool.
- define components in the entity definition.

You configure `EntityAdvancedField`*N* in the same way as EntityFieldN. Specify a comma-separated list of advanced fields that you want to return. The value of the advanced field is the output of simple operations (`min`, `max`, `sum`, and `ave`) on the values of entity components.

For example, for the following configuration:

```
Entity0=testgrammar/testentity
EntityField0=FIELD0
EntityAdvancedField0=OfferPrice:max(price1 price2),BidPrice:min(price1 price2)
```

And the following data:

```
share price1 price2
Com1  165    167
Com2  1890   1880
```

An entity with the following pattern:

```
<grammar name="testgrammar">
<entity name="testentity" type="public">
<pattern>(?A=price1:\d+)\s+(?A=price2:\d+)</pattern>
</entity>
</grammar>
```

Returns the following results as fields:

```
#DREFIELD FIELD0="165 167"
#DREFIELD OfferPrice="167"
#DREFIELD BidPrice="165"
#DREFIELD FIELD0="1890 1880"
```

```
#DREFIELD OfferPrice="1890"
#DREFIELD BidPrice="1880"
```

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings

Example:	`Entity0=testgrammar/testentity`
	`EntityField0=FIELD0`
	`EntityAdvancedField0=OfferPrice:max(price1 price2),BidPrice:min(price1 price2)`

See Also:	EntityN, on page 70 EntityZoneN, on page 72

EntityComponentField*N*

A comma-separated list of entity components that you want to return as fields.

To use this option you must:

- set `OutputSimpleMatch` to **`False`** for edktool.
- set `EnableComponents` to **`True`** for edktool.
- define components in the entity definition.

You configure `EntityComponentField`*N* in the same way as EntityFieldN. Specify a comma-separated list of entity components that you want to return as fields.

For example, for the following configuration:

```
Entity0=testgrammar/testentity
EntityField0=FIELD0
EntityComponentField0=Name,Age
```

And the following data:

```
name    age
geoff   45
jane    54
```

An entity with the following pattern:

```
<grammar name="testgrammar">
<entity name="testentity" type="public">
<pattern>name\s+age(\n(?A=Name:\w+)\s+(?A=Age:\d+)){1,}</pattern>
</entity>
</grammar>
```

Returns the following values as fields:

```
#DREFIELD Name="geoff"
#DREFIELD Age="45"
#DREFIELD Name="jane"
#DREFIELD Age="54"
```

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`Entity0=testgrammar/testentity`
	`EntityField0=FIELD0`
	`EntityComponentField0=Name,Age`

See Also:	EntityN, on page 70 EntityZoneN, on page 72

EntityField*N*

A comma-separated list of document fields to associate with the entities specified by the `EntityN` parameter. If entities are identified in a document, the text is saved in the fields specified by this parameter. The entity field number *N* must match the corresponding `Entity`*N* number.

A many-to-many relationship exists between the `Entity`*N* and `EntityField`*N* parameters. If an `Entity`*N* setting does not have an `EntityField`*N* setting, text matching the entity is not passed to Eduction.

If no `Entity`*N* settings are provided, `EntityField`*N* settings are ignored, because Eduction automatically generates `EntityField`*N* settings corresponding to each EntityN, on page 70 setting that exists in the selected grammars.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`Entity0=edk_common_entities/ss_number`
	`EntityField0=SOCIAL_SECURITY_NUMBER`
	
	`Entity1=edk_common_entities/postal_address`
	`EntityField1=SHIPPING_ADDRESS`

See Also:	EntityN, on the next page EntitySearchFieldsN, on page 71 EntityZoneN, on page 72

EntityMatchRange*N*

A range of matching instances of the entity that are returned. The entity match range number *N* must match the corresponding EntityN number. The format of the range is as follows:

```
<match>\[{-\|,}<match>\]\[,...\]*
```

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`Entity0=edk_common_entities/ss_number`
	`EntityMatchRange0=1-3,6,9-`
	This example specifies the first through third match for the `ss_number` entity, as well as the sixth match and all matches starting with the ninth.

See Also:	EntityN, on the next page

EntityMinScore*N*

The minimum score that is required for a match to be returned. The lowest possible score is 0. The upper limit varies depending on the entity.

As the minimum score is increased above **0**, towards (and even past) **1**, you are indicating that matches must meet a higher confidence level to be returned.

> **NOTE:**
> The minimum score threshold is applied before Eduction runs any post-processing tasks (seePost-Processing, on page 87). If a post-processing task reduces the score for a match so that it is lower than the threshold specified by this parameter, the match is not discarded. You could discard the match in the post-processing task, or run an additional post-processing task to check the scores on all matches and discard those below a certain value.

The *score* for an entity is defined by the author of the grammar and defaults to **1**. See the Eduction Grammar Syntax for a description of the score attribute.

The entity number (*N*) in `EntityMinScore`*N* must match the corresponding entity number in the `Entity`*N* entry.

Type:	Long
Default:	0 (returns all matches)
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	To specify a minimum score of `0.5` for `Entity0`:
	`Entity0=edk_common_entities/ss_number`
	`EntityMinScore0=0.5`

See Also:	EntityN, below MinScore, on page 76

Entity*N*

A comma-separated list of entities to extract. Entities are defined in the resource file identified in the ResourceFiles parameter. Replace *N* with the zero-based rank of the entity.

You must associate each entity with a field by using the EntityFieldN parameter.

You cannot use the entity name `entities/ZoneStart`*N* or `entities/ZoneEnd`*N* (where *N* is a numeric value). These entity names are reserved for use by Eduction.

If you do not define an `Entity`*N* parameter, Eduction looks for all entities in all loaded grammar files. In this case, the `EntityField`*N* settings are automatically generated from the entities found in grammar files by converting the entity names to uppercase and replacing slashes with an underscore. For example, if the entity `edk_common_entities/place` is found, Eduction generates the entity field: `EDK_COMMON_ENTITIES_PLACE`.

If you want to use several entities, you can use wildcard expressions instead of typing a lengthy comma-separated list. For example:

```
Entity0=place/city1/*,place/city2/*
EntityField0=CITY
Entity1=place/*/spabo
EntityField1=BOLIVIAN_PLACE
```

You can use the * wildcard to match any number of characters, or the ? wildcard to match a single character.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`Entity0=edk_common_entities/ss_number`
	`EntityField0=SOCIAL_SECURITY_NUMBER`
	`EntityZone0=0`
	
	`Entity1=edk_common_entities/postal_address`
	`EntityField1=SHIPPING_ADDRESS`
	`EntityZone1=1`
	
	`ZoneStart0=Social Security:`
	`ZoneEnd0=Shipping Address`
	
	`ZoneStart1=Shipping Address:`
	`ZoneEnd1=Billing Address`

See Also:	EntityFieldN, on page 68 EntityMinScoreN, on page 69 EntitySearchFieldsN, below EntityZoneN, on the next page

EntitySearchFields*N*

The `EntitySearchFields`*N* parameter specifies the document fields to search for the corresponding `Entity`*N* entity.

Use this parameter if you want to search a different set of fields than is specified by SearchFields, on page 81. If you do not set `EntitySearchFields`*N*, Eduction searches the fields specified by the `SearchFields` parameter.

Type:	String
Default:	The value of SearchFields, on page 81
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings

Example:	In the following example, matches for `Entity0` (`airport/icao`) are returned only if they occur in the `STARTAIRPORT` or `DESTAIRPORT` fields.
	
	The `EntityFields`*N* parameter is not set for `Entity4` (`place/state/engus`), so matches are returned if they are present in the fields specified by the `SearchFields` parameter.
	
	`[Eduction]`
	`SearchFields=DRECONTENT`
	
	`Entity0=airport/icao`
	`EntityField0=AIRPORTCODE`
	`EntitySearchFields0=STARTAIRPORT,DESTAIRPORT`

	Entity1=person/femalefirstname/engus EntityField1=FIRSTNAME EntitySearchFields1=PASSENGER_FIRSTNAME Entity2=person/malefirstname/engus EntityField2=FIRSTNAME EntitySearchFields2=PASSENGER_FIRSTNAME Entity3=person/lastname/engus EntityField3=SURNAME EntitySearchFields3=PASSENGER_SURNAME Entity4=place/state/engus EntityField4=STATE
See Also:	EntityN, on page 70 EntityFieldN, on page 68 SearchFields, on page 81

EntityZone*N*

Associates an EntityN entity with one or more zones defined using the ZoneStartN and ZoneEndN parameters. Type the number of the `ZoneStart`*`N`* and `ZoneEnd`*`N`* parameters to associate with the EntityN. Eduction searches for the entity in the specified zones. The entity zone number *N* must match the corresponding EntityN number.

Type:	Long
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings
Example:	Entity0=edk_common_entities/ss_number EntityField0=SOCIAL_SECURITY_NUMBER EntityZone0=0 Entity1=edk_common_entities/postal_address EntityField1=SHIPPING_ADDRESS EntityZone1=1 ZoneStart0=Social Security: ZoneEnd0=Shipping Address ZoneStart1=Shipping Address: ZoneEnd1=Billing Address
See Also:	ZoneEndN, on page 84 ZoneStartN, on page 85

LanguageDirectory

Enables tokenization of Chinese, Japanese, Korean, and Thai languages. Set `LanguageDirectory` to the path of an IDOL Server language directory that contains the relevant sentence breaking libraries and associated data files.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`C:\Program Files\IDOLServer\IDOL\langfiles`
See Also:	Locale, below

Locale

Enables tokenization of Chinese, Japanese, Korean, and Thai languages. Set `Locale` to one of **`CHI`**, **`JPN`**, **`KOR`**, or **`THA`**.

> **NOTE:**
> The standard grammar files are developed without this setting; Micro Focus recommends that you use this parameter only when you are using custom grammar files that have been developed with the specific tokenization.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`Locale=THA`
See Also:	LanguageDirectory, above

MatchCase

By default, Eduction is case sensitive when matching characters. This default value applies only when no grammar-specific case attribute has been specified for an entity.

To ignore case when matching characters, set this parameter to **`False`**.

Type:	Boolean
Default:	True
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`MatchCase=False`
See Also:	

MatchTimeout

The maximum amount of time (in seconds) to spend searching for matches (to all chosen entities) at a specific offset. If the timeout is reached, Eduction returns the best match it has found (if any) and continues looking for matches later in the text.

Eduction usually finds matches very quickly, so in most cases setting this timeout is not required.

Type:	Integer
Default:	60
Required:	No
Configuration Section:	**Eduction**
Example:	`MatchTimeout=30`
See Also:	RequestTimeout, on page 80

MatchWholeWord

To match only terms in the text that begin and end on a whole word boundary, set this parameter to **`True`**.

To match terms that start and end anywhere, including in the middle of a word in the text, set this parameter to **`False`**.

For example, if `MatchWholeWord=True`, a search for the term `80` does not find a match in the text string `80mph`. If `MatchWholeWord=False`, a search for the term `par` finds a match in the text string `separated`.

For more information on modifying the matching behavior by using `MatchWholeWord`, refer to *IDOL Expert*.

Type:	Boolean
Default:	True

Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`MatchWholeWord=False`
See Also:	TangibleCharacters, on page 82 TokenWithPunctuation, on page 83

MaxEntityLength

The maximum number of characters in a returned entry.

Reducing this number can assist performance by preventing Eduction from scanning a long string of text for an entity that is expected to be small.

Type:	Integer
Default:	256
Allowed Range:	Minimum: 1 Maximum: 1024
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`MaxEntityLength=100`
See Also:	EntityN, on page 70 ZoneEndN, on page 84 ZoneStartN, on page 85

MaxMatchesPerDoc

The maximum number of matches to allow in each document.

Type:	Integer
Default:	Unlimited
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`MaxMatchesPerDoc=15`
See Also:	

MinScore

The minimum score that is required for a match to be returned. The lowest possible score is 0. The upper limit varies depending on the entity.

As the minimum score is increased above **0**, towards (and even past) **1**, you are indicating that matches must meet a higher confidence level to be returned.

This parameter applies to all entities. You can also set EntityMinScoreN, on page 69, which applies to the entities specified by the corresponding EntityN, on page 70 parameter. If you set both parameters, a match is only returned if it exceeds both thresholds.

> **NOTE:**
> This threshold is applied before Eduction runs any post-processing tasks (see Post-Processing, on page 87). To filter matches after all post-processing tasks have completed, use the parameter PostProcessThreshold, on page 78.

Type:	Long
Default:	0 (all matches are returned)
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`MinScore=0.5`
See Also:	EntityMinScoreN, on page 69

NonGreedyMatch

A Boolean that specifies whether to return the shortest match. To configure Eduction to return the shortest match, set `NonGreedyMatch` to **`True`**. If two matches from two different entities start at the same word, and `NonGreedyMatch` is set to **`True`**, Eduction returns only the shortest match.

Setting this parameter to **`True`** implicitly disables the `AllowOverlaps` and `AllowMultipleResults` parameters. If you have set these parameters, `NonGreedyMatch` takes precedence.

For more information on how to configure the Eduction matching behavior using `NonGreedyMatch`, refer to *IDOL Expert*.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`NonGreedyMatch=True`
See Also:	AllowMultipleResults, on page 57 AllowOverlaps, on page 59

NumTasks

The number of post-processing tasks that you want to configure.

See Post-Processing, on page 87 for more information.

Type:	Integer
Default:	None
Required:	No
Configuration Section:	**[PostProcessingTasks]**
Example:	`NumTasks=1`
See Also:	TaskN, on page 83

OutputScores

Set this parameter to **`True`** to include the score associated with a match in the output from an extraction task. If the output is in .IDX format, the score is added as a new `DREFIELD`, with the field name `SCORE`. If the output is in XML format, the score is added as an attribute with the name `"score"`.

> **NOTE:**
> This parameter is used by `edktool` only.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`OutputScores=True`
See Also:	

OutputSimpleMatchInfo

When `edktool` is used with both the extract option and the option to generate a list of matches, setting `OutputSimpleMatchInfo` to **`True`** generates basic match information only, such as document, entity,

position, and original text.

If `OutputSimpleMatchInfo=True`, the `EnableComponents` setting has no effect and reverts to **`False`**.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`OutputSimpleMatchInfo=True`
See Also:	EnableComponents, on page 64

PostProcessThreshold

The minimum score that is required, after all post-processing tasks have completed, for a match to be returned.

The threshold applies to all entities. When you specify a higher threshold, you are indicating that matches must meet a higher confidence level to be returned.

This parameter is similar to the MinScore, on page 76 parameter. Use `PostProcessThreshold` when you want to filter the matches after all post-processing tasks have completed, rather than before post-processing begins.

Type:	Number
Default:	All matches are returned
Required:	No
Configuration Section:	`PostProcessingTasks`
Example:	`[PostProcessingTasks]` `PostProcessThreshold=0.4`
See Also:	MinScore, on page 76

ProcessEnMasse

Configures an en masse post-processing task. If you set `ProcessEnMasse` to **`True`**, your post-processing script takes the entire set of educed matches as its input argument, rather than a single match. The script can thus look at all the matches at once and modify them accordingly.

A Boolean that specifies whether to consider and modify all of the matches at the same time. For example, to increase the score of a match if it is found near other matches, you must consider all of the matches together.

- When `ProcessEnMasse=False`, the Lua post processing script takes an individual match as its input argument.
- When `ProcessEnMasse=True`, the Lua post processing script takes the entire set of matches as its input argument.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for an Eduction post processing task.

Example:	`PostProcessingTask0=EductionLuaPostProcessing`
	
	`[EductionLuaPostProcessing]`
	`Script=scripts/eduction_post_process.lua`
	`ProcessEnMasse=True`

See Also:	Entities, on page 65 Script, on page 81 TaskN, on page 83

RedactedOutput

Set this parameter to `True` to enable redaction of sensitive information in the output text.

You can also set **one** of RedactionOutputString or RedactionReplacementCharacter; if neither are set, the default behavior is to replace redacted text with `[redacted]` in the output. If both are configured, `RedactionReplacementCharacter` takes precedence.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`RedactedOutput=False`
See Also:	RedactionOutputString, below RedactionReplacementCharacter, on the next page

RedactionOutputString

A string that replaces redacted information in the output text.

Type:	String
Default:	[redacted]
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`RedactionOutputString=[censored]`
See Also:	RedactionReplacementCharacter, below

RedactionReplacementCharacter

A single character that replaces each character in redacted text.

Type:	String
Default:	Use [redacted] instead.
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`RedactionReplacementCharacter=*`
See Also:	RedactionOutputString, on the previous page

RequestTimeout

The maximum amount of time (in seconds) to spend processing a single input file or document. If the timeout is reached, Eduction stops processing and returns any results that were found. In most cases the default timeout is not reached, but it can prevent Eduction running for a long time with abnormal documents.

Type:	Integer
Default:	300
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings
Example:	`RequestTimeout=120`
See Also:	MatchTimeout, on page 74

ResourceFiles

The full path to a compiled ECR file containing Eduction grammar entries. At least one resource file is required.

You can specify multiple resource files by separating them with commas, or by using wildcard expressions. You can use the * wildcard to match any number of characters, or the ? wildcard to match a single character. For example, set `ResourceFiles` to `<grammar_files_directory>/sentiment_*.ecr` to use all available sentiment grammars without having to type a lengthy comma-separated list.

Type:	String
Default:	None
Required:	Yes
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`ResourceFiles=C:\MyGrammar\gram1.ecr,C:\MyGrammar\gram2.ecr`
See Also:	

Script

The path to the Lua script that you want to run to process the data returned by the Eduction module. See Post-Processing, on page 87 for more information.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for an Eduction post processing task.
Example:	`Script=./scripts/checksum.lua`
See Also:	Entities, on page 65 ProcessEnMasse, on page 78 TaskN, on page 83

SearchFields

A comma-separated list of fields to search for entities, for example `DRECONTENT` or `DRETITLE`.

To search for a specific entity only in specific fields, you can set EntitySearchFieldsN, on page 71, which overrides the value of this parameter for specific entities.

You must search at least one field, otherwise Eduction does not return any results.

Type:	String
Default:	`DRETITLE,SUMMARY,DRECONTENT`
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings
Example:	`SearchFields=DRECONTENT,DRETITLE`
See Also:	EntityN, on page 70 EntitySearchFieldsN, on page 71

SuppressMatchLogging

Set this parameter to **`True`** to suppress log entries for every entity and zone pattern found in a document.

When logging is set to **`Full`** in the Eduction configuration file, Eduction makes a log entry for every entity and zone pattern found in a document. If you set this parameter to **`True`**, these log entries are suppressed. This option is useful when you want to log the performance timing information, but do not want the verbose match entries.

You can also set this parameter in Eduction Server. If you set logging to **`Full`** in the Eduction Server configuration file, the server records a log entry for every entity match found. You can set `SuppressMatchLogging` to **`True`** to suppress these log entries.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`SuppressMatchLogging=True`
See Also:	

TangibleCharacters

A list of punctuation characters to treat as part of the word, rather than as word boundaries. By default almost all punctuation characters are treated as word boundaries.

> **NOTE:**
> You cannot specify spaces, returns, and tabs as `TangibleCharacters`.

This parameter has no effect when MatchWholeWord is set to **`False`**.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`TangibleCharacters=-/\@`
See Also:	MatchWholeWord, on page 74 TokenWithPunctuation, below

For more information on using `TangibleCharacters` to specify punctuation characters to match, or to match punctuation at the start of a match, refer to *IDOL Expert*.

Task*N*

The name of an Eduction post-processing task to run. This parameter specifies the name of a section in the Eduction configuration file that contains the parameters required to run the task. To run multiple tasks, use numbered parameters (`Task0`,`Task1`, and so on).

You can use a post processing task to modify the output from the Eduction module, or format the output to meet your requirements. See Post-Processing, on page 87 for more information.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.

Example:	`Task0=EductionLuaPostProcessing`
	
	`[EductionLuaPostProcessing]`
	`Script=scripts/eduction_post_process.lua`

See Also:	Script, on page 81 Entities, on page 65 ProcessEnMasse, on page 78 NumTasks, on page 77

TokenWithPunctuation

A Boolean that specifies whether to treat all punctuation characters as part of a word token, rather than treating them as word boundaries. Setting this parameter to **`True`** is equivalent to setting the `TangibleCharacters` parameter to all punctuation characters.

This parameter has no effect when `MatchWholeWord` is set to **`False`**.

Type:	Boolean
Default:	False
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings.
Example:	`TokenWithPunctuation=True`
See Also:	MatchWholeWord, on page 74 TangibleCharacters, on page 82

For more information on using `TokenWithPunctuation` to configure all punctuation marks as tangible characters, refer to *IDOL Expert*.

ZoneEnd*N*

A regular expression that defines the end point of a zone.

A zone is a section of a field defined by a start and end pattern. Zones locate entities in parts of a field. If you do not add zone entries, Eduction searches the entire field. If the end pattern is absent, the search begins at a match for the start pattern and continues until the end of the field.

Use the EntityZoneN parameter to associate an entity identified in an EntityN parameter with one or more zones defined using the ZoneStartN and `ZoneEnd`*N* parameters.

> **NOTE:**
> You must choose start and end patterns that do not match the same text in a field.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings

Example:	`ZoneStart0=Social Security:`
	`ZoneEnd0=Shipping Address`
	
	`ZoneStart1=Shipping Address:`
	`ZoneEnd1=Billing Address`

See Also:	EntityN, on page 70 EntityZoneN, on page 72 ZoneStartN, on the next page

ZoneStart*N*

A regular expression that defines the start point of a zone.

A zone is a section of a field defined by a start and end pattern. Zones locate entities in parts of a field. If you do not add zone entries, Eduction searches the entire field. If the start pattern is absent, the search begins at the start of the field and continues until a match for the end pattern.

Use the EntityZoneN parameter to associate an entity identified in an EntityN parameter with one or more zones defined using the `ZoneStart`*`N`* and ZoneEndN parameters.

> **NOTE:**
> You must choose start and end patterns that do not match the same text in a field.

Type:	String
Default:	None
Required:	No
Configuration Section:	Any section that you have defined for Eduction settings

Example:	`ZoneStart0=Social Security:`
	`ZoneEnd0=Shipping Address`
	
	`ZoneStart1=Shipping Address:`
	`ZoneEnd1=Billing Address`

See Also:	EntityN, on page 70 EntityZoneN, on page 72 ZoneEndN, on the previous page

Match Validity

The order in which a match is assessed for its validity is as follows:

- If the match is not found inside one of the required zones, discard it.
- If the match does not meet the minimum score requirement, discard it.
- If duplicates are allowed:
  - If the instance of the match is allowable, count this instance and return the match.
  - Otherwise, count this instance and discard it.
- If duplicates are not allowed for the entity field:

- If the matched text has been found before, discard it.
- Otherwise, if the instance of the match is allowable, count this instance and return the matched text.
- If the instance of the match is not allowable, count this instance and discard it.

Order of Returned Matches

When multiple fields in a document are selected for parsing, the field order in which matches are returned is as follows:

- `DREREFERENCE`
- `DRETITLE`
- `DRECONTENT`
- Any remaining fields in the order in which they are specified.

Chapter 6: Post-Processing

This section describes post-processing.

Introduction

Post-processing performs additional processing on the matches that are found by Eduction.

A common reason for post-processing is to validate matches. Some entities, such as credit card numbers, can be validated by calculating a checksum. A match with an invalid checksum can be discarded, because even though it matches the correct format, it cannot be genuine. If a match has a valid checksum then you might increase its score, because it is likely to be valid.

Another reason for post-processing is to normalize the output from Eduction. For example, if you are extracting monetary values Eduction might find matches that look like "£5.3 million" or "£25". You can use post-processing to normalize these values to "£5,300,000" and "£25", so that IDOL Content or another application can compare and sort the values correctly.

A post-processing task passes the matches found by Eduction into a Lua function, either one at a time or en masse (for more information about processing matches en masse, see Write a Lua Script for Post-Processing, on the next page).

Configure Post-Processing in Eduction Server

Post-processing tasks are configured in the `[PostProcessingTasks]` section of the Eduction Server configuration file. Use the `NumTasks` parameter to specify the number of tasks and use the `Task`*`N`* parameter to specify names for the tasks. Start numbering the tasks from zero. For example:

```
[PostProcessingTasks]
NumTasks=2
Task0=ValidateWithChecksum
Task1=FilterScore
```

Then, create a section for each of the tasks using the names that you defined:

```
[ValidateWithChecksum]
Type=lua
Entities=number/creditcard
Script=./scripts/checksum.lua
ProcessEnMasse=FALSE
```

The `Entities` parameter specifies the entities to process. You can use wildcards to match multiple entities. The `Script` parameter specifies the path to the Lua script that you want to run. Eduction includes some example scripts, and you can write your own. For information about how to write a post-processing script, see Write a Lua Script for Post-Processing, below.

For more information about Eduction Server configuration parameters, refer to the *Eduction Server Reference*.

Post-Processing with the Eduction API

To perform post-processing in an application built on the Eduction API, add post-processing tasks to the Eduction engine. You can add the tasks:

- At the time the engine is created, if you construct the engine by supplying a configuration. When you create an Eduction engine from a configuration, by calling `EdkEngineCreateFromConfigFile` in the C API, or using the `EdkEngine` constructor in the Java API, any post-processing tasks in the configuration are added to the engine automatically.
- After the engine is created, by calling the appropriate function:
  - `EdkEngineAddPostProcessingTask` in the C API.
  - `addPostProcessingTask` in the Java API.

To specify the minimum score that is required for a match to be returned, after all post-processing tasks have completed, you can use the following functions:

- `EdkEngineSetPostProcessingThreshold` in the C API.
- `setPostProcessingThreshold` in the Java API.

The matches returned by the `EdkGetNextMatch` function in the C API, or by iterating over the matches in the Java API, reflect any modifications made by post-processing. If a post-processing task discards a match or its score does not meet the threshold you have specified, it is not returned at all.

If you configure a post-processing task that processes matches en masse, the API does not return matches until all input has been received. This is necessary because an en masse post-processing task requires all of the matches at the same time.

The Eduction SDK includes reference documentation for the API. For more information about the SDK, see Eduction SDK, on page 25.

Write a Lua Script for Post-Processing

An Eduction post-processing task runs a Lua script.

Your script must define a function named `processmatch`, which takes a single argument, an `edkMatch` object. The matches that are found by Eduction are passed into the script one at a time. The script must return a Boolean value: `true` to keep the match or `false` to discard it.

The following example changes the score for every match to `0.5`:

```
function processmatch(edkmatch)
    if edkmatch then
```

```
        -- change the score for the match
        edkmatch:setScore(0.5)
    end
    return true
end
```

Process Matches En Masse

Sometimes, you might prefer to process all of the matches together. For example, you might want to increase the scores of matches that appear near other matches. It is easier to do this if you process all of the matches at the same time.

To process all of the matches at the same time, modify your Eduction configuration and set the parameter `ProcessEnMasse` to `TRUE`. When `ProcessEnMasse=TRUE`, all of the matches found by Eduction are passed into the script together.

Your script must define a function named `processmatches`, which takes a single argument, a Lua table of `edkEnMasseMatch` objects. Each of these objects represents a single match, but you must call the `getMatch` method to obtain an `edkMatch` object. You can then use the `edkMatch` object to manipulate the match. If you want to discard a match, call the method `setOutput` on the relevant `edkEnMasseMatch` object.

The following example demonstrates how to iterate over the elements in the table and discard any match with a score that is less than `0.5`:

```
function processmatches(matches)
    -- example that discards matches with score < 0.5
    for k,v in ipairs (matches) do
        local edkmatch = v:getMatch()
        if edkmatch:getScore() < 0.5 then
            v:setOutput(false)
        end
    end
end
```

For information about the objects and methods that you can use in your Lua post-processing scripts, see Eduction Lua Methods Reference, on page 284.

Pass Parameters into the Lua Script

You can pass additional parameters into post-processing tasks that you run through the Eduction API. To add an additional parameter (to all post-processing tasks that run during the session), call the appropriate function:

- `EdkSessionSetUserParamValue` in the C API.
- `setUserParamValue` in the Java API.

Any parameters that you set using these functions are passed into the `processmatch` or `processmatches` function of the Lua script as a table of key-value pairs. For example:

```
function processmatch(edkmatch, params)
    for k,v in pairs (params) do
        --print ("Custom parameter ", k, " has value ", v)
    end
    return true
end
```

Example Scripts

Eduction includes the following example post-processing scripts.

Checksum Validation

The `checksum_luhn.lua` script verifies the checksum digit of each match using the *Luhn algorithm*, and reduces the score associated with the match if the checksum is wrong. The `checksum_luhn_enmasse.lua` script performs checksum validation as an en masse processing task, discards incorrect matches, and alters the score of correct matches to equal the proportion of matches that have the correct checksum digit.

You can use these scripts with the `number_cc.ecr` and `number_sin_ca.ecr` grammar files to validate most credit card numbers.

Spanish Identity Card Number Validation

You can use the `checksum_dni_es.lua` script with the `number_dni_es.ecr` grammar file to validate Spanish Documento Nacional de Identidad (national identity card) numbers.

Dutch Citizen Service Number Validation

You can use the `checksum_bsn_nl.lua` script with the `number_bsn_nl.ecr` grammar file to validate Dutch Citizen Service Numbers (Burgerservicenummer, or BSNs).

Geographical Coordinate Standardization

You can use the `lat_long.lua` script with the `place_lat_long.ecr` grammar file to convert and standardize the output of geographical coordinates.

Date and Time Standardization

You can use the `datetime.lua` script with the `datetime_advanced_eng.ecr` grammar file to convert and standardize the output of dates and times (and ranges) in English into a standardized format in cases where there are matches on several formats. For example, you can convert both *23/11/13* and *Nov 23 2013* to *2013-11-23*.

The `datetime_advanced_eng.ecr` grammar file can understand English natural language, and relative dates such as *last Saturday morning*. You can provide a reference date for `<today>` in the Lua script to enable normalization of relative dates into standard formats.

For date and time range matches, this script sets the normalized text to `<start>/<end>`, and additionally adds `STARTPOINT` and `ENDPOINT` components that contain the associated dates or times. When there is a multiple date match (for example, *5th and 8th July* matches as *5th July* and *8th July*), the script returns a comma-separated list, with a POINT component for each date.

Filter Matches by Case

You can use the `case_filter.lua` example script to filter out matches by case, for example in personal name grammars.

To use this option, you must set `MatchCase` to **`False`** for the grammar. The script filters out any match that is not one of:

- an exact match as specified in the grammar.
- an upper case match (for example, JANE SMITH).
- a title case match (for example Jane Smith).

> **NOTE:**
> You might need to update this script to include case mappings for uncommon non-ASCII characters. The script provides sample mappings for common Latin characters with diacritics.

Chapter 7: Standard Grammars

This chapter contains specific information concerning the standard grammars that come with Eduction.

File Names

File names consist of up to four parts:

- **Basic entity type**. For example, *place*, *number,* or *person*.
- **Further detail on the basic type**. For example, *malefirstname* or *ss* for Social Security number. This part is optional, and is preceded by an underscore.
- **Language**. The three-character ISO 639-2/B code in which the grammar was written. For example, *eng* for English. It is preceded by an underscore.
- **Country**. The two-character ISO 3166-1 code describing the country for which the grammar was written. For example: *us* for the United States. This part is optional if the grammar does not target a specific country (for example, a credit card number). It is preceded by an underscore.

> **NOTE:**
> Entity names follow the same four-part structure, except for the basic type. The further detail and language/country parts are separated by forward slashes. The language code and the optional country code are concatenated.

Sentiment Grammars

Eduction includes standard grammars designed to identify those phrases in a passage of text that indicate positive or negative sentiment. These grammars can also identify which sentiments are expressed for which topics.

The sentiment grammar files have ‘lite’ counterparts. These can process data up to twice as fast compared to the full versions, depending on language. The ‘lite’ versions are identical to the full versions in most respects, but they do not support components or user modification. Micro Focus recommends that you use the ‘lite’ versions except in cases where you want to enable components or modify the built-in dictionaries.

The 'lite' versions are distinguished from the full versions by the addition of *lite* to the file name, preceded by an underscore. For example, the file name of the Chinese sentiment grammar file is `sentiment_chi.ecr`, and the file name of the ‘lite’ version is `sentiment_chi_lite.ecr`.

All sentiment analysis grammar files except `sentiment_basic_eng.ecr` now support components. You can extract the SENTIMENT and TOPIC components in most matches.

Polarity Scoring

The Arabic, Chinese, Czech, English, French, German, Italian, Polish, Portuguese, Russian, Spanish, and Turkish sentiment grammars (`sentiment_ara.ecr`, `sentiment_chi.ecr`, `sentiment_cze.ecr`, `sentiment_dut.ecr`, `sentiment_eng.ecr`, `sentiment_fre.ecr`, `sentiment_ger.ecr`, `sentiment_ita.ecr`, `sentiment_pol.ecr`, `sentiment_por.ecr`, `sentiment_rus.ecr`, `sentiment_spa.ecr`, and `sentiment_tur.ecr`) support *polarity scoring*. This is a number, usually between 0.50 and 1.50, that represents the strength of the sentiment in the matched phrase. For example:

- a strongly positive or negative phrase might have a score of 1.35
- a typical phrase might have a score of 1.00
- a match where the sentiment is weak or ambiguous might have a score of 0.60

You can edit the user modification files (`sentiment_user_ara.xml`, `sentiment_user_chi.xml`, `sentiment_user_cze.xml`, `sentiment_user_dutch.xml`, `sentiment_user_eng.xml`, `sentiment_user_fre.xml`, `sentiment_user_ita.xml`, `sentiment_user_pol.xml`, `sentiment_user_por.xml`, `sentiment_user_rus.xml`, `sentiment_user_spa.xml`, and `sentiment_user_tur.xml`) to increase the scores of words in the dictionaries. For example, add the following on a new line in the user modification file to modify the existing entry `flexible` so that it has a score of `1.23`:

```
"      <entry score="1.23" headword="flexible"/>"
```

> **NOTE:**
> `sentiment_basic_eng.ecr` does not support polarity scoring.

Verb Sentiment Transitivity

The sentiment analysis files for Arabic, Chinese, Czech, English, French, German, Italian, Polish, Portuguese, Russian, Spanish, and Turkish (`sentiment_ara_ecr`, `sentiment_chi.ecr`, `sentiment_cze.ecr`, `sentiment_eng.ecr`, `sentiment_fre.ecr`, `sentiment_ger.ecr`, `sentiment_ita.ecr`, `sentiment_pol.ecr`, `sentiment_por.ecr`, `sentiment_rus.ecr`, `sentiment_spa.ecr`, and `sentiment_tur.ecr`) support *verb sentiment transitivity*. This enables the TOPIC components of the matches to determine what the sentiment is about with more accuracy by using advanced contextual understanding of whether that sentiment is being expressed about the subject or object of the sentence. For example, given two matches, `x likes y` and `x wins at y`, the grammar files can determine that the first match is a positive statement about `y`, whereas the second match is a positive statement about `x`.

Place Name Disambiguation

Ambiguous names in all place grammars have been given a score of 0.98 so that you can filter them out by setting `EntityMinScore`*N* to **`0.99`**. For example, if you want to use the `place/state/engau` entity to extract Australian state names using the `place_engau.ecr` grammar file, you can set `EntityMinScore`*N* to **`0.99`** to filter out ambiguous names such as *Victoria*.

Standard Grammar – Compiled

The following sections list the compiled grammar files included with Eduction.

> **NOTE:**
> All the Chinese grammar files support traditional Chinese.

A

address_au.ecr

Entity	Description
address/postcode/au	Australian postal codes. For example, *2600*.
address/state_postcode/au	Australia state or territory, and postal code. For example, *NSW 2060*.
address/city_state_postcode/au	Australian city, state or territory, and postal code. For example, *North Sydney, NSW 2060*.
address/au	Any Australian address. For example: *Shop 17, Winnellie Shopping Centre, 347 Stuart Hwy, Winnellie, NT, 0820*. *P.O.Box 27, Armadale North, Victoria, 3143, AUSTRALIA*. *121 North Seal Way, Cocos Keeling Islands, WA, 6799*. Eduction supports all common delimiters, including newlines.

address_ca.ecr

Entity	Description
address/postcode/ca	Canadian postal codes. For example, *T2P-0B4*, *T2P0B4*, or *T2P 0B4*.
address/region_postcode/ca	Canadian province or territory, and postal code. For example, *Alberta, T2P0B4*.
address/city_region_postcode/ca	Canadian city, province or territory, and postal code. For example, *Calgary, Alberta, T2P 0B4*.
address/ca	Any Canadian address. For example:

address_ca.ecr, continued

Entity	Description
	240 4th Avenue S.W., Suite 600, Calgary, Alberta T2P 4H4, Canada. *124 Av de la Peine, Montreal QC, H3Z 2Y7*. *Suite 600, 222-3rd Ave S.W., Calgary Alberta, T2P 0B4*. Eduction supports all common delimiters, including newlines.

address_cn.ecr

Entity	Description
address/pc/chicn	Chinese postal code. For example, *266033*.
address/chicn	Any Chinese address. For example, 中国，山东省，青岛市 香港东路6号，5号楼，8号室 李小方 (先生)收.
address/engcn	A Chinese address in English. For example. *63 Renmin Lu, Qingdao Shi, 266033 Shandong, China*.
address/cn	A Chinese address in Chinese or English.

address_de.ecr

Entity	Description
address/postcode/de	German postal code. For example, *80639*.
address/postcode_city/de	German postal code, and city. For example, *80639, München*.
address/de	Any German address. For example: *Hewlett-Packard-Straße 1, 61352, Bad Homburg vor der Höhe*. *Postfach 10 01 65, 32547, Bad Oeynhausen, GERMANY*. *Grüner Weg 6, 61169, Friedberg, GERMANY*. Eduction supports all common delimiters, including newlines.

address_eng.ecr

Entity	Description
address/strnum/eng	Street numbers. For example, *12a* or *14-17B*.
address/pobox/eng	Post office box numbers. For example, *PO Box 26*.
address/pmb/eng	Private mail box number. For example, *Private Mail Box 26*.
address/pmb_or_pobox/eng	Post office box or private mail box number.
address/street_pre/eng	Special street type that prefixes street numbers. For example, *Highway Contract, HC*.
address/street_hwy/eng	Highway. For example, *City Route*.
address/street_grid/eng	Grid address. For example, *400W350N*.
address/street/eng	A street. For example, *Cowley Road* or *5th Street NW*.
address/street_corner/eng	A street corner. For example, *Corner King Street & Queen Street*.
address/street_all/eng	Any street For example, *12a Carlisle Lane*.
address/suite/eng	Suite number. For example, *Suite 1*.
address/floor/eng	Floor or level number. For example, *3rd Floor*, *Second Floor*, *Level 8*.
address/floor_or_suite/eng	A floor or suite number.
address/unitshipmil/eng	A military address analogous to a street address.
address/building/eng	A building. For example, *Spear Tower*.

address_es.ecr

Entity	Description
address/postcode/es	Spanish postal code. For example, *19208*.
address/postcode_city/es	Spanish postal code and city. For example, *19208 Guadalajara*.
address/es	Any Spanish address. For example: *Av. de las Cortes de Cádiz, s/n, C. C. El Corte Inglés, 11011, Cádiz*. *Avda. Alfonso XIII, 6, Santander, España*.

address_es.ecr, continued

Entity	Description
	Calle de la Fundición, 3, 33206, Gijón, Spain. Eduction supports all common delimiters, including newlines.

address_fr.ecr

Entity	Description
address/postcode/fr	French postal codes. For example, *75008*.
address/postcode_city/fr	French postal code, city, and optional CEDEX. For example, *75008, Paris*.
address/fr	Any French address. For example: *3, Avenue Denis Semeria, Saint-Jean-Cap-Ferrat, Provence-Alpes-Côte d'Azur, 06230, France*. *950 route des Colles - BP 27, 06901 Valbonne Sophia Antipolis*. *Bât G1 147 r Oberkampf, 75011 PARIS*. Eduction supports all common delimiters, including newlines.

address_fre.ecr

Entity	Description
address/strnum/fre	A street number. For example, *12a* or *14-17B*.
address/pobox/fre	Post office box number in French. For example, *Boite Postale 26*.
address/park/fre	A business park in French. For example, *Technopark de Marseille*.
address/building/fre	A building. For example, *Château de Chambord*.
address/delivery_point/fre	A delivery point in French. For example, *BÂTIMENT 15*.
address/street_type/fre	A street type in French. For example, *Rue*.
address/street/fre	A street in French. For example, *Rue Pierre Charron*.

address_fre.ecr, continued

Entity	Description
address/street_all/fre	Any street in French.
address/house_type/fre	A house type in French. For example, *Residence*.

address_gb.ecr

Entity	Description
address/postcode/gb	United Kingdom postal codes. For example, *GY9 3UX*.
address/city_county_postcode/gb	UK city, optional county/country name, post code, and optional place name. For example, *Cambridge, CB4 0WZ*.
address/gb	Any United Kingdom address. For example: *Cambridge Business Park, Cowley Road, Cambridge, CB4 0WZ*. *12-14 The Diamond, Londonderry, Northern Ireland, BT48 6HW*. *105 Piccadilly, (First Floor), London, W1J 7NJ*. *Unit D, Acorn Business Park, Ling Road, Tower Park, Poole, Dorset, BH12 4NZ*. *44 Dorset Road, Providenciales, TURKS AND CAICOS ISLANDS*. Eduction supports all common delimiters, including newlines.

address_ger.ecr

Entity	Description
address/strnum/ger	A street number. For example, *12a*.
address/pobox/ger	A post office box number in German. For example, *Postfach 26*.
address/street/ger	A street in German. For example, *12 Romanstr*.

address_it.ecr

Entity	Description
address/postcode/it	Italian postal code. For example, *12345* or *IT-98765*.
address/postcode_city/it	Italian postal code and city. For example, *52100 Arezzo*.
address/it	Any Italian address. For example: *Strada del Masarone 67, 13900 Biella (MI)*. *Via Balbi 3 e 40 16126 Genova*. *Via Mascarella n° 21/3, 40131 Bologna, Italia*. Eduction supports all common delimiters, including newlines.

address_ita.ecr

Entity	Description
address/strnum/ita	Italian street number. For example, *12a*.
address/pobox/ita	A post office box number in Italian. For example, *Casella postale 26*.
address/street_type/ita	A street type in Italian. For example, *Via* or *Lungomare*.
address/street/ita	An entire street name in Italian. For example, *Via del Fosso de Dragoncello*.

address_jp.ecr

Entity	Description
address/postcode/jp	Japanese postal code. For example, 青森市.

address_spa.ecr

Entity	Description
address/strnum/spa	A street number. For example, *12a* or *14-17B.*
address/pobox/spa	A post office box number in Spanish. For example, *Apartado de correos 26*.
address/street_type/spa	A street type in Spanish or in another language spoken

address_spa.ecr, continued

Entity	Description
	in Spain. For example, *Calle* or *Passeig*.
address/street_name/spa	A Spanish name that may refer to a street. For example, *26 de Marzo de 1824* or *Trujillo*. These are often used for street names in South America without a street type such as *Calle*.
address/street/spa	An entire street name in Spanish. For example, *Calle de La Habana*.
address/business_area/spa	A shopping centre or business park in Spanish. For example, *Parque Tecnológico de Andalucía*.

address_us.ecr

Entity	Description
address/zipcode/us	U.S. ZIP codes. For example, *94070-1234.*
address/city_state_zipcode/us	U.S. city, state, and ZIP code. For example, *Chicago, IL 80803.*
address/military/us	U.S. military address. For example, *Unit 45013, Box 2666, USAG J, APO AP 96338.*
address/us	Any U.S. address. For example: *30 South Wacker Drive, 22nd Floor, Chicago, IL 60606.* *P.O. Box 29, Sometown, AL 12345.* *5758 West Las Positas Blvd, Suite 100, Pleasanton, CA 94588.* *1 Market Street, Spear Tower, Suite 1900, San Francisco, CA 94105.* Eduction supports all common delimiters, including newlines.

age_eng.ecr

Entity	Description
age/all/eng	An age in English.

age_fre.ecr

Entity	Description
age/all/fre	An age in French.

B

bank.ecr

Entity	Description
bank/engca	Canadian banks. For example, *Canadian Imperial Bank of Commerce*.
bank/engb	UK banks. For example, *HSBC*.
bank/engus	U.S. banks. For example, *Morgan Stanley*.

C

company_chicn.ecr

Entity	Description
company/all/chicn	Any Chinese company.

company_dutnl.ecr

Entity	Description
company/top500/dutnl	Top 500 Dutch companies.
company/designator/dutnl	Dutch company identifiers.

company_engau.ecr

Entity	Description
company/law/engau	Law firms in Australia.

company_engca.ecr

Entity	Description
company/tsx60/engca	A Canadian TSX60 company.
company/TSXVenture50/engca	A Canadian TSX Venture 50 company.
company/all/engca	Any Canadian company. This entity includes all companies matched by the other entities in this section, as well as several hundred other significant companies.

company_enggb.ecr

Entity	Description
company/LSE/enggb	A United Kingdom company listed on the London Stock Exchange.
company/law/enggb	Law firms in the United Kingdom.
company/ftse100/enggb	A FTSE 100 United Kingdom company.
company/all/enggb	Any United Kingdom company. This entity includes all companies matched by the other entities in this section, as well as dozens of other significant companies.

company_engjp.ecr

Entity	Description
company/nikkei225/engjp	A Nikkei225 Japanese company.
company/all/engjp	Any Japanese company. This entity includes all companies matched by the other entities in this section, as well as several hundred other significant companies.

company_engus.ecr

Entity	Description
company/fortune_1000_2008/engus	The 2008 list of Fortune 1000 companies.
company/sp500/engus	U.S. S&P 500 companies.

company_engus.ecr, continued

Entity	Description
company/major_company/engus	Major U.S. companies.
company/law/engus	Law firms in the United States.
company/fortune_500/engus	A company that has featured in the Fortune 500 list at any time since 2011.
company/forbes_largest_private_companies2010/engus	The 2010 list of Forbes largest companies.
company/all/engus	Any U.S. company. This entity includes all companies matched by the other entities in this section, as well as several hundred other significant companies.

company_frefr.ecr

Entity	Description
company/CAC_40/frefr	A French CAC 40 company.
company/CAC_40_stocksymbol/frefr	A French CAC 40 company stock symbol.
company/CAC_next_20/frefr	A French CAC Next 20 company.
company/CAC_next_20_stocksymbol/frefr	A French CAC Next 20 company stock symbol.
company/CAC_mid_60/frefr	A French CAC Mid 60 company.
company/CAC_small/frefr	A French CAC Small company.
company/SBF_120/frefr	A French SBF 120 company.
company/all/frefr	Any French company. This entity includes all companies matched by the other entities in this section.

company_gerde.ecr

Entity	Description
company/dax/gerde	A German DAX company.
company/dax_stocksymbol/gerde	A German DAX company stock symbol.
company/cdax/gerde	A German CDAX company.

company_gerde.ecr, continued

Entity	Description
company/hdax/gerde	A German HDAX company.
company/mdax/gerde	A German MDAX company.
company/sdax/gerde	A German SDAX company.
company/tecdax/gerde	A German TecDAX company.
company/all/gerde	Any German company. This entity includes all companies matched by the other entities in this section.

company_jpnjp.ecr

Entity	Description
company/nikkei225/jpnjp	A Japanese Nikkei 225 company.
company/all/jpnjp	Any Japanese company. This entity includes all companies matched by the other entities in this section, as well as several hundred other significant companies.

company_korkr.ecr

Entity	Description
company/all/korkr	Any Korean company.

company_law_eng.ecr

Entity	Description
company/law_sgl/eng	Law firms with single-word names.
company/law_multi/eng	Law firms with multiple-word names. When names include commas and ampersand characters, the entity includes up to three versions of the name: • full name • with commas removed • with commas and ampersand removed All suffixes are removed for data in these entities.

company_other_eng.ecr

Entity	Description
company/designator/eng	A company designator. For example, *Corp, Inc.*
company/org_legal/eng	Legal practice extensions. For example, *LLC*, *PC*.
company/common_end_word/eng	A common company name end word. For example, *Partners, Bros*.
company/non_name/eng	A non-specific name used in a company name. For example, *American, National*.
company/business/eng	A business term in a company name. For example, *Resorts, Capital, Accountants*.

company_rusru.ecr

Entity	Description
company/all/rusru	Any Russian company.

D

date_chi.ecr

Entity	Description
date/season/chi	The four seasons in Chinese.
date/season_simplified/chi	The four seasons in simplified Chinese.
date/solar_term/chi	The solar terms in Chinese.
date/solar_term_simplified/chi	The solar terms in simplified Chinese.
date/yyyy/chi	The year in Chinese.
date/yyyy_simplified/chi	The year in simplified Chinese and ASCII numbers.
date/mm/chi	The month in Chinese.
date/mm_simplified/chi	The month in simplified Chinese and ASCII numbers.
date/ddd/chi	The day of the week in Chinese.
date/ddd_simplified/chi	The day of the week in simplified Chinese.
date/rel_period/chi	A period relative to the current date in Chinese.

date_chi.ecr, continued

Entity	Description
date/rel_period_simplified/chi	A period relative to the current date in simplified Chinese.
date/period/chi	A fixed period of time in Chinese.
date/period_simplified/chi	A fixed period of time in simplified Chinese.
date/rel_day/chi	A day relative to the current date in Chinese.
date/rel_day_simplified/chi	A day relative to the current date in simplified Chinese.
date/ddd_dd/chi	The day of the week and the day of the month in Chinese.
date/ddd_dd_simplified/chi	The day of the week and the day of the month in simplified Chinese and ASCII numbers.
date/ddd_mmdd/chi	The day of the week and the month and day in Chinese.
date/ddd_mmdd_simplified	The day of the week and the month and day in simplified Chinese and ASCII numbers.
date/mmdd/chi	The month and day in Chinese.
date/mmdd_simplified	The month and day in simplified Chinese and ASCII numbers.
date/mmdd_ddd/chi	The month, day, and day of the week in Chinese.
date/mmdd_ddd_simplified/chi	The month, day, and day of the week in simplified Chinese and ASCII numbers.
date/yyyymmdd/chi	The year, month, and day in Chinese.
date/yyyymmdd_simplified/chi	The year, month, and day in simplified Chinese and ASCII numbers.
date/yyyymmdd_ddd/chi	The year, month, day, and day of the week in Chinese.
date/yyyymmdd_ddd_simplified/chi	The year, month, day, and day of the week in simplified Chinese and ASCII numbers.
date/lunar_mmdd/chi	The month and the day of the lunar calendar in Chinese.
date/lunar_mmdd_simplified/chi	The month and the day of the lunar calendar in simplified Chinese and ASCII numbers.

date_chi.ecr, continued

Entity	Description
date/chi	A date in any format in Chinese.
date/simplified/chi	A date in any format in simplified Chinese and ASCII numbers.
date/day_and_time/chi	A time of day on a specific or relative date in Chinese.
date/day_and_time_simplified/chi	A time of day on a specific or relative date in simplified Chinese and ASCII numbers.

date_eng.ecr

Entity	Description
date/season/eng	The four seasons in English. For example, *Winter*, *Spring*.
date/year/eng	A year in English, in any format.
date/mmm/eng	The month in English, written in full or in short form. For example, *September*, *Sept*.
date/ddd/eng	The day of the week in English. For example, *Monday*, *Tuesday*.
date/rel_period/eng	A period relative to the current date in English.
date/rel_day/eng	A day relative to the current date in English.
date/mmmdd/eng	The month and day in English. For example, *January 5th*, *January 5*, or *January the 5th*.
date/ddmmm/eng	The day and month in English. For example, *5th January*, *5 January*, or *5th of January*.
date/day_date/eng	The date preceded by the day of the week in English. For example, *Sat January 5*, *Saturday the 5th Jan*.
date/month_dd_year/eng	The month, day, and year in English. For example, *January 5th, 2008*.
date/dd_month_year/eng	The day, month, and year in English. For example, *5th January, 2008*.
date/day_date_year/eng	The date and year, preceded by the day of the week, in English. For example, *Saturday, January 5th, 2008*.
date/mmm_year/eng	The month and year in English. For example, *January*

date_eng.ecr, continued

Entity	Description
	2008.
date/eng	A date in any format in English. Supported formats include: Date and month, with optional day and optional year: • 04 Oct 2008 • 4th October 2008 • 4 Oct • 4th of October 2008 • October 4th 2008 • 4th Oct '08 • 04 OCTOBER '08 • Saturday, October the 4th • Sat 4th of Oct • SATURDAY 4 OCTOBER 2008 • SAT OCT 4 • Sat. 4 Oct. 2008 Extra delimiter support for formats where the year is present: • 04_OCT_2008 • 4.10.08 • 04/10/2008 • Saturday 4-10-08 • 04102008 (years 1970-2029 only) • 28-10-2008 • 10/28/08 • OCT 28 2008

date_fre.ecr

Entity	Description
date/season/fre	The seasons in French. For example, *l'Hiver*, *saison des pluies*.
date/ddd/fre	A day of the week, in French. For example, *Lundi*, *Mardi, VEN*.
date/mmm/fre	Month, written in full or in short form, in French. For example, *Septembre*, *Sept*.
date/year/fre	A year in any format.
date/ddmmm/fre	The day and month in French. For example, *5e Janvier*, *5 Janvier*.
date/day_date/fre	The day and month in French, preceded by the day of the week. For example, *Samedi, 5 Janvier*.
date/date_year/fre	The day, month, and year in French. For example, *5 Janvier, 2008*.
date/day_date_year/fre	The day, month, and year in French, preceded by the day of the week. For example, *Samedi, 5 Janvier, 2008*.
date/mmm_year/fre	The month and year in French. For example, *Janvier, 2008*.
date/fre	A date in any format in French. Supported formats include: Date and month, with optional day and optional year: • 04 OCT. 2008 • 4ième Octobre 2008 • 4 Oct • 4 10 '08 • 04 OCTOBRE '08 • Samedi, 4 Oct • SAMEDI 4 OCTOBRE 2008 • Sam. 4 Oct. 2008 Extra delimiter support for formats where the year is present: • 04_OCT_2008

date_fre.ecr, continued

Entity	Description
	• 04/10/2008 • Samedi 4-10-08 • 04102008 (years 1970-2029 only)

date_ger.ecr

Entity	Description
date/ddd/ger	A day of the week in German. For example, *Montag*, *Dienstag*.
date/mmm/ger	A month in German. For example, *März*.
date/year/ger	A year in any format.
date/ddmmyyyy_dotspace/ger	dd. mm. yyyy. For example, *5. 1. 2008*.
date/ddmmm/ger	The day and month in German. For example, *5 Januar*.
date/day_date/ger	The day and month in German, preceded by the day of the week. For example*, Samstag, 5. Januar*.
date/date_year/ger	The day, month, and year in German. For example, *5. Januar, 2008*.
date/day_date_year/ger	The day, month, and year in German, preceded by the day of the week. For example*, Samstag, 5. Januar, 2008*.
date/ger	A date in any numeric format in German. Supported formats include: Date and month, with optional day and optional year: • 04 Okt 2008 • 4 OKTOBER 2008 • 4. okt • 4 Oktober '08 • 04 OCT. '08 • 04. 2. 2007 • Samstag, 03.2.2007 • SONNABEND 4 OKTOBER 2008

date_ger.ecr, continued

Entity	Description
	• SA 04 OKT • Sa. 4. Okt. 2008 Extra delimiter support for formats where the year is present: • 04_OKT_2008 • 04/10/2008 • SA. 04-Okt-2008 • 04102008 (years 1970-2029 only) • 28-10-2008
date/mmm_year/ger	The month and year in German. For example, *Januar 2008*.

date_ita.ecr

Entity	Description
date/season/ita	The seasons in Italian. For example, *la primavera*, *l'inverno*.
date/ddd/ita	A day of the week in Italian. For example, *lunedì*, *MAR*.
date/mmm/ita	A month in Italian. For example, *gen.*, *FEBBRAIO*.
date/year/ita	A year in any format.
date/ddmmm/ita	The day and month in Italian. For example, *5 di gennaio*.
date/day_date/ita	The day and month in Italian, preceded by the day of the week. For example, *sabato 5 di gennaio.*
date/date_year/ita	The day, month, and year in Italian. For example, *5 di gennaio del 2008*.
date/day_date_year/ita	The day, month, and year in Italian, preceded by the day of the week. For example, *sabato 5 di gennaio del 2008*.
date/ita	A date in any format in Italian. Supported formats include:

date_ita.ecr, continued

Entity	Description
	Date and month, with optional day and optional year: • 04 Ott 2008 • 4 OTTOBRE 2008 • 4 ott • 04 di Ottobre 2008 • 4 di Ott del '08 • 4 Ott. '08 • Venerdi', 03 di Ottobre • Sab 4 di Ott • VENERDÌ 03 DI OTTOBRE DEL 2008 • SAB 4 OTT • Sab. 4 Ott. 2008 Extra delimiter support for formats where the year is present: • 04_OTT_2008 • 04/10/2008 • Venerdì 3-10-08 • 04102008 (years 1970-2029 only) • 28-10-2008
date/mmm_year/ita	Month and year in Italian. For example, *gennaio del 2008*.

date_jpn.ecr

Entity	Description
date/season/jpn	The seasons in Japanese.
date/ddd/jpn	A day of the week in Japanese.
date/mmm/jpn	A month in Japanese (Kanji, numerals and fullwidth numerals).
date/year_gregorian/jpn	A year in the Gregorian calendar, in Japanese, in any format, with optional A.D./B.C.

date_jpn.ecr, continued

Entity	Description
date/year_imperial/jpn	Japanese imperial calendar year from 1868 onwards, in any format.
date/mmmdd/jpn	The month and day in Japanese.
date/day_date/jpn	The day and month in Japanese, preceded by the day of the week.
date/date_year_gregorian/jpn	The year, month, and day in the Gregorian calendar, in Japanese.
date/date_year_imperial/jpn	The year, month, and day in the Japanese imperial calendar, in Japanese.
date/day_date_year_gregorian/jpn	The day, month, and year in the Gregorian calendar, in Japanese, preceded by the day of the week.
date/day_date_year_imperial/jpn	The day, month, and year in the Japanese imperial calendar, in Japanese, preceded by the day of the week.
date/jpn	A date in any numeric format in Japanese.
date/mmm_year_gregorian/jpn	The month and year in the Gregorian calendar, in Japanese.
date/mmm_year_imperial/jpn	The month and year in the Japanese imperial calendar, in Japanese.

date_numeric.ecr

Entity	Description
date/dd	A day from 1 to 31.
date/dd_fullwidth	A day from 1 to 31, in fullwidth characters.
date/dd2	A day from 01 to 31.
date/dd2_fullwidth	A day from 01 to 31, in fullwidth characters.
date/mm	A month from 1 to 12.
date/mm_fullwidth	A month from 1 to 12, in fullwidth characters.
date/mm2	A month from 01 to 12.
date/mm2_fullwidth	A month from 01 to 12, in fullwidth characters.

date_numeric.ecr, continued

Entity	Description
date/yy	The last two digits of the year. For example, *67*, *08*.
date/yy_fullwidth	The last two digits of the year, in fullwidth characters.
date/yyyy	A three- or four-digit year, from 100 to 2099.
date/yyyy_fullwidth	A three- or four-digit year in fullwidth characters, from 100 to 2099.
date/yyyy4	A four-digit year, from 1000 to 2099.
date/yyyy4_fullwidth	A four-digit year in fullwidth characters, from 1000 to 2099.
date/year	A year in any numerical format.
date/year_fullwidth	A year in any numerical format in fullwidth characters.
date/yyyymmddsep	yyyy-mm-dd. For example, *2008-10-28*.
date/yyyymmdd	yyyymmdd. For example, *20081028*.
date/yyyymmdd_safe	yyyymmdd for a date between 19700101 and 20291231. For example, *20081028*.
date/yymmddsep	yy-mm-dd. For example, *08-10-28*.
date/yymmdd	yymmdd. For example, *081028*.
date/ddmmyyyysep	dd-mm-yyyy. For example, *28-10-2008*.
date/ddmmyyyy	ddmmyyyy. For example, *28102008*.
date/ddmmyyyy_safe	ddmmyyyy for a date between 01011970 and 31122029. For example, *28102008*.
date/ddmmyysep	dd-mm-yy. For example, *28-10-08*.
date/ddmmyy	ddmmyy. For example, *281008*.
date/mmddyyyysep	mm-dd-yyyy. For example, *10-28-2008*.
date/mmddyyyy	mmddyyyy. For example, *10282008*.
date/mmddyyyy_safe	mmddyyyy for a date between 01011970 and 12312029. For example, *10282008*.
date/mmddyysep	mm-dd-yy. For example, *10-28-2008*.
date/mmddyy	mmddyy. For example, *102808*.

date_por.ecr

Entity	Description
date/season/por	The seasons in Portuguese. For example, *Verão, Outono.*
date/ddd/por	A day of the week in Portuguese. For example, *Segunda-feira*, *Terça-feira, DOM*.
date/mmm/por	A month in Portuguese. For example, *Setembro*.
date/year/por	A year in any format.
date/ddmmm/por	The day and month in Portuguese. For example, *5 de Janeiro*.
date/day_date/por	The day and month in Portuguese, preceded by the day of the week. For example, *Sábado 5 de Janeiro*.
date/date_year/por	The day, month, and year in Portuguese. For example, *5 de maio 2008*.
date/day_date_year/por	The day, month, and year in Portuguese, preceded by the day of the week. For example, *Sábado 5 de janiero de 2008*.
date/por	Any date in Portuguese. Supported formats include: Date and month, with optional day and optional year: • 04 Out. 2008 • 4 OUTUBRO 2008 • 04 de Outubro 2008 • 4 de Out de '08 • SÁB 04 OUT 2008 • Sábado, 04 de Outubro • Terça-feira 14 Out. 1947 • SÁBADO 04 DE OUTUBRO DE 2008 • Quinta-feira, 12 de Setembro de 2013 EC • 4 de Março de 2012 Extra delimiter support for formats where the year is present: • 04_OUT_2008 • 04/10/2008

date_por.ecr, continued

Entity	Description
	• Quarta feira 30-12-1953 • 04102008 (years 1970-2029 only) • 28-10-2008
date/mmm_year/por	The month and year in Portuguese. For example, *Junho de 2008.*

date_spa.ecr

Entity	Description
date/season/spa	The seasons in Spanish. For example, *el invierno, la primavera.*
date/ddd/spa	A day of the week in Spanish. For example, *Lunes, Domingo*.
date/mmm/spa	A month in Spanish. For example, *Septiembre*.
date/year/spa	A year in any format.
date/ddmmm/spa	The day and month in Spanish. For example, *5 de enero*.
date/date_year/spa	The day, month, and year in Spanish. For example, *5 de enero 2008*.
date/day_date/spa	The day and month in Spanish, preceded by the day of the week. For example, *Sábado 5 de enero*.
date/day_date_year/spa	The day, month, and year in Spanish, preceded by the day of the week. For example, *Sábado 5 de enero de 2008*.
date/mmm_year/spa	The month and year in Spanish. For example, *Januar 2008*.
date/spa	Any date in Spanish. Supported formats include: Date and month, with optional day and optional year: • 04 Oct 2008 • 4 OCTUBRE 2008 • 4 OCT • 4 de Octubre 2008

date_spa.ecr, continued

Entity	Description
	• 4 de Oct de '08 • 04 OCT. '08 • Sábado, 04 de Octubre • Jueves, 12 de Septiembre de 2013 d. J.C. • SÁBADO 04 DE OCTUBRE DE 2008 • SAB 4 OCT • Sab. 4 Oct. 2008 Extra delimiter support for formats where the year is present: • 04_OCT_2008 • 04/10/2008 • Sábado 4-10-08 • 04102008 (years 1970-2029 only) • 28-10-2008

datetime_advanced_eng.ecr

Entity	Description
datetime/advanced_hms24/eng	Time in hh:mm:ss.ss ZZZ format (seconds, fractional seconds, and timezone are optional). For example, *04:35*, *18:56:00*, *21:42:56.45 +0100*.
datetime/advanced_hms24_range/eng	Time range in hh:mm:ss.ss ZZZ format (seconds, fractional seconds, and timezone are optional). For example, *04:35-04:36*, *18:56:00-21:00:00*, *21:42:56.45 to 23:59:59.99 +0100*.
datetime/advanced_hm24_dot/eng	Time in hh.mm format. For example, *04.56*.
datetime/advanced_hm24_dot_range/eng	Time range in hh.mm format. For example, *04.56 to 12.34*.
datetime/advanced_hm24tz_nosep/eng	Time in hhmm ZZZ format. For example, *2100 GMT*.
datetime/advanced_hm24tz_nosep_ range/eng	Time range in hhmm ZZZ format. For example, *2100-2330 GMT*.
datetime/advanced_hm24_nosep/eng	Time in hhmm format, with higher scores if the number

datetime_advanced_eng.ecr, continued

Entity	Description
	of minutes is a multiple of 5. For example, *2100*.
datetime/advanced_hm24_nosep_range/eng	Time range in hhmm format, with higher scores if the number of minutes is a multiple of 5. For example, *2100-2330*.
datetime/advanced_hms12/eng	Time in 12-hour h:mm:ss am/pm ZZZ format (seconds and timezone are optional). For example, *9:30am*, *9:30:00pm GMT*.
datetime/advanced_hms12_range/eng	Time range in 12-hour h:mm:ss am/pm ZZZ format (seconds and timezone are optional). For example, *9:30-10:30am*, *9:30:00am to 9:30:00pm GMT*.
datetime/advanced_hm12_noampm/eng	Time in 12-hour h:mm:ss ZZZ format without am or pm specified (seconds and timezone are optional). For example, *9:30*, *9:30:00 GMT*.
datetime/advanced_hm12_noampm_range/eng	Time range in 12-hour h:mm:ss ZZZ format without am or pm specified (seconds and timezone are optional). For example, *9:30-10:30*, *9:30:00 to 9:30:00 GMT*.
datetime/advanced_hm12_dot/eng	Time in 12-hour h.mm am/pm ZZZ format (am, pm, and timezone are optional, but scores are lower without them, although multiples of 5 minutes are boosted). For example, *6.30am*, *8.45 GMT*, *11.35*.
datetime/advanced_hm12_dot_range/eng	Time range in 12-hour h.mm am/pm ZZZ format (am, pm, and timezone are optional, but scores are lower without them, although multiples of 5 minutes are boosted). For example, *6-7.30am*, *8.45am-6.30pm GMT*, *11-12.35*.
datetime/advanced_hm12_nosep/eng	Time in 12-hour hmm am/pm ZZZ format (am, pm, and timezone are optional, but scores are lower without them, although multiples of 5 minutes are boosted). For example, *630am*, *845 GMT*, *1135*.
datetime/advanced_hm12_nosep_range/eng	Time range in 12-hour hmm am/pm ZZZ format (am, pm, and timezone are optional, but scores are lower without them, although multiples of 5 minutes are boosted). For example, *630-730am*, *845-945 GMT*, *1135 to 345*.
datetime/advanced_namedtime/eng	Times of the day with a specific name in the English language. For example, *noon*, *midnight*.

datetime_advanced_eng.ecr, continued

Entity	Description
datetime/advanced_clocktime_loose/eng	Time of the day, or time range, described in English (low confidence, scores reduced). For example, *twelve*.
datetime/advanced_clocktime_loose_range/eng	Time range, described in English (low confidence, scores reduced). For example, *two to three*.
datetime/advanced_clocktime_strict/eng	Time of the day, described in English (high confidence). For example, *twelve o'clock*, *two fifteen*, *ten past one*, *quarter to midnight*.
datetime/advanced_clocktime_strict_range/eng	Time range, described in English (high confidence). For example, *ten to ten forty-five*.
datetime/advanced_clocktime/eng	Time of the day, described in English (high and low confidence, scored appropriately). For example, *twelve o'clock*, *two fifteen in the afternoon*, *ten past one*, *quarter to midnight*, *twelve at night*.
datetime/advanced_clocktime_range/eng	Time range, described in English (high and low confidence, scored appropriately). For example, *ten to ten forty-five*, *two to three*.
datetime/nameddays_strict/eng	Specially named days (confident matched only). For example, *Christmas Day*, *Easter Monday*.
datetime/nameddays_all/eng	Specially named days (high and low confidence matches, scored appropriately). For example, *Christmas*, *Easter*.
datetime/advanced_yyyymmdd/eng	Dates in yyyy-mm-dd, yyyy-Mmm-dd, yyyy.mm.dd, yyyy.Mmm.dd, yyyy/mm/dd, or yyyy/Mmm/dd formats. For example, *2008-10-28*, *2008.Oct.28*.
datetime/advanced_yyyymmdd_range/eng	Date range in yyyy-mm-dd, yyyy-Mmm-dd, yyyy.mm.dd, yyyy.Mmm.dd, yyyy/mm/dd, or yyyy/Mmm/dd formats. For example, *2008-10-28 to 2008-11-03*, *2008.Oct.28-2008.Nov.03*.
datetime/advanced_yyyymmdd_nosep/eng	Dates in yyyy-mm-dd, yyyy.mm.dd, or yyyy/mm/dd formats. For example, *20081028*.
datetime/advanced_yyyymmdd_nosep_range/eng	Date range in yyyy-mm-dd, yyyy.mm.dd, or yyyy/mm/dd formats. For example, *20081028-20081103*.
datetime/advanced_ddmmyyyy/eng	Dates in dd-m-yyyy, dd.m.yyyy, or dd/m/yyyy formats. For example, *28-10-2008*.

datetime_advanced_eng.ecr, continued

Entity	Description
datetime/advanced_ddmmyyyy_range/eng	Date range in dd-m-yyyy, dd.m.yyyy, or dd/m/yyyy formats. For example, *28-10-2008 to 03-11-2008*.
datetime/advanced_ddmmyy/eng	Dates in dd-mm-yy, dd.mm.yy, or dd/mm/yy formats. For example, *28.10.08*.
datetime/advanced_ddmmyy_range/eng	Date range in dd-mm-yy, dd.mm.yy, or dd/mm/yy formats. For example, *28.10.08 -03.11.08*.
datetime/advanced_ddmm/eng	Dates in dd-mm, dd.mm, or dd/mm formats. For example, *28/10*.
datetime/advanced_ddmm_range/eng	Date range in dd-mm, dd.mm, or dd/mm formats. For example, *28/10-03/11*.
datetime/advanced_ddMmmyyyy/eng	Dates in dd-Mmm-yyyy, dd.Mmm.yyyy, or dd/Mmm/yyyy formats. For example, *28-Oct-2008*.
datetime/advanced_ddMmmyyyy_ range/eng	Date range in dd-Mmm-yyyy, dd.Mmm.yyyy, or dd/Mmm/yyyy formats. For example, *28-Oct-2008 to 03-Nov-2008*.
datetime/advanced_ddMmmyy/eng	Dates in dd-Mmm-yy, dd.Mmm.yy, or dd/Mmm/yy formats. For example, *28.Oct.08*.
datetime/advanced_ddMmmyy_range/eng	Date range in dd-Mmm-yy, dd.Mmm.yy, or dd/Mmm/yy formats. For example, *28.Oct.08-03.Nov.08*.
datetime/advanced_ddMmm/eng	Dates in dd-Mmm, dd.Mmm, or dd/Mmm formats. For example, *28/Oct*.
datetime/advanced_ddMmm/eng	Date range in dd-Mmm, dd.Mmm, or dd/Mmm formats. For example, *28/Oct-03/Nov*.
datetime/advanced_mmddyyyy/eng	Dates in m-dd-yyyy, m.dd.yyyy, or m/dd/yyyy formats. For example, *10-28-2008*.
datetime/advanced_mmddyyyy_range/eng	Date range in m-dd-yyyy, m.dd.yyyy, or m/dd/yyyy formats. For example, *10-28-2008 to 11-03-2008*.
datetime/advanced_mmddyy/eng	Dates in mm-dd-yy, mm.dd.yy, or mm/dd/yy formats. For example, *10.28.08*.
datetime/advanced_mmddyy_range/eng	Date range in mm-dd-yy, mm.dd.yy, or mm/dd/yy formats. For example, *10.28.08 to 11.03.08*.
datetime/advanced_mmdd/eng	Dates in mm-dd, mm.dd, mm/dd formats. For example, *10/28*.

datetime_advanced_eng.ecr, continued

Entity	Description
datetime/advanced_mmdd_range/eng	Date range in mm-dd, mm.dd, mm/dd formats. For example, *10/28-11/03*.
datetime/advanced_Mmmddyyyy/eng	Dates in Mmm-dd-yyyy, Mmm.dd.yyyy, or Mmm/dd/yyyy formats. For example, *Oct-28-2008*.
datetime/advanced_Mmmddyyyy_ range/eng	Date range in Mmm-dd-yyyy, Mmm.dd.yyyy, or Mmm/dd/yyyy formats. For example, *Oct-28-2008 to Nov-03-2008*.
datetime/advanced_Mmmddyy/eng	Dates in Mmm-dd-yy, Mmm.dd.yy, or Mmm/dd/yy formats. For example, *Oct.28.08*.
datetime/advanced_Mmmddyy_range/eng	Date range in Mmm-dd-yy, Mmm.dd.yy, or Mmm/dd/yy formats. For example, *Oct.28.08 to Nov.03.08*.
datetime/advanced_Mmmdd/eng	Dates in Mmm-dd, Mmm.dd, Mmm/dd formats. For example, *Oct/28*.
datetime/advanced_Mmmdd_range/eng	Date range in Mmm-dd, Mmm.dd, Mmm/dd formats. For example, *Oct/28-Nov/03*.
datetime/advanced_Mmmyyyy/eng	Named month/year. For example, *Oct 2008*, *October of 2008*, *October '08*.
datetime/advanced_Mmmyyyy_range/eng	Named month/year range. For example, *Oct 2008 to Feb 2009*.
datetime/advanced_yyyymm/eng	Dates in yyyy-mm, yyyy.mm, yyyy/mm and yyyy mm formats.
datetime/advanced_yyyymm_range/eng	Date range in yyyy-mm, yyyy.mm, yyyy/mm and yyyy mm formats.
datetime/advanced_textdate_noyear/eng	A date, without a year, described in English. For example, *July 4th*, *July the 4th*, *The morning of Wednesday July fourth*, *4th July,the 4th of July*, *In the morning on Wednesday fourth July*, *Christmas Eve*, *Easter Day*.
atetime/advanced_textdate_noyear_ range/eng	A date range, without a year, described in English. For example, *July 4th-8th*, *July 4th-October 8th*, *July the 4th to the 8th*, *Wednesday 4th to Sunday 8th July*, *4th July-8th October*.
datetime/advanced_textdate_noyear_ multiple/eng	Multiple dates, without a year, described in English. For example, *July 4th and 8th*, *4th, 5th and 6th of July*,

datetime_advanced_eng.ecr, continued

Entity	Description
	Wednesday 4th and Sunday 8th July.
datetime/advanced_textdate_withyear/eng	A date, with a year, described in English. For example, *July 4th 2008*, *July the 4th 2008*, *The morning of Wednesday July fourth 2008*, *4th July 2008*, *the 4th of July 2008*, *In the morning on Wednesday fourth July 2008*, *Christmas Day 2012*, *Easter '02*.
datetime/advanced_textdate_withyear_range/eng	A date range, with a year, described in English. For example, *July 4th-8th 2008*, *July 4th-October 8th 2008*, *July the 4th to the 8th, 2008*, *Wednesday 4th to Sunday 8th July 2008*, *4th July-8th October 2008*.
datetime/advanced_textdate_withyear_multiple/eng	Multiple dates, with a year, described in English. For example, *July 4th and 8th, 2008*, *4th, 5th and 6th of July 2008*, *Wednesday 4th and Sunday 8th July 2008*.
datetime/advanced_textdate/eng	A date, described in English (with or without a year).
datetime/advanced_textdate_range/eng	A date range, described in English (with or without a year).
datetime/advanced_textdate_multiple/eng	Multiple dates, described in English (with or without a year).
datetime/advanced_reldate/eng	A day, or part of a day, relative to today, described in English. Less confident matches are scored lower. For example, *This morning*, *tomorrow evening*, *Today* (score 0.9), *yesterday* (score 0.9), *the day after tommorrow* [sic] (score 0.9), *Two weeks ago on Monday*, *This coming Tuesday AM*, *Two weeks on Wednesday afternoon*, *Tuesday week*, *Tomorrow fortnight*, *Two weeks ago last Monday*, *Monday last week*, *Tuesday this wk*, *Wednesday next*, *Not this Tuesday but next* (score 0.9), *Not last Monday but the one before* (score 0.9), *4th* (score 0.5), *the fourth* (score 0.8), *Wednesday the fourth of next month*, *last month on Wednesday 4th*, *The first Sunday of next month*, *Second Monday in July*, *Last Tuesday of April but one*, *Monday morning* (score 0.9), *Tuesday* (score 0.7).
datetime/advanced_reldate_range/eng	A date range, relative to today, described in English. Less confident matches are scored lower. For example, *Wednesday 4th to Friday 6th* (score 1.0), *4th-6th next month* (score 0.9), *4th to 6th* (score 0.8).

datetime_advanced_eng.ecr, continued

Entity	Description
datetime/advanced_reldate_multiple/eng	Multiple dates, relative to today, described in English. For example, *4th and 6th of next month*.
datetime/advanced_relmonth/eng	A month, relative to today, described in English. Less confident matches are scored lower. For example, *In January* (score 0.6), *This July* (score 0.6), *September* (score 0.4) *Last September* (score 0.8), *January next year* (score 0.9), *Next month* (score 0.6), *last month* (score 0.6).
datetime/advanced_date_and_time/eng	Any date with time, in any recognized format. Relevant components are extracted. Score indicates the confidence that the matched text is a genuine reference to a date and time.
datetime/advanced_date_only/eng	Any date (without a time), in any recognized format. Relevant components are extracted. Score indicates the confidence that the matched text is a genuine reference to a date.
datetime/advanced_month_only/eng	Any date to month precision, in any recognised format. Relevant components are extracted. Score indicates the confidence that the matched text is a genuine reference to a date.
datetime/advanced_time_only/eng	Any time (without a date), in any recognized format. Relevant components are extracted. Score indicates the confidence that the matched text is a genuine reference to a time.
datetime/advanced/eng	Any date, with optional time, in any recognized format. Relevant components are extracted. Score indicates the confidence that the matched text is a genuine reference to a date and time. You can use the `datetime.lua` script to standardize the output of these entities. To allow precision to the month of the year rather than just day, set `RelaxedPrecision` to **`True`** in your eduction request.

E

ethnicity_eng.ecr

Entity	Description
ethnicity/nationality/eng	A nationality. For example, *Andorran*, *Welsh*.

ethnicity_engca.ecr

Entity	Description
ethnicity/aboriginal/engca	A Canadian aboriginal group. For example, *Inuit*.
ethnicity/population_group/engca	A Canadian population group. For example, *Arab*, *White*.

ethnicity_enggb.ecr

Entity	Description
ethnicity/enggb	Ethnicity classification in England. For example, *Irish*, *Indian*.
ethnicity/identity_code/enggb	United Kingdom identity code. For example, *IC1*, *IC2*.

ethnicity_engus.ecr

Entity	Description
ethnicity/races/engus	A United States race. For example, *Japanese*, *White*.
ethnicity/races_lowercase/engus	A U.S. race in lowercase. For example, *japanese*, *white*.
ethnicity/native_american/engus	A U.S. native. For example, *Cherokee*, *Lambee*.
ethnicity/asian/engus	A U.S. ethnicity of Asian origin. For example, *Pakistani*, *Korean*.
ethnicity/pacific/engus	A U.S. ethnicity of Pacific origin. For example, *Fijian*, *Tongan*.
ethnicity/hispanic/engus	A U.S. ethnicity of Hispanic origin. For example, *Cuban*, *Spanish*.
ethnicity/engus	Any U.S. ethnicity.

ethnicity_fre.ecr

Entity	Description
ethnicity/nationality/fre	Nationality in French. For example, *Andorrane*, *Vietnamien*.
ethnicity/ethnic_groups/fre	Ethnic groups in the French language. For example, *Africain*, *Autres*.

G

gender_eng.ecr

Entity	Description
gender/gender_word/eng	A word that describes a family relation or gender in English. For example, *lady, father*.
gender/gender_context/eng	A gender in the context of English language.
gender/all/eng	A gender in the English language, either in a word or in context.

gender_fre.ecr

Entity	Description
gender/gender_word/fre	A word that describes a family relation or gender in French. For example, *Dame*, *voisines*.
gender/gender_context/fre	A gender in the context of French language.
gender/all/fre	A gender in the French language, either in a word or in context.

gender_ger.ecr

Entity	Description
gender/gender_word/ger	A word that describes a family relation or gender in German. For example, *mensch*, *Frau*.
gender/gender_context/ger	A gender in the context of German language.
gender/all/ger	A gender in the German language, either in a word or in context.

gov_chicn.ecr

Entity	Description
org/gov/chicn	A Chinese government agency.

gov_engca.ecr

Entity	Description
org/gov/engca	A Canadian government agency.

H

holiday_ca.ecr

Entity	Description
holiday/statutory/engca	Statutory Canadian holidays in English. For example, *Good Friday.*
holiday/statutory/freca	Statutory Canadian holidays in French. For example, *Le vendredi saint.*
holiday/statutory/ca	Statutory Canadian holidays, in English or French.
holiday/federal/engca	Federal Canadian holidays in English. For example, *Victoria Day*.
holiday/federal/freca	Federal Canadian holidays in French. For example, *La fête de la Reine.*
holiday/federal/ca	Federal Canadian holidays, in English or French.
holiday/statother/engca	Other statutory Canadian holidays in English. For example, *Family Day.*
holiday/statother/freca	Other statutory Canadian holidays in French. For example, *La fête du Travail.*
holiday/statother/ca	Other statutory Canadian holidays, in English or French.
holiday/alberta/engca holiday/britishcolumbia/engca holiday/manitoba/engca holiday/newbrunswick/engca	Holidays for each Canadian province and territory in English.

holiday_ca.ecr, continued

Entity	Description
holiday/newfoundlandlabrador/engca holiday/northwestterritories/engca holiday/novascotia/engca holiday/nunavut/engca holiday/ontario/engca holiday/princeedwardisland/engca holiday/quebec/engca holiday/saskatchewan/engca holiday/yukon/engca	
holiday/prov_terr/engca	Holidays for Canadian provinces and territories in English.
holiday/other/engca	Other Canadian holidays and observances in English.
holiday/ca	All Canadian holidays.

holiday_enggb.ecr

Entity	Description
holiday/bank_holiday/enggb	British Bank Holiday name.
holiday/holiday/enggb	Traditional days celebrated. For example, *Mother's Day*.

holiday_engus.ecr

Entity	Description
holiday/federal/engus	U.S. federal holidays. For example, *Memorial Day*.
holiday/traditional/engus	Traditional U.S. days celebrated. For example, *Mother's Day*.
holiday/engus	All U.S. holidays.

I

internet.ecr

Entity	Description
internet/host_domain	A host name. For example, *www.myhost.com*.
internet/host_ip/ipv4	An IPv4 IP address. For example, *127.0.0.1*.
internet/host_ip/ipv6	An IPv6 IP address. For example, *1234:5678:90AB:CDEF.*
internet/host_ip/ipv4mapped	An IPv4-mapped IP address. For example, *::FFFF:129.144.52.38*.
internet/host_ip	Any IP address.
internet/addr_host	Host address. For example, *www.myhost.com* or *192.231.21.2*.
internet/addr_email	Email address. For example, *jsmith@mailserver.com*.
internet/addr_email_mailto	Email address with *mailto:* prefix. For example, *mailto:jsmith@mailserver.com*.
internet/addr_https	HTTP or HTTPS address.
internet/addr_file	file:// address.
internet/addr_ftp	FTP address.
internet/addr_news	news:// address.
internet/addr_telnet	Telnet address.
internet/addr_gopher	Gopher address.

J

jobtitledicts_eng.ecr

Entity	Description
person/titleprefix_camelcase/eng	Job title prefix in camel case. For example, *Acting*.
person/titleprefix_lowercase/eng	Job title prefix in lowercase. For example, *acting*.
person/titlesuffix_camelcase/eng	Job title suffix in camel case. For example, *Associate*, *Advisor*.

jobtitledicts_eng.ecr, continued

Entity	Description
person/titlesuffix_lowercase/eng	Job title suffix in lowercase. For example, *educator*, *trainee*.
person/govdep/engus	U.S. government departments and abbreviations. For example, *National Security Council*, *FBI.*
person/titlegeneric_camelcase/eng	Generic job titles in camel case. For example, *Sales Assistant*.
person/titlegeneric_lowercase/eng	Generic job titles in lowercase. For example, *sales assistant*.
person/titlefull_camelcase/eng	Full job title in camel case, including prefixes and suffixes. For example, *Head of Customer Communications*.
person/titlefull_lowercase/eng	Full job title in lower case, including prefixes and suffixes. For example, *head of customer communications*.
person/titlecorp/eng	Corporate job titles. For example, *Chief Financial Officer.*
person/titlecorpabb/eng	Abbreviated version of corporate job titles. For example, *CFO.*
person/titlegov/eng	Government and cabinet titles. For example, *President*, *Secretary of Defense.*
person/titleroyal/eng	Royal titles. For example, *King.*
person/titlepolitical/eng	Political titles. For example, *Foreign Minister*, *Governor.*
person/titlereligious/eng	Religious titles. For example, *Pope*, *Father*, *Imam.*

L

languages.ecr

Entity	Description
language/iso_lowercase	Three-letter ISO 639-2/B language code. For example, *fin*, *ger*.
language/all	Language name in a local language, English, or other

languages.ecr, continued

Entity	Description
	major language.
language/output_iso	Language name in a local language, English, or other major language (output is normalized to the ISO 639-2/B code)

legal_engus.ecr

Entity	Description
legal/citsupr/engus	Supreme Court Citations. For example, *Roe v. Wade, 410 U.S. 113 (1973)*.
legal/citcofa/engus	Federal Court Reporter Citations. For example, *Universal City Studios, Inc. v. Corley, 273 F.3d 429 (2d Cir. 2001)*.

M

measure_eng.ecr

Entity	Description
measure/len/met/eng	Metric measures of length. For example, *mm.*, *kilometre.*
measure/len/usuk/eng	U.S. and UK measures of length. For example, *foot*, *mile*, *in.*
measure/area/met/eng	Metric measures of area. For example, *sq. m.*, *square kilometres.*
measure/area/usuk/eng	U.S. and UK measures of area. For example, *sq. in.*, *acres.*
measure/vol/met/eng	Metric measures of volume. For example, *microlitres*, *cubic centimetres.*
measure/vol/usuk/eng	U.S. and UK measures of volume. For example, *pinches*, *cups*, *gal.*
measure/mass/met/eng	Metric measures of mass. For example, *gram*, *tonnes.*
measure/mass/usuk/eng	U.S. and UK measures of mass. For example, *pound*, *lb.*

medical_condition.ecr

Entity	Description
medical/disability/social_security/engus	Impairment for the purpose of disability evaluation under social security.
medical/disease_condition	Disease or medical condition.
medical/lifestyle	Lifestyle that relates to medical condition.

medical_drug.ecr

Entity	Description
drug/brand	Trade name of medical drugs.
drug/generic	Generic name of medical drugs.
drug/medication	Description of a medication.

medical_healthcare_engus.ecr

Entity	Description
healthcare/provider/AK/engus	U.S. healthcare provider in Alaska.
healthcare/provider/AL/engus	U.S. healthcare provider in Alabama.
healthcare/provider/AR/engus	U.S. healthcare provider in Arkansas.
healthcare/provider/AZ/engus	U.S. healthcare provider in Arizona.
healthcare/provider/CA/engus	U.S. healthcare provider in California.
healthcare/provider/CO/engus	U.S. healthcare provider in Colorado.
healthcare/provider/CT/engus	U.S. healthcare provider in Connecticut.
healthcare/provider/DC/engus	U.S. healthcare provider in Washington, D.C.
healthcare/provider/DE/engus	U.S. healthcare provider in Delaware.
healthcare/provider/FL/engus	U.S. healthcare provider in Florida.
healthcare/provider/GA/engus	U.S. healthcare provider in Georgia.
healthcare/provider/HI/engus	U.S. healthcare provider in Hawaii.
healthcare/provider/IA/engus	U.S. healthcare provider in Iowa.

medical_healthcare_engus.ecr, continued

Entity	Description
healthcare/provider/ID/engus	U.S. healthcare provider in Idaho.
healthcare/provider/IL/engus	U.S. healthcare provider in Illinois.
healthcare/provider/IN/engus	U.S. healthcare provider in Indiana.
healthcare/provider/KS/engus	U.S. healthcare provider in Kansas.
healthcare/provider/KY/engus	U.S. healthcare provider in Kentucky.
healthcare/provider/LA/engus	U.S. healthcare provider in Louisiana.
healthcare/provider/MA/engus	U.S. healthcare provider in Massachusetts.
healthcare/provider/MD/engus	U.S. healthcare provider in Maryland.
healthcare/provider/ME/engus	U.S. healthcare provider in Maine.
healthcare/provider/MI/engus	U.S. healthcare provider in Michigan.
healthcare/provider/MN/engus	U.S. healthcare provider in Minnesota.
healthcare/provider/MO/engus	U.S. healthcare provider in Missouri.
healthcare/provider/MS/engus	U.S. healthcare provider in Mississippi.
healthcare/provider/MT/engus	U.S. healthcare provider in Montana.
healthcare/provider/NC/engus	U.S. healthcare provider in North Carolina.
healthcare/provider/ND/engus	U.S. healthcare provider in North Dakota.
healthcare/provider/NE/engus	U.S. healthcare provider in Nebraska.
healthcare/provider/NH/engus	U.S. healthcare provider in New Hampshire.
healthcare/provider/NJ/engus	U.S. healthcare provider in New Jersey.
healthcare/provider/NM/engus	U.S. healthcare provider in New Mexico.
healthcare/provider/NV/engus	U.S. healthcare provider in Nevada.
healthcare/provider/NY/engus	U.S. healthcare provider in New York.
healthcare/provider/OH/engus	U.S. healthcare provider in Ohio.
healthcare/provider/OK/engus	U.S. healthcare provider in Oklahoma.
healthcare/provider/OR/engus	U.S. healthcare provider in Oregon.
healthcare/provider/PA/engus	U.S. healthcare provider in Pennsylvania.

medical_healthcare_engus.ecr, continued

Entity	Description
healthcare/provider/PR/engus	U.S. healthcare provider in Puerto Rico.
healthcare/provider/RI/engus	U.S. healthcare provider in Rhode Island.
healthcare/provider/SC/engus	U.S. healthcare provider in South Carolina.
healthcare/provider/SD/engus	U.S. healthcare provider in South Dakota.
healthcare/provider/TN/engus	U.S. healthcare provider in Tennessee.
healthcare/provider/TX/engus	U.S. healthcare provider in Texas.
healthcare/provider/UT/engus	U.S. healthcare provider in Utah.
healthcare/provider/VA/engus	U.S. healthcare provider in Virginia.
healthcare/provider/VT/engus	U.S. healthcare provider in Vermont.
healthcare/provider/WA/engus	U.S. healthcare provider in Washington.
healthcare/provider/WI/engus	U.S. healthcare provider in Wisconsin.
healthcare/provider/WV/engus	U.S. healthcare provider in West Virginia.
healthcare/provider/WY/engus	U.S. healthcare provider in Wyoming.
healthcare/provider/all/engus	Any U.S. healthcare provider.

medical_procedure.ecr

Entity	Description
medical/blood_test	Blood test.
medical/lab_test	Lab test.
medical/surgical_procedure	Surgical procedure.
medical/specialty	Medical specialty.

monetary_value.ecr

Entity	Description
monetary_value/full_value	A monetary value without context. For example, *$5 billion*, *£9.8m*, *EUR50bn*, *£15,275,486*, *25 cents*. This entity has the following components:

monetary_value.ecr, continued

Entity	Description
	• PRECURRENCY, for example *$* from *$100*. • POSTCURRENCY, for example *cents* from *25 cents*. • VALUE, for example *50.0* from *50.0 million* or *1.26* from *£1.26k*. • BIGUNITS, for example *million* from *$50.0 million*. You can use the Lua post processing script `normalize_money.lua` to normalize the values. The script adds a component named NORMALIZED_VALUE to each match. For example, the input *£9.8m* would have a VALUE of *9.8* but a NORMALIZED_VALUE of *9,800,000*.

money_eng.ecr

Entity	Description
money/fracunits	Fractional units of currency such as *Cent* or *Penny*.
money/iso4217	ISO 4217 currency codes. For example, *AUD* or *USD*.
money/currency	Currency name. For example, *Algerian dinar*.
money/currencyabbrev	Abbreviated currency name. For example, *dinar* or *dollar*.
money/denom_us	U.S. denominations. For example, *penny* or *quarter*.
money/symbol	Currency symbols. For example, *$* or *€*.

N

number_banking_au.ecr

Entity	Description
number/bsb/au	Australian bank state branch number. For example, *34* or *985*.

number_banking_ca.ecr

Entity	Description
number/cpa_transit_micr/ca	Canadian Payments Association MICR transit number, in the format *BBBBB-AAA*, where *BBBB* is a five-digit code that identifies the branch, and *AAA* is a three-digit code that identifies the institution. For example, *25539-001*.
number/cpa_transit_eft/ca	Canadian Payments Association EFT transit number, in the format *0AAABBBBB*, where *AAA* is a three-digit code that identifies the institution, and *BBBB* is a five-digit code that identifies the branch. The first digit is always a leading zero. For example, *000125539*.
number/cpa_transit/ca	Canadian Payments Association transit number.
number/bankaccount/ca	Canadian bank account number. The account number format can be a known format for particular banks, or a generic seven- or 12-digit number. Known formats are given higher scores. This entity does not include the CPA transit numbers.

number_banking_de.ecr

Entity	Description
number/sort_code/de	8-digit German bank sort code. For example, *10019610*.
number/bank_number/de	German bank account number.

number_banking_fr.ecr

Entity	Description
number/bankaccount/fr	French bank account number.

number_banking_gb.ecr

Entity	Description
number/sortcode/gb	United Kingdom bank sort code. *For example, 301007, 30-10-07, or 30 10 07.* This entity recognizes any valid sort code, but assigns higher scores to known formats from several banks.

number_banking_gb.ecr, continued

Entity	Description
number/bankaccount/gb	United Kingdom bank account number, including the sort code. The sort code and account number must be separated by white space. The account number can be any eight-digit number.

number_banking_ie.ecr

Entity	Description
number/sortcode/ie	Ireland bank sort code. For example, *906005*, *90-60-05*, or *90 60 05*.
number/bankaccount/ie	Ireland bank account number.

number_banking_us.ecr

Entity	Description
number/aba_micr/us	American Bankers Association MICR transit number, in the format *XXXXYYYYC*, where *XXXX* is the Federal Reserve Routing Symbol, *YYYY* is the ABA Institution Identifier, and *C* is the check digit. For example, *129131673*.
number/aba_fraction/us	American Bankers Association fraction transit number, in the format *PP-YYYY/XXXX*, where *PP* is a one-digit or two-digit prefix that represents the bank's check processing center location, *YYYY* is the ABA Institution identifier, and *XXXX* is the Federal Reserve Routing Symbol.
number/aba_routing/us	American Bankers Association transit number.
number/bankaccount/us	United States bank account number, including the American Bankers Association routing number, in fraction or MICR format. The routing information and account information must be separated by a single space. The account number can be four to 17 digits, but nine-, ten-, and 12-digit numbers are given higher scores.

number_bsn_nl.ecr

Entity	Description
number/bsn/nl	Dutch Citizen Service Numbers (burgerservicenummer). BSNs always consist of nine digits.

number_cc.ecr

Entity	Description
number/cc12dn	12-digit credit card numbers with no delimiters.
number/cc12dh	12-digit credit card numbers with hyphen delimiters.
number/cc12ds	12-digit credit card numbers with space delimiters.
number/cc12	All 12-digit credit card numbers.
number/cc13dn	13-digit credit card numbers with no delimiters.
number/cc13dh	13-digit credit card numbers with hyphen delimiters.
number/cc13ds	13-digit credit card numbers with space delimiters.
number/cc13	All 13-digit credit card numbers.
number/cc14dn	14-digit credit card numbers with no delimiters.
number/cc14dh	14-digit credit card numbers with hyphen delimiters.
number/cc14ds	14-digit credit card numbers with space delimiters.
number/cc14	All 14-digit credit card numbers.
number/cc15dn	15-digit credit card numbers with no delimiters.
number/cc15dh	15-digit credit card numbers with hyphen delimiters.
number/cc15ds	15-digit credit card numbers with space delimiters.
number/cc15	All 15-digit credit card numbers.
number/cc16dn	16-digit credit card numbers with no delimiters.
number/cc16dh	16-digit credit card numbers with hyphen delimiters.
number/cc16ds	16-digit credit card numbers with space delimiters.
number/cc16	All 16-digit credit card numbers.
number/cc17dn	17-digit credit card numbers with no delimiters.

number_cc.ecr, continued

Entity	Description
number/cc17dh	17-digit credit card numbers with hyphen delimiters.
number/cc17ds	17-digit credit card numbers with space delimiters.
number/cc17	All 17-digit credit card numbers.
number/cc18dn	18-digit credit card numbers with no delimiters.
number/cc18dh	18-digit credit card numbers with hyphen delimiters.
number/cc18ds	18-digit credit card numbers with space delimiters.
number/cc18	All 18-digit credit card numbers.
number/cc19dn	19-digit credit card numbers with no delimiters.
number/cc19dh	19-digit credit card numbers with hyphen delimiters.
number/cc19ds	19-digit credit card numbers with space delimiters.
number/cc19	All 19-digit credit card numbers.
number/ccdn	All credit card numbers with no delimiters.
number/ccdh	All credit card numbers with hyphen delimiters.
number/ccds	All credit card numbers with space delimiters.
number/cc	Any credit card number. Micro Focus Eduction supports the following credit card formats: • American Express • Bankcard • China Union Pay • DanKort • Diners Club Carte Blanche • Diners Club International • Diners Club enRoute • Discover • InstaPayment • JCB • Laser

number_cc.ecr, continued

Entity	Description
	• Maestro • Mastercard • Solo • Switch • Visa
number/cc_amex	American Express credit card number. American Express credit card account numbers are 15 digits in lengths, and generally start with either *34* or *37*. For example, *378124403602370*.
number/cc_bankcard	Bankcard credit card number (discontinued in 2006).
number/cc_china_union_pay	China UnionPay credit card number. Most China UnionPay card numbers have prefixes from *620* to *625*, and range in length from 16 to 19 characters.
number/cc_diners_club	Diners Club credit card number. Most Diners Club credit card numbers are 16 or 14 digits long. For example, *30544726571210* (Carte Blanche), *36072371463677* (International), or *5484308289255581* (North America).
number/cc_discover	Discover credit card number. Discover credit card numbers start with *6011*, *622126* to *622925*, *644*, *645*, *646*, *647*, *648*, *649*, or *65*, and are 16 digits long. For example, *6011541256841963*.
number/cc_instapayment	InstaPayment credit card number. InstaPayment credit card numbers start with either *637*, *638*, or *639*, and are 16 digits long. For example, *6393519709142682*.
number/cc_jcb	JCB credit card number. JCB credit card numbers consist of 16 digits. Either the first four digits must be *3088*, *3096*, *3112*, *3158*, or *3337*, or the first eight digits must be in the range *35280000* to *35899999*. For example, *3158745776935953*.
number/cc_laser	Laser credit card number (discontinued in 2014). Laser credit card numbers start with *6304*, *6706*, *6771*, or *6709*, and are between 16 to 19 digits long. For example, *6709682431878947*.
number/cc_maestro	Maestro credit card number. Maestro credit card

number_cc.ecr, continued

Entity	Description
	numbers start with *5018*, *5020*, *5038*, *5893*, *6304*, *6759*, *6761*, *6762*, or *6763*, and are between 16 to 19 digits long (although they can have as few as 12 digits). For example, *5018452935461261*.
number/cc_mastercard	Mastercard credit card number. Mastercard credit card numbers start with *51*, *52*, *53*, *54*, or *55*, and are between 16 to 19 digits long.
number/cc_solo	16-digit, 18-digit, or 19-digit Solo credit card number (discontinued in 2011). For example,*6331101999990016*.
number/cc_switch	16-digit, 18-digit, or 19-digit Switch credit card number (rebranded as Maestro). Switch credit card numbers begin with *4903*, *4905*, *4911*, *4936*, *564182*, *633110*, *6333*, or *6759*.
number/cc_visa	Visa credit card number. Most Visa credit card numbers start with *4* and are 16 digits long; however, there are a few that consist of 13 digits. The numbers are always spaced in four groups of four digits each. For example, *4929 8198 5006 5312*.

number_dni_es.ecr

Entity	Description
number/dni/es	Spanish DNI.

number_driverlic_ca.ecr

Entity	Description
number/driverlic/AB/ca number/driverlic/BC/ca number/driverlic/MB/ca number/driverlic/NB/ca number/driverlic/NL/ca number/driverlic/NS/ca number/driverlic/NT/ca	Driver's licence number for each Canadian province and territory. The two-letter codes are defined by ISO 3166-2.

number_driverlic_ca.ecr, continued

Entity	Description
number/driverlic/NU/ca number/driverlic/ON/ca number/driverlic/PE/ca number/driverlic/QC/ca number/driverlic/SK/ca number/driverlic/YT/ca	
number/driverlic/ca	All Canadian driver's licence numbers.

number_driverlic_de.ecr

Entity	Description
number/driverlic/de	German driver's licence number.

number_driverlic_fr.ecr

Entity	Description
number/driverlic/fr	French driver's licence number.

number_driverlic_gb.ecr

Entity	Description
number/driverlic/gb	United Kingdom driving licence number.

number_driverlic_us.ecr

Entity	Description
number/driverlic/AL/us number/driverlic/AK/us number/driverlic/AR/us number/driverlic/AZ/us number/driverlic/CA/us number/driverlic/CO/us	Driver's licence number for each U.S. state.

number_driverlic_us.ecr, continued

Entity	Description
number/driverlic/CT/us	
number/driverlic/DC/us	
number/driverlic/DE/us	
number/driverlic/FL/us	
number/driverlic/GA/us	
number/driverlic/HI/us	
number/driverlic/IA/us	
number/driverlic/ID/us	
number/driverlic/IL/us	
number/driverlic/IN/us	
number/driverlic/KS/us	
number/driverlic/KY/us	
number/driverlic/LA/us	
number/driverlic/MA/us	
number/driverlic/MD/us	
number/driverlic/ME/us	
number/driverlic/MI/us	
number/driverlic/MN/us	
number/driverlic/MO/us	
number/driverlic/MS/us	
number/driverlic/MT/us	
number/driverlic/NC/us	
number/driverlic/ND/us	
number/driverlic/NE/us	
number/driverlic/NH/us	
number/driverlic/NJ/us	
number/driverlic/NM/us	
number/driverlic/NV/us	
number/driverlic/NY/us	
number/driverlic/OH/us	

number_driverlic_us.ecr, continued

Entity	Description
number/driverlic/OK/us number/driverlic/OR/us number/driverlic/PA/us number/driverlic/RI/us number/driverlic/SC/us number/driverlic/SD/us number/driverlic/TN/us number/driverlic/TX/us number/driverlic/UT/us number/driverlic/VA/us number/driverlic/VT/us number/driverlic/WA/us number/driverlic/WV/us number/driverlic/WI/us number/driverlic/WY/us	
number/driverlic/us	All U.S. driver's licence numbers.

number/iban.ecr

Entity	Description
number/ibandn/albania number/ibands/albania number/ibandn/andorra number/ibands/andorra number/ibandn/austria number/ibands/austria number/ibandn/bahrain number/ibands/bahrain number/ibandn/belgium number/ibands/belgium number/ibandn/bosniaherzegovina	Undelimited (dn) or space-delimited (ds) International Bank Account Number (IBAN) for each country. For more information on IBAN formatting requirements for each country, see https://www.iban.com/structure.html.

number/iban.ecr, continued

Entity	Description
number/ibands/bosniaherzegovina	
number/ibandn/bulgaria	
number/ibands/bulgaria	
number/ibandn/costarica	
number/ibands/costarica	
number/ibandn/croatia	
number/ibands/croatia	
number/ibandn/cyprus	
number/ibands/cyprus	
number/ibandn/czechrepublic	
number/ibands/czechrepublic	
number/ibandn/denmark	
number/ibands/denmark	
number/ibandn/dominicanrepublic	
number/ibands/dominicanrepublic	
number/ibandn/estonia	
number/ibands/estonia	
number/ibandn/finland	
number/ibands/finland	
number/ibandn/france	
number/ibands/france	
number/ibandn/georgia	
number/ibands/georgia	
number/ibandn/germany	
number/ibands/germany	
number/ibandn/gibraltar	
number/ibands/gibraltar	
number/ibandn/greece	
number/ibands/greece	
number/ibandn/hungary	

number/iban.ecr, continued

Entity	Description
number/ibands/hungary	
number/ibandn/iceland	
number/ibands/iceland	
number/ibandn/ireland	
number/ibands/ireland	
number/ibandn/israel	
number/ibands/israel	
number/ibandn/italy	
number/ibands/italy	
number/ibandn/kazakhstan	
number/ibands/kazakhstan	
number/ibandn/kuwait	
number/ibands/kuwait	
number/ibandn/latvia	
number/ibands/latvia	
number/ibandn/lebanon	
number/ibands/lebanon	
number/ibandn/liechtenstein	
number/ibands/liechtenstein	
number/ibandn/lithuania	
number/ibands/lithuania	
number/ibandn/luxembourg	
number/ibands/luxembourg	
number/ibandn/macedonia	
number/ibands/macedonia	
number/ibandn/malta	
number/ibands/malta	
number/ibandn/mauritania	
number/ibands/mauritania	
number/ibandn/mauritius	

number/iban.ecr, continued

Entity	Description
number/ibands/mauritius	
number/ibandn/monaco	
number/ibands/monaco	
number/ibandn/montenegro	
number/ibands/montenegro	
number/ibandn/netherlands	
number/ibands/netherlands	
number/ibandn/norway	
number/ibands/norway	
number/ibandn/poland	
number/ibands/poland	
number/ibandn/portugal	
number/ibands/portugal	
number/ibandn/romania	
number/ibands/romania	
number/ibandn/sanmarino	
number/ibands/sanmarino	
number/ibandn/saudiarabia	
number/ibands/saudiarabia	
number/ibandn/serbia	
number/ibands/serbia	
number/ibandn/slovakrepublic	
number/ibands/slovakrepublic	
number/ibandn/slovenia	
number/ibands/slovenia	
number/ibandn/spain	
number/ibands/spain	
number/ibandn/sweden	
number/ibands/sweden	
number/ibandn/switzerland	

number/iban.ecr, continued

Entity	Description
number/ibands/switzerland number/ibandn/tunisia number/ibands/tunisia number/ibandn/turkey number/ibands/turkey number/ibandn/unitedarabemirates number/ibands/unitedarabemirates number/ibandn/unitedkingdom number/ibands/unitedkingdom	
number/ibandn	All IBAN numbers without delimiters.
number/ibands	All IBAN numbers with space delimiters.

number_insee_fr.ecr

Entity	Description
number/insee/fr	French INSEE number. INSEE numbers are composed of 13 digits and a two-digit key. `Score="0.2"` is used for examples with unspecified months.

number_licenseplate_ca.ecr

Entity	Description
number/licenseplate/AB/ca number/licenseplate/BC/ca number/licenseplate/MB/ca number/licenseplate/NB/ca number/licenseplate/NL/ca number/licenseplate/NT/ca number/licenseplate/NS/ca number/licenseplate/NU/ca	Licence plate numbers for each Canadian province and territory.

number_licenseplate_ca.ecr, continued

Entity	Description
number/licenseplate/ON/ca number/licenseplate/PE/ca number/licenseplate/QC/ca number/licenseplate/SK/ca number/licenseplate/YT/ca	
number/licenseplate/ca	All Canadian licence plate numbers.

number_licenseplate_de.ecr

Entity	Description
number/licenseplate/de	German vehicle number plate.

number_licenseplate_es.ecr

Entity	Description
number/licenseplate/es	Spanish vehicle number plate.

number_licenseplate_fr.ecr

Entity	Description
number/licenseplate/fr	French vehicle registration number.

number_licenseplate_gb.ecr

Entity	Description
number/licenseplate/gb	United Kingdom vehicle registration number.

number_licenseplate_us.ecr

Entity	Description
number/licenseplate/AL/us number/licenseplate/AK/us	Licence plate numbers for each U.S. state.

number_licenseplate_us.ecr, continued

Entity	Description
number/licenseplate/AR/us	
number/licenseplate/AZ/us	
number/licenseplate/CA/us	
number/licenseplate/CO/us	
number/licenseplate/CT/us	
number/licenseplate/DE/us	
number/licenseplate/DC/us	
number/licenseplate/FL/us	
number/licenseplate/GA/us	
number/licenseplate/HI/us	
number/licenseplate/IA/us	
number/licenseplate/ID/us	
number/licenseplate/IL/us	
number/licenseplate/IN/us	
number/licenseplate/KS/us	
number/licenseplate/KY/us	
number/licenseplate/LA/us	
number/licenseplate/MA/us	
number/licenseplate/MD/us	
number/licenseplate/ME/us	
number/licenseplate/MI/us	
number/licenseplate/MN/us	
number/licenseplate/MO/us	
number/licenseplate/MS/us	
number/licenseplate/MT/us	
number/licenseplate/NC/us	
number/licenseplate/ND/us	
number/licenseplate/NE/us	
number/licenseplate/NH/us	
number/licenseplate/NJ/us	

number_licenseplate_us.ecr, continued

Entity	Description
number/licenseplate/NM/us number/licenseplate/NV/us number/licenseplate/NY/us number/licenseplate/OH/us number/licenseplate/OK/us number/licenseplate/OR/us number/licenseplate/PA/us number/licenseplate/RI/us number/licenseplate/SC/us number/licenseplate/SD/us number/licenseplate/TN/us number/licenseplate/TX/us number/licenseplate/UT/us number/licenseplate/VA/us number/licenseplate/VT/us number/licenseplate/WA/us number/licenseplate/WV/us number/licenseplate/WI/us number/licenseplate/WY/us	
number/licenseplate/us	United States license plate number.

number_mac_address.ecr

Entity	Description
number/EUI48dh	MAC address in EUI-48 format (hyphen-separated). For example, *01-23-45-67-89-Ab*
number/EUI48dc	MAC address in EUI-48 format (colon-separated). For example, *01:23:45:67:89:Ab*
number/EUI48	MAC address in EUI-48 format. For example, *01-23-45-67-89-Ab*
number/EUI64dh	MAC address in EUI-64 format (hyphen-separated).

number_mac_address.ecr, continued

Entity	Description
	For example, *01-23-45-67-89-ab-CD-eF*
number/EUI64dc	MAC address in EUI-64 format (colon-separated). For example, *01:23:45:67:89:ab:CD:eF*
number/EUI64	MAC address in EUI-64 format. For example, *01-23-45-67-89-ab-CD-eF*

number_ni_gb.ecr

Entity	Description
number/nids/gb	UK National Insurance number with space delimiters.
number/nidn/gb	UK National Insurance number without delimiters.
number/nidh/gb	UK National Insurance number with hyphen delimiters.
number/ni/gb	Any UK National Insurance number. The format of the number is two prefix letters, six digits, and one suffix letter.

number_passport_engca.ecr

Entity	Description
number/passport_number/engca	Canadian passport number (in any context).
number/passport_context/engca	Canadian passport number (when found in English-language context).

number_passport_enggb.ecr

Entity	Description
number/passport_context/enggb	UK passport number (when found in English-language context).

number_passport_engus.ecr

Entity	Description
number/passport_context/engus	U.S. passport number (when found in English-language context).

number_passport_freca.ecr

Entity	Description
number/passport_number/freca	French Canadian passport number (in any context).
number/passport_context/freca	French Canadian passport number (when found in French-language context).

number_passport_frefr.ecr

Entity	Description
number/passport_number/frefr	French passport number (in any context).
number/passport_context/frefr	French passport number (when found in French-language context).

number_passport_gerde.ecr

Entity	Description
number/passport_context/gerde	German passport number (when found in German-language context).

number_phone_au.ecr

Entity	Description
phone/landline/au	A complete landline phone number in Australia. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/au	A complete mobile phone number in Australia. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/au	A complete 08- or 09- phone number in Australia. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/au	Any complete phone number in Australia. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_be.ecr

Entity	Description
phone/landline/be	A complete landline phone number in Belgium. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/be	A complete mobile phone number in Belgium. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/be	A complete 08- or 09- phone number in Belgium. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/be	Any complete phone number in Belgium. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_ca.ecr

Entity	Description
phone/numds/ca	A numeric-only Canadian phone number, delimited by spaces. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdh/ca	A numeric-only Canadian phone number, delimited by hyphens. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdd/ca	A numeric-only Canadian phone number, delimited by dots. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdn/ca	An undelimited, numeric-only Canadian phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/num/ca	Any numeric-only Canadian phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/alphanumds/ca	An alphanumeric Canadian phone number, delimited by spaces.
phone/alphanumdh/ca	An alphanumeric Canadian phone number, delimited by hyphens. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '

number_phone_ca.ecr, continued

Entity	Description
	(.
phone/alphanumdd/ca	An alphanumeric Canadian phone number, delimited by dots.
phone/alphanumdn/ca	An undelimited, alphanumeric Canadian phone number.
phone/alphanum/ca	Any alphanumeric Canadian phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_cn.ecr

Entity	Description
phone/landline/cn	A Chinese landline phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/cn	A Chinese mobile phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/tollfree/cn	A Chinese toll free phone number.
phone/all/cn	Any Chinese phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_de.ecr

Entity	Description
phone/landline/de	A complete landline phone number in Germany. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/de	A complete mobile phone number in Germany. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/de	A complete freephone or premium phone number in Germany. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_de.ecr, continued

Entity	Description
phone/all/de	Any complete German phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_es.ecr

Entity	Description
phone/landline/es	A complete landline phone number in Spain. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/es	A complete mobile phone number in Spain. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/es	A complete freephone or premium phone number in Spain. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/es	Any complete phone number in Spain. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_fr.ecr

Entity	Description
phone/landline/fr	A complete landline phone number in France. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/fr	A complete mobile phone number in France. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/fr	A complete 08- or 09- phone number in France. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/fr	Any complete phone number in France. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_gb.ecr

Entity	Description
phone/areacode/gb	United Kingdom area code.
phone/landline/gb	A complete landline phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/gb	A complete mobile phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/freephone/gb	A complete freephone phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/business/gb	A complete 08- or 09- phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/non_geographic/gb	A complete non-geographic phone number in the United Kingdom. For example, *0345 678 579 40.* To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/personal/gb	A complete 070- phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/gb	Any complete phone number in the United Kingdom. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_it.ecr

Entity	Description
phone/landline/it	A complete landline phone number in Italy. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/it	A complete mobile phone number in Italy. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/it	A premium rate, freephone, or shared-cost phone number in Italy. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '

number_phone_it.ecr, continued

Entity	Description
	(.
phone/all/it	Any complete phone number in Italy. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_lu.ecr

Entity	Description
phone/landline/lu	A complete landline phone number in Luxembourg. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/lu	A complete mobile phone number in Luxembourg. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/lu	Any complete phone number in Luxembourg. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_nl.ecr

Entity	Description
phone/landline/nl	A complete landline phone number in the Netherlands. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/nl	A complete mobile phone number in the Netherlands. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/nl	A complete 08- or 09- phone number in the Netherlands. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/nl	Any complete phone number in the Netherlands. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_pt.ecr

Entity	Description
phone/landline/pt	A complete landline phone number in Portugal. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/mobile/pt	A complete mobile phone number in Portugal. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/other/pt	Other complete phone number in Portugal. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/all/pt	Any complete phone number in Portugal. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.

number_phone_us.ecr

Entity	Description
phone/numds/us	A numeric-only U.S. phone number, delimited by spaces. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdh/us	A numeric-only U.S. phone number, delimited by hyphens. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdd/us	A numeric-only U.S. phone number, delimited by dots.To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/numdn/us	An undelimited, numeric-only U.S. phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/num/us	Any numeric-only U.S. phone number. To ensure that this entity performs correctly, set `TangibleCharacters` to include '+' and '('.
phone/alphanumds/us	An alphanumeric U.S. phone number, delimited by spaces.
phone/alphanumdh/us	An alphanumeric U.S. phone number, delimited by hyphens.

number_phone_us.ecr, continued

Entity	Description
phone/alphanumdd/us	An alphanumeric U.S. phone number, delimited by dots.
phone/alphanumdn/us	An undelimited, alphanumeric U.S. phone number.
phone/alphanum/us	Any alphanumeric U.S. phone number.

number_sin_ca.ecr

Entity	Description
number/sindh/ca	Canadian social insurance number with dash delimiters.
number/sinds/ca	Canadian social insurance number with space delimiters.
number/sindn/ca	Canadian social insurance number without delimiters.
number/sin/ca	Any Canadian social security number.

number_ss_us.ecr

Entity	Description
number/ssdh/us	Social Security number with dash delimiters.
number/ssdsh/us	Social Security number with soft hyphen delimiters.
number/ssds/us	Social Security number with space delimiters.
number/ssdnbs/us	Social Security number with non-breaking space delimiters.
number/ssdn/us	Social Security number without delimiters.
number/ss/us	Any Social Security number.
number/medicareid/us	Medicare ID.

number_swiftcode.ecr

Entity	Description
number/swiftcode	Swift code.

number_telecoms.ecr

Entity	Description
number/imei	International Mobile Station Equipment Identity number.
number/imeisv	International Mobile Station Equipment Identity Software Version number.
number/meid_hex	Mobile Equipment Identifier (hexadecimal format).
number/iccid	Integrated Circuit Card Identifier number.
number/imsi	International Mobile Subscriber Identity number.
number/plmn	Public Land Mobile Network number.
number/msisdn	Mobile Subscriber Integrated Services Digital Network number.

number_types_chi.ecr

Entity	Description
number/one_to_nine/chi	The numbers one to nine in Chinese.
number/zero_to_nine/chi	The numbers zero to nine in Chinese.
number/zero_to_twelve/chi	The numbers zero to twelve in Chinese.
number/zero_to_twenty_four/chi	The numbers zero to twenty-four in Chinese.
number/one_to_thirty_one/chi	The numbers one to thirty-one in Chinese.
number/zero_to_fifty_five/chi	The numbers zero to fifty-five in steps of five in Chinese.
number/zero_to_fifty_nine/chi	The numbers zero to fifty-nine in Chinese.
number/one_to_ninety_nine/chi	The numbers one to ninety-nine in Chinese.
number/one_to_one_hundred/chi	The numbers one to one hundred in Chinese.
number/all/chi	Large numbers in Chinese.
number/num/chi	A simple string of digits that does not start with a zero in Chinese.
number/digits/chi	String of digits in Chinese.
number/fraction/chi	A simple fraction consisting of two strings of digits that do not start with a zero. For example, *-12/13, 1/5*.

number_types_eng.ecr

Entity	Description
number/num/eng	A simple string of digits that does not start with a zero. For example, *123*.
number/ncomma/eng	A number without commas. For example, *123456.*
number/comma/eng	A number with commas. For example, *123,456.*
number/sign/eng	A sign. For example +, -, *plus*, *minus.*
number/natural/eng	A natural number. For example, *123,456.*
number/int/eng	An integer. For example, *-123*,*456*, *minus 2*, *+20.*
number/real/eng	A real number. For example, *123.456*, *-123.456.*
number/fraction/eng	A simple fraction of two unsigned strings of digits that do not start with a zero. For example, *12/13*.
number/fracalpha/eng	An alphabetical fraction. For example, *one half*, *three sixths*, *three and five ninths.*
number/fracnum/eng	Numeric fractions. For example, *-12/13.*
number/fracmixed/eng	Mixed fractions. For example, *1 twelfth*, *8 fourteenths*, *3 and five ninths.*
number/pcnt/eng	Percent. For example, *100%*, *100 percent*, *12.78%.*
number/suff/eng	Numbers with a suffix. For example, *1st*, *3rd.*
number/suffalpha/eng	Fractions with mixed alphabetical and numeric terms. For example, *a 12th*, *three 3rds.*
number/doz/eng	Number based on dozen. For example, *half a dozen*, *2 dozen*, *three and a half dozen*.
number/alpha/eng	Alphabetical numbers less than 100. For example, *one*, *ten*, *thirty-one.*
number/bigalpha/eng	Big alphabetical numbers.
number/bignum/eng	Big numeric abbreviated numbers.
number/big/eng	A big number, alphabetical or numeric abbreviated.
number/sci/eng	A number in scientific notation. For example, *1.23x10^11*, *1.23E+5, 6.1^-3.*
number/fullalpha/eng	A fully written out number up to 999,999,999,999,999. For example, *one thousand two hundred and thirty*

number_types_eng.ecr, continued

Entity	Description
	four.
number/numord/eng	An ordinal number up to 999. For example, *thirty fourth*.
number/num_plurals/eng	Plural numbers. For example, *dozens, millions*.

number_types_fre.ecr

Entity	Description
number/num/fre	A simple string of digits that does not start with 0. For example, *123*.
number/num_sep/fre	A number with separators. For example, *123.456.789*.
number/digits/fre	A string of digits. For example, *00123*.
number/natural/fre	A natural number. For example, *123.456.789* or *123456789*.
number/fraction/fre	A simple fraction of two unsigned strings of digits that do not start with a zero. For example, *12/13*.
number/int/fre	An integer. For example, *-123.456.789*; *moins 2*; *20*.
number/real/fre	A real number. For example, *123.456*, *-123.456*.
number/numalpha/fre	A fully written out number up to 999. For example, *deux cent trente-quatre*.
number/numord/fre	An ordinal number up to 999. For example, *trente quatrième*.

number_vin.ecr

Entity	Description
number/vin/wmi	The world manufacturer identifier section (3 characters) of a vehicle identification number.
number/vin/vds	The vehicle descriptor section (6 characters) of a vehicle identification number.
number/vin/model_year	The model year character of a vehicle identification number.

number_vin.ecr, continued

Entity	Description
number/vin/plant_code	The plant code character of a vehicle identification number.
number/vin/seq_number	The vehicle identifier section sequential number (6 characters) of a vehicle identification number.
number/vin/vis	The vehicle identifier section (8 characters) of a vehicle identification number.
number/vin	A vehicle identification number (17 characters).
number/vin/anonymized	An anonymized vehicle identification number (first 9 or 11 characters).

O

organization.ecr

Entity	Description
org/organization	An organization.

P

person_name_chicn.ecr

Entity	Description
person/femalefirstname_s/chicn	Popular simplified Chinese female first name.
person/malefirstname_s/chicn	Popular simplified Chinese male first name.
person/lastname_s/chicn	Popular simplified Chinese last name.
person/firstname/chicn	Chinese first name.
person/namelastfirst/chicn	Chinese last and first name.

person_name_dutnl.ecr

Entity	Description
person/femalefirstname/dutnl	Popular Dutch female first name.

person_name_dutnl.ecr, continued

Entity	Description
person/malefirstname/dutnl	Popular Dutch male first name.
person/firstname/dutnl	Dutch first name.
person/surname/dutnl	Dutch surname.
person/namefirstmiddlelast/dutn	Dutch first, optional middle, and last name.

person_name_engcn.ecr

Entity	Description
person/femalefirstname/engcn	Popular Chinese female first name in English.
person/femalefirstname_lowercase/engcn	Popular Chinese female first name in lowercase English.
person/malefirstname/engcn	Popular Chinese male first name in English.
person/malefirstname_lowercase/engcn	Popular Chinese male first name in lowercase English.
person/lastname/engcn	Popular Chinese last name in English.
person/firstname/engcn	Chinese first name in English.
person/namelastfirst/engcn	Chinese last and first name in English.
person/namefirstlast/engcn	Chinese first and last name in English.

person_name_enggb.ecr

Entity	Description
person/femalefirstname/enggb	Popular UK female first name.
person/malefirstname/enggb	Popular UK male first name.
person/lastname/enggb	Popular UK last name.
person/firstname/enggb	UK first name.
person/namefirstlast/enggb	UK first and last name.
person/namefirstmiddlelast/enggb	UK first, optional middle, and last name.

person_name_enggr.ecr

Entity	Description
person/femalefirstname/enggr	Popular Greek female first name.
person/malefirstname/enggr	Popular Greek male first name.
person/lastname/enggr	Popular Greek last name.
person/firstname/enggr	Greek first name.
person/namefirstlast/enggr	Greek first and last name.

person_name_engin.ecr

Entity	Description
person/femalefirstname/engin	Popular Indian female first name.
person/malefirstname/engin	Popular Indian male first name.
person/lastname/engin	Popular Indian last name.
person/firstname/engin	Indian first name.
person/namefirstlast/engin	Indian first and last name.
person/namefirstmiddlelast/engin	Indian first, optional middle, and last name.

person_name_engjp.ecr

Entity	Description
person/femalefirstname/engjp	Popular Japanese female first name in English.
person/malefirstname/engjp	Popular Japanese male first name in romanji.
person/lastname/engjp	Popular Japanese last name in English.
person/firstname/engjp	Japanese first name in English.
person/namelastfirst/engjp	Japanese last and first name in English.

person_name_engru.ecr

Entity	Description
person/femalefirstname/engru	Popular Russian female first name in English.

person_name_engru.ecr, continued

Entity	Description
person/malefirstname/engru	Popular Russian male first name in English.
person/lastname/engru	Popular Russian last name in English.
person/firstname/engru	Russian first name in English.
person/namefirstlast/engru	Russian first and last name in English.
person/namefirstmiddlelast/engru	Russian first, optional middle, and last name in English.

person_name_engus.ecr

Entity	Description
person/femalefirstname/engus	Popular U.S. female first name.
person/malefirstname/engus	Popular U.S. male first name.
person/lastname/engus	Popular U.S. last name.
person/firstname/engus	U.S. first name.
person/compoundlastname/engus	U.S. last name that might be compound.
person/namefirstlast/engus	U.S. first and last name.
person/namefirstmiddlelast/engus	U.S. first, optional middle, and last name.
person/nameinitial/engus	U.S. initialed name.
person/namelastsuffix/engus	Last name and suffix.
person/namefirstneelast/engus	First name and maiden name.
person/namelastcommafirst/engus	Full name in address book format (*Last Name, First Name*).

person_name_frefr.ecr

Entity	Description
person/femalefirstname/frefr	Popular French female first name.
person/malefirstname/frefr	Popular French male first name.
person/lastname/frefr	Popular French last name.

person_name_frefr.ecr, continued

Entity	Description
person/firstname/frefr	French first name.
person/namefirstlast/frefr	French first and last name.
person/namefirstmiddlelast/frefr	French first, optional middle, and last name.

person_name_gerde.ecr

Entity	Description
person/femalefirstname/gerde	Popular German female first name.
person/malefirstname/gerde	Popular German male first name.
person/lastname/gerde	Popular German last name.
person/firstname/gerde	German first name.
person/namefirstmiddlelast/gerde	German first, optional middle, and last name.

person_name_itait.ecr

Entity	Description
person/femalefirstname/itait	Popular Italian female first name.
person/malefirstname/itait	Popular Italian male first name.
person/lastname/itait	Popular Italian last name.
person/firstname/itait	Italian first name.
person/namefirstlast/itait	Italian first and last name.
person/namefirstmiddlelast/itait	Italian first, optional middle, and last name.

person_name_jpnjp.ecr

Entity	Description
person/femalefirstname/jpnjp	Popular Japanese female first name.
person/malefirstname/jpnjp	Popular Japanese male first name in kanj.
person/lastname/jpnjp	Popular Japanese last name in kanj.

person_name_jpnjp.ecr, continued

Entity	Description
person/firstname/jpnjp	Japanese first name.
person/namelastfirst/jpnjp	Japanese last and first name.

person_name_norno.ecr

Entity	Description
person/femalefirstname/norno	Popular Norwegian female first name.
person/malefirstname/norno	Popular Norwegian male first name.
person/lastname/norno	Popular Norwegian last name.
person/firstname/norno	Norwegian first name.
person/namefirstlast/norno	Norwegian first and last name.
person/namefirstmiddlelast/norno	Norwegian first, optional extra given name, optional middle name, and last name.

person_name_rusru.ecr

Entity	Description
person/femalefirstname_unambiguous	Common Russian female first name in Russian that rarely has any alternative meaning.
person/malefirstname_unambiguous	Common Russian male first name in Russian that rarely has any alternative meaning.
person/femalefirstname/rusru	Russian female first name in Russian.
person/femalelastname/rusru	Russian female last name in Russian.
person/malefirstname/rusru	Russian male first name in Russian.
person/malelastname/rusru	Russian male last name in Russian.
person/firstname/rusru	Russian first name in Russian.
person/lastname/rusru	Russian last name in Russian.
person/fullname/rusru	Russian full name in Russian.

person_name_spaes.ecr

Entity	Description
person/femalefirstname/spaes	Popular Spanish female first name.
person/malefirstname/spaes	Popular Spanish male first name.
person/lastname/spaes	Popular Spanish last name.
person/firstname/spaes	Spanish first name.
person/compoundlastname/spaes	Spanish compound last name.
person/namefirstoptionallast/spaes	Spanish first and optional last name.
person/namefirstlast/spaes	Spanish first and last name.
person/namelastfirst/spaes	Spanish last name, comma, and first name.
person/fullname	Spanish full name.

person_name_swese.ecr

Entity	Description
person/femalefirstname/swese	Popular Swedish female first name.
person/malefirstname/swese	Popular Swedish male first name.
person/lastname/swese	Popular Swedish last name.
person/firstname/swese	Swedish first name.
person/namefirstlast/swese	Swedish first and last name.

person_politician_engus.ecr

Entity	Description
person/poli_hor/engus	Full names of members of the U.S. House of Representatives. For example, *Robert E. Cramer*, *Robert Cramer.*
person/poli_last_hor/engus	Last name of House of Representatives members. For example, *Cramer.*
person/poli_sen/engus	Full name of U.S. senate members.
person/poli_last_sen/engus	Last name of U.S. senate members.

person_politician_engus.ecr, continued

Entity	Description
person/poli_gov/engus	Full name of U.S. governors.
person/poli_last_gov/engus	Last name of U.S. governors.
person/poli_cabinet_gw_bush/engus	Full name of a member of the George W. Bush administration.
person/poli_last_cabinet_gw_bush/engus	Last name of a member of the George W. Bush administration.
person/poli_cabinet_obama/engus	Full name of a member of the Barack Obama administration.
person/poli_last_cabinet_obama/engus	Last name of a member of the Barack Obama administration.
person/poli_other_2012/engus	Full name of other currently active politician. For example, a Presidential nominee.
person/poli_last_other_2012/engus	Last name of other currently active politician. For example, a Presidential nominee.
person/poli_president/engus	Past and present U.S. Presidents.
person/poli_title_hor/engus	Formal title for legislative members. For example, *Congressman Cramer.*
person/poli_title_sen/engus	Formal title for senate members.
person/poli_title_gov/engus	Formal title for governors.

person_politician_jpnjp.ecr

Entity	Description
person/politician/jpnjp	Japanese politician.

person_public_figure_chi.ecr

Entity	Description
person/politician/chicn	Chinese politician.
person/legislativecouncil/chihk	Hong Kong legislative council member.
person/entertainer/chi	Chinese entertainer.
person/npc/chicn	Chinese National People's Congress delegate.

person_public_figure_eng.ecr

Entity	Description
person/public_figure/eng	A list of public figures in English.

person_public_figure_jpn.ecr

Entity	Description
person/public_figure/jpn	A list of public figures in Japanese.

person_salutation_eng.ecr

Entity	Description
person/salutation/common/eng	Common salutation. For example, *Mr*.
person/salutation/military/eng	Military salutation.
person/salutation/political/eng	Political salutation.
person/salutation/religious/eng	Religious salutation.
person/salutation/nobility/eng	Salutation of nobility.
person/salutation/eng	Any salutation in English.

person_salutation_fre.ecr

Entity	Description
person/salutation/fre	French salutations. For example, *Madame*, *Mlle*.

person_suffix_eng.ecr

Entity	Description
person/suffixjr/eng	Name suffixes. For example, *Jr.*
person/suffixrmn/eng	Roman suffixes. For example, *III.*
person/suffixacab/eng	Academic suffix – Bachelor's. For example, *BA.*
person/suffixacam/eng	Academic suffix – Master's. For example, *MA.*
person/suffixacad/eng	Academic suffix – Doctoral. For example, *PhD*.
personal/suffixprof/eng	Professional suffix. For example, *MD.*

place_albal.ecr

Entity	Description
place/city1/albal	Albanian settlement with over 100,000 inhabitants.
place/city1_uppercase/albal	Albanian settlement with over 100,000 inhabitants, in uppercase.
place/city2/albal	Albanian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/albal	Albanian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/albal	Albanian county.
place/county_uppercase/albal	Albanian county in uppercase.
place/district/albal	Albanian district.
place/district_uppercase/albal	Albanian district in uppercase.

place_albxk.ecr

Entity	Description
place/city1/albxk	Kosovan settlement with over 100,000 inhabitants.
place/city1_uppercase/albxk	Kosovan settlement with over 100,000 inhabitants, in uppercase.
place/city2/albxk	Kosovan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/albxk	Kosovan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/district/albxk	Kosovan district.
place/district_uppercase/albxk	Kosovan district in uppercase.

place_bosba.ecr

Entity	Description
place/city1/bosba	Settlement of Bosnia and Herzegovina with over 100,000 inhabitants.
place/city1_uppercase/bosba	Settlement of Bosnia and Herzegovina with over 100,000 inhabitants, in uppercase.

place_bosba.ecr, continued

Entity	Description
place/city2/bosba	Settlement of Bosnia and Herzegovina with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/bosba	Settlement of Bosnia and Herzegovina with between 10,000 and 100,000 inhabitants, in uppercase.

place_chicn.ecr

Entity	Description
place/city/chicn	Chinese city.
place/province/chicn	Chinese province.

place_chihk.ecr

Entity	Description
place/district/chihk	District in Hong Kong.
place/island/chihk	Island in Hong Kong.
place/port/chihk	Port in Hong Kong.
place/hospital/chihk	Hospital in Hong Kong.
place/tunnel/chihk	Tunnel in Hong Kong.
place/bridge/chihk	Bridge in Hong Kong.
place/hotel/chihk	Hotel in Hong Kong.
place/locality/chihk	Place in Hong Kong.

place_chitw.ecr

Entity	Description
place/city/chitw	Major divisions of Taiwan, including six special municipalities, three provincial cities, and 13 counties.
place/district/chitw	Subdivisions of the major divisions of Taiwan. The subdivisions include city districts, county-controlled cities, urban townships, and rural townships.

place_countries.ecr

Entity	Description
country/iso_lowercase	ISO 3166-1 alpha-2 country code.
country/all	Country in a local or major language.
country/output_iso	Country in a local or major language (output is normalized to the ISO 3166-1 alpha-2 code).

place_czecz.ecr

Entity	Description
place/city1/czecz	Czech settlement with over 100,000 inhabitants.
place/city1_uppercase/czecz	Czech settlement with over 100,000 inhabitants, in uppercase.
place/city2/czecz	Czech settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/czecz	Czech settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/czecz	Czech region.
place/region_uppercase/czecz	Czech region in uppercase.

place_dandk.ecr

Entity	Description
place/city1/dandk	Danish settlement with over 100,000 inhabitants.
place/city1_uppercase/dandk	Danish settlement with over 100,000 inhabitants, in uppercase.
place/city2/dandk	Danish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/dandk	Danish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/dandk	Danish region.
place/region_uppercase/dandk	Danish region in uppercase.
place/municipality/dandk	Danish municipality.

place_dandk.ecr, continued

Entity	Description
place/municipality_uppercase/dandk	Danish municipality in uppercase.
place/island/dandk	Danish island
place/island_uppercase/dandk	Danish island in uppercase

place_dutnl.ecr

Entity	Description
place/city1/dutnl	Dutch settlement with over 100,000 inhabitants.
place/city1_uppercase/dutnl	Dutch settlement with over 100,000 inhabitants, in uppercase.
place/city2/dutnl	Dutch settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/dutnl	Dutch settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/dutnl	Dutch county.
place/county_uppercase/dutnl	Dutch county in uppercase.
place/municipality/dutnl	Dutch municipality.
place/municipality_uppercase/dutnl	Dutch municipality in uppercase.
place/island/dutnl	Dutch island.
place/island_uppercase/dutnl	Dutch island in uppercase.

place_dutsr.ecr

Entity	Description
place/city1/dutsr	Surinamese settlement with over 100,000 inhabitants.
place/city1_uppercase/dutsr	Surinamese settlement with over 100,000 inhabitants, in uppercase.
place/city2/dutsr	Surinamese settlement with under 100,000 inhabitants.
place/city2_uppercase/dutsr	Surinamese settlement with under 100,000 inhabitants, in uppercase.

place_dutsr.ecr, continued

Entity	Description
place/district/dutsr	Surinamese district.
place/district_uppercase/dutsr	Surinamese district in uppercase.

place_engae.ecr

Entity	Description
place/city1/engae	Settlement of the United Arab Emirates with over 100,000 inhabitants.
place/city1_uppercase/engae	Settlement of the United Arab Emirates with over 100,000 inhabitants, in uppercase.
place/city2/engae	Settlement of the United Arab Emirates with under 100,000 inhabitants.
place/city2_uppercase/engae	Settlement of the United Arab Emirates with under 100,000 inhabitants, in uppercase.
place/emirate/engae	Emirate of the United Arab Emirates.
place/emirate_uppercase/engae	Emirate of the United Arab Emirates in uppercase.

place_engau.ecr

Entity	Description
place/state/engau	Australian state or territory.
place/state_uppercase/engau	Australian state or territory in uppercase.
place/state_abbrev/engau	Australian state or territory abbreviations.
place/state_capital/engau	Australian state or territory capitals.
place/state_capital_uppercase/engau	Australian state or territory capitals in uppercase.
place/city1/engau	Australian city with population greater than 100,000.
place/city1_uppercase/engau	Australian city with population greater than 100,000, in uppercase.
place/city2/engau	Australian city with population between 10,000 and 100,000.
place/city2_uppercase/engau	Australian city with population between 10,000 and

place_engau.ecr, continued

Entity	Description
	100,000, in uppercase.
place/city/NSW/engau	Settlement in New South Wales.
place/city_uppercase/NSW/engau	Settlement in New South Wales, in uppercase.
place/city/QLD/engau	Settlement in Queensland, Australia.
place/city_uppercase/QLD/engau	Settlement in Queensland, Australia, in uppercase.
place/city/SA/engau	Settlement in South Australia.
place/city_uppercase/SA/engau	Settlement in South Australia, in uppercase.
place/city/TAS/engau	Settlement in Tasmania.
place/city_uppercase/TAS/engau	Settlement in Tasmania, in uppercase.
place/city/VIC/engau	Settlement in Victoria, Australia.
place/city_uppercase/VIC/engau	Settlement in Victoria, Australia, in uppercase.
place/city/WA/engau	Settlement in Western Australia.
place/city_uppercase/WA/engau	Settlement in Western Australia, in uppercase.
place/city/NT/engau	Settlement in Northern Territory, Australia.
place/city_uppercase/NT/engau	Settlement in Northern Territory, Australia, in uppercase.
place/city/ACT/engau	Settlement in Australian Capital Territory.
place/city_uppercase/ACT/engau	Settlement in Australian Capital Territory, in uppercase.
place/city/engau	Australian cities.
place/city_uppercase/engau	Australian cities in uppercase.

place_engbd.ecr

Entity	Description
place/city1/engbd	Bangladeshi settlement with over 100,000 inhabitants.
place/city1_uppercase/engbd	Bangladeshi settlement with over 100,000 inhabitants, in uppercase.

place_engbd.ecr, continued

Entity	Description
place/city2/engbd	Bangladeshi settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engbd	Bangladeshi settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/division/engbd	Bangladeshi division.
place/division_uppercase/engbd	Bangladeshi division in uppercase.
place/district/engbd	Bangladeshi district.
place/district_uppercase/engbd	Bangladeshi district in uppercase.

place_engbg.ecr

Entity	Description
place/city1/engbg	Bulgarian settlement with over 100,000 inhabitants.
place/city1_uppercase/engbg	Bulgarian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engbg	Bulgarian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engbg	Bulgarian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engbg	Bulgarian province.
place/province_uppercase/engbg	Bulgarian province in uppercase.

place_engby.ecr

Entity	Description
place/city1/engby	Belarusian settlement with over 100,000 inhabitants.
place/city1_uppercase/engby	Belarusian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engby	Belarusian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engby	Belarusian settlement with between 10,000 and 100,000 inhabitants, in uppercase.

place_engby.ecr, continued

Entity	Description
place/region/engby	Belarusian region.
place/region_uppercase/engby	Belarusian region in uppercase.

place_engca.ecr

Entity	Description
place/region/engca	Canadian province or territory.
place/region_uppercase/engca	Canadian province or territory in uppercase.
place/region_abbrev/engca	Canadian province or territory abbreviation.
place/region_all/engca	Canadian province or territory full name or abbreviation.
place/region_all_uppercase/engca	Canadian province or territory full name or abbreviation, in uppercase.
place/region_capitals/engca	Canadian provincial or territorial capital.
place/region_capitals_uppercase/engca	Canadian provincial or territorial capital in uppercase.
place/city1/engca	Canadian settlement with over 100,000 inhabitants.
place/city1_uppercase/engca	Canadian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engca	Canadian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engca	Canadian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/city/AB/engca place/city_uppercase/AB/engca place/city/BC/engca place/city_uppercase/BC/engca place/city/MB/engca place/city_uppercase/MB/engca place/city/NB/engca place/city_uppercase/NB/engca	Settlements in each Canadian province or territory, in normal or uppercase.

place_engca.ecr, continued

Entity	Description
place/city/NL/engca place/city_uppercase/NL/engca place/city/NS/engca place/city_uppercase/NS/engca place/city/ON/engca place/city_uppercase/ON/engca place/city/PE/engca place/city_uppercase/PE/engca place/city/QC/engca place/city_uppercase/QC/engca place/city/SK/engca place/city_uppercase/SK/engca place/city/NT/engca place/city_uppercase/NT/engca place/city/NU/engca place/city_uppercase/NU/engca place/city/YT/engca place/city_uppercase/YT/engca	
place/city/engca	Canadian settlement.
place/city_uppercase/engca	Canadian settlement in uppercase.

place_engcn.ecr

Entity	Description
place/city0/engcn	Chinese settlement with over 1,000,000 inhabitants.
place/city0_uppercase/engcn	Chinese settlement with over 1,000,000 inhabitants, in uppercase.
place/city1/engcn	Chinese settlement with over 100,000 inhabitants.
place/city1_uppercase/engcn	Chinese settlement with over 100,000 inhabitants, in uppercase.

place_engcn.ecr, continued

Entity	Description
place/city2/engcn	Chinese settlement with over 10,000 inhabitants.
place/city2_uppercase/engcn	Chinese settlement with over 10,000 inhabitants, in uppercase.
place/province/engcn	Chinese province.
place/province_uppercase/engcn	Chinese province in uppercase.

place_enggb.ecr

Entity	Description
place/possession/enggb	UK crown dependencies.
place/possession_uppercase/enggb	UK crown dependencies in uppercase.
place/country/enggb	UK countries.
place/country_uppercase/enggb	UK countries in uppercase.
place/country_capital/enggb	UK country capitals.
place/country_capital_uppercase/enggb	UK country capitals in uppercase.
place/county/england/enggb	Counties in England.
place/county_uppercase/england/enggb	Counties in England, in uppercase.
place/county/northern_ireland/enggb	Counties in Northern Ireland.
place/county_uppercase/northern_ ireland/enggb	Counties in Northern Ireland, in uppercase.
place/county/scotland/enggb	Counties in Scotland.
place/county_uppercase/scotland/enggb	Counties in Scotland, in uppercase.
place/county/wales/enggb	Counties in Wales.
place/county_uppercase/wales/enggb	Counties in Wales, in uppercase.
place/county/enggb	Counties in UK.
place/county_uppercase/enggb	Counties in UK in uppercase.
place/city1/enggb	Settlement in the UK with over 100,000 inhabitants.
place/city1_uppercase/enggb	Settlement in the UK with over 100,000 inhabitants, in

place_enggb.ecr, continued

Entity	Description
	uppercase.
place/city/england/enggb	Settlements in each UK country, in normal or uppercase.
place/city_uppercase/england/enggb	
place/city/scotland/enggb	
place/city_uppercase/scotland/enggb	
place/city/wales/enggb	
place/city_uppercase/wales/enggb	
place/city/northern_ireland/enggb	
place/city_uppercase/northern_ ireland/enggb	
place/city/enggb	UK settlements.
place/city_uppercase/enggb	UK settlements in uppercase.
place/londonborough/enggb	London borough.
place/island/enggb	Major islands of the United Kingdom.
place/island_uppercase/enggb	Major islands of the United Kingdom in uppercase.

place_enggr.ecr

Entity	Description
place/city1/enggr	Greek settlement with over 100,000 inhabitants.
place/city1_uppercase/enggr	Greek settlement with over 100,000 inhabitants, in uppercase.
place/city2/enggr	Greek settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/enggr	Greek settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/enggr	Greek region.
place/region_uppercase/enggr	Greek region in uppercase.

place_enggr.ecr, continued

Entity	Description
place/prefecture/enggr	Greek prefecture (obsolete after 2010).
place/prefecture_uppercase/enggr	Greek prefecture in uppercase (obsolete after 2010).
place/municipality/enggr	Greek municipality.
place/municipality_uppercase/enggr	Greek municipality in uppercase.
place/island/enggr	Greek island.
place/island_uppercase/enggr	Greek island in uppercase.

place_enggy.ecr

Entity	Description
place/city1/enggy	Guyanan settlement with over 100,000 inhabitants.
place/city1_uppercase/enggy	Guyanan settlement with over 100,000 inhabitants, in uppercase.
place/city2/enggy	Guyanan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/enggy	Guyanan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/enggy	Guyanan region.
place/region_uppercase/enggy	Guyanan region in uppercase.

place_enghk.ecr

Entity	Description
place/district/enghk	District in Hong Kong.
place/island/enghk	Island in Hong Kong.
place/enghk	Street in Hong Kong.

place_engid.ecr

Entity	Description
place/city1/engid	Indonesian settlement with over 100,000 inhabitants.
place/city1_uppercase/engid	Indonesian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engid	Indonesian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engid	Indonesian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engid	Indonesian province.
place/province_uppercase/engid	Indonesian province in uppercase.
place/regency/engid	Indonesian regency.
place/regency_uppercase/engid	Indonesian regency in uppercase.
place/island/engid	Indonesian island.
place/island_uppercase/engid	Indonesian island in uppercase.

place_engie.ecr

Entity	Description
place/city1/engie	Irish settlement with over 100,000 inhabitants.
place/city1_uppercase/engie	Irish settlement with over 100,000 inhabitants, in uppercase.
place/city2/engie	Irish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engie	Irish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county_engie	Irish county.
place/county_uppercase/engie	Irish county in uppercase.
place/island/engie	Irish island.
place/island_uppercase/engie	Irish island in uppercase.

place_engin.ecr

Entity	Description
place/city1/engin	Indian settlement with over 100,000 inhabitants.
place/city1_uppercase/engin	Indian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engin	Indian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engin	Indian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/state/engin	Indian state.
place/state_uppercase/engin	Indian state in uppercase.
place/union_territory/engin	Indian union territory.
place/union_territory_uppercase/engin	Indian union territory in uppercase.
place/district/engin	Indian district.
place/district_uppercase/engin	Indian district in uppercase.
place/island/engin	Indian island.
place/island_uppercase/engin	Indian island in uppercase.

place_engir.ecr

Entity	Description
place/city1/engir	Iranian settlement with over 100,000 inhabitants.
place/city1_uppercase/engir	Iranian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engir	Iranian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engir	Iranian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engir	Iranian province.
place/province_uppercase/engir	Iranian province in uppercase.
place/county/engir	Iranian county.
place/county_uppercase/engir	Iranian county in uppercase.

place_engjp.ecr

Entity	Description
place_city1/engjp	Japanese settlement with over 100,000 inhabitants, in English.
place_city1_uppercase/engjp	Japanese settlement with over 100,000 inhabitants, in uppercase English.
place/city2/engjp	Japanese settlement with between 10,000 and 100,000 inhabitants, in English.
place/city2_uppercase/engjp	Japanese settlement with between 10,000 and 100,000 inhabitants, in uppercase English.
place/special_ward/engjp	Special ward of Tokyo in English.
place/special_ward_uppercase/engjp	Special ward of Tokyo in uppercase English.
place/island/engjp	Japanese island in English.
place/island_uppercase/engjp	Japanese island in uppercase English.
place/prefecture/engjp	Japanese prefectures in English.
place/prefecture_uppercase/engjp	Japanese prefectures in uppercase English.
place/region/engjp	Japanese regions in English.
place/region_uppercase/engjp	Japanese regions in uppercase English.
place/city/aichi/engjp place/city_uppercase/aichi/engjp place/city/akita/engjp place/city_uppercase/akita/engjp place/city/aomori/engjp place/city_uppercase/aomori/engjp place/city/chiba/engjp place/city_uppercase/chiba/engjp place/city/ehime/engjp place/city_uppercase/ehime/engjp place/city/fukui/engjp place/city_uppercase/fukui/engjp place/city/fukuoka/engjp place/city_uppercase/fukuoka/engjp	Cities in each Japanese prefecture in English, in normal or uppercase.

place_engjp.ecr, continued

Entity	Description
place/city/fukushima/engjp	
place/city_uppercase/fukushima/engjp	
place/city/gifu/engjp	
place/city_uppercase/gifu/engjp	
place/city/gunma/engjp	
place/city_uppercase/gunma/engjp	
place/city/hiroshima/engjp	
place/city_uppercase/hiroshima/engjp	
place/city/hokkaido/engjp	
place/city_uppercase/hokkaido/engjp	
place/city/hyogo/engjp	
place/city_uppercase/hyogo/engjp	
place/city/ibaraki/engjp	
place/city_uppercase/ibaraki/engjp	
place/city/ishikawa/engjp	
place/city_uppercase/ishikawa/engjp	
place/city/iwate/engjp	
place/city_uppercase/iwate/engjp	
place/city/kagawa/engjp	
place/city_uppercase/kagawa/engjp	
place/city/kagoshima/engjp	
place/city_uppercase/kagoshima/engjp	
place/city/kanagawa/engjp	
place/city_uppercase/kanagawa/engjp	
place/city/kochi/engjp	
place/city_uppercase/kochi/engjp	
place/city/kumamoto/engjp	
place/city_uppercase/kumamoto/engjp	
place/city/kyoto/engjp	
place/city_uppercase/kyoto/engjp	

place_engjp.ecr, continued

Entity	Description
place/city/mie/engjp	
place/city_uppercase/mie/engjp	
place/city/miyagi/engjp	
place/city_uppercase/miyagi/engjp	
place/city/miyazaki/engjp	
place/city_uppercase/miyazaki/engjp	
place/city/nagano/engjp	
place/city_uppercase/nagano/engjp	
place/city/nagasaki/engjp	
place/city_uppercase/nagasaki/engjp	
place/city/nara/engjp	
place/city_uppercase/nara/engjp	
place/city/niigata/engjp	
place/city_uppercase/niigata/engjp	
place/city/oita/engjp	
place/city_uppercase/oita/engjp	
place/city/okayama/engjp	
place/city_uppercase/okayama/engjp	
place/city/okinawa/engjp	
place/city_uppercase/okinawa/engjp	
place/city/osaka/engjp	
place/city_uppercase/osaka/engjp	
place/city/saga/engjp	
place/city_uppercase/saga/engjp	
place/city/saitama/engjp	
place/city_uppercase/saitama/engjp	
place/city/shiga/engjp	
place/city_uppercase/shiga/engjp	
place/city/shimane/engjp	
place/city_uppercase/shimane/engjp	

place_engjp.ecr, continued

Entity	Description
place/city/shizuoka/engjp place/city_uppercase/shizuoka/engjp place/city/tochigi/engjp place/city_uppercase/tochigi/engjp place/city/tokushima/engjp place/city_uppercase/tokushima/engjp place/city/tokyo/engjp place/city_uppercase/tokyo/engjp place/city/tottori/engjp place/city_uppercase/tottori/engjp place/city/toyama/engjp place/city_uppercase/toyama/engjp place/city/wakayama/engjp place/city_uppercase/wakayama/engjp place/city/yamagata/engjp place/city_uppercase/yamagata/engjp place/city/yamaguchi/engjp place/city_uppercase/yamaguchi/engjp place/city/yamanashi/engjp place/city_uppercase/yamanashi/engjp	
place/city/engjp	Japanese cities in English.

place_engkr.ecr

Entity	Description
place/city1_rr/engkr	South Korean settlement with over 100,000 inhabitants, in revised romanization.
place/city1_rr_uppercase/engkr	South Korean settlement with over 100,000 inhabitants, in uppercase revised romanization.
place/city2_rr/engkr	South Korean settlement with between 10,000 and 100,000 inhabitants, in revised romanization.

place_engkr.ecr, continued

Entity	Description
place/city2_rr_uppercase/engkr	South Korean settlement with between 10,000 and 100,000 inhabitants, in uppercase revised romanization.
place/province_rr/engkr	South Korean province in revised romanization.
place/province_rr_uppercase/engkr	South Korean province in uppercase revised romanization.
place/county_rr/engkr	South Korean county in revised romanization.
place/county_rr_uppercase/engkr	South Korean county in uppercase revised romanization.
place/island_rr/engkr	South Korean island in revised romanization.
place/island_rr_uppercase/engkr	South Korean island in uppercase revised romanization.
place/city1_mcr/engkr	South Korean settlement with over 100,000 inhabitants in McCune-Reischauer romanization.
place/city1_mcr_uppercase/engkr	South Korean settlement with over 100,000 inhabitants, in uppercase McCune-Reischauer romanization.
place/city2_mcr/engkr	South Korean settlement with between 10,000 and 100,000 inhabitants, in McCune-Reischauer romanization.
place/city2_mcr_uppercase/engkr	South Korean settlement with between 10,000 and 100,000 inhabitants, in uppercase McCune-Reischauer romanization.
place/province_mcr/engkr	South Korean province in McCune-Reischauer romanization.
place/province_mcr_uppercase/engkr	South Korean province in uppercase McCune-Reischauer romanization.
place/county_mcr/engkr	South Korean county in McCune-Reischauer romanization.
place/county_mcr_uppercase/engkr	South Korean county in uppercase McCune-Reischauer romanization.
place/island_mcr/engkr	South Korean island in McCune-Reischauer romanization.

place_engkr.ecr, continued

Entity	Description
place/island_mcr_uppercase/engkr	South Korean island in uppercase McCune-Reischauer romanization.

place_englk.ecr

Entity	Description
place/city1/englk	Sri Lankan settlement with over 100,000 inhabitants.
place/city1_uppercase/englk	Sri Lankan settlement with over 100,000 inhabitants, in uppercase.
place/city2/englk	Sri Lankan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/englk	Sri Lankan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/englk	Sri Lankan province.
place/province_uppercase/englk	Sri Lankan province in uppercase.
place/district/englk	Sri Lankan district.
place/district_uppercase/englk	Sri Lankan district in uppercase.

place_engmk.ecr

Entity	Description
place/city1/engmk	Macedonian settlement with over 100,000 inhabitants.
place/city1_uppercase/engmk	Macedonian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engmk	Macedonian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engmk	Macedonian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/municipality/engmk	Macedonian municipality.
place/municipality_uppercase/engmk	Macedonian municipality in uppercase.

place_engmn.ecr

Entity	Description
place/city1/engmn	Mongolian settlement with over 100,000 inhabitants.
place/city1_uppercase/engmn	Mongolian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engmn	Mongolian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engmn	Mongolian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engmn	Mongolian province.
place/province_uppercase/engmn	Mongolian province in uppercase.

place_engmy.ecr

Entity	Description
place/city1/engmy	Malaysian settlement with over 100,000 inhabitants.
place/city1_uppercase/engmy	Malaysian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engmy	Malaysian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engmy	Malaysian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/state/engmy	Malaysian state.
place/state_uppercase/engmy	Malaysian state in uppercase.
place/district/engmy	Malaysian district.
place/district_uppercase/engmy	Malaysian district in uppercase.

place_engnz.ecr

Entity	Description
place/city1/engnz	New Zealand settlement with over 100,000 inhabitants.
place/city1_uppercase/engnz	New Zealand settlement with over 100,000 inhabitants, in uppercase.

place_engnz.ecr, continued

Entity	Description
place/city2/engnz	New Zealand settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engnz	New Zealand settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/engnz	New Zealand region.
place/region_uppercase/engnz	New Zealand region in uppercase.
place/terr_auth/engnz	New Zealand territorial authority.
place/terr_auth_uppercase/engnz	New Zealand territorial authority in uppercase.
place/island/engnz	New Zealand island.
place/island_uppercase/engnz	New Zealand island in uppercase.

place_engph.ecr

Entity	Description
place/city1/engph	Philippine settlement with over 100,000 inhabitants.
place/city1_uppercase/engph	Philippine settlement with over 100,000 inhabitants, in uppercase.
place/city2/engph	Philippine settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engph	Philippine settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/engph	Philippine region.
place/region_uppercase/engph	Philippine region in uppercase.
place/province/engph	Philippine province.
place/province_uppercase/engph	Philippine province in uppercase.
place/island/engph	Philippine island.
place/island_uppercase/engph	Philippine island in uppercase.

place_engpk.ecr

Entity	Description
place/city1/engpk	Pakistani settlement with over 100,000 inhabitants.
place/city1_uppercase/engpk	Pakistani settlement with over 100,000 inhabitants, in uppercase.
place/city2/engpk	Pakistani settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engpk	Pakistani settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engpk	Pakistani province.
place/province_uppercase/engpk	Pakistani province in uppercase.
place/district/engpk	Pakistani district.
place/district_uppercase/engpk	Pakistani district in uppercase.

place_engqa.ecr

Entity	Description
place/city1/engqa	Qatari settlement with over 100,000 inhabitants.
place/city1_uppercase/engqa	Qatari settlement with over 100,000 inhabitants, in uppercase.
place/city2/engqa	Qatari settlement with fewer than 100,000 inhabitants.
place/city2_uppercase/engqa	Qatari settlement with fewer than 100,000 inhabitants, in uppercase.
place/municipality/engqa	Qatari municipality.
place/municipality_uppercase/engqa	Qatari municipality in uppercase.

place_engru.ecr

Entity	Description
place/city1/engru	Russian settlement with over 100,000 inhabitants.
place/city1_uppercase/engru	Russian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engru	Russian settlement with between 10,000 and 100,000

place_engru.ecr, continued

Entity	Description
	inhabitants.
place/city2_uppercase/engru	Russian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/republic/engru	Russian republic (type of region).
place/republic_uppercase/engru	Russian republic (type of region), in uppercase.
place/oblast/engru	Russian oblast (type of region).
place/oblast_uppercase/engru	Russian oblast (type of region), in uppercase.
place/krai/engru	Russian krai (type of region).
place/krai_uppercase/engru	Russian krai (type of region), in uppercase.
place/okrug/engru	Russian okrug (type of region).
place/okrug_uppercase/engru	Russian okrug (type of region), in uppercase.
place/federal_city/engru	Russian federal city (type of region).
place/federal_city_uppercase/engru	Russian federal city (type of region), in uppercase.
place/region/engru	Russian region.
place/region_uppercase/engru	Russian region, in uppercase.
place/island/engru	Russian island.
place/island_uppercase/engru	Russian island, in uppercase.

place_engsa.ecr

Entity	Description
place/city1/engsa	Saudi Arabian settlement with over 100,000 inhabitants.
place/city1_uppercase/engsa	Saudi Arabian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engsa	Saudi Arabian settlement with fewer than 100,000 inhabitants.
place/city2_uppercase/engsa	Saudi Arabian settlement with fewer than 100,000 inhabitants, in uppercase.

place_engsa.ecr, continued

Entity	Description
place/province/engsa	Saudi Arabian province.
place/province_uppercase/engsa	Saudi Arabian province in uppercase.

place_ength.ecr

Entity	Description
place/city1/ength	Thai settlement with over 100,000 inhabitants.
place/city1_uppercase/ength	Thai settlement with over 100,000 inhabitants, in uppercase.
place/city2/ength	Thai settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/ength	Thai settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/ength	Thai province.
place/province_uppercase/ength	Thai province in uppercase.

place_engtw.ecr

Entity	Description
place/city1/engtw	Taiwanese settlement with over 100,000 inhabitants.
place/city1_uppercase/engtw	Taiwanese settlement with over 100,000 inhabitants, in uppercase.
place/city2/engtw	Taiwanese settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engtw	Taiwanese settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/engtw	Taiwanese county.
place/county_uppercase/engtw	Taiwanese county in uppercase.

place_engua.ecr

Entity	Description
place/city1/engua	Ukrainian settlement with over 100,000 inhabitants.
place/city1_uppercase/engua	Ukrainian settlement with over 100,000 inhabitants, in uppercase.
place/city2/engua	Ukrainian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engua	Ukrainian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/engua	Ukrainian region.
place/region_uppercase/engua	Ukrainian region in uppercase.

place_engus.ecr

Entity	Description
place/possession/engus	U.S. possessions in long form. For example, *American Samoa*.
place/possession_uppercase/engus	U.S. possessions in long form, in uppercase.
place/possession_abbrev/engus	U.S. possession abbreviations. For example, *GU*.
place/state/engus	U.S. states. For example, *New Hampshire*.
place/state_uppercase/engus	U.S. states, in uppercase.
place/state_abbrev/engus	U.S. states abbreviations. For example, *AL*.
place/poss_state/engus	U.S. possessions and states.
place/poss_state_abbrev/engus	U.S. possession and state abbreviations.
place/statecapital/engus	U.S. state capitals.
place/statecapital_uppercase/engus	U.S. state capitals, in uppercase.
place/city/AL/engus place/city_uppercase/AL/engus place/city/AK/engus place/city_uppercase/AK/engus place/city/AZ/engus place/city_uppercase/AZ/engus	Settlements in each U.S. state, in normal or uppercase.

place_engus.ecr, continued

Entity	Description
place/city/AR/engus	
place/city_uppercase/AR/engus	
place/city/CA/engus	
place/city_uppercase/CA/engus	
place/city/CO/engus	
place/city_uppercase/CO/engus	
place/city/CT/engus	
place/city_uppercase/CT/engus	
place/city/DE/engus	
place/city_uppercase/DE/engus	
place/city/FL/engus	
place/city_uppercase/FL/engus	
place/city/GA/engus	
place/city_uppercase/GA/engus	
place/city/HI/engus	
place/city_uppercase/HI/engus	
place/city/ID/engus	
place/city_uppercase/ID/engus	
place/city/IL/engus	
place/city_uppercase/IL/engus	
place/city/IN/engus	
place/city_uppercase/IN/engus	
place/city/IA/engus	
place/city_uppercase/IA/engus	
place/city/KS/engus	
place/city_uppercase/KS/engus	
place/city/KY/engus	
place/city_uppercase/KY/engus	
place/city/LA/engus	
place/city_uppercase/LA/engus	

place_engus.ecr, continued

Entity	Description
place/city/ME/engus	
place/city_uppercase/ME/engus	
place/city/MD/engus	
place/city_uppercase/MD/engus	
place/city/MA/engus	
place/city_uppercase/MA/engus	
place/city/MI/engus	
place/city_uppercase/MI/engus	
place/city/MN/engus	
place/city_uppercase/MN/engus	
place/city/MS/engus	
place/city_uppercase/MS/engus	
place/city/MO/engus	
place/city_uppercase/MO/engus	
place/city/MT/engus	
place/city_uppercase/MT/engus	
place/city/NE/engus	
place/city_uppercase/NE/engus	
place/city/NV/engus	
place/city_uppercase/NV/engus	
place/city/NH/engus	
place/city_uppercase/NH/engus	
place/city/NJ/engus	
place/city_uppercase/NJ/engus	
place/city/NM/engus	
place/city_uppercase/NM/engus	
place/city/NY/engus	
place/city_uppercase/NY/engus	
place/city/NC/engus	
place/city_uppercase/NC/engus	

place_engus.ecr, continued

Entity	Description
place/city/ND/engus	
place/city_uppercase/ND/engus	
place/city/OH/engus	
place/city_uppercase/OH/engus	
place/city/OK/engus	
place/city_uppercase/OK/engus	
place/city/OR/engus	
place/city_uppercase/OR/engus	
place/city/PA/engus	
place/city_uppercase/PA/engus	
place/city/RI/engus	
place/city_uppercase/RI/engus	
place/city/SC/engus	
place/city_uppercase/SC/engus	
place/city/SD/engus	
place/city_uppercase/SD/engus	
place/city/TN/engus	
place/city_uppercase/TN/engus	
place/city/TX/engus	
place/city_uppercase/TX/engus	
place/city/UT/engus	
place/city_uppercase/UT/engus	
place/city/VA/engus	
place/city_uppercase/VA/engus	
place/city/VT/engus	
place/city_uppercase/VT/engus	
place/city/WA/engus	
place/city_uppercase/WA/engus	
place/city/WI/engus	
place/city_uppercase/WI/engus	

place_engus.ecr, continued

Entity	Description
place/city/WV/engus place/city_uppercase/WV/engus place/city/WY/engus place/city_uppercase/WY/engus	
place/city1/engus	U.S. city with over 100,000 inhabitants.
place/city1_uppercase/engus	U.S. city with over 100,000 inhabitants, in uppercase.
place/city2/engus	U.S. city with over 10,000 inhabitants.
place/city2_uppercase/engus	U.S. city with over 10,000 inhabitants, in uppercase.
place/county/AL/engus	County in Alabama.
place/county/AK/engus	County in Alaska.
place/county/AZ/engus	County in Arizona.
place/county/AR/engus	County in Arkansas.
place/county/CA/engus	County in California.
place/county/CO/engus	County in Colorado.
place/county/CT/engus	County in Connecticut.
place/county/DE/engus	County in Delaware.
place/county/FL/engus	County in Florida.
place/county/GA/engus	County in Georgia.
place/county/HI/engus	County in Hawaii.
place/county/ID/engus	County in Idaho.
place/county/IL/engus	County in Illinois.
place/county/IN/engus	County in Indiana.
place/county/IA/engus	County in Iowa.
place/county/KS/engus	County in Kansas.
place/county/KY/engus	County in Kentucky.
place/county/LA/engus	County in Louisiana.
place/county/ME/engus	County in Maine.
place/county/MD/engus	County in Maryland.
place/county/MA/engus	County in Massachusetts.

place_engus.ecr, continued

Entity	Description
place/county/MI/engus	County in Michigan.
place/county/MN/engus	County in Minnesota.
place/county/MS/engus	County in Mississippi.
place/county/MO/engus	County in Missouri.
place/county/MT/engus	County in Montana.
place/county/NE/engus	County in Nebraska.
place/county/NV/engus	County in Nevada.
place/county/NH/engus	County in New Hampshire.
place/county/NJ/engus	County in New Jersey.
place/county/NM/engus	County in New Mexico.
place/county/NY/engus	County in New York.
place/county/NC/engus	County in North Carolina.
place/county/ND/engus	County in North Dakota.
place/county/OH/engus	County in Ohio.
place/county/OK/engus	County in Oklahoma.
place/county/OR/engus	County in Oregon.
place/county/PA/engus	County in Pennsylvania.
place/county/RI/engus	County in Rhode Island.
place/county/SC/engus	County in South Carolina.
place/county/SD/engus	County in South Dakota.
place/county/TN/engus	County in Tennessee.
place/county/TX/engus	County in Texas.
place/county/UT/engus	County in Utah.
place/county/VT/engus	County in Vermont.
place/county/VA/engus	County in Virginia.
place/county/WA/engus	County in Washington.
place/county/WV/engus	County in West Virginia.
place/county/WI/engus	County in Wisconsin.
place/county/WY/engus	County in Wyoming.
place/county_uppercase/AL/engus	County in Alabama in uppercase.

place_engus.ecr, continued

Entity	Description
place/county_uppercase/AK/engus	County in Alaska in uppercase.
place/county_uppercase/AZ/engus	County in Arizona in uppercase.
place/county_uppercase/AR/engus	County in Arkansas in uppercase.
place/county_uppercase/CA/engus	County in California in uppercase.
place/county_uppercase/CO/engus	County in Colorado in uppercase.
place/county_uppercase/CT/engus	County in Connecticut in uppercase.
place/county_uppercase/DE/engus	County in Delaware in uppercase.
place/county_uppercase/FL/engus	County in Florida in uppercase.
place/county_uppercase/GA/engus	County in Georgia in uppercase.
place/county_uppercase/HI/engus	County in Hawaii in uppercase.
place/county_uppercase/ID/engus	County in Idaho in uppercase.
place/county_uppercase/IL/engus	County in Illinois in uppercase.
place/county_uppercase/IN/engus	County in Indiana in uppercase.
place/county_uppercase/IA/engus	County in Iowa in uppercase.
place/county_uppercase/KS/engus	County in Kansas in uppercase.
place/county_uppercase/KY/engus	County in Kentucky in uppercase.
place/county_uppercase/LA/engus	County in Louisiana in uppercase.
place/county_uppercase/ME/engus	County in Maine in uppercase.
place/county_uppercase/MD/engus	County in Maryland in uppercase.
place/county_uppercase/MA/engus	County in Massachusetts in uppercase.
place/county_uppercase/MI/engus	County in Michigan in uppercase.
place/county_uppercase/MN/engus	County in Minnesota in uppercase.
place/county_uppercase/MS/engus	County in Mississippi in uppercase.
place/county_uppercase/MO/engus	County in Missouri in uppercase.
place/county_uppercase/MT/engus	County in Montana in uppercase.
place/county_uppercase/NE/engus	County in Nebraska in uppercase.
place/county_uppercase/NV/engus	County in Nevada in uppercase.
place/county_uppercase/NH/engus	County in New Hampshire in uppercase.
place/county_uppercase/NJ/engus	County in New Jersey in uppercase.
place/county_uppercase/NM/engus	County in New Mexico in uppercase.

place_engus.ecr, continued

Entity	Description
place/county_uppercase/NY/engus	County in New York in uppercase.
place/county_uppercase/NC/engus	County in North Carolina in uppercase.
place/county_uppercase/ND/engus	County in North Dakota in uppercase.
place/county_uppercase/OH/engus	County in Ohio in uppercase.
place/county_uppercase/OK/engus	County in Oklahoma in uppercase.
place/county_uppercase/OR/engus	County in Oregon in uppercase.
place/county_uppercase/PA/engus	County in Pennsylvania in uppercase.
place/county_uppercase/RI/engus	County in Rhode Island in uppercase.
place/county_uppercase/SC/engus	County in South Carolina in uppercase.
place/county_uppercase/SD/engus	County in South Dakota in uppercase.
place/county_uppercase/TN/engus	County in Tennessee in uppercase.
place/county_uppercase/TX/engus	County in Texas in uppercase.
place/county_uppercase/UT/engus	County in Utah in uppercase.
place/county_uppercase/VT/engus	County in Vermont in uppercase.
place/county_uppercase/VA/engus	County in Virginia in uppercase.
place/county_uppercase/WA/engus	County in Washington in uppercase.
place/county_uppercase/WV/engus	County in West Virginia in uppercase.
place/county_uppercase/WI/engus	County in Wisconsin in uppercase.
place/county_uppercase/WY/engus	County in Wyoming in uppercase.
place/county/engus	Any U.S. county.
place/county_uppercase/engus	Any U.S. county in uppercase.

place_engvn.ecr

Entity	Description
place/city1/engvn	Vietnamese settlement with over 100,000 inhabitants.
place/city1_uppercase/engvn	Vietnamese settlement with over 100,000 inhabitants, in uppercase.
place/city2/engvn	Vietnamese settlement with between 10,000 and 100,000 inhabitants.

place_engvn.ecr, continued

Entity	Description
place/city2_uppercase/engvn	Vietnamese settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engvn	Vietnamese province.
place/province_uppercase/engvn	Vietnamese province in uppercase.
place/district/engvn	Vietnamese district.
place/district_uppercase/engvn	Vietnamese district in uppercase.

place_engza.ecr

Entity	Description
place/city1/engza	South African settlement with over 100,000 inhabitants.
place/city1_uppercase/engza	South African settlement with over 100,000 inhabitants, in uppercase.
place/city2/engza	South African settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/engza	South African settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/engza	South African province.
place/province_uppercase/engza	South African province in uppercase.
place/district/engza	South African district.
place/district_uppercase/engza	South African district in uppercase.
place/island/engza	South African island.
place/island_uppercase/engza	South African island in uppercase.

place_estee.ecr

Entity	Description
place/city1/estee	Estonian settlement with over 100,000 inhabitants.
place/city1_uppercase/estee	Estonian settlement with over 100,000 inhabitants, in uppercase.

place_estee.ecr, continued

Entity	Description
place/city2/estee	Estonian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/estee	Estonian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/estee	Estonian county.
place/county_uppercase/estee	Estonian county in uppercase.

place_finfi.ecr

Entity	Description
place/city1/finfi	Finnish settlement with over 100,000 inhabitants.
place/city1_uppercase/finfi	Finnish settlement with over 100,000 inhabitants, in uppercase.
place/city2/finfi	Finnish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/finfi	Finnish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/finfi	Finnish region.
place/region_uppercase/finfi	Finnish region in uppercase.
place/island/finfi	Finnish island.
place/island_uppercase/finfi	Finnish island in uppercase.

place_frefr.ecr

Entity	Description
place/city1/frefr	French settlement with over 100,000 inhabitants.
place/city1_uppercase/frefr	French settlement with over 100,000 inhabitants, in uppercase.
place/region_metro/frefr	French metropolitan regions.
place/region_metro_uppercase/frefr	French metropolitan regions in uppercase.

place_frefr.ecr, continued

Entity	Description
place/department_metro/Alsace/frefr	Departments of each French metropolitan region, in normal or uppercase.
place/department_metro_uppercase/Alsace/frefr	
place/department_metro/Aquitaine/frefr	
place/department_metro_uppercase/Aquitaine/frefr	
place/department_metro/Auvergne/frefr	
place/department_metro_uppercase/Auvergne/frefr	
place/department_metro/BasseNormandie/frefr	
place/department_metro_ uppercase/BasseNormandie/frefr	
place/department_metro/Bourgogne/frefr	
place/department_metro_uppercase/Bourgogne/frefr	
place/department_metro/Brittany/frefr	
place/department_metro_uppercase/Brittany/frefr	
place/department_metro/Centre/frefr	
place/department_metro_uppercase/Centre/frefr	
place/department_metro/ChampagneArdenne/frefr	
place/department_metro_ uppercase/ChampagneArdenne/frefr	
place/department_metro/Corsica/frefr	
place/department_metro_uppercase/Corsica/frefr	
place/department_metro/FrancheComte/frefr	
place/department_metro_ uppercase/FrancheComte/frefr	
place/department_metro/HauteNormandie/frefr	
place/department_metro_ uppercase/HauteNormandie/frefr	
place/department_metro/IleDeFrance/frefr	
place/department_metro_	

place_frefr.ecr, continued

Entity	Description
uppercase/IleDeFrance/frefr	
place/department_metro/LanguedocRoussillon/frefr	
place/department_metro_ uppercase/LanguedocRoussillon/frefr	
place/department_metro/Limousin/frefr	
place/department_metro_uppercase/Limousin/frefr	
place/department_metro/Lorraine/frefr	
place/department_metro_uppercase/Lorraine/frefr	
place/department_metro/MidiPyrenees/frefr	
place/department_metro_ uppercase/MidiPyrenees/frefr	
place/department_metro/NordPasDeCalais/frefr	
place/department_metro_ uppercase/NordPasDeCalais/frefr	
place/department_metro/PaysDeLaLoire/frefr	
place/department_metro_ uppercase/PaysDeLaLoire/frefr	
place/department_metro/Picardie/frefr	
place/department_metro_uppercase/Picardie/frefr	
place/department_metro/PoitouCharentes/frefr	
place/department_metro_ uppercase/PoitouCharentes/frefr	
place/department_ metro/ProvenceAlpesCoteDAzur/frefr	
place/department_metro_ uppercase/ProvenceAlpesCoteDAzur/frefr	
place/department_metro/RhoneAlpes/frefr	
place/department_metro_ uppercase/RhoneAlpes/frefr	

place_frefr.ecr, continued

Entity	Description
place/department_metro/frefr	French metropolitan departments.
place/department_metro_uppercase/frefr	French metropolitan departments in uppercase.
place/departmentcode_metro/frefr	French metropolitan department INSEE codes.
place/departmentcode_overseas/frefr	French overseas department INSEE codes.
place/communecode_metro/frefr	French metropolitan commune INSEE codes.
place/communecode_overseas/frefr	French overseas commune INSEE codes.
place/city/alsace/Bas_Rhin/frefr	Settlements in each French department, in normal or uppercase.
place/city_uppercase/alsace/Bas_Rhin/frefr	
place/city/alsace/Haut_Rhin/frefr	
place/city_uppercase/alsace/Haut_Rhin/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/aquitaine/Dordogne/frefr	
place/city_uppercase/aquitaine/Dordogne/frefr	
place/city/aquitaine/Gironde/frefr	
place/city_uppercase/aquitaine/Gironde/frefr	
place/city/aquitaine/Landes/frefr	
place/city_uppercase/aquitaine/Landes/frefr	
place/city/aquitaine/Lot_et_Garonne/frefr	
place/city_uppercase/aquitaine/Lot_et_ Garonne/frefr	
place/city/aquitaine/Pyrenees_Atlantiques/frefr	
place/city_uppercase/aquitaine/Pyrenees_ Atlantiques/frefr	
place/city/auvergne/Allier/frefr	
place/city_uppercase/auvergne/Allier/frefr	
place/city/auvergne/Cantal/frefr	
place/city_uppercase/auvergne/Cantal/frefr	
place/city/auvergne/Haute_Loire/frefr	
place/city_uppercase/auvergne/Haute_Loire/frefr	
place/city/auvergne/Puy_de_Dome/frefr	
place/city_uppercase/auvergne/Puy_de_Dome/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/basseNormandie/Calvados/frefr	
place/city_ uppercase/basseNormandie/Calvados/frefr	
place/city/basseNormandie/Manche/frefr	
place/city_ uppercase/basseNormandie/Manche/frefr	
place/city/basseNormandie/Orne/frefr	
place/city_uppercase/basseNormandie/Orne/frefr	
place/city/bourgogne/Cote_dOr/frefr	
place/city_uppercase/bourgogne/Cote_dOr/frefr	
place/city/bourgogne/Nievre/frefr	
place/city_uppercase/bourgogne/Nievre/frefr	
place/city/bourgogne/Saone_et_Loire/frefr	
place/city_uppercase/bourgogne/Saone_et_ Loire/frefr	
place/city/bourgogne/Yonne/frefr	
place/city_uppercase/bourgogne/Yonne/frefr	
place/city/brittany/Cotes_dArmor/frefr	
place/city_uppercase/brittany/Cotes_dArmor/frefr	
place/city/brittany/Finistere/frefr	
place/city_uppercase/brittany/Finistere/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/brittany/Ille_et_Vilaine/frefr	
place/city_uppercase/brittany/Ille_et_Vilaine/frefr	
place/city/brittany/Morbihan/frefr	
place/city_uppercase/brittany/Morbihan/frefr	
place/city/centre/Cher/frefr	
place/city_uppercase/centre/Cher/frefr	
place/city/centre/Eure_et_Loir/frefr	
place/city_uppercase/centre/Eure_et_Loir/frefr	
place/city/centre/Indre/frefr	
place/city_uppercase/centre/Indre/frefr	
place/city/centre/Indre_et_Loire/frefr	
place/city_uppercase/centre/Indre_et_Loire/frefr	
place/city/centre/Loir_et_Cher/frefr	
place/city_uppercase/centre/Loir_et_Cher/frefr	
place/city/centre/Loiret/frefr	
place/city_uppercase/centre/Loiret/frefr	
place/city/champagneArdenne/Ardennes/frefr	
place/city_ uppercase/champagneArdenne/Ardennes/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/champagneArdenne/Aube/frefr	
place/city_ uppercase/champagneArdenne/Aube/frefr	
place/city/champagneArdenne/Marne/frefr	
place/city_ uppercase/champagneArdenne/Marne/frefr	
place/city/champagneArdenne/Haute_Marne/frefr	
place/city_uppercase/champagneArdenne/Haute_ Marne/frefr	
place/city/corsica/Corse_du_Sud/frefr	
place/city_uppercase/corsica/Corse_du_Sud/frefr	
place/city/corsica/Haute_Corse/frefr	
place/city_uppercase/corsica/Haute_Corse/frefr	
place/city/francheComte/Doubs/frefr	
place/city_uppercase/francheComte/Doubs/frefr	
place/city/francheComte/Jura/frefr	
place/city_uppercase/francheComte/Jura/frefr	
place/city/francheComte/Haute_Saone/frefr	
place/city_uppercase/francheComte/Haute_ Saone/frefr	
place/city/francheComte/Territoire_de_Belfort/frefr	
place/city_uppercase/francheComte/Territoire_de_ Belfort/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/hauteNormandie/Eure/frefr	
place/city_uppercase/hauteNormandie/Eure/frefr	
place/city/hauteNormandie/Seine_Maritime/frefr	
place/city_uppercase/hauteNormandie/Seine_Maritime/frefr	
place/city/ileDeFrance/Seine_et_Marne/frefr	
place/city_uppercase/ileDeFrance/Seine_et_Marne/frefr	
place/city/ileDeFrance/Yvelines/frefr	
place/city_uppercase/ileDeFrance/Yvelines/frefr	
place/city/ileDeFrance/Essonne/frefr	
place/city_uppercase/ileDeFrance/Essonne/frefr	
place/city/ileDeFrance/Hauts_de_Seine/frefr	
place/city_uppercase/ileDeFrance/Hauts_de_Seine/frefr	
place/city/ileDeFrance/Seine_Saint_Denis/frefr	
place/city_uppercase/ileDeFrance/Seine_Saint_Denis/frefr	
place/city/ileDeFrance/Val_de_Marne/frefr	
place/city_uppercase/ileDeFrance/Val_de_Marne/frefr	
place/city/ileDeFrance/Val_dOise/frefr	
place/city_uppercase/ileDeFrance/Val_dOise/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/ileDeFrance/Paris/frefr	
place/city_uppercase/ileDeFrance/Paris/frefr	
place/city/languedocRoussillon/Aude/frefr	
place/city_ uppercase/languedocRoussillon/Aude/frefr	
place/city/languedocRoussillon/Gard/frefr	
place/city_ uppercase/languedocRoussillon/Gard/frefr	
place/city/languedocRoussillon/Herault/frefr	
place/city_ uppercase/languedocRoussillon/Herault/frefr	
place/city/languedocRoussillon/Lozere/frefr	
place/city_ uppercase/languedocRoussillon/Lozere/frefr	
place/city/languedocRoussillon/Pyrenees_ Orientales/frefr	
place/city_ uppercase/languedocRoussillon/Pyrenees_ Orientales/frefr	
place/city/limousin/Correze/frefr	
place/city_uppercase/limousin/Correze/frefr	
place/city/limousin/Creuse/frefr	
place/city_uppercase/limousin/Creuse/frefr	
place/city/limousin/Haute_Vienne/frefr	
place/city_uppercase/limousin/Haute_Vienne/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/lorraine/Meurthe_et_Moselle/frefr	
place/city_uppercase/lorraine/Meurthe_et_ Moselle/frefr	
place/city/lorraine/Meuse/frefr	
place/city_uppercase/lorraine/Meuse/frefr	
place/city/lorraine/Moselle/frefr	
place/city_uppercase/lorraine/Moselle/frefr	
place/city/lorraine/Vosges/frefr	
place/city_uppercase/lorraine/Vosges/frefr	
place/city/midiPyrenees/Ariege/frefr	
place/city_uppercase/midiPyrenees/Ariege/frefr	
place/city/midiPyrenees/Aveyron/frefr	
place/city_uppercase/midiPyrenees/Aveyron/frefr	
place/city/midiPyrenees/Haute_Garonne/frefr	
place/city_uppercase/midiPyrenees/Haute_ Garonne/frefr	
place/city/midiPyrenees/Gers/frefr	
place/city_uppercase/midiPyrenees/Gers/frefr	
place/city/midiPyrenees/Lot/frefr	
place/city_uppercase/midiPyrenees/Lot/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/midiPyrenees/Hautes_Pyrenees/frefr	
place/city_uppercase/midiPyrenees/Hautes_ Pyrenees/frefr	
place/city/midiPyrenees/Tarn/frefr	
place/city_uppercase/midiPyrenees/Tarn/frefr	
place/city/midiPyrenees/Tarn_et_Garonne/frefr	
place/city_uppercase/midiPyrenees/Tarn_et_ Garonne/frefr	
place/city/nordPasDeCalais/Nord/frefr	
place/city_uppercase/nordPasDeCalais/Nord/frefr	
place/city/nordPasDeCalais/Pas_de_Calais/frefr	
place/city_uppercase/nordPasDeCalais/Pas_de_ Calais/frefr	
place/city/paysDeLaLoire/Loire_Atlantique/frefr	
place/city_uppercase/paysDeLaLoire/Loire_ Atlantique/frefr	
place/city/paysDeLaLoire/Maine_et_Loire/frefr	
place/city_uppercase/paysDeLaLoire/Maine_et_ Loire/frefr	
place/city/paysDeLaLoire/Mayenne/frefr	
place/city_uppercase/paysDeLaLoire/Mayenne/frefr	
place/city/paysDeLaLoire/Sarthe/frefr	
place/city_uppercase/paysDeLaLoire/Sarthe/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/paysDeLaLoire/Vendee/frefr	
place/city_uppercase/paysDeLaLoire/Vendee/frefr	
place/city/picardie/Aisne/frefr	
place/city_uppercase/picardie/Aisne/frefr	
place/city/picardie/Oise/frefr	
place/city_uppercase/picardie/Oise/frefr	
place/city_uppercase/picardie/Somme/frefr	
place/city/poitouCharentes/Charente/frefr	
place/city_ uppercase/poitouCharentes/Charente/frefr	
place/city/poitouCharentes/Charente_Maritime/frefr	
place/city_uppercase/poitouCharentes/Charente_ Maritime/frefr	
place/city/poitouCharentes/Deux_Sevres/frefr	
place/city_uppercase/poitouCharentes/Deux_ Sevres/frefr	
place/city/poitouCharentes/Vienne/frefr	
place/city_uppercase/poitouCharentes/Vienne/frefr	
place/city/provenceAlpesCoteDAzur/Alpes_de_ Haute_Provence/frefr	
place/city_ uppercase/provenceAlpesCoteDAzur/Alpes_de_ Haute_Provence/frefr	
place/city/provenceAlpesCoteDAzur/Hautes_ Alpes/frefr	
place/city_ uppercase/provenceAlpesCoteDAzur/Hautes_ Alpes/frefr	
place/city/provenceAlpesCoteDAzur/Alpes_ Maritimes/frefr	

place_frefr.ecr, continued

Entity	Description
place/city_ uppercase/provenceAlpesCoteDAzur/Alpes_ Maritimes/frefr	
place/city/provenceAlpesCoteDAzur/Bouches_du_ Rhone/frefr	
place/city_ uppercase/provenceAlpesCoteDAzur/Bouches_du_ Rhone/frefr	
place/city/provenceAlpesCoteDAzur/Var/frefr	
place/city_ uppercase/provenceAlpesCoteDAzur/Var/frefr	
place/city/provenceAlpesCoteDAzur/Vaucluse/frefr	
place/city_ uppercase/provenceAlpesCoteDAzur/Vaucluse/frefr	
place/city/rhoneAlpes/Ain/frefr	
place/city_uppercase/rhoneAlpes/Ain/frefr	
place/city/rhoneAlpes/Ardeche/frefr	
place/city_uppercase/rhoneAlpes/Ardeche/frefr	
place/city/rhoneAlpes/Drome/frefr	
place/city_uppercase/rhoneAlpes/Drome/frefr	
place/city/rhoneAlpes/Isere/frefr	
place/city_uppercase/rhoneAlpes/Isere/frefr	

place_frefr.ecr, continued

Entity	Description
place/city/rhoneAlpes/Loire/frefr	
place/city_uppercase/rhoneAlpes/Loire/frefr	
place/city/rhoneAlpes/Rhone/frefr	
place/city_uppercase/rhoneAlpes/Rhone/frefr	
place/city/rhoneAlpes/Savoie/frefr	
place/city_uppercase/rhoneAlpes/Savoie/frefr	
place/city/rhoneAlpes/Haute_Savoie/frefr	
place/city_uppercase/rhoneAlpes/Haute_ Savoie/frefr	
place/city/frefr	French cities.
place/city_uppercase/frefr	French cities in uppercase.

place_fregf.ecr

Entity	Description
place/city2/fregf	French Guianan settlement with over 10,000 inhabitants.
place/city2_uppercase/fregf	French Guianan settlement with over 10,000 inhabitants, in uppercase.
place/canton/fregf	French Guianan canton.
place/canton_uppercase/fregf	French Guianan canton in uppercase.

place_geo_dut.ecr

Entity	Description
place/country/dut	Country in Dutch.
place/country_capital/dut	Country capital in Dutch.

place_geo_eng.ecr

Entity	Description
place/region/eng	Regions. For example, *Asia-Pacific*.
place/region_uppercase/eng	Regions in uppercase.
place/continent/eng	Continents. For example, *Africa*.
place/continent_uppercase/eng	Continents in uppercase.
place/ocean/eng	Oceans. For example, *Pacific*.
place/ocean_uppercase/eng	Oceans in uppercase.
place/country/eng	Countries. For example, *Australia*.
place/country_uppercase/eng	Countries in uppercase.
place/country_capital/eng	Country capitals. For example, *Canberra*.
place/country_capital_uppercase/eng	Country capitals in uppercase.
place/direction/eng	Directions. For example, *Southwest*.
place/direction_uppercase/eng	Directions in uppercase.
place/direction_abb/eng	Direction abbreviations. For example, *SW.*
place/direction_mod/eng	Direction modifiers. For example, *Southwestern, Central, Downtown*.
place/direction_mod_uppercase/eng	Direction modifiers in uppercase.
place/area/eng	Areas. For example, *Cape, Canyon, Grassland, Peninsula.*
place/area_uppercase/eng	Areas in uppercase.
place/street_type/eng	Street types. For example, *Ave*, *Street*, *Place.*
place/street_type_uppercase/eng	Street types in uppercase.

place_geo_fre.ecr

Entity	Description
place/street_type/fre	Street types in French. For example, *Chauss*, *Cloitre*.
place/street_type_lowercase/fre	Street types in lowercase French. For example, *chauss*, *cloitre*.

place_geo_fre.ecr, continued

Entity	Description
place/street_type_uppercase/fre	Street types in uppercase French. For example, *CHAUSS*, *CLOITRE*.
place/house_type/fre	House types in French. For example, *Residence*, *Batiment*.
place/house_type_uppercase/fre	House types in uppercase French.
place/direction/fre	Directions in French. For example, *Sudouest*.
place/direction_uppercase/fre	Directions in uppercase French.
place/direction_abb/fre	Direction abbreviations in French. For example, *NO*.

place_gerat.ecr

Entity	Description
place/city1/gerat	Austrian settlement with over 100,000 inhabitants.
place/city1_uppercase/gerat	Austrian settlement with over 100,000 inhabitants, in uppercase.
place/city2/gerat	Austrian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/gerat	Austrian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/state/gerat	Austrian state.
place/state_uppercase/gerat	Austrian state in uppercase.

place_gerde.ecr

Entity	Description
place/state/gerde	German states.
place/state_uppercase/gerde	German states in uppercase.
place/state_abbrev/gerde	German state abbreviations.
place/city/state_capital/gerde	German state capitals.
place/city_uppercase/state_capital/gerde	German state capitals in uppercase.

place_gerde.ecr, continued

Entity	Description
place/city/bw/gerde place/city_uppercase/bw/gerde place/city/by/gerde place/city_uppercase/by/gerde place/city/be/gerde place/city_uppercase/be/gerde place/city/bb/gerde place/city_uppercase/bb/gerde place/city/hb/gerde place/city_uppercase/hb/gerde place/city/hh/gerde place/city_uppercase/hh/gerde place/city/he/gerde place/city_uppercase/he/gerde place/city/mv/gerde place/city_uppercase/mv/gerde place/city/ni/gerde place/city_uppercase/ni/gerde place/city/nw/gerde place/city_uppercase/nw/gerde place/city/rp/gerde place/city_uppercase/rp/gerde place/city/sl/gerde place/city_uppercase/sl/gerde place/city/sn/gerde place/city_uppercase/sn/gerde place/city/st/gerde place/city_uppercase/st/gerde place/city/sh/gerde place/city_uppercase/sh/gerde	Settlements in each German state, in normal or uppercase.

place_gerde.ecr, continued

Entity	Description
place/city/th/gerde place/city_uppercase/th/gerde	
place/city1/gerde	German settlement with more than 100,000 inhabitants.
place/city1_uppercase/gerde	German settlement with more than 100,000 inhabitants, in uppercase.
place/city/gerde	German cities.
place/city_uppercase/gerde	German cities in uppercase.

place_hrvhr.ecr

Entity	Description
place/city1/hrvhr	Croatian settlement with over 100,000 inhabitants.
place/city1_uppercase/hrvhr	Croatian settlement with over 100,000 inhabitants, in uppercase.
place/city2/hrvhr	Croatian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/hrvhr	Croatian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/hrvhr	Croatian county.
place/county_uppercase/hrvhr	Croatian county in uppercase.

place_hunhu.ecr

Entity	Description
place/city1/hunhu	Hungarian settlement with over 100,000 inhabitants.
place/city1_uppercase/hunhu	Hungarian settlement with over 100,000 inhabitants, in uppercase.
place/city2/hunhu	Hungarian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/hunhu	Hungarian settlement with between 10,000 and 100,000 inhabitants, in uppercase.

place_hunhu.ecr, continued

Entity	Description
place/county/hunhu	Hungarian county.
place/county_uppercase/hunhu	Hungarian county in uppercase.

place_itait.ecr

Entity	Description
place/city1/itait	Italian settlement with over 100,000 inhabitants.
place/city1_uppercase/itait	Italian settlement with over 100,000 inhabitants, in uppercase.
place/city2/itait	Italian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/itait	Italian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region_abbreviation/itait	2-letter abbreviation for an Italian region. For example, RM (includes SCV and RSM).
place/region/itait	Italian region.
place/region_uppercase/itait	Italian region in uppercase.
place/municipality/itait	Italian municipality.
place/municipality_uppercase/itait	Italian municipality in uppercase.
place/island/itait	Italian island.
place/island_uppercase/itait	Italian island in uppercase.
place/locality/itait	Italian place.
place/locality_uppercase/itait	Italian place in uppercase.

place_jpnjp.ecr

Entity	Description
place/prefecture/jpnjp	Japanese prefectures.
place/region/jpnjp	Japanese regions.

place_jpnjp.ecr, continued

Entity	Description
place/city/aichi/jpnjp	Settlements in each Japanese prefecture.
place/city/akita/jpnjp	
place/city/aomori/jpnjp	
place/city/chiba/jpnjp	
place/city/ehime/jpnjp	
place/city/fukui/jpnjp	
place/city/fukuoka/jpnjp	
place/city/fukushima/jpnjp	
place/city/gifu/jpnjp	
place/city/gunma/jpnjp	
place/city/hiroshima/jpnjp	
place/city/hokkaido/jpnjp	
place/city/hyogo/jpnjp	
place/city/ibaraki/jpnjp	

place_jpnjp.ecr, continued

Entity	Description
place/city/ishikawa/jpnjp	
place/city/iwate/jpnjp	
place/city/kagawa/jpnjp	
place/city/kagoshima/jpnjp	
place/city/kanagawa/jpnjp	
place/city/kochi/jpnjp	
place/city/kumamoto/jpnjp	
place/city/kyoto/jpnjp	
place/city/mie/jpnjp	
place/city/miyagi/jpnjp	
place/city/miyazaki/jpnjp	
place/city/nagano/jpnjp	
place/city/nagasaki/jpnjp	
place/city/nara/jpnjp	
place/city/niigata/jpnjp	
place/city/oita/jpnjp	
place/city/okayama/jpnjp	
place/city/okinawa/jpnjp	

place_jpnjp.ecr, continued

Entity	Description
place/city/osaka/jpnjp	
place/city/saga/jpnjp	
place/city/saitama/jpnjp	
place/city/shiga/jpnjp	
place/city/shimane/jpnjp	
place/city/shizuoka/jpnjp	
place/city/tochigi/jpnjp	
place/city/tokushima/jpnjp	
place/city/tokyo/jpnjp	
place/city/tottori/jpnjp	
place/city/toyama/jpnjp	
place/city/wakayama/jpnjp	
place/city/yamagata/jpnjp	
place/city/yamaguchi/jpnjp	
place/city/yamanashi/jpnjp	
place/city/jpnjp	Japanese settlements.
place/misc/jpnjp	Japanese places.

place_kokr.ecr

Entity	Description
place/province/korkr	Province of South Korea, in Korean language.
place/province_DPRK/korkr	Province of North Korea (DPRK) as claimed by South Korea (Republic of Korea), in Korean language.
place/district/korkr	District of South Korea, in Korean language.
place/city1/korkr	Settlement in South Korea with over 100,000 inhabitants, in Korean language.
place/city_DPRK/korkr	Settlement in North Korea (DPRK) as claimed by

place_kokr.ecr, continued

Entity	Description
	South Korea (Republic of Korea), in Korean language.
place/city2/korkr	Settlement in South Korea with between 10,000 and 100,000 inhabitants, in Korean language.

place_lat_long.ecr

Entity	Description
place/lat_long	Geographical co-ordinate in any format (minimum precision is 1/10 degree or one minute of a degree). Supports the components NS, EW, LAT_DEGREES, LAT_DECIMAL, LAT_MINUTES, LAT_SECONDS, LONG_DEGREES, LONG_DECIMAL, LONG_MINUTES, and LONG_SECONDS You can use the `lat_long.lua` script to process this entity.
place/utm	Geographical co-ordinate written using the Universal Transverse Mercator convention. Supports no components.

place_lavlv.ecr

Entity	Description
place/city1/lavlv	Latvian settlement with over 100,000 inhabitants.
place/city1_uppercase/lavlv	Latvian settlement with over 100,000 inhabitants, in uppercase.
place/city2/lavlv	Latvian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/lavlv	Latvian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/municipality/lavlv	Latvian municipality.
place/municipality_uppercase/lavlv	Latvian municipality in uppercase.

place_litlt.ecr

Entity	Description
place/city1/litlt	Lithuanian settlement with over 100,000 inhabitants.
place/city1_uppercase/litlt	Lithuanian settlement with over 100,000 inhabitants, in uppercase.
place/city2/litlt	Lithuanian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/litlt	Lithuanian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/litlt	Lithuanian county.
place/county_uppercase/litlt	Lithuanian county in uppercase.

place_mil_engus.ecr

Entity	Description
place/mil/engus	U.S. military places.
place/mil_uppercase/engus	U.S. military places in uppercase.

place_mulbe.ecr

Entity	Description
place/city1/mulbe	Belgian settlement with over 100,000 inhabitants.
place/city1_uppercase/mulbe	Belgian settlement with over 100,000 inhabitants, in uppercase.
place/city2/mulbe	Belgian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/mulbe	Belgian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/mulbe	Belgian province.
place/province_uppercase/mulbe	Belgian province in uppercase.
place/region/mulbe	Belgian region.
place/region_uppercase/mulbe	Belgian region in uppercase.

place_mulch.ecr

Entity	Description
place/city1/mulch	Swiss settlement with over 100,000 inhabitants.
place/city1_uppercase/mulch	Swiss settlement with over 100,000 inhabitants, in uppercase.
place/city2/mulch	Swiss settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/mulch	Swiss settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/canton/mulch	Swiss canton.
place/canton_uppercase/mulch	Swiss canton in uppercase.
place/canton_abbr/mulch	Two-letter abbreviation for a Swiss canton (always uppercase).

place_mullu.ecr

Entity	Description
place/city2/mullu	Luxembourgish city.
place/city2_uppercase/mullu	Luxembourgish city in uppercase.
place/district/mullu	Luxembourgish district.
place/district_uppercase/mullu	Luxembourgish district in uppercase.
place/canton/mullu	Luxembourgish canton.
place/canton_uppercase/mullu	Luxembourgish canton in uppercase.

place_norno.ecr

Entity	Description
place/city1/norno	Norwegian settlement with over 100,000 inhabitants.
place/city1_uppercase/norno	Norwegian settlement with over 100,000 inhabitants, in uppercase.
place/city2/norno	Norwegian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/norno	Norwegian settlement with between 10,000 and

place_norno.ecr, continued

Entity	Description
	100,000 inhabitants, in uppercase.
place/county/norno	Norwegian county.
place/county_uppercase/norno	Norwegian county in uppercase.
place/island/norno	Norwegian island.
place/island_uppercase/norno	Norwegian island in uppercase.

place_polpl.ecr

Entity	Description
place/city1/polpl	Polish settlement with over 100,000 inhabitants.
place/city1_uppercase/polpl	Polish settlement with over 100,000 inhabitants, in uppercase.
place/city2/polpl	Polish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/polpl	Polish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/polpl	Polish province.
place/province_uppercase/polpl	Polish province in uppercase.
place/county/polpl	Polish county.
place/county_uppercase/polpl	Polish county in uppercase.
place/province/polpl	Polish province (in English).
place/province_uppercase/polpl	Polish province in uppercase (in English).
place/county/polpl	Polish county (in English).
place/county_uppercase/polpl	Polish county in uppercase (in English).

place_porbr.ecr

Entity	Description
place/city1/porbr	Brazilian settlement with over 100,000 inhabitants.

place_porbr.ecr, continued

Entity	Description
place/city1_uppercase/porbr	Brazilian settlement with over 100,000 inhabitants, in uppercase.
place/city2/porbr	Brazilian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/porbr	Brazilian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/state/porbr	Brazilian state.
place/state_uppercase/porbr	Brazilian state in uppercase.
place/island/porbr	Brazilian island.
place/island_uppercase/porbr	Brazilian island in uppercase.

place_porpt.ecr

Entity	Description
place/city1/porpt	Portuguese settlement with over 100,000 inhabitants.
place/city1_uppercase/porpt	Portuguese settlement with over 100,000 inhabitants, in uppercase.
place/city2/porpt	Portuguese settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/porpt	Portuguese settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/district/porpt	Portuguese district.
place/district_uppercase/porpt	Portuguese district in uppercase.
place/island/porpt	Portuguese island.
place/island_uppercase/porpt	Portuguese island in uppercase.

place_rummd.ecr

Entity	Description
place/city1/rummd	Moldovan settlement with over 100,000 inhabitants.
place/city1_uppercase/rummd	Moldovan settlement with over 100,000 inhabitants, in

place_rummd.ecr, continued

Entity	Description
	uppercase.
place/city2/rummd	Moldovan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/rummd	Moldovan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/district/rummd	Moldovan district.
place/district_uppercase/rummd	Moldovan district in uppercase.

place_rumro.ecr

Entity	Description
place/city1/rumro	Romanian settlement with over 100,000 inhabitants.
place/city1_uppercase/rumro	Romanian settlement with over 100,000 inhabitants, in uppercase.
place/city2/rumro	Romanian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/rumro	Romanian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/rumro	Romanian county.
place/county_uppercase/rumro	Romanian county in uppercase.

place_slksk.ecr

Entity	Description
place/city1/slksk	Slovakian settlement with over 100,000 inhabitants.
place/city1_uppercase/slksk	Slovakian settlement with over 100,000 inhabitants, in uppercase.
place/city2/slksk	Slovakian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/slksk	Slovakian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/slksk	Slovakian region.

place_slksk.ecr, continued

Entity	Description
place/region_uppercase/slksk	Slovakian region in uppercase.

place_slvsi.ecr

Entity	Description
place/city1/slvsi	Slovenian settlement with over 100,000 inhabitants.
place/city1_uppercase/slvsi	Slovenian settlement with over 100,000 inhabitants, in uppercase.
place/city2/slvsi	Slovenian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/slvsi	Slovenian settlement with between 10,000 and 100,000 inhabitants, in uppercase.

place_spaar.ecr

Entity	Description
place/city1/spaar	Argentinian settlement with over 100,000 inhabitants.
place/city1_uppercase/spaar	Argentinian settlement with over 100,000 inhabitants, in uppercase.
place/city2/spaar	Argentinian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spaar	Argentinian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/spaar	Argentinian province.
place/province_uppercase/spaar	Argentinian province in uppercase.
place/island/spaar	Argentinian island.
place/island_uppercase/spaar	Argentinian island in uppercase.

place_spabo.ecr

Entity	Description
place/city1/spabo	Bolivian settlement with over 100,000 inhabitants.
place/city1_uppercase/spabo	Bolivian settlement with over 100,000 inhabitants, in uppercase.
place/city2/spabo	Bolivian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spabo	Bolivian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/department/spabo	Bolivian department.
place/department_uppercase/spabo	Bolivian department in uppercase.
place/province/spabo	Bolivian province.
place/province_uppercase/spabo	Bolivian province in uppercase.

place_spacl.ecr

Entity	Description
place/city1/spacl	Chilean settlement with over 100,000 inhabitants.
place/city1_uppercase/spacl	Chilean settlement with over 100,000 inhabitants, in uppercase.
place/city2/spacl	Chilean settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spacl	Chilean settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/spacl	Chilean region.
place/region_uppercase/spacl	Chilean region in uppercase.
place/commune/spacl	Chilean commune.
place/commune_uppercase/spacl	Chilean commune in uppercase.

place_spaco.ecr

Entity	Description
place/city1/spaco	Colombian settlement with over 100,000 inhabitants.
place/city1_uppercase/spaco	Colombian settlement with over 100,000 inhabitants, in uppercase.
place/city2/spaco	Colombian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spaco	Colombian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/department/spaco	Colombian department.
place/department_uppercase/spaco	Colombian department in uppercase.

place_spaec.ecr

Entity	Description
place/city1/spaec	Ecuadorian settlement with over 100,000 inhabitants.
place/city1_uppercase/spaec	Ecuadorian settlement with over 100,000 inhabitants, in uppercase.
place/city2/spaec	Ecuadorian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spaec	Ecuadorian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/province/spaec	Ecuadorian province.
place/province_uppercase/spaec	Ecuadorian province in uppercase.
place/island/spaec	Ecuadorian island.
place/island_uppercase/spaec	Ecuadorian island in uppercase.

place_spaes.ecr

Entity	Description
place/city1/spaes	Spanish settlements with over 100,000 inhabitants.
place/city1_uppercase/spaes	Spanish settlements with over 100,000 inhabitants, in uppercase.

place_spaes.ecr, continued

Entity	Description
place/city2/spaes	Spanish settlements with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spaes	Spanish settlements with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/spaes	Region in Spain.
place/region_uppercase/spaes	Region in Spain in uppercase.
place/province/spaes	Province in Spain.
place/province_uppercase/spaes	Province in Spain in uppercase.
place/island/spaes	Balearic and Canary Islands.
place/island_uppercase/spaes	Balearic and Canary Islands in uppercase.

place_spamx.ecr

Entity	Description
place/city1/spamx	Mexican settlements with over 100,000 inhabitants.
place/city1_uppercase/spamx	Mexican settlements with over 100,000 inhabitants, in uppercase.
place/city2/spamx	Mexican settlements with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spamx	Mexican settlements with between 10,000 and 100,000 inhabitants, in uppercase.
place/state/spamx	States in Mexico.
place/state_uppercase/spamx	States in Mexico in uppercase.
place/islands/spamx	Mexican islands.
place/islands_uppercase/spamx	Mexican islands in uppercase.

place_spape.ecr

Entity	Description
place/city1/spape	Peruvian settlement with over 100,000 inhabitants.

place_spape.ecr, continued

Entity	Description
place/city1_uppercase/spape	Peruvian settlement with over 100,000 inhabitants, in uppercase.
place/city2/spape	Peruvian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spape	Peruvian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/spape	Peruvian region.
place/region_uppercase/spape	Peruvian region in uppercase.

place_spapy.ecr

Entity	Description
place/city1/spapy	Paraguayan settlement with over 100,000 inhabitants.
place/city1_uppercase/spapy	Paraguayan settlement with over 100,000 inhabitants, in uppercase.
place/city2/spapy	Paraguayan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spapy	Paraguayan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/spapy	Paraguayan region.
place/region_uppercase/spapy	Paraguayan region in uppercase.
place/commune/spapy	Paraguayan commune.
place/commune_uppercase/spapy	Paraguayan commune in uppercase.

place_spauy.ecr

Entity	Description
place/city1/spauy	Uruguayan settlement with over 100,000 inhabitants.
place/city1_uppercase/spauy	Uruguayan settlement with over 100,000 inhabitants, in uppercase.
place/city2/spauy	Uruguayan settlement with between 10,000 and 100,000 inhabitants.

place_spauy.ecr, continued

Entity	Description
place/city2_uppercase/spauy	Uruguayan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/department/spauy	Uruguayan department.
place/department_uppercase/spauy	Uruguayan department in uppercase.

place_spave.ecr

Entity	Description
place/city1/spave	Venezuelan settlement with over 100,000 inhabitants.
place/city1_uppercase/spave	Venezuelan settlement with over 100,000 inhabitants, in uppercase.
place/city2/spave	Venezuelan settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/spave	Venezuelan settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/spave	Venezuelan region.
place/region_uppercase/spave	Venezuelan region in uppercase.
place/state/spave	Venezuelan state.
place/state_uppercase/spave	Venezuelan state in uppercase.
place/island/spave	Venezuelan island.
place/island_uppercase/spave	Venezuelan island in uppercase.

place_srpme.ecr

Entity	Description
place/city1/srpme	Montenegrin settlement with over 100,000 inhabitants.
place/city1_uppercase/srpme	Montenegrin settlement with over 100,000 inhabitants, in uppercase.
place/city2/srpme	Montenegrin settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/srpme	Montenegrin settlement with between 10,000 and

place_srpme.ecr, continued

Entity	Description
	100,000 inhabitants, in uppercase.
place/municipality/srpme	Montenegrin municipality.
place/municipality_uppercase/srpme	Montenegrin municipality in uppercase.

place_srprs.ecr

Entity	Description
place/city1/srprs	Serbian settlement with over 100,000 inhabitants.
place/city1_uppercase/srprs	Serbian settlement with over 100,000 inhabitants, in uppercase.
place/city2/srprs	Serbian settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/srprs	Serbian settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/district/srprs	Serbian district.
place/district_uppercase/srprs	Serbian district in uppercase.

place_swese.ecr

Entity	Description
place/city1/swese	Swedish settlement with over 100,000 inhabitants.
place/city1_uppercase/swese	Swedish settlement with over 100,000 inhabitants, in uppercase.
place/city2/swese	Swedish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/swese	Swedish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/county/swese	Swedish county.
place/county_uppercase/swese	Swedish county in uppercase.
place/island/swese	Swedish island.
place/island_uppercase/swese	Swedish island in uppercase.

place_turtr.ecr

Entity	Description
place/city1/turtr	Turkish settlement with over 100,000 inhabitants.
place/city1_uppercase/turtr	Turkish settlement with over 100,000 inhabitants, in uppercase.
place/city2/turtr	Turkish settlement with between 10,000 and 100,000 inhabitants.
place/city2_uppercase/turtr	Turkish settlement with between 10,000 and 100,000 inhabitants, in uppercase.
place/region/turtr	Turkish region.
place/region_uppercase/turtr	Turkish region in uppercase.
place/province/turtr	Turkish province.
place/province_uppercase/turtr	Turkish province in uppercase.
place/district/turtr	Turkish district.
place/district_uppercase/turtr	Turkish district in uppercase.

profanity_chi.ecr

Entity	Description
profanity/biological/chi	Potentially offensive term in Chinese pertaining to biological processes (including obscured representations).
profanity/sexual/chi	Potentially offensive term in Chinese pertaining to sex (including obscured representations).
profanity/personal/chi	Directly insulting term in Chinese (including obscured representations).
profanity/exclaim/chi	Potentially offensive term in Chinese pertaining to exclamation (including obscured representations).
profanity/chi	Any potentially offensive Chinese term (including obscured representations). Eduction gives higher scores to matches with a greater tendency to offend. The following `MinScore` parameter values are provided as a guide:

profanity_chi.ecr, continued

Entity	Description
	• `MinScore=0.7` removes many weakly offensive terms and phrases • `MinScore=1.1` returns moderately-offensive terms and phrases • `MinScore=1.3` returns only strongly offensive terms and phrases • `MinScore=2.5` returns no matches at all
profanity/phrase/chi	Any potentially offensive Chinese phrase (including obscured representations). The following `MinScore` parameter values are provided as a guide: • `MinScore=0.7` removes many weakly offensive terms and phrases • `MinScore=1.1` returns moderately-offensive terms and phrases • `MinScore=1.3` returns only strongly offensive terms and phrases • `MinScore=3.0` returns no matches at all

profanity_eng.ecr

Entity	Description
profanity/blasphemous/eng	Religious term often used for blasphemy (including obscured representations).
profanity/homophobic/eng	Homophobic term (including obscured representations).
profanity/racial/eng	Racial derogatory term (including obscured representations).
profanity/personal/eng	Personally insulting term. Contains all racial and homophobic offensive terms (including obscured representations).
profanity/sexual/eng	Potentially offensive term pertaining to sex (including obscured representations).
profanity/biological/eng	Potentially offensive term pertaining to biological

profanity_eng.ecr, continued

Entity	Description
	processes (including obscured representations).
profanity/censored/eng	Word that appears in the text in a fully-censored format.
profanity/eng	Any potentially-offensive English term (including obscured representations) Eduction gives higher scores to matches with a greater tendency to offend. The following `MinScore` parameter values are provided as a guide: • `MinScore=0.1` removes false matches, for example from URL shorteners • `MinScore=0.7` removes many weakly offensive terms and phrases • `MinScore=1.1` returns moderately-offensive terms and phrases • `MinScore=1.3` returns only strongly offensive terms and phrases • `MinScore=2.5` returns no matches at all

S

The sentiment grammar files have ‘lite’ counterparts. These can process data up to twice as fast compared to the full versions, depending on language. The ‘lite’ versions are identical to the full versions in most respects, but they do not support components or user modification. Micro Focus recommends that you use the ‘lite’ versions except in cases where you want to enable components or modify the built-in dictionaries.

The 'lite' versions are distinguished from the full versions by the addition of *lite* to the file name, preceded by an underscore. For example, the file name of the Chinese sentiment grammar file is `sentiment_chi.ecr`, and the file name of the ‘lite’ version is `sentiment_chi_lite.ecr`.

sentiment_ara.ecr and sentiment_ara_lite.ecr

Entity	Description
sentiment/positive/ara	An Arabic phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/ara	An Arabic phrase that expresses a negative

sentiment_ara.ecr and sentiment_ara_lite.ecr, continued

Entity	Description
	statement. Supports the TOPIC and SENTIMENT components.
sentiment/ara	A positive or negative phrase in Arabic. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_chi.ecr and sentiment_chi_lite.ecr

Entity	Description
sentiment/positive/chi	A Chinese phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/chi	A Chinese phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/chi	A positive or negative phrase in Chinese. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_cze.ecr and sentiment_cze_lite.ecr

Entity	Description
sentiment/positive/cze	A Czech phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/cze	A Czech phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.

sentiment_cze.ecr and sentiment_cze_lite.ecr, continued

Entity	Description
sentiment/cze	A positive or negative phrase in Czech. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_dut.ecr and sentiment_dut_lite.ecr

Entity	Description
sentiment/positive/dut	A Dutch phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/dut	A Dutch phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_eng.ecr and sentiment_eng_lite.ecr

Entity	Description
sentiment/positive/eng	An English phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/eng	An English phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/eng	A positive or negative phrase in English. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_basic_eng.ecr

Entity	Description
sentiment/positive/eng sentiment/negative/eng sentiment/eng	If recall with `sentiment_eng.ecr` is too low, and your documents are generally short comments, use `sentiment_basic_eng.ecr` to extract additional matches. This grammar contains carefully-selected lists of positive and negative terms that help determine the sentiment of a document in which `sentiment_eng.ecr` found no matches. TOPIC and SENTIMENT components are **not** supported. `sentiment_basic_eng.ecr` contains terms in title case, but research shows that for most data these impair recall, so these are given a lower score. Micro Focus recommends that you set `EntityMinScore`*`N`* to `0.4` to filter out these terms unless you need them.

sentiment_fre.ecr and sentiment_fre_lite.ecr

Entity	Description
sentiment/positive/fre	A French phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/fre	A French phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/fre	A positive or negative phrase in French. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_ger.ecr and sentiment_get_lite.ecr

Entity	Description
sentiment/positive/ger	A German phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/ger	A German phrase that expresses a negative

sentiment_ger.ecr and sentiment_get_lite.ecr, continued

Entity	Description
	statement. Supports the TOPIC and SENTIMENT components.
sentiment/ger	A positive or negative phrase in German. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_ita.ecr and sentiment_ita_lite.ecr

Entity	Description
sentiment/positive/ita	An Italian phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/ita	An Italian phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/ita	A positive or negative phrase in Italian. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_pol.ecr and sentiment_pol_lite.ecr

Entity	Description
sentiment/positive/pol	A Polish phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/pol	A Polish phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/pol	A positive or negative phrase in Polish. This entity adds a POSITIVE or NEGATIVE component wrapper

sentiment_pol.ecr and sentiment_pol_lite.ecr, continued

Entity	Description
	to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_por.ecr and sentiment_por_lite.ecr

Entity	Description
sentiment/positive/por	A Portuguese phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/por	A Portuguese phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/por	A positive or negative phrase in Portuguese. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_rus.ecr and sentiment_rus_lite.ecr

Entity	Description
sentiment/positive/rus	A Russian phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/rus	A Russian phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/rus	A positive or negative phrase in Russian. This entity adds a POSITIVE or NEGATIVE component wrapper

sentiment_rus.ecr and sentiment_rus_lite.ecr, continued

Entity	Description
	to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_spa.ecr and sentiment_spa_lite.ecr

Entity	Description
sentiment/positive/spa	A Spanish phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/spa	A Spanish phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/spa	A positive or negative phrase in Spanish. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable. Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

sentiment_tur.ecr and sentiment_tur_lite.ecr

Entity	Description
sentiment/positive/tur	A Turkish phrase that expresses a positive statement. Supports the TOPIC and SENTIMENT components.
sentiment/negative/tur	A Turkish phrase that expresses a negative statement. Supports the TOPIC and SENTIMENT components.
sentiment/tur	A positive or negative phrase in Turkish. This entity adds a POSITIVE or NEGATIVE component wrapper to an empty string after the match. You can use this component to determine the sentiment of the phrase. Use this entity when faster performance is desirable.

sentiment_tur.ecr and sentiment_tur_lite.ecr, continued

Entity	Description
	Micro Focus recommends that you configure Eduction to allow all duplicates, and set `TangibleCharacters` to `:;@#`.

T

team_american_football.ecr

Entity	Description
org/football/us	American Football team in the U.S.
org/football/ca	Canadian Football team in Canada. All synonyms for team names produce the same normalized text (for example, *The Bears* normalizes to *Chicago Bears*) to identify variant team names.

team_baseball.ecr

Entity	Description
org/baseball/mlb	Major League baseball team in the U.S. and Canada. All synonyms for team names produce the same normalized text (for example, *LA Dodgers* normalizes to *Los Angeles Dodgers*) to identify variant team names.

team_basketball.ecr

Entity	Description
org/basketball/nba	Basketball team in the NBA. All synonyms for team names produce the same normalized text (for example, *Sixers* normalizes to *Philadelphia 76ers*) to identify variant team names.

team_hockey.ecr

Entity	Description
org/hockey/nhl	Hockey team in the NHL. All synonyms for team names produce the same normalized text (for example, *NJ Devils* normalizes to *New Jersey Devils*) to identify variant team names.

team_soccer.ecr

Entity	Description
org/soccer/us	Soccer team in U.S. and Canada (Major League Soccer).
org/soccer/gb	Football (soccer) team in the United Kingdom. Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Celtic*.
org/soccer/de	Football (soccer) team in Germany (current Bundesliga teams). Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Wolfsburg*.
org/soccer/fr	Football (soccer) team in France. Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Nice*.
org/soccer/nl	Football (soccer) team in the Netherlands. Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Ajax*.
org/soccer/es	Football (soccer) team in Spain (current Primera & Segunda Divisiónes teams). Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Barcelona*.
org/soccer/it	Football (soccer) team in Italy (current teams in Serie A and Serie B). Set `EntityMinScoreN=0.99` to filter out ambiguous names such as *Inter*. All synonyms for team names produce the same normalized text (for example, *Man United* normalizes to *Manchester United*) to identify variant team names.

time_chi.ecr

Entity	Description
time/time_of_day/chi	A descriptive time of day in Chinese.
time/time_of_day_simplified/chi	A descriptive time of day in simplified Chinese.
time/period/chi	An amount of time in Chinese.
time/period_simplified/chi	An amount of time in simplified Chinese.
time/alpha_time/chi	Time of the day in Chinese words.
time/alpha_time_simplified/chi	Time of the day in simplified Chinese words and ASCII numbers.
time/hms/chi	Time in hours and minutes with optional seconds and fractions thereof.
time/hms_simplified/chi	Time in hours and minutes with optional seconds and fractions thereof, in simplified Chinese and ASCII numbers.
time/chi	Any time of day in Chinese, in a variety of formats.
time/simplified/chi	Any time of day in simplified Chinese and ASCII numbers, in a variety of formats.

time_eng.ecr

Entity	Description
time/time_of_day/eng	A descriptive time of day in English. For example, *dawn*, *morning*, *Mid-afternoon.*
time/period/eng	An amount of time. For example, *day*, *quarter*, *month*, *decades.*
time/alpha_time/eng	Time of day in English words, for example, *4 o'clock*, *ten past five*.
time/hms/eng	Time in hours and minutes with optional seconds and fractions thereof.
time/eng	Any time in English or numeric format. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54

time_eng.ecr, continued

Entity	Description
	• 20:20:20.2020202020202020 • 04:54 a.m. • 02:20 at night • quarter past midnight • 20 to midnight • ten past six • One o'clock • 6.10pm • 1.49 in the afternoon • noon • 5:00 UTC+1 • 19:15 Hawaii-Aleutian Time

time_fre.ecr

Entity	Description
time/time_of_day/fre	A descriptive time of day in French. For example, *l'aube*, *Matin*.
time/period/fre	An amount of time. For example, *une décennie*, *un siècle*.
time/alpha_time/fre	Time of day in French words. For example, *sept heures du matin*, *trois heures de l'après-midi*.
time/hms/fre	Time in hours and minutes with optional seconds and fractions thereof.
time/fre	Any time in French or numeric format. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54 • 20:20:20.2020202020202020 • 04:54 du matin

time_fre.ecr, continued

Entity	Description
	• 4.54 de la nuit • minuit et 15 • midi moins vingt • 6 heures 20 du soir • 1 heure 49 de l'apres midi • une heure trente cinq • six heures et dix • midi • 5:00 UTC+1 • 19:15 PDT

time_ger.ecr

Entity	Description
time/time_of_day/ger	A descriptive time of day. For example, *Nachmittag*.
time/period/ger	An amount of time in German (all declensions). For example, *Jahrzehnt*.
time/alpha_time/ger	Time of day in German words. For example, *fünf nach zehn*.
time/hms/ger	Time in hours and minutes with optional seconds and fractions thereof.
time/ger	Any time in German or numeric format. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54 • 20:20:20.2020202020202020 • 04:54 morgens • 4.54 nachts • viertel nach mitternacht • 6.20 nachmittags

time_ger.ecr, continued

Entity	Description
	• 1 Uhr 49 nachmittags • Fünf Uhr • Sechs Uhr Zehn • mittag • 5:00 UTC+1 • 19:15 PDT

time_ita.ecr

Entity	Description
time/time_of_day/ita	A descriptive time of day in Italian. For example, *pomeriggio*.
time/period/ita	An amount of time in Italian. For example, *giorno*, *Mesi*, *secolo*.
time/alpha_time/ita	Time of day in Italian words. For example, *Sono le 4*, *5 y 10 delpomeriggio*, *mezzanotte meno cinque.*
time/hms/ita	Time in hours and minutes with optional seconds and fractions thereof.
time/ita	Any time in Italian or numeric format. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54 • 20:20:20.2020202020202020 • 4.54 del mattino • Tre e tre quarti di notte • un quarto alle sette • dieci all'una • sono le due meno cinque • 6 e 20 • 1 e 49 del pomeriggio

time_ita.ecr, continued

Entity	Description
	• 13:35 • sei e dieci • Mezzo giorno • 5:00 UTC+1 • 19:15 PDT

time_numeric.ecr

Entity	Description
time/hms12	12-hour time in hours and minutes, with optional seconds and fractions.
time/hms24	24-hour time in hours and minutes, with optional seconds and fractions.
time/tz_abbr	Standard timezone abbreviations.
time/tz_abbr_plus	Standard timezone abbreviations with optional +/- hh:mm modifier.

time_por.ecr

Entity	Description
time/time_of_day/por	A descriptive time of day in Portuguese. For example, *manhã*, *pôr do dol.*
time/period/por	An amount of time in Portuguese. For example, *dia*, *Mês*, *séculos.*
time/alpha_time/por	Time of day in Portuguese words. For example, *São dez*, *doze e um quarto da noite*, *meia-noite menos 15*.
time/hms/por	Time in hours and minutes with optional seconds and fractions thereof.
time/por	Any time in Portugese. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54

time_por.ecr, continued

Entity	Description
	• 20:20:20.2020202020202020 • 4.54 da manhã • doze e quarto da noite (Brazilian Portuguese) • São vinte e cinco para as cinco da manhã • cinco e vinte da manhã • 1 e 49 da tarde • 13:35 • seis e dez • Meio-dia • 5:00 UTC+1 • 19:15 PDT • 7 em ponto

time_spa.ecr

Entity	Description
time/time_of_day/spa	A descriptive time of day in Spanish. For example, *a la medianoche*, *al amanecer*.
time/period/spa	An amount of time in Spanish. For example, *década*.
time/alpha_time/spa	Time of day in Spanish words. For example, *a media mañana*.
time/hms/spa	Time in hours and minutes with optional seconds and fractions thereof.
time/spa	Any time in Spanish or numeric format. Supported formats include: • 20:20 GMT+0100 • 00:15 • 4:54 • 20:20:20.2020202020202020 • 04:54 de la mañana • 4.54 por la noche

time_spa.ecr, continued

Entity	Description
	• doce y cuarto de la noche • Son las cinco menos veinticinco de la mañana • cinco y veinte de la mañana • 1 y 49 de la tarde • 13:35 • seis y diez • mediodía • 5:00 UTC+1 • 19:15 PDT • 7 en punto

transport_airport.ecr

Entity	Description
airport/icao	Airport ICAO code.
airport/iata	Airport IATA code.

transport_car.ecr

Entity	Description
car/make_model	Make and model of car.

U

university.ecr

Entity	Description
org/university	A university.

Standard Grammar – Source

Eduction includes standard grammar files in source form (XML) and their compiled equivalents (ECR). The source files import compiled Eduction standard grammar files and illustrate sample usage. Customers can modify these XML source files and recompile them to customize a grammar for the needs of an Eduction application. The following table lists public entities defined in the XML source files. It excludes the public entities that are republished from the imported Eduction ECR grammar files.

File	Entity	Description
measure.xml	measure/all/eng	An editable collection of patterns that match length, area, volume, and mass.
money.xml[1]	money/all	All currency amounts. **NOTE:** This grammar file supports some English alphabetic numbers, for example, *seven cents*, *$12 million*, *one hundred dollars*, *£5m*.
pci_dss.xml	pci_dss/person_name/engus pci_dss/date/engus pci_dss/credit_card/engus pci_dss/bank_names/engus	Person names. Dates. Credit and debit card numbers. Bank names.
pii.xml	pii/person_name/engus pii/phone_number/engus pii/email_address/engus pii/ip_address/engus	Personal names. Phone numbers. Email addresses. IP addresses.

[1]When matching symbols in the money entities, the Eduction option `MatchWholeWord` must be set to `0` (false). Otherwise, when encountering a string such as $10.70, Eduction will not recognize that *$* is the start of a token. Instead, it looks only for matches starting on the *1* and on the *7,* and will not return *$10.70*.

File	Entity	Description
	pii/social_security/engus	Social Security numbers.
	pii/car_numberplate/engus	Car license plate numbers.
	pii/driver_license/engus	Driver's license numbers.
	pii/credit_card/engus	Credit and debit card numbers.
	pii/date/engus	Dates.
	pii/country	Countries.
	pii/state/engus	U.S. states or possessions.
	pii/county/engus	U.S. counties.
	pii/city/engus	U.S. cities.
	pii/address/engus	Geographical addresses.
	pii/zipcode/engus	U.S. zipcodes.
	pii/age/engus	Age.
	pii/gender/engus	Gender.
	pii/race/engus	Race.
	pii/job_title/engus	Job title.
	pii/disease_and_condition/engus	Disease or medical condition.
	pii/account_number/engus	Generic account number with 6-8 digits in a predictable context.
	pii/license_number/engus	Generic license number with specific alphanumeric format.
	pii/facebook_url/engus	Example URL for a personal Web page (Facebook).

File	Entity	Description
place_europe.xml	place/country/europe	European country in English (and some local languages).
	place/country_uppercase/europe	European country in English and local languages (uppercase).
	place/city1/europe	European settlement with over 100,000 inhabitants, in local language.
	place/city1_uppercase/europe	European settlement with over 100,000 inhabitants, in local language (uppercase).
	place/city2/europe	European settlement with between 10,000 and 100,000 inhabitants, in local language.
	place/city2_uppercase/europe	European settlement with between 10,000 and 100,000 inhabitants, in local language (uppercase).
	place/region/Europe	High-level administrative division, in local language.
	place/region_uppercase/Europe	High-level administrative division, in local language (uppercase).

File	Entity	Description
place_south_ america.xml	place/country/south_america	South American country in English, Spanish, or Portuguese.
	place/country_uppercase/south_america	South American country in English, Spanish, or Portuguese (uppercase).
	place/city1/south_america	South American settlement with over 100,000 inhabitants, in local language.
	place/city1_uppercase/south_america	South American settlement with over 100,000 inhabitants, in local language (uppercase).
	place/city2/south_america	South American settlement with between 10,000 and 100,000 inhabitants, in local language.
	place/city2_uppercase/south_america	South American settlement with between 10,000 and 100,000 inhabitants, in local language (uppercase).
	place/island/south_america	South American island, in local language.
	place/island_uppercase/south_america	South American island, in local language (uppercase).
	place/region/south_america	High-level administrative division, in local language.
	place/region_uppercase/south_america	High-level administrative division, in local language (uppercase).
retention.xml	retention/admission_date retention/discharge_date retention/birth_date retention/age/eng	Admission date. Discharge date. Birth date. Age.
sample.xml	sample/solar_system	A simple entity for planets of the solar system.
sentiment_user_ chi.xml	sentiment/user_client_name sentiment/user_client_brand sentiment/user_client_rv1_name	You can use these files to modify the sentiment analysis grammar files for the relevant languages to give access to extra domain-specific vocabulary.

File	Entity	Description
	sentiment/user_client_rv1_brand sentiment/user_third_party_company_name sentiment/user_third_party_company_brand sentiment/user_positive_adjective sentiment/user_negative_adjective sentiment/user_positive_noun sentiment/user_negative_noun sentiment/user_neutral_noun sentiment/user_positive_verb sentiment/user_negative_verb sentiment/user_neutral_verb sentiment/user_positive_idiom sentiment/user_negative_idiom	
sentiment_user_ara.xml sentiment_user_cze.xml sentiment_user_dutch.xml sentiment_user_eng.xml sentiment_user_	sentiment/user_positive_adjective sentiment/user_negative_adjective sentiment/user_neutral_adjective sentiment/user_positive_adverb sentiment/user_negative_adverb sentiment/user_neutral_adverb sentiment/user_positive_noun sentiment/user_negative_noun	

File	Entity	Description
fre.xml sentiment_user_ger.xml sentiment_user_ita.xml sentiment_user_pol.xml sentiment_user_por.xml sentiment_user_rus.xml sentiment_user_spa.xml sentiment_user_tur.xml	sentiment/user_neutral_noun sentiment/user_positive_verb sentiment/user_negative_verb sentiment/user_neutral_verb sentiment/user_positive_match sentiment/user_negative_match sentiment/user_good_noun (English only)	

The entities above incorporate the compiled Eduction entities in combination with Eduction XML grammar to create additional entities. The XML illustrates how to use the compiled Eduction entities. You can modify these XML files and compile them into Eduction ECR files that can then be used for specific applications.

The Eduction grammar files have three advantages:

- Allows for fined-grained access to basic entities that include more complex entities. Allows you to customize the complex entities to increase the precision and recall of the matching process.
- Provides both the compiled ECR grammar files as well as source-form XML grammar files that reference them.
- Separate ECR files reduce the memory footprint and file size.

Chapter 8: Grammar Reference

Eduction uses Grammar files to identify and tag entities in documents. They are written in XML in a format specific to Eduction. They are then compiled, using the Eduction command-line tool, into ECR files that Eduction can easily read at runtime. Eduction includes a collection of standard grammar files that make it easy to identify common entities such as names and phone numbers. These are described in .

Create and Edit Grammar Files

An Eduction grammar defines patterns for matching text in a document. A pattern is a combination of characters and operators. An operator is a sequence of special characters that match text by following the rules associated with the operator.

Pattern	Description	Matches
Smith|John	Match either *Smith* or *John*	*Smith*

Pattern	Description	Matches
		John
[0-9]{3}	Match a sequence of three characters in the range 0 through 9	*123* *456*

In the above example, the square bracket operators [] are used to match on any of the characters 0 through 9 and the curly braces {} are used to repeat the previous pattern three times.

Grammars are described using XML. The template that defines the XML that Eduction understands is contained in the file `edk.dtd`. When writing grammars for Eduction, Micro Focus recommends that you reference `edk.dtd` at the start of the XML grammar file using the include statement, and that you use a DTD-compatible XML authoring tool to eliminate syntax errors and save time.

Here is an example of a simple Eduction grammar:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE grammars SYSTEM "edk.dtd">
<grammars>
   <grammar name="mygrammar">
      <entity name="name" type="public">
         <pattern>Smith|John</pattern>
      </entity>
      <entity name="digits" type="public">
         <pattern>[0-9]{3}</pattern>
      </entity>
   </grammar>
</grammars>
```

This grammar defines two entities: `mygrammar/name` and `mygrammar/digits`.

For a more extensive set of example Eduction grammar files, see Example Grammar Files, on page 278.

Compile Grammars

After a grammar is written, compile the XML file into an ECR file using the Eduction command-line tool `edktool`. XML files are easy for people to read, but inefficient for computers to process. `edktool` transforms the XML file into an ECR file that is efficient for Eduction to use directly. An example of the `edktool` compile command is:

```
edktool c mygrammar.xml
```

This command produces the output file `mygrammar.ecr`.

Eduction Grammar Syntax

The tables in this section describe the Eduction grammar syntax defined in the `edk.dtd` (see Eduction Grammar DTD, on page 281).

In the tables, terms shown in *italics* are deprecated, but are kept for backward compatibility. The terms in angled brackets <> describe the value that must be inserted. Note that the XML elements, attributes, and values are defined in lower case. Although the Eduction compiler accepts uppercase element and attribute names, this functionality is deprecated, but retained for backward compatibility. The `edk.dtd` file represents the current definition for Eduction grammar files, and must be followed for all Eduction grammars.

> **NOTE:**
> Two deprecated elements are missing from the tables below: `dictionary` and `entryset`. These are synonymous with `grammar` and `entity` respectively, and, although not documented, are retained for backward compatibility.

<grammars>

Element: grammars

Child Elements: include, grammar

Description: This is the top-level element in an Eduction grammar.

Example: `<grammars version="1.0" debug="true" case="sensitive">`

Attribute	Value	Default	Description
`version`	<version string>	none	An optional character string providing version information for the grammar.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` takes the value from the application level, which in the case of Eduction applications is usually `sensitive`.
`debug`	true *1/on/yes/true*	false	Displays verbose information for the grammars element while `edktool` compiles the grammar.

Attribute	Value	Default	Description
	false *0/off/no/false*		

<include>

Element: include

Child Elements: publish

Description: References another Eduction grammar file for inclusion.

Example: `<include path="winter_names.ecr" type="private"/>`

Attribute	Value	Default	Description
`path`	<path to the grammar file>		A value is required.
`type`	public private	public	The default setting of public allows entities in included XML grammars to retain their private/public visibility. (Included ECR grammars, by definition of a compiled grammar, only contain public entities.) Setting the type attribute to **`private`** hides the included public entities from being visible in the file that includes the grammar.

<publish>

Element: publish

Child Elements: <none>

Description: Makes a private entity public. The entity can be anywhere in an included XML file chain. Note that private entities cannot be accessed in a compiled ECR file, so that even if the name of the private entity is known, publish is not able to make it public.

Example: `<publish name="grammar2/g2e2"/>`

Attribute	Value	Default	Description
`name`	<entity name>		Makes a private entity in an included XML file public. A value is required.

<grammar>

Element: grammar

Child Elements: extern, entity

Description: Defines a grammar, which is a collection of entities. Entities are used for matching.

Example: `<grammar name="grammar1" case="inherited" extend="disallow" debug="inherited">`

Attribute	Value	Default	Description
`name`	<grammar name>		A value is required.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `grammars` parent.
`extend`	append replace disallow	disallow	Extends or replaces an existing grammar definition, or disallows this if one already exists.
`debug`	true *1/on/yes/true* false *0/off/no/false* inherited	inherited	Displays verbose information for the dictionary element during compilation. The value `inherited` accepts the debug mode of the `grammars` parent.

<extern>

Element: extern

Child Elements: <none>

Description: Identifies an external grammar by name so that the entities contained by the grammar do not have to explicitly name the grammar. For example, if another grammar is `grammar1` and an entity within it is `entity1`, then in the current grammar, the entity can be referred to as simply `entity1` rather than `grammar1/entity1`.

Example: `<extern name="grammar2"/>`

Attribute	Value	Default	Description
`name`	<grammar name>		Identifies the name of the grammar. A value is required.

<entity>

Element: entity

Child Elements: entry, pattern

Description: Defines an entity used for matching.

Example: `<entity name="entity1" type="public" case="insensitive" extend="disallow" debug="true">`

Attribute	Value	Default	Description
`name`	<grammar name>		A value is required.
`type`	public private	private	Defines the entity as public or private.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `grammars` parent.
`extend`	append replace disallow	disallow	Extends or replaces an existing entity definition.
`debug`	true *1/on/yes/true* false *0/off/no/false* inherited	inherited	Displays verbose information for the entity element during compilation. The value `inherited` accepts the debug mode of the `grammar` parent.

<entry>

Element: entry

Child Elements: headword, synonym

Description: An entry represents an individual entry that is matched in an entity. The entry has one or more attributes such as the actual phrase that is returned (the *headword*), the case, and so on.

Example:`<entry headword="mat" score=".3" case="inherited" debug="inherited">`

Attribute	Value	Default	Description
`headword`	#CDATA		The dictionary entry. Headword can be an attribute or a subelement, but it must be one or the other.
`score`	>= 0	1	Can be used to assign any weightings to the matches. • A score of `1` is the default score.

Attribute	Value	Default	Description
			• A score of `0` always excludes the matching tag from the results, and can be used to specify exceptions to grammar rules. You can use these weightings for a variety of purposes: • They can represent the confidence the grammar author has in the accuracy of the match (where a value of `1` represents certainty, and lower values represent lesser confidence). • They can represent the importance of a match - for example, in the sentiment grammars the scoring represents the strength of the sentiment in the match. Multiple scores are multiplied. For example, if a match on an entity has a score of `1.5`, and that entity is used in another entity that also has a score of `0.4`, the resulting score is `0.6`. If a minimum score is specified during extraction, only those matches with a sufficiently high score are extracted. You can also display the exact scores of any match during extraction. **NOTE:** Entries and patterns should be assigned a score no lower than `0.01` and no higher than `100`.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `entryset` parent.
`debug`	true *1/on/yes/true* false *0/off/no/false* inherited	inherited	Displays verbose information for the entry element during compilation. The value `inherited` accepts the debug mode of the `entryset` parent.

<headword>

Element: headword

Child Elements: <none>

Description: A headword is the sequence of characters that produce an entity match.

Example: See example in <entry>, on the previous page.

Attribute	Value	Default	Description
`<element contents>`	<the headword>		The headword value. Note that if the `entry` element contains a headword attribute, it cannot have a headword subelement.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `entry` parent.
`score`	>= 0	1	See description in <entry>, on page 271.

<synonym>

Element: synonym

Child Elements: <none>

Description: A synonym is an alternative sequence of characters to a headword. Synonym matching produces an entity match, but returns the headword in place of the matching synonym. For example, if you search for *dog* with the synonym *canine* enabled, matches for *canine* return as if they matched *dog*.

Example:

```
<entry headword="Vatican City">
   <synonym>The Vatican</synonym>
   <synonym>Holy See</synonym>
   <synonym>Città del Vaticano</synonym>
   <synonym>Citta del Vaticano</synonym>
</entry>
```

Attribute	Value	Default	Description
`<element contents>`	<the synonym>		The synonym value.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `entry` parent.

<pattern>

Element: pattern

Child Elements: <none>

Description: Defines a pattern used for matching.

Example: `<pattern score=".1" case="insensitive" replace="replacechars" insert_before="prefix_" insert_after="_suffix">cat</pattern>`

Attribute	Value	Default	Description
`pattern`	<actual pattern>		A value is required.
`score`	>= 0	1	See description in <entry>, on page 271.
`case`	sensitive *1/on/yes/true* insensitive *0/off/no/false* inherited	inherited	Determines whether a match is case sensitive. The value `inherited` accepts the case matching mode of the `grammars` parent.
`replace`	<text to replace the match>	<no default>	The matched text is replaced with the specified text.
`insert_before`	<text to insert before the match>	<no default>	Matched text is prefixed with the specified text.
`insert_after`	<text to insert after the match>	<no default>	Matched text is suffixed with the specified text.
`debug`	true *1/on/yes/true* false *0/off/no/false* inherited	inherited	Displays verbose information for the pattern element during compilation. The value `inherited` accepts the debug mode of the `entity` parent.

Regular Expressions

This section describes the regular expressions syntax that Eduction supports.

The engine's parser interprets regular expression syntax nearly identically to the UNIX regular expression syntax. The engine's regular expression syntax also includes some extensions for matching substrings.

Operators

The following table the base regular expression operators available in the Eduction engine and the pattern the operator matches.

Operator	Matched Pattern
`\`	Quote the next metacharacter.
`^`	Match the beginning of a line.
`$`	Match the end of a line.
`.`	Match any character (except newline).
`|`	Alternation.
`()`	Used for grouping to force operator precedence.
`[xy]`	The character `x` or `y`.
`[x-z]`	The range of characters between `x` and `z`.
`[^z]`	Any character except `z`. **NOTE:** For performance reasons, Micro Focus recommends that you explicitly list all the characters that you want to match, rather than using this operator. **NOTE:** To use negated character classes in case-insensitive entities, you must include letters in both cases, for example `[^Zz]` rather than `[^z]`.

Quantifiers

Operator	Matched Pattern
`*`	Match 0 or more times.
`+`	Match 1 or more times.
`?`	Match 0 or 1 times.
`{n}`	Match exactly *n* times.
`{n,}`	Match at least *n* times.
`{n,m}`	Match at least *n* times, but no more than *m* times.

Metacharacters

Operator	Matched Pattern
`\t`	Match tab.

Operator	Matched Pattern
`\n`	Match newline.
`\r`	Match return.
`\f`	Match formfeed.
`\a`	Match alarm (bell, beep, and so on).
`\e`	Match escape.
`\v`	Match vertical tab.
`\021`	Match octal character (in this example, 21 octal).
`\xF0`	Match hex character (in this example, F0 hex).
`\x{263a}`	Match wide hex character (Unicode).
`\w`	Match word character: `[A-Za-z0-9_]`.
`\W`	Match non-word character: `[^A-Za-z0-9_]`.
`\s`	Match whitespace character. This metacharacter also includes `\n` and `\r`: `[ \t\n\r]`.
`\S`	Match non-whitespace character: `[^ \t\n\r]`.
`\d`	Match digit character: `[0-9]`.
`\D`	Match non-digit character: `[^0-9]`.
`\b`	Match word boundary.
`\B`	Match non-word boundary.
`\A`	Match start of string (never match at line breaks).
`\Z`	Match end of string. Never match at line breaks; only match at the end of the final buffer of text submitted for matching.

Extensions

Operator	Matched Pattern

`(?A:` *`entity`*`)`	Match a previously defined entity, which is then copied into the new entity's definition.
	For example:
	`<include path="number_types_eng.ecr"/>`
	`<entity name="fracpos" type="private">`
	`<pattern>(?A:number/fracalpha/eng)</pattern>`
	`</entity>`

Operator	Matched Pattern
	Copying an entity improves pattern execution speed, but increases compilation time and memory usage. It is recommended unless the copied entry is large and is copied multiple times.
(?A^ *entity*)	Match a previously defined entity, which is then referenced by the new entity. Referencing an entity minimizes the size and memory usage of the grammar, but decreases performance. The performance impact can vary from unnoticeable to significant, depending on the size and structure of the grammar.

(?A! *expr*)	Match the expression *expr* but exclude its output. Designates an expression that helps identify an entity, but is not part of it.
	For example:
	`<grammars>`
	`  <grammar name="person">`
	`  <entity name="age" type="public">`
	`  <pattern>(?A!Age:\s)[1-9][0-9]?</pattern>`
	`  </entity>`
	`  </grammar>`
	`</grammars>`
	If this grammar is used to search the text
	`  Name: Simon. Age: 32. Address. 12 Fifth Street, Las Vegas.`
	the text `32` is returned but `12` is ignored because it does not have the prefix "`Age:`", which is matched upon but excluded from the output.

(?A= *component:expr*)	Define a component within an entity's definition. A component is a named part of an entity.
	For example, the following grammar defines `areacode` and `main` as components:
	`<grammars>`
	`  <grammar name="number">`
	`  <entity name="phone" type="public">`
	`  <pattern>(?A=areacode:[0-9]{3})-(?A=main:[0-9]{3}-[0-9]{4})</pattern>`
	`  </entity>`
	`  </grammar>`
	`</grammars>`
	If the data is as follows
	`  The phone number is 408-555-1342.`
	and the following configuration options are set
	`  <OutputSimpleMatchInfo>false</OutputSimpleMatchInfo>`
	`  <EnableComponents>true</EnableComponents>`

Operator	Matched Pattern
	then the output displays the `areacode` value `408` and the main value `555-1342` separately.

Token Properties

> **CAUTION:**
> Token properties will be deprecated in a future release. Users should use the equivalent explicit regular expresions instead of token properties.

Operator	Match Pattern
`(?A:{`*`properties`*`})`	Matches a token that satisfies the list of properties provided. The properties are specified in a comma-separated list of one or more of the following: • num, alpha_num • all_caps, mixed_case, capword Any of these properties can be prefixed with the negation operator '!' for exclusion.

Example Grammar Files

The following sample grammar files contains the `gram_edk_place.xml` grammar.

grammar.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE grammars SYSTEM "../published/edk.dtd">
<!-- Sample Eduction grammar file showing all elements and attributes in the DTD --
>
<grammars debug="true" case="sensitive">
  <include path="grammar_include.xml" type="private">
    <publish name="grammar2/g2e2"/>  <!-- publish previously private entity -->
  </include>
  <grammar name="grammar1" case="inherited" extend="disallow" debug="inherited">
    <extern name="grammar2"/> <!-- removes the need to refer explicitly to grammar2
-->
```

```
    <entity name="entity1" type="public" case="insensitive" extend="disallow"
debug="true">
    <!-- the following entity definitions are not useful but are provided only to
illustrate the options and combinations of elements and attributes available -->
    <pattern score=".1" case="insensitive" replace="replacechars" insert_
before="prefix_" insert_after="_suffix">cat</pattern>
    <pattern score=".2">sat</pattern>
    <entry headword="mat" score=".3" case="inherited" debug="inherited">
      <synonym case="inherited">rug</synonym> <!-- will locate rug but return mat -
->
      <!-- will locate rug but return mat -->
      <synonym case="inherited"><![CDATA[carpet]]></synonym> <!-- illustrates
allowing CDATA in this element -->
      </entry>
      <entry headword="dog" score=".6"/>
      <entry>
        <headword score=".8"><![CDATA[rabbit<hi!>&abc&amp;]]></headword>
        <synonym>bunny</synonym>
      </entry>
    </entity>
    <entity name="entity2" type="public">
      <pattern>(?A:g2e1)</pattern>
    </entity>
  </grammar>
</grammars>
```

grammar_include.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE grammars SYSTEM "../published/edk.dtd">
<grammars>
   <grammar name="grammar2">
      <entity name="g2e1">
         <pattern>animal</pattern> <!-- default visibility -->
      </entity>
      <entity name="g2e2" type="private"> <!-- explicitly private -->
         <pattern>mineral</pattern>
      </entity>
      <entity name="g2e3" type="public"> <!-- explicitly public -->
         <pattern>vegetable</pattern>
      </entity>
   </grammar>
</grammars>
```

Example Grammar File to Match Months

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE grammars SYSTEM "../published/edk.dtd">
```

```
<grammars version="4.0">
   <include path="winter_names.ecr" type="private"/>
   <grammar name="example">

      <entity name="spring_month" type="private">
         <pattern>[Mm]ar(ch|\.)</pattern>
         <entry headword="April"/>
         <entry headword="april"/>
         <entry headword="Apr"/>
         <entry headword="apr"/>
         <entry headword="Apr."/>
         <entry headword="apr."/>
         <pattern replace="May">[Mm]ay\.?</pattern>
         <entry headword="June">
            <synonym>Jun</synonym>
            <synonym>Jun.</synonym>
            <synonym>june</synonym>
            <synonym>jun</synonym>
            <synonym>jun.</synonym>
         </entry>
      </entity>

      <entity name="summer_month" type="private" case="insensitive">
         <entry headword="June"/>
         <entry headword="July"/>
         <entry headword="August"/>
         <entry headword="September"/>
      </entity>

      <entity name="month" type="public">
         <pattern>(?A^spring_month)</pattern>
         <pattern>(?A:summer_month)</pattern>
         <entry headword="September"/>
         <entry headword="October"/>
         <entry headword="November"/>
         <entry headword="December"/>
         <pattern>(?A^winter_month)</pattern>
         <!-- spelling mistakes -->
         <entry score="0.5" headword="Febuary"/>
      </entity>

   </grammar>
</grammars>
```

Simplified Grammar File Containing a Dictionary of Place Names

> **NOTE:**
> The following grammar file is a simplified version provided for example purposes, rather than actual source code.

```
<entity name="city/spain" type="public">
    <entry headword="Barcelona"/>
    <entry headword="Ciudad Real"/>
    <entry headword="Granada"/>
    <entry headword="Madrid"/>
</entity>
<entity name="city/germany" type="headword">
    <entry headword="Berlin"/>
    <entry headword="Frankfurt"/>
    <entry headword="München"/>
    <entry headword="Leipzig"/>
</entity>
```

Simplified Grammar File Containing Patterns to Match Times of Day

> **NOTE:**
> The following grammar file is a simplified version provided for example purposes, rather than actual source code.

```
<entity name="time_24_hour" type="public">
   <pattern>[01][0-9]:[0-5][0-9]</pattern>
   <pattern>2[0-3]:[0-5][0-9]</pattern>
</entity>

<entity name="time_all" type="public">
   <pattern>(?A:time_24_hour)</pattern>
   <entry headword="Midnight"/>
   <entry headword="midnight"/>
   <pattern>([1-9]|10|11|12) ?[ap]\.?m\.?</pattern>
</entity>
```

Eduction Grammar DTD

The XML DTD describing the Eduction grammar (such as, `edk.dtd`) is as follows:

```
<!ELEMENT grammars (include*, grammar*)>
<!ATTLIST grammars
version CDATA #IMPLIED
case (sensitive|insensitive|inherited) "inherited"
```

```
debug (true|false) "false"
>
<!ELEMENT include (publish*)>
<!ATTLIST include
path CDATA #REQUIRED
type (private|public) "public"
>
<!ELEMENT publish EMPTY>
<!ATTLIST publish
name CDATA #IMPLIED
>
<!ELEMENT grammar (extern*,entity+)>
<!ATTLIST grammar
name CDATA #REQUIRED
case (sensitive|insensitive|inherited) "inherited"
extend (append|replace|disallow) "disallow"
debug (true|false|inherited) "inherited"
>
<!ELEMENT extern EMPTY>
<!ATTLIST extern
name CDATA #REQUIRED
>
<!ELEMENT entity (entry*,pattern*)+>
<!ATTLIST entity
name CDATA #REQUIRED
type (private|public) "private"
case (sensitive|insensitive|inherited) "inherited"
extend (append|replace|disallow) "disallow"
debug (true|false|inherited) "inherited"
>
<!ELEMENT entry (headword?,synonym*)>
<!ATTLIST entry
headword CDATA #IMPLIED
score CDATA "1"
case (sensitive|insensitive|inherited) "inherited"
debug (true|false|inherited) "inherited"
>
<!ELEMENT headword (#PCDATA)>
<!ATTLIST headword
score CDATA "1"
case (sensitive|insensitive|inherited) "inherited"
>
<!ELEMENT synonym (#PCDATA)>
<!ATTLIST synonym
case (sensitive|insensitive|inherited) "inherited"
>
<!ELEMENT pattern (#PCDATA)>
<!ATTLIST pattern
score CDATA "1"
```

```
case (sensitive|insensitive|inherited) "inherited"
replace CDATA #IMPLIED
insert_before CDATA #IMPLIED
insert_after CDATA #IMPLIED
>
```

Appendix A: Eduction Lua Methods Reference

This section describes the methods you can use in your Lua post-processing scripts.

edkComponent Methods

The following methods are available on `edkComponent` objects.

You can obtain an `edkComponent` object using the getComponent method of an edkmatch object.

Method	Description
getName	Returns the name of a component.
getText	Returns the text that is matched by a component.
setName	Edits the name of a component.
setText	Edits the text that is matched by a component.

getName

Retrieves the name of a component.

Syntax

```
edkcomponent:getName()
```

Returns

The component name. You can use setName to edit the component name.

getText

Returns the output text that is matched by a particular component.

Syntax

```
edkcomponent:getText()
```

Returns

The matched text for a specified component. You can use setText to edit the text.

setName

Edits the name of the component that you retrieved with getName.

Syntax

```
edkcomponent:setName(new_name)
```

Arguments

Argument	Description
`new_name`	The new name for the component.

Returns

The new component name.

setText

Edits the matched text for a particular component that you retrieved with getText.

Syntax

```
edkcomponent:setText(new_text)
```

Arguments

Argument	Description
`new_text`	The new matched text for the component.

Returns

The new matched text for the specified component.

edkEnMasseMatch Methods

An `edkEnMasseMatch` object represents a match that is being processed in an en-masse post processing task. You can not manipulate the match directly, instead call the getMatch method to obtain an edkmatch object.

The following methods are available on `edkEnMasseMatch` objects.

Method	Description
getMatch	Returns an edkmatch object that represents the match.
getOutput	Returns whether the match is included in the results.
setOutput	Sets whether to include the match in the results.

getMatch

Returns an edkmatch object that you can use to manipulate the match.

Syntax

```
edkEnMasseMatch:getMatch()
```

Returns

An edkmatch object.

Example

The following example Lua script uses the `getMatch` method to obtain an `edkmatch` object:

```
function processmatches(matches)
    -- example that discards matches with score < 0.5
    for k,v in ipairs (matches) do
        local edkmatch = v:getMatch()
        if edkmatch:getScore() < 0.5 then
            v:setOutput(false)
        end
    end
end
```

getOutput

Returns whether the match is included in the results.

Syntax

```
edkEnMasseMatch:getOutput()
```

Returns

Boolean. `True` if the match is going to be included in the results, or `false` if it is going to be discarded.

setOutput

Sets whether to include the match in the results.

Syntax

```
edkEnMasseMatch:setOutput(output)
```

Arguments

Argument	Description
`output`	A Boolean value that specifies whether to include the match in the results (true) or discard it (false).

Example

The following script uses the `setOutput` method to discard matches with a score less than `0.5`:

```
function processmatches(matches)
    -- example that discards matches with score < 0.5
    for k,v in ipairs (matches) do
        local edkmatch = v:getMatch()
        if edkmatch:getScore() < 0.5 then
            v:setOutput(false)
        end
    end
end
```

edkMatch Methods

The following methods are available on `edkMatch` objects.

Method	Description
addComponent	Adds a new component to the match.
getComponent	Returns a specific component.
getComponentCount	Returns the total number of components in a match.
getContribution	Returns a specific scoring contribution.
getContributionsCount	Returns the number of scoring contributions, that contributed to the score for the match.
getEntityName	Returns the name of an entity in a match.
getMatchedText	Returns the input text for a match.
getOffset	Returns the position of a match (in bytes).
getOffsetLength	Returns the position of a match (in characters).
getOutputText	Returns the output text for a match.
getScore	Returns the score of a match.
setEntityName	Edits an entity name in a match.
setMatchedText	Edits the input text for a match.
setOffset	Edits the position of a match (in bytes).
setOffsetLength	Edits the position of a match (in bytes).
setOutputText	Assigns a new value to the output text for a match.
setScore	Edits the score of a match.

addComponent

Adds a new component to the match. For example, if your Eduction task returns an email address as a match, you can use `addComponent` to extract the text after the @ symbol and add it as a DOMAIN component for the match.

You can also use `addComponent` to add metadata from other sources. For example, if you have extracted a place name, you can add components called “LATITUDE” and “LONGITUDE”, and populate them with data from a different source, regardless of the fact that they were not components of the original text.

Syntax

```
edkmatch:addComponent(name, offset, offsetLength)
```

Arguments

Argument	Description
`name`	The name of the new component (for example, TOPIC, or SENTIMENT)
`offset`	The position of the text in the match to use as the new component (in bytes).
`offsetLength`	The position of the text in the match to use as the new component (in characters).

> **NOTE:**
> If you are unsure of the correct `offset` or `offsetLength`, you can specify `offset=0` or `offsetLength=0`.

Returns

The new empty component object.

Related Topics

- getName, on page 285
- setName, on page 285
- getText, on page 285
- setText, on page 286

getComponent

The `getComponent` method returns a specified component object. The components are zero-indexed. For example, if you have six components, you can get the last component by using `edkmatch:getComponent(5)`.

Syntax

```
edkmatch:getComponent(index)
```

Arguments

Argument	Description
`index`	The number of the component to get (where the first component has the index zero).

Returns

The component object at the specified index position in the match.

Related Topics

- getName, on page 285
- setName, on page 285
- getText, on page 285
- setText, on page 286

getComponentCount

Returns the total number of components in a match.

Syntax

```
edkmatch:getComponentCount()
```

Returns

The number of components.

getContribution

Returns a specified scoring contribution.

The final score for a match (as retrieved through the getScore method) can be the product of multiple scoring contributions. For some entities the score is then normalized (for example so that it is always a value between 0 and 1).

Syntax

```
edkmatch:getContribution(index)
```

Arguments

Argument	Description
`index`	The number of the contribution to get (where the first contribution has the index zero).

Returns

The scoring contribution.

Example

The following example demonstrates how to obtain scoring contributions:

```
function processmatch(edkmatch)
    local contributionsCount = edkmatch:getContributionsCount()
    print ("Contributions count: ", contributionsCount)
    if contributionsCount >= 1 then
        for i=0, contributionsCount-1, 1 do
            local contribution = edkmatch:getContribution(i)
            print ("Contribution " .. i .. ": " , contribution)
        end
    end
    print ("Score: " , edkmatch:getScore())
    return true
end
```

This script produces output similar to:

```
Contributions count:    4.0
Contribution 0:         1.0
Contribution 1:         0.75
Contribution 2:         1.0
Contribution 3:         0.6
Score:  0.45
```

getContributionsCount

Returns the number of scoring contributions, that contributed to the score for the match.

Syntax

```
edkmatch:getContributionsCount()
```

Returns

The number of scoring contributions.

Example

For an example that demonstrates how to obtain scoring contributions, see the example for the getContribution method.

getEntityName

Gets an entity name from a match.

Syntax

```
edkmatch:getEntityName()
```

Returns

The name of the entity in a match. You can use setEntityName to edit the name.

getMatchedText

Returns the input text for a particular match, that is, the text **before** any normalization or modification that occurs as part of the extraction process.

Syntax

```
edkmatch:getMatchedText()
```

Returns

The input text for a match. You can use setMatchedText to edit the text.

Related Topics

- getOutputText, on the next page

getOffset

Provides information on where in a document a particular match is found.

Syntax

```
edkmatch:getOffset()
```

Returns

The position of the match, in bytes. You can use setOffset to edit this information.

getOffsetLength

Provides information on where in a document a particular match is found.

Syntax

```
edkmatch:getOffsetLength()
```

Returns

The position of the match, in characters. You can use setOffsetLength to edit this information.

getOutputText

Returns the output text for a match, that is, the text **after** any normalization or modification that takes place as part of the extraction process.

Syntax

```
edkmatch:getOutputText()
```

Returns

The output text of a match. You can use setOutputText to edit the text.

Related Topics

- getMatchedText, on the previous page

getScore

Retrieves the score for a match.

Syntax

```
edkmatch:getScore()
```

Returns

The score for the match. You can use setScore to edit the score.

setEntityName

Edits the name of the entity that you retrieved by using getEntityName.

Syntax

```
edkmatch:setEntityName(new_name)
```

Arguments

Argument	Description
`new_name`	The new name for the entity in the match.

Returns

The new entity name.

setMatchedText

Edits the input text that you retrieved by using getMatchedText.

The input text is the text **before** any normalization or modification that takes place as part of the extraction process. By contrast, setOutputText enables you to edit the output text after any changes.

Syntax

```
edkmatch:setMatchedText(new_text)
```

Arguments

Argument	Description
`new_text`	The new value that you want to assign to the input text.

Returns

The new input text.

setOffset

Edits the position of a match in a document.

Syntax

```
edkmatch:setOffset(new_offset)
```

Arguments

Argument	Description
`new_offset`	The new position of the match (in bytes).

Returns

The new position of the match (in bytes).

Related Topics

- getOffset, on page 293

setOffsetLength

Edits the position of a match in a document.

Syntax

```
edkmatch:setOffsetLength(new_length)
```

Arguments

Argument	Description
`new_length`	The new position of the match (in characters).

Returns

The new position of the match (in characters).

setOutputText

Edits the output text that you retrieved by using getOutputText.

The output text is the text **after** any normalization or modification that takes place as part of the extraction process. By contrast, setMatchedText enables you to edit the input text before any changes are made.

Syntax

```
edkmatch:setOutputText(new_text)
```

Arguments

Argument	Description
`new_text`	The new value that you want to assign to the output text for a match.

Returns

The new output text.

setScore

Edits the match score that you retrieved with getScore.

Syntax

```
edkmatch:setScore(new_score)
```

Arguments

Argument	Description
`new_score`	The new score for the match.

Returns

The new score for the match.

Appendix B: Open Source and Third-Party Software License Agreements

Micro Focus acknowledges the redistribution of the following open source and third-party components under the licenses shown below.

Boost.org Boost C++ Libraries

```
Boost Software License - Version 1.0 - August 17th, 2003

Permission is hereby granted, free of charge, to any person or organization
obtaining a copy of the software and accompanying documentation covered by
this license (the "Software") to use, reproduce, display, distribute,
execute, and transmit the Software, and to prepare derivative works of the
Software, and to permit third-parties to whom the Software is furnished to
do so, all subject to the following:

The copyright notices in the Software and this entire statement, including
the above license grant, this restriction and the following disclaimer,
must be included in all copies of the Software, in whole or in part, and
all derivative works of the Software, unless such copies or derivative
works are solely in the form of machine-executable object code generated by
a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT
SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE
FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE,
ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS IN THE SOFTWARE.
```

CPAN.org GSSAPI

Perl5 is Copyright (C) 1993-2005, by Larry Wall and others.
It is free software; you can redistribute it and/or modify it under the terms of either:
a) the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version, or
b) the "Artistic License".
The Artistic License 2.0
Copyright (c) 2000-2006, The Perl Foundation.
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.
Preamble
This license establishes the terms under which a given free software Package may be copied, modified, distributed, and/or redistributed. The intent is that the Copyright Holder maintains some artistic control over the development of that Package while still keeping the Package available as open source and free software. You are always permitted to make arrangements wholly outside of this license directly with the Copyright Holder of a given Package. If the terms of this license do not permit the full use that you propose to make of the Package, you should contact the Copyright Holder and seek a different licensing arrangement.
Definitions
"Copyright Holder" means the individual(s) or organization(s) named in the copyright notice for the entire Package.
"Contributor" means any party that has contributed code or other material to the Package, in accordance with the Copyright Holder's procedures.
"You" and "your" means any person who would like to copy, distribute, or modify the Package.
"Package" means the collection of files distributed by the Copyright Holder, and derivatives of that collection and/or of those files. A given Package may consist of either the Standard Version, or a Modified Version.
"Distribute" means providing a copy of the Package or making it accessible to anyone else, or in the case of a company or organization, to others outside of your company or organization.
"Distributor Fee" means any fee that you charge for Distributing this Package or providing support for this Package to another party. It does not mean licensing fees.
"Standard Version" refers to the Package if it has not been modified, or has been modified only in ways explicitly requested by the Copyright Holder.
"Modified Version" means the Package, if it has been changed, and such changes were not explicitly requested by the Copyright Holder.
"Original License" means this Artistic License as Distributed with the Standard Version of the Package, in its current version or as it may be modified by The Perl Foundation in the future.
"Source" form means the source code, documentation source, and configuration files for the Package.
"Compiled" form means the compiled bytecode, object code, binary, or any other form

resulting from mechanical transformation or translation of the Source form.
Permission for Use and Modification Without Distribution
(1) You are permitted to use the Standard Version and create and use Modified Versions for any purpose without restriction, provided that you do not Distribute the Modified Version.

Permissions for Redistribution of the Standard Version
(2) You may Distribute verbatim copies of the Source form of the Standard Version of this Package in any medium without restriction, either gratis or for a Distributor Fee, provided that you duplicate all of the original copyright notices and associated disclaimers. At your discretion, such verbatim copies may or may not include a Compiled form of the Package.
(3) You may apply any bug fixes, portability changes, and other modifications made available from the Copyright Holder. The resulting Package will still be considered the Standard Version, and as such will be subject to the Original License.

Distribution of Modified Versions of the Package as Source
(4) You may Distribute your Modified Version as Source (either gratis or for a Distributor Fee, and with or without a Compiled form of the Modified Version) provided that you clearly document how it differs from the Standard Version, including, but not limited to, documenting any non-standard features, executables, or modules, and provided that you do at least ONE of the following:
(a) make the Modified Version available to the Copyright Holder of the Standard Version, under the Original License, so that the Copyright Holder may include your modifications in the Standard Version.
(b) ensure that installation of your Modified Version does not prevent the user installing or running the Standard Version. In addition, the Modified Version must bear a name that is different from the name of the Standard Version.
(c) allow anyone who receives a copy of the Modified Version to make the Source form of the Modified Version available to others under
(i) the Original License or
(ii) a license that permits the licensee to freely copy, modify and redistribute the Modified Version using the same licensing terms that apply to the copy that the licensee received, and requires that the Source form of the Modified Version, and of any works derived from it, be made freely available in that license fees are prohibited but Distributor Fees are allowed.

Distribution of Compiled Forms of the Standard Version or Modified Versions without the Source
(5) You may Distribute Compiled forms of the Standard Version without the Source, provided that you include complete instructions on how to get the Source of the Standard Version. Such instructions must be valid at the time of your distribution. If these instructions, at any time while you are carrying out such distribution, become invalid, you must provide new instructions on demand or cease further distribution. If you provide valid instructions or cease distribution within thirty days after you become aware that the instructions are invalid, then you do not forfeit any of your rights under this license.
(6) You may Distribute a Modified Version in Compiled form without the Source, provided that you comply with Section 4 with respect to the Source of the Modified

Version.

Aggregating or Linking the Package
(7) You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.
(8) You are permitted to link Modified and Standard Versions with other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

Items That are Not Considered Part of a Modified Version
(9) Works (including, but not limited to, modules and scripts) that merely extend or make use of the Package, do not, by themselves, cause the Package to be a Modified Version. In addition, such works are not considered parts of the Package itself, and are not subject to the terms of this license.

General Provisions
(10) Any use, modification, and distribution of the Standard or Modified Versions is governed by this Artistic License. By using, modifying or distributing the Package, you accept this license. Do not use, modify, or distribute the Package, if you do not accept this license.
(11) If your Modified Version has been derived from a Modified Version made by someone other than you, you are nevertheless required to ensure that your Modified Version complies with the requirements of this license.
(12) This license does not grant you the right to use any trademark, service mark, tradename, or logo of the Copyright Holder.
(13) This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package. If you institute patent litigation (including a cross-claim or counterclaim) against any party alleging that the Package constitutes direct or contributory patent infringement, then this Artistic License to you shall terminate on the date that such litigation is filed.
(14) Disclaimer of Warranty:
THE PACKAGE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS "AS IS' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES. THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT ARE DISCLAIMED TO THE EXTENT PERMITTED BY YOUR LOCAL LAW. UNLESS REQUIRED BY LAW, NO COPYRIGHT HOLDER OR CONTRIBUTOR WILL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING IN ANY WAY OUT OF THE USE OF THE PACKAGE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Github FST

James Clark Expat XML Parser

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Lua

MIT License
Copyright (c) <year> <copyright holders>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

MIT Kerberos Consortium Kerberos

MIT Kerberos License information
Copyright 1985-2016 by the Massachusetts Institute of Technology.
All rights reserved.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT

NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
Downloading of this software may constitute an export of cryptographic software from the United States of America that is subject to the United States Export Administration Regulations (EAR), 15 CFR 730-774. Additional laws or regulations may apply. It is the responsibility of the person or entity contemplating export to comply with all applicable export laws and regulations, including obtaining any required license from the U.S. government.
The U.S. government prohibits export of encryption source code to certain countries and individuals, including, but not limited to, the countries of Cuba, Iran, North Korea, Sudan, Syria, and residents and nationals of those countries.
Documentation components of this software distribution are licensed under a Creative Commons Attribution-ShareAlike 3.0 Unported License. (http://creativecommons.org/licenses/by-sa/3.0/)
Individual source code files are copyright MIT, Cygnus Support, Novell, OpenVision Technologies, Oracle, Red Hat, Sun Microsystems, FundsXpress, and others.
Project Athena, Athena, Athena MUSE, Discuss, Hesiod, Kerberos, Moira, and Zephyr are trademarks of the Massachusetts Institute of Technology (MIT). No commercial use of these trademarks may be made without prior written permission of MIT. Commercial use means use of a name in a product or other for-profit manner. It does NOT prevent a commercial firm from referring to the MIT trademarks in order to convey information (although in doing so, recognition of their trademark status should be given).

The following copyright and permission notice applies to the OpenVision Kerberos Administration system located in kadmin/create, kadmin/dbutil, kadmin/passwd, kadmin/server, lib/kadm5, and portions of lib/rpc:
Copyright, OpenVision Technologies, Inc., 1993-1996, All Rights Reserved
WARNING: Retrieving the OpenVision Kerberos Administration system source code, as described below, indicates your acceptance of the following terms. If you do not agree to the following terms, do not retrieve the OpenVision Kerberos administration system.
You may freely use and distribute the Source Code and Object Code compiled from it, with or without modification, but this Source Code is provided to you AS IS EXCLUSIVE OF ANY WARRANTY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR ANY OTHER WARRANTY, WHETHER EXPRESS OR IMPLIED. IN NO EVENT WILL OPENVISION HAVE ANY LIABILITY FOR ANY LOST PROFITS, LOSS OF DATA OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, OR FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS AGREEMENT, INCLUDING, WITHOUT LIMITATION, THOSE RESULTING FROM THE USE OF THE SOURCE CODE, OR THE FAILURE OF THE SOURCE CODE TO PERFORM, OR FOR ANY OTHER REASON.
OpenVision retains all copyrights in the donated Source Code. OpenVision also retains copyright to derivative works of the Source Code, whether created by OpenVision or by a third party. The OpenVision copyright notice must be preserved if derivative works are made based on the donated Source Code.
OpenVision Technologies, Inc. has donated this Kerberos Administration system to

MIT for inclusion in the standard Kerberos 5 distribution. This donation underscores our commitment to continuing Kerberos technology development and our gratitude for the valuable work which has been performed by MIT and the Kerberos community.

Portions contributed by Matt Crawford crawdad@fnal.gov were work performed at Fermi National Accelerator Laboratory, which is operated by Universities Research Association, Inc., under contract DE-AC02-76CHO3000 with the U.S. Department of Energy.

Portions of src/lib/crypto have the following copyright:
Copyright 1998 by the FundsXpress, INC.
All rights reserved.
Export of this software from the United States of America may require a specific license from the United States Government. It is the responsibility of any person or organization contemplating export to obtain such a license before exporting.
WITHIN THAT CONSTRAINT, permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of FundsXpress. not be used in advertising or publicity pertaining to distribution of the software without specific, written prior permission.
FundsXpress makes no representations about the suitability of this software for any purpose. It is provided as is without express or implied warranty.
THIS SOFTWARE IS PROVIDED AS IS AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The implementation of the AES encryption algorithm in src/lib/crypto/builtin/aes has the following copyright:
Copyright 2001, Dr Brian Gladman brg@gladman.uk.net, Worcester, UK.
All rights reserved.
LICENSE TERMS
The free distribution and use of this software in both source and binary form is allowed (with or without changes) provided that:
1. distributions of this source code include the above copyright notice, this list of conditions and the following disclaimer;
2. distributions in binary form include the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other associated materials;
3. the copyright holders name is not used to endorse products built using this software without specific written permission.
DISCLAIMER
This software is provided as is with no explcit or implied warranties in respect of any properties, including, but not limited to, correctness and fitness for purpose.

Portions contributed by Red Hat, including the pre-authentication plug-in framework and the NSS crypto implementation, contain the following copyright:
Copyright 2006 Red Hat, Inc.

```
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS AS IS AND ANY
EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
POSSIBILITY OF SUCH DAMAGE.

The implementations of GSSAPI mechglue in GSSAPI-SPNEGO in src/lib/gssapi,
including the following files:
lib/gssapi/generic/gssapi_err_generic.et
lib/gssapi/mechglue/g_accept_sec_context.c
lib/gssapi/mechglue/g_acquire_cred.c
lib/gssapi/mechglue/g_canon_name.c
lib/gssapi/mechglue/g_compare_name.c
lib/gssapi/mechglue/g_context_time.c
lib/gssapi/mechglue/g_delete_sec_context.c
lib/gssapi/mechglue/g_dsp_name.c
lib/gssapi/mechglue/g_dsp_status.c
lib/gssapi/mechglue/g_dup_name.c
lib/gssapi/mechglue/g_exp_sec_context.c
lib/gssapi/mechglue/g_export_name.c
lib/gssapi/mechglue/g_glue.c
lib/gssapi/mechglue/g_imp_name.c
lib/gssapi/mechglue/g_imp_sec_context.c
lib/gssapi/mechglue/g_init_sec_context.c
lib/gssapi/mechglue/g_initialize.c
lib/gssapi/mechglue/g_inquire_context.c
lib/gssapi/mechglue/g_inquire_cred.c
lib/gssapi/mechglue/g_inquire_names.c
lib/gssapi/mechglue/g_process_context.c
lib/gssapi/mechglue/g_rel_buffer.c
lib/gssapi/mechglue/g_rel_cred.c
lib/gssapi/mechglue/g_rel_name.c
lib/gssapi/mechglue/g_rel_oid_set.c
lib/gssapi/mechglue/g_seal.c
lib/gssapi/mechglue/g_sign.c
lib/gssapi/mechglue/g_store_cred.c
lib/gssapi/mechglue/g_unseal.c
lib/gssapi/mechglue/g_userok.c
lib/gssapi/mechglue/g_utils.c
lib/gssapi/mechglue/g_verify.c
lib/gssapi/mechglue/gssd_pname_to_uid.c
lib/gssapi/mechglue/mglueP.h
lib/gssapi/mechglue/oid_ops.c
lib/gssapi/spnego/gssapiP_spnego.h
```

PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COMPUTING RESEARCH LAB OR NEW MEXICO STATE UNIVERSITY BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The following notice applies to src/util/support/strlcpy.c:
Copyright 1998 Todd C. Miller Todd.Miller@courtesan.com
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.
THE SOFTWARE IS PROVIDED AS IS AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

The following notice applies to src/util/profile/argv_parse.c and src/util/profile/argv_parse.h:
Copyright 1999 by Theodore Tso.
Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED AS IS AND THEODORE TSO (THE AUTHOR) DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE. (Isnt it sick that the U.S. culture of lawsuit-happy lawyers requires this kind of disclaimer?)

The following notice applies to SWIG-generated code in src/util/profile/profile_tcl.c:
Copyright 1999-2000, The University of Chicago
This file may be freely redistributed without license or fee provided this copyright message remains intact.

The following notice applies to portiions of src/lib/rpc and src/include/gssrpc:
Copyright 2000 The Regents of the University of Michigan. All rights reserved.
Copyright 2000 Dug Song dugsong@UMICH.EDU. All rights reserved, all wrongs reversed.
Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. Neither the name of the University nor the names of its contributors may be used

to endorse or promote products derived from this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED AS IS AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Implementations of the MD4 algorithm are subject to the following notice:
Copyright 1990, RSA Data Security, Inc. All rights reserved.
License to copy and use this software is granted provided that it is identified as the RSA Data Security, Inc. MD4 Message Digest Algorithm in all material mentioning or referencing this software or this function.
License is also granted to make and use derivative works provided that such works are identified as derived from the RSA Data Security, Inc. MD4 Message Digest Algorithm in all material mentioning or referencing the derived work.
RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided as is without express or implied warranty of any kind.
These notices must be retained in any copies of any part of this documentation and/or software.

Implementations of the MD5 algorithm are subject to the following notice:
Copyright 1990, RSA Data Security, Inc. All rights reserved.
License to copy and use this software is granted provided that it is identified as the RSA Data Security, Inc. MD5 Message- Digest Algorithm in all material mentioning or referencing this software or this function.
License is also granted to make and use derivative works provided that such works are identified as derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm in all material mentioning or referencing the derived work.
RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided as is without express or implied warranty of any kind.
These notices must be retained in any copies of any part of this documentation and/or software.

The following notice applies to src/lib/crypto/crypto_tests/t_mddriver.c:
Copyright 1990-2, RSA Data Security, Inc. Created 1990. All rights reserved.
RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided as is without express or implied warranty of any kind.
These notices must be retained in any copies of any part of this documentation

XMLsoft.org libxml

Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform), it is free software available under the MIT License. XML itself is a metalanguage to design markup languages, i.e. text language where semantic and structure are added to the content using extra "markup" information enclosed between angle brackets. HTML is the most well-known markup language. Though the library is written in C a variety of language bindings make it available in other environments.

Libxslt is the XSLT C library developed for the GNOME project. XSLT itself is a an XML language to define transformation for XML. Libxslt is based on libxml2 the XML C library developed for the GNOME project. It also implements most of the EXSLT set of processor-portable extensions functions and some of Saxon's evaluate and expressions extensions.
People can either embed the library in their application or use xsltproc the command line processing tool. This library is free software and can be reused in commercial applications (see the intro)
Daniel Veillard
MIT License follows:

MIT License
Copyright (c) <year> <copyright holders>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

XMLsoft.org libxslt

Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform), it is free software available under the MIT License. XML itself is a metalanguage to design markup languages, i.e. text language where semantic and structure are added to the content using extra "markup" information enclosed between angle brackets. HTML is the most well-known markup

language. Though the library is written in C a variety of language bindings make it available in other environments.

Libxslt is the XSLT C library developed for the GNOME project. XSLT itself is a an XML language to define transformation for XML. Libxslt is based on libxml2 the XML C library developed for the GNOME project. It also implements most of the EXSLT set of processor-portable extensions functions and some of Saxon's evaluate and expressions extensions.
People can either embed the library in their application or use xsltproc the command line processing tool. This library is free software and can be reused in commercial applications (see the intro)
Daniel Veillard
MIT License follows:

MIT License
Copyright (c) <year> <copyright holders>
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
THE SOFTWARE.

Zlib.net zlib

License
/* zlib.h -- interface of the 'zlib' general purpose compression library
  version 1.2.2, October 3rd, 2004

  Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler

  This software is provided 'as-is', without any express or implied
  warranty. In no event will the authors be held liable for any damages
  arising from the use of this software.

  Permission is granted to anyone to use this software for any purpose,
  including commercial applications, and to alter it and redistribute it
  freely, subject to the following restrictions:

```
   1. The origin of this software must not be misrepresented; you must not
      claim that you wrote the original software. If you use this software
      in a product, an acknowledgment in the product documentation would be
      appreciated but is not required.
   2. Altered source versions must be plainly marked as such, and must not be
      misrepresented as being the original software.
   3. This notice may not be removed or altered from any source distribution.

   Jean-loup Gailly jloup@gzip.org
   Mark Adler madler@alumni.caltech.edu

*/
```

Glossary

A

ACI (Autonomy Content Infrastructure)
A technology layer that automates operations on unstructured information for cross-enterprise applications. ACI enables an automated and compatible business-to-business, peer-to-peer infrastructure. The ACI allows enterprise applications to understand and process content that exists in unstructured formats, such as email, Web pages, Microsoft Office documents, and IBM Notes.

ACI Server
A server component that runs on the Autonomy Content Infrastructure (ACI).

ACL (access control list)
An ACL is metadata associated with a document that defines which users and groups are permitted to access the document.

action
A request sent to an ACI server.

active directory
A domain controller for the Microsoft Windows operating system, which uses LDAP to authenticate users and computers on a network.

C

Category component
The IDOL Server component that manages categorization and clustering.

chunking
The process of dividing a sentence into a sequence of non-overlapping text regions, or chunks. See also: shallow parsing.

Community component
The IDOL Server component that manages users and communities.

compiled grammar
A grammar file that has been compiled from XML into ECR file format using the Eduction command-line tool edktool, so that Eduction can use it directly. See also: XML, ECR file, grammar, standard grammar, user grammar.

connector
An IDOL component (for example File System Connector) that retrieves information from a local or remote repository (for example, a file system, database, or Web site).

Connector Framework Server (CFS)
Connector Framework Server processes the information that is retrieved by connectors. Connector Framework Server uses KeyView to extract document content and metadata from over 1,000 different file types. When the information has been processed, it is sent to an IDOL Server or Distributed Index Handler (DIH).

Content component
The IDOL Server component that manages the data index and performs most of the search and retrieval operations from the index.

D

DAH (Distributed Action Handler)
DAH distributes actions to multiple copies of IDOL Server or a component. It allows

you to use failover, load balancing, or distributed content.

database
An IDOL server data pool that stores indexed information. The administrator can set up one or more databases, and specifies how data is fed to the databases. By default IDOL server contains the databases Profile, Agent, Activated, Deactivated, News and Archive.

dictionary
An XML file that provides a vocabulary for an entity. Eduction uses the dictionary to scan a document and extract the defined entities that match the search pattern. See also: XML, entity, extraction.

DIH (Distributed Index Handler)
DIH allows you to efficiently split and index extremely large quantities of data into multiple copies of IDOL Server or the Content component. DIH allows you to create a scalable solution that delivers high performance and high availability. It provides a flexible way to batch, route, and categorize the indexing of internal and external content into IDOL Server.

E

ECR file
ECR is a proprietary format for grammar files that Eduction can easily read at runtime. You can write grammar files in XML, then use the Eduction command-line tool edktool to compile them into ECR format. See also: XML, compiled grammar.

Eduction
The process of extracting entities (patterns of text) from documents.

entity
In Eduction, an entity is a word, phrase, or block of information that the Eduction component can match and extract from documents. An entity can be a specific text string, such as a name, or it can be a pattern of text such as an address or phone number. You define the pattern in a grammar, which Eduction uses to find the entities in documents.

extraction
Eduction extracts entities from documents based on the rules you have created in your dictionaries and grammars, and returns an XML list of matches, or adds the matches to the source document as new fields. See also: XML, grammar, dictionary.

F

field
Fields define different parts of content in IDOL documents, such as the title, content, and metadata information.

G

grammar
In Eduction, a grammar is a pattern that defines an entity.

I

IDOL
The Intelligent Data Operating Layer (IDOL) Server, which integrates unstructured, semi-structured and structured information from multiple repositories through an understanding of the content. It delivers a real-time environment in which operations across applications and content are automated.

IDOL Proxy component
An IDOL Server component that accepts incoming actions and distributes them to the appropriate subcomponent. IDOL Proxy also performs some maintenance operations to make sure that the subcomponents are running, and to start and stop them when necessary.

IDOL server
The Micro Focus Intelligent Data Operating Layer (IDOL) server, which integrates unstructured, semi-structured and structured information from multiple repositories through an understanding of the content, delivering a real time environment in which operations across applications and content are automated, removing all the manual processes involved in getting the right information to the right people at the right time.

IDX
A structured file format that can be indexed into IDOL server. You can use a connector to import files into this format or you can manually create IDX files.

importing
After a document has been downloaded from the repository in which it is stored, it is imported to an IDX or XML file format. This process is called “importing”.

index
The IDOL server data index contains document content and field information for analysis and retrieval.

indexing
The process of storing data in IDOL server. IDOL server stores data in different field types (such as, index, numeric and ordinary fields). It is important to store data in appropriate field types to ensure optimized performance.

Intellectual Asset Protection System (IAS)
An integrated security solution to protect your data. At the front end, authentication checks that users are allowed to access the system that contains the result data. At the back end, entitlement checking and authentication combine to ensure that query results contain only documents that the user is allowed to see, from repositories that the user has permission to access. For more information, refer to the IDOL Document Security Administration Guide.

K

KeyView
The IDOL component that extracts data, including text, metadata, and subfiles from over 1,000 different file types. KeyView can also convert documents to HTML format for viewing in a Web browser.

L

LDAP
Lightweight Directory Access Protocol. Applications can use LDAP to retrieve information from a server. LDAP is used for directory services (such as corporate email and telephone directories) and user authentication. See also: active directory, primary domain controller.

License Server
License Server enables you to license and run multiple IDOL solutions. You must have a License Server on a machine with a known, static IP address.

Lua
An embedded scripting language that you can use to write custom scripts to expand certain IDOL functionality.

Luhn algorithm
A formula used to validate identification numbers, such as credit card numbers and social security numbers. The formula checks for errors by performing mathematical operations in the number to calculate a number that must agree with the final digit of the number.

M

metadata
Data that describes and gives information about other data. For example, the metadata for a text document might include information about the author of the document, the date it was written, or a short summary.

O

OmniGroupServer (OGS)
A server that manages access permissions for your users. It communicates with your repositories and IDOL Server to apply access permissions to documents.

P

parsing
The process of analyzing text according to the rules of a formal grammar.

pattern
A pattern is a description of the entity you want to extract, that enables Eduction to produce a list of matches based on that pattern. A pattern can explicitly list what Eduction should look for (for example, a list of names), or can specify in general terms what a match should look like (for example, phone numbers). See also: entity, extraction, grammar.

polarity scoring
A number, usually between 0.50 and 1.50, that represents the strength of the sentiment in the matched phrase.

precision
Precision is the percentage of extracted entities that are true entities. See also: recall.

primary domain controller
A server computer in a Microsoft Windows domain that controls various computer resources. See also: active directory, LDAP.

R

recall
The recall of an extraction is the percentage of matches that are actually returned, out of the total number of matches that should return in theory. See also: precision.

regular expressions
A string that allows you to define a particular string pattern in a concise format. In IDOL server, matching in Eduction and Connectors uses regular expressions to define what you want to match.

relevance
The similarity that a particular query result has to the initial query. IDOL server assigns results a percentage relevance score according to how closely it matches the query criteria.

S

sentiment analysis
A form of Eduction that identifies positive and negative sentiment in text.

shallow parsing
A form of sentence analysis that identifies the constituent parts of the sentence, such as noun phrases, but not their structure or their role in the sentence. See also: chunking.

standard grammar
Eduction includes a set of standard grammars that allow you to extract the most common entities, such as person, place, or company names, legal terms, addresses, dates, and times. See also: entity, compiled grammar, grammar, user grammar.

T

tagging
The process of adding extra information to documents. The tag might be a category, or entities returned from Eduction. Tagging usually adds a field to a document, which you can use to search by the name of a tag.

tokens
IDOL Server stores document text as a series of tokens. Generally, a token is a word, but it can also include other strings of characters (such as a phone number or e-mail address).

U

user grammar
XML files created by the user that describe entities that can locate patterns in text using the Eduction grammar language.

V

View
An IDOL component that converts files in a repository to HTML formats for viewing in a Web browser.

W

Wildcard
A character that stands in for any character or group of characters in a query.

X

XML
Extensible Markup Language. XML is a language that defines the different attributes of document content in a format that can be read by humans and machines. In IDOL Server, you can index documents in XML format. IDOL Server also returns action responses in XML format.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Programming Guide (Micro Focus Eduction 12.3)

Add your feedback to the email and click **Send**.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to swpdl.idoldocsfeedback@microfocus.com.

We appreciate your feedback!