

Eduction

Software Version 12.3.0

Release Notes



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

Legal notices

Copyright notice

© Copyright 2019 Micro Focus or one of its affiliates.

The only warranties for products and services of Micro Focus and its affiliates and licensors (“Micro Focus”) are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without 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](#).

Contents

New in this Release	4
Resolved Issues	7
Requirements	8
Minimum System Requirements	8
Software Dependencies	8
Supported Operating System Platforms	8
Documentation	9

New in this Release

The following new features were released in Eduction version 12.3.0.

Enhancements to Eduction Server and Eduction SDK

- Eduction includes a new grammar file, `monetary_value.ecr`, for extracting monetary values such as *\$5 billion*, *£15,275,486*, or *25 cents*.
- The grammar file `number_driverlic_ca.ecr`, which contains entities for matching Canadian driver's license numbers, supports numbers from Newfoundland and Labrador (`number/driverlic/NL/ca`) and Nunavut (`number/driverlic/NU/ca`), and new formats for Manitoba (MB), New Brunswick (NB), Northwest Territories (NT), Ontario (ON), and Quebec (QC).
- Processing input that contains Chinese, Japanese, and Korean (CJK) characters is much faster when none of the entities you configure can begin with a CJK character. To benefit from this performance improvement, use the standard grammar files that are included with Eduction 12.3.0 or later, and re-compile any custom grammar files using `edktool` version 12.3.0 or later.
- The `datetime_advanced_eng.ecr` grammar (and the associated `datetime.lua` postprocessing script) have been extended to match and process date and time ranges. For example, it can now detect dates and ranges such as *the 16th to 20th October*, *November 4-8*, *29 Nov-2 Dec 2019*, *29/11/19-02/12/19*, *10:30-11:30*, *ten 'til ten thirty*, and *16th Oct 10-10:30*.

In some cases, this grammar can also match multiple dates as a single entity, where educating separately would lose information. For example, *5th and 6th July* means *5th July* and *6th July*.

- The `datetime_advanced_eng.ecr` grammar has been extended to match the additional formats *year ending*, *quarter ending*, or *month ending*, and similar (*up to*, *ended*).

The year or quarter can be followed by a month-precision date (for example, *year ending September 2019*). The year is taken to end on the last day of the specified month. This match produces an inclusive date range, for example *2018-10-01/2019-09-30*.

It can also match *last day of* or *end of* a specified month or year.

There is also a fuller range of month-only patterns, grouped under the `datetime/advanced_month_only/eng` entity. If Eduction matches a month-only entity, it returns the date in YYYY-MM format.

- Case normalization did not handle Turkic dotted and dotless "i" characters correctly. To use Eduction with Turkic languages, set the new configuration parameter `CaseNormalizationBehavior=Turkic`.
- The `organization.ecr` grammar has been updated to map variants of an organization name as synonyms, rather than separate headwords.
- Scoring has been added to the `number/swiftcode` entity in the `number_swiftcode.ecr` grammar. Potential SWIFT codes that are (ALL CAPS) dictionary words are scored lower. Codes from "tier 1" financial countries are scored higher.

Score adjustment is implemented in Lua post-processing, so you must configure a post-processing task for swiftcode entity matches using the provided script `scripts/swiftcode_stoplist.lua` to enable this feature.

Enhancements to Education Server

- Education Server supports the configuration parameter `EntityMinScoreN`.
- Communications can be secured with TLS version 1.3.

Enhancements to Education SDK

- Education can perform case normalization (converting input text to upper or lower case). The following functions have been added:
 - `EdkSetCaseNormalization` and `EdkSetCaseNormalizationBehavior` in the C API.
 - `setCaseNormalization` and `setCaseNormalizationBehavior` in the Java API.
- Several improvements have been made to the processing of inclusions (when Education grammar files are compiled).
 - Compilation no longer fails when a grammar file is included more than once (for example when you compile grammar file A which includes files B and C, and both B and C include grammar file D).
 - When a grammar file is included privately, all entities added as a result of that include become private, even if they were defined as public entities.
 - If a grammar file is included privately in XML format but then publicly in ECR format, the entities are included publicly.
- The matches returned by the API now reflect any modifications made by post-processing. This makes it easier to include post-processing in your application.
- In the C API, the `EdkFreeBuffer(void)` and `EdkFreeStringArray(char***, int*)` utility functions have been added to allow you to safely free buffers and arrays.
- In the C API, the `EdkNormalizeCJKText` and `EdkRedactText` functions have been added, which use non-const `char*` variables to return string buffers that need to be freed. These functions replace the functions `EdkCJKNormalizeText` and `EdkGetRedactedText`. The older functions are still available but are deprecated. They might be deleted in future.
- The Java API has been updated to improve usability:
 - There are new interfaces for engine, session, and input stream. The `EDKEngine`, `EDKSession`, and `EDKJNIInputStream` classes have been updated to implement them.
 - The `EDKEngine` and `EDKSession` classes now implement `AutoCloseable` so that you can use them in `try-with-resources` blocks.
 - Engine enums for `CaseNormalization`, `CaseNormalizationBehaviour`, and `AllowMultipleResults`, have been moved into the interface.
 - The following new functions have been added, to specify multiple grammar files or entities, and to make it easier to enable or disable various settings. In addition, you can now create a session by calling an engine function.

```
public void enableComponents();
public void disableComponents();
public void enableUniqueMatches();
public void disableUniqueMatches();
public void enableCaseSensitiveFieldName();
public void disableCaseSensitiveFieldName();
public void enableMatchWholeWord();
public void disableMatchWholeWord();
public void enableTokenWithPunctuation();
public void disableTokenWithPunctuation();
public void enableAllowOverlaps();
public void disableAllowOverlaps();
public void enableNonGreedyMatch();
public void disableNonGreedyMatch();
public void enableMatchCase();
public void disableMatchCase();
public void loadResourceFiles(final Iterable<String> resourceFiles) throws
Exception;
public void addTargetEntities(final Iterable<String> entities) throws
Exception;
public TextExtractionSession<T> createSession();
public TextExtractionSession<T> createTimedSession(long start);
```

- You can now create an education engine from a configuration file, which automatically respects the settings in the configuration file. Similarly, you can now create a postprocessor from an education engine, which reflects any task configurations in the engine settings.
 - In the C API, use the new functions `EdkEngineCreateFromConfigFile` and `EdkPostProcessorCreateFromSettings`.
 - In the Java API, use the new constructors for `EDKEngine` and `EDKPostProcessor`.
- Normalization of Chinese, Japanese, and Korean (CJK) text is now performed inline. You can turn on CJK normalization by using the `CJKNormalization` parameter in the configuration file, or by calling the new `EdkEngineSetCJKNormalization` function with a comma-separated list of settings.

The `normalize-CJK` buffer functions are no longer required, and have been marked as deprecated. These functions might be removed in a future release.

Resolved Issues

The following issues were resolved in Education version 12.3.0.

- Some matches could be missed if they were followed only by whitespace.
- Timeouts specified by the configuration parameters `MatchTimeout` and `RequestTimeout` were not always respected by Education.
- An issue with case-insensitive matching could result in some matches being missed.
- The `numbers/ibands` entity in `number_iban.ecr` failed to match Austrian IBANs.

Resolved Issues for Education Server

- Education Server could stop unexpectedly after receiving multiple actions with any of the parameters `TangibleCharacters`, `RedactionOutputString`, or `RedactionReplacementCharacter` set.

Resolved Issues for Education SDK

- The Java EDK could leak memory during calls to CJK normalization.
- `Edktool` could exit unexpectedly if it tried to compile an XML file to produce a grammar in a nonexistent directory.
- In some cases, `EDKJNIInputStream` objects that were passed into `EDKSession.setInputStream` were never released, and memory that was allocated to track the stream object was not freed.

Requirements

This section describes the system requirements, supported platforms, and software dependencies for Education 12.3.0.

Minimum System Requirements

The following are minimum system requirements for Education 12.3.0.

- 512 MB RAM
- 1 GB disk space
- 800 MHz or higher processor
- 400 MB of space designated for `/tmp` or `/var/tmp` (UNIX), or user-defined temporary directory (Windows)

Software Dependencies

Java

Development with the Education Java SDK requires the Java 6.0 JDK and JRE.

Microsoft Visual C++ Redistributable Package

Windows versions of Education require installation of the Microsoft Visual C++ 2017 redistributable package.

Supported Operating System Platforms

The following operating system platforms are supported by Education 12.3.0.

- Windows x86 64
- Linux x86 64
- Solaris x86 64
- Solaris SPARC 64

Documentation

The following documentation was updated for this release.

- *Education SDK Programming Guide*
- *Education User Guide*
- *Education Server Reference*