

# IDOL Education

Software Version 12.13.0

## Release Notes



Document Release Date: October 2022  
Software Release Date: October 2022

## Legal notices

© Copyright 2022 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 as may be 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.

To check for updated documentation, visit <https://www.microfocus.com/support-and-services/documentation/>.

## 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:

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

Many areas of the portal require you to sign in. If you need an account, you can create one when prompted to sign in.

# Contents

New in this Release .....	4
Resolved Issues .....	8
Notes .....	9
Requirements .....	11
Minimum System Requirements .....	11
Software Dependencies .....	11
Supported Operating System Platforms .....	12
Documentation .....	13

## New in this Release

The following new features were released in Education version 12.13.0.

- The edktool command-line tool now has a configuration validation mode, `validate`. For information about the available options, refer to the *Education User and Programming Guide*.
- The edktool command-line tool benchmark mode now reports engine initialization time.
- In table mode, you can now configure Education to check the first N rows of a table until it finds at least one configured header match. Set the new `MaxSearchHeaderRow` configuration parameter to the number of rows that Education must search. Education searches for up to this number of rows, until it finds a header match. For example, if you set `MaxSearchHeaderRow` to 5, and Education finds a match on the third row, Education stops searching for header matches in further rows.

The default value is 1 (that is, it searches only the first row). You can also set `MaxSearchHeaderRow` to -1 to search the whole table to find a header match.

- In table mode, you can now configure entity match limits for the header and cell entities. The `HeaderEntityMatchLimitN` configuration parameter sets the maximum number of columns that Education can match against the header entities. The `CellEntityMatchLimitN` configuration parameter sets the maximum number of matches of the cell entity that Education returns from the body of the table (across all columns if there were multiple matches for the corresponding header entity). These limits apply only to values in a single table, and are reset if Education detects a table delimiter.
- In table mode, Education can now process multiple worksheets from a single text stream, when they are separated by the correct table delimiters. The distinct delimiters are:
  - Start table: "<blank line><tab line>" (`\n\n\t\n`)
  - End table: "<tab line><blank line>" (`\n\t\n\n`)

These delimiters reset the header matches, so that Education looks for a new header row.

**NOTE:** To retain existing behavior, the first table does not need a start delimiter. If there is text outside table delimiters, Education currently treats it as a new separate table.

You can configure `KeyView` to give the correct delimiters format when extracting tables, by:

- Setting the target character set to UTF-8.
- Enabling the Tab Delimited option.
- Enabling the Output Table Delimiters option.

For more information, refer to the *KeyView Filter SDK Programming Guide*.

- Compilation times for grammars that include repetitions of standard classes (such as `\w`) and Unicode character classes (such as `\pN`) are now vastly improved, following internal optimizations.

## Enhancements to Education Grammars

- For the `pii/address` grammar, matching has been improved for Canadian addresses that contain a fraction, for example *225 1/2 Front Street, Toronto*.
- The `pii/passport` grammar now supports Indian passport numbers. Landmarks are supported in English and Hindi.
- For PII post-processing, you can now disable the simple validity checks that eliminate likely false positive ID numbers, if you feel that these checks are too strict. To disable these checks, set the `IDOL_PII_DISABLE_BASIC_VALIDITY_CHECKS` environment variable to **TRUE**.
- The `pii/date` grammar has been updated to match the ISO-8601 date formats that KeyView FilterSDK outputs when it extracts dates from an Excel spreadsheet with the `StandardizeCellFormats` configuration option turned on (that is `YYYY-MM-DDTHH:MM:SS`, for example `1972-12-25T00:00.000`).
- The `pii/date` grammar has been updated so that `nocontext` date matches in the format `mm/dd` have a score higher than the 0.4 threshold.

**NOTE:** The score has been increased for `nocontext` matches for both `dd/mm` and `mm/dd`.

- The PII address grammars precision has been improved, by adding some low population Romanian villages to the stop list.
- The PII nationality grammar context entity precision has been improved:
  - *US* is now a case-sensitive match (that is, it cannot match *us* any more).
  - *Ísland* is now a case- and accent-sensitive match (that is, it cannot match *island* any more)
- English language PII landmarks have been improved where the qualifier *number* was included. For example, *number* has been removed from the landmarks for *social security number*, *bank account number*, and *passport number*.
- The PII `name_cjkvt.ecr` grammar has new `nocontext` `post_title` and `title_surname` entities for China, Korea, and Taiwan.
- The way that name matches interact with stopwords has been improved. Names with leading or trailing stopwords are now returned as matches, as long as the trimmed name is still a valid name. The match object returned is adjusted accordingly.
- Matching of standalone landmark entities in the PII address grammar has been improved.
- The score has been increased for IBAN `nocontext` entities in the PII banking.ecr grammar.
- Landmark matching has been improved in the `phi/account` banking grammar.
- Scoring has been improved for the PHI `location.ecr` grammar `phi/location/us` entity:
  - The score has been increased for cases where the state is retrieved, for example *Abbeville in the state of Alabama*.
  - The score has been decreased for cities without state information, for example *Edison*.
  - The score has been increased for city and postcode.

- The score has been increased for well-known US cities that are often used without state information, for example *Baltimore*.
- Matching of HIBCC identifiers by the `phi/device/udi/nocontext` entity has been improved.
- The PCI PAN entity `pci/pan` now supports the following additional types of credit card: `China_T_Union`, `UkrCard`, `RuPay`, `InterPayment`, `Dankort`, `Mir`, `BORICA`, `NPS Pridnestrovie`, `Troy`, `UATP`, `Verve`, `LankaPay`, `UzCard` and `Humo`. There are individual new entities for each of these types, and they are also matched in the `number/cc` entity. These credit cards are also available individually from the `number_cc.ecr` Education standard grammar.
- The Education standard grammar `place_engus.ecr` has been updated with two new public entities:
  - `place/city1_trimmed/engus` to match US cities with population over 100k, without a state identifier. For example *Edison*.
  - `place/city1_uppercase_trimmed/engus` to match uppercase US cities with population over 100k, without a state identifier. For example *EDISON*.

### Enhancements to Education Server

- Education Server has been improved so that when you specify entities on an ACI action, the server does not need to create a new education engine for the new set of entities.
- ACI servers that are using an OEM license (or Enterprise OEM license) can now accept HTTP multipart/form-data in a POST request. The request URL must be encrypted to ensure that the client has access to the OEM keys. The multipart/form-data POST request is in plain text.

### Enhancements to Education SDK

- You can now obtain a detailed error message if a factory fails to create an engine for some reason. Each SDK has its own method for doing this. The following examples show an attempt to create an engine from a non-existent configuration file, and assume a factory has already been created.
  - C SDK. The C SDK defines a dedicated function for obtaining the error message for the last error returned by a factory.

```
EdkEngineHandle engine = NULL;
EdkError err = EdkFactoryMakeEngineFromConfigFile(factory, &engine, "no_
such_file.cfg");
if (err != EdkSuccess)
{
    const char *szError = NULL;
    EdkGetLastFactoryError(factory, &szError);
    fprintf(stdout, "%s\n", szError);
}
```

- Java SDK. The Java SDK provides a more detailed message in the exceptions it throws if a factory fails to create an engine.

```
try (TextExtractionEngine<EDKMatch> engine =
factory.createEngineFromConfigFile("no_such_file.cfg"))
{
```

```
        //do something with engine  
    }  
    catch (EDKRuntimeException ex)  
    {  
        System.out.println(ex.getMessage());  
    }  
}
```

- .NET SDK. The .NET SDK provides a more detailed message in the exceptions it throws if a factory fails to create an engine.

```
try  
{  
    using (ITextExtractionEngine engine = factory.GetTextExtractionEngine  
("no_such_file.cfg"))  
    {  
        //do something with engine  
    }  
}  
catch (EDKException ex)  
{  
    Console.WriteLine(ex.Message);  
}
```

- The .NET SDK now provides more detailed error messages for any exception that is thrown by a factory or engine, for example if you try to compile an invalid grammar file.
- Methods have been added to the SDK to allow you to set a limit on the number of matches for a specified entity on a per-session basis. This option overrides the `EntityMatchLimitM` configuration parameter set for the engine. The following methods are available:
  - C: `EdkError EdkSessionSetEntityMatchLimit(const char* entity, int limit)`
  - .NET: `void ITextExtractionSession::SetEntityMatchLimit(string entity, int limit)`
  - Java: `void TextExtractionSession<?>::setEntityMatchLimit(final String entity, int limit)`

Set these values to -1 to mean no limit, and 0 to mean do not match this entity.

## Resolved Issues

The following issues were resolved in Education version 12.13.0.

- When a single Education engine configuration included `PreFilterMaxReturnedBytes` and multiple prefilters, Education sometimes merged prefilter windows incorrectly, and produced incorrect match offsets or an `EdkErrInvalidArg` exception.
- Setting the `RedactionReplacementCharacter` configuration parameter to a multibyte Unicode character caused an error to be reported.

### Resolved Issues for Education SDK

- The `repeated_matches` sample code was missing from the ZIP for the Education SDK.
- Using `EdkGetNextRepeatedMatch` (or equivalent) when your configuration has `AllowMultipleResults` set to `true` and a prefilter, could result in the next `EdkGetNextMatch` call returning an `EdkErrException (203)` error.
- `EdkGetNextRepeatedMatch` did not always respect the configured `EntityMatchLimit`.
- `EdkGetNextRepeatedMatch` did not work correctly with EJR grammars.
- The Java SDK did not return the correct byte and codepoint offsets for matches when the input features SMP (Supplementary Multilingual Plane) UTF-8 characters.

**NOTE:** This issue affected only the Java SDK. The `edktool` and C SDK returned the correct offset information.

# Notes

These notes provide extra information about installing and using Education.

- The sample Education configuration files have been updated to remove deprecated settings.

## Deprecated Features

The following features have been deprecated.

Category	Deprecated Feature	Deprecated Since
PII Grammar	In the <code>name_cjkvt.ecr</code> grammar, the <code>pre_title</code> and <code>post_title</code> entities in the format <code>pii/name/pre_title/CC</code> and <code>pii/name/post_title/CC</code> have been deprecated (where <code>CC</code> is the country code <code>cn</code> , <code>kr</code> , or <code>tw</code> ). You can now specify these entities in the format <code>pii/name/pre_title/nocontext/CC</code> , for consistency with the <code>jp</code> entities.	12.13
Education SDK	<p>The ability to specify a license key by supplying a file path has been deprecated. Micro Focus recommends embedding your license key in your application as a string, to avoid having the license key in a file on disk.</p> <p>In the C API the following functions have been deprecated:</p> <ul style="list-style-type: none"><li>• <code>EdkEngineCreate()</code></li><li>• <code>EdkEngineCreateFromConfigFile()</code></li><li>• <code>EdkSetLicenseKey()</code></li><li>• <code>EdkSetLicenseKeyFromFile()</code></li></ul> <p>In the Java API the following have been deprecated:</p> <ul style="list-style-type: none"><li>• Both of the constructors for <code>EDKEngine</code>.</li><li>• The <code>setLicenseKey</code> method of <code>EDKEngine</code>.</li></ul> <p>In the .NET API, the following has been deprecated:</p> <ul style="list-style-type: none"><li>• The <code>EDKFactory</code> constructor: <code>public EDKFactory(string license_key_path)</code></li></ul> <p>Micro Focus recommends supplying the license key as a string, and creating an Education engine by using an engine factory. For more information, refer to the API reference documentation.</p>	12.11

.NET Education SDK	Education SDK support for .NET Standard 1.1 has been deprecated and might be removed in future. Micro Focus recommends using a .NET implementation that supports .NET Standard 2.0.	12.11
C Education SDK	EdkSessionSetMatchTimeout EdkSessionSetRequestTimeout EdkGetNextMatchTimed	12.8
.NET Education SDK	ITextExtractionSession::SetMatchTimeout ITextExtractionSession::SetRequestTimeout ITextExtractionSession::SetStartTime	12.8
Java Education SDK	TextExtractionSession<T>::setMatchTimeout TextExtractionSession<T>::setRequestTimeout	12.8
Java Education SDK	EDKMatch::getTimedOut function	12.6
edktool configuration	SearchFields parameter. This parameter applies only to IDX document input, which was deprecated in 12.5.0.	12.6
Configuration	[PostProcessingTasks] configuration section. Use the PostProcessingTaskN and PostProcessThreshold parameters in the [Education] section.	12.5
edktool	The OutputScores and OutputSimpleMatchInfo configuration parameters.  <b>NOTE:</b> The default values of the parameters OutputSimpleMatchInfo and OutputScores have been updated, to False and True respectively.	12.5
edktool	XML configuration files are now deprecated. Use CFG format configuration files instead.	12.5
edktool	Education on IDX documents. The following configuration parameters that apply only to IDX are also deprecated as part of this change: <ul style="list-style-type: none"> <li>• AllowDuplicates</li> <li>• CantHaveFieldCSVs</li> <li>• Databases</li> <li>• DocumentDelimiterCSVs</li> </ul>	12.5

# Requirements

This section describes the system requirements, supported platforms, and software dependencies for Education 12.13.0.

## Minimum System Requirements

The following are minimum system requirements for Education 12.13.0.

- 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 8.0 JDK and JRE.

### Microsoft Visual C++ Redistributable Package

To run Education version 12.13.0 on the Microsoft Windows operating system, you might need to install Microsoft Visual C++ Redistributable packages. The Education SDK and Education Server stand-alone zip packages include the required redistributable files for Microsoft Visual C++ 2017. You can also update your packages by using the latest version at: <http://support.microsoft.com/kb/2019667>.

### UNIX Libraries

To run Education version 12.13.0 on UNIX platforms (except MacOS on the M1 platform), the server must have the following minimum versions of libraries:

- GLIBC\_2.17
- GLIBCXX\_3.4.30
- GCC\_4.8.0

**NOTE:** The Education SDK and Education Server stand-alone zip packages provide these libraries in the `libgcc_s` and `libstdc++` shared libraries.

You might need to set the `LD_LIBRARY_PATH` to include the `InstalLDir/bin` directory, to ensure that Education can access the installed shared libraries.

You can also copy the shared libraries to the component working directory.

## Supported Operating System Platforms

The following operating system platforms are supported by Education 12.13.0.

- Windows x86-64
- Linux x86-64
- Linux ARM 64 (centos AArch 64) (Education SDK only)
- MacOS x86-64 and M1 (Education SDK only)

# Documentation

The following documentation was updated for Education version 12.13.0.

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