IDOL PHI Package

Software Version 12.4

Technical Note

Document Release Date: October 2019
Software Release Date: October 2019

Legal notices

Copyright notice

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.

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:

- 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. If you need an account, 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

Introduction

The IDOL PHI Package contains tools that allow you to locate Protected Healthcare Information (PHI) in your data, to ensure compliance with regulations such as the *Standards of Privacy of Individually Identifiable Health Information* implemented as part of the Health Insurance Portability and Accountability Act (HIPAA).

The IDOL PHI Package has two types of tools:

- IDOL Eduction Grammars (.ecr files). IDOL Eduction is a tool for finding entities (small pieces of information such as names and phone numbers) in text. Eduction grammars contain descriptions of the entities. In some cases, this might be a list of fixed values (such as names), and in others it might be pattern matching tools that find data of a particular type (such as a set of digits that make up a phone number). The Eduction grammars included in the IDOL PHI Package describe different kinds of personally identifiable information, so that you can find these in your data.
- IDOL AgentBoolean IDX. IDOL AgentBoolean is a method of storing entities and querying for them that uses the IDOL Agentstore component (a specially configured IDOL Content component), rather than Eduction. The IDX files are index files that contain the details of the entities, which you can index into the IDOL Agentstore component.

Data Sources

The IDOL PHI Package contains a variety of different kinds of entities to describe healthcare information that is protected by regulations such as HIPAA. The following sections provide some information about how this information is compiled.

For each entity type, extensive testing is performed to ensure the precision and recall metrics are optimized. Many millions of examples are run through the package to test full coverage of the patterns and algorithms involved.

Names

An international database containing over 100 million individuals is analyzed to identify the structure and characteristics of names in each country. In doing so, extensive lists of the frequencies of occurrence of given names and family names are used to generate strong identification grammars for names.

In addition, rules are included to handle linguistic information, such as transliteration (for example, from the Cyrillic or Greek alphabets), or the use or removal of diacritic marks.

Age

The linguistic patterns of usage of both unstructured and semi-structured text have been analyzed in all supported languages to determine the range of formats used to refer to a patient's age or age demographic. The resulting grammar establishes a confidence measure to distinguish references to age as opposed to other information, and includes all elements of dates that allow the determination of age.

Dates

A large corpus of documents from public sources is processed to analyze the occurrence and format of dates. In this way, coverage of all common and less-common formats is built up, while enabling a *likelihood* measure to indicate the confidence that the characters identified are a date of birth, rather than an unrelated date or other alphanumeric string.

Dates of any type that relate to an individual's healthcare, other than a single year, are covered by PHI regulation. The IDOL PHI Package allows determination of all such dates from analysis of linguistic patterns in all supported languages. In addition, the package can identify dates of particular types, such as date of death, and hospital admission and discharge dates.

Postal Codes

For each country, the publications of the national Postal Services are used as the authoritative source on the postal code.

In addition, testing against widely-gathered examples allows the identification and inclusion of non-standard formats and common errors (such as mixing the letter O with the digit 0), with an appropriately adjusted likelihood measure.

Addresses and Locations

The identification of addresses consists of a number of steps, each of which is used as additional evidence that a piece of text represents a postal address. These are:

1. The format of the text.
2. The house number / street-name portion.
3. The village / town / county / region portion.
4. The postal code.

These components are not necessarily always present for a particular address, but each is taken as evidence that the text does indeed contain an address, combining to form an overall likelihood.

- Few countries have prescribed formats for addresses, while most have conventions defined by the national Postal Service that is generally adhered to, but also frequently ignored.

  The IDOL Web Connector is used to gather many millions of web documents to identify candidate addresses in each applicable country. From there, the variety of formats that are used in practice are identified. In addition, any recommendations published by the national Postal Services are also used.

- For the street-address portion, the extensive OpenStreetMap project is used, and a database of every named street in each of the supported countries is obtained and analyzed. From this database, rules are derived to allow the identification of the vast majority of street-address strings.
- The de facto authority for geographical place names is the GeoNames database, with 11 million locations identified by data including country, population and type. In particular the *type* field is used to generate complete lists of populated settlements and administrative regions (such as county / department / region) for the countries that frequently use those in addresses. In addition, the names are available in different character sets and transliteration schemes to ensure internationalization.
- The patterns derived for matching Postal Codes are also used here (see Postal Codes, on the previous page).

For locations, the IDOL PHI Package identifies any address portion smaller than a state.

Telephone Number

The general schemes for the creation of telephone numbers and fax numbers are readily available from the appropriate government department of each country. However, the formats of such numbers when written down varies considerably within a country, and even more so when numbers are referred to in a foreign document.

The strategy for creating comprehensive phone number matching grammars is centered on several key methods:

- Knowledge of the national scheme for assigning numbers.
- Databases of international and area codes in each country, obtained from authoritative sources.
- Analysis of many millions of examples of the usage of telephone numbers, obtained from a wide variety of public sources.

This final point is the most important. Only through examination of real-world usage of such numbers is the full range of formats obtained for each country.

The proximity of keywords indicating that the digits represent a telephone or fax number is used to strengthen the likelihood of the match.

Email Address

The IETF publications RFC 5321 and RFC 5322 define the standards of validity of email addresses, and so the IDOL PHI Package uses these for this purpose. In addition, it uses metrics of likelihood

derived from the analysis of the most common email domains, to allow the grammars to differentiate between likely email addresses and those that are unlikely but still valid (for example, example@example.test).

IP Address/URL

The formats of IP Addresses are defined by the IETF in RFC 791 (IPv4) and RFC 4291 (IPv6) with later modifications. These allow the location of potentially identifiable information in candidate text by the IDOL PHI Package.

In the same way, Uniform Resource Locators (URL) were defined in RFC 1738.

National Identification Number

Each country has a different scheme for the use of National Identification. For countries with National ID cards, the format of the number is derived from governmental sources. In other countries, the formats of National Health, National Social Security, or National Insurance numbers are obtained from governmental sites, with the exception of a few cases in which other sources are used.

Vehicle Identifiers

Each country has a different scheme for the identification of vehicles by license plates. In each case, the national Vehicle Licensing Authority is used as the authoritative source of such information.

In each type, there are often standard and non-standard formats, with the former following a prescribed system more tightly. In the identification of such plates, a likelihood metric is used to take into account such formats and give a confidence that an identifier is actually a vehicle license.

Medical

Documents that contain mention of medical procedures or conditions are identified with the Medical categories, available in each of the supported languages. The categories are generated from the Medical Subject Headings (MeSH) taxonomy published by the United States National Library of Medicine using the C hierarchy (diseases and conditions).

Profession

Documents that contain mention of an individual's occupation or profession are identified by the IDOL PHI Package in each supported language. The items are generated from an international database of over 60 million items.

Unique Device Identifiers

The IDOL PHI Package identifies Unique Device Identifiers (UDI) for medical devices. The formats match the standard formats issued by the three agencies accredited by the US Food and Drug Administration (FDA): GS1, HIBCC, and ICCBA.

Health Plan and Medical Record Numbers

The IDOL PHI Package identifies health plan numbers and Medical Record Numbers (MRN).

Health plan numbers have a standard format, which is readily available from governmental sources.

MRN formats differ for different healthcare provider. In this case, example formats are used to create as broad an identifier as possible, and landmark text is used to locate likely numbers. A grammar extension is also used to allow you to restrict the MRN detection to known formats, when you have specific formats you would like to detect. See Medical Record Numbers, on page 19.

IDOL Eduction Grammars

The following section describes the Eduction grammars available in the IDOL PHI Package.

You can use these grammars with IDOL Eduction, by using Eduction Server, the edktool command-line utility, or the Eduction SDK. For more information, refer to the *IDOL Eduction User Guide* and the *Eduction SDK Programming Guide*.

> **IMPORTANT:** To use the Eduction grammars in the IDOL PHI Package, you must have a license that enables them. To obtain a license, contact Micro Focus Support.

Configure Post Processing

When you use the IDOL PHI Package Eduction grammars it is essential to configure a Lua post-processing task to run the script `pii_postprocessing.lua`. This script contains post-processing to improve results for various entities, such as stop list filtering, and checksum validation (see Validated ID Numbers, on page 20).

> **IMPORTANT:** If you do not run this script, you might encounter unexpected behavior.

Add a post-processing task to your Eduction configuration. For example:

```
[PostProcessingTasks]
NumTasks=1
Task0=MyPostProcessingSection

[MyPostProcessingSection]
Type=Lua
Script=scripts/phi_postprocessing.lua
Entities=phi/*
```

For more information about configuring post-processing tasks, refer to the *Eduction User and Programming Guide*.

Configure Pre-Filtering

Pre-filtering allows the IDOL PHI Package to run a quick initial check to find potential matches in your input text. It then selects match windows around these potential matches, reducing the amount of text that it must match against your grammars. This process can improve the performance in certain cases.

Micro Focus recommends that you use the following pre-filtering configuration with the `address.ecr` grammar.

```
[Eduction]
PrefilterTask0=AddressPrefilter
```

```
[AddressPrefilter]
Regex=\d{1,7}
WindowCharsBeforeMatch=100
WindowCharsAfterMatch=100
```

> **NOTE:** Pre-filter tasks run for all configured entities, so you must configure it only for the appropriate entities to ensure that it does not affect the results for other entities.

For more information about pre-filtering, refer to the *Eduction User and Programming Guide*.

Entity Context

Some of the entities are available in two versions, with and without context. The context-based entities match the entity when it occurs in an easily identifiable location in text. For example, it might match a telephone number that occurs next to the prefix **Phone:**.

The entities that do not have context attempt to match the entity wherever it occurs. This version might over-match significantly (that is, it is likely to return values that are similar to the entity patterns, such a number that is not a telephone number). However, it also reduces the number of false negatives (that is, it misses fewer matches).

You can configure Eduction to use both versions of an entity; matches located with context are given a higher score in the results.

Balance Precision and Recall

In many cases, Eduction is able to locate entities that are ambiguous, such as a postal code which is simply a five-digit number. In some situations it is desirable to match as many entities as possible ("high recall") and in others only entities with a high likelihood of being a useful match ("high precision"). So that you can filter the results produced by Eduction, each match is given a 'score' value.

As described in Entity Context, matches located by an entity that requires context are assigned higher scores than matches located by the corresponding entity without context. Most matches extracted without context have a score of `0.4`. For example, a context-free date ("January 18, 1998") might be returned by the Date Of Birth entity with a score of `0.4`. But with context to suggest that it is indeed a date of birth ("DOB: January 18, 1998"), the score should be above `0.5`.

The PII post-processing script (see Configure Post Processing, on the previous page) includes a step to validate matches (some ID numbers can be validated by calculating a checksum). The script increases the score of matches that have valid checksums, because this is an indication that the match is more likely to be genuine. Any match that has an invalid checksum is immediately discarded because it cannot be genuine.

When you configure Eduction, use the parameters `MinScore` and `PostProcessThreshold` to achieve the desired balance between precision and recall. Eduction discards any match with a score lower than `MinScore`. Matches with scores that meet or exceed `MinScore` are then processed by post-processing tasks. After post-processing has finished, Eduction discards any match with a score lower than `PostProcessThreshold`.

In the example configuration that is included with the IDOL PHI Package, `MinScore` is set to `0.4` and `PostProcessThreshold` is set to `0.5`. These values have been chosen to return results only if they have a relatively high likelihood of being a useful match. Any match that is located without context can proceed to post-processing, but, unless its score is increased through successful validation, it is then discarded. If you prefer to maximize recall rather than precision, you can reduce or remove these thresholds.

For more information about Eduction configuration parameters, refer to the *Eduction User Guide*.

Configure Tangible Characters

`TangibleCharacters` is a configuration parameter that you can set when using the Eduction SDK, the Eduction Server, or the Eduction command-line utility (`edktool`). It specifies a list of characters to treat as part of a word, rather than as word boundaries.

Some of the entities in the IDOL PHI Package Eduction Grammars require tangible characters to be set in order to perform correctly (see the descriptions of the entities in Eduction Grammar Reference, on page 13).

When you use Eduction to search for matches, `TangibleCharacters` applies across all of your chosen entities. If you use multiple entities that have different recommended tangible character sets, you might need to take some extra steps. For example:

- If you are using the Eduction SDK, create a separate EDK engine for each distinct set of tangible characters, and configure the tangible characters for the engine using the appropriate API call:

C	`EdkSetTangibleCharacters`
Java	`EDKEngine.setTangibleCharacters`

  After configuring an engine with the correct tangible characters, you can add the relevant entities. You will need to create a session from each engine to process your input text.

- If you are using an Eduction Server, send a separate action (`EduceFromText` or `EduceFromFile`) for each distinct set of tangible characters. In each action, set the `TangibleCharacters` and `Entities` action parameters to specify which set of tangible characters and which entities to use.

- If you are using the command line `edktool`, create a separate configuration file for each distinct set of tangible characters and associated entities, and process your input text once with each configuration file.

For more information about the `TangibleCharacters` configuration parameter, refer to the *Eduction User Guide*.

Customize Stop Lists

The IDOL PHI Package post-processing script (see Configure Post Processing, on page 9) uses stop lists to discard matches that are likely to be false positives. You can add entries to the stop lists, or remove entries, by modifying the following files.

- `scripts/address_stoplist.lua` contains a list of common words that are likely to indicate a false positive when returned as the `STREET` or `CITY` component of an address match.
- `scripts/names_stoplist.lua` contains two stop lists to discard names. In the first stop list, each component is plausible but the entire match is likely to be a false positive, for example "Christian Church" or "Norman Conquest". The second stop list contains common words that are likely to indicate a false positive when returned as either the `FORENAME` or `SURNAME` component of a name match. The stop lists in this file can be customized such that a name can be considered a false positive in one country but not another.

Eduction Grammar Reference

The following table describes the grammar files that are available in the IDOL PHI Package, and the entities that each provides.

File	Entity	Description
account.ecr	phi/account/bank/context/us	A US bank account number, with context.
	phi/account/bank/nocontext/us	A US bank account number, without context.
	phi/account/bank/landmark/us	A bank account landmark, such as "Bank Account Number".
	phi/account/swiftcode/context/us	A SWIFT code, with context.
	phi/account/swiftcode/nocontext/us	A SWIFT code, without context.
	phi/account/swiftcode/landmark/us	A SWIFT code landmark, such as "SWIFT Code".
address.ecr	phi/address/us	A postal address. In general, a score of one is given to an address that includes a numbered, common format street address (for example "23 North Road"), a known city (for example "London"), and a postal code in a viable format for the country (for example "SW1A 2AA"). Deviations from this form lead to score penalties. The ordering of these elements varies by country. Micro Focus recommends that you use pre-filtering to improve the performance for this grammar. See Configure Pre-Filtering, on page 9. Example matches: "Schlosshoferstrasse 20, 1210 Vienna", "Avenida Juan Xxiii 20, 41006, Sevilla", "162-168 Regent Street, London, W1B 5TG".

File	Entity	Description
age.ecr	phi/age/context/us	A US age statement with context. For example "Age: 99".
	phi/age/nocontext/us	A US age statement without context. For example "99 years old".
	phi/age/landmark/us	A US age landmark. For example "Age".
certificate.ecr	phi/certificate/birth/context/us	A US birth certificate number with context. For example "Birth Certificate: 160 99 123456".
	phi/certificate/birth/nocontext/us	A US birth certificate number without context. For example "160 99 123456".
	phi/certificate/birth/landmark/us	A US birth certificate landmark. For example "Birth Certificate".
	phi/certificate/generic/context/us	A US generic certificate number with context. For example "Certifcate number "MX-123-456/78".
	phi/certificate/generic/nocontext/us	A US generic certificate number without context. This option is available only for the certificate user extension. See Generic Certificate Numbers, on page 20.
	phi/certificate/generic/landmark/us	A US generic certificate landmark. For example "Certifcate number".

File	Entity	Description
date.ecr	phi/date/nocontext/eng	An English date, without context. For example, "01/13/1981".
	phi/date/noyear/nocontext/eng	An English date without the year, without context. For example, "01/13"
	phi/date/dob/context/eng	An English date of birth, with context. For example, "DOB: 1/13/1981".
	phi/date/dob/noyear/context/eng	An English date of birth without the year, with context. For example, "DOB: 01/13".
	phi/date/dob/landmark/eng	An English date of birth landmark. For example, "DOB".
	phi/date/dod/context/eng	An English date of death, with context. For example, "Died on 01/13/1981".
	phi/date/dod/noyear/context/eng	An English date of death without the year, with context. For example, "Died on 01/13".
	phi/date/dod/landmark/eng	An English date of death landmark. For example, "Died on".
	phi/date/medical/context/eng	An English medical date with context. For example, "Admission date: 01/13/1981".
	phi/date/medical/noyear/context/eng	An English medical date without the year, with context. For example, "Admission date: 01/13".
	phi/date/medical/landmark/eng	An English medical date landmark. For example, "Admission date".
device.ecr	phi/device/udi/nocontext	A Unique Device Identifier (UDI) required by the United States FDA, as issued by the three accredited agencies, GS1, HIBCC, and ICCBA. For example: "+X999123ABC0/$$31905151234AB/S5678EDFG/16D20151001J" **NOTE:** To ensure that this entity performs correctly, set your `TangibleCharacters` configuration to include the following characters: `(+=`. For more information, see Configure Tangible Characters, on page 11.
healthplan.ecr	phi/healthplan/context/us	A PHI health plan beneficiary number. These numbers consist of fourteen alphanumeric characters, preceded by a suitable landmark. For example,

File	Entity	Description
		"Plan ID 12345678LMNOP9".
	phi/healthplan/nocontext/us	A PHI health plan beneficiary number without context. For example, "12345678LMNOP9".
	phi/healthplan/landmark/us	A health plan number landmark. For example, "Health Plan Number".
	phi/mrn/context/us	A Medical Record Number. See Medical Record Numbers, on page 19.
	phi/mrn/nocontext/us	A Medical Record Number without context. This option is available only for the healthplan user extension. See Medical Record Numbers, on page 19.
	phi/mrn/landmark/us	A Medical Record Number landmark, such as "MRN".
internet.ecr	phi/inet/email/context	An email address with context. For example, "e-mail: jsmith@mailserver.com".
	phi/inet/email/nocontext	An email address without context. For example, "jsmith@mailserver.com".
	phi/inet/email/landmark	An email address landmark. For example, "e-mail".
	phi/inet/ip/context	An IP address with context. For example, "ip address: 5.5.5.5".
	phi/inet/ip/nocontext	An IP address without context. For example, "10.12.14.16".
	phi/inet/ip/landmark	An IP address landmark. For example, "ip address".
	phi/inet/url/context	A URL with context. For example, "uri: https://www.example.com".
	phi/inet/url/nocontext	A URL without context. For example, "www.example.com".
	phi/inet/url/landmark	A URL landmark. For example, "url".

File	Entity	Description
laboratory.ecr	phi/laboratory/context/us	A US laboratory number with context. For example "CLIA No: 01D1234567".
	phi/laboratory/nocontext/us	A US laboratory number without context. For example "01D1234567".
	phi/laboratory/landmark/us	A US laboratory number landmark. For example, "CLIA No".
license.ecr	phi/license/driving/context/us	A US driving license number with context. For example, "Driving license: 012AB3456".
	phi/license/driving/nocontext/us	A US driving license number without context. For example, "012AB3456".
	phi/license/driving/landmark/us	A US driving license landmark. For example, "Driving license".
	phi/license/generic/context/us	A US generic license number with context. For example, "License number: MX-123-456/78".
	phi/license/generic/nocontext/us	A US generic license number without context. This option is available only for the license user extension. See Generic License Numbers, on page 19.
	phi/license/generic/landmark/us	A US generic license landmark. For example, "License number".
location.ecr	phi/location/us	A US subdivision smaller than a state, such as towns, cities, and counties. For example, "Houston". Scores are boosted by the presence of a state, a zipcode, or a nearby landmark value. For example, "Houston" scores 0.4, while "Houston, Texas" scores 0.65 and "city of Houston, Texas" scores 1.
name.ecr	phi/name/us	A full personal name, in title case or upper case.
national_id.ecr	phi/id/context/us	A national identity number (US Social Security Number) with context.
	phi/id/nocontext/us	A national identity number (US Social Security Number) without context.
	phi/id/landmark/us	A national identity number landmark, such as "Social security number".
telephone.ecr	phi/telephone/context/us	A telephone number with context. For example "Tel: +44 1234 224050",

File	Entity	Description
		"Telephone: (204)-243-9955", or "numéro de téléphone: +1-902-861-7000". **NOTE:** To ensure that this entity performs correctly, set your `TangibleCharacters` configuration to include the following characters: `()+-`. For more information, see Configure Tangible Characters, on page 11.
	phi/telephone/nocontext/us	A telephone number without context. For example: "+39 055 326 43 11", or "44 20 7499 9000". **NOTE:** To ensure that this entity performs correctly, set your `TangibleCharacters` configuration to include the following characters: `()+-`. For more information, see Configure Tangible Characters, on page 11.
	phi/telephone/landmark/us	A telephone number landmark, such as "Tel:".
vehicle.ecr	phi/vehicle/licenseplate/context/us	A vehicle license place number with context. For example, "License Plate Number: ABC 123".
	phi/vehicle/licenseplate/nocontext/us	A vehicle license place number without context. For example, "ABC 123".
	phi/vehicle/licenseplate/landmark/us	A vehicle license plate number landmark. For example, "License Plate Number".
	phi/vehicle/vin/context/us	A vehicle identification number with context. For example, "VIN: LJCPCBLCX11000237".
	phi/vehicle/vin/nocontext/us	A vehicle identification number without context. For example, "LJCPCBLCX11000237".
	phi/vehicle/vin/landmark/us	A vehicle identification number landmark. For example, "VIN".

User Grammar Extensions

In some cases, the PHI grammar files provide entities for values that are very broad, to allow you to find values that do not have very well-defined formats. To reduce the number of false positives in these cases, only the context form of the entity is available by default (that is, with a suitable landmark).

You can use a user extension to expand the entity to include more specific formats, which enables the `nocontext` entity.

Medical Record Numbers

Medical Record Numbers (MRN) are provider-specific, and have a large number of possible formats. Therefore, the provided MRN entities in the `healthplan.ecr` grammar are generic, and matches a string of 7-15 alphanumeric characters, with optional - and / separators.

Because the MRN pattern is so broad, it can match a lot of values that are not actually MRNs. To reduce the number of false positives, only the context form of the entity is available by default (that is, with a suitable landmark).

If you have a specific set of MRN values with well-defined patterns that you want to match, you can use the `healthplan_user.xml` extension grammar. This XML file allows you to expand the MRN grammar with more specific patterns, and compile them into a grammar. In this case, the original `context` entity is available, with your additional entities, and it also enables the `nocontext` entity.

For details of how to expand and compile the user XML grammar, refer to the *Eduction User and Programming Guide*.

Generic License Numbers

The generic entities in the `license.ecr` grammar attempt to match a variety of different license numbers. The `phi/license/generic` entity matches a string of 5-20 alphanumeric characters (which must include at least one number), with optional - and / separators.

If you have a specific set of license values with well-defined patterns that you want to match, you can use the `license_user.xml` extension grammar. This XML file allows you to expand the grammar with more specific patterns, and compile them into a grammar. In this case, the original `context` entity is available, with your additional entities, and it also enables the `nocontext` entity.

For details of how to expand and compile the user XML grammar, refer to the *Eduction User and Programming Guide*.

Generic Certificate Numbers

The generic entities in the `certificate.ecr` grammar attempt to match a variety of different license numbers. The `phi/certificate/generic` entity matches a string of 5-20 alphanumeric characters (which must include at least one number), with optional - and / separators.

If you have a specific set of certificate values with well-defined patterns that you want to match, you can use the `certificate_user.xml` extension grammar. This XML file allows you to expand the grammar with more specific patterns, and compile them into a grammar. In this case, the original `context` entity is available, with your additional entities, and it also enables the `nocontext` entity.

For details of how to expand and compile the user XML grammar, refer to the *Eduction User and Programming Guide*.

Validated ID Numbers

The script `phi_postprocessing.lua` (see Configure Post Processing, on page 9) includes steps to validate ID numbers that are found by Eduction. This improves accuracy by discarding results that match the pattern for a valid ID number, but cannot be genuine because they have an invalid checksum. The script increases the score for matches that have a valid checksum, because this is an indication that the match is more likely to be genuine.

The following tables list the entities that are validated.

Unique Device Identifiers (`device.ecr`)
phi/device/udi/nocontext

IDOL AgentBoolean IDX

IDOL AgentBoolean provides another way of finding pieces of information in text. In this case, you index the entities that you want to find into an IDOL Agentstore component.

The IDOL Agentstore component is a specially configured IDOL Content component. It uses IDOL AgentBoolean queries for entity matching.

When you use AgentBoolean for entity matching, each entity becomes a document in Agentstore. You then send a piece of text as a query to Agentstore, and it returns the entity documents that match the text.

The IDOL PHI Package contains IDX documents that describe entities for medical data (such as conditions and procedures), and professions and occupations. You can use these IDX documents as another tool to find data that is protected by PHI regulations.

The package also contains example Agentstore configuration files to allow you to set up your Agentstore component more easily.

After you configure and set up your Agentstore, you can index the IDX documents and use Agentstore for entity matching.

For more information about how to set up and use IDOL querying, refer to the *IDOL Server Administration Guide* and the *IDOL Content Component Reference*.

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 Technical Note (Micro Focus IDOL PHI Package 12.4)

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!