PUBLIC

SAP HANA Platform SPS 12
Document Version: 1.2 – 2018-01-24

SAP HANA Text Analysis Developer Guide

Content

1 SAP HANA Text Analysis Developer Guide

Text analysis is a feature enabled with the full-text index to discover and classify entities in your documents.

Text analysis provides a vast number of possible entity types and analysis rules for many industries in many languages. You do not have to deal with this complexity when analyzing your individual set of documents. The language modules included with the software contain system dictionaries and provide an extensive set of predefined entity types. The extraction process can extract entities using these lists of specific entities. It can also discover new entities using linguistic models. Extraction classifies each extracted entity by entity type and presents this metadata in a standardized format. You can also customize the text analysis process and even define your own entity types.

The following data types are enabled for text analysis: TEXT, BINTEXT, NVARCHAR, VARCHAR, NCLOB, CLOB, and BLOB.

Individual text analysis options are grouped into text analysis configurations, which are stored in the SAP HANA system in an XML format. The system includes a number of predefined configurations. You can use any of these, or create your own custom text analysis configurations. To use your own text analysis extraction dictionaries and extraction rules, you need to create a custom text analysis configuration.

The following text analysis configurations are delivered by SAP:

Text Analysis Configurations

Name of Text Analysis Configuration	Description
`LINGANALYSIS_BASIC`	This configuration provides the following language processing capabilities for linguistic analysis of unstructured data: • Segmentation, also known as tokenization - the separation of input text into its elements
`LINGANALYSIS_STEMS`	This configuration provides the following language processing capabilities for linguistic analysis of unstructured data: • Segmentation, also known as tokenization - the separation of input text into its elements • Stemming - the identification of word stems or dictionary base forms
`LINGANALYSIS_FULL`	This configuration provides the following language processing capabilities for linguistic analysis of unstructured data: • Segmentation, also known as tokenization - the separation of input text into its elements • Stemming - the identification of word stems or dictionary base forms • Tagging - the labeling of words' parts of speech
`EXTRACTION_CORE`	This configuration extracts entities of interest from unstructured text, such as people, organizations, or places mentioned. In most use cases, this option is sufficient.

Name of Text Analysis Configuration	Description
EXTRACTION_CORE_ENTERPRISE	This configuration includes a set of entity types and rules for extracting information about organizations, such as management changes, product releases, mergers, acquisitions, and affiliations.
EXTRACTION_CORE_PUBLIC_SECTOR	This configuration includes a set of entity types and rules for extracting information about events, persons, organizations, and their relationships, specifically oriented towards security-related events.
EXTRACTION_CORE_VOICEOFCUSTOMER	Voice of the customer content includes a set of entity types and rules that address requirements for extracting customer sentiments and requests. You can use this content to retrieve specific information about your customers' needs and perceptions when processing and analyzing text. This configuration involves complex linguistic analysis and pattern matching that includes processing parts of speech, syntactic patterns, negation, and so on, to identify the patterns to be extracted. The keyword dictionaries used to identify and classify sentiments can also be customized, if needed, for specific applications. Refer to the topic *Creating Custom Text Analysis Dictionaries* for more information.
GRAMMATICAL_ROLE_ANALYSIS	This configuration provides the capability to identify functional (grammatical) relationships between elements in an input sentence (e.g., Subject, DirectObject). **Note** This feature is only supported for English.

To use the text analysis function, create a full-text index on the column containing your texts with the following parameters:

```
TEXT ANALYSIS ON

CONFIGURATION '<NAME OF TEXT ANALYSIS CONFIGURATION>'
```

Note

The technical names of the text analysis configurations are case-sensitive.

Not all configurations are supported in all languages. For details, see the Text Analysis Language Reference Guide.

If your tables contain a language indicator, enter the name of the column:

```
LANGUAGE COLUMN <NAME OF COLUMN CONTAINING THE LANGUAGE INDICATOR>
```

If no language is specified, EN is used by default.

Once indexing starts, the text analysis runs in the background. Depending on the number and size of the texts, a single analysis can take several minutes or more. To check the status of the text analysis, you can use the default monitoring view `SYS.M_FULLTEXT_QUEUES`.

For each full-text index, the system creates an additional table with the name `$TA_<index_name>` in the same schema that contains the source table.

This table stores the extracted entities and the analysis results. You can use this table to build joins with other search-enabled views, for example to use the data for interactive navigation or auto-completion in search input fields.

For detailed information on this table, see Structure of the $TA Table [page 6].

To track deletions in the source table, the keys in the `$TA` table need to be aligned to the keys of the source table. To do this, use the following SQL statement:

```
ALTER TABLE "<schema>"."$TA_INDEX_NAME" ADD CONSTRAINT <constraint name> FOREIGN KEY("key_1", "key_2", "key_n") REFERENCES "<schema>"."<name of source table>"("key_1","key_2","key_n") ON DELETE CASCADE
```

If it becomes too large, you can partition the `$TA_<index_name>` table. Partitioning improves manageability and performance. For example, you can use the following command to partition the `$TA` table using the hash partition strategy: `ALTER TABLE "$TA_<index_name>" PARTITION BY HASH (<PRIMARY_KEY_ATTR_1>, ... , <PRIMARY_KEY_ATTR_N>) PARTITIONS <N>`

Example

Use the `CREATE FULLTEXT INDEX` statement to create an index named `CUSTOMER_INDEX` on your `CUSTOMERS` table to index the `customername` column: `CREATE FULLTEXT INDEX CUSTOMER_INDEX ON "MY_SCHEMA"."CUSTOMERS" ('customername')[<fulltext_parameter_list>]`

If you are triggering the text analysis using the `EXTRACTION_CORE` option, specify the following additional parameters for the full-text index:

```
TEXT ANALYSIS ON

CONFIGURATION 'EXTRACTION_CORE'

LANGUAGE COLUMN "LANG"

ALTER TABLE "MY_SCHEMA"."$TA_CUSTOMER_INDEX" ADD CONSTRAINT ALTER_COMMAND FOREIGN KEY("KEY_1", "KEY_2") REFERENCES "MY_SCHEMA"."CUSTOMERS"("KEY_1","KEY_2") ON DELETE CASCADE
```

Related Information

SAP HANA Text Analysis Language Reference Guide

2 Structure of the $TA Table

The `$TA_<index_name>` table is generated automatically when you trigger the creation of the index. The table is built from the key fields of the source table, additional key fields `TA_RULE` and `TA_COUNTER`, and several additional fields.

Structure of TA table

Column ID	Key	Description	Data Type
<n key columns from source table>	Yes	To support a foreign key definition linking from the `$TA` table to its source table, the `$TA` table has to use exactly the same key columns as its source table (in data type and ID). The `$TA` table includes all keys from the source table.	Same as in source table
`TA_RULE`	Yes	Stores the source that yielded the token. This is also required to distinguish between linguistic analysis output, output from entity extraction from grammatical role analysis and document metadata.	NVARCHAR(200)
`TA_COUNTER`	Yes	The token counter counts all tokens across the document. The order is only unique for a given processing type (hence the TA_RULE as the key).	BIGINT
`TA_TOKEN`	-	Term, entity, or metadata - depending on the processing type.	NVARCHAR(250)
`TA_LANGUAGE`	-	The language of the document is usually stated in the source table. In rare cases where no language is specified, the language code is stored here. Since there is no support for multi-language documents, the language code is identical for all result records of a document.	NVARCHAR(2)
`TA_TYPE`	-	The token type contains the linguistic or semantic type of the token, for instance "noun" (if configuration = `LINGANALYSIS_*`), "company" (if configuration = `EXTRACTION_*`), "Subject" (if configuration = `GRAMMATICAL_ROLE_ANALYSIS`), or "Author" (if OutputMetadata is enabled). The note listed below this table contains a table that includes all SAP HANA `TA_TYPE` values for linguistic analysis.	NVARCHAR(100)

Column ID	Key	Description	Data Type
TA_NORMALIZED	-	Stores a normalized representation of the token. This is relevant, for example, in the case of German, with umlauts, or ß/ss. Normalization includes the following steps: • Words are converted to lowercase • Umlauts are "normalized" (ä to ae, for example) • Diacritics are removed **Note** This column will be NULL for tokens of type "punctuation".	NVARCHAR(250)
TA_STEM	-	Stores the linguistic stemming information, for example the singular nominative for nouns, or the infinitive for verbs. If text analysis yields several stems, only the best stem is stored. **Note** This column will be NULL unless the token has a stem, and the stem is different from the token.	NVARCHAR(300)
TA_PARAGRAPH	-	Stores the relative paragraph number containing the TA_TOKEN (states that the nth paragraph contains TA_TOKEN). This column will be NULL for metadata.	INTEGER
TA_SENTENCE	-	Stores the relative sentence number containing the TA_TOKEN (states that the nth sentence contains TA_TOKEN). This column will be NULL for metadata.	INTEGER
TA_CREATED_AT	-	Stores the creation time. Used only for administrative information, for example, for reorganization.	TIMESTAMP
TA_OFFSET	-	Stores the offset in characters relative to the beginning of the document. This column will be NULL for metadata.	BIGINT

Column ID	Key	Description	Data Type
TA_PARENT		Stores the TA_COUNTER value of the parent token, or NULL if the token has no parent. This field is used to indicate that there is a lingustic relationship between two tokens. For example, it is used by the EXTRACTION_CORE_VOICEOFCUSTOMER rules to relate topics to their enclosing sentiments. **Note** This column will only appear in $TA tables created after installing SAP HANA SPS09. Previously created $TA tables will not have this column. It does not cause any problems, but obviously the parent/child information will not be available.	BIGINT

The `$TA` table can be partitioned.

Note

If the source table has a key field name identical to one of the standard fields in the `$TA` table, you will receive an error message after the `CREATE FULLTEXT INDEX` statement, prompting you to rename the field in the source table. Once you have renamed the corresponding field, you can execute the `CREATE FULLTEXT INDEX` statement again.

Note

SAP HANA will not display the same token (part-of-speech) types in the `$TA` table that are documented in the *SAP HANA Text Analysis Language Reference Guide*. The following table shows all of the `TA_TYPE` values that can be displayed in SAP HANA. This applies only to output from the `LINGANALYSIS_*` configurations.

HANA TA_TYPE Values (from $TA table)	Text Analysis Parts-of-Speech (from Language Reference)
abbreviation	Abbr
adjective	Adj, Adj-*
adverb	Adv, Adv-*
auxiliary verb	Aux, Modal
conjunction	Conj-*, Conj/*
determiner	Det, Det-*, Det/*, Art-*
interjection	Interj
noun	Nn, Nn-*

HANA TA_TYPE Values (from $TA table)	Text Analysis Parts-of-Speech (from Language Reference)
number	Num
particle	Part-*
preposition	Prep, Prep-*
pronoun	Pron, Pron-*
proper name	Prop, Symb
punctuation	Punct, Punct-*
verb	V-*, V/*
unknown	anything not listed above

3 Custom Text Analysis Configurations

Custom text analysis configurations are frequently used to incorporate custom text analysis dictionaries and extraction rule sets. You can customize the features and options used for text analysis by creating your own configuration files.

You can specify named entities with a large number of variations, aliases, and so on by creating custom text analysis dictionaries. Dictionaries also allow you to specify a standard name for each entity. For more complex entity types, text analysis rule sets might be a better choice.

You can specify your own entity types to be used with text analysis by creating custom text analysis extraction rules. Whereas text analysis dictionaries are ideal for specifying named entities, extraction rules enable you to identify more complex entity types, including events, relationships, etc. Extraction rules can leverage the full power of text analysis, including linguistic properties, core entities, and custom text analysis dictionaries.

For detailed information about text analysis dictionaries and extraction rules, refer to the *SAP HANA Text Analysis Extraction Customization Guide*.

Related Information

Managing Custom Text Analysis Configurations with the SAP HANA Repository [page 22]
Managing Custom Text Analysis Configurations with the SAP HANA Deployment Infrastructure [page 27]
Managing Custom Text Analysis Configurations with Stored Procedures [page 33]

3.1 Text Analysis Configuration File Syntax

Text analysis configurations are stored in XML format. They specify the text analysis processing steps to be performed and the options to use for each step. To edit text analysis configurations, you need some basic knowledge about the structure of XML files.

3.1.1 Complete Syntax of the Text Analysis Configuration File

To create or modify text analysis configurations, you need to understand the XML syntax and be aware of the options listed below.

Text analysis options are grouped into individual `<configuration>` elements, which usually relate to one step or component of the overall text analysis processing pipeline. Each configuration is identified by a unique `name`, which must be specified exactly as shown.

Some of the `<configuration>` elements contain `<property>` elements, which represent text analysis options that you can modify. The available options are described in subsequent topics, grouped by a configuration element.

> **Note**
>
> A `<configuration>` must still be included, even if it does not contain any `<property>` elements. It must be specified exactly as shown.

```
<?xml version="1.0" encoding="utf-8" ?>
<tasdk-configuration xmlns="http://www.sap.com/ta/config/4.0">
 <configuration name="SAP.TextAnalysis.AnalysisModel.AggregateAnalyzer.Aggregator">
   <property name="Analyzers" type="string-list">
     <string-list-
value>SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.F
C</string-list-value>
     <string-list-
value>SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA</
string-list-value>
     <string-list-
value>SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX</
string-list-value>
     <string-list-
value>SAP.TextAnalysis.DocumentAnalysis.Extraction.ExtractionAnalyzer.TF</string-
list-value>
     <string-list-
value>SAP.TextAnalysis.DocumentAnalysis.GrammaticalRoles.GrammaticalRoleAnalyzer.GR
A</string-list-value>
   </property>
 </configuration>
 <configuration name="CommonSettings" />
 <configuration
name="SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.F
C" based-on="CommonSettings" />
 <configuration
name="SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA"
based-on="CommonSettings">
   <property name="MinimumInputLength" type="integer">
     <integer-value>30</integer-value>
   </property>
   <property name="EvaluationSampleSize" type="integer">
     <integer-value>300</integer-value>
   </property>
   <property name="MinimumConfidence" type="integer">
     <integer-value>50</integer-value>
   </property>
 </configuration>
 <configuration
name="SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX"
based-on="CommonSettings">
   <property name="GetTokenStem" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
   <property name="EnableStemGuesser" type="boolean">
     <boolean-value>false</boolean-value>
   </property>
   <property name="GetTokenPartOfSpeech" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
   <property name="DisambiguatePartOfSpeech" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
   <property name="DisambiguateStem" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
```

```
   <property name="EnableCustomDictionaries" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
   <property name="VariantString" type="string">
     <string-value>expanded</string-value>
   </property>
 </configuration>
 <configuration
name="SAP.TextAnalysis.DocumentAnalysis.Extraction.ExtractionAnalyzer.TF" based-
on="CommonSettings">
   <property name="ExtractionRules" type="string-list">
     <string-list-value>RULE_SET_NAME</string-list-value>
   </property>
   <property name="Dictionaries" type="string-list">
     <string-list-value>DICTIONARY_NAME</string-list-value>
   </property>
 </configuration>
 <configuration
name="SAP.TextAnalysis.DocumentAnalysis.GrammaticalRoles.GrammaticalRoleAnalyzer.GR
A" based-on="CommonSettings">
   <property name="InputEntityCategories" type="string-list">
     <string-list-value>MWT_ABBR</string-list-value>
     <string-list-value>MWT_ADJ</string-list-value>
     <string-list-value>MWT_ADJ_COMP</string-list-value>
     <string-list-value>MWT_ADJ_ORD</string-list-value>
   </property>
   <property name="EnableDependencyParser" type="boolean">
     <boolean-value>true</boolean-value>
   </property>
 </configuration>
 <configuration name="PreProcessor">
   <property name="EntityTypes" type="string-list">
     <string-list-value> ENTITY_TYPE_NAME</string-list-value>
   </property>
   <property name="OutputMetadata" type="boolean">
    <boolean-value>false</boolean-value>
   </property>
 </configuration>
</tasdk-configuration>
```

3.1.2 SAP.TextAnalysis.AnalysisModel.AggregateAnalyzer.Aggregator

This configuration specifies the sequence of text analysis steps to be performed. You can decide to include or exclude the extraction and grammatical role analysis steps.

property name="Analyzers"

> **Note**
>
> All lines must appear exactly as shown.

The following lines specify the sequence of text analysis steps. They are mandatory in every configuration file.

```
<string-list-
value>SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.F
```

```
C</string-list-value>
<string-list-
value>SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA</
string-list-value>
<string-list-
value>SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX</
string-list-value>
```

The following line enables entity and relationship extraction, which includes custom dictionary extraction. If you only want linguistic analysis to be performed, which includes tokenization, identification of word base forms (stems), and tagging parts of speech, you can omit this line.

```
<string-list-
value>SAP.TextAnalysis.DocumentAnalysis.Extraction.ExtractionAnalyzer.TF</string-
list-value>
```

The following line enables the grammatical role analyzer. The primary goal of this analyzer is to identify functional relationships between elements in an input sentence.

```
<string-list-
value>SAP.TextAnalysis.DocumentAnalysis.GrammaticalRoles.GrammaticalRoleAnalyzer.GR
A</string-list-value>
```

3.1.3 SAP.TextAnalysis.DocumentAnalysis.StructureAnalysis.StructureAnalyzer.SA

This configuration specifies the options for automatic language detection.

property name="MinimumInputLength"

The `MinimumInputLength` option specifies the minimum input length for which automatic language identification is attempted. The default language is assumed for shorter inputs. For more information on the parameter LANGUAGE DETECTION, see *Full-Text Index Parameters*.

> **Note**
>
> The length is measured in Unicode UTF-16 code units, which usually equals the number of characters. However, inputs that make significant use of supplementary Unicode characters will use two UTF-16 code units for each supplementary character.

property name="EvaluationSampleSize"

To improve performance with large inputs, automatic language detection only examines a sample of the input text. Use this option, `EvaluationSampleSize`, to specify the size of the input sample used for language identification.

> **Note**
>
> The size is measured in Unicode UTF-16 code units, which usually equals the number of characters. However, inputs that make significant use of supplementary Unicode characters use two UTF-16 code units for each supplementary character.

property name="MinimumConfidence"

The `MinimumConfidence` option specifies the minimum confidence level required to accept the result of automatic language detection. The default language is assumed if the confidence falls below this level. For more information on the parameter LANGUAGE DETECTION, see *Full-Text Index Parameters*.

Values must be in the range from 0 to 100.

Related Information

Refer to chapter "Full-Text Index Parameters" in the SAP HANA Search Developer Guide

3.1.4 SAP.TextAnalysis.DocumentAnalysis.LinguisticAnalysis.LinguisticAnalyzer.LX

This configuration specifies the options for linguistic analysis.

property name="GetTokenStem"

The option `GetTokenStem` specifies whether word stems (base forms) are returned for each token.

Valid values are `true` and `false`.

property name="EnableStemGuesser"

The option `EnableStemGuesser` specifies whether word stems (base forms) are inferred ("guessed") for tokens that are not found in one of the text analysis lexicons.

Valid values are `true` and `false`.

property name="GetTokenPartOfSpeech"

The option `GetTokenPartOfSpeech` specifies whether the part of speech (for example, noun or verb) is returned for each token.

Valid values are `true` (default) and `false`.

property name="DisambiguatePartOfSpeech"

The option `DisambiguatePartOfSpeech` specifies whether the most probable part of speech should be chosen in cases where the part of speech is ambiguous.

Valid values are `true` and `false`.

> **Note**
>
> You should normally leave this property set to `true`, which is the default. SAP HANA will only display a single part of speech, regardless of the setting used. However, setting this property to `false` may allow additional word stems to be included in the full text index, which may, in turn, slightly improve search recall.

property name="DisambiguateStem"

The option `DisambiguateStem` specifies whether the most probable stem should be chosen in cases where the stem is ambiguous.

Valid values are `true` and `false` (default).

> **Note**
>
> You should normally leave this property set to `false`, which is the default. Setting this property to `true` also forces the `GetTokenPartOfSpeech` property to be `true`, since the part of speech is used to determine the most probable stem.

property name="EnableCustomDictionaries"

The option `EnableCustomDictionaries` is for internal SAP use only.

> **Note**
>
> This option should always be included and set to `true`.
>
> Do not confuse this option with the `Dictionaries` option in the configuration `SAP.TextAnalysis.DocumentAnalysis.Extraction.ExtractionAnalyzer.TF`.

property name="VariantString"

Text analysis supports alternate implementations of stemming and tokenization for many languages. The option `VariantString` can be used to specify which implementation to use.

Valid values for all languages are `std` (default) and `expanded`. Additional variants may be supported for selected languages. For more information about the variants supported by each language, see the *SAP HANA Text Analysis Language Reference Guide*.

> **Note**
>
> For search applications you should normally set this property to `expanded`. This implements more tolerant stemming in white-space languages, and more granular tokenization in non-white-space languages, which typically improves search recall.

3.1.5 SAP.TextAnalysis.DocumentAnalysis.Extraction.ExtractionAnalyzer.TF

This configuration specifies the options for entity and relationship extraction.

property name="ExtractionRules"

The option `ExtractionRules` specifies a list of text analysis extraction rule sets to be used for entity extraction.

> **Note**
>
> Only specify this property if you are actually using extraction rules. If you are not using text analysis extraction rules, omit the `ExtractionRules` property element completely. Otherwise, deployment fails if the `ExtractionRules` property is empty or blank.

Each string item in the list should be the name of a previously-deployed text analysis extraction rule set. The order of the rule sets does not matter.

> **Note**
>
> Normally, all of the rule sets listed in the `ExtractionRules` option are used for all inputs, regardless of their language. However, if the rule set name begins with a recognized language name followed by a hyphen, for example `german-myrules`, that rule set is used only for the specified language.

> **Note**
>
> Simplified and Traditional Chinese language modules provided by SAP are consolidated into modules that work for both languages simultaneously. For example, language modules used for Voice of the Customer

extraction in both Simplified and Traditional Chinese have names like `chinese-tf-voc-*.fsm` rather than `simplified-chinese-tf-voc-*.fsm` or `traditional-chinese-tf-voc-*.fsm`. When using an extraction rule set provided by SAP, the old naming convention (`simplified-chinese-*`, `traditional-chinese-*`) will continue to work, but you are encouraged to use the new names (`chinese-*`). Note that Simplified Chinese and Traditional Chinese remain distinct languages within text analysis, and in particular, language detection will continue to detect each language individually.

property name="Dictionaries"

The `Dictionaries` option specifies a list of text analysis dictionaries to be used for entity extraction.

> **Note**
>
> Only specify this property if you are actually using dictionaries. If you are not using custom text analysis dictionaries, omit the `Dictionaries` property element completely. Otherwise, deployment fails if the `Dictionaries` property is empty or blank.

Each string item in the list should be the name of a previously-deployed text analysis dictionary. The order of the dictionary names does not matter.

> **Note**
>
> Normally, all of the dictionaries listed in the `Dictionaries` option are used for all inputs, regardless of their language. However, if the dictionary name begins with a recognized language name followed by a hyphen, for example `german-mydictionary`, that dictionary is used only for the specified language.

> **Note**
>
> As with `Extraction Rules`, each Simplified and Traditional Chinese dictionary provided by SAP is consolidated into a single dictionary that works for both languages. For example, dictionaries used for Voice of the Customer extraction in both Simplified and Traditional chinese have names like `chinese-tf-voc-*.nc`. The old naming convention will continue to work, but you are encouraged to use the new names.

Related Information

SAP HANA Text Analysis Language Reference Guide

3.1.6 SAP.TextAnalysis.DocumentAnalysis.GrammaticalRoles.GrammaticalRoleAnalyzer.GRA

This configuration specifies the options for Grammatical Role Analysis.

property name="InputEntityCategories"

The option `InputEntityCategories` specifies which entity categories should use as input for grammatical role analysis. By default, none of the entity categories is used.

property name="EnableDependencyParser"

The option `EnableDependencyParser` indicates whether all grammatical roles should be extracted. Set this property to false to extract noun phrases only.

3.1.7 PreProcessor

This configuration specifies additional considerations for including data in the output of a text analysis process.

property name="EntityTypes"

The option `EntityTypes` specifies a list of entity types to be returned by entity extraction. If one or more entity types are provided, SAP HANA generates only entities that have one of the specified types. If no entity types are specified, or if the `EntityTypes` option is omitted, all supported entity types are returned.

Each string item in the list should be the fully-qualified name of the entity type. The order of type names does not matter.

> **Note**
>
> This configuration should only be set if only a subset of entity types is required in the output. By default all entity types are extracted.

> **Note**
>
> For a description of the available entity types for each language, refer to the *SAP HANA Text Analysis Language Reference Guide*.

property name="OutputMetadata"

The option `OutputMetadata` indicates whether document metadata is included in the `$TA` table. Document metadata values (such as Author, Date, Subject) are properties of the entire document. If metadata is required, the property value should be specified with a Boolean "true" value: `<boolean-value>true</boolean-value>`.

> **Note**
>
> If metadata is not required, the property value can be specified with a Boolean "false" value, or the property can simply be omitted. By default, metadata is omitted.

The following metadata properties are extracted:

- Author
- Date
- Date Created
- Date Modified
  Description
- Keyword
- Language
- Subject
- Title
- Version
- FromEmailAddress
- FromName
- ToEmailAddress
- ToName
- CcEmailAddress
- CcName
- BccEmailAddress
- BccName

property name="OutputLinguisticTokens"

The option `OutputLinguisticTokens` indicates whether linguistic output is included in the `$TA` table. Linguistic output includes tokens, word base forms (stems), and parts of speech. If linguistic output is required, the property value should be specified with a Boolean `true` value: `<boolean-value>true</boolean-value>`.

> **Note**
>
> If linguistic output is not required, the property value can be specified with a Boolean `false` value, or the property can simply be omitted. By default, linguistic output is omitted when extraction analysis or grammatical role analysis is enabled.

Related Information

SAP HANA Text Analysis Language Reference Guide

3.2 Text Analysis Dictionaries

You can specify your own entity types and entity names to be used with text analysis by creating custom text analysis dictionaries.

Text analysis dictionaries are ideal for specifying named entities with a large number of variations, aliases, and so on. Dictionaries also allow you to specify a standard name for each entity. For more complex entity types, text analysis rule sets might be a better choice.

A dictionary is stored in a single file using an XML syntax.

The file extension must be `.hdbtextdict`. If the dictionary name begins with a recognized language name followed by a hyphen, `german-mydictionary` for example the dictionary is used only for inputs in the language specified (in this case German). Otherwise, the dictionary is used for all inputs, regardless of their language.

> **Note**
>
> The recognized language names are as follows: Arabic, Bokmal, Catalan, Croatian, Czech, Danish, Dutch, English, Farsi, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Nynorsk, Polish, Portuguese, Romanian, Russian, Serbian, Serbian-lat, Simplified-Chinese, Slovak, Slovenian, Spanish, Swedish, Thai, Traditional-Chinese, Turkish.

For a complete description of the text analysis dictionary syntax, see the *SAP HANA Text Analysis Extraction Customization Guide*.

The keyword dictionaries used by text analysis to identify and classify sentiments can also be customized for specific applications if required. These dictionaries are used by the standard `EXTRACTION_CORE_VOICEOFCUSTOMER` text analysis configuration provided with SAP HANA.

> **Note**
>
> The keyword dictionaries are not installed by default. See *Obtaining Predefined Text Analysis Configurations* for instructions on how to obtain the standard text analysis configurations that are shipped with SAP HANA.

For more information, see the *SAP HANA Text Analysis Extraction Customization Guide* about customizing the sentiment analysis feature.

3.3 Text Analysis Extraction Rules

You can specify your own entity types to be used with text analysis by creating custom text analysis extraction rules. Whereas text analysis dictionaries are ideal for specifying named entities, extraction rules enable you to

identify more complex entity types, including events, relationships, etc. Extraction rules can leverage the full power of text analysis, including linguistic properties, core entities, and custom text analysis dictionaries.

Several rules are included in a rule set which is stored as a single file. The file extension must be `.hdbtextrule`. If the rule file name begins with a recognized language name followed by a hyphen, for example `german-myrules.hdbtextrule`, the rule set is used only for inputs in that language (in this case, German). Otherwise, the rule set is used for all inputs, regardless of their language.

> **Note**
>
> The following language names are recognized: Arabic, Bokmal, Catalan, Croatian, Czech, Danish, Dutch, English, Farsi, French, German, Greek, Hebrew, Hungarian, Indonesian, Italian, Japanese, Korean, Nynorsk, Polish, Portuguese, Romanian, Russian, Serbian, Serbian-lat, Simplified Chinese, Slovak, Slovenian, Spanish, Swedish, Thai, Traditional Chinese, Turkish.

For complex rule sets, you can divide your rules into multiple files and use rule directives to combine them during deployment. Text analysis defines three different types of rule files:

Extension of Rule File	Description
`.hdbtextrule`	Defines the top-level rule sets. These are the objects that are referenced in text analysis configurations. They are compiled during deployment, and may include one or more of the rule file types `.hdbtextinclude` and `.hdbtextlexicon`.
`.hdbtextinclude`	Defines rule definitions to be used in one or more top-level rule sets using the `#include` directive. These are not compiled during deployment. Instead they are compiled during the deployment of the including top-level rule set.
`.hdbtextlexicon`	Includes word lists to be used in one or more top-level rule sets using the `#lexicon` directive. These are not compiled during deployment. Instead they are compiled during the deployment of the including top-level rule set.

All `#include` and `#lexicon` statements must appear at the top of the rule file and the # must be the first non-space character on the line. As soon as a line is encountered that does not begin with `#include` or `#lexicon`, no further #include or #lexicon will be processed, which will likely result in compilation errors during deployment of the top-level rule set. The `#include` and `#lexicon` directives must specify the complete name of the rule file to be included (together with the package or namespace), even if the files reside in the same location as the including rule file.

Example

Assuming your rule files are all located in the `my.ta.rules` namespace (or package), and rule set `main` includes shared rules from `common-rules` and a word list from `common-words`, your `main.hdbtextrule` file should begin with the following lines:

Sample Code

```
#include <my.ta.rules::common-rules.hdbtextinclude>
#include <my.ta.rules::common-words.hdbtextlexicon>
```

For a complete description of the text analysis rule file syntax, see the *SAP HANA Text Analysis Extraction Customization Guide*.

3.4 Managing Custom Text Analysis Configurations with the SAP HANA Repository

Custom text analysis configuration files can be stored in the HANA Repository and edited using standard HANA development tools just like any other design-time development objects. You must activate the text analysis configuration objects before you can use them.

Artifact References

All SAP HANA Repository artifacts belong to packages. You define your project's package hierarchy when you create your project and when you organize your content into folders within your project. When one artifact refers to another artifact, such as when a text analysis configuration refers to a rule set or dictionary, the reference must be fully qualified. This means that the reference must contain the full package name followed by a double colon, "::", followed by the artifact name.

Sample Code

```
<property name="Dictionaries" type="string-list">
  <string-list-value>acme.myproject.ta::MyDictionary.hdbtextdict</string-list-
value>
</property>
```

3.4.1 Creating a Text Analysis Configuration with the SAP HANA Repository

You can create a new text analysis configuration file using the New/File wizard in the SAP HANA studio.

Prerequisites

- You have created a development workspace.
- You have created and shared a project.

Context

To create a new text analysis configuration from scratch, perform the following steps.

Procedure

1. In the Project Explorer view in the SAP HANA Development perspective, right-click the project for which you want to create the new configuration and, from the context menu, choose New/File.
2. In the wizard, enter or select a parent folder and enter the file name. The file extension must be .hdbtextconfig. Choose Finish. Your text analysis configuration file is created locally. Your configuration opens as an empty file in the text editor.

   > **Note**
   >
   > You can also create a folder first and add a file. To do so, right-click the project name and choose New/Folder. The New Folder wizard appears. Enter or select the project, enter the folder name, and choose Finish.

3. Enter your text analysis configuration options in your new file and save it locally. At this point, your text analysis configuration has not been committed or activated.

   > **Note**
   >
   > To avoid typing in the complete syntax, you can also copy the contents of one of the standard text analysis configurations that ships with SAP HANA. See Obtaining Predefined Text Analysis Configurations for instructions on how to obtain the standard text analysis configurations that ship with HANA.

4. To commit your new configuration or make changes to an existing one, save it, open the context menu for the configuration file, and choose Team and then Commit. Your configuration is now synchronized with the repository as a design time object, and the icon shows that your configuration is committed.
5. When you have finished editing your configuration and you are ready to activate it, open the context menu for the configuration file and choose Team/Activate. Your configuration is created in the repository as a runtime object, and the icon shows that your configuration is activated. This allows you and other users to use the configuration for text analysis.

   > **Note**
   >
   > You can also activate your configuration at the project and folder levels.

3.4.2 Creating Custom Text Analysis Rule Sets with the SAP HANA Repository

You can create a new text analysis rule file using the *New/File* wizard in the SAP HANA Studio.

Prerequisites

- You have created a development workspace.
- You have created and shared a project.

> **Note**
>
> You can also share your project after you create your rule specification files.

Context

You can specify your own entity types to be used with text analysis by creating custom text analysis extraction rules. Whereas text analysis dictionaries are ideal for specifying named entities, extraction rules enable you to identify more complex entity types, including events, relationships, etc. Extraction rules can leverage the full power of text analysis, including linguistic properties, core entities, and custom text analysis dictionaries.

Several rules are included in a rule set which is stored as a file in the SAP HANA repository.

Procedure

1. In the SAP HANA Studio open the SAP HANA Development perspective. In the *Project Explorer* view choose the project to contain the new rule set and choose *New/File* from the context menu.
2. In the wizard, enter or select a parent folder and enter the rule file name. The file extension must be `.hdbtextrule`. Choose *Finish*.

   Your text analysis rule file is created locally and opens as an empty file in the text editor.

   > **Note**
   >
   > You can also create a folder first and add a file. Open the context menu of the project name and choose *New/Folder*. In the wizard enter or select the project, enter the folder name, and choose *Finish*.
3. Enter your text analysis rules set specification into your new file and save it locally. At this point, your text analysis rules are not committed and not activated.
4. To commit your new rule set or make changes to an existing one, save it, open the context menu for the rule file, and choose *Team*, and then *Commit*.

Your rule set is now synchronized with the repository as a design time object and the icon shows that your rule set has been committed.

5. When you have finished editing your rule set and you are ready to activate it, open the context menu for the rule file and choose Team/Activate.

   Your rule set is created in the repository as a runtime object and the icon shows that your rule set has been activated. Activation allows you and other users to use the rule set for text analysis.

   You can also activate your rule set at the project and folder levels.

Next Steps

Reference your rule set in a custom text analysis configuration.

Related Information

Creating a Text Analysis Configuration with the SAP HANA Repository [page 22]

3.4.3 Creating Custom Text Analysis Dictionaries with the SAP HANA Repository

You can create a new text analysis dictionary file using the *New/File* wizard in the SAP HANA studio.

Prerequisites

- You have created a development workspace.
- You have created and shared a project.

> **Note**
>
> You can also share your project after creating your dictionary specification file.

Context

Text analysis dictionaries are ideal for specifying named entities with a large number of variations, aliases, and so on. Dictionaries also allow you to specify a standard name for each entity. For more complex entity types, text analysis rule sets might be a better choice.

Procedure

1. In the *Project Explorer* view in the SAP HANA Development perspective, choose the project that you want to create the new dictionary in and choose *New/File* from the context menu.
2. In the wizard, enter or select a parent folder and enter the dictionary file name. The file extension must be `.hdbtextdict`. Choose *Finish*.

   Your text analysis dictionary file is created locally. Your dictionary file opens as an empty file in the text editor.

   > **Note**
   >
   > You can also create a folder first, and add a file to it subsequently. Right-click the project name and choose *New/Folder*. The New Folder wizard appears. Enter or select the project, enter the folder name, and choose *Finish*.

3. Enter your text analysis dictionary specification into your new file and save it locally.

   At this point, your text analysis dictionary is not committed or activated.
4. To commit your new dictionary or make changes to an existing one, save it, open the context menu for the dictionary file, and choose *Team*, and then *Commit*.

   Your dictionary is now synchronized with the repository as a design time object, and the icon shows that your dictionary has been committed.
5. When you have finished editing your dictionary and are ready to activate it, open the context menu for the dictionary file and choose Team/Activate.

   Your dictionary is created in the repository as a runtime object, and the icon shows that your dictionary has been activated. This allows you and other users to use the dictionary for text analysis.

   > **Note**
   >
   > You can also activate your dictionary at the project and folder levels.

Next Steps

Reference your custom dictionary in your custom text analysis configuration.

Related Information

Text Analysis Configuration File Syntax [page 10]

3.5 Managing Custom Text Analysis Configurations with the SAP HANA Deployment Infrastructure

Custom text analysis configuration files can be added to an XS Advanced Application Project just like any other database artifacts. The text analysis configuration objects will become available for use when the application is deployed. Database artifacts in XS Advanced are deployed by the SAP HANA Deployment Infrastructure.

XS Advanced Applications are described in the SAP HANA Developer Guide for SAP HANA XS Advanced Model. Refer to that guide for more information about setting up XS Advanced projects and about defining database artifacts in XS Advanced.

Artifact References

XS Advanced Application projects usually use namespaces to organize artifacts. Namespaces are defined in the runtime namespace configuration (`.hdinamespace`) files. When one artifact refers to another artifact, such as when a text analysis configuration refers to a rule set or dictionary, the reference must be fully qualified. This means that the reference must contain the namespace followed by a double colon, "::", followed by the artifact name.

Sample Code

```
<property name="Dictionaries" type="string-list">
  <string-list-value>acme.myproject.ta::MyDictionary.hdbtextdict</string-list-
value>
</property>
```

It is possible to omit namespace names in the runtime namespace configuration (`.hdinamespace`) files by using a blank namespace name. If your project omits namespaces, then artifact references should omit the namespace and double colon, and consist of just the artifact name.

Sample Code

```
<property name="Dictionaries" type="string-list">
  <string-list-value>MyDictionary.hdbtextdict</string-list-value>
</property>
```

Related Information

SAP HANA Developer Guide for SAP HANA Studio

3.5.1 Creating a Text Analysis Configuration with SAP HANA DI

You can add a new text analysis configuration to an XS Advanced project.

Prerequisites

- You have created an XS Advanced project.
- Your project includes at least the following:
  - the deployment descriptor files (`manifest.yml` and `mtad.yaml`)
  - a folder for database artifacts, which includes a `package.json` file containing details of dependencies, the container configuration (`.hdiconfig`), and runtime namespace configuration (`.hdinamespace`) files
- You have created an HDI container and bound the HDI container to the application that you want to use.

Context

To create and deploy a new text analysis configuration, perform the following steps.

Procedure

1. Add the `com.sap.hana.di.textconfig` HDI plugin to the container configuration (`.hdiconfig`).
2. Save your text analysis configuration file in the folder for database artifacts. Your file should use `.hdbtextconfig` as its file extension.
3. Deploy your project.

Results

Your configuration is now stored in the SAP HANA database.

Sample Code

Sample code for the container configuration file:

```
"hdbtextconfig" : {
"plugin_name" : "com.sap.hana.di.textconfig",
"plugin_version": "12.0.0"
}
```

Sample Code

Sample code for project folder structure:

```
<MyAppName>
|- db/                           # Database deployment artifacts
| |- package.json                # Database details/dependencies
| |- src/                        # Database artifacts: tables, views, etc.
| | |- .hdiconfig                # HDI build plug-in configuration
| | |- .hdinamespace             # HDI run-time name-space configuration
| \ \- myTAConfig.hdbtextconfig # Text Analysis Configuration definition
|- manifest.yml
\- mtad.yaml
```

Next Steps

Create a fulltext index definition that references your configuration, and add it to your project.

Related Information

Text Analysis Configuration File Syntax [page 10]

3.5.2 Creating Custom Text Analysis Rule Sets with SAP HANA DI

You can specify your own entity types to be used with text analysis by creating custom text analysis extraction rules. Whereas text analysis dictionaries are ideal for specifying named entities, extraction rules enable you to identify more complex entity types, including events, relationships, etc. Extraction rules can leverage the full power of text analysis, including linguistic properties, core entities, and custom text analysis dictionaries. Several rules are included in a rule set.

Prerequisites

- You have created an XS Advanced project.
- Your project has at least the following:
  - The deployment descriptor files (`manifest.yml` and `mtad.yaml`)
  - A folder for database artifacts which includes a `package.json` file containing details of dependencies, the container configuration (`.hdiconfig`), and run-time namespace configuration (`.hdinamespace`) files
- You have created an HDI container and bound the HDI container to the application that you want to use it.

Context

To create and deploy a new text analysis rule set, perform the following steps.

Procedure

1. Add the `com.sap.hana.di.textrule` HDI plugin to the container configuration (`.hdiconfig`).
2. (Optional): add the `com.sap.hana.di.textrule.include` and `com.sap.hana.di.textrule.lexicon` HDI plugins to the container configuration (`.hdiconfig`).
3. Save your text analysis rule set file in the folder for database artifacts. Your file should use `.hdbtextrule` as its file extension.
4. (Optional): save any text analysis rule include and lexicon files in the folder for database artifacts. Include files should use the extension `.hdbtextinclude` and lexicon files should use `.hdbtextlexicon`.
5. Deploy your project.

Results

Your rule set is now stored in the SAP HANA database.

Sample Code

Sample code for the container configuration file:

```
"hdbtextrule" : {
"plugin_name" : "com.sap.hana.di.textrule",
"plugin_version": "12.0.0"
},
"hdbtextinclude" : {
"plugin_name" : "com.sap.hana.di.textrule.include",
"plugin_version": "12.0.0"
},
"hdbtextlexicon" : {
"plugin_name" : "com.sap.hana.di.textrule.lexicon",
"plugin_version": "12.0.0"
}
```

Sample Code

Sample code for project folder structure:

```
<MyAppName>
|- db/                          # Database deployment artifacts
| |- package.json               # Database details/dependencies
| |- src/                       # Database artifacts: tables, views, etc.
| | |- .hdiconfig               # HDI build plug-in configuration
| | |- .hdinamespace            # HDI run-time name-space configuration
| | |- myRule.hdbtextrule       # Text Analysis Rule definition
| | |- myInclude.hdbtextinclude # Text Analysis Rule include (optional)
| \ \- myLexicon.hdbtextlexicon # Text Analysis Word List (optional)
```

```
|- manifest.yml
\- mtad.yaml
```

Next Steps

Create a text analysis configuration that references your rule set, and add it to your project.

Related Information

Creating a Text Analysis Configuration with SAP HANA DI [page 28]

3.5.3 Creating Custom Text Analysis Dictionaries with SAP HANA DI

Prerequisites

- You have created an XS Advanced project.
- Your project has at least the following:
  - The deployment descriptor files (`manifest.yml` and `mtad.yaml`)
  - A folder for database artifacts which includes a `package.json` file containing details of dependencies, the container configuration (`.hdiconfig`), and run-time namespace configuration (`.hdinamespace`) files
- You have created an HDI container and bound the HDI container to the application that you want to use it.

Context

To create and deploy a new text analysis dictionary, perform the following steps.

Procedure

1. Add the `com.sap.hana.di.textdictionary` HDI plugin to the container configuration (`.hdiconfig`).
2. Save your text analysis dictionary file in the folder for database artifacts. Your file should have `.hdbtextdict` as its file extension.

3. Deploy your project.

Results

Your dictionary is now stored in the HANA database.

Sample Code

Sample code for the container configuration file:

```
"hdbtextdict" : {
"plugin_name" : "com.sap.hana.di.textdictionary",
"plugin_version": "12.0.0"
}
```

Sample Code

Sample code for project folder structure:

```
<MyAppName>
|- db/                           # Database deployment artifacts
| |- package.json                # Database details/dependencies
| |- src/                        # Database artifacts: tables, views, etc.
| | |- .hdiconfig                # HDI build plug-in configuration
| | |- .hdinamespace             # HDI run-time name-space configuration
| \ \- myDictionary.hdbtextdict # Text Analysis Dictionary definition
|- manifest.yml
\- mtad.yaml
```

Next Steps

Create a text analysis configuration that references your dictionary, and add it to your project.

Related Information

Creating a Text Analysis Configuration with SAP HANA DI [page 28]
Creating Custom Text Analysis Rule Sets with SAP HANA DI [page 29]

3.6 Managing Custom Text Analysis Configurations with Stored Procedures

You can use stored procedures to create and manage configurations and other resources for customizing text analysis.

Artifact References

When one artifact refers to another artifact, such as when a text analysis configuration refers to a rule set or dictionary, the reference should simply use the name that was used when the other artifact was created.

Sample Code

```
<property name="Dictionaries" type="string-list">
  <string-list-value>MyDictionary.hdbtextdict</string-list-value>
</property>
```

If you wish to use hierarchical names for your artifacts, like the names used with the SAP HANA Repository and SAP HANA DI, simply use a fully qualified name when you create your artifact. A fully qualified name is a dot separated package/namespace name, followed by a double colon, "::", followed by a simple name. For example, you could create a dictionary with the name `acme.myproject.ta::MyDictionary`, and then reference it with that name in a configuration as follows:

Sample Code

```
<property name="Dictionaries" type="string-list">
  <string-list-value>acme.myproject.ta::MyDictionary.hdbtextdict</string-list-
value>
</property>
```

3.6.1 Stored Procedures for Managing Text Analysis and Text Mining Resources

TEXT_CONFIGURATION_CREATE

Create or update a configuration or other resource for customizing text analysis and text mining.

Parameters

Parameter Name	Data Type	Description
SCHEMA_NAME	NVARCHAR (256)	Specifies the name of the schema that will contain the resource.
NAME	NVARCHAR (256)	Specifies the name of the resource.
TYPE	VARCHAR (16)	Specifies the resource type.
DATA	BLOB	Specifies the resource content.

Values of the TYPE Parameter

Type	Description of Resource
hdbtextconfig	A text analysis configuration
hdbtextdict	A text analysis dictionary
hdbtextrule	A text analysis rule set
hdbtextinclude	A text analysis rule set include
hdbtextlexicon	A text analysis rule set word list
textminingconfig	A text mining configuration

TEXT_CONFIGURATION_DROP

Delete an existing resource so it can no longer be used.

Parameters

Parameter Name	Data Type	Description
SCHEMA_NAME	NVARCHAR (256)	Specifies the name of the schema that contains the resource.
NAME	NVARCHAR (256)	Specifies the name of the resource.
TYPE	VARCHAR (16)	Specifies the resource type.

TEXT_CONFIGURATION_CLEAR

Notify the system that custom resources have changed.

Parameters

Parameter Name	Data Type	Description
SCHEMA_NAME	NVARCHAR (256)	Specifies the name of the schema that contains the resource.
NAME	NVARCHAR (256)	Optional: Specifies the name of the resource.
TYPE	VARCHAR (16)	Optional: Specifies the resource type.

Related Information

Dropping Custom Text Analysis Resources with Stored Procedures [page 39]
Notifying the System of Changes with Stored Procedures [page 41]
Creating a Text Analysis Configuration with Stored Procedures [page 35]
Creating Custom Text Analysis Rule Sets with Stored Procedures [page 36]
Creating Custom Text Analysis Dictionaries with Stored Procedures [page 38]

3.6.2 Creating a Text Analysis Configuration with Stored Procedures

You can create a new text analysis configuration using the TEXT_CONFIGURATION_CREATE stored procedure.

Prerequisites

You have created a schema.

Context

To create a new text analysis configuration from scratch, perform the following steps:

Procedure

1. In SAP HANA studio, right click your system and select *Open SQL Console*.
2. In the console, type the following, replacing <schema_name> with your schema's name, replacing <name> with the name for your new configuration, and replacing <configuration> with your configuration.

```
CALL TEXT_CONFIGURATION_CREATE('<schema_name>', '<name>', 'hdbtextconfig',
'<configuration>');
```

3. Click *Execute*.

Results

Your configuration is now stored in the HANA database and will appear in the `TEXT_CONFIGURATIONS` system view.

Next Steps

> **Note**
>
> After creating a configuration, you can update it by making another call to `TEXT_CONFIGURATION_CREATE` with the updated configuration content. After updating a configuration, notify the system of your changes by calling `TEXT_CONFIGURATION_CLEAR`.

Related Information

Stored Procedures for Managing Text Analysis and Text Mining Resources [page 33]
Text Analysis Configuration File Syntax [page 10]
Notifying the System of Changes with Stored Procedures [page 41]
Obtaining Predefined Text Analysis Configurations [page 42]

3.6.3 Creating Custom Text Analysis Rule Sets with Stored Procedures

Prerequisites

You have created a schema.

Context

You can specify your own entity types to be used with text analysis by creating custom text analysis extraction rules. Whereas text analysis dictionaries are ideal for specifying named entities, extraction rules enable you to identify more complex entity types, including events, relationships, etc. Extraction rules can leverage the full power of text analysis, including linguistic properties, core entities, and custom text analysis dictionaries.

Several rules are included in a rule set.

To create a new text analysis rule set from scratch, perform the following steps:

Procedure

1. In SAP HANA studio, right click your system and select *Open SQL Console*.
2. 2. In the console, type the following, replacing `<schema_name>` with your schema's name, replacing `<name>` with the name for your new rule set, and replacing `<rules>` with your rule set specification.

   ```
   CALL TEXT_CONFIGURATION_CREATE('<schema_name>', '<name>', 'hdbtextrule',
   '<rules>');
   ```
3. Click *Execute*.

Results

Your rule set has been compiled to a binary format and is now stored in the SAP HANA database and will appear in the `TEXT_CONFIGURATIONS` system view with the type ‘textrule' that represents compiled rule sets.

Next Steps

> **Note**
>
> After creating a rule set, you can update it by making another call to `TEXT_CONFIGURATION_CREATE` with the updated rule set specification. After updating a rule set, notify the system of your changes by calling `TEXT_CONFIGURATION_CLEAR`.

> **Note**
>
> If your rule set references any Text Analysis Include files or Text Analysis Word Lists, create those first using `TEXT_CONFIGURATION_CREATE` and using type 'hdbtextinclude' or 'hdbtextlexicon'. Once your rule is created, the include files and lexicons are no longer needed and can be dropped using the `TEXT_CONFIGURATION_DROP` procedure.

Reference your rule set in a custom text analysis configuration.

Related Information

Creating a Text Analysis Configuration with Stored Procedures [page 35]
Managing Custom Text Analysis Configurations with Stored Procedures [page 33]
Notifying the System of Changes with Stored Procedures [page 41]

3.6.4 Creating Custom Text Analysis Dictionaries with Stored Procedures

You can create a new text analysis dictionary using the `TEXT_CONFIGURATION_CREATE` stored procedure.

Prerequisites

You have created a schema.

Context

Text analysis dictionaries are ideal for specifying named entities with a large number of variations, aliases, and so on. Dictionaries also allow you to specify a standard name for each entity. For more complex entity types, text analysis rule sets might be a better choice.

Procedure

1. In SAP HANA studio, right click your system and select `Open SQL Console`.
2. In the console, type the following, replacing `<schema_name>` with your schema's name, replacing `<name>` with the name for your new dictionary, and replacing `<dictionary>` with your dictionary definition.

   ```
   CALL TEXT_CONFIGURATION_CREATE('<schema_name>', '<name>', 'hdbtextdict',
   '<dictionary>');
   ```
3. Click *Execute*.

Results

Your dictionary has been compiled to a binary format and is now stored in the HANA database and will appear in the `TEXT_CONFIGURATIONS` system view with the type 'textdict' that represents compiled dictionaries.

> **Note**
>
> After creating a dictionary, you can update it by making another call to `TEXT_CONFIGURATION_CREATE` with the updated dictionary definition. After updating a dictionary, notify the system of your changes by calling `TEXT_CONFIGURATION_CLEAR`.

Next Steps

Reference your dictionary in a custom text analysis configuration.

Related Information

Stored Procedures for Managing Text Analysis and Text Mining Resources [page 33]
Creating a Text Analysis Configuration with Stored Procedures [page 35]
Creating Custom Text Analysis Rule Sets with Stored Procedures [page 36]
Notifying the System of Changes with Stored Procedures [page 41]

3.6.5 Dropping Custom Text Analysis Resources with Stored Procedures

You can drop custom configurations, rule sets, and dictionaries once you are no longer using them for any fulltext indexes.

Prerequisites

You have created a text analysis configuration, rule set, or dictionary.

Context

To drop a text analysis resource, perform the following steps:

Procedure

1. In SAP HANA studio, right click your system and select *Open SQL Console*.
2. In the console, type the following, replacing `<schema_name>` with your schema's name, replacing `<name>` with the name of your resource, and replacing `<type>` with the type of your resource, 'hdbtextconfig', 'textrule', or 'textdict'.

   ```
   CALL TEXT_CONFIGURATION_DROP('<schema_name>', '<name>', '<type>');
   ```
3. Click *Execute*.

Results

Your configuration has been removed from the HANA database and will no longer appear in the `TEXT_CONFIGURATIONS` system view.

> **Note**
>
> The type used when dropping a rule set or dictionary should be 'textrule' or 'textdict'. This is the type of the compiled resource that appears in the `TEXT_CONFIGURATIONS` system view, and differs from the type you used when you created the resource.

Related Information

Stored Procedures for Managing Text Analysis and Text Mining Resources [page 33]

3.6.6 Notifying the System of Changes with Stored Procedures

After changing or dropping custom configurations, rule sets, and dictionaries, you should notify the system to stop using the old versions.

Prerequisites

You have updated or dropped one or more text analysis configurations, rule sets, or dictionaries.

Context

The system will continue to use old versions of configurations, dictionaries and rules until you notify the system of changes. To notify the system of changes, perform the following steps:

Procedure

1. In SAP HANA studio, right click your system and select *Open SQL Console*.
2. In the console, type the following, replacing `<schema_name>` with your schema's name.

   ```
   CALL TEXT_CONFIGURATION_CLEAR('<schema_name>');
   ```
3. Click *Execute*.

Results

The system has been notified of your changes.

> **Note**
>
> When changing multiple configurations, rule sets or dictionaries from the same schema, make all your changes, and then call `TEXT_CONFIGURATION_CLEAR` for the schema just once after all changes are complete.

Related Information

Stored Procedures for Managing Text Analysis and Text Mining Resources [page 33]

3.7 Obtaining Predefined Text Analysis Configurations

The system includes a number of predefined text analysis configurations. To avoid typing in the complete XML syntax into your own custom configurations, you can copy and paste the contents of one of the standard text analysis configurations into your file.

The standard text analysis configurations are automatically installed into the SAP HANA Repository as part of the standard SAP HANA installation from the `HANA_TA_CONFIG` delivery unit. To copy one of the standard configuration files, go to the SAP HANA Repositories view in the SAP HANA Development perspective and navigate to the `sap.hana.ta.config` package. Open one of the standard configuration files, copy the complete contents of the file, and paste it into your new configuration file.

> **Note**
>
> You should not modify the standard text analysis configuration files. Instead, you should always copy the contents of the standard configuration into a new file before making any modifications.

The keyword dictionaries used by text analysis to identify and classify sentiments can also be customized for specific applications, if needed. These dictionaries are used by the standard `EXTRACTION_CORE_VOICEOFCUSTOMER` text analysis configuration provided with SAP HANA. However, the keyword dictionaries are not installed by automatically during the installation of SAP HANA.

The keyword dictionaries are contained in a separate delivery unit named `HANA_TA_VOC`, which must be manually imported into the SAP HANA Repository by a SAP HANA administrator using the SAP HANA Lifecycle Management tools or SAP HANA Studio. Once the `HANA_TA_VOC` delivery unit has been imported, the keyword dictionaries will be located in the `sap.hana.ta.voc` package in the SAP HANA repository.

> **Note**
>
> You should not modify the standard keyword dictionary files. Instead, you should always copy the contents of a standard dictionary file into a new file before making any modifications.

Related Information

SAP HANA Text Analysis Developer Guide [page 3]

4 Using the Text Analysis XS API

Text Analysis can be used via the SAP HANA Extended Application Services (SAP HANA XS) API.

To create a SAP HANA XS application, you set up a project in SAP HANA Studio. You use JavaScript to develop your application.

To run your application, you can use any internet browser and open the application URL on the SAP HANA server which is hosting your application.

To load data into the database tables you use the standard SAP HANA tools.

Related Information

SAP HANA Text Analysis XS JS API

4.1 Text Analysis XS API Example Application

The example describes an SAP HANA XS API JavaScript application which analyzes text using the `EXTRACTION_CORE` configuration and then displays the results.

Sample Code

```
// Create a text analysis session that uses an out-of-the-box configuration
EXTRACTION_CORE
var oTextAnalysisSession  = new
$.text.analysis.Session({configuration:'EXTRACTION_CORE'});
// Input text to be analyzed
var sText = "New York, New York, this city's a dream";
// Call the analyze method. Explicitly set the language, although the default
is English anyway
var oAnalysisResult = oTextAnalysisSession.analyze({inputDocumentText: sText,
language: 'en'});
// Send the results back
$.response.contentType = 'text/json';
$.response.setBody(JSON.stringify(oAnalysisResult));
```

5 Important Disclaimer for Features in SAP HANA Platform, Options and Capabilities

SAP HANA server software and tools can be used for several SAP HANA platform and options scenarios as well as the respective capabilities used in these scenarios. The availability of these is based on the available SAP HANA licenses and the SAP HANA landscape, including the type and version of the back-end systems the SAP HANA administration and development tools are connected to. There are several types of licenses available for SAP HANA. Depending on your SAP HANA installation license type, some of the features and tools described in the SAP HANA platform documentation may only be available in the SAP HANA options and capabilities, which may be released independently of an SAP HANA Platform Support Package Stack (SPS). Although various features included in SAP HANA options and capabilities are cited in the SAP HANA platform documentation, each SAP HANA edition governs the options and capabilities available. Based on this, customers do not necessarily have the right to use features included in SAP HANA options and capabilities. For customers to whom these license restrictions apply, the use of features included in SAP HANA options and capabilities in a production system requires purchasing the corresponding software license(s) from SAP. The documentation for the SAP HANA options is available in SAP Help Portal. If you have additional questions about what your particular license provides, or wish to discuss licensing features available in SAP HANA options, please contact your SAP account team representative.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: https://help.sap.com/viewer/disclaimer).

go.sap.com/registration/contact.html