Content

1  Text Analysis. ................................................................. 4

2  Structure of the $TA Table. ............................................. 7

3  Custom Text Analysis Configurations. .............................. 11
   3.1 Text Analysis Configuration File Syntax. ...................... 11
      Complete Syntax of the Text Analysis Configuration File. ... 11
      PreProcessor. .......................................................... 19
   3.2 Text Analysis Dictionaries. ....................................... 21
   3.3 Text Analysis Extraction Rules. .................................. 22
   3.4 Managing Custom Text Analysis Configurations with the SAP HANA Repository. ........... 23
      Creating a Text Analysis Configuration with the SAP HANA Repository. ............... 24
      Creating Custom Text Analysis Rule Sets with the SAP HANA Repository. .............. 25
      Creating Custom Text Analysis Dictionaries with the SAP HANA Repository. .......... 27
   3.5 Managing Custom Text Analysis Configurations with XS Advanced. ......................... 28
      Getting Started. ..................................................... 29
      Creating Custom Text Analysis Dictionaries with XS Advanced. ......................... 31
      Creating Custom Text Analysis Rule Sets with XS Advanced. ............................. 34
      Testing Dictionaries and Rules with XS Advanced. ....................................... 37
      Creating a Text Analysis Configuration with XS Advanced. .................................. 42
      Packaging and Deploying Text Analysis Customizations. .................................. 44
   3.6 Managing Custom Text Analysis Configurations with Stored Procedures. .................. 44
      Stored Procedures for Managing Text Analysis and Text Mining Resources. ............ 45
      Creating a Text Analysis Configuration with Stored Procedures. ......................... 47
      Creating Custom Text Analysis Rule Sets with Stored Procedures. ....................... 48
      Creating Custom Text Analysis Dictionaries with Stored Procedures. .................... 50
      Dropping Custom Text Analysis Resources with Stored Procedures. ..................... 51
      Notifying the System of Changes with Stored Procedures. .................................. 52
   3.7 Obtaining Predefined Text Analysis Configurations. .................. 53

4  Using the Text Analysis SQL API. ................................ 55
   4.1 Performing Linguistic Analysis Using TA_ANALYZE. .............. 57
4.2 Performing Entity Extraction Using TA_ANALYZE ................................................. 57

5 Using the Text Analysis XS API .................................................................................. 58
5.1 Text Analysis XS API Example Application ................................................................. 58
5.2 Text Analysis XS Advanced API ................................................................................ 58
    Text Analysis XS Advanced API Example Application .................................................. 59
5.3 Text Analysis XSJS Compatibility Layer ................................................................. 59
1 Text Analysis

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:

<table>
<thead>
<tr>
<th>Name of Text Analysis Configuration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINGANALYSIS_BASIC</td>
<td>This configuration provides the following language processing capabilities for linguistic analysis of unstructured data:</td>
</tr>
<tr>
<td></td>
<td>● Segmentation, also known as tokenization - the separation of input text into its elements</td>
</tr>
<tr>
<td>LINGANALYSIS_STEMS</td>
<td>This configuration provides the following language processing capabilities for linguistic analysis of unstructured data:</td>
</tr>
<tr>
<td></td>
<td>● Segmentation, also known as tokenization - the separation of input text into its elements</td>
</tr>
<tr>
<td></td>
<td>● Stemming - the identification of word stems or dictionary base forms</td>
</tr>
<tr>
<td>LINGANALYSIS_FULL</td>
<td>This configuration provides the following language processing capabilities for linguistic analysis of unstructured data:</td>
</tr>
<tr>
<td></td>
<td>● Segmentation, also known as tokenization - the separation of input text into its elements</td>
</tr>
<tr>
<td></td>
<td>● Stemming - the identification of word stems or dictionary base forms</td>
</tr>
<tr>
<td></td>
<td>● Tagging - the labeling of words' parts of speech</td>
</tr>
<tr>
<td>EXTRACTION_CORE</td>
<td>This configuration extracts entities of interest from unstructured text, such as people, organizations, or places mentioned.</td>
</tr>
<tr>
<td></td>
<td>In most use cases, this option is sufficient.</td>
</tr>
<tr>
<td>Name of Text Analysis Configuration</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>EXTRACTION_CORE_ENTERPRISE</td>
<td>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.</td>
</tr>
<tr>
<td>EXTRACTION_CORE_PUBLIC_SECTOR</td>
<td>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.</td>
</tr>
<tr>
<td>EXTRACTION_CORE_VOICEOFCUSTOMER</td>
<td>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.</td>
</tr>
<tr>
<td>GRAMMATICAL_ROLE_ANALYSIS</td>
<td>This configuration provides the capability to identify functional (grammatical) relationships between elements in an input sentence (e.g., Subject, DirectObject).</td>
</tr>
</tbody>
</table>

**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>'**

**i 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 7].

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')
```

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.

Table 2: Structure of TA table

<table>
<thead>
<tr>
<th>Column ID</th>
<th>Key</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;n key columns from source table&gt;</td>
<td>Yes</td>
<td>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 $TA table includes all keys from the source table.</td>
<td>Same as in source table</td>
</tr>
<tr>
<td>$TA_RULE</td>
<td>Yes</td>
<td>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.</td>
<td>NVARCHAR(200)</td>
</tr>
<tr>
<td>$TA_COUNTER</td>
<td>Yes</td>
<td>The token counter counts all tokens across the document. The order is only unique for a given processing type (hence the $TA_RULE as the key).</td>
<td>BIGINT</td>
</tr>
<tr>
<td>$TA_TOKEN</td>
<td>-</td>
<td>Term, entity, or metadata - depending on the processing type.</td>
<td>NVARCHAR(250)</td>
</tr>
<tr>
<td>$TALANGUAGE</td>
<td>-</td>
<td>The language of the document is usually stated in the source table. In rare cases where no language is specified, the language code is stored here. Since there is no support for multi-language documents, the language code is identical for all result records of a document.</td>
<td>NVARCHAR(2)</td>
</tr>
<tr>
<td>$TA_TYPE</td>
<td>-</td>
<td>The token type contains the linguistic or semantic type of the token, for instance &quot;noun&quot; (if configuration = LINGANALYSIS_<em>), &quot;company&quot; (if configuration = EXTRACTION_</em>), &quot;Subject&quot; (if configuration = GRAMMATICAL_ROLE_ANALYSIS), or &quot;Author&quot; (if OutputMetadata is enabled). The note listed below this table contains a table that includes all SAP HANA $TA_TYPE values for linguistic analysis.</td>
<td>NVARCHAR(100)</td>
</tr>
<tr>
<td>Column ID</td>
<td>Key</td>
<td>Description</td>
<td>Data Type</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------</td>
</tr>
<tr>
<td>TA_TYPE_EXPANDED</td>
<td>-</td>
<td>By default, the column is empty. If the configuration used to create the full text index contains the <code>OutputExpandedTypes</code> option set to true, the column will contain the expanded form of the linguistic type information, for example V-Pres-Sg-be.</td>
<td>NVARCHAR(200)</td>
</tr>
</tbody>
</table>
| 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.                                                                                                                                          | 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 |
### Column ID

<table>
<thead>
<tr>
<th>Column ID</th>
<th>Key</th>
<th>Description</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA_OFFSET</td>
<td>-</td>
<td>Stores the offset in characters relative to the beginning of the document. This column will be NULL for metadata.</td>
<td>BIGINT</td>
</tr>
<tr>
<td>TA_PARENT</td>
<td></td>
<td>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 linguistic relationship between two tokens. For example, it is used by the EXTRATION_CORE_VOICEOFCUSTOMER rules to relate topics to their enclosing sentiments.</td>
<td>BIGINT</td>
</tr>
</tbody>
</table>

**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.

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.

More detailed type values can be stored in the TA_TYPE_EXPANDED column if the OutputExpandedTypes option is set in the text analysis configuration used to create the full text index.

This applies only to output from the LINGANALYSIS_* configurations.

### Table 3:

<table>
<thead>
<tr>
<th>HANA TA_TYPE Values (from $TA table)</th>
<th>Text Analysis Parts-of-Speech (from Language Reference)</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbreviation</td>
<td>Abbr</td>
</tr>
<tr>
<td>adjective</td>
<td>Adj, Adj-*</td>
</tr>
<tr>
<td>HANA TA_TYPE Values (from $TA table)</td>
<td>Text Analysis Parts-of-Speech (from Language Reference)</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>adverb</td>
<td>Adv, Adv-*</td>
</tr>
<tr>
<td>auxiliary verb</td>
<td>Aux, Modal</td>
</tr>
<tr>
<td>conjunction</td>
<td>Conj-<em>, Conj/</em></td>
</tr>
<tr>
<td>determiner</td>
<td>Det, Det-<em>, Det/</em>, Art-*</td>
</tr>
<tr>
<td>interjection</td>
<td>Interj</td>
</tr>
<tr>
<td>noun</td>
<td>Nn, Nn-*</td>
</tr>
<tr>
<td>number</td>
<td>Num</td>
</tr>
<tr>
<td>particle</td>
<td>Part-*</td>
</tr>
<tr>
<td>preposition</td>
<td>Prep, Prep-*</td>
</tr>
<tr>
<td>pronoun</td>
<td>Pron, Pron-*</td>
</tr>
<tr>
<td>proper name</td>
<td>Prop, Symb</td>
</tr>
<tr>
<td>punctuation</td>
<td>Punct, Punct-*</td>
</tr>
<tr>
<td>verb</td>
<td>V-* , V/*</td>
</tr>
<tr>
<td>unknown</td>
<td>anything not listed above</td>
</tr>
</tbody>
</table>
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 23]
Managing Custom Text Analysis Configurations with Stored Procedures [page 44]

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
<?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.FC</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>
    </property>
  </configuration>
  <configuration name="CommonSettings" />
  <configuration name="SAP.TextAnalysis.DocumentAnalysis.FormatConversion.FormatConversionAnalyzer.FC" 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>
  </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.


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.


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.


### 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.
i 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.
The option `OutputExpandedTypes` indicates whether the expanded type information is stored in the `TA_TYPE_EXPANDED` column of the `STA` table. If the expanded types are required, the property value should be specified with a Boolean `true` value: `<boolean-value>true</boolean-value>`.

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, for example `german-mydictionary` 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, Bohemian, 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.

### Related Information

http://help.sap.com/saphelp_hanaplatform/helpdata/en/20/31dfe5e9754d0fb09b5ca24fd0329f/frameset.htm

#### 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:

<table>
<thead>
<tr>
<th>Extension of Rule File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.hdbtextrule</td>
<td>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 <code>.hdbtextinclude</code> and <code>.hdbtextlexicon</code>.</td>
</tr>
<tr>
<td>.hdbtextinclude</td>
<td>Defines rule definitions to be used in one or more top-level rule sets using the <code>#include</code> directive. These are not compiled during deployment. Instead they are compiled during the deployment of the including top-level rule set.</td>
</tr>
</tbody>
</table>
Extension of Rule File

<table>
<thead>
<tr>
<th>Extension of Rule File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.hdbtextlexicon</td>
<td>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.</td>
</tr>
</tbody>
</table>

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:

```
#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.

Related Information

http://help.sap.com/saphelp_hanaplatform/helpdata/en/20/31de5e9754d0fb09b5ca24fd0329f/frameset.htm

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

```xml
<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.
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 24]
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 11]

3.5 Managing Custom Text Analysis Configurations with XS Advanced

Custom text analysis configuration files can be added to an XS Advanced multi-target application (MTA) project just like any other database artifacts. You can edit and test text analysis artifacts interactively using tools provided by the SAP Web IDE. Once the artifacts are ready, you can deploy them to production SAP HANA systems using the standard tools provided with the SAP HANA Deployment Infrastructure.

The following sections assume that you are familiar with the basic concepts and operations of XS Advanced and the SAP Web IDE.

Note
For more information about the concepts and operation of XS Advanced and the SAP Web IDE, see the SAP HANA Developer Guide for SAP HANA XS Advanced Model.

Artifact References
XS Advanced Application projects usually use namespaces to organize artifacts. Namespaces are defined when you create the HDB module in the Web IDE, and are stored in the runtime namespace configuration.
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 double colons, "::", followed by the artifact name. Refer to HDI Run-Time Name spaces in the SAP HANA Developer Guide for XS Advanced Model for more information about using namespaces in database artifacts.

Sample Code

```xml
<property name="Dictionaries" type="string-list">
  <string-list-value>myproject.db::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 or removing the namespace file. If your project omits namespaces, then artifact references should omit the namespace and double colon, and consist of just the artifact name.

Sample Code

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

Related Information

SAP HANA Developer Guide for SAP HANA XS Advanced Model

3.5.1 Getting Started

Start working with SAP Web IDE for SAP HANA by logging on and preparing a multi-target application (MTA) project where you will develop your text analysis dictionaries and rules.

Prerequisites

- Access to a running SAP HANA system where XS Advanced (XSA) and the SAP Web IDE are installed and configured.
- The URL and logon credentials for accessing the SAP Web IDE.
Context

Text analysis dictionaries and rules represent HANA database artifacts, and are therefore developed as part of your application’s overall HANA data-persistence model. In the XS Advanced environment, this means that text analysis artifacts are developed and deployed as part of a database module within a multi-target application (MTA) project.

Text analysis dictionary and rule files are design-time definitions that are used to generate the corresponding run-time database objects, when the dictionary or rule file is deployed to the SAP HANA database. SAP Web IDE uses the HANA Deployment Infrastructure (HDI) tools to deploy database artifacts to an HDI run-time container. Design-time database objects are typically located in the `db/src` folder of the application design-time hierarchy.

Procedure

1. Start the SAP Web IDE for SAP HANA.
2. Create a new project to use for developing your text analysis artifacts.

   This procedure describes how to create a new project from scratch. There are a number of other ways to create a project, for example, by importing it or cloning it. Refer to the SAP HANA Developer Guide for SAP HANA XS Advanced Model for more options.

   1. Choose `File > New > Project from Template`.
   2. Choose the project template type. Select `Multi-Target Application Project` and choose `Next`.
   3. Specify a name for the new project and choose `Next`.
   4. Specify the application ID, version number, and target space for the new project and choose `Next`.
   5. Choose `Finish` to create the new project.

   A project folder is created. The project folder initially contains one essential file, the application’s deployment descriptor `mta.yaml`, which defines the prerequisites and dependencies for the deployment of a multi-target application. At this point in time, the file contains only the name and version of the project and an optional short description.

3. Add a database module to contain your new text analysis artifacts.

   1. Right-click the root folder of your new project and, in the context menu, choose `New > HDB Module`.
   2. Provide a name (e.g., `db`) for the new database module and choose `Next`.
   3. Customize the details of the new database module, if required, and choose `Next`. SAP Web IDE automatically assigns the namespace for the database artifacts to be the project name, followed by a dot (.), followed by the HDB module name. If you specify the schema name for your new module, the HDI container for your run-time objects will have the corresponding database schema having the following name format: `<WebIDE workspace ID>-<chosen schema name>`. Otherwise, the schema name will be `<WebIDE workspace ID>-<automatically generated name>`.
   4. Choose `Finish` to create the new module.

   A new database module is created with the specified name. The database module contains a single folder `src/` where you should put all of your text analysis (and other database) artifacts. You may create additional sub-folders below the `src/` folder to further organize your text analysis dictionaries and rules.
Results

The `src/` folder initially contains two files that are normally hidden by default. To view them, select `View Show Hidden Files` from the SAP Web IDE menu. One file is the runtime namespace configuration (.hdinamespace) file, which records the namespace that will be used for the database artifacts generated by this module. The other file is the container configuration (.hdiconfig), which specifies the HDI plug-ins that will be used to deploy the artifacts contained in this module. You do not normally need to edit these files.

Next Steps

You are ready to begin creating text analysis dictionary and rule files.

Related Information

SAP HANA Developer Guide for SAP HANA XS Advanced Model

3.5.2 Creating Custom Text Analysis Dictionaries with XS Advanced

Text analysis dictionaries can be used to extract custom entities from your input data, whether the entities are composed of common words that have special meaning in your application's domain, or the entities are alphanumeric sequences such as specialized vocabulary or part numbers. Dictionaries are represented using the XML markup language.

Prerequisites

- You have created a multi-target application (MTA) project.
- You have created an HDB (database) module within the project.
- You are familiar with the features of text analysis dictionaries.

Note

For detailed information on the structure of dictionaries and how they affect the results of text analysis, refer to the SAP HANA Text Analysis Extraction Customization Guide.
Context

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

Procedure

1. Locate the folder in the project hierarchy that will contain the text analysis dictionary. This should either be the HDB module's src/ folder, or some sub-folder of that folder.

2. Create a new dictionary.
   1. Right-click on the folder, select New in the context menu, and select Text Analysis Dictionary.
   2. In the New Text Analysis Dictionary dialog box, enter a name for the dictionary and choose Create. A new dictionary file will be created within the folder with the file extension .hdbtextdict appended to its name, and a new tab containing a few lines of XML will open in the editor panel. The new dictionary will already contain the top-level XML syntax required by all text analysis dictionaries. Alternatively, to import an existing dictionary from a local file:
      1. Right-click on the folder, select Import in the context menu, and select From File System.
      2. In the Import dialog box, browse to the desired file and click the OK button. Be sure that the local file has the correct (.hdbtextdict) file extension. A new dictionary will be created beneath the folder, and a new tab containing the dictionary’s XML will open in the editor panel.

3. Add an entity category.

   Entities are organized into groups, called categories, based on their semantic types. Each entity category is defined by an <entity_category> XML element, which must be a child of the <dictionary> XML element.

   1. Position the cursor in the editor panel on the blank line that follows the opening <dictionary> tag. Or, if you are editing a dictionary that already contains some categories, position the cursor in the editor panel at the start of the line preceding an opening <entity_category> tag or following a closing </entity_category> tag.

   2. Insert a new <entity_category> element by doing one of the following:
      1. Click Edit Insert Snippet Dictionary Category on the menu bar.
      2. Right-click and choose Insert Snippet Dictionary Category from the context menu.
      3. Press the keyboard shortcut for Insert Snippet Dictionary Category. This inserts the XML markup for a dictionary category and moves the cursor to the correct location for typing the name of the category.

### Note

The category’s XML markup is always inserted at the current location of the cursor. Therefore, before you insert a category you must make sure the cursor is between <dictionary>…</dictionary> tags and outside of any other <entity_category>…</entity_category> tags. Otherwise, the dictionary’s resulting XML will be incorrect.
4. Add a new entity (instance).

Individual entities (i.e., instances) are defined using the `<entity_name>` XML element. Each entity must be contained within the `<entity_category>` XML element to which it belongs. That is, it must appear between the `<entity_category>` and `</entity_category>` XML tags for its semantic type.

1. Position the cursor in the editor panel on a line preceding an `<entity_name>` or `</entity_category>` tag, or on a line following an `</entity_name>` or `<entity_category>` tag.
2. Insert a new `<entity_name>` element by doing one of the following:
   1. Click `Edit > Insert Snippet > Dictionary Entity` on the menu bar.
   2. Right-click and choose `Insert Snippet > Dictionary Entity` from the context menu.

   This inserts the XML markup for an entity and moves the cursor to the correct location for typing the entity’s name.

   **Note**

   The entity’s XML markup is always inserted at the current location of the cursor. Therefore, before you insert an entity you must make sure the cursor is the correct position (see step a above). Otherwise, the dictionary’s resulting XML will be incorrect.

   **Note**

   You can also manually type, or copy & paste, the XML for an entity into the editor panel.

5. Optional: Add a new variant name.

Variant names are alternative names for entity instances, usually shorter, less standard or complete than a standard entity name, and may include abbreviations, different spellings, nicknames, and so on. Variants are defined using the `<variant>` XML element. All of the variants for a single entity must appear between the associated entity’s `<entity_name>` and `</entity_name>` XML elements.

1. Position the cursor in the editor panel on a line preceding a `<variant>` or `</entity_name>` tag, or on a line following a `</variant>` or `<entity_name>` tag.
2. Insert a new `<variant>` element by doing one of the following:
   1. Click `Edit > Insert Snippet > Dictionary Variant` on the menu bar.
   2. Right-click and choose `Insert Snippet > Dictionary Variant` from the context menu.
   3. Press the keyboard shortcut for `Insert Snippet > Dictionary Variant`.

   This inserts the XML markup for a variant and moves the cursor to the correct location for typing the variant’s name.

   **Note**

   The variant’s XML markup is always inserted at the current location of the cursor. Therefore, before you insert a variant you must make sure the cursor is the correct position (see step a above).

   Otherwise, the dictionary’s resulting XML will be incorrect.
You can also manually type, or copy & paste, the XML for a variant into the editor panel.

6. Repeat steps 3 through 5 as needed to enter all of your entities into the dictionary.
7. Save the changes to your dictionary file.
8. Build the dictionary by doing one of the following:
   1. Select the HDB module in the workspace view and choose Build from the menu bar.
   2. Select the HDB module in the workspace view and choose Build from the context menu.

Building the dictionary deploys the dictionary to your HANA system and checks for any errors. The progress of the build is reported in the console view.

If you have modified any other files in the HDB module since the last build, those modified files will be re-built as well.

If an error is detected in the dictionary syntax, a dialog box will appear and the errors will be reported in the console view. Alternatively, you can view the error in the Problem view. Choose View to display it. You will need to re-build the dictionary after locating and repairing the syntax error.

If the build is successful, the dictionary is now ready to be used.

Next Steps

Test your dictionary interactively. Optionally write custom text analysis rules that depend on the custom dictionary. Create a text analysis configuration that references your dictionary, and add it to your project.

Related Information

Testing your Dictionaries and Rule Sets [page 38]

3.5.3 Creating Custom Text Analysis Rule Sets with XS Advanced

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 a multi-target application (MTA) project.
- You have created an HDB (database) module within the project.
- You are familiar with writing custom text analysis extraction rules.

Note

For detailed information on the syntax of extraction rules and how they affect the results of text analysis, refer to the *SAP HANA Text Analysis Extraction Customization Guide*.

Context

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

Procedure

1. Locate the folder in the project hierarchy that will contain the text analysis rule set. This should either be the HDB module’s `src` folder, or some sub-folder of that folder.
2. Create a new rule set file.
   1. Right-click on the folder, select New in the context menu, and select Text Analysis Rule Set.
   2. In the New Text Analysis Rule Set dialog box, enter a name for the rule set and click Create.

A new rule set file will be created within the folder with the file extension `.hdbtextrule` appended to its name, and the (initially empty) file will open in a new tab in the editor panel.

Alternatively, to import an existing rule set from a local file:

1. Right-click on the folder, select Import in the context menu, and select From File System.
2. In the Import dialog box, browse to the desired file and click OK. Be sure that the local file has the correct `.hdbtextrule` file extension.

A new rule set will be created beneath the folder, and a new tab containing the rule set file’s content will open in the editor panel.

3. Enter your custom extraction rules into the rule set file using the SAP Web IDE text editor. Refer to the *SAP HANA Text Analysis Extraction Customization Guide* for detailed information on writing and editing rules.
4. Save the changes to your rule set file.
5. Optional: Create a file containing common rule definition.

In advanced applications, large rule sets may need to be split into multiple files, and common rule definitions may need to be shared across those rule sets. For this scenario, you can create text analysis
**include** and **lexicon** rule files in your HDB module, which allow you to define common rule definitions and word lists, respectively.

1. Right-click on the folder to contain your shared rule files, select **New** in the context menu, and select **File**.
2. In the **New File** dialog box, enter a name for the shared rules file, including the file extension. Include files should use the extension `.hdbtextinclude` and lexicon files should use `.hdbtextlexicon`. Click **Create**.
3. A new file will be created within the folder, and the (initially empty) file will open in a new tab in the editor panel.
4. Enter your custom extraction rules into the rule set file using the SAP Web IDE text editor. Refer to the **SAP HANA Text Analysis Extraction Customization Guide** for detailed information about using `#include` and `#lexicon` directives in rule files.
5. Save the changes to your include or lexicon file.
6. Repeat steps a through e for each common file needed by your rule sets.

6. **Build** the rule set by doing one of the following:

   1. Select the HDB module in the workspace view and choose **Build** from the menu bar.
   2. Select the HDB module in the workspace view and choose **Build** from the context menu.

   Building the rules set deploys the rules set to your HANA system and checks for any errors. The progress of the build is reported in the console view.

   **Note**
   If you have modified any other files in the HDB module since the last build, those modified files will be re-built as well.

   If an error is detected in the rule syntax, a dialog box will appear and the errors will be reported in the console view. Alternatively, you can view the error in the Problem view. Choose **View > Problem** to display it. You will need to re-build the rule set after locating and repairing the syntax error.

   If the build is successful, the rule set is now ready to be used.

**Next Steps**

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

**Related Information**

*Testing your Dictionaries and Rule Sets [page 38]*
3.5.4 Testing Dictionaries and Rules with XS Advanced

You can run and test your text analysis dictionaries and rule sets using the tools provided by SAP Web IDE.

Prerequisites

- You have created a multi-target application (MTA) project.
- You have created an HDB (database) module within the project.
- You have created and successfully built one or more custom text analysis dictionaries or rule sets.

3.5.4.1 Preparing a Text Analysis Run Configuration

You can configure how to run and test text analysis dictionaries and rules in your project.

Context

Before you can test your text analysis artifacts, you must first create a text analysis run configuration.

Procedure

1. Right-click on the HDB (database) module containing your text analysis dictionaries and rule sets, select Run in the context menu, and select Run Configurations.

   Alternatively, select the HDB (database) module containing your text analysis dictionaries and rule sets, then choose Run > Run Configurations from the menu bar.

2. In the Run Configurations dialog, click + and choose the type of run configuration, Text Analysis Batch Test.

   A new configuration with a default name appears under the selected category in the left panel.

3. Edit the name for the new run configuration if desired.

4. Perform the following steps on the Input Data tab:

   1. Choose Enter Plain Text for the Data Source. Refer to Using Test Data from HANA Tables for information on using the Select Database Table option to test with data stored in SAP HANA database tables.
   2. Enter input text against which the test should be run.
   3. Select the input language for the test, or choose Auto-detect to let text analysis determine the language automatically.

5. Perform the following steps on the Dictionaries and Rule Sets tab:
1. Choose a **Base Configuration**. This is a text analysis configuration that will provide the default text analysis options for the test. The text analysis configurations available for selection are the standard **EXTRACTION** configurations provided by SAP.

2. Optional: Select **Dictionaries** by clicking the search box and choosing the dictionaries in your HDB module that should be included in the test.

3. Optional: Select **Rule Sets** by clicking the search box and choosing the rule sets in your HDB module that should be included in the test.

6. To save the configuration and run your text analysis artifacts with this configuration, choose **Save and Run**. To save the configuration without running, choose **OK**. To discard all your changes, choose **Cancel**.

### Note

These settings are persistent and are used every time that you run the selected run configuration, until you edit or delete the settings.

### Next Steps

If you chose **Save and Run**, a new **Text analysis batch test results** window will open in the browser showing the results from the test.

### Related Information

- Testing your Dictionaries and Rule Sets [page 38]
- Exploring the Text Analysis Test Results [page 39]

### 3.5.4.2 Testing your Dictionaries and Rule Sets

You can run and test text analysis dictionaries and rules in your project.

### Prerequisites

You have created a text analysis run configuration for the HDB (database) module containing your text analysis dictionaries and rule sets.
Procedure

1. Right-click on the HDB (database) module containing your text analysis dictionaries and rule sets, select Run in the context menu, and select your text analysis run configuration.

   Note

   You can also re-run the last test by simply selecting the Run button on the global toolbar.

   A new Text analysis batch test results window will open in the browser showing the results from the test.

2. Review the test results. (Refer to section 3.5.4.3 for information about the features of the text analysis test results window.)

3. Optional: Modify your dictionaries and rule sets, save and rebuild them, and return to step 1 to re-run the test.

4. Optional: Edit the text analysis run configuration to change the input text to test different aspects of your dictionaries and rule sets.

5. Terminate the test by closing the text analysis test results window.

Next Steps

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

Related Information

Exploring the Text Analysis Test Results [page 39]

3.5.4.3 Exploring the Text Analysis Test Results

You can explore the results of your text analysis tests using the Text analysis batch test results window.

Overview

The Text analysis batch test results window shows the results of your text analysis test.

The left sidebar lists each of the test inputs. If you specified a single plain text input, there will only be one entry in the left sidebar. If you are using test data from a HANA table, there will be one entry in the left sidebar for each row in the HANA table. Each entry is shown with the name and value of the Document ID Column you chose when you created the text analysis run configuration.

Click an entry in the left sidebar to see the text analysis results for that test input in the right pane. The test input will appear with highlighting showing where text analysis entities (i.e., dictionary and rule matches)
occurred. You can also view the text analysis entities that were extracted in a tabular form. Choose the 
*Highlight* or *Tabular* tabs to switch back and forth between views.

**Highlight View**

The *Highlight* view displays the test document with the extracted entities highlighted and in bold. Click on the 
*Highlight* tab to switch to this view. You can hover the mouse pointer over a highlighted entity to see its entity type.

**Tabular View**

The *Tabular* view shows the extracted entities in order of appearance in the text, one entity per row. Right-click 
on a column heading to select sorting options or choose which columns are displayed. The following columns are supported:

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Counter</td>
<td>The sequence number of the entity within the set of all entities in the test results.</td>
</tr>
<tr>
<td>Token</td>
<td>The extracted entity as it appeared in the document.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of the extracted entity. This may be a built-in type (e.g., NOUN_GROUP), a type defined in a particular text analysis configuration (e.g., StrongPositiveSentiment in the EXTRACTION_CORE_VOICEOFCUSTOMER configuration), or one defined in a custom dictionary or rule set.</td>
</tr>
<tr>
<td>Normalized</td>
<td>The normalized form of the entity (as specified in a text analysis dictionary using the ‘standard_form’ attribute).</td>
</tr>
<tr>
<td>Paragraph</td>
<td>The relative paragraph number containing the token.</td>
</tr>
<tr>
<td>Sentence</td>
<td>The relative sentence number containing the token.</td>
</tr>
<tr>
<td>Offset</td>
<td>The character offset (0-based) of the entity’s first character within the test input.</td>
</tr>
<tr>
<td>Parent</td>
<td>If the entity is defined as part of a larger entity, this column will contain the Counter value of the larger (“parent”) entity.</td>
</tr>
</tbody>
</table>
Filtering the Results

Click the arrow (>) on an entry in the left sidebar to open the entity filter. When the entity filter is expanded, each type of entity found in that test input is displayed on the left with a checkbox. Uncheck an entity type on the left and click the Update button to remove that entity type from the Highlight and Tabular views. Click the topmost All checkbox to quickly check or uncheck all boxes.

Related Information

Using Test Data from HANA Tables [page 41]

3.5.4.4 Using Test Data from HANA Tables

You can use data stored in HANA tables to test your custom text analysis dictionaries and rule sets.

Prerequisites

You have data stored in a database table against which you would like to run your test. Since the test is run from the HDI container where dictionaries and rule sets are deployed, there are two options:

- The database table needs to be deployed to the same HDI container. If you are developing text analysis artifacts at the same time as your application data model, then the latter can be used as target database tables in your tests. Refer to Configuring the HDI Deployer in the SAP HANA Developer Guide for SAP HANA XS Advanced Model for more information.
- The database table needs to have a synonym deployed to the same HDI container, in cases where you would like to run tests against existing HANA database data. For more information, refer to the Creating Synonyms section of the SAP HANA Developer Guide for SAP HANA XS Advanced Model.

Context

Before you can test your text analysis artifacts using data from SAP HANA database tables, you must first create a text analysis run configuration.

Procedure

1. Right-click on the HDB (database) module containing your text analysis dictionaries and rule sets, select Run in the context menu, and select Run Configurations.
2. To edit an existing text analysis run configuration, click its name in the panel. See Related Information on how to create a new text analysis configuration.

3. Perform the following steps on the Input Data tab:
   1. For the Data Source, choose Select Database Table.
   2. Select the Table Name containing the data with which you wish to test.
   3. For the Document ID Column, choose a column that contains a uniquely identifying value for each document. An identifying value could be a document ID number, a filename, URL, the document’s title, etc.
   4. Set the value of the Document Content Column control to the column containing the text data that you want to use to test your dictionaries and rules.
   5. Select the input language for the test. You can either choose the language, choose a column in the database table that will contain the language, or choose Auto-detect to let text analysis determine the language automatically.

4. Continue editing the remainder of the text analysis run configuration (see Related Information).

5. To save the configuration and run your text analysis artifacts with this configuration, choose Save and Run. To save the configuration without running, choose OK. To discard all your changes, choose Cancel.

These settings are persistent and are used every time that you run the selected run configuration, until you edit or delete the settings.

Related Information

Preparation a Text Analysis Run Configuration [page 37]

3.5.5 Creating a Text Analysis Configuration with XS Advanced

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

Prerequisites

- You have created a multi-target application (MTA) project.
- You have created an HDB (database) module within the project.
- You are familiar with writing text analysis configuration files.

Context

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

1. Locate the folder in the project hierarchy that will contain the text analysis configuration. This should either be the HDB module’s src/ folder, or some sub-folder of that folder.

2. Right-click on the folder, select New in the context menu, and select File.

3. In the New File dialog box, enter a name for the text analysis configuration file, including the .hdbtextconfig file extension. Click the Create button.

4. Enter your custom configuration settings into the text analysis configuration file using the SAP Web IDE text editor. See Related Information for detailed information about the text analysis configuration syntax.

5. Build the configuration by doing one of the following:
   1. Select the configuration file in the workspace view and choose Build ➤ Build Selected Files from the menu bar.
   2. Right-click on the configuration file in the workspace view and choose Build Selected Files from the context menu.
   3. Select the HDB module in the workspace view and choose ➤ Build ➤ Build from the menu bar.
   4. Select the HDB module in the workspace view and choose Build from the context menu.

Building the configuration deploys the configuration to your SAP HANA system and checks for any errors. The progress of the build is reported in the console view.

**Note**

If you have modified any other files in the HDB module since the last build, those modified files will be re-built as well.

If an error is detected in the configuration syntax, a dialog box will appear and the errors will be reported in the console view. Alternatively, you can view the error in the Problem view. Choose ➤ View ➤ Problem to display it. You will need to re-build the configuration after locating and repairing the syntax error.

If the build is successful, the configuration is now ready to be used.

Next Steps

Create a fulltext index definition that references your configuration, and add it to your project. Package and deploy your text analysis customizations.

Related Information

Text Analysis Configuration File Syntax [page 11]
3.5.6 Packaging and Deploying Text Analysis Customizations

At the last stage of text analysis customization, you need to package the multi-target application (MTA) containing your text analysis customizations and deploy it to the SAP HANA XS Advanced Model production system.

Prerequisites

All the HDB database artifacts (including text analysis customizations) in your application are implemented and tested.

Procedure

1. Build the whole application by choosing Build from the context menu of the project root. The application is packaged into an MTA archive named <ID>_<version>.mtar, which appears in the project under mta_archives<ID>, where ID and version are the properties you specified when you created the application. These properties are actually defined in the MTA descriptor, mta.yaml, which is created automatically at the root of your project folder.

2. Export the archive to the file system by choosing Export from the archive’s context menu. The archive file appears in your default download folder. This archive contains all the artifacts required for the deployment of your application, including the auto-generated deployment version of the MTA descriptor (mtad.yaml).

Next Steps

The exported MTA archive file can now be used to deploy the text analysis customizations to a production SAP HANA XS Advanced system.

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**

```xml
<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**

```xml
<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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHEMA_NAME</td>
<td>NVARCHAR (256)</td>
<td>Specifies the name of the schema that will contain the resource.</td>
</tr>
<tr>
<td>NAME</td>
<td>NVARCHAR (256)</td>
<td>Specifies the name of the resource.</td>
</tr>
<tr>
<td>TYPE</td>
<td>VARCHAR (16)</td>
<td>Specifies the resource type.</td>
</tr>
<tr>
<td>DATA</td>
<td>BLOB</td>
<td>Specifies the resource content.</td>
</tr>
</tbody>
</table>
Table 7: Values of the TYPE Parameter

<table>
<thead>
<tr>
<th>Type</th>
<th>Description of Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>hdbtextconfig</td>
<td>A text analysis configuration</td>
</tr>
<tr>
<td>hdbtextdict</td>
<td>A text analysis dictionary</td>
</tr>
<tr>
<td>hdbtextrule</td>
<td>A text analysis rule set</td>
</tr>
<tr>
<td>hdbtextinclude</td>
<td>A text analysis rule set include</td>
</tr>
<tr>
<td>hdbtextlexicon</td>
<td>A text analysis rule set word list</td>
</tr>
<tr>
<td>textminingconfig</td>
<td>A text mining configuration</td>
</tr>
</tbody>
</table>

**TEXT_CONFIGURATION_DROP**

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

Table 8: Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHEMA_NAME</td>
<td>NVARCHAR (256)</td>
<td>Specifies the name of the schema that contains the resource.</td>
</tr>
<tr>
<td>NAME</td>
<td>NVARCHAR (256)</td>
<td>Specifies the name of the resource.</td>
</tr>
<tr>
<td>TYPE</td>
<td>VARCHAR (16)</td>
<td>Specifies the resource type.</td>
</tr>
</tbody>
</table>

**TEXT_CONFIGURATION_CLEAR**

Notify the system that custom resources have changed.

Table 9: Parameters

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHEMA_NAME</td>
<td>NVARCHAR (256)</td>
<td>Specifies the name of the schema that contains the resource.</td>
</tr>
<tr>
<td>NAME</td>
<td>NVARCHAR (256)</td>
<td>Optional: Specifies the name of the resource.</td>
</tr>
<tr>
<td>TYPE</td>
<td>VARCHAR (16)</td>
<td>Optional: Specifies the resource type.</td>
</tr>
</tbody>
</table>
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.
   ```sql
   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 45]
- Text Analysis Configuration File Syntax [page 11]
- Notifying the System of Changes with Stored Procedures [page 52]
- Obtaining Predefined Text Analysis Configurations [page 53]

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. 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.

   ```sql
   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 47]
- Managing Custom Text Analysis Configurations with Stored Procedures [page 44]
- Notifying the System of Changes with Stored Procedures [page 52]
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 45]
- Creating a Text Analysis Configuration with Stored Procedures [page 47]
- Creating Custom Text Analysis Rule Sets with Stored Procedures [page 48]
- Notifying the System of Changes with Stored Procedures [page 52]

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'.
   ```sql
   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.

i 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 45]

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 45]

### 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

Text Analysis [page 4]
4 Using the Text Analysis SQL API

The stored procedure `TA_ANALYZE` returns the output from performing text analysis on a single input value. The procedure accepts all text analysis inputs and options, and returns all text analysis outputs that can be obtained using either FULLTEXT INDEXes or the XS JavaScript API in SAP HANA. This includes all TA annotations that can be retrieved via the $TA table (LA, EA, GRA, and metadata), as well as document-level data that can be retrieved using the `LANGUAGE`, `MIMETYPE`, and `PLAINTEXT` SQL functions.

Table 10:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOCUMENT_BINARY</td>
<td>The input document can either be specified as a string or a binary/text document. If the document is binary (i.e. Word, PDF, etc.) use this parameter and set DOCUMENT_TEXT to NULL.</td>
</tr>
<tr>
<td>DOCUMENT_TEXT</td>
<td>If the document is text (i.e. XML, HTML, etc.) or a string use this parameter and set DOCUMENT_BINARY to NULL.</td>
</tr>
<tr>
<td>LANGUAGE_CODE</td>
<td>Specifies the language of the document. Only one of <code>LANGUAGE</code> and <code>LANGUAGE_DETECTION</code> parameters should be non-empty. If language is ' ' then <code>LANGUAGE_DETECTION</code> is used to detect the language and set this parameter. If <code>LANGUAGE_DETECTION</code> = NULL and <code>LANGUAGE</code> = ' ' then it defaults to English.</td>
</tr>
</tbody>
</table>

**Note**

Specify the "neutral" language (two-letter code "UD") to perform only basic analysis on the document, such as tokenization, case normalization, and umlaut conversion. You may want to do this when a document is in a language not listed in the SAP HANA Text Analysis Language Reference Guide (for example, Irish Gaelic). No entities will be extracted unless you have custom rules or dictionaries written in the neutral language.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIME_TYPE</td>
<td>Specifies the default document mime type. If MIME_TYPE is '', then automatic format detection is used and this parameter is set to the detected mime type.</td>
</tr>
<tr>
<td></td>
<td><strong>i</strong> Note                                                                                                                                                                                                                           <strong>The only mime types that can be used are text/plain, text/html, and text/xml. Binary types may be specified (i.e., no error is reported), but automatic format detection is always used for binary formats.</strong></td>
</tr>
<tr>
<td>TOKEN_SEPARATORS</td>
<td>The set of ASCII characters used for token separation. If this is NULL then the default list is used: &lt;/&gt;;,.:-_()[]&lt;&gt;!?@+{}=&quot;&amp;&gt;</td>
</tr>
<tr>
<td>LANGUAGE_DETECTION</td>
<td>Specifies the set of languages to be considered during language detection as a string literal list of comma-separated ISO 639-1 two character codes. If NULL is given, then the LANGUAGE_CODE parameter is assumed to specify the language.</td>
</tr>
<tr>
<td></td>
<td><strong>i</strong> Note                                                                                                                                                                                                                           <strong>In addition, you may specify the “neutral” language code (“UD”) as the first language in the LANGUAGE_DETECTION list. This causes “neutral” language processing to be used when the actual language cannot be determined automatically.</strong></td>
</tr>
<tr>
<td>CONFIGURATION_SCHEMA_NAME</td>
<td>The schema containing the configuration. Can be left blank for a standard configuration or a configuration that is in the SAP HANA repository.</td>
</tr>
<tr>
<td>CONFIGURATION</td>
<td>Specifies the path to a custom configuration file for text analysis. If no configuration is specified, the default LINGANALYSIS_FULL is used.</td>
</tr>
<tr>
<td>RETURN_PLAINTEXT</td>
<td>1: specifies to return the plain text extracted from the input document by the format conversion analyzer. If set to 1, the text will be returned in the plaintext output parameter, 0 otherwise.</td>
</tr>
<tr>
<td>TA_ANNOTATIONS</td>
<td>The output table containing text analysis annotations, similar to the $TA table.</td>
</tr>
<tr>
<td>PLAINTEXT</td>
<td>The plain text metadata extracted from the document.</td>
</tr>
</tbody>
</table>
4.1 Performing Linguistic Analysis Using TA_ANALYZE

The following example returns Linguistic Analyzer part of speech tokens for the sentence *I love that dirty water, Boston, you're my home!* with English detected as the language.

Execute the following SQL statement in a SQL console in SAP HANA Studio:

```
CALL TA_ANALYZE ( DOCUMENT_TEXT => 'I love that dirty water, Boston, you''re my home!', LANGUAGE_CODE=>?, MIME_TYPE =>?, LANGUAGE_DETECTION =>'DE, EN, KR', CONFIGURATION=>'LINGANALYSIS_FULL', RETURN_PLAINTEXT=>0, TA_ANNOTATIONS => ?, PLAINTEXT => ? );
```

4.2 Performing Entity Extraction Using TA_ANALYZE

The following example performs core extraction for the input HTML code and extracts the plain text inside HTML tags.

```
CALL TA_ANALYZE ( DOCUMENT_BINARY => '<!DOCTYPE html><html><body><h1>This is heading 1</h1><h2>TechABC Co. is one of the companies that has spent about $50 billion on acquisitions in the past decade.</h2></body></html>', DOCUMENT_TEXT=>'', LANGUAGE_CODE=>?, MIME_TYPE =>?, LANGUAGE_DETECTION =>'EN, DE, KR', CONFIGURATION=>'EXTRACTION_CORE', RETURN_PLAINTEXT=>1, TA_ANNOTATIONS => ?, PLAINTEXT => ? );
```
5 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

5.1 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.

```
// 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));
```

5.2 Text Analysis XS Advanced API

Text analysis is available in SAP HANA XS Advanced as the sap-textanalysis Node.js package. The package is part of the XS_JAVASCRIPT software component available on the SAP Service Marketplace.
Installation instructions for the XS_JAVASCRIPT bundle can be found in the XS Advanced Developer Guide.

sap-textanalysis is implemented as an interface to the TA_ANALYZE SQL stored procedure.

It provides the following single API function: analyze().

5.2.1 Text Analysis XS Advanced API Example Application

To use sap-textanalysis, first a HANA database connection must be established. Then the client database object can be passed to the analyze method along with the input parameters. The input parameters set the input variables to the TA_ANALYZE stored procedure.

```javascript
var ta = require('sap-textanalysis');
var client;
async.series(
    function connect(callback) {
        client = hdb.createClient(options);
        client.connect(callback);
    },
    function analyze(callback) {
        var values = {
            DOCUMENT_TEXT: '<!DOCTYPE html><html><body><h1>My First Heading</h1><p>My first paragraph.</p></body></html>',
            LANGUAGE_CODE: 'EN',
            CONFIGURATION: 'EXTRACTION_CORE',
            RETURN_PLAINTEXT: 0
        };
        ta.analyze(values, client, function done(err, parameters, rows) {
            if (err) { return console.error('error', err); }
            callback();
        });
    },
    function end(callback) {
        client.end(callback);
    },

Sample Code

```

5.3 Text Analysis XSJS Compatibility Layer

Text analysis is available in the XSJS compatibility layer that allows for SAP HANA extended application services, classic model (SAP HANA XS Classic) applications to run on Node.js in SAP HANA extended application services, advanced model.

For an installation tutorial see Use the XSJS Compatibility Layer in XS Advanced in SAP HANA Developer Guide for SAP HANA XS Advanced Model.
Related Information

SAP HANA Developer Guide for SAP HANA XS Advanced Model
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.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

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: http://help.sap.com/disclaimer).