



# Informatica<sup>TM</sup>

## Out-of-the-Box DQ rules for Product 360

Informatica MDM - Product 360

Version: 10.1

## Table of Contents

<b>1</b>	<b>Project Structure within Informatica Data Quality (Design Environment) .....</b>	<b>7</b>
1.1	Upgrading from one IDQ Version to another .....	8
<b>2</b>	<b>Importing Rules into Product 360 .....</b>	<b>10</b>
<b>3</b>	<b>Rule types and reference tables .....</b>	<b>10</b>
3.1	Generic rules.....	10
3.2	Structure Feature / Attribute rules.....	11
3.3	Product code validation rules .....	11
3.4	Classification rules .....	11
3.5	Lookup and standardization rules .....	11
3.6	Reference tables.....	11
<b>4</b>	<b>Generic Rules.....</b>	<b>12</b>
4.1	Check_DependentFields.....	12
4.1.1	General description.....	12
4.1.2	Input ports.....	12
4.1.3	Output ports.....	12
4.1.4	Meta data and reference tables.....	12
4.1.5	Example usage .....	13
4.1.5.1	Example output.....	13
4.2	Check_Equal.....	13
4.2.1	General description.....	13
4.2.2	Input ports.....	13
4.2.3	Output ports.....	13
4.2.4	Meta data and reference tables.....	14
4.2.5	Example usage .....	14
4.2.5.1	Example output.....	14
4.3	Check_isEmpty.....	14
4.3.1	General description.....	14
4.3.2	Input ports.....	14
4.3.3	Output ports.....	14
4.3.4	Meta data and reference tables.....	15
4.3.5	Example usage .....	15

4.3.5.1	Example output.....	15
4.4	<b>Check_isEmpty_SubentityLevel .....</b>	<b>15</b>
4.4.1	General description.....	15
4.4.2	Input ports.....	15
4.4.3	Output ports.....	16
4.4.4	Meta data and reference tables.....	16
4.4.5	Example usage .....	16
4.4.5.1	Example output.....	16
4.5	<b>Check_MaxLength .....</b>	<b>16</b>
4.5.1	General description.....	16
4.5.2	Input ports.....	17
4.5.3	Output ports.....	17
4.5.4	Meta data and reference tables.....	17
4.5.5	Example usage .....	17
4.5.5.1	Example output.....	17
4.6	<b>Check_MinLength .....</b>	<b>18</b>
4.6.1	General description.....	18
4.6.2	Input ports.....	18
4.6.3	Output ports.....	18
4.6.4	Meta data and reference tables.....	18
4.6.5	Example usage .....	18
4.6.5.1	Example output.....	19
5	<b>Feature / Attribute Rules.....</b>	<b>19</b>
5.1	<b>Check_DataTypes .....</b>	<b>19</b>
5.1.1	General description.....	19
5.1.2	Sub-Entity type for Input .....	19
5.1.3	Input ports.....	19
5.1.4	Output ports.....	20
5.1.5	Meta data and reference tables.....	20
5.1.6	Example usage .....	21
5.1.6.1	Example output.....	21
5.2	<b>Check_MandatoryValues .....</b>	<b>21</b>
5.2.1	General description.....	21
5.2.2	Sub-Entity type for Input .....	22

5.2.3	Input ports.....	22
5.2.4	Output ports.....	22
5.2.5	Meta data and reference tables.....	23
5.2.6	Example usage .....	23
5.2.6.1	Example output.....	23
5.3	<b>Check_PresetValues.....</b>	<b>24</b>
5.3.1	General description.....	24
5.3.2	Sub-Entity type for Input .....	24
5.3.3	Input ports.....	24
5.3.4	Output ports.....	25
5.3.5	Meta data and reference tables.....	25
5.3.6	Example usage .....	25
5.3.6.1	Example output.....	26
5.4	<b>Check_MissingAttributes .....</b>	<b>26</b>
5.4.1	General description.....	26
5.4.2	Sub-Entity type for Input .....	26
5.4.3	Input ports.....	26
5.4.4	Output ports.....	27
5.4.5	Meta data and reference tables.....	27
5.4.6	Example usage .....	27
5.4.6.1	Example output.....	28
6	<b>Product Code Validation Rules.....</b>	<b>28</b>
6.1	<b>Validate_GTIN .....</b>	<b>28</b>
6.1.1	General description.....	28
6.1.2	Input ports.....	28
6.1.3	Output ports.....	28
6.1.4	Meta data and reference tables.....	29
6.1.5	Example usage .....	29
6.1.5.1	Example output.....	29
6.2	<b>Validate_UPC.....</b>	<b>29</b>
6.2.1	General description.....	29
6.2.2	Input ports.....	29
6.2.3	Output ports.....	29
6.2.4	Meta data and reference tables.....	30

6.2.5	Example usage .....	30
6.2.5.1	Example output.....	30
<b>7</b>	<b>Classification Rules .....</b>	<b>30</b>
7.1	Classify_Product .....	30
7.1.1	General description.....	30
7.1.2	Input ports.....	31
7.1.3	Output ports.....	31
7.1.4	Meta data and reference tables.....	31
7.1.5	Example usage .....	31
7.1.5.1	Example output.....	32
7.2	Identify_Language .....	32
7.2.1	General description.....	32
7.2.2	Input ports.....	32
7.2.3	Output ports.....	32
7.2.4	Meta data and reference tables.....	33
7.2.5	Example usage .....	33
7.2.5.1	Example output.....	33
<b>8</b>	<b>Lookup and Standardization Rules.....</b>	<b>33</b>
8.1	Check_Profanity.....	33
8.1.1	General description.....	33
8.1.2	Input ports.....	34
8.1.3	Output ports.....	34
8.1.4	Meta data and reference tables.....	34
8.1.5	Example usage .....	34
8.1.5.1	Example output.....	34
8.2	Parse_Color .....	35
8.2.1	General description.....	35
8.2.2	Input ports.....	35
8.2.3	Output ports.....	35
8.2.4	Meta data and reference tables.....	36
8.2.5	Example usage .....	36
8.2.5.1	Example output.....	36
8.3	Standardize_Color .....	36
8.3.1	General description.....	36

8.3.2	Input ports.....	37
8.3.3	Output ports.....	37
8.3.4	Meta data and reference tables.....	37
8.3.5	Example usage .....	37
8.3.5.1	Example output.....	37
8.4	<b>Standardize_CompanyName .....</b>	<b>38</b>
8.4.1	General description.....	38
8.4.2	Input ports.....	38
8.4.3	Output ports.....	38
8.4.4	Meta data and reference tables.....	38
8.4.5	Example usage .....	39
8.4.5.1	Example output.....	39
8.5	<b>Standardize_UOM .....</b>	<b>39</b>
8.5.1	General description.....	39
8.5.2	Input ports.....	39
8.5.3	Output ports.....	39
8.5.4	Meta data and reference tables.....	40
8.5.5	Example usage .....	40
8.5.5.1	Example output.....	40
9	<b>Reference Tables.....</b>	<b>41</b>
9.1	Product 360 administrative reference tables .....	41
9.2	Reference tables for data look ups and standardizations .....	43

### **Out-of-the-Box DQ Rules for Product 360**

The Out-of-the-Box data quality rule package enables customers to use pre-built templates in order to get started quickly.

This document describes the use cases and set up for all standard data quality rules that are part of the Informatica Data Quality for Product 360 delivery.

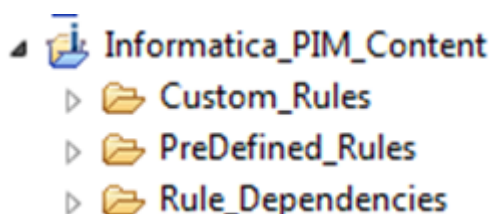
## 1 Project Structure within Informatica Data Quality (Design Environment)

The out-of-the-box DQ rules for Product 360 package is structured into two files that make up the Informatica Data Quality project that can be used in Product 360:

1. Informatica\_PIM\_Content.xml  
This file contains the rules itself in the defined structure. See details on the rules and their functionality under "Rule Types"
2. Informatica\_PIM\_Content.zip  
This file contains all the meta data like demo classifiers for the auto classification of items and reference tables used by some of the rules. See details on the reference tables and their functionality under "Reference Tables"


Importing the Product 360 rules into Informatica Data Quality is only necessary if you want to add additional custom rules for Product 360.

When importing the XML and the ZIP file into Informatica Data Quality it should reflect the following structure:



Only Maplets included in following 2 paths will be surfaced through the Product 360 UI:

- Informatica\_PIM\_Content\**Custom\_Rules**: Used for custom rules built by customer
- Informatica\_PIM\_Content\**PreDefined\_Rules**: Contains all pre-built / shipped rules for Product 360

 Please make sure to put custom rules always under **Custom\_Rules** as they won't be available through the UI if structured differently!

Without filtering of views, a Product 360 user is overwhelmed with the amount of maplets displayed and has no indication which ones are ready for consumption in Product 360 and which ones are nested maplets.

Place mapplets / objects that should not be surfaced to the user selection screen in Product 360 in **Rule\_Dependencies** folder or other project.

On import of the ZIP file all content for the sub-folder Rule\_Dependencies is transferred into the Informatica Data Quality project.

- Used for Product 360 specific rule dependencies – mapplets not shown to users, reference tables, content sets, etc.
- Use of other projects (e.g. Informatica\_DQ\_Content, custom project, etc.)
- Allowed and recommended to support single source of truth for rule design
- Limitation in Product 360 only applies to viewing rules from XML in the Product 360 User interface, all other dependencies are required during execution and included in exports

**i** Make sure custom rules are exported with a **different XML filename than the standard rules (Informatica\_PIM\_Content.xml)**. You can have either one XML for all custom rules (e.g. "Custom\_Rules\_of\_MyEnterprise.xml") or even better one XML for each of them ("my\_color\_standardization.xml", etc.). The export of custom rules should not include the **PreDefined\_Rules** folder of the project so you can update the pre-built rules with any Product 360 upgrade and without any additional effort just by using the XML shipped with the install packages.

## 1.1 Upgrading from one IDQ Version to another

In case the embedded IDQ SDK deployed inside Product 360 has received a version upgrade it is also needed to upgrade the IDQ design environment containing your custom DQ rules (Informatica Developer). This can be achieved by performing an upgrade of the IDQ instance itself to the higher version while having the rules used for Product 360 in the IDQ repository. The upgrade of Informatica Developer will automatically upgrade the rules to that higher version as well.

**i** For more information on upgrading the Informatica Developer (IDQ Server) to a higher version please refer to the standard documentation of Informatica Data Quality.

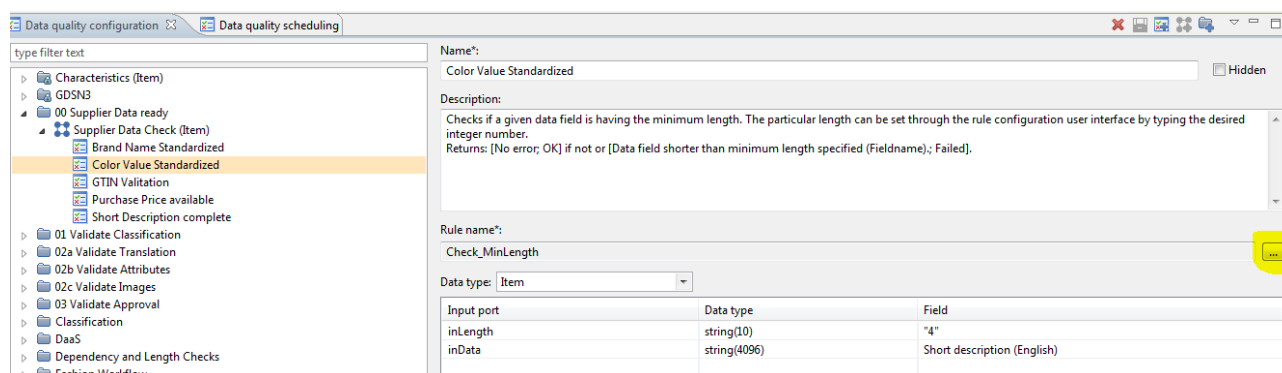
Next step is now to export your custom rules and potentially additional reference data as XML and ZIP.

**i** Make sure custom rules are exported with a **different XML filename than the standard rules (Informatica\_PIM\_Content.xml)**. You can have either one XML for all custom rules (e.g. "Custom\_Rules\_of\_MyEnterprise.xml") or even better one XML for each of them ("my\_color\_standardization.xml", etc.). The export of custom rules should not include the **PreDefined\_Rules** folder of the project so you can update the pre-built rules with any Product 360 upgrade and without any additional effort just by using the XML shipped with the install packages.

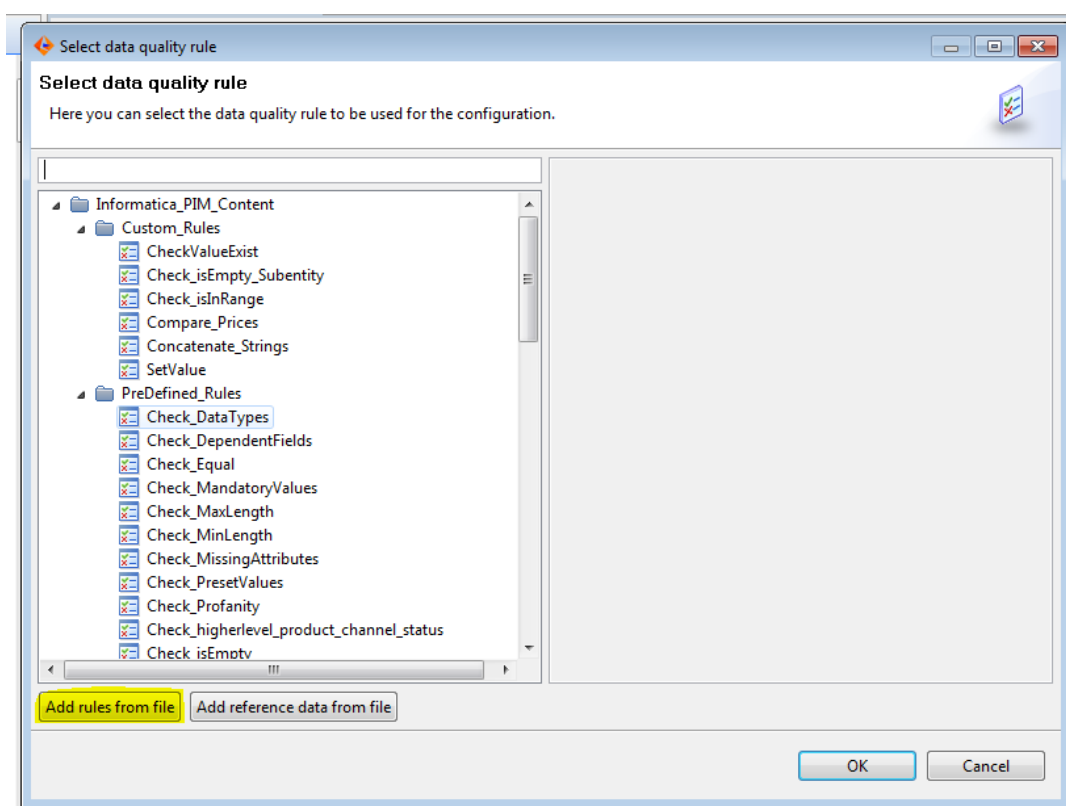
On the Product 360 Server the "dataquality" folder should be backed up now.

To upload the new XML version of your rules open the Desktop UI and navigate to the DQ perspective to open up the rule configuration dialog via the "..." button:





The dialog will allow you to upload the new XML files:



If you would have updated meta data as well you can additionally use the "Add reference data..." button from above screen to import them. If not you can skip that step (only needed if the reference data has changed).

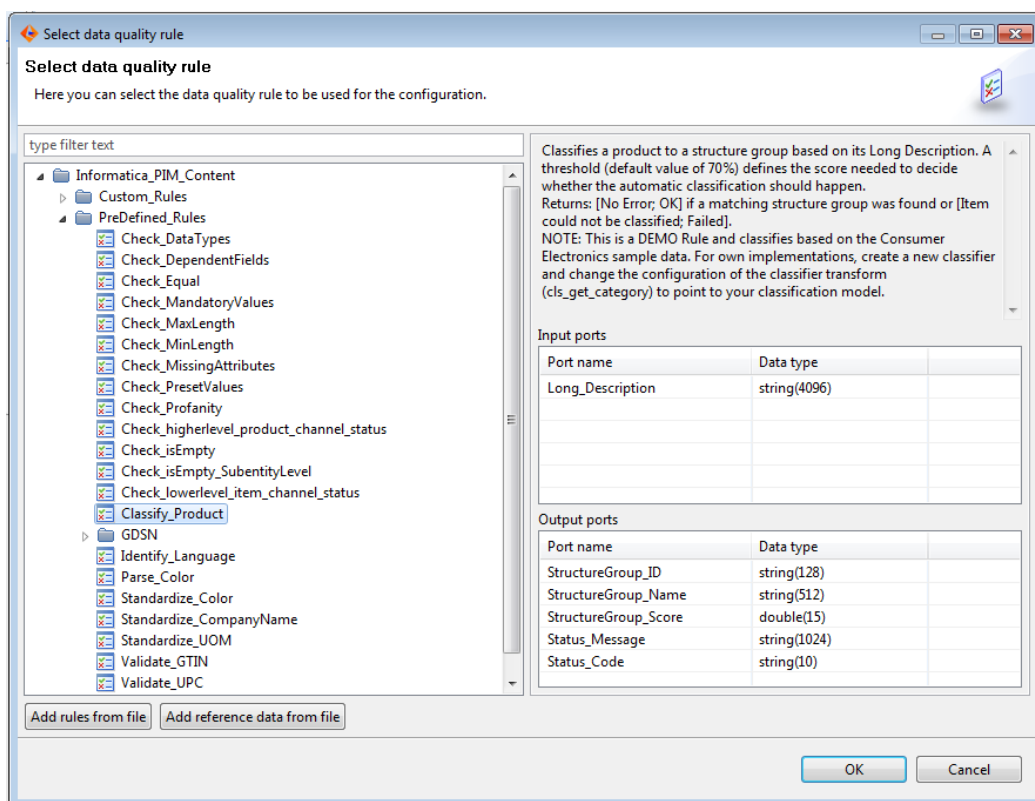
Upload all custom rule XML files from the upgraded IDQ environment.

Do this also with the pre-built rules XML ("Informatica\_PIM\_Content.xml") shipped with the version of Product 360 that carries the IDQ SDK update so you have the latest and greatest of the out of the box rules as well.

Lastly go to your "dataquality" folder on the server and check for any potential duplicate or dated XML files to remove them and avoid duplicate rules (remember to backup first). The Date modified may serve as hint on whether this is a new XML or a dated one. In case you found duplicates or dated rules it is recommended to perform a server restart in the end so any memory load of rules on the server is also cleaned up.

## 2 Importing Rules into Product 360

In order to use the rules in Product 360 it is needed to upload them through the UI. In order to upload the files it is needed to create a Rule Configuration within the "Data quality configuration" view. When selecting a rule the following dialog pops up:



Add the XML file via the "Add rules from file" button and add the ZIP file via the "Add reference data from file" button.

## 3 Rule types and reference tables

### 3.1 Generic rules

Basic data quality rules that can be used in multiple use cases and scenarios to validate any data of products, variants and items in Product 360.

- Check\_DependentFields
- Check\_Equal
- Check\_isEmpty
- Check\_isEmpty\_SubentityLevel (**since 8.1**)
- Check\_MaxLength
- Check\_MinLength

## 3.2 Structure Feature / Attribute rules

Data quality rules that can be used specifically for feature / attribute relational checks of products, variants and items stored in Product 360.

- Check\_DataTypes
- Check\_MandatoryValues
- Check\_PresetValues
- Check\_MissingAttributes

## 3.3 Product code validation rules

Data quality rules that can be used specifically to validate product code specific norms.

- Validate\_GTIN
- Validate\_UPC

## 3.4 Classification rules

Data quality rules that can be used specifically to classify objects based on the individual input values.

- Classify Product
- Identify\_Language

## 3.5 Lookup and standardization rules

Data Quality rules that can be used specifically to standardize input values based on reference table lookups.

- Check\_Profanity
- Parse\_Color
- Standardize\_Color
- Standardize\_CompanyName
- Standardize\_UOM

## 3.6 Reference tables

Reference tables are being used to store look up values and static information that will be accessed by the DQ rules during their runtime. Reference Tables that consist of one or two columns can be directly imported into Product 360 and maintained through the "Dictionary" perspective of the Desktop UI. Reference tables that consist of more than two columns will not show up on the Product 360 UI but directly be used by the DQ rules during runtime.

## 4 Generic Rules

Basic data quality rules that can be used in multiple use cases and scenarios to validate any product data.

- [Check\\_DependentFields](#)(see page 12)
- [Check\\_Equal](#)(see page 13)
- [Check\\_isEmpty](#)(see page 14)
- [Check\\_isEmpty\\_SubentityLevel](#)(see page 15)
- [Check\\_MaxLength](#)(see page 16)
- [Check\\_MinLength](#)(see page 18)

### 4.1 Check\_DependentFields

#### 4.1.1 General description

Checks if a dependent field has a value if its driver field is complete.

#### 4.1.2 Input ports

<b>inDriverField</b>	Data field that defines (based on whether it is empty or not) if the dependent field needs to be filled.
<b>inDependentField</b>	Dependent field which may not be empty if driver field has a value.

#### 4.1.3 Output ports

<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

#### 4.1.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

### 4.1.5 Example usage

If the attribute for "Hazardous Material" (inDriverField) is set to "true" the attribute value for "Risk & Security Sentences" (inDependentField) may not be empty

#### 4.1.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Checked field requires a value to be completed according to the dependent field.

## 4.2 Check\_Equal

### 4.2.1 General description

Checks if 2 field values assigned as input ports are equal.

### 4.2.2 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>inCheckField1</b>	Data field 1 that will be checked for equality against data field 2
<b>inCheckField2</b>	Data field 2 that will be checked for equality against data field 1

### 4.2.3 Output ports

<b>outObjectID</b>	Returns the identifier of the parent object. ( <i>needed to aggregate status messages from multiple attribute checks to a single parent object status message</i> )
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )

<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )
-----------------------	---

#### 4.2.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

#### 4.2.5 Example usage

Check if the value of Manufacturer (inCheckField1) is the same as for the attribute of brand name (inCheckField2)

##### 4.2.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	The input values are not equal.

### 4.3 Check\_isEmpty

#### 4.3.1 General description

Checks if a given data field is null or empty.

#### 4.3.2 Input ports

<b>Field_Value</b>	Any product data field that will be checked whether it is 'null' or empty.
--------------------	--

#### 4.3.3 Output ports

<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
--------------------	---

<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )
-----------------------	---

#### 4.3.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

#### 4.3.5 Example usage

Check if the "English short description" (Field\_Value) of an item is null or empty.

##### 4.3.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Input data value is blank or null.

### 4.4 [Check\\_isEmpty\\_SubentityLevel](#)

#### 4.4.1 General description

Checks if a given data field is null or empty. While running this rule on sub entities the inObjectID needs to be set to a valid object code of the parent.

Note: sub entities need to exist to get evaluated by this rule, if no sub entities exists, this rule will never be executed and status will automatically be set to OK.

#### 4.4.2 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>inCheckField</b>	Any data field on sub entity level that will be checked whether it is 'null' or empty.

### 4.4.3 Output ports

<b>Object_id</b>	Returns the identifier of the parent object. <i>(needed to aggregate status messages from multiple attribute checks to a single parent object status message)</i>
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). <i>(QualityStatusEntry.Status)</i>
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. <i>(QualityStatusEntry.Message)</i>

### 4.4.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

### 4.4.5 Example usage

Check if the any attribute value of an item is null or empty and report back failed if at least one of the attributes has no value set.

#### 4.4.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Input data value is blank or null.

## 4.5 Check\_MaxLength

### 4.5.1 General description

Checks if a given data field is exceeding the allowed length. The particular length can be set through the rule configuration user interface by typing the desired integer number into the "inLength" field.



## 4.5.2 Input ports

<b>inLength</b>	User Input of maximum length allowed
<b>inData</b>	Data field that will be checked for its length

## 4.5.3 Output ports

<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

## 4.5.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

## 4.5.5 Example usage

Check if the "English short description" (inData) is longer than the defined maximum length (inLength) specified.

### 4.5.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Data Field exceeds maximum length specified (10).

## 4.6 Check\_MinLength

### 4.6.1 General description

Checks if a given data field is having a minimum length. The particular length can be set through the rule configuration user interface by typing the desired integer number into the "inLength" field. Empty values are treated as zero-length and cause the rule to fail.

### 4.6.2 Input ports

<b>inLength</b>	User Input of minimum length needed.
<b>inData</b>	Data field that will be checked for its length

### 4.6.3 Output ports

<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

### 4.6.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

### 4.6.5 Example usage

Check if the "English short description" (inData) is shorter than the allowed minimum length (inLength) specified.

#### 4.6.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Data field shorter than minimum length specified (7).

## 5 Feature / Attribute Rules

Data quality rules that can be used specific for feature / attribute relational checks of products, variants and items stored in Product 360.

- [Check\\_DataTypes](#)(see page 19)
- [Check\\_MandatoryValues](#)(see page 21)
- [Check\\_PresetValues](#)(see page 24)
- [Check\\_MissingAttributes](#)(see page 26)

### 5.1 Check\_DataTypes

#### 5.1.1 General description

Check that the data type of an attribute is the same as the one of the assigned feature if the feature is not set to "character string" (e.g. The data type for the attribute "size" has to be set the same as for the assigned feature if the feature is not set to "Character String")

#### 5.1.2 Sub-Entity type for Input

Attributes

#### 5.1.3 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>inAttributeName</b>	Name of the attribute in the key language of the system. ( <i>ArticleAttribute:name</i> )
<b>inLanguage</b>	Language of the specific attribute value. ( <i>ArticleAttributeLang:language</i> )

<b>inStructure</b>	Structure system of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.structure</i> )
<b>inStructureGroup</b>	Structure group of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.structure</i> )
<b>inStructureFeature</b>	Feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.StructureAttribute</i> )
<b>inAttributeValue</b>	Attribute value in its specific language. ( <i>ArticleAttributeValue.Value</i> )
<b>inValueIdentifier</b>	Identifier of the attribute value. ( <i>ArticleAttributeValue.identifier</i> )
<b>inStructurePresetValue</b>	Preset values defined by the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.DomainValue</i> )
<b>inDataType</b>	Data type defined for the attribute. ( <i>ArticleAttribute.Datatype</i> )

#### 5.1.4 Output ports

<b>Object_id</b>	Returns the identifier of the parent object. ( <i>needed to aggregate status messages from multiple attribute checks to a single parent object status message</i> )
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

#### 5.1.5 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
<b>Language_Inclusion</b>	References all "languages" considered for the attribute value checks. The languages need to be written like their UI label text in the global standard language of the repository. (e.g. "English" and/or "German" if system language is set to "en" for English).

<b>Structure_checks_inclusion</b>	References the "identifier" values for each Structure System of Product 360 considered by the Rule.
<b>System_Language</b>	References the global standard language of the Product 360 repository (e.g. "en" for English) using a single line. Language needs to be value recognized by Product 360.
<b>java_structure_name_lookup</b>	Java transformation that synchronizes the database internal and external identifier (that is visible through the UI) of a structure system during rule execution.
<b>java_CheckAttribute</b>	Java Transformation that contains all the logic to gather the required values of the feature each attribute is assigned to.

### 5.1.6 Example usage

Check for all attributes of an item (that are assigned to a feature of a structure system mentioned in "Structure\_checks\_inclusion") whether their data type is the same as the data type of the assigned feature if the feature isn't set to "Character String".

#### 5.1.6.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	<p>Pattern: <i>[Structure system], [Structure group], Feature '[Feature name]': The data type ([Attribute data type]) of attribute '[Attribute name]' does not match the assigned feature data type ([Feature data type]). [line break; optional]</i></p> <p>Sample Data Consumer Electronics, HDTV Series, Feature 'USB 2.0':The data type (Character string) of attribute 'USB 2.0' does not match the assigned feature data type (Integer).</p>

## 5.2 Check\_MandatoryValues

### 5.2.1 General description

Check that the attribute has a value if assigned to a feature that is defined as mandatory (e.g. a T-Shirt item needs to have a value for the attribute "Size" if this is a mandatory feature in the "T-Shirt" structure group)

## 5.2.2 Sub-Entity type for Input

Attributes

## 5.2.3 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>inAttributeName</b>	Name of the attribute in the key language of the system. ( <i>ArticleAttribute:name</i> )
<b>inLanguage</b>	Language of the specific attribute value. ( <i>ArticleAttributeLang:language</i> )
<b>inStructure</b>	Structure system of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap:structure</i> )
<b>inStructureGroup</b>	Structure group of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap:structure</i> )
<b>inStructureFeature</b>	Feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.StructureAttribute</i> )
<b>inAttributeValue</b>	Attribute value in its specific language. ( <i>ArticleAttributeValue.Value</i> )
<b>inValueIdentifier</b>	Identifier of the attribute value. ( <i>ArticleAttributeValue:identifier</i> )
<b>inStructurePresetValue</b>	Preset values defined by the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.DomainValue</i> )

## 5.2.4 Output ports

<b>Object_id</b>	Returns the identifier of the parent object. ( <i>needed to aggregate status messages from multiple attribute checks to a single parent object status message</i> )
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )

<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )
-----------------------	---

### 5.2.5 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
<b>Language_Inclusion</b>	References all "languages" considered for the attribute value checks. The languages need to be written like their UI label text in the global standard language of the repository. (e.g. "English" and/or "German" if system language is set to "en" for English).
<b>Structure_checks_inclusion</b>	References the "identifier" values for each Structure System of Product 360 considered by the Rule.
<b>System_Language</b>	References the global standard language of the Product 360 repository (e.g. "en" for English) using a single line. Language needs to be value recognized by Product 360.
<b>java_structure_name_lookup</b>	Java transformation that synchronizes the database internal and external identifier (that is visible through the UI) of a structure system during rule execution.
<b>java_CheckAttribute</b>	Java Transformation that contains all the logic to gather the required values of the feature each attribute is assigned to.

### 5.2.6 Example usage

Check for all attributes of an item (that are assigned to a feature of a structure system mentioned in "Structure\_checks\_inclusion") whether they have a value (in the languages mentioned in "Language\_Inclusion") if the assigned feature is defined as mandatory.

#### 5.2.6.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	Pattern: <i>[Structure system], [Structure group], Feature '[Feature name]'</i> : The mandatory value <i>[Language]</i> of attribute ' <i>[Attribute name]</i> ' is missing. <i>[line break; optional]</i>

Sample Data Consumer Electronics, MultyVision ISIO, Feature 'Color': The mandatory value (French) of attribute 'Color' is missing.

## 5.3 Check\_PresetValues

### 5.3.1 General description

Check that the value of an attribute is conform to the preset values defined for the assigned feature (e.g. The value for the attribute "Size" may only be "S", "M", "L" or "XL" according to the preset values given by the assigned feature)

### 5.3.2 Sub-Entity type for Input

Attributes

### 5.3.3 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>inAttributeName</b>	Name of the attribute in the key language of the system. ( <i>ArticleAttribute:name</i> )
<b>inLanguage</b>	Language of the specific attribute value. ( <i>ArticleAttributeLang:language</i> )
<b>inStructure</b>	Structure system of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap:structure</i> )
<b>inStructureGroup</b>	Structure group of the feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap:structure</i> )
<b>inStructureFeature</b>	Feature the attribute is assigned to. ( <i>ArticleAttributeStructureGroupAttributeMap.StructureAttribute</i> )
<b>inAttributeValue</b>	Attribute value in its specific language. ( <i>ArticleAttributeValue.Value</i> )
<b>inValueIdentifier</b>	Identifier of the attribute value. ( <i>ArticleAttributeValue:identifier</i> )
<b>inStructurePresetValue</b>	Optional different value for the attribute. ( <i>ArticleAttributeStructureGroupAttributeMap.DomainValue</i> )



### 5.3.4 Output ports

<b>Object_id</b>	Returns the identifier of the parent object. <i>(needed to aggregate status messages from multiple attribute checks to a single parent object status message)</i>
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). <i>(QualityStatusEntry.Status)</i>
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. <i>(QualityStatusEntry.Message)</i>

### 5.3.5 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
<b>Language_Inclusion</b>	References all "languages" considered for the attribute value checks. The languages need to be written like their UI label text in the global standard language of the repository. (e.g. "English" and/or "German" if system language is set to "en" for English).
<b>Structure_checks_inclusion</b>	References the "identifier" values for each Structure System of Product 360 considered by the Rule.
<b>System_Language</b>	References the global standard language of the Product 360 repository (e.g. "en" for English) using a single line. Language needs to be value recognized by Product 360.
<b>java_structure_name_lookup</b>	Java transformation that synchronizes the database internal and external identifier (that is visible through the UI) of a structure system during rule execution.
<b>java_CheckAttribute</b>	Java Transformation that contains all the logic to gather the required values of the feature each attribute is assigned to.

### 5.3.6 Example usage

Check for all attributes of an item (that are assigned to a feature of a structure system mentioned in "Structure\_checks\_inclusion") whether their preset value is similar to one of the values from the list of the assigned feature.

### 5.3.6.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	<p>Pattern: <i>[Structure system], [Structure group], Feature '[Feature name]'</i>: The value '<i>[Attribute value]</i>' (<i>[Language]</i>, <i>[Identifier of the value]</i>) of attribute '<i>[Attribute name]</i>' doesn't conform to the preset values. <i>[line break; optional]</i></p> <p>Sample Data Consumer Electronics, HDTV Series, Feature 'USB 2.0': The value 'enabled' (English, DEFAULT) of attribute 'USB 2.0' doesn't conform to the preset values.</p>

## 5.4 Check\_MissingAttributes

### 5.4.1 General description

Checks that an item has attributes for all features defined by the structure group it is assigned to (e.g. a T-Shirt needs to have an attribute for "Size" if this is a feature in the "T-Shirt" structure group)

### 5.4.2 Sub-Entity type for Input

Structure assignments

### 5.4.3 Input ports

<b>inObjectID</b>	Identifier of the parent object. ( <i>Object_ID</i> )
<b>Structure</b>	Identifier of the assigned Structure. ( <i>ArticleStructureMap:structureId</i> )
<b>StructureGroup</b>	Structure group the object is assigned to. ( <i>ArticleStructureMap.StructureGroup</i> )

#### 5.4.4 Output ports

<b>Object_id</b>	Returns the identifier of the parent object. <i>(needed to aggregate status messages from multiple structure assignment checks to a single parent object status message)</i>
<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). <i>(QualityStatusEntry.Status)</i>
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. <i>(QualityStatusEntry.Message)</i>

#### 5.4.5 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
<b>Structure_checks_inclusion</b>	References the "identifier" values for each Structure System of Product 360 considered by the Rule.
<b>System_Language</b>	References the global standard language of the Product 360 repository (e.g. "en" for English) using a single line. Language needs to be value recognized by Product 360.
<b>java_StructureIdentifierLookup</b>	Java transformation that synchronizes the database internal and external identifier (that is visible through the UI) of a structure system during rule execution.
<b>java_missing_attributes_check</b>	Java Transformation that contains all the logic to gather the required values of the assignment to identify missing attributes.

#### 5.4.6 Example usage

Check for all items (that have an assignment to a structure system mentioned in "Structure\_checks\_inclusion") whether they have a corresponding attribute for each of the features of their assigned structure group.

#### 5.4.6.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	<p>Pattern: <i>[Structuresystem], [Structuregroup], Feature '[Feature name]'</i>: There is no attribute assigned. <i>[line break; optional]</i></p> <p>Sample Data Home Improvement, Tool boxes, Feature 'Version': There is no attribute assigned.</p> <p>Sample Data Home Improvement, Tool boxes, Feature 'Weight': There is no attribute assigned.</p>

## 6 Product Code Validation Rules

Data quality rules that can be used specifically to validate product code specific norms.

- [Validate\\_GTIN](#)(see page 28)
- [Validate\\_UPC](#)(see page 29)

### 6.1 [Validate\\_GTIN](#)

#### 6.1.1 General description

Validates any given GTIN8, GTIN12, GTIN13 or GTIN14 number including length, check sum, etc.

#### 6.1.2 Input ports

<b>In_Field_Value</b>	Any data field with a GTIN.
-----------------------	-----------------------------

#### 6.1.3 Output ports

<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

## 6.1.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

## 6.1.5 Example usage

Check if the GTIN (In\_Field\_Value) of an item is valid and consistent.

### 6.1.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	GTIN / EAN value is not valid.

## 6.2 [Validate\\_UPC](#)

### 6.2.1 General description

Validates any given Universal Product Code (UPC) number.

### 6.2.2 Input ports

<b>Field_Value</b>	Any data field with a UPC.
--------------------	----------------------------

### 6.2.3 Output ports

<b>Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

## 6.2.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

## 6.2.5 Example usage

Check if the UPC (Field\_Value) of an item is valid and consistent.

### 6.2.5.1 Example output

<b>Status_Code</b>	Failed
<b>Status_Message</b>	UPC value is not valid.

# 7 Classification Rules

Data quality rules that can be used specifically to classify objects based on their individual input values.

- [Classify\\_Product](#)(see page 30)
- [Identify\\_Language](#)(see page 32)

## 7.1 Classify\_Product

### 7.1.1 General description

Classifies a product to a structure group based on its "Long Description". A threshold (default value of 70%) defines the score needed to decide whether the classification should happen or not.

NOTE: This is a DEMO rule and classifies based on the "Sample Data Consumer Electronics" demo data. For own implementations, create a new classifier and change the configuration of the classifier transform (cls\_get\_category) to point to your classification model.

NLP Classifiers need to be trained with actual live data. This happens by mapping e.g. long descriptions of products to structure groups of Product 360. The broader the variety of long descriptions that have been mapped to a specific structure group value the better is the classification of new items working.

### 7.1.2 Input ports

<b>Long_Description</b>	Long description that will be used to classify the object against a certain structure group.
-------------------------	--

### 7.1.3 Output ports

<b>StructureGroup_ID</b>	Returns the identifier of the structure group found for the product.
<b>StructureGroup_Name</b>	Returns the English name of the structure group found for the product.
<b>StructureGroup_Score</b>	Returns the probability score for the classification to the structure group found. ( <i>QualityStatusEntry.Score</i> )
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

### 7.1.4 Meta data and reference tables

<b>cs_classify_products_demo</b>	Sample classifier based on the Consumer Electronics "Sample Data Consumer Electronics" demo data.
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

### 7.1.5 Example usage

Check an item for a proposed structure group of the "Sample Data Consumer Electronics" structure system.

The "English Long Description" (Long\_Description) of the item is:

*"46 inch multimedia television set (Full-HD) with high-quality aluminum front frame and contrast pane - comes equipped with Dual HDTV multi-tuner and digital video recorder (integrated hard drive)"*

Since the classifier of the rule has been trained with multiple possible long descriptions for all the structure groups of the system it suggests that the item should be assigned to the structure group called "HDTV Series"

### 7.1.5.1 Example output

<b>StructureGroup_ID</b>	1267688379455
<b>StructureGroup_Name</b>	HDTV Series
<b>StructureGroup_Score</b>	1.0000
<b>Status_Code</b>	OK
<b>Status_Message</b>	No Error

## 7.2 Identify\_Language

### 7.2.1 General description

Identifies the language of a text field by a probability score given after the identification. A threshold (default value of 70%) defines the score needed to decide whether the language classification was successful or not.

### 7.2.2 Input ports

<b>Text_Value</b>	Text field that will be checked for its language.
-------------------	---

### 7.2.3 Output ports

<b>Language</b>	Returns the language identified for the text field.
<b>Language_Score</b>	Returns the probability score for the identified language. ( <i>QualityStatusEntry.Score</i> )
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )



## 7.2.4 Meta data and reference tables

<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.
-----------------------------------	---

## 7.2.5 Example usage

Check whether the content of the "English Long Description" is actually written in English

The "English Long Description" (Text\_Value) of the item is:

*"46 inch multimedia television set (Full-HD) with high-quality aluminum front frame and contrast pane - comes equipped with Dual HDTV multi-tuner and digital video recorder (integrated hard drive)"*

### 7.2.5.1 Example output

<b>Language</b>	en
<b>Language_Score</b>	0.9487
<b>Out_Status_Code</b>	OK
<b>Out_Status_Message</b>	No Error

# 8 Lookup and Standardization Rules

Data quality rules that can be used specifically to standardize input values based on reference table lookups.

- [Check\\_Profanity](#)(see page 33)
- [Parse\\_Color](#)(see page 35)
- [Standardize\\_Color](#)(see page 36)
- [Standardize\\_CompanyName](#)(see page 38)
- [Standardize\\_UOM](#)(see page 39)

## 8.1 Check\_Profanity

### 8.1.1 General description

Parses inappropriate language and returns original text with profanity words removed.

### 8.1.2 Input ports

<b>Text_Value</b>	Text field that will be checked for profanities.
-------------------	--

### 8.1.3 Output ports

<b>Profanity_Value</b>	Returns the profanity value(s) found in the text.
<b>Masked_Data</b>	Returns the input text and exchanges profanity words with the value [CENSORED].
<b>Cleansed_Text</b>	Returns the input text with the profanity words removed.
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

### 8.1.4 Meta data and reference tables

<b>profanity_infa</b>	Dictionary that contains a variety of profanity values that are being used during rule execution for look ups.
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

### 8.1.5 Example usage

Check whether the "English Long Description" (Text\_Value) contains any profanity values referenced by "profanity\_infa".

The "English Long Description" of the item is:

*"HDTV with a crappy 50 GB hard drive"*

#### 8.1.5.1 Example output

<b>Profanity_Value</b>	crappy
------------------------	--------

<b>Masked_Data</b>	HDTV with a CENSORED 50 GB hard drive
<b>Cleansed_Text</b>	HDTV with a 50 GB hard drive
<b>Status_Code</b>	Failed
<b>Status_Message</b>	Profanity -crappy- was found in input data.

## 8.2 Parse\_Color

### 8.2.1 General description

Parses color names as determined by a reference table. The rule will return 2 color values – for each, as found in the data as well as a cleansed / cased version.

### 8.2.2 Input ports

<b>Text_Value</b>	Text field that will be parsed for color values.
-------------------	--

### 8.2.3 Output ports

<b>Standard_Text</b>	Returns the text field with standardized color values.
<b>Color_ID1</b>	Returns the name of the first color value found.
<b>Color_ID2</b>	Returns the name of the second color value found.
<b>Color_Standardized1</b>	Returns the standardized color value found for the first color.
<b>Color_Standardized2</b>	Returns the standardized color value found for the second color.
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

## 8.2.4 Meta data and reference tables

<b>colors_infa</b>	Dictionary that contains the color values in a standardized format.
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

## 8.2.5 Example usage

Parse out two color values from the "English long description" (Text\_Value) and exchange them with the standardized format of the same color (Standard\_Text).

The "English Long Description" of the item is:

*"This nice HDTV has a ferrari red frame and a cool black screen color."*

### 8.2.5.1 Example output

<b>Standard_Text</b>	This nice HDTV has a Ferrari Red frame and a Cool black screen color.
<b>Color_ID1</b>	ferrari red
<b>Color_ID2</b>	cool black
<b>Color_Standardized1</b>	Ferrari Red
<b>Color_Standardized2</b>	Cool black
<b>Status_Code</b>	OK
<b>Status_Message</b>	No Error

## 8.3 Standardize\_Color

### 8.3.1 General description

Returns a base color value (Base\_Color) for an input color value (Text\_Value).

### 8.3.2 Input ports

<b>Text_Value</b>	Color value that will be standardized.
-------------------	--

### 8.3.3 Output ports

<b>Base_Color</b>	Returns the standardized color value.
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

### 8.3.4 Meta data and reference tables

<b>colors_base_infa</b>	Reference table that maps different color values to a standardized color value (e.g. "Deep sky blue" to "Blue")
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

### 8.3.5 Example usage

Take the color value provided by the manufacturer of the item and derive a base color value out of it that can be used for web shop search classification of that particular item.

The attribute "Color" provide by the supplier has the value "*Midnight black*".

#### 8.3.5.1 Example output

<b>Base_Color</b>	Black
<b>Status_Code</b>	OK
<b>Status_Message</b>	No Error

## 8.4 Standardize\_CompanyName

### 8.4.1 General description

Standardizes a company name and additionally provides its acronym if possible.

### 8.4.2 Input ports

<b>CompanyName</b>	Company name that will be standardized.
--------------------	---

### 8.4.3 Output ports

<b>Standardized_CompanyName</b>	Returns the standardized company name.
<b>Acronym_ComapnyName</b>	Returns the acronym for the standardized company name.
<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

### 8.4.4 Meta data and reference tables

<b>usa_company_acronyms_infa</b>	Reference table that maps company names to their corresponding acronyms (e.g. "Hewlett-Packard Co" to "HP")
<b>usa_company_names_std_infa</b>	Reference table that maps company names to their standardized spelling format (e.g. "Abercrombie and Fitch" to "Abercrombie & Fitch Co")
<b>usa_company_sufx_abrv_infa</b>	Reference table that contains set of American suffixes for company names and their abbreviations (e.g. "Co.")
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

## 8.4.5 Example usage

Take the manufacturer name "Hewlett Packard" (CompanyName) attached to an item and standardize it to the defined format as determined by the reference table.

### 8.4.5.1 Example output

<b>Standardized_CompanyName</b>	Hewlett-Packard
<b>Acronym_CompanyName</b>	HP
<b>Status_Code</b>	OK
<b>Status_Message</b>	No Error

## 8.5 Standardize\_UOM

### 8.5.1 General description

Separates the quantity and unit of measure, and outputs the unstandardized and standardized values. It also outputs the full string with the Unit of Measure standardized.

### 8.5.2 Input ports

<b>Text_Value</b>	Text field to be checked for unit of measures.
-------------------	--

### 8.5.3 Output ports

<b>Standardized_Unit</b>	Returns the standardized unit of measure found in the text.
<b>Parsed_Unit</b>	Returns the unit of measure found in the text.
<b>Standardized_Text</b>	Returns the text field with standardized UOM values.
<b>Additional_Parsed_Values</b>	Returns any additional parsed values found in the text.
<b>Unparsed_Field</b>	Returns the part of the text that hasn't been parsed by the rule.

<b>Out_Status_Code</b>	Returns the overall Status Code after the rule execution (OK or Failed). ( <i>QualityStatusEntry.Status</i> )
<b>Out_Status_Message</b>	Returns the overall Status Message after the rule execution. ( <i>QualityStatusEntry.Message</i> )

## 8.5.4 Meta data and reference tables

<b>uom_infa</b>	Reference table that contains the unit values in a standardized format.
<b>Error_messages_by_Language</b>	Contains a 4 digit error code plus a language to indicate the preferred error message to be output.

## 8.5.5 Example usage

The "English Long Description" of the item that will be parsed by that rule is:

*"This nice HDTV weighs 10 kilogram and can be delivered in 24 hours."*

### 8.5.5.1 Example output

<b>Standardized_Unit</b>	10 kg
<b>Parsed_Unit</b>	10 kilogram
<b>Standardized_Text</b>	This nice HDTV weighs 10 kg and can be delivered in 24 hrs.
<b>Additional_Parsed_Values</b>	24 hours
<b>Unparsed_Field</b>	This nice HDTV weighs and can be delivered hours.
<b>Status_Code</b>	OK
<b>Status_Message</b>	No Error




## 9 Reference Tables

Reference tables are being used to store look up values and static information that will be accessed by the DQ rules during their runtime. Reference Tables that consist of one or two columns can be directly imported into Product 360 and maintained through the "Dictionary" perspective of the rich client. Reference tables that consist of more than two columns will not show up on the Product 360 UI but directly be used by the DQ rules during runtime.

- [Product 360 administrative reference tables](#)(see page 41)
- [Reference tables for data look ups and standardizations](#)(see page 43)

### 9.1 Product 360 administrative reference tables

Name	Description	Importable into Product 360?						
Custom_Error_messages_by_language	<p>Reference table that contains the error messages for custom rules.</p> <div> Important: Please add error messages for custom rules always into this table.</div> <p>Contains a 4 digit error code plus a language to indicate the preferred error message to be output.</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>Error_message</td><td>Text that builds up the error message of a custom DQ rule in case of a "failed" status result.</td></tr><tr><td>Error_code</td><td>Identifier that is used by the custom DQ rule to reference the error_message.</td></tr></table>	Field	Description	Error_message	Text that builds up the error message of a custom DQ rule in case of a "failed" status result.	Error_code	Identifier that is used by the custom DQ rule to reference the error_message.	Yes
Field	Description							
Error_message	Text that builds up the error message of a custom DQ rule in case of a "failed" status result.							
Error_code	Identifier that is used by the custom DQ rule to reference the error_message.							
Error_messages_by_Language	<p>Reference table that contains the error messages for standard rules.</p> <p>Contains a 4 digit error code plus a language to indicate the preferred error message to be output.</p>	Yes						

	<table><tr><th>Field</th><th>Description</th></tr><tr><td>Error_message</td><td>Text that builds up the error message of a DQ rule in case of a "failed" status result.</td></tr><tr><td>Error_code</td><td>Identifier that is used by the DQ rule to reference the error_message.</td></tr></table>	Field	Description	Error_message	Text that builds up the error message of a DQ rule in case of a "failed" status result.	Error_code	Identifier that is used by the DQ rule to reference the error_message.	
Field	Description							
Error_message	Text that builds up the error message of a DQ rule in case of a "failed" status result.							
Error_code	Identifier that is used by the DQ rule to reference the error_message.							
Language_Inclusion	<p>References all "languages" considered for the attribute value checks. The languages need to be written like their UI label text in the global standard language of the repository. (e.g. "English" and "German" if system language is set to "en, de").</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>language_list</td><td>Semi colon separated list of languages that are included into the DQ checks that use this reference table (e.g. "English;German;French").</td></tr><tr><td>lookup_val</td><td>Fixed string "lang_list" that is used as an identifier for the DQ rules.</td></tr></table>	Field	Description	language_list	Semi colon separated list of languages that are included into the DQ checks that use this reference table (e.g. "English;German;French").	lookup_val	Fixed string "lang_list" that is used as an identifier for the DQ rules.	Yes
Field	Description							
language_list	Semi colon separated list of languages that are included into the DQ checks that use this reference table (e.g. "English;German;French").							
lookup_val	Fixed string "lang_list" that is used as an identifier for the DQ rules.							
Structure_checks_inclusion	<p>References the "identifier" values for each Structure System of Product 360 considered by rules that do structure/feature/attribute checks.</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>flag</td><td>Indicates whether the structure system will be taken into consideration by the DQ rule that uses this reference table.</td></tr><tr><td>Structure_identifier</td><td>List of all structure system identifiers that could be taken into consideration by the DQ</td></tr></table>	Field	Description	flag	Indicates whether the structure system will be taken into consideration by the DQ rule that uses this reference table.	Structure_identifier	List of all structure system identifiers that could be taken into consideration by the DQ	Yes
Field	Description							
flag	Indicates whether the structure system will be taken into consideration by the DQ rule that uses this reference table.							
Structure_identifier	List of all structure system identifiers that could be taken into consideration by the DQ							

	<table><tr><th>Field</th><th>Description</th></tr><tr><td></td><td>rule that uses this reference table.</td></tr></table>	Field	Description		rule that uses this reference table.			
Field	Description							
	rule that uses this reference table.							
System_Language	<p>References the global standard language of the Product 360 repository (e.g. "en" for English) using a single line. Language needs to be value recognized by Product 360.</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>SysLang</td><td>The short spelling of the overall system language / repository language (e.g. "en" for English). This will have effect on data access through Web API during rule execution and the language the error messages are written back into the object status.</td></tr><tr><td>Lookup_value</td><td>Fixed string "sys_lang" that is used as an identifier for the DQ rules.</td></tr></table>	Field	Description	SysLang	The short spelling of the overall system language / repository language (e.g. "en" for English). This will have effect on data access through Web API during rule execution and the language the error messages are written back into the object status.	Lookup_value	Fixed string "sys_lang" that is used as an identifier for the DQ rules.	Yes
Field	Description							
SysLang	The short spelling of the overall system language / repository language (e.g. "en" for English). This will have effect on data access through Web API during rule execution and the language the error messages are written back into the object status.							
Lookup_value	Fixed string "sys_lang" that is used as an identifier for the DQ rules.							

## 9.2 Reference tables for data look ups and standardizations

Name	Description	Importable into Product 360?						
color_base_infa	Reference table that maps different color values to a standardized color value (e.g. "Deep sky blue" to "Blue")	Yes						
	<table><tr><th>Field</th><th>Description</th></tr><tr><td>column1</td><td>The base color value (e.g. "Blue") a DQ rule that uses this table will take for standardization.</td></tr><tr><td>column2</td><td>The variation of the color value (e.g. "Deep sky blue") a DQ rule uses as look up</td></tr></table>		Field	Description	column1	The base color value (e.g. "Blue") a DQ rule that uses this table will take for standardization.	column2	The variation of the color value (e.g. "Deep sky blue") a DQ rule uses as look up
	Field		Description					
	column1		The base color value (e.g. "Blue") a DQ rule that uses this table will take for standardization.					
column2	The variation of the color value (e.g. "Deep sky blue") a DQ rule uses as look up							

Name	Description	Importable into Product 360?														
	<table><tr><th>Field</th><th>Description</th></tr><tr><td></td><td>for a base color value return.</td></tr></table>	Field	Description		for a base color value return.											
Field	Description															
	for a base color value return.															
colors_infa	<p>Dictionary that contains the color values in a standardized format.</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>column1</td><td>The base color value (e.g. "Blue") a DQ rule that uses this table will look up to identify a color string.</td></tr><tr><td>column2</td><td>Variation of the base color.</td></tr><tr><td>column3</td><td>Variation of the base color.</td></tr><tr><td>column4</td><td>Variation of the base color.</td></tr><tr><td>column5</td><td>Variation of the base color.</td></tr><tr><td>column6</td><td>Variation of the base color.</td></tr></table>	Field	Description	column1	The base color value (e.g. "Blue") a DQ rule that uses this table will look up to identify a color string.	column2	Variation of the base color.	column3	Variation of the base color.	column4	Variation of the base color.	column5	Variation of the base color.	column6	Variation of the base color.	No
Field	Description															
column1	The base color value (e.g. "Blue") a DQ rule that uses this table will look up to identify a color string.															
column2	Variation of the base color.															
column3	Variation of the base color.															
column4	Variation of the base color.															
column5	Variation of the base color.															
column6	Variation of the base color.															
profanity_infa	<p>Dictionary that contains a variety of profanity values that are being used during rule execution for look ups.</p> <table><tr><th>Field</th><th>Description</th></tr><tr><td>COLUMN 1</td><td>The profanity word a DQ rule that uses this table will look up to identify bad words.</td></tr><tr><td>COLUMN 2</td><td>The profanity word a DQ rule that uses this table will look up to identify bad words (the same value as in column 1).</td></tr></table>	Field	Description	COLUMN 1	The profanity word a DQ rule that uses this table will look up to identify bad words.	COLUMN 2	The profanity word a DQ rule that uses this table will look up to identify bad words (the same value as in column 1).	Yes								
Field	Description															
COLUMN 1	The profanity word a DQ rule that uses this table will look up to identify bad words.															
COLUMN 2	The profanity word a DQ rule that uses this table will look up to identify bad words (the same value as in column 1).															
uom_infa	Reference table that contains the unit values in a standardized format.	No														

Name	Description		Importable into Product 360?						
	Field	Description							
	column1	The base unit (e.g. "sec") a DQ rule that uses this table will look up to identify unit.							
	column2	Variation of the base unit.							
	column3	Variation of the base unit.							
	column4	Variation of the base unit.							
	column5	Variation of the base unit.							
usa_company_acronyms_infa	Reference table that maps company names to their corresponding acronyms (e.g. "Hewlett-Packard Co" to "HP") <table><tr><th>Field</th><th>Description</th></tr><tr><td>column1</td><td>The abbreviation of a given company name (e.g. "hp") a DQ rule uses as look up for a standardized company name value return.</td></tr><tr><td>column2</td><td>The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.</td></tr></table>		Field	Description	column1	The abbreviation of a given company name (e.g. "hp") a DQ rule uses as look up for a standardized company name value return.	column2	The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.	Yes
Field	Description								
column1	The abbreviation of a given company name (e.g. "hp") a DQ rule uses as look up for a standardized company name value return.								
column2	The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.								
usa_company_names_std_inf	Reference table that maps company names to their standardized spelling format (e.g. "Abercrombie and Fitch" to "Abercrombie & Fitch Co") <table><tr><th>Field</th><th>Description</th></tr><tr><td>column1</td><td>The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.</td></tr></table>		Field	Description	column1	The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.	No		
Field	Description								
column1	The standardized company name (e.g. "Hewlett Packard") a DQ rule that uses this table will take for standardization.								

Name	Description		Importable into Product 360?
	<b>Field</b>	<b>Description</b>	
	column2	Variation of the company name.	
	column3	Variation of the company name.	
	column4	Variation of the company name.	
	column5	Variation of the company name.	
	column6	Variation of the company name.	
	etc.	etc.	
usa_company_sufx_a brv_infa	Reference table that contains set of American suffixes for company names and their abbreviations (e.g. "Co.")		No
	<b>Field</b>	<b>Description</b>	
	column1	The standardized company suffix (e.g. "Inc.") a DQ rule that uses this table will take for standardization.	
	column2	Variation of the company suffix .	
	column3	Variation of the company suffix .	
	column4	Variation of the company suffix .	
	column5	Variation of the company suffix .	
	column6	Variation of the company suffix .	
	etc.	etc.	

# Copyright

Copyright (c) 1993-2020 Informatica LLC. All rights reserved.

This Software and documentation contain proprietary information of Informatica LLC and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC. This Software may be protected by [U.S. and/or](#) international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013©(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14

(ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica LLC in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved.

Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright

© EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved.

This product includes software developed by the Apache Software Foundation ( <http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla ( <http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at [http:// www.gnu.org/licenses/lgpl.html](http://www.gnu.org/licenses/lgpl.html). The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (©) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, < [daniel@haxx.se](mailto:daniel@haxx.se) >. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http:// curl.haxx.se/docs/ copyright.html](http://curl.haxx.se/docs/copyright.html). Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http://www.dom4j.org/ license.html](http://www.dom4j.org/license.html).

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http://dojotoolkit.org/ license](http://dojotoolkit.org/license).

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at [http://www.gnu.org/software/kawa/ Software-License.html](http://www.gnu.org/software/kawa/Software-License.html).

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost ( <http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http:// www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt).

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>. This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at [http:// www.eclipse.org/org/documents/ epl-v10.php](http://www.eclipse.org/org/documents/epl-v10.php) and at <http://www.eclipse.org/org/documents/edl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, [http:// www.stlport.org/doc/ license.html](http://www.stlport.org/doc/license.html), <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqLicense.html>, <http://unit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html), [http://www.openldap.org/software/release/ license.html](http://www.openldap.org/software/release/license.html), <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, [http:// fusesource.com/downloads/license- agreements/fuse-message-broker-v-5-3- licenseagreement](http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-licenseagreement); [http:// antlr.org/license.html](http://antlr.org/license.html); <http://aopalliance.sourceforge.net/>; <http://www.bouncycastle.org/licence.html>; <http://www.jgraph.com/jgraphdownload.html>; <http://www.jcraft.com/jsch/LICENSE.txt>; [http://jotm.objectweb.org/ bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html); [http://www.w3.org/Consortium/Legal/ 2002/copyright-software-20021231](http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231); [http://](http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231)



[www.slf4j.org/license.html](http://www.slf4j.org/license.html); <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/jaservice/>; <http://www.postgresql.org/about/licence.html>; <http://www.sqlite.org/copyright.html>; <http://www.tcl.tk/software/tcltk/license.html>; <http://www.jaxen.org/faq.html>; <http://www.jdom.org/docs/faq.html>; <http://www.slf4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt); <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; and <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>) the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSDLicense (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mitlicense.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 2003-2006 Joe Walnes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.

This product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,243,110; 7,254,590; 7,281,001; 7,421,458; 7,496,588; 7,523,121; 7,584,422; 7,676,516; 7,720,842; 7,721,270; 7,774,791; 8,065,266; 8,150,803; 8,166,048; 8,166,071; 8,200,622; 8,224,873; 8,271,477; 8,327,419; 8,386,435; 8,392,460; 8,453,159; 8,458,230; 8,707,336; 8,886,617 and RE44,478, International Patents and other Patents Pending.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.iv>

## NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY,

MISREPRESENTATION AND OTHER TORTS.