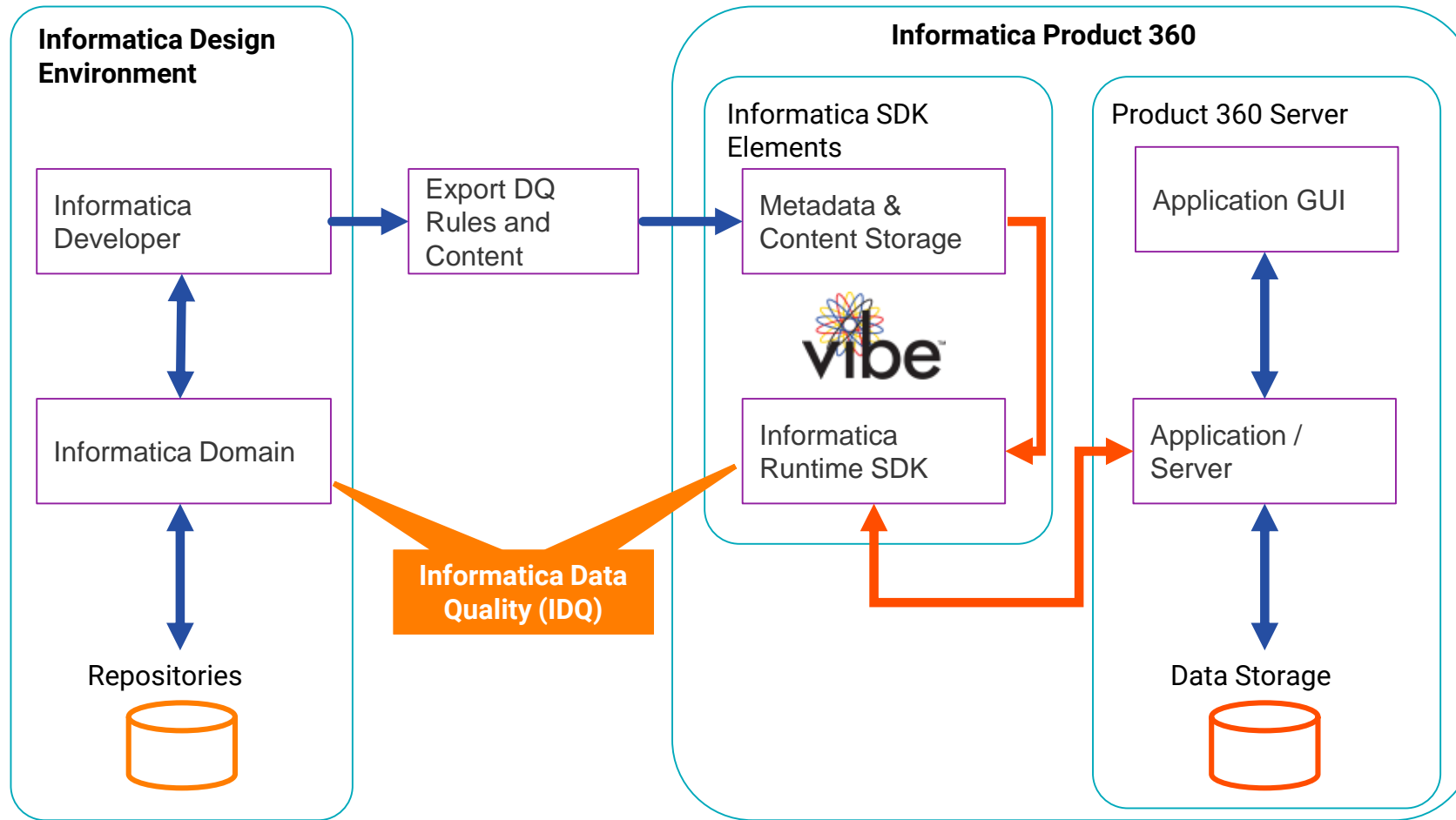


Best Practice Guide

# IDQ Rules for Product 360

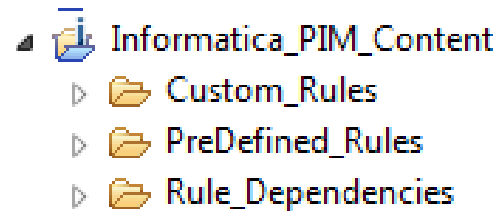
# Architecture Overview



**Versions in Product 360 and Design need to be the same**

# Project Structures 1/2

## Default Project Structure for IDQ Rules for Product 360



- Only Mapplets included in following 2 paths will be surfaced through Product 360 UI:
  - **Informatica\_PIM\_Content\Custom\_Rules**: Used for custom rules built by customer
  - **Informatica\_PIM\_Content\PreBuilt\_Rules**: Contains all pre-built / shipped rules for Product 360

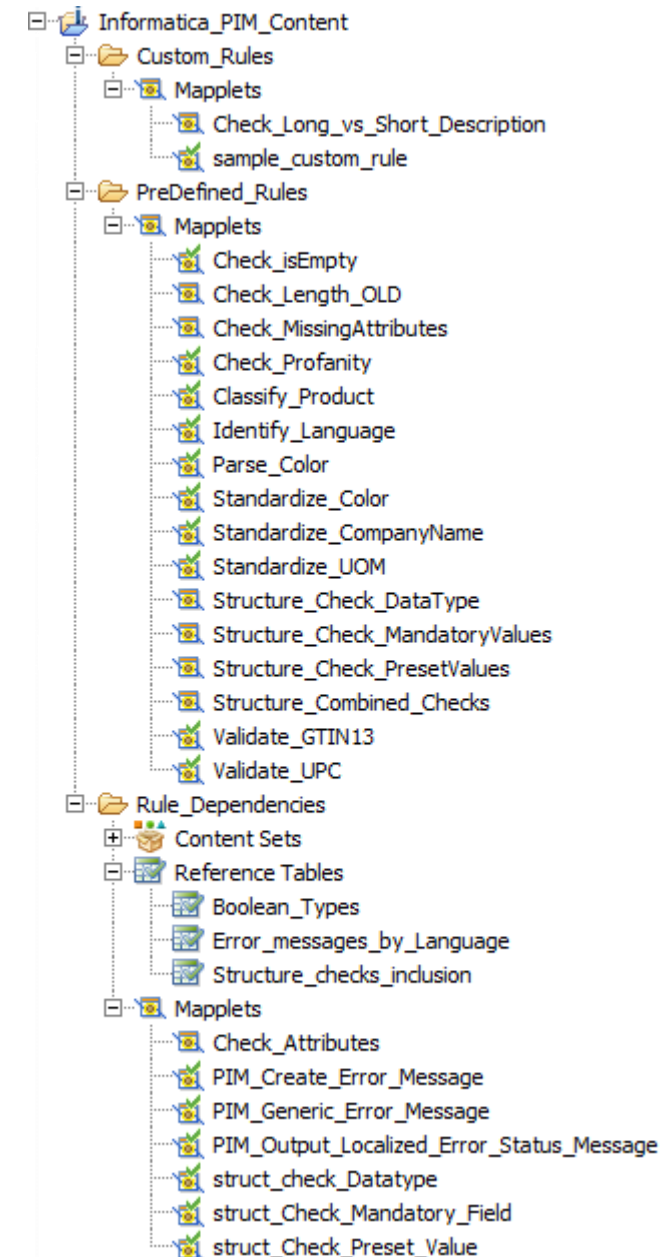
# Project Structures 2/2

- Informatica\_PIM\_Content\Rule\_Dependencies
  - 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

# Project Structures

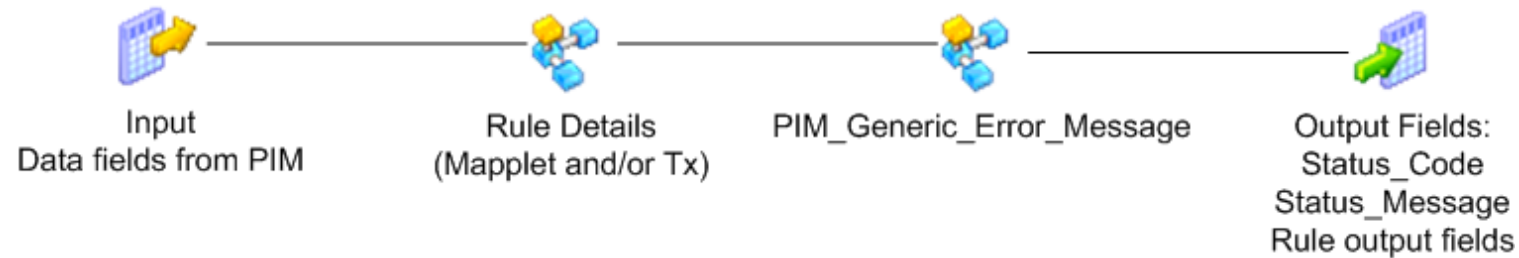
## Background

- Without filtering of views, Product 360 business users are overwhelmed with amount of mapplets displayed and have no indication which ones are ready for consumption in Product 360 and which ones are nested mapplets



# Rules for PIM – General Structure

## High Level View of Product 360 Rules



Item	Fields	Detail
Input	Dependent on Rule	Data that is expected to be received from Product 360
Rule Details	Rule Logic, can be mapplet or Tx	Dependent on requirements
Error Message	Pre-Built Mapplet: (PIM_Generic_Error_Message)	Standard Approach for error message generation (see further slides)
Output	Status_Code and Status_Message are required for Product 360 status settings	Additional outputs dependent on rule (e.g. standardised value)

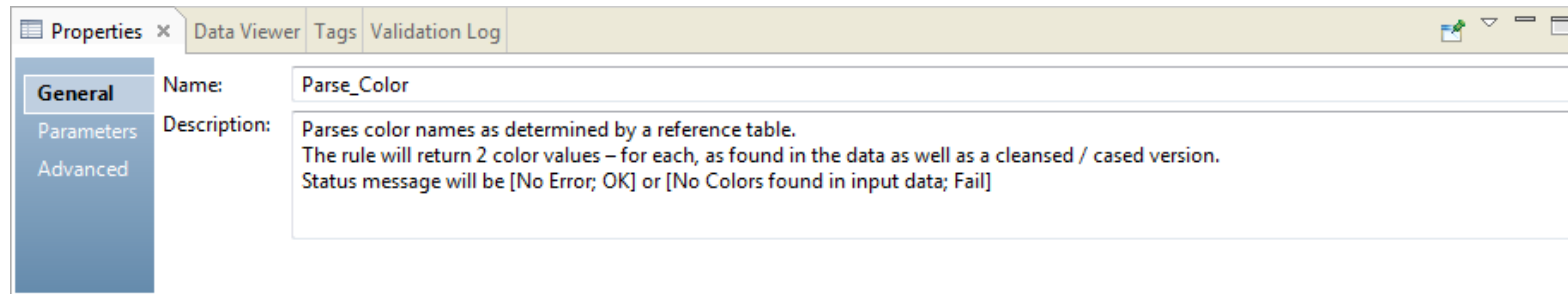
# Base setup and Error Handling

- Save new / custom rules in **Custom\_Rules** folder
- New Rules can use existing mapplets in all folders in Informatica\_PIM\_Rules folder and other projects
- Place mapplets / objects that should not be surfaced to the user selection screen in Product 360 in **Rule\_Dependencies** folder or other project
- Set up error reporting for rule:
  - Enter new value in reference table *Error\_messages\_by\_Language*
  - Re-Use *PIM\_Generic\_Error\_Message* mapplet for error message generation

# Naming Conventions

## Rules and Descriptions

- **Rule Names** should be business friendly, for example
  - **Check\_isEmpty** instead of **rule\_Completeness**
- Set **Description** for maplet including special instructions if applicable





# Naming Conventions

## Port Names for Input/Output

- Set **Descriptions for Port Names** on input and output transformations
  - These may be surfaced as tool-tips in Product 360

	Name	Type	Precision	Scale	Default Value	Description
1	Standard_Text	string	200	0		Returns the text field with standardized color values.
2	Color_ID1	string	30	0		Returns the name of the first color value found.
3	Color_ID2	string	30	0		Returns the name of the second color found.
4	Color_Standardize...	string	40	0		Returns the standardized color value for the first color.
5	Color_Standardize...	string	40	0		Returns the standardized color value for the second color.
6	Status_Message	string	1024	0		Returns the overall Status Message after the rule execution.
7	Status_Code	string	10	0		Returns the overall Status Code after the rule execution.

	Name	Type	Precision	Scale	Default Value	Description
Group (1)						
1	CompanyName	string	200	0		Company name that will be standardized.

# Output Status Codes

Product 360 environment expects 2 status values

- Status\_Code: Indicates whether the rule passed or failed. Use 1 for pass, 0 for fail
- Status\_Message: Error message associated to rule, default for pass is “No Error.” for fail use pre-defined mapplet to generate error messages

	Name	Type	Precisi...	Scale	Default Value	Description
1	Status_Message	string	1024	0		Returns the overall Status Message after the rule execution. ⓘ
2	Status_Code	string	10	0		Returns the overall Status Code after the rule execution.

# Setting Rule Status

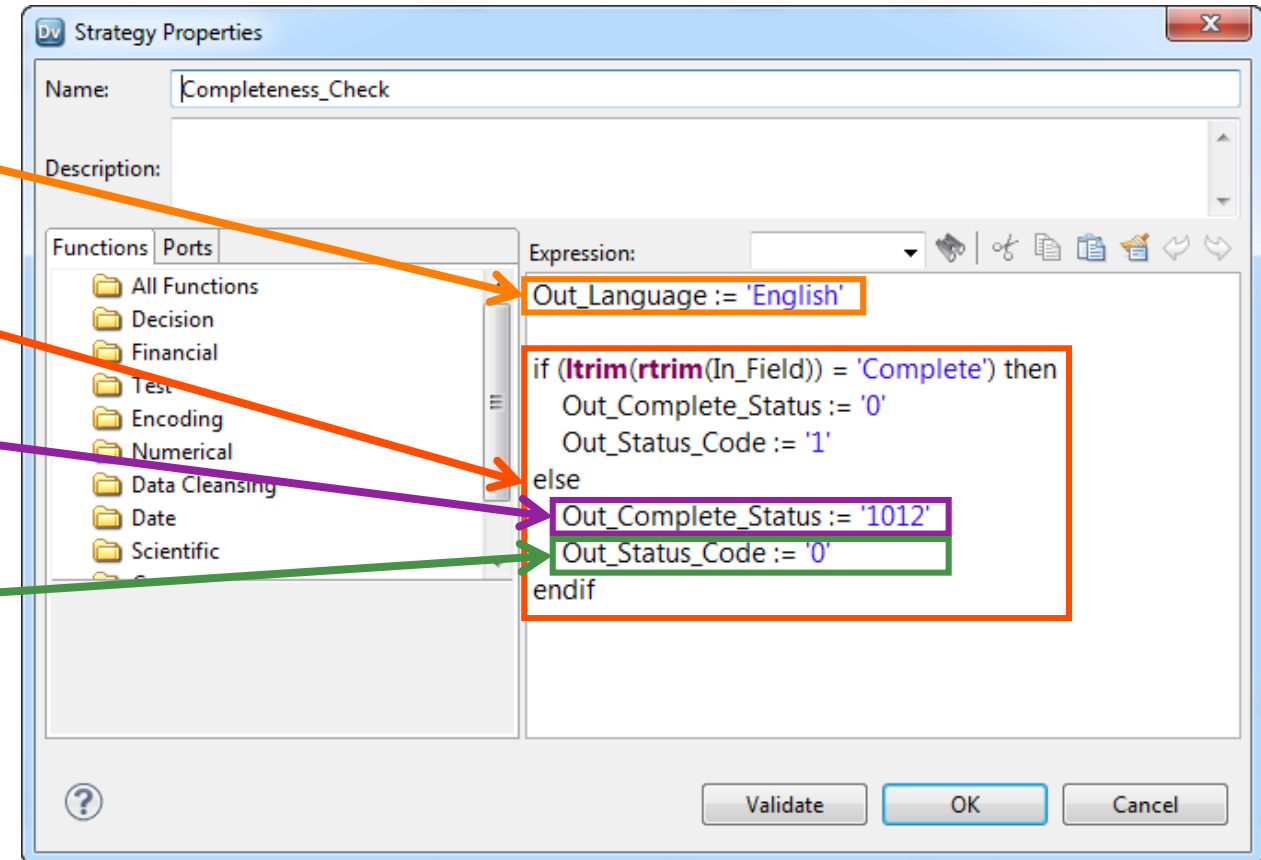
- Usually Set through Decision Tx
- Example: (same approach used for all pre-built rules)

**Set output language  
(Hard-Coded to English  
for all pre-built rules)**

**Error Logic**

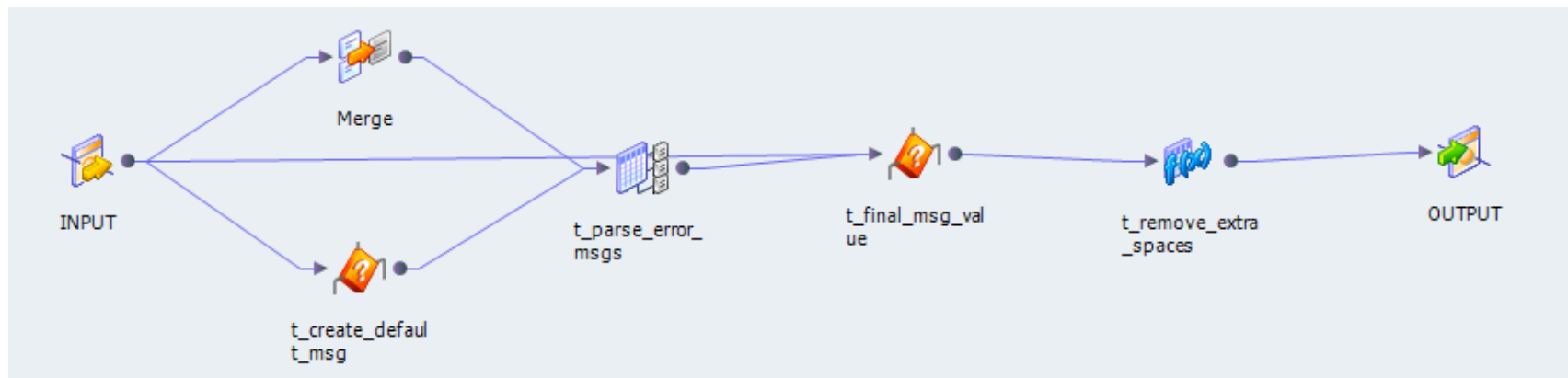
**Rule Error Code for  
lookup in Reference  
Table**

**Rule overall status for  
PIM System**



# Error Message Generation

- Based on standard mapplet:  
*PIM\_Generic\_Error\_Message*
- Used to generate locale specific error message for Product 360
  - Based on Reference Table *Error\_messages\_by\_Language*
  - Pre-built rules set up for English Messages only
  - Multiple languages require additional logic and entries in reference table



# Managing Error Messages

- Rule *PIM\_Output\_Localized\_Error\_Status\_Message* contains import ports that can be used in error message generation
- Based on entries in Reference Table *Error\_messages\_by\_Language* which reflects input fields
  - Each rule needs to have unique Error\_Code

Error_message	Error_code	Error_code_alt
1 Maximum Length value provided (  param1  ) is not a valid integer.	1020_English	[NULL]
2 Data Field exceeds maximum length specified (  param1  ).	1021_English	[NULL]
3 Profanity -  Object_no  - was found in input data.	1014_English	[NULL]
4   Object_no   is missing attributes for structure group,   structgrp  , a...	1003_English	[NULL]
5   Object_no   was not found in the reference table.	1007_English	[NULL]
6   param1  ,   param2  , Feature "  param3  ": The value   param6  ...	1001_English	[NULL]
7   param1  ,   param2  , Feature	1000_English	[NULL]
8 Pflichtfeld für Objekt   Object_no   ist nicht für Struktur komplette Gr...	1000_German	1000_Deutsch
9 Input data scored below threshold level and is not classified.	1010_English	[NULL]
10 No UOM found in input data value.	1013_English	[NULL]
11 GTIN / EAN value   Object_no   is not valid.	1011_English	[NULL]
12 No Colors found in input data.	1008_English	[NULL]
13 Input data value is blank or null.	1012_English	[NULL]
14 No Error.	0_English	[NULL]
15 UPC value   Object_no   is not valid.	1009_English	[NULL]
16 Demo custom check error message	2000_English	[NULL]
17 Item could not be auto-classified.	1015_English	[NULL]
18   param1  ,   param2  , Feature	1016_English	[NULL]
19 For product   param1  , the Short and Long Descriptions are the same.	2005_English	[NULL]

**Error message with parameters  
between double-pipes ||**

**Lookup Value for error Message**

# Deploying IDQ Rules to Product 360

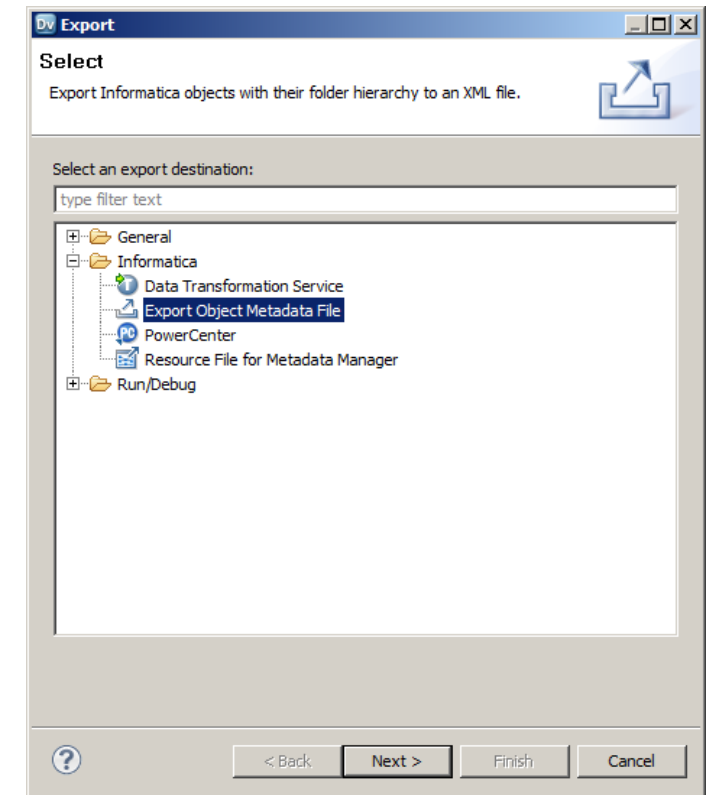
## Process Overview



# Deploying IDQ Rules to PIM

## IDQ Actions

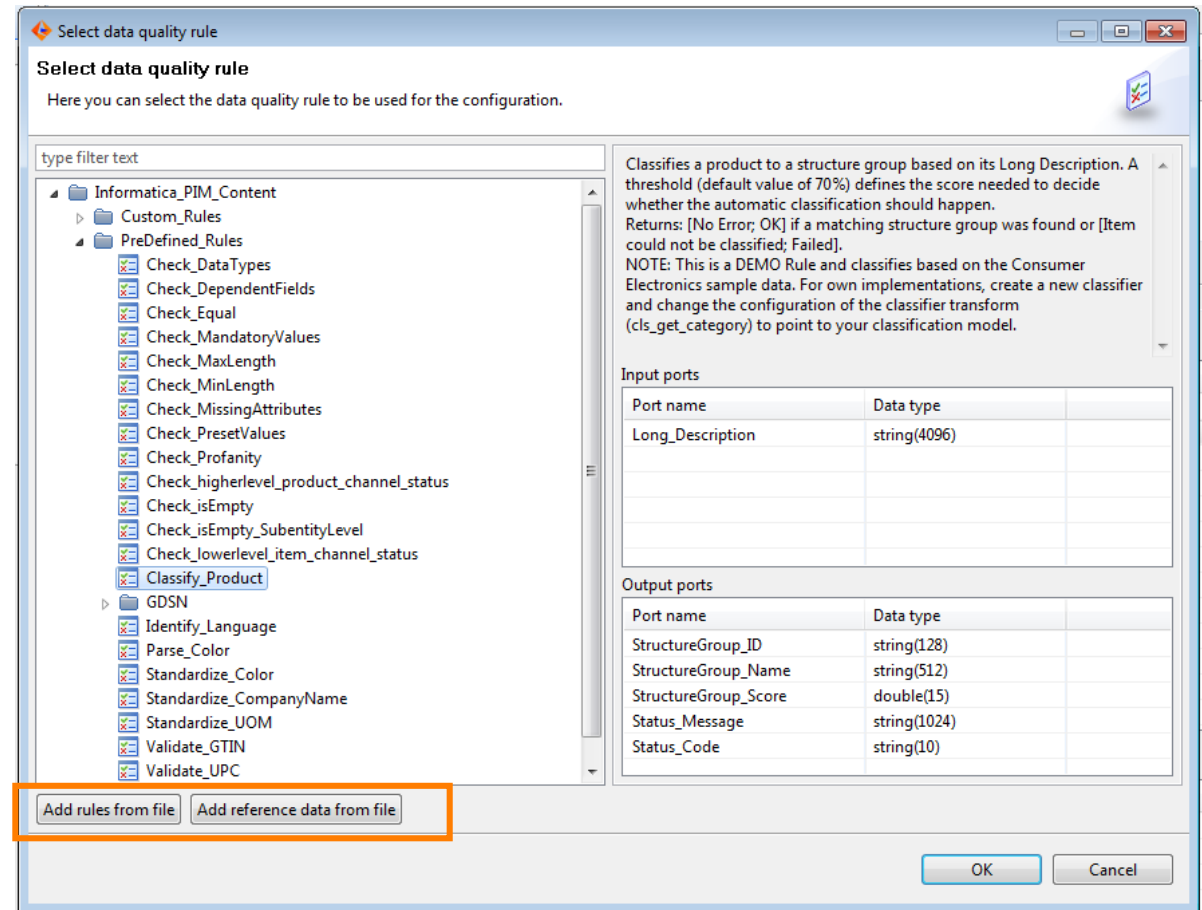
- Use Standard export option “**Export Object Metadata File**”
  - Include export of Reference Data in zip file
    - For NLP models, trained only option is OK
  - One or Multiple export files may be used if required
- 
- Custom rules should be exported with a **dedicated XML filename**
    - Either one XML for all custom rules (e.g. “Custom\_Rules\_of\_MyEnterprise.xml”) or one for each rule (“my\_color\_standardization.xml”, etc.). The export should not include the **PreDefined\_Rules** folder so you can update the pre-built rules with any Product 360 upgrade and without any additional effort by using the XML shipped with the install packages.



# Deploying IDQ Rules to PIM

## File Transfer

- FTP / Copy XML and ZIP files from IDQ to location readable by Product 360
- Use Product 360 UI to import metadata (xml) and reference data (zip)





# Use of Product 360 Service API in Data Quality Rules

Advanced use only!

- Used to retrieve data from Product 360 in IDQ processing
- REST API Accessed through Java Tx in IDQ rule
- Recommendations:
  - Test Java code used for data extraction outside of IDQ environment
  - Use data samples from external java tests for building and de-bugging IDQ rules

*If you need to use this functionality, please contact PM beforehand to evaluate alternate options / discuss existing development integrations*