# IFRS 16

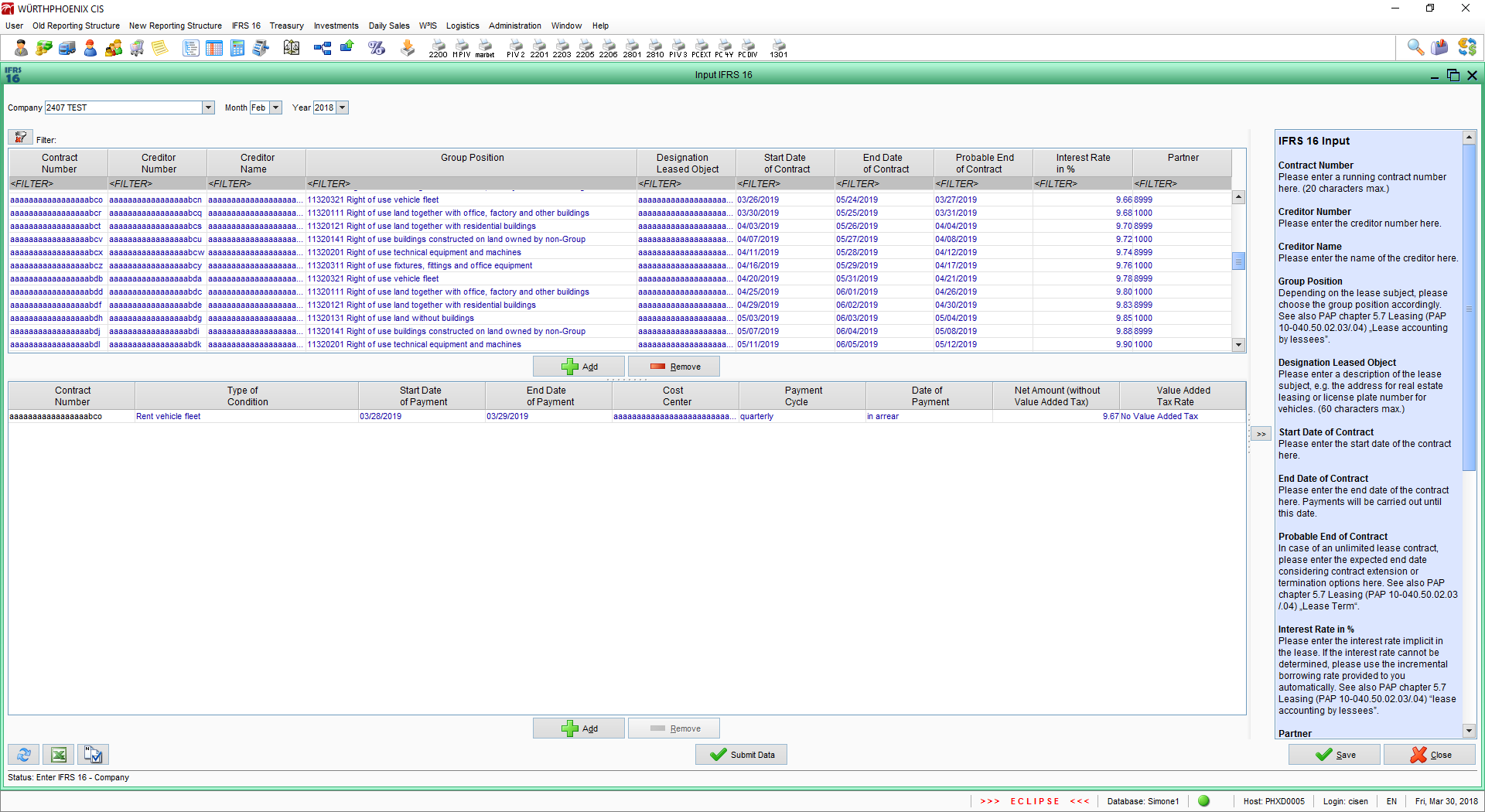
“IFRS 16 is an International Financial Reporting Standard (IFRS) promulgated by the International Accounting Standards Board (IASB) providing guidance on accounting for leases. IFRS 16 was issued in January 2016 and will be effective for most companies that report under IFRS in 2019” (en.wikipedia.org)

More information: <http://www.ifrs.org/issued-standards/list-of-standards/ifrs-16-leases/>

# CIS tasks

## Input Task

The user can edit (insert, update and delete) data about the contracts



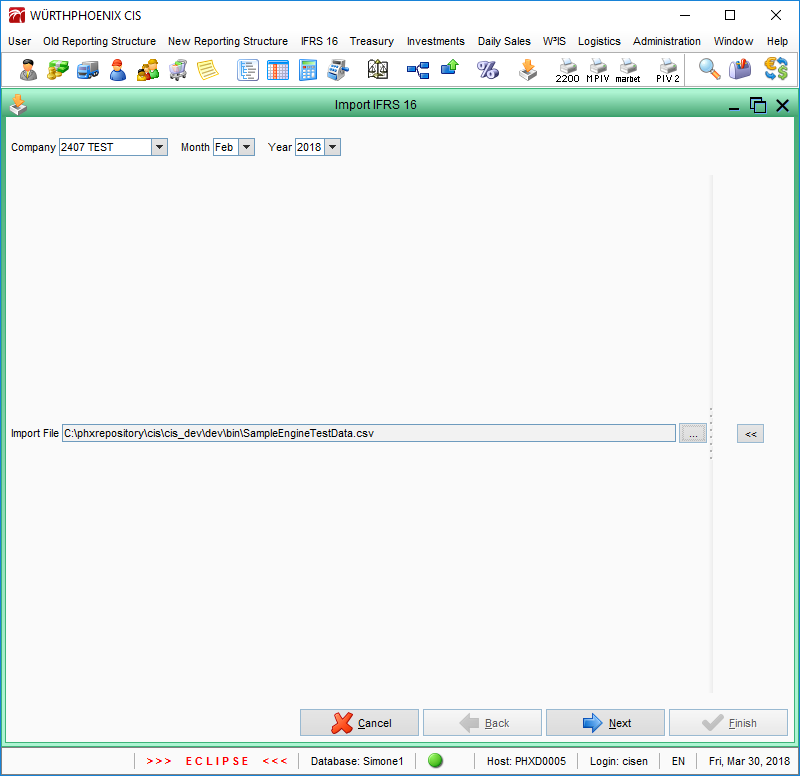
## Import Task

The user can import data into the system.

The engine of this task is the target of the test described in this document.

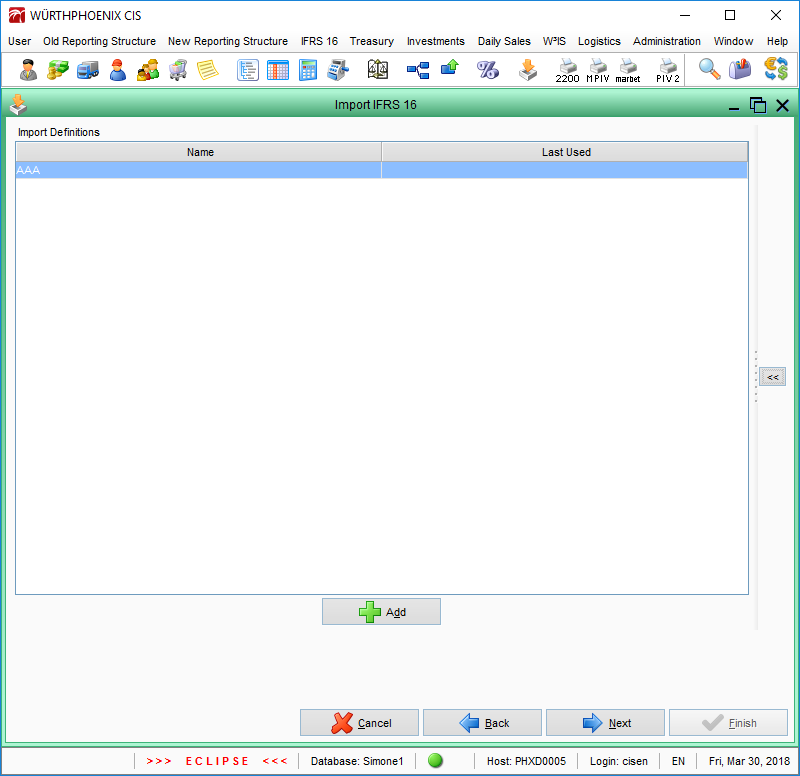
### File choose

The user select the target ImplementedCompany, the year, the month and the file to import



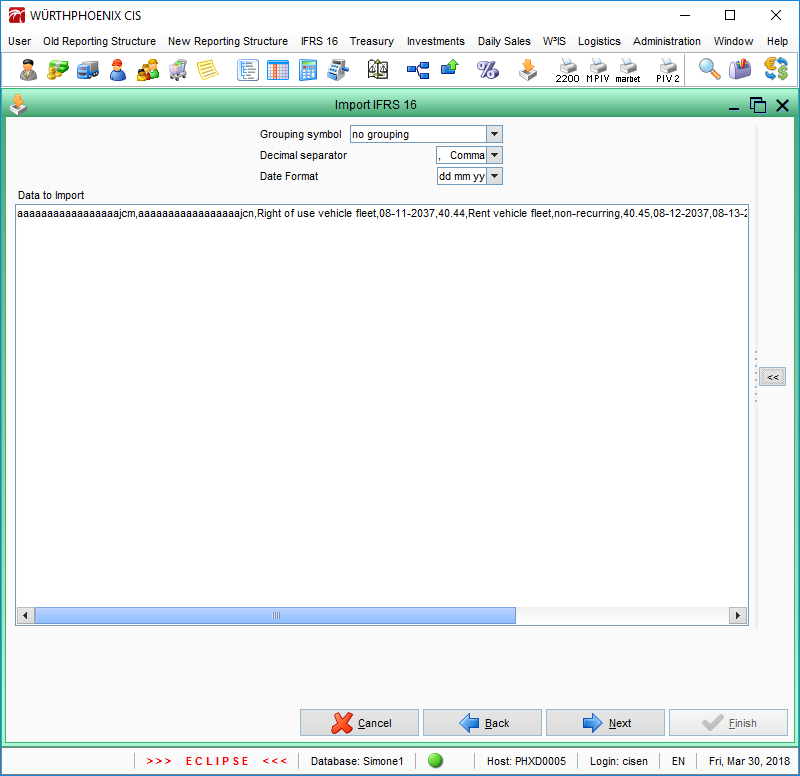
### Template choose

The user create a new template or select an existing one. A template contains all information about the import file format (number format, date format, column types, etc.)



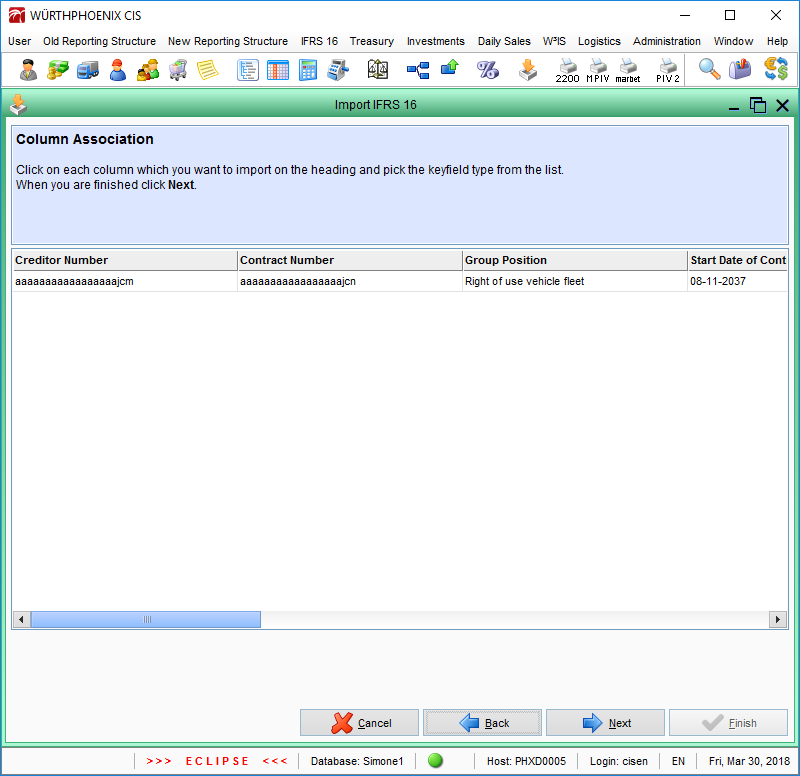
### Format

The user select the number format (thousand and decimal separator characters) and the date pattern



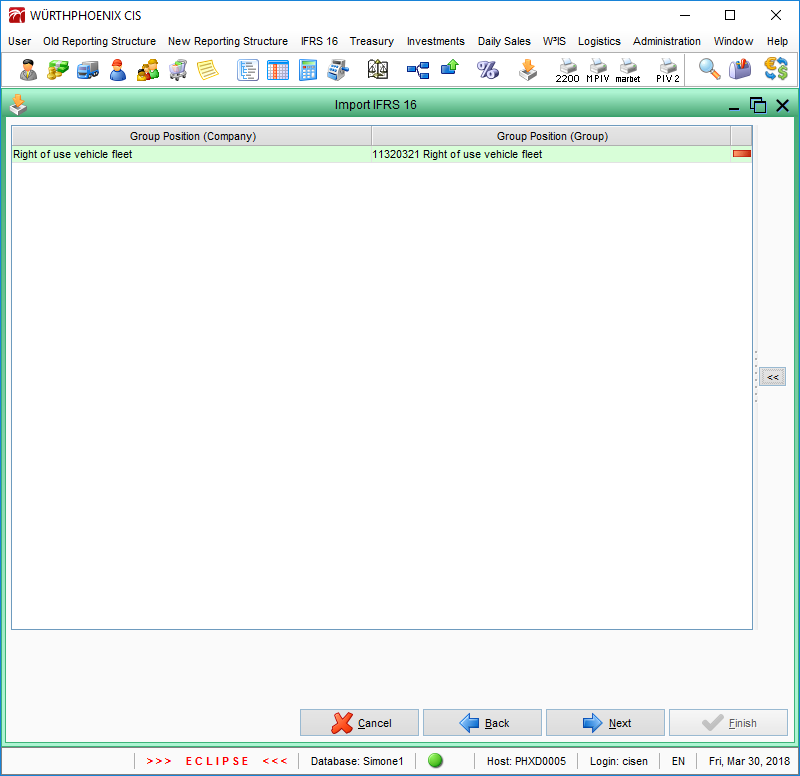
### Column association

The user associates the columns of the import file to a type (Creditor Name, Contract Number, Group Position, etc.)



### Row association

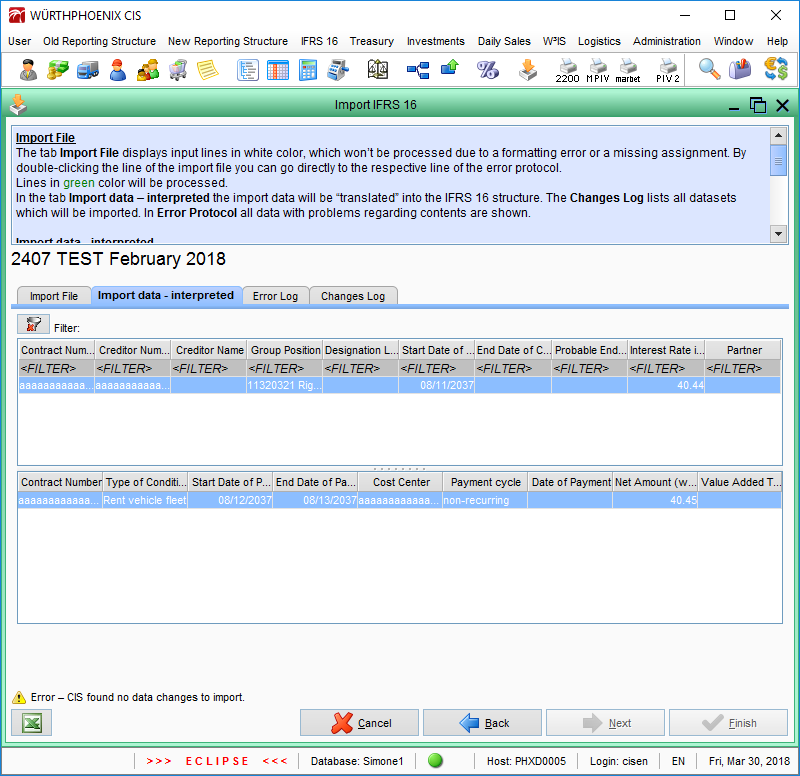
The user associates import file values to Corporate values (values shared by all Companies of the group)



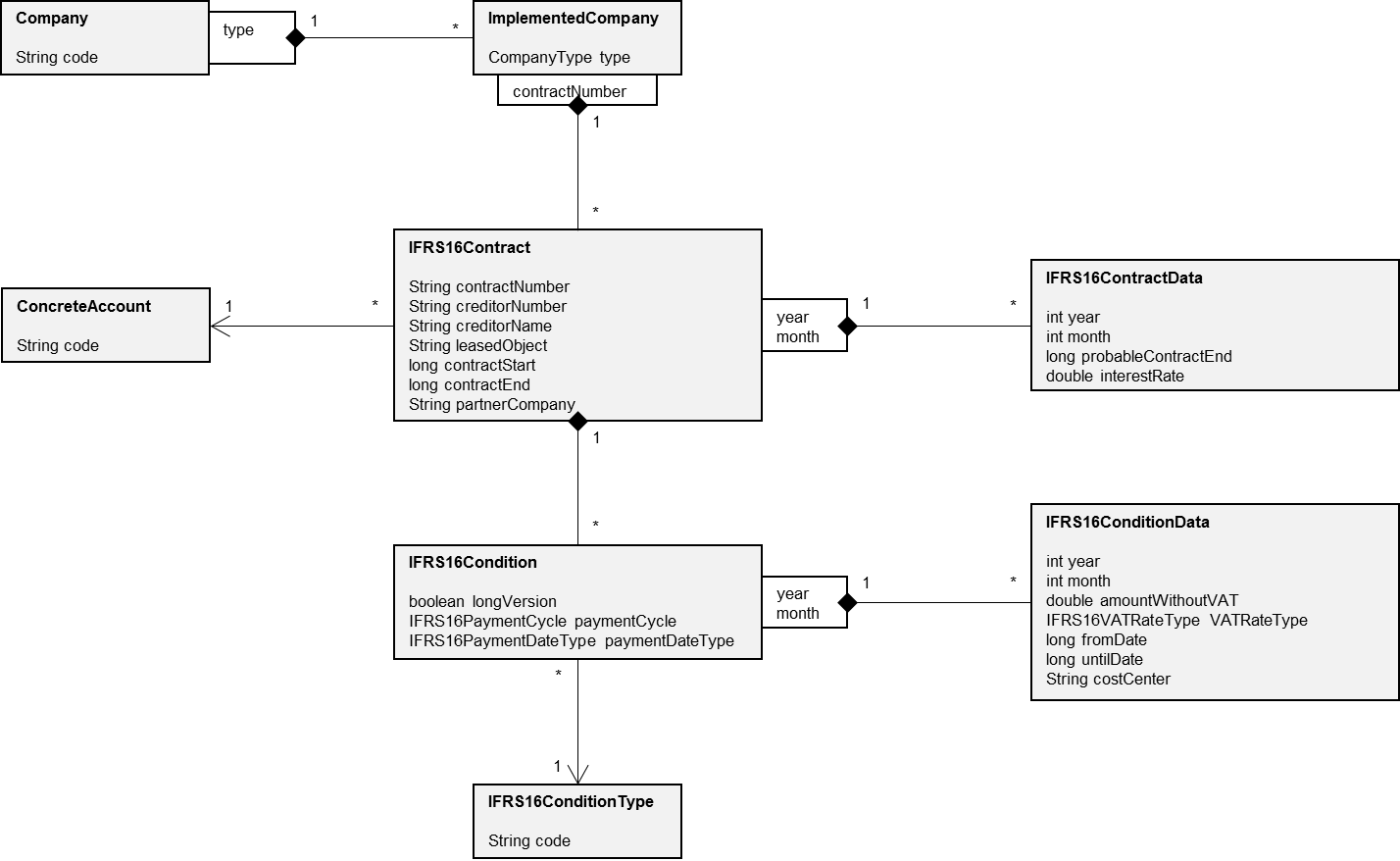
### Changes

The parsed values (interpretation of the import file), the errors (not valid values) and the merged values (merging of the interpreted valid values and the values existing in the DB) are shown to the user.

If possible, the user can import the data pushing the ‘Finish’ button



# Class diagram



## Description of the class diagram

### IFRS16Contract

All information about a Contract.

The class is aggregated to the class ImplementedCompany. More than one IFRS16Contract with the same ‘contractNumber’ cannot be aggregated to the same ImplementedCompany.

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| contractNumber | String, alphanumeric | Contract Number |
| creditorNumber | String, alphanumeric | Creditor Number |
| creditorName | String, alphanumeric | Creditor Name |
| leasedObject | String, alphanumeric | Designation Leased Object |
| contractStart | long, 0 or date in milliseconds | Start Date of Contract |
| contractEnd | long, 0 or date in milliseconds | End Date of Contract |
| partnerCompany | String, alphanumeric | Partner |

#### Associations

|  |  |
| --- | --- |
| **Name** | **Description** |
| ConcreteAccount | Group Position |

### IFRS16ContractData

All monthly information about a Contract.

The class is aggregated to the class IFRS16Contract. More than one IFRS16ContractData with the same ‘year’ and ‘month’ cannot be aggregated to the same IFRS16Contract.

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| year | int | Year |
| month | int, [1-12] | Month |
| probableContractEnd | long, 0 or date in milliseconds | Probable End of Contract |
| interestRate | double, 0 or positive, 2 decimals | Interest Rate in % |

### IFRS16Condition

All information about a Condition.

The class is aggregated to the class IFRS16Contract. More than one IFRS16Condition with the same ‘Condition Type’, ‘IFRS16ConditionData From Date’ and ‘IFRS16ConditionData Cost Center’ cannot be aggregated to the same IFRS16Contract.

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| longVersion | boolean, true (Long Version) or false (Short Version) | Version of the Condition |
| paymentCycle | IFRS16PaymentCycle, enumerator, not null | Payment cycle |
| paymentDateType | IFRS16PaymentDateType, enumerator, not null | Date of Payment |

#### Associations

|  |  |
| --- | --- |
| **Name** | **Description** |
| IFRS16ConditionType | Type of Condition |

### IFRS16ConditionData

All monthly information about a Condition Data.

The class is aggregated to the class IFRS16Condition. More than one IFRS16ConditionData with the same ‘year’ and ‘month’ cannot be aggregated to the same IFRS16Condition.

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| year | int | Year |
| month | int, [1-12] | Month |
| amountWithoutVAT | double, 0 or positive, 2 decimals | Net Amount (Without Value Added Tax) |
| VATRateType | IFRS16VATRateType, enumerator, not null | Value Added Tax Rate |
| fromDate | long, 0 or date in milliseconds | Start Date of Payment |
| untilDate | long, 0 or date in milliseconds | End Date of Payment |
| costCenter | String, alphanumeric | Cost Center |

### Company

All information about a Company (legal entity of the Würth Group).

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| code | String, alphanumeric, not empty | Unique code of the Company |

### ImplementedCompany

Container of the Company data. Data can be ‘Productive’ (not validated) or ‘Consolidated’ (accepted by the Group Controlling).

The class is aggregated to the class Company. More than one ImplementedCompany with the same ‘type’ cannot be aggregated to the same Company.

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| type | CompanyType, enumerator, not null | Type of the data (Productive or Consolidated) |

### ConcreteAccount

Balance account (shared by all Companies).

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| code | String, alphanumeric, not empty | Unique code of the Account |

#### Items

|  |  |
| --- | --- |
| **Code** | **Name** |
| NDS\_BS11320111 | Right of use land together with office, factory and other buildings |
| NDS\_BS11320121 | Right of use land together with residential buildings |
| NDS\_BS11320131 | Right of use land without buildings |
| NDS\_BS11320141 | Right of use buildings constructed on land owned by non-Group |
| NDS\_BS11320201 | Right of use technical equipment and machines |
| NDS\_BS11320311 | Right of use fixtures, fittings and office equipment |
| NDS\_BS11320321 | Right of use vehicle fleet |

### IFRS16ConditionType

Type of the Condition (shared by all Companies).

#### Attributes

|  |  |  |
| --- | --- | --- |
| **Name** | **Value** | **Description** |
| code | String, alphanumeric, not empty | Unique code of the Condition Type |

#### Items

|  |  |
| --- | --- |
| **Code** | **Name** |
| IFRS16ConditionType.01 | Rent I |
| IFRS16ConditionType.02 | Grant for remodeling costs |
| IFRS16ConditionType.03 | Utilities |
| IFRS16ConditionType.04 | Power |
| IFRS16ConditionType.05 | Water |
| IFRS16ConditionType.06 | Heating |
| IFRS16ConditionType.07 | Property tax |
| IFRS16ConditionType.08 | Rent parking area |
| IFRS16ConditionType.09 | Rent II |
| IFRS16ConditionType.10 | Rent III |
| IFRS16ConditionType.11 | Rent technical equipment and machines |
| IFRS16ConditionType.12 | Rent fixtures, fittings and office equipment |
| IFRS16ConditionType.13 | Rent vehicle fleet |
| IFRS16ConditionType.14 | Service fee technical equipment and machines |
| IFRS16ConditionType.15 | Service fee fixtures, fittings and office equipment |
| IFRS16ConditionType.16 | Service fee vehicle fleet |

### IFRS16PaymentCycle

Enumerator: Payment cycle of a condition (shared by all Companies).

#### Items

|  |  |
| --- | --- |
| **Value** | **Name** |
| NULL | - |
| NONRECURRING | non-recurring |
| MONTHLY | monthly |
| QUARTERLY | quarterly |
| SEMIANNUALLY | semiannually |
| ANNUALLY | annually |

### IFRS16PaymentDateType

Enumerator: Date of payment of a condition (shared by all Companies).

#### Items

|  |  |
| --- | --- |
| **Value** | **Name** |
| NULL | - |
| INADVANCE | in advance |
| MIDTERM | midterm |
| INARREAR | in arrear |

### IFRS16VATRateType

Enumerator: Value Added Tax rate (shared by all Companies).

#### Items

|  |  |
| --- | --- |
| **Value** | **Name** |
| NULL | - |
| FULLVAT | Full Value Added Tax |
| REDUCEDVAT | Reduced Value Added Tax |
| NOVAT | No Value Added Tax |

## Data

### Date

Date are stored in the DB as long (time in milliseconds). Year, Month and Day are considered, the other fields (Hour, Minute, Second and Millisecond) are reset (set as 0)

# Import File

Import File can be written as XLS (Excel) or CSV (Comma Separated Value).

An Import File is used to insert/update the data of the IFRS16Contract, IFRS16ContractData, IFRS16Condition and IFRS16ConditionData.

All the data of the Import File refer to a specified month. Data of different months cannot be specified into a single Import File.

A company can accept Import File in ‘Long Version’ or ‘Short Version’ (depends on a property of the Company).

A ‘Long Version’ Import File can include all columns. A ‘Short Version’ Import File can include only the columns marked with an ‘X’

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Class** | **Atribute/Association** | **Long** | **Short** |
| Contract Number | IFRS16Contract | contractNumber | X | X |
| Creditor Number | IFRS16Contract | creditorNumber | X |  |
| Creditor Name | IFRS16Contract | creditorName | X |  |
| Group Position | IFRS16Contract | ConcreteAccount | X | X |
| Designation Leased Object | IFRS16Contract | leasedObject | X | X |
| Start Date of Contract | IFRS16Contract | contractStart | X | X |
| End Date of Contract | IFRS16Contract | contractEnd | X | X |
| Probable End of Contract | IFRS16ContractData | probableContractEnd | X |  |
| Interest Rate | IFRS16ContractData | interestRate | X | X |
| Partner Company | IFRS16Contract | partnerCompany | X | X |
| Condition Type | IFRS16Condition | IFRS16ConditionType | X |  |
| From Date | IFRS16ConditionData | fromDate | X | X |
| Until Date | IFRS16ConditionData | untilDate | X | X |
| Cost Center | IFRS16ConditionData | costCenter | X | X |
| Payment Cycle | IFRS16Condition | paymentCycle | X |  |
| Payment Date Type | IFRS16Condition | paymentDateType | X |  |
| Amount Without Value Added Tax | IFRS16ConditionData | amountWithoutVAT | X | X |
| VAT Rate Type | IFRS16ConditionData | VATRateType | X |  |

## Column validation

The default mandatory column is ‘Contract Number’.

A column type can be specified only 0 or 1 time.

### Long version

‘Condition Type’, ‘From Date’ and ‘Cost Center’ are the logical key of a ‘Condition’. If the Import File contains one or more of the following columns, the following columns are mandatory:

* Condition Type
* From Date
* Cost Center

If the Import File contains one of the following columns

* From Date
* Until Date
* Cost Center
* Amount Without Value Added Tax
* VAT Rate Type

the following columns are mandatory

* Condition Type
* From Date
* Cost Center

The ‘Condition Type’ depends on the ‘Group Position’. If the Import File contains the column ‘Condition Type’, the column ‘Group Position’ is mandatory

### Short version

‘From Date’ and ‘Cost Center’ are the logical key of a ‘Condition’. If the Import File contains one or more of the following columns, the following columns are mandatory:

* From Date
* Cost Center

If the Import File contains one of the following columns

* From Date
* Until Date
* Cost Center
* Amount Without Value Added Tax

the following columns are mandatory

* From Date
* Cost Center

## Data validation

This paragraph describes the not importable data and the errors during the import

### Different attribute – Contract

In the Import File exists more than one line that refer to the same Contract (equal ‘Contract Number’). These lines have one or more different data for:

* Creditor Number
* Creditor Name
* Group Position
* Designation Leased Object
* Start Date of Contract
* End Date of Contract
* Partner Company

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Creditor Number** | **Creditor Name** |
| 1 | 0000001 | CN-001 | John Smith |
| 2 | 0000001 | CN-001 | John Smit |

### Different attribute – Contract Data

In the Import File exist more than one line that refer to the same Contract (equal ‘Contract Number’). These lines have one or more different data for:

* Probable End of Contract
* Interest Rate

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Probable End of Contract** | **Interest Rate** |
| 1 | 0000001 | 31-12-2018 | 1,25 |
| 2 | 0000001 | 31-12-2018 | 1,24 |

### Different attribute – Condition

In the Import File exist more than one line that refer to the same Contract (equal ‘Contract Number’) and Condition (equal ‘Condition Type’, ‘From Date’, ‘Cost Center’). These lines have one or more different data for:

* Payment Cycle
* Payment Date Type

Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Index** | **Contract Number** | **Condition Type** | **From Date** | **Cost Center** | **Payment Cycle** |
| 1 | 0000001 | 01 | 01-12-2018 | Auto | monthly |
| 2 | 0000001 | 01 | 01-12-2018 | Auto | annually |

### Different attribute – Condition Data

In the Import File exist more than one line that refer to the same Contract (equal ‘Contract Number’) and Condition (equal ‘Condition Type’, ‘From Date’, ‘Cost Center’). These lines have one or more different data for:

* Until Date
* Amount Without Value Added Tax
* VAT Rate Type

Example

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Index** | **Contract Number** | **Condition Type** | **From Date** | **Cost Center** | **Until Date** |
| 1 | 0000001 | 01 | 01-12-2018 | Auto | 31-12-2018 |
| 2 | 0000001 | 01 | 01-12-2018 | Auto | 30-12-2018 |

### No Group Position

In the Import File exist a line with a ‘Condition Type’ and without a ‘Group Position’

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Condition Type** | **Group Position** |
| 1 | 0000001 | 01 | 11320111 |
| 2 | 0000002 | 01 |  |

### Condition Type not available

In the Import File exist a line with a ‘Group Position’ and a ‘Condition Type’.

The following table shows the map of the valid relations. The prefix ‘NDS\_BS’ is omitted from the ‘ConcreteAccount.code’ and the prefix ‘IFRS16ConditionType.’ is omitted from the ‘IFRS16ConditionType.code’.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **11320111**  **11320121**  **11320131**  **11320141** | **11320201** | **11320311** | **11320321** |
| **01** | **Rent I** | X |  |  |  |
| **02** | **Grant for remodeling costs** | X |  |  |  |
| **03** | **Utilities** | X |  |  |  |
| **04** | **Power** | X |  |  |  |
| **05** | **Water** | X |  |  |  |
| **06** | **Heating** | X |  |  |  |
| **07** | **Property tax** | X |  |  |  |
| **08** | **Rent parking area** | X |  |  |  |
| **09** | **Rent II** | X |  |  |  |
| **10** | **Rent III** | X |  |  |  |
| **11** | **Rent technical equipment and machines** |  | X |  |  |
| **12** | **Rent fixtures, fittings and office equipment** |  |  | X |  |
| **13** | **Rent vehicle fleet** |  |  |  | X |
| **14** | **Service fee technical equipment and machines** |  | X |  |  |
| **15** | **Service fee fixtures, fittings and office equipment** |  |  | X |  |
| **16** | **Service fee vehicle fleet** |  |  |  | X |

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Condition Type** | **Group Position** |
| 1 | 0000001 | 01 | 11320111 |
| 2 | 0000002 | 01 | 11320201 |

### Empty

In the Import File exist a line without a ‘Contract Number’

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Creditor Number** | **Creditor Name** |
| 1 | 0000001 | CN-001 | John Smith |
| 2 |  | CN-002 | John Johnson |

### Exceeding text

In the Import File exist a line with one or more of the following columns that exceed the maximum number of characters

|  |  |
| --- | --- |
| **Column** | **Maximum number of characters (less or equal)** |
| Contract Number | 20 |
| Creditor Number | 20 |
| Creditor Name | 100 |
| Designation Leased Object | 250 |
| Cost Center | 50 |

Example

|  |  |  |
| --- | --- | --- |
| **Index** | **Contract Number** | **Creditor Number** |
| 1 | 012345678901234567890 | 01234567890123456789 |
| 2 | 01234567890123456789 | 012345678901234567890 |

### Unparsable

The data in the Import File cannot be used for the associated column.

The following table show the allowed value types

|  |  |
| --- | --- |
| **Column** | **Allowed value types** |
| Contract Number | Text |
| Creditor Number | Text |
| Creditor Name | Text |
| Group Position | Text |
| Designation Leased Object | Text |
| Start Date of Contract | Date object (Excel) or a ‘date’ formatted text |
| End Date of Contract | Date object (Excel) or a ‘date’ formatted text |
| Probable End of Contract | Date object (Excel) or a ‘date’ formatted text |
| Interest Rate | Number object (Excel) or a ‘number’ formatted text |
| Partner Company | Text |
| Condition Type | Text |
| From Date | Date object (Excel) or a ‘date’ formatted text |
| Until Date | Date object (Excel) or a ‘date’ formatted text |
| Cost Center | Text |
| Payment Cycle | Text |
| Payment Date Type | Text |
| Amount Without Value Added Tax | Number object (Excel) or a ‘number’ formatted text |
| VAT Rate Type | Text |

Example

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **Start Date of Contract** | **Interest Rate** |
| 1 | 0000001 | Abcd | 0,25 |
| 2 | 0000002 | 01-12-2018 | Abcd |

#### Date formatted text

Date can be written in ‘us’ mode (mm-dd-yyyy) or ‘de’ mode (dd-mm-yyyy).

The date format has to be the same for all data of the Import File.

#### Number formatted text

Numbers can be written with a ‘Digit Grouping Symbol’ (‘Point’, ‘Comma’, ‘No Break Space’, ‘Single Quotation Mark’, ‘Normal Space’) and a ‘Decimal Separator’ (‘Point’, ‘Comma’).

‘Digit Grouping Symbol’ and ‘Decimal Separator’ are not mandatory. If specified, cannot be equal.

The number format has to be the same for all data of the Import File

### Historical

The Import File contains a line that refers to an ‘historical’ Contract.

Historical Contract: the Contract is stored in the DB and the stored date for ‘End Date of Contract’ is before the date selected in the Import Wizard

Example

Contracts stored in the DB

|  |  |
| --- | --- |
| **Contract Number** | **End Date of Contract** |
| 0000001 | 31-10-2018 |
| 0000002 | 30-11-2018 |
| 0000003 | 31-12-2018 |

Import File (Import Wizard date: November 2018)

|  |  |  |  |
| --- | --- | --- | --- |
| **Index** | **Contract Number** | **End Date of Contract** |  |
| 1 | 0000001 | 31-12-2018 | *31-10-2018 is before 01-11-2018* |
| 2 | 0000002 | 31-12-2018 | *30-11-2018 is not before 01-11-2018* |
| 3 | 0000003 | 31-12-2018 | *31-12-2018 is not before 01-11-2018* |

### Not editable

The Import File contains a line that refers to an existing (stored in the DB) ‘not historical’ Contract. The Import File also contains data for:

* Group Position
* Start Date of Contract
* Partner Company
* Payment Cycle
* Payment Date Type

## Import blockers

Import files with ‘Column validation’ errors cannot be imported.

The following table shows the behavior of the ‘Data validation’ errors

|  |  |  |  |
| --- | --- | --- | --- |
| **Error** | **Column** | **Mode** | **Description** |
| Different attribute – Contract |  | Blocker | All data of the line are not imported |
| Different attribute – Contract Data |  | Blocker | The data referring to the Contract Data are not imported |
| Different attribute – Condition |  | Blocker | The data referring to the Condition and Condition Data are not imported |
| Different attribute – Condition Data |  | Blocker | The data referring to the Condition and Condition Data are not imported |
| No Group Position |  | Blocker | All data of the line are not imported |
| Condition Type not available |  | Blocker | All data of the line are not imported |
| Empty | Contract Number | Blocker | All data of the line are not imported |
| Empty | Others | Warning | Empty data are imported |
| Exceeding text | Contract Number | Blocker | All data of the line are not imported |
| Exceeding text | Others | Warning | Exceeding texts are reduced and imported |
| Unparsable |  | Warning | Unparsable data are not imported |
| Historical |  | Blocker | All data of the line are not imported |
| Not editable |  | Warning | Not editable data are not imported |

# Source Code

## com.wuerth.phoenix.cis.university.example2.adapters

**--- NO CHANGES ARE ALLOWED IN THIS PACKAGE ---**

The package contains the replica of the persistent classes of the CIS

### Persistent objects

The following classes replicated a subset of the persistent classes of the CIS (paragraph Class diagram). Each class contains the relevant attributes (columns) of the related table of the DB.

The classes can be used for:

* Define the initial data (data stored in the DB before the import check)
* Combination check (transformation of a Combination in a list of IFRS16Contract)

|  |  |
| --- | --- |
| **Class name** | **Description** |
| APersistentObject | Abstract class. Each class of this block extends this class |
| Company | Replica of the Company (paragraph Company) |
| ImplementedCompany | Replica of the ImplementedCompany class (paragraph ImplementedCompany) |
| IFRS16Contract | Replica of the IFRS16Contract class (paragraph IFRS16Contract) |
| IFRS16ContractData | Replica of the IFRS16ContractData class (paragraph IFRS16ContractData) |
| IFRS16Condition | Replica of the IFRS16Condition class (paragraph IFRS16Condition) |
| IFRS16ConditionData | Replica of the IFRS16ConditionData class (paragraph IFRS16ConditionData) |
| ConcreteAccount | Replica of the ConcreteAccount class (paragraph ConcreteAccount) |
| IFRS16ConditionType | Replica of the IFRS16ConditionType class (paragraph IFRS16ConditionType) |

### Enumerators

The following classes replicated a subset of the enumerators of the CIS.

An enumerator is a class that contains:

* A constants (short) for each item of the enumerator
* A constants class instance for each constants (short)
* A method (public static List<AEnumerator> getAllValues()) that returns all constants class instances

An enumerator is stored in the DB as ‘short’ (value of the constants class instance)

|  |  |  |
| --- | --- | --- |
| **Class name** | **Description** | **Used for** |
| AEnumerator | Abstract class. Each class of this block extends this class |  |
| CompanyType | Replica of the CompanyType enumerator | ImplementedCompany.type  Items:   * PRODUCTIVE (data aggregated to this type of ImplementedCompany are not validated by the group controlling) * CONSOLIDATED (data aggregated to this type of ImplementedCompany are validated by the group controlling) * TEMPORARY (obsolete item, do not use) |
| IFRS16PaymentCycle | Replica of the IFRS16PaymentCycle enumerator (paragraph IFRS16PaymentCycle) | IFRS16Condition.paymentCycle |
| IFRS16PaymentDateType | Replica of the IFRS16PaymentDateType enumerator (paragraph IFRS16PaymentDateType) | IFRS16Condition.paymentDateType |
| IFRS16VATRateType | Replica of the IFRS16VATRateType enumerator (paragraph IFRS16VATRateType) | IFRS16ConditionData.VATRateType |

|  |  |  |
| --- | --- | --- |
| IFRS16ImportAssignmentType | Replica of the IFRS16ImportAssignmentType enumerator | Defines:   * the type of a CombinationItem * the type of a column of an import file   It exists an item for each attribute of the following classes:   * IFRS16Contract * IFRS16ContractData * IFRS16Condition * IFRS16ConditionData |
| DigitGroupingSymbol | Replica of the DigitGroupingSymbol enumerator | Defines:   * the thousand separator of a formatted number * the decimal separator of a formatted number   Items:   * NULL (no separator) * POINT (.) * COMMA (,) * NO BREAK SPACE (char 160, a space character that prevents an automatic line break at its position) * SINGLE QUOTATION MARK (‘) * NORMAL SPACE ( )   Only POINT and COMMA can be used as decimal separator |

### Util

|  |  |
| --- | --- |
| **Class name** | **Description** |
| Key | This class is used to define an id for an aggregation between two persistent classes |

## com.wuerth.phoenix.cis.university.example2.util

**--- NO CHANGES ARE ALLOWED IN THIS PACKAGE ---**

The package contains the classes used for the Combination generation, parse and check

### Combination

|  |  |
| --- | --- |
| **Class name** | **Description** |
| CombinationItem | A single item, like a cell, of an import file. An instance of this class contains the value of the item and its text format  Type of values:   * String * Long * Double * ConcreteAccount * IFRS16ConditionType * IFRS16PaymentCycle * IFRS16PaymentDateType * IFRS16VATRateType   The class provides static methods to create an instance for each type |
| CombinationLine | A list of items, like a row, of an import file.  One or more CombinationItem can be added to this class.  The items are identified by type (IFRS16ImportAssignmentType). Duplicated types are not allowed |
| Combination | A list of lines, like an import file  One or more CombinationLine can be added to this class.  The class FileGenerator transforms this class in a CSV import file |
| Settings | This class contains the parameters for the import functionality   * locale: the login language of the user (Locale.GERMAN or Locale.ENGLISH) * implementedCompany: the target ImplementedCompany of the import * longVersion: the type of the import * year, month: the month selected for the import (attributes of the classes IFRS16ContractData and IFRS16ConditionData) * filePath: the target file folder and name of the import file * name: the name of the import configuration * digitGroupingSymbol, decimalSeparator: the formatting chars of a number * dateFormat: the pattern of a formatted date (Constants.SETTTING\_DATE\_DE or Constants.SETTTING\_DATE\_EN) * typeList: the list of the column types of the import file |
| Scenario | This class contains the data for an import check   * combination: import file creation and data check during the import functionality * settings: parameters for the import functionality * companyList: data to create on the DB before the import (paragraph Class diagram) |

### Checker

|  |  |
| --- | --- |
| **Class name** | **Description** |
| FileGenerator | This class provides to transform a Combination in an import file using the parameters specified in the Settings class |
| CombinationParser | This class provides to   * create IFRS16Contract data structure from a CombinationLine * manage the parsed data during an import check (on parse, on merge, on import) |
| Checker | This class provides to parse and manage the data during the import checks (on parse, on merge, on import) |

### Util

|  |  |
| --- | --- |
| **Class name** | **Description** |
| Constants | This class contains constants data needed for the test |
| Util | This class contains util methods needed for the test |

## com.wuerth.phoenix.cis.university.example2.test

**--- NO CHANGES ARE ALLOWED IN THIS PACKAGE ---**

This package contains the interface (IEngineTestData) used for the import test.

All tests have to implements this interface and declared as ’abstract’.

The method ‘public boolean check(Scenario scenario)’ has not to be implemented. The method has to be called from the class for each generated Scenario

### public boolean start()

The method is called on test start. In this place have to be generated all Scenario. For each generated Scenario has to be called the method ‘public boolean check(Scenario scenario)’.

### public boolean check(Scenario scenario)

The method is implemented by the import tester (not provided) and checks the generated Scenario

## com.wuerth.phoenix.cis.university.example2.test.sample

**--- NO CHANGES ARE ALLOWED IN THIS PACKAGE ---**

This package contains a sample import test (SampleEngineTestData).

## com.wuerth.phoenix.cis.university.example2.test.X

A package named ‘com.wuerth.phoenix.cis.university.example2.test.X’ has to be created for each work group (replace the X with the name of the group). The package has to contain an implementation of the class ‘IEngineTestData’ (like the provided ‘SampleEngineTestData’)

# Test Case

