XBRL US Data Quality Committee

Campbell Pryde

Overview of DQC

- Develop freely available guidance and validation rules for public companies to prevent or detect inconsistencies or errors in XBRL submissions to the SEC.
- Proposed guidance and rules are subject to public comment before final publication.
- The Committee holds periodic meetings to update the SEC staff on its progress.
- Supports US-GAAP and IFRS submissions
- Works with FASB and IFRS to identify rules for taxonomy updates and new taxonomies

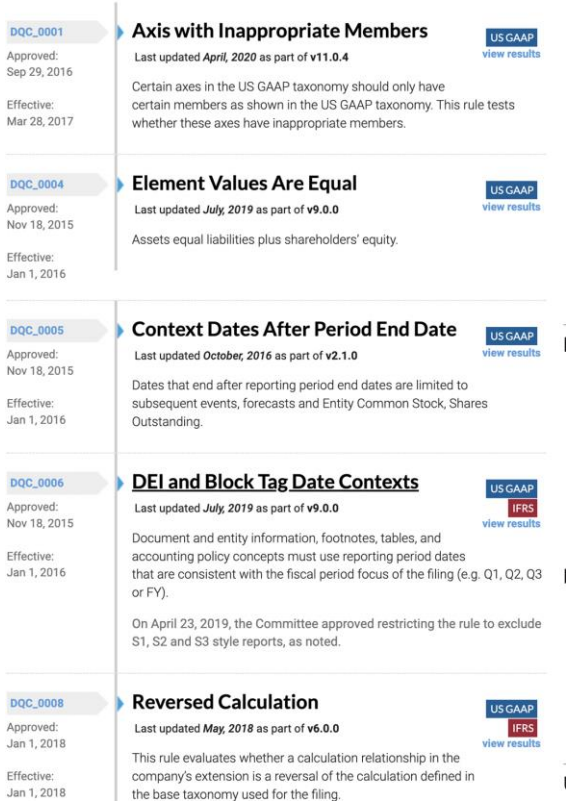
Work with the SEC

- Meet with the SEC twice a year to discuss issues in filings
- Work with SEC to incorporate US-GAAP rules into Edgar validation routines
- Edgar validation does not include DQC rules for IFRS filings.
- Make findings available to the SEC.

Rule Manifest

- 138 US GAAP DQC Rules over 26 version releases.
  - Each rule can contain multiple components
  - I.e. there is one non negative rule which checks 1,000's of elements
- 32 IFRS DQC Rules over 26 version releases.
- IFRS and US GAAP are separate rule sets
- There are separate rules for each version of the taxonomy
- Listing of Rules

View Published Rules



DQC_0001
Approved: Sep 29, 2016
Effective: Mar 28, 2017

Axis with Inappropriate Members

US GAAP
view results

Last updated *April, 2020* as part of **v11.0.4**

Certain axes in the US GAAP taxonomy should only have certain members as shown in the US GAAP taxonomy. This rule tests whether these axes have inappropriate members.

DQC_0004
Approved: Nov 18, 2015
Effective: Jan 1, 2016

Element Values Are Equal

US GAAP
view results

Last updated *July, 2019* as part of **v9.0.0**

Assets equal liabilities plus shareholders' equity.

DQC_0005
Approved: Nov 18, 2015
Effective: Jan 1, 2016

Context Dates After Period End Date

US GAAP
view results

Last updated *October, 2016* as part of **v2.1.0**

Dates that end after reporting period end dates are limited to subsequent events, forecasts and Entity Common Stock, Shares Outstanding.

DQC_0006
Approved: Nov 18, 2015
Effective: Jan 1, 2016

DEI and Block Tag Date Contexts

US GAAP
IFRS
view results

Last updated *July, 2019* as part of **v9.0.0**

Document and entity information, footnotes, tables, and accounting policy concepts must use reporting period dates that are consistent with the fiscal period focus of the filing (e.g. Q1, Q2, Q3 or FY).

On April 23, 2019, the Committee approved restricting the rule to exclude S1, S2 and S3 style reports, as noted.

DQC_0008
Approved: Jan 1, 2018
Effective: Jan 1, 2018

Reversed Calculation

US GAAP
IFRS
view results

Last updated *May, 2018* as part of **v6.0.0**

This rule evaluates whether a calculation relationship in the company's extension is a reversal of the calculation defined in the base taxonomy used for the filing.

The published rules are available at https://xbrl.us/data-quality/rules-guidance/

This lists all the published rules and an explanation of each rule.

The code of each rule is published on github.

xBRL US

General Requirements

Every filing has its own taxonomy so company taxonomy inspection is required to run many rules.

Rules will run differently depending on the company extension taxonomy.

Rules should eliminate false positives.

Rules should handle exceptions that apply

- This means the rules are not always straight forward
- Usually 70% of the rule is dealing with the exception to the rule.
- This often means inspecting the taxonomy to determine what has been done, or looking up values that indicate an exception.

General Requirements

Rules check the values reported in the instance

Rules check the structure of the DTS

Rules compare the structure of the filer provided DTS against the base taxonomy.

Existing formula lacks the functions to effectively test the validity of the DTS.

Rule Development Process

Identify errors through the following process:

- Feedback from data users
- Feedback from FASB and IFRS teams
- Normalization Analysis
- Review of disclosures
- Creation of rules

Assess the impact of the errors across the filing base

Design a rule to capture the error condition.

Test the rule

Run the rule over a large sample of filings and have filing agents review the results.

Define test suite for each rule to ensure that it operates as expected through time

Rule Development Technology

Capture all filing data in a Postgres database.

Use SQL and API queries to quantify the extent of potential issues

Develop rules using the XULE syntax in a vscode editor

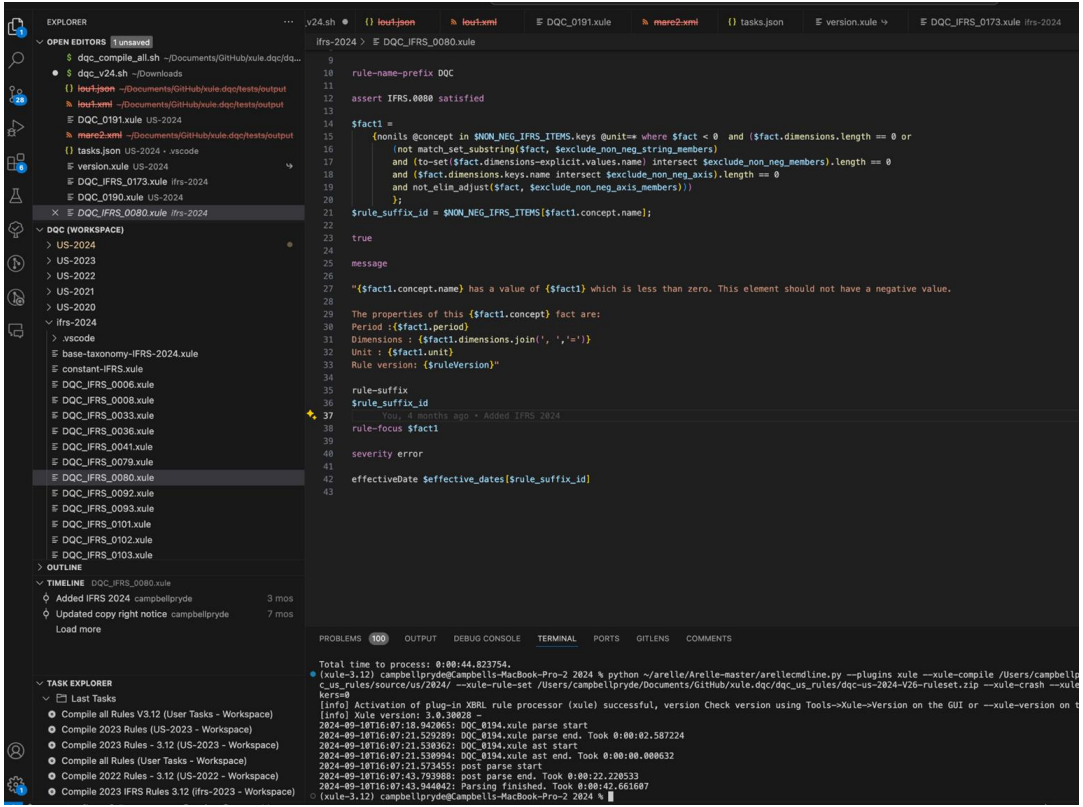
All rule are managed and published on github

Run filings against the rules using a XULE processor.

Database filing rule results in the postgres database for review

All software used is open source and freely available

Vscode Development Environment



Uses a vscode plugin to highlight, autocomplete and syntax check the rules as they are written.

Why do we not use XBRL Formula?

- Lacks the ability to navigate the company taxonomy independently of the instance document.
- Cannot create rules that inspect the base taxonomy
  - Many rules operate by comparing the filing DTS to the base taxonomy
- Difficult to control alignment across multiple dimensions
- The XULE model does not reference an XML format and allows rules to be written at an object level.
  - This means XULE expressions will work for XML, xhtml, json and csv submissions.
- At the time of original deployment the EF syntax for formula did not exist
  - This meant formula would have need to be maintained in an XML format.
- Working to make XULE an XBRL specification. XBRL International has published an XBRL query specification for public review that incorporates the functionality of XULE.

Example Rules - Negative Items

Check for items that cannot be negative.

Complicated by fact that in certain circumstances values can be negative

This occurs when combined with:

- A specific dimension
- A specific member
- Members containing a specific string
- A combination of dimensions and members

The items to check and the exceptions are defined in the taxonomy

This is written as a single rule.

See DQC Rule 80

Example Rules - Negative Items

```
$fact1 = {nonils @concept in $NON_NEG_ITEMS.keys @unit = * where $fact < 0
            and
            ($fact.dimensions.length == 0
                  or (not match_set_substring($fact, $EXCLUDE_NON_NEG_STRING_MEMBERS)
                                                      and (to-set($fact.dimensions-explicit.values.name)
intersect $EXCLUDE_NON_NEG_MEMBERS).length == 0
                                                      and ($fact.dimensions.keys.name intersect
$EXCLUDE_NON_NEG_AXIS).length == 0
                                                      and not_elim_adjust($fact, $EXCLUDE_NON_NEG_AXIS_MEMBERS)
                  )
            )
            };
```

Negative Items - Reads Taxonomy

The following constant is defined by reading the relationships defined in the US-GAAP taxonomy.

This means the rule does not have to be updated when additional non neg axis are added to the taxonomy.

```
constant $EXCLUDE_NON_NEG_AXIS = navigate rule-concept descendants from
list(dqcrules15:Dqc_0015_ExcludeNonNegAxisAbstract) taxonomy $DQC_RULES
returns set (target-name)
```

Identify the DQC rules taxonomy

```
constant $DQC_RULES = taxonomy('https://xbrl.fasb.org/us-
gaap/2023/dqcrules/dqcrules-entire-2023.xsd')
```

Example Rules - Period Comparison

Check that where contiguous periods are reported and the sum of those periods are reported that the values match.

Take periods out of alignment and compare based on start and end dates.

Allows to check that Q1 + Q2 + Q3 + Q4 = YR or Q1 + Q2 + Q3 = 3QCUM

The use case is used to ensure that elements are selected consistently from one period to the next.

See DQC Rule 115

Compare Calculations in Filer DTS vs Base DTS

Compare calculation relationships to the bases DTS to ensure they are not reversed.

I.e. CurrentAssets = Assets + NoncurrentAssets

The CurrentAssets to Assets relationship should be flagged as the inverse of the base taxonomy.

See DQC Rule 8

Compare Calculations in Filer DTS vs Base DTS

Check that sibling elements in the base taxonomies are not promoted to parents in the calculation relationships.

I.e. CurrentAssets = Assets + NoncurrentAssets

In this case current assets has been defined as a parent of NoncurrentAssets in the filer taxonomy, but is a sibling in the base taxonomy.

Of course there are some exceptions to the rule which have to be excluded.

See DQC Rule 0081

Recalculate Calculations with Dimensional Breakdowns

For each aggregated amount the rule selects the child elements and retrieves the value from the instance document.

If the child value does not have a value in the default the rule checks if dimensional values are reported for the element and derives a value by aggregating the dimension.

The aggregation is only done for those dimensions that comprise the financial statement table. The rule works on the assumption that the dimensional breakdown of the amount is complete and does not represent a portion of the default value for the line item. If a value can be derived from adding two different dimensions, then the dimension which results in the highest value is used.

If the sum of the child elements does not equal the value reported for the aggregation then an error is reported.

The rule will only run for the period representing the required context. This ensures that no false positives are generated for periods that are only partially reported.

See DQC Rule 0127

Recalculate Calculations with Dimensional Breakdowns

Errors will typically occur for the following reasons:

Missing Calculation Child

In these cases the filer has excluded a calculation relationship in the filing. This will result in a calculation inconsistency. All components of a calculation have to be reported in the calculation linkbase.

Incorrect Addition

In some cases the components of a value are incorrect because one of the numbers comprising the calculation is entered incorrectly.

Dimensional Misalignment

A line item has calculation children but one of the children is the parent line with a dimensional component. Because the calculation components of the parent can not include the parent itself there is no child element defined to capture the value in the calculation tree.

Recalculate Calculations with Dimensional Breakdowns

Functionality Required	Syntax
Ability to identify Financial Statement Cubes	`FILTER taxonomy().cubes where $item.drs-role.description.contains('- Statement -')`
Ability to Identify concepts in the cube that use facts	`filter $cube.facts returns $item.concept.name;`
Ability to identify calculation children of a concept	`navigate summation-item children from $concept_item role $cube.drs-role.uri returns (target-name, weight);`
Ability to select only values present in the cube.	`{nonils @concept = $targetElement @cube.drs-role = $cube.drs-role}`
Ability to get dimension Sums	
Determine nesting on Axis	`$members = navigate dimensions descendants from $dim drs-role $cube.drs-role returns target-name`
Only members with values that are subtotals	`{nonils @concept = $concept @cube.drs-role = $cube.drs-role @$dim = $member }`