IBM Planning Analytics

Submission Requirements

Management and Validation

Solution Template

Last Updated:
January 27, 2020

Author:

Andreas Kugelmeier
Executive Consultant, FOPM
Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-215-384-7302
Email: kugelmeier@us.ibm.com

Document Version History

| Date	Version	Author	Description
2019	0.9	Andreas Kugelmeier	
01/27/2020	1.0	Andreas Kugelmeier	

Table of Contents

1   About this Document

This document describes the design and basic implementation of an IBM Planning Analytics financial data submission grid management, control & validation asset template.

Without a submission requirement management model, submission validations have to be conducted manually and via custom reports or forms. Those custom reports will undergo frequent changes and are error prone. The effort to Building the reports and maintaining them for just one closing cycle by far exceeds the effort to just enter the initial submission requirements grid into the model. Forms will have to be carefully maintained and build such that data is only entered where it should be entered.

The template described in this document provides functionality that allows to enter submission requirements ('who is to input what and where and what validation criteria are to be met and which data has to tie to what data') into a 'Submission Grid' or 'Submission Requirements Management' model. The solution then translates those business submission requirements into a matrix that can be interpreted by TM1/PA and hence utilized to

- Provide submission requirements guidance for users
- Block input for non-permissible intersections of a model (i.e. prevent users from inputting data where data shall not be entered for a particular submission)
- Allow for submission requirements to be applied downstream and dynamically, while keeping an archive (history) of prior submission requirements.
- Provide automated validation (or aide in validation), by analysing required inputs against validation and tie-out requirements, all meta-data driven (based on data in the submission grid model.
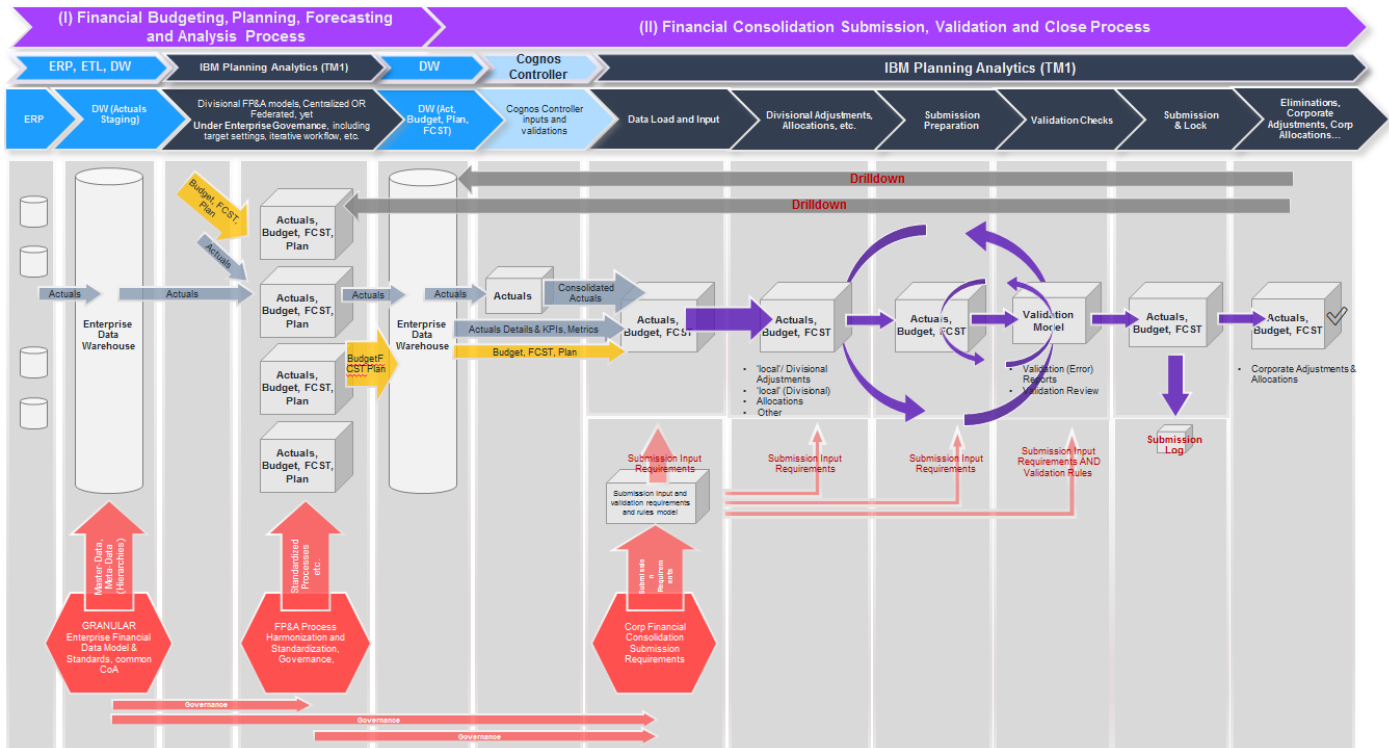
2 Submission Requirements Management and Lookup

2.1 Submission Requirements Model

2.1.1 Submission Requirements Cube

The 'Submission Requirements' model is used for entering and managing submission requirements and financial data validation rules at a high, aggregated level, by

➢ Company (or Business Unit, Legal Entity or similar),
➢ Time (YYYYMM),
➢ Version (Actual, Budget, Forecast, Plan, etc.)
➢ Data Source,
➢ Account, and
➢ Submission Model.
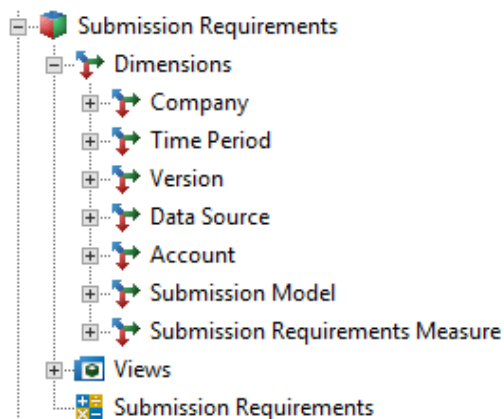
The Submission Requirements model and its measures is designed to allow for both rule and TI-process-based interpretation of submission requirements. Requirements can be 'queried' from the model by intersection of Company, Time, Version, Data Source, Account, and Submission Model. Multiple Tie-Out requirements are possible because multiple tie-out rules can be defined for every such intersection.

2.1.2 Submission Requirements Measures

- **Reference Company**: Reference Company for submission requirements. If <> none or <> '' all submission requirements for required accounts (see 'Is Required' below) are taken from this company. A reference company may be a leaf or a consolidation.
  Recommendation: use Consolidation reference companies for requirements that are applicable for all or most of its descendants.

- **Is Required**: set to Y if submission for the account is required for Company, Version, Data Source, Submission Model, Month.

- **Is Required (Numeric)**: calculated. 1 if submission is required for leaf-level accounts and data sources.

- **Is non-$ Account**: set to Y if the account is a non-currency account.

- **Is level-specific Validation Requirement:** Set this flag to yes if the submission requirement is not to be applied to other companies (i.e. if the submission requirement is level-specific). Example: a tie-out requirement at Total Company level such as Intercompany = 0 at total

company). It follows that if this flag is set to yes for a reference company, the companies that are configured to use this reference company will not receive the corresponding submission requirement.

- **--- Submission Thresholds and Restrictions ---**: Grouping
  - o **=0** (Y/N/empty): Value is to be = 0 (for balances or 'differences' for example)
  - o **>=0** (Y/N/empty): Value is to be >=0
  - o **<=0** (Y/N/empty)**:** Value is to be <=0
  - o **<=1** (Y/N/empty): Value is to be <=1
  - o **Threshold:** for =0 set to Y, will allow setting an absolute threshold. Example: if a balance is supposed to be = 0, yet a small variance shall be allowed, the absolute variance is to be entered here. Setting a threshold of 1000 for example on a balance sheet = 0 requirement will only lead to a violation or the variance between assets and liabilities & equity is equal or greater than 1000.

  Notes:
  - To require a value between 0 and 1, set both >=0 and <=1 to Y
  - In the Submission Validation Model, Submission Thresholds and Restrictions are evaluated against translated USD values only.

- **--- Tie-Out Requirements ---**: Grouping. The Tie-Out Requirements are leveraged by the <Submission Validation> model to automatically calculate tie-out violations (variances <> 0).

  - o **Tie-Out Submission A**: name of the data source that the required submission entry needs to tie out with. May be the same or a different submission. If left empty, the tie-out will occur against the same data source as the submission value.
  - o **Tie-Out Account A**: name of the account within Tie-Out Submission A that the submission entry needs to tie out with. May be the same of a different account.
  - o **Tie-Out Model A**: name of the submission model where the value for Tie-Out Account A can be found for Tie-Out Submission A.
  - o **Tie-Out to Account A Closing Balance**: if set to Y, the prior year closing balance is used for the Tie-Out Account Value
  - o **Tie-Out Account A Variance Threshold:** Allowable variance threshold (absolute) between submission value and tie-out value

  - o **Tie-Out Submission B**: name of the data source that the required submission entry needs to tie out with. May be the same or a different submission. If left empty, the tie-out will occur against the same data source as the submission value.
  - o **Tie-Out Account B**: name of the account within Tie-Out Submission A that the submission entry needs to tie out with. May be the same of a different account.
  - o **Tie-Out Model B**: name of the submission model where the value for Tie-Out Account B can be found for Tie-Out Submission B.
  - o **Tie-Out to Account B Closing Balance**: if set to Y, the prior year closing balance is used for the Tie-Out Account Value
  - o **Tie-Out Account B Variance Threshold:** Allowable variance threshold (absolute) between submission value and tie-out value

  - o **Tie-Out Submission C**: name of the data source that the required submission entry needs to tie out with. May be the same or a different submission. If left empty, the tie-out will occur against the same data source as the submission value.
  - o **Tie-Out Account C**: name of the account within Tie-Out Submission A that the submission entry needs to tie out with. May be the same of a different account.
  - o **Tie-Out Model C**: name of the submission model where the value for Tie-Out Account C can be found for Tie-Out Submission C.
  - o **Tie-Out to Account C Closing Balance**: if set to Y, the prior year closing balance is used for the Tie-Out Account Value

- o **Tie-Out Account C Variance Threshold:** Allowable variance threshold (absolute) between submission value and tie-out value

Note: In the Submission Validation Model, Tie-Out Account Variance Thresholds are evaluated against translated USD values only.

- **--- Submission Model Granularity ---**: Grouping. The Granularity section allows setting filters/restrictions on where (into which dimensions) the data may be entered for the dimensions that are applicable to the submission model. If no Filter (= Filter Type empty) is set, no restrictions are applied.

  - o **<Dimension> Filter Type**: Dimension to be filtered by 'Element', 'Attribute', or 'Ancestor'.
    - If Filter Type = Element, then a specific element is to be entered in the <Dimension> Filter field.
    - If Filter Type = Attribute, then a specific Attribute is to be selected in the <Dimension> Filter field. The Filter will apply to all elements with value = Y for this attribute.
    - If Filter Type = Ancestor, then a specific Ancestor element is to be entered in the <Dimension> Filter field. The filter is applied to all descendants of the Ancestor

  - o **<Dimension> Filter**: Enter the <Element>, <Attribute>, or <Ancestor>.
    - Region Filter Type, Region Filter,
    - Channel Filter Type, Channel Filter
    - Product Filter Type, Product Filter
    - … Filter Type, … Filter

The following formats are allowable for entering Dimension Granularity Filters:

Format Description	Format	Example 1
Single	A	Company A
Multiples	A+B+C+…	Company A + Company B
All except …	1-(A+B+…)	1-(North America Companies)
Inclusion with Exclusion (with A = Aggregation and B = Aggregation or Leaf)	((A)-(B+…))	((North America Companies)-(Company K + Company L + Company M))
Inclusion with Exclusion (with A,B,C,D = Aggregation or Leaf)	((A+B)-(C+D))	((North America Companies + LATAM Companies) - (Company K + Company L + Company M))
All	1 or ALL	1
empty (no value) => if account belongs to a particular sub-model/cube, entry is allowed for all elements of the dimension where Granularity = "" (empty) => Empty = ALL	empty (no value)	

Filters are applied via rules in the corresponding Submission Models. See section <6> for details.

2.2 Copying, Moving or Deleting Submission Requirements

Submission Requirements - CopyMoveDelete.pro: Process to copy, move, or delete (via parameter pAction) submission requirements data from a source intersection to a target intersection (for copy and move).

2.3   User Interfaces

Submission Requirements Grid Management.xlsx

2.4   Submission Requirements Import

Rather than inputting submission requirements into the 'Submission Requirements' cube, a bulk-import of requirements can be performed via running process 'Submission Requirements - Import.pro' against a Submission Requirements import csv file. The Import is based on the data in the import file. It is recommended to only import data for one version (Actual, Budget, Forecast, Plan) and time period (YYYYMM) at a time, because the process allows zeroing-out the existing submission requirements data by version and time period).

Parameters for process **'Submission Requirements - Import.pro:**
- **pFilename**: Name of the csv file to import
- **pPath**: path/location of the import file. If empty, the 'Outbound Directory Path' as per 'SYS_IBM_Control' cube will be used
- **pZeroOutAsPerDimensionAndElementParameters**: If Y, will zero-out 'Submission Requirements' cube as per the below intersection definition:
- **pCompanyDimension**: Name of Company Dimension
- **pCompany**: Company (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pTimePeriodDimension**
- **pTimePeriod**: Time Period (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pVersionDimension**: Name of Version Dimension
- **pVersion**: Version (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pDataSourceDimension**: Name of Data Source Dimension
- **pDataSource**: Data Source (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pAccountDimension:** Name of Account Dimension
- **pAccount**: Account (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pSubmissionModelDimension**: Name of Submission Model Dimension
- **pSubmissionModel**: Submission Model (suffixes ND;, AD;, IC;, Multi:, MDX;, … are permissible)
- **pLogging**: if set to Y, will log source records & target records and skipped records in 'Process Log Directory path as per 'SYS_IBM_Control' cube
- **pProcessEmptyMeasureValues**: If set to Y, columns (=Measures) with empty values will be processed (cleared-out). If not set to Y, empty values will not be processed.
- **pMaxNoOfRecordsToLog**: Allows logging to be restricted. The max value applies not to the import file rows, but the resulting import records (which could be many times higher because an import file row can apply to many records as per Company and Data Source prefixes)

2.4.1   Import templates and Samples

Import file template (save as csv):

Microsoft Excel
97-2003 Worksheet

2.4.2   Import file formatting requirements and import options

CSV (Comma Separated File)
1st row: column headers (see requirements below)
Rows 2-N: import data (intersection and measure values)

Example:
Company,Time Period,Version,Data Source,Account,Submission Model,Is Required
Corporate,201801,Budget,Post-Allocation,12345678,Financials,Y
Company B,201801,Budget,KPI,23456789,Financials,Y

Columns 1-6:

- o   Column 1: **Company**,
- o   Column 2: **Time** (only YYYYMM),
- o   Column 3: **Version** (only leaf elements),
- o   Column 4: **Data Source** (leafs or consolidations),
- o   Column 5: **Account** (leafs of consolidations), and
- o   Column 6: **Submission Model** (leaf elements)

For Company and Data Source the following prefixes may be used to apply the data from an import row to one or more than one row:

- o   **ND;**<Element/Node>: Apply to all leaf-levels under <Element/Node>
- o   **AD;**<Element/Node>: Apply to all descendants (all consolidations and leaves) under <ElementNode>
- o   **IC;**<Element/Node>: Apply to immediate Children of <Element/Node>
- o   **CD;**<Element/Node>: Apply to C-Level Descendants (including the ;<Element/Node>)
- o   <Element/Node>: apply to single Element/Node
- o   **Multi:**<ElementName1>;<ElementName2>;...;<ElementNameN> = apply to multiple specific Elements/Nodes
- o   **MDX;<MDX Query>**: apply to elements as per MDX query

Note: IF requirements are managed primarily via import file upload, the use of the 'ND;' prefix (and other prefixes) should be used in lieu of defining reference companies. In other words: The concept of a reference company was originally introduced to allow defining one requirement for many companies , because the requirement had to be entered only for the reference company and was then automatically applied to the 'other' companies'. Alternatively, requirements imported like in the example above will also be populated directly for all the corresponding companies. The difference is that in the case of the reference company there will be one record that gets propagated to all other companies via rule, while in the case of the import with prefix 'ND;' the requirement gets directly imported to all corresponding companies

Columns 7-N: Submission Requirements Measures:

- o   Submission Requirements measures need to be in the header row (1st row) of the import, and following the columns for Company, Time, Version, Data Source, Account, and Model. The header names need to be = their corresponding measure element name.
- o   The import process does retrieve the measures from the first row of the import file.
- o   Not all measures need to be provided.

- o Required measures: **'Is Required'** if = Y (because 'Is Required' = Y will determine a submission requirement.
- o Data is imported in the order of the columns in the import file. This means that the order of columns is important where a specific measure depends on a 'previous'/other configuration value. For example: in order to import values for '=0', 'Is Required' needs to be set to Y.
- o 'Is Required (numeric)' is always calculated and hence does not need to be imported

3 Submission Validation

3.1 Submission Validation Model

3.1.1 Submission Validation Cube

The submission validation cube is used to validate submission data from all submission models. Submission Model data is not automatically pulled into the Submission Validation model; it is processed via 'Submission for Validation.pro' (see <Submit data from a submission model into the Submission Validation Model> for details).

Submission Validation occurs by
- Company
- Time
- Version,
- Data Source (Submission)
- Account,
- Currency, and
- Submission Model
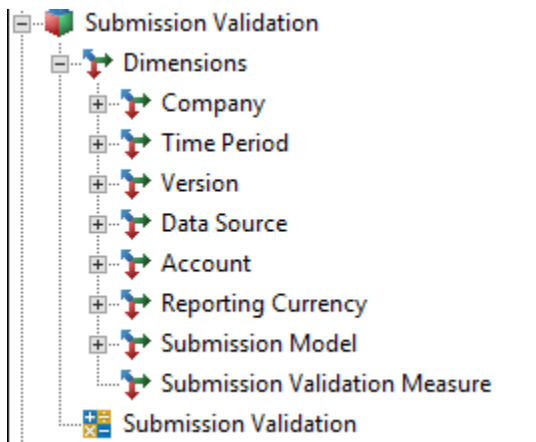
Because the submission validation model will flag missing submission entries as well as submission tie-out violations at the leaf level (via a numeric flag), a drilldown from consolidated Company/Acct/Time/Data Source and Submission Models (all Submission Models) is possible to view and analyze validation details. Conversely, at the consolidated level (or for a division for example), the submission validation model can be used to check validity of a submission as a whole (without needing to check granular details). See section <Submission Validation Measures> below for details on built-in violation checks.

3.1.2 Submission Validation Measures

- **Is Required (Numeric)**: = 1 at leaf level if the submission is required at the intersection of Company, Version, Data Source, Time, & Account. At consolidated level, a number > 0 indicates the number of required submissions.

- **Submission Value (Numeric)**: corresponding submission value (numeric)
- **Submission Violation**: A number > 0 indicates the number of Submission Violations.

- **Tie-Out Account A Value**: submission value for Tie-Out Account A (from Tie-Out Submission A and Submission Model A), as per 'Submission Requirements' Model.
- **Tie-Out Account A Variance**: Variance between 'Submission Value (numeric)' and 'Tie-Out Account A Value'.
- **Tie-Out Account B Value**: submission value for Tie-Out Account B (from Tie-Out Submission B and Submission Model B), as per 'Submission Requirements' Model.
- **Tie-Out Account B Variance**: Variance between 'Submission Value (numeric)' and 'Tie-Out Account B Value'.
- **Tie-Out Account C Value**: submission value for Tie-Out Account C (from Tie-Out Submission C and Submission Model C), as per 'Submission Requirements' Model.
- **Tie-Out Account C Variance**: Variance between 'Submission Value (numeric)' and 'Tie-Out Account C Value'.

- **Tie-Out Violation**: A number > 0 indicates the number of Tie-Out Violations.

Notes:
- Submission Violations, Tie-Out Violations, Tie-Out Values and Variances are only calculated for USD values.
- 'Submission Violation' and 'Tie-Out Validation' will be calculated at a leaf-level and consolidated 'up'. Exception: If a submission requirement is entered under a C-level (parent) company AND if '*Is level-specific Validation Requirement'* is set to Y, the violation will be calculated at the C-level intersection itself.
- If/when modifying Submission Validation Measures, it is not recommended to add any string measures/elements. The Submission Validation model processes submission and tie-out violations via rules at runtime and hence features a performance-optimized design that allows considering and leveraging the last (measures) dimension for dimension sort order optimization. If string measures were added to the dimension, it could not be used for dimension sort order optimization.
- To prevent the submission validation model from being used for reporting and analysis purposes and to improve validation model performance and reduce (rule)maintenance work, the following restrictions apply to the submission validation model:
  o C-level time periods are ruled to 0 (i.e. validations occur at the month level)
  o Version variances are ruled to 0
  o Leaf level data is only calculated for the 'submission versions' Budget, Actual, Forecast, Plan. Versions may be copied to 'historical versions' and c-level rules will apply for the historical versions:
- Models to include in version management process (process used when copying versions to historical versions): 'Submission Requirements', 'Submission Requirements - Company to Reference Company Mapping', 'Submission Validation', & 'Submission Validation Commentary'.
- Commentary and workflow related to Submission Validation should be handled via a supplementary model such as 'Submission Validation Commentary':

3.1.3   Submission Validation User Interface

Submission Validation.xlsx

The Submission Validation Workbook includes both the Submission Validation Measures as well as corresponding Submission Requirements Values from the Submission Requirements model:

Submission Value, Submission Violations, Tie-Out Violations, Tie-Out Values, Tie-Out Variances:

3.2   Submission Validation Commentary

Allows entering commentary pertaining to submission value and tie-out violations:
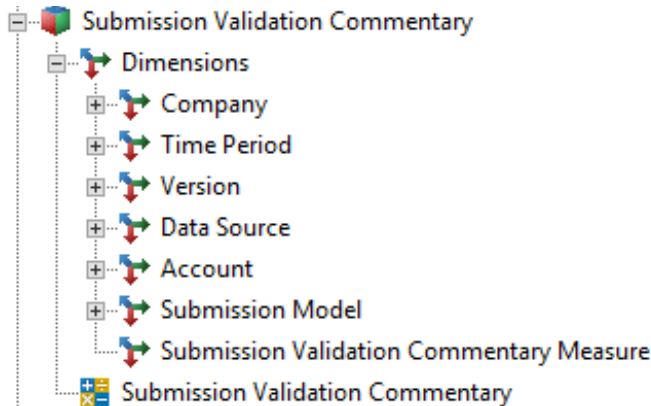
Measures:

- **Submission Value Violation**: rule in from Submission Validation Model. Values > 0 indicate a submission violation. If > 0, a comment, remediation data and business contact may be entered.

- **Tie-Out Violation**: rule in from Submission Validation Model. Values > 0 indicate a tie-out violation. If > 0, a comment, remediation data and business contact may be entered.

- **Comment**: free text input

- **Remediation Date**: YYYYMM (up to 3 months into future, based on YYYYMM of violation)

- **Business Contact**: free text input of business contact

3.3   Submit data from a submission model into the Submission Validation Model

3.3.1   Technical Submission Methods

Options for submitting data to the submission model:

a)   Via Cognos Command Center or Rest API or equivalent

b)   Via TI-process (for intra-model/database/instance submissions)

3.3.2   Ti-process-based submissions

Submission for Validation.pro: Processes data from a submission model to the submission validation cube. If the parameter pWorkflowCheck is set to Y and pLock is set to Y, will lock all submitted records in **Workflow - Account Status**. Data may be re-submitted using this process even if the Workflow Status is locked, because the locked status needs to (as per corresponding SecurityOverlay or CellSecurity rules) the source data from changing, hence leading to a re-submission of the same data.

Parameters:

- pSourceCube: Source Cube
- pVersion: Actual, Forecast, Plan, …
- pDataSource: Data Source? (optional for source, but needs to be specified for the target)
- pCompany: Company/BU/… (company dimension can be used in multiple forms)
- pAccount: Account?
- pTimePeriod: YYYYMM or YYYY or YYYYQ
- pSourceViewIsCLevel: if the source data is to be queried at a consolidated level, set to Y, otherwise = N (empty = N)
- pLogging = process logging Y/N
- pWorkflowCheck: Set to Y to check workflow/locking status
- pLock: Set to Y to lock further input into specified accounts by time and company (requires pWorkflowCheck = Y)
- pThread: leave empty - will be populated if needed by parallel processing governor
- pCompletionTriggerFile: leave empty - will be populated if needed by parallel processing governor

For parameters pVersion, pDataSource, pCompany, pAccount, pTimePeriod, the following options are available:

- ○   <Element/Node>: specific element/node (if = node, then pSourceViewIsCLevel needs to be set to Y)
- ○   **ND;**<Element/Node>: Apply to all leaf-levels under <Element/Node>
- ○   **AD;**<Element/Node>: Apply to all descendants (all consolidations and leaves) under <ElementNode>
- ○   **IC;**<Element/Node>: Apply to immediate Children of <Element/Node>
- ○   **CD;**<Element/Node>: Apply to C-Level Descendants (including the ;<Element/Node>)
- ○   <Element/Node>: apply to single Element/Node
- ○   **Multi:**<ElementName1>;<ElementName2>;...;<ElementNameN> = apply to multiple specific Elements/Nodes
- ○   **MDX;<MDX Query>**: apply to elements as per MDX query

For mapping of source models to the submission models, the process 'Submission for Valdiation.pro' leverages mappings as per the IBM Planning Analytics Cube, Dimension, and Data-Load Framework. Consequently, source Cube to 'Submission Validation' cube mappings are to be defined as per 'TM1 Cube to Data Source Mapping.cub'. **Submission for Validation.pro** will pass Version, Company, Account, Data Source, Time Period, and Currency (see MDX in prolog) filter information to Process 'Manage Cube - Load or Update Data - 23Dim - V2.pro':

- pVersion: passed to pSource_DimensionA_Element
- pDataSource: passed to pSource_DimensionB_Element
- pCompany: passed to pSource_DimensionC_Element

- pAccount: passed to pSource_DimensionD_Element
- pTimePeripd: passed to pSource_DimensionE_Element
- pCurrency: passed to pSource_DimensionF_Element

Corresponding dimension mapping will have to be provided in 'TM1 Cube to Data Source Mapping.cub' as follows:

- Source Dimension A Name = <SourceCubeVersionDimension>
- Source Dimension B Name = <SourceCubeDataSourceDimension>
- Source Dimension C Name = <SourceCubeCompanyDimension>
- Source Dimension D Name = <SourceCubeAccountDimension>
- Source Dimension E Name = <SourceCubeTimePeriodDimension>
- Source Dimension F Name = <SourceCubeCurrencyDimension>
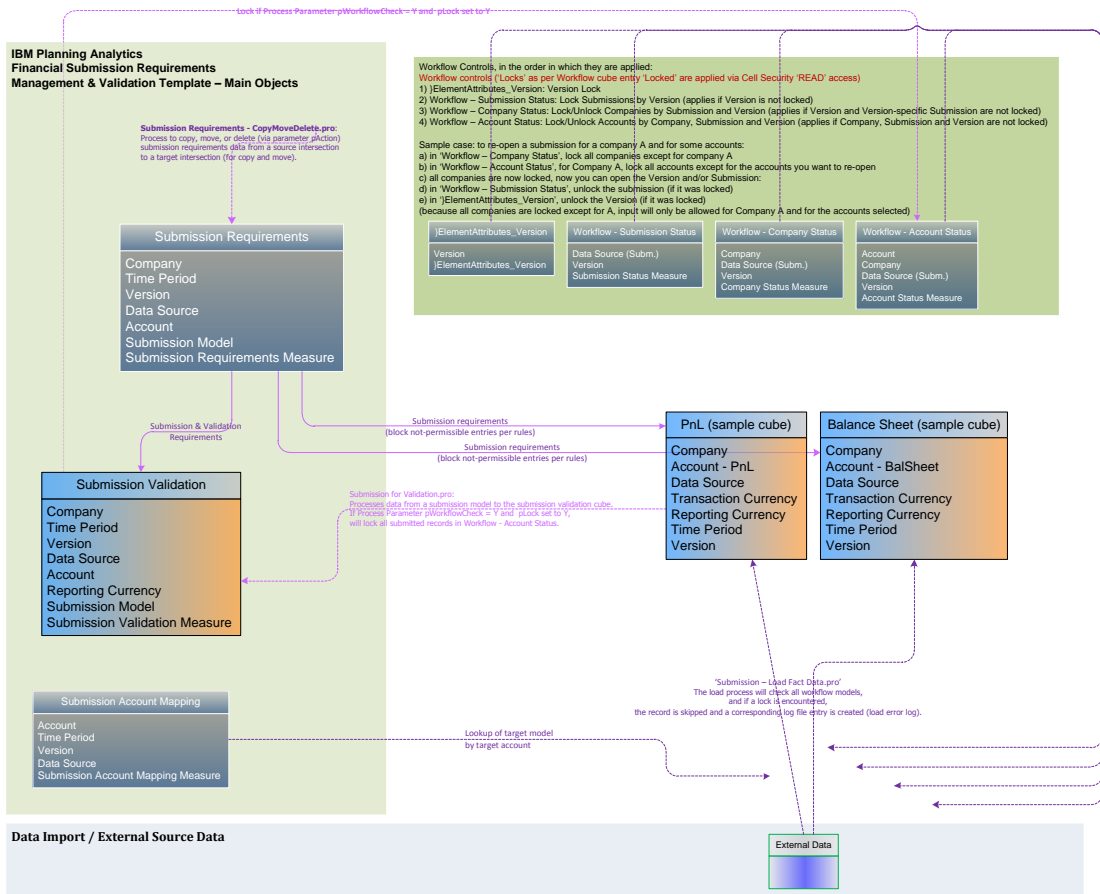
4 Submission Requirements Management Model Architecture Overview

IBM Planning Analytics
Financial Submission Requirements
Management & Validation Template

Created By:
Andreas Kugelmeier, IBM

Date Last Saved:
**January 27, 2020**

Version: 1.2

Arrows:
 Blue arrow = ACTUALS
 Orange arrow = FCST/Budget/Budget
 Purple arrow = ACTUALS and FCST/Budget/Budget
 dashed (-----): Process
 dotted (.....): user input/contribution/ analysis via User Interface (web and Excel)
 straight (-): rule

Fonts/Border-Colors:
 Blue font color (Arrows): Actuals
 Orange font color (Arrows): FCST/Budget/Budget
 Purple font color (Arrows): Actuals and FCST/Budget/Budget

Cube
 Black font (Cube) = Read/Write, FCST/Budget/Budget Input, Reporting & Analysis
 Blue Font (Cube) = Reporting & Analysis cube (no input)

Background Colors:
 Blue Model color = ACTUALS
 Orange Model color = FCST/Budget/Budget

IBM Planning Analytics
Financial Submission Requirements
Management & Validation Template – Main Objects

Lock if Process Parameter pWorkflowCheck = Y and pLock set to Y

Submission Requirements - CopyMoveDelete.pro:
Process to copy, move, or delete (via parameter pAction) submission requirements data from a source intersection to a target intersection (for copy and move).

Workflow Controls, in the order in which they are applied:
Workflow controls ('Locks' as per Workflow cube entry 'Locked' are applied via Cell Security 'READ' access)
1))ElementAttributes_Version: Version Lock
2) Workflow – Submission Status: Lock Submissions by Version (applies if Version is not locked)
3) Workflow – Company Status: Lock/Unlock Companies by Submission and Version (applies if Version and Version-specific Submission are not locked)
4) Workflow – Account Status: Lock/Unlock Accounts by Company, Submission and Version (applies if Company, Submission and Version are not locked)

Sample case: to re-open a submission for a company A and for some accounts:
a) in 'Workflow – Company Status', lock all companies except for company A
b) in 'Workflow – Account Status', for Company A, lock all accounts except for the accounts you want to re-open
c) all companies are now locked, now you can open the Version and/or Submission:
d) in 'Workflow – Submission Status', unlock the submission (if it was locked)
e) in ')ElementAttributes_Version', unlock the Version (if it was locked)
(because all companies are locked except for A and for the accounts selected)

Submission Requirements
Company
Time Period
Version
Data Source
Account
Submission Model
Submission Requirements Measure

)ElementAttributes_Version
Version
)ElementAttributes_Version

Workflow - Submission Status
Data Source (Subm.)
Version
Submission Status Measure

Workflow - Company Status
Company
Data Source (Subm.)
Version
Company Status Measure

Workflow - Account Status
Account
Company
Data Source (Subm.)
Version
Account Status Measure

Submission requirements
(block not-permissible entries per rules)

Submission requirements
(block not-permissible entries per rules)

Submission & Validation
Requirements

Submission Validation
Company
Time Period
Version
Data Source
Account
Reporting Currency
Submission Model
Submission Validation Measure

Submission for Validation.pro:
Processes data from a submission model to the submission validation cube.
If Process Parameter pWorkflowCheck = Y and pLock set to Y,
will lock all submitted records in Workflow - Account Status.

PnL (sample cube)
Company
Account - PnL
Data Source
Transaction Currency
Reporting Currency
Time Period
Version

Balance Sheet (sample cube)
Company
Account - BalSheet
Data Source
Transaction Currency
Reporting Currency
Time Period
Version

Submission Account Mapping
Account
Time Period
Version
Data Source
Submission Account Mapping Measure

Lookup of target model
by target account

'Submission – Load Fact Data.pro'
The load process will check all workflow models,
and if a lock is encountered,
the record is skipped and a corresponding log file entry is created (load error log).

Data Import / External Source Data

External Data

5 Submission Data Load & Mappings

- **Submission - Fact Data Load.pro**: This process is used for loading external fact data from flat files (and can be enhanced to also load from SQL). The load process will check all workflow models, and if a lock is encountered, the record is skipped and a corresponding log file entry is created (load error log).

The process will leverage the 'Submission Account Mapping' model to tie the file-based Account, Version, Data Source, and Month to a submission contribution model:
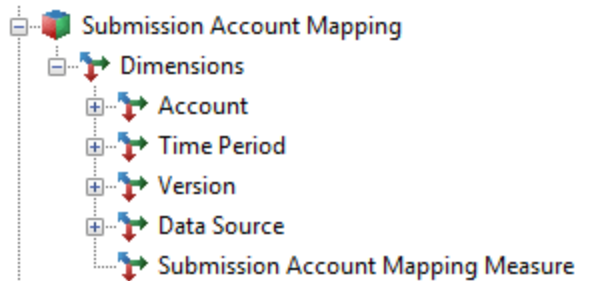
For example, if the load file contains a record for Company X, Account A, Version Actual, Submission Data Source 'Input', Month 201305, the 'Submission Account Mapping' model will be queried to retrieve the target model/cube Y.

Columns are mapped dynamically to the target models via mapping the column names to corresponding metadata information (and dimension names) as specified in the prolog (see below for variables and parameters that need to be configured). Also, security and workflow checks are conducted during data load.

Note that this process should only be used for loading smaller to medium size data sets.

Variables & Parameters to be configured in the prolog of Submission – Fact Data Load:

- o sAccountDimension = '<AccountDimensionName>'
- o sVersionDimension = '<VersionDimensionName>'
- o sTimePeriodDimension = '<TimePeriodDimensionName>';
- o sSubmissionDataSourceDimension = '<DataSourceDimensionName>';
- o sSubmissionModelDimension = '<SubmissionModelDimensionName>';
- o sCompanyDimension = '<CompanyDimensionName>';
- o sValueColumn = '<ValueColumnHeader>';
- o sUnitOfMeasureColumn = '<UnitOfMeasureColumnsHeader>';
- o sReportingCurrencyDimension = '<ReportingCurrencyDimensionName>';
- o sTransactionCurrencyDimension = '<TransactionCurrencyDimensionName';
- o sReportingCurrencyDefault = 'USD';
- o sNoCurrencyDefault = '<DefaultElementForNoCurrency>';
- o sRequiredAccess = 'WRITE';
- o sCompanyAncestorAsPerFileName = 'N'; (set to Y if the submitting company is to be derived as part of the file name)
- o nTargetCompanyInRow = 0; (if the submitting company is in a specific row, set to this row #, otherwise, leave at 0 in which case the submitting company is read per row)
- o nTargetVersionInRow = 0; (if the version for the submission is in a specific row, set to this row #, otherwise, leave at 0 in which case the version is read per row)
- o nTargetDataSourceInRow = 0; (if the data source for the submission is in a specific row, set to this row #, otherwise, leave at 0 in which case the data source is read per row)

- nTargetTimePeriodInRow = 0; (if the YYYYMM period for the submission is in a specific row, set to this row #, otherwise, leave at 0 in which case the time period is read per row)
- nFirstFactRecord = 2; (set to the first row with fact records. Example: row 1 is for the headers, row 2 could be used to defined submitting company and version, and fact would be in rows 3-N)
- sSecurityCheck = 'N';
- sWorkflowCheck = 'N';
- sUserPartOfFileName = 'N'; (set to Y if the user is part of the file name, user name/ID is then extracted from the file name and used for security checks. If set to N, the user processing the file is used)
- sCharacterFollowingUserName = ''; (a special character to be used to allow for parsing out the user name)
- sCompanyIsPartOfFileName = 'N'; (set to Y if the company is part of the file name, the company is then extracted from the file name and used as the importing company. If set to N, the company as per file content is used)
- sCharacterFollowingCompanyName = ''; (a special character to be used to allow for parsing out the company)

6   Submission Granularity Filters and Restrictions

Submission granularity filters - which in the Submission Requirements model may be used to define needed granularity of a submission - are enforced not in the submission validation model (which has a low granularity and hence would not suffice), but in the submission models themselves. This also ensures that a requirement is input at a desired granularity during contribution.

Corresponding '<Dimension> Filter Type' elements and '<Dimension> Filter' elements will have to be added to the measures dimension of the Submission Requirements model

```
# rule to block input for non-required submissions
# ONLY use this rule IF
# A) submission requirements will and must cover/include all needed accounts
# B) AND if non-required accounts are to be blocked !!!
[{<Versions/Scenarios>},{<DataSources>},'Transaction Currency'} = N:
        IF ( DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', 'Is Required (numeric)') = 0,
                0,
                Continue);


# rules to enforce submission granularity WHERE defined
# apply the submission granularity filter rule(s) for the dimensions that apply to each submission model
# examples:
# - no submisison granularity rules for Financials (b/c Financials does not have any addtl. dimensions for which filters apply)
# - Sales and Profit shall use submisison granularity rules for dimensions Product, Channel, Region, Brands
# <Dimension> Filter
[{'Actual', 'Budget','Forecast', 'Plan'},'Reporting Currency':'Transaction Currency'] = N:
        IF ( DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', '<Dimension> Filter Type') @= '',
                continue,
                # if filter type = element, only allow entry against specific element
                IF ( DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', '<Dimension> Filter Type') @= 'Element',
                        IF ( !<Dimension> @<> DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', '<Dimension> Filter'),
                                0,
                                continue),
                # if filter type = ancestor, only allow entry if leafs are ancestors of element
                IF ( DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', '<Dimension> Filter Type') @= 'Ancestor',
                        IF ( ELISANC ( '<Dimension>', DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>',
                        '<Dimension> Filter'), !<Dimension> ) = 0,
                                0,
                                continue),
                # if filter type = attribute, only allow entry if attribute value for element is = Y
                IF ( DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source, '<AccountDimension>, '<Submission Model>', '<Dimension> Filter Type') @= 'Attribute',
                        IF ( ATTRS ( '<Dimension>', !<Dimension>, DB('Submission Requirements',!Company,!Time Period,!Version,!Data Source,  , '<AccountDimension>, '<Submission Model>',
                        '<Dimension> Filter')) @<> 'Y',
                                0,
                                continue),
        continue))));
```

7   TM1/Planning Analytics Objects

IBM Planning Analytics Submission Requirements Management and Validation Template Objects