Workflow with IBM Planning Analytics and TM1:

Types, Design Options, Caveats, Guidelines, Tips & Tricks

Prepared:
July 2016
Last updated:
January 2020

Created By:

Andreas Kugelmeier
Executive Consultant, FOPM
TM1 & Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-215-384-7302
Email: kugelmeier@us.ibm.com

William McBride
Senior Managing Consultant,
TM1 & Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-817-807-1951
Email: wmcbride@us.ibm.com

Notices & Disclaimers

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations and papers (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS document is distributed "AS IS" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Document Version History

Date	Version	Author	Description
5/25/2016	0.9	Andreas Kugelmeier	1st Draft Version
7/6/2016	0.95	Bill McBride	Review 1st Draft, Misc. Edits and additions => 2nd Draft Version
7/7/2016	1.0		1st Version
7/8/2016	1.01	Andreas Kugelmeier	Correct typo
10/13/2016	1.02	Andreas Kugelmeier	Misc. minor edits to section 3.1
4/16/2017	1.03	Andreas Kugelmeier	Misc. minor edits
01/16/2020	1.04	Andreas Kugelmeier	• Update to 4.2.5: If CellSecurity is in place and requires a similar dimensionality as SecurityOverlay, and if a Security Overlay can be achieved via CellSecurity rules , and if the # of user groups is not too small, it is an alternate practice to use CellSecurity (to reduce complexity for security management and auditing). • Update to diagram 4.3

Table of Contents

1. About this document 5

2. Workflow Types 6

2.1 Version/Scenario Control Workflow 6

2.2 Responsibility Workflow 6

2.3 Approval Workflow 7

2.4 Workflow Types: Overview 7

3. Workflow Method Decision Support: Overview 8

3.1 Workflow Design Options by Type: Advantages and Disadvantages 8

3.2 Custom Workflow in TM1 vs. 'TM1 Applications' Workflow 9

4. Workflow Implementation: Functionality, Practices, Caveats, & Guidelines 10

4.1 Scenario Control Workflow 10

4.1.1 Proven Practices, Assets for Version Management Design 10

4.1.2 Workflow Management using Version dimension attributes 10

4.1.2.1 Example Approach 10

4.1.2.2 Cell Security Examples 10

4.2 Responsibility and Approval Workflow 11

4.2.1 Enforcing Security Rights in TM1 Applications Workflow 11

4.2.2 CellSecurity 12

4.2.3 ElementSecurity 13

4.2.4 Integrating (Non-)Approval Hierarchy Dimension Element Security into TM1 Applications Workflow 15

4.2.5 TM1 Workflow Management 16

4.3 Workflow Cube Security Decision Support Matrix 19

5. Misc. Guidelines, Tips & Tricks 20

1. About this document

Workflow has different meanings depending on the context of its use. Within the functional context, Workflow describes how the collection of data as well as the underlying business process is managed and distributed amongst contributors. Within planning and budgeting, Workflow is often referred to as managed contribution. For example, an application that distributes the collection of plan data by cost center will use Cost Center as "Workflow", and plan contributors are distributed by Cost Center.

Within the technical context of TM1, Workflow is defined by how TM1 manages and controls user access to data. TM1 attempts to align the management of data to common functional Workflow types via three TM1 Application types. Additionally, TM1 controls user access to data through specific configuration options (both TM1 server and application type), specialized application security, and traditional security. Application types include Central, Responsibility, and Approval, and are configured through Performance Modeler and deployed using TM1 Applications Web.

This document provides workflow decision support guidelines and design methods in TM1. Information is presented within the context of typical Workflow examples such as version control, managed contribution, and plan submission. Typical Workflow types are described and guidance is provided in choosing a technical approach, and which security methods to use. Workflow design and implementation caveats, guidelines, tips and tricks are also provided and discussed.

2. Workflow Types

2.1 Version/Scenario Control Workflow

Characteristics:

- Workflow is controlled through the simple attachment of credentials to members of a version or scenario dimension.
- Typically includes a version or scenario dimension that will be locked (set to read-only) or open for contribution within the context of a defined business process or timeline.
- Does not use submission or user initiated locking of data by design. Upon reaching a deadline or milestone, the entire version or scenario will be locked.
- Version or Scenario Workflow is good practice in environments where managed contribution doesn't require a formal hierarchy of submission, review, and rejection/approval, and the organizational structure or division of labor for contribution is very flat.
- Single Model Version Control Workflow: Only one model needs to be workflow-controlled/secured.
- Multi-Model Version Control Workflow: Multiple models need to be workflow-controlled/secured.

Typical Design Methodologies:

- Single Model Version Control Workflow: Control via Version element security on Version Dimension (no rules, but TI-process driven[1]).
- Multi-Model Version Control Workflow (multiple models in an instance, where the version status may differ, i.e. where the models may not move in lockstep): Control via model-specific Version Dimension Attributes and corresponding Cell Security Rules.
- TM1 Applications:
  - Build using the Application Type: 'Central'.
  - Note: TM1 Applications as an interface or deployment tool is not required for Version/Scenario Workflow.

2.2 Responsibility Workflow

Characteristics:

- An ongoing rolling type of workflow with view and edit rights.
- Requires a managed contribution approach, but doesn't require a formal submission process where data submission and approval are required.
- Data collection is partitioned using a hierarchy dimension elements.
- Data entry is ongoing, i.e. absent of fixed deadlines or deadlines that must be jointly adhered to by plan contributors, or the system is not required enforce a lock upon completion of data entry.
- In most cases, data entry is limited to a time period and/or a scenario/version (such as Forecast/Budget, or Trial Balance Adjustments for Actuals during period close).

Typical Design Methodologies:

- Custom Workflow:
  - Control via Version element security on Version Dimension (no rules, but TI-process driven) & Version attributes (best way to 'open' and 'close' a Workflow Process via CellSecurity rules).
  - Control via ElementSecurity on the Approval Hierarchy Dimension (like Cost Center for example).
  - Control via CellSecurity rules based on Version Dimension attributes.

[1] Cube Security ('}CubeSecurity.cub'), Dimension Security ('}DimensionSecurity.cub'), Process Security ('}ProcessSecurity.cub'), and Element Security Data ('}ElementSecurity_<Dimension>.cub') should be processed via TI instead of cube rules. If rules are used, a security metadata change - for example due to a hierarchy change (with corresponding/resulting security changes for parent and/or child nodes) or due to a new element being added to a hierarchy will require running the 'SecurityRefresh()' command in TM1, effectively rendering all cached security settings invalid and hence renewing/refreshing all security credentials. A security refresh on large models will typically lead to a multi- to many minute lock of all user activity due to TM1 refreshing security access credentials for all active users and groups. If security is manually entered or processed via TI (and hence directly stored in the corresponding security cube), a security refresh is not necessary. The security changes will propagate automatically and with only very short locks.

- TM1 Applications:
  - Build using the Application Type: 'Responsibility'.
  - Will require definition and use of an 'Approval Hierarchy'.
  - Optional: Use of a 'Control' Dimension (such as Version/Scenario); not typically used for a single model, but can be useful for multi-model.

2.3 Approval Workflow

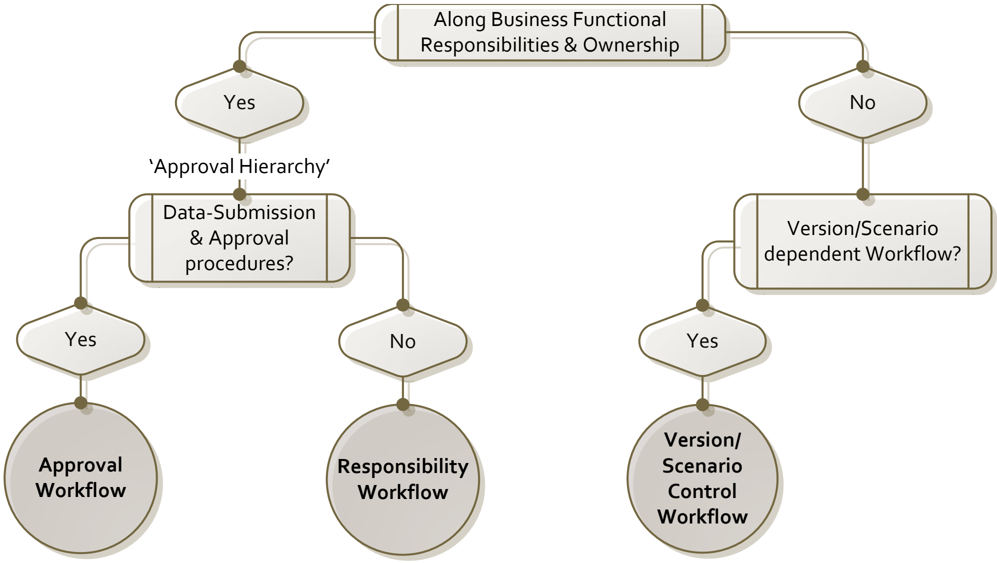
Characteristics:

- A bottom-up, submission-based workflow requiring Edit, Submit, and Review/Reject capability.
- Requires a managed contribution approach, and a formal submission process where ownership and locking are required using an 'Approval Hierarchy'.
- In most cases, data entry is limited to a time period and/or a Version/Scenario (such as Forecast/Budget or Trial Balance Adjustments for Actuals during period close).

Typical Design Methodologies:

- Custom Workflow:
  - Control via Version element security on Version Dimension (no rules, but TI-process driven) & Version attributes (best way to 'open' and 'close' a Workflow Process via CellSecurity rules)
  - Control via ElementSecurity on the Approval Hierarchy Dimension (like Cost Center for example)
  - Control via ElementAttributes on the Approval Hierarchy Dimension (like Cost Center for example) or a Workflow Control cube, including the approval hierarchy dimension, the version dimension & workflow status measures.
  - Control via CellSecurity rules based on Version Dimension attributes (for 'opening' and 'locking' the entire process) and Approval Hierarchy Dimension ElementAttributes or the Workflow Control cube (to lock/unlock sections of the cube based on workflow status).
  - Data Reservations may be acquired against approval hierarchy nodes and/or other dimensions
- TM1 Applications:
  - Build using the Application Type: 'Approval'.
  - Requires defining and using an 'Approval Hierarchy'.
  - Optional: Using a 'Control' Dimension (such as Version/Scenario); not typically used for a single model, but can be useful for multi-model.
  - Data Reservations are acquired against approval hierarchy nodes

2.4 Workflow Types: Overview


Along Business Functional Responsibilities & Ownership
Yes
No
'Approval Hierarchy'
Data-Submission & Approval procedures?
Version/Scenario dependent Workflow?
Yes
No
Yes
Approval Workflow
Responsibility Workflow
Version/ Scenario Control Workflow

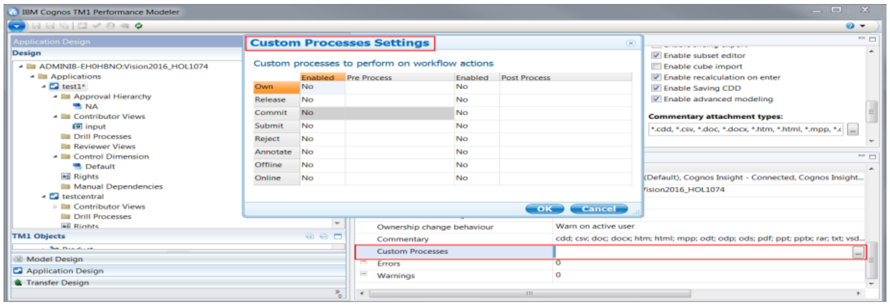
3. Workflow Method Decision Support: Overview

TM1 provides two principal Workflow design options
a) built-in workflow functionality via TM1 'Contributor' Applications and
b) custom workflow

3.1 Workflow Design Options by Type: Advantages and Disadvantages

Workflow Type	Design Option	Advantages	Disadvantages
Scenario (Version) Control	**Custom**	• Simple controls • High transparency of security regime • Fast performance • Highly scalable	
	TM1 Applications	n/a	n/a
Responsibility	**Custom**	• Code simplicity • High transparency of security regime • Fast performance • Highly scalable • Ability to optimize for testing, validation and auditing efforts around security and workflow	• Higher initial development effort: ○ Custom UI needs to be built ○ Commentary, email integration, warnings, etc. require custom development ○ Custom Workflow management utilities • No Offline Planning with Cognos Insight (Distributed Mode)
	TM1 Applications	• Out-of-the-box functionality: ○ Out of the box UI ○ Out-of-the-box Workflow management utilities & functionalities*** • Good performance for smaller to mid-sized environments (for larger environments, dependent on specific Workflow Security Implementation, see section 3.2 below & column to the right) • Offline Planning with Cognos Insight in Distributed Mode	• Code complexity and lack of transparency** • Performance impact for larger environments • Performance impact if ElementSecurity needs to be enforced for Approval Hierarchies, requiring issuing a SecurityRefresh every time Approval Hierarchy ElementSecurity changes • Increased testing, validation and auditing efforts around security and Workflow • Customization may introduce additional code complexity as auto-generated code will have to be overruled
Approval	**Custom**	• Code simplicity • High transparency of security regime • Fast performance • Highly scalable • Ability to optimize for testing, validation and auditing efforts around security and workflow • Higher Flexibility*	• Higher initial development effort: ○ Custom UI needs to be built ○ Commentary, email integration, warnings, etc. require custom development ○ Custom Workflow management utilities • No Offline Planning with Cognos Insight in Distributed Mode
	TM1 Applications	• Out-of-the-box functionality: ○ Out of the box UI ○ Out-of-the-box workflow management utilities & functionalities*** • Good performance for smaller to mid-sized environments (for larger environments, dependent on specific Workflow Security Implementation, see section 3.2 below & column to the right) • Offline Planning with Cognos Insight in Distributed Mode	• Code complexity and lack of transparency** • Performance impact for larger environments • Performance impact if ElementSecurity needs to be enforced for Approval Hierarchies • Reduced transparency of security regime • Increased testing, validation and auditing efforts around security and workflow • Limited to 'standard' Workflow (Approval Hierarchy & Control Dimension) • Customization may introduce additional code complexity as auto-generated code will have to be overruled.

*: Taking Ownership does not have to be tied to approval hierarchy and control dimension alone. Other dimensions may be integrated.
**: High implicit complexity of workflow security objects & management ('under the hood')
***: Notifications, Commentary, Warning of Ownership Changes, Workflow-related actions in can be configured to automatically trigger a turbo-integrator process:



Custom Processes Settings

Custom processes to perform on workflow actions

	Enabled	Pre Process	Enabled	Post Process
Own	No		No	
Release	No		No	
Commit	No		No	
Submit	No		No	
Reject	No		No	
Annotate	No		No	
Offline	No		No	
Online	No		No	

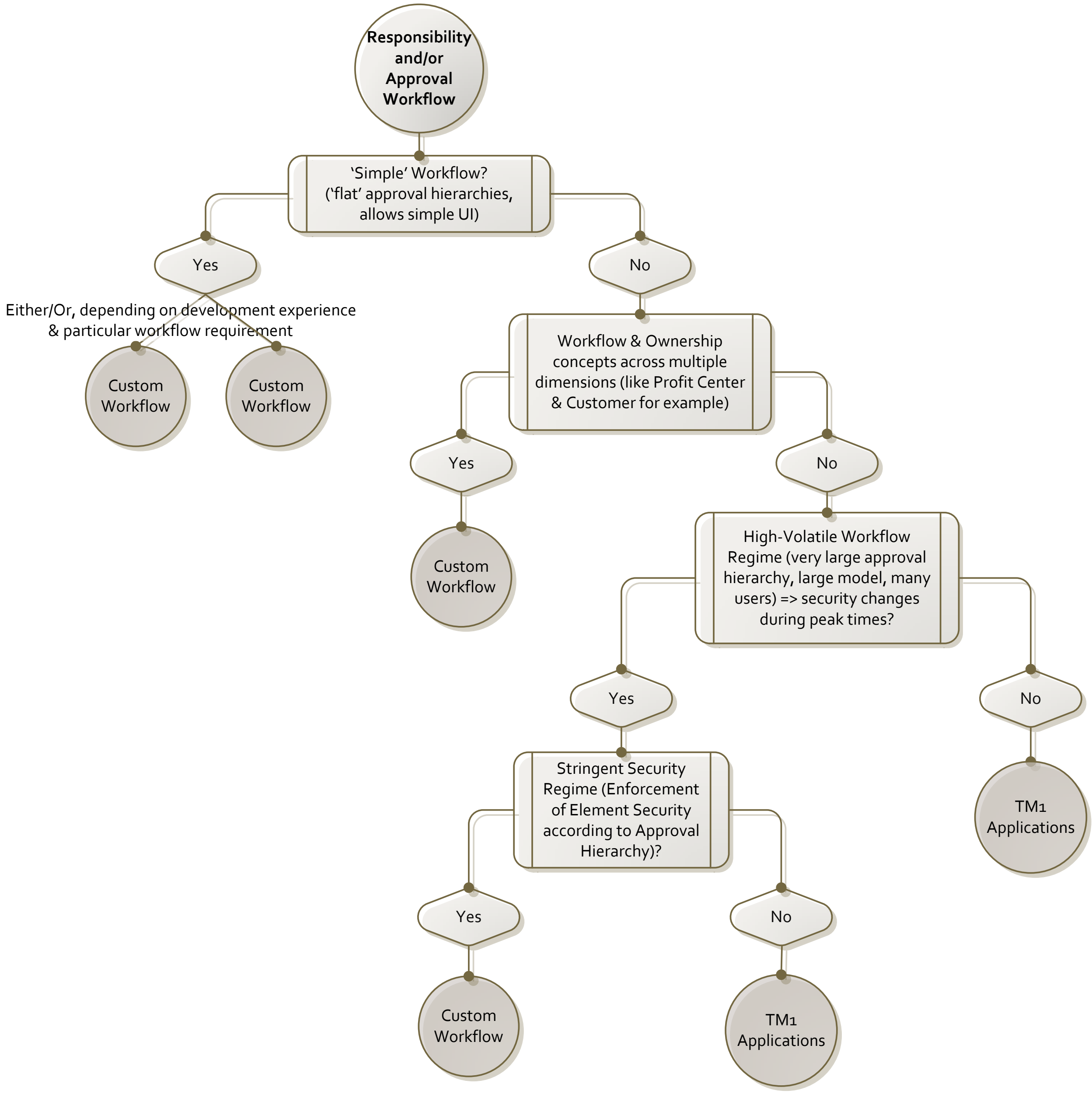
☑ Enable subset editor
□ Enable cube import
☑ Enable recalculation on enter
☑ Enable Saving CDD
☑ Enable advanced modeling

Commentary attachment types: *.cdd, *.csv, *.doc, *.docx, *.htm, *.html, *.mpp, *.

(Default), Cognos Insight - Connected, Cognos Insight...
Vision2016_HOL1074

Ownership change behaviour	Warn on active user
Commentary	cdd; csv; doc; docx; htm; html; mpp; odt; odp; ods; pdf; ppt; pptx; rar; txt; vsd...
Custom Processes	
Errors	0
Warnings	0

3.2 Custom Workflow in TM1 vs. 'TM1 Applications' Workflow


Responsibility and/or Approval Workflow
'Simple' Workflow? ('flat' approval hierarchies, allows simple UI)
Yes
No
Either/Or, depending on development experience & particular workflow requirement
Custom Workflow
Custom Workflow
Workflow & Ownership concepts across multiple dimensions (like Profit Center & Customer for example)
Yes
No
Custom Workflow
High-Volatile Workflow Regime (very large approval hierarchy, large model, many users) => security changes during peak times?
Yes
No
TM1 Applications
Stringent Security Regime (Enforcement of Element Security according to Approval Hierarchy)?
Yes
No
Custom Workflow
TM1 Applications

4. Workflow Implementation: Functionality, Practices, Caveats, & Guidelines

4.1 Scenario Control Workflow

4.1.1 Proven Practices, Assets for Version Management Design

Whenever workflow is to be tied to a Version or Scenario dimension, proven practices for Version Dimension Design & Scenario Management should be followed: Details can be found in Proven Practices for TM1 Version and Scenario Management and Variance Analysis incl design template.docx

4.1.2 Workflow Management using Version dimension attributes

4.1.2.1 Example Approach

1.) Create Version dimension attributes of type '<Sub-Model> Lock', with a Y/N picklist. Example: Attributes 'P&L Lock', 'Workforce Lock'.

2.) Create cell security rules on the corresponding sub-models that will overrule potential WRITE access with READ in case the attribute is set to '* Locked' = Y. Sample cell security rules are outlined in the following section:

4.1.2.2 Cell Security Examples

In the following example (& following the above sample approach) we are applying simple workflow security to a P&L cube, depending on the P&L Lock status as well as Element Security for Version, Cost Center and Account[2]:

```
#Workflow Rule: will overrule non-READ access with READ access
[] = S:
    IF ( ISLEAF () = 1,
        IF ( '}ElementAttrbutes_Version, !Version, 'P&L Lock' ) @= 'Y',
                IF ( DB('}ElementSecurity_Version', !Version, !}Groups ) @= 'READ'
                   % DB ('}ElementSecurity_Version', !Version, !}Groups ) @= '',
                        continue,
                        IF ( DB ( '}ElementSecurity_Account', !Account, !}Groups ) @<> ''
                        & DB( '}ElementSecurity_CostCenter', !CostCenter, !}Groups ) @<> '',
                                'READ',
                                Continue),
                continue),
        continue);

#Rule for Rolling Versions/Scenarios such as Rolling Forecast, to prevent overwriting of 'actualized' periods. Common rule in models where
actuals are nor ruled in, but processed into a FCST or Plan scenario
[] = S:
    IF ( ISLEAF () = 1,
        IF ( NUMBR ( !Time Period) < DB('}ElementAttributes_Version', !Version, 'Actuals Through Date'),
                IF ( DB('}ElementSecurity_Version', !Version, !}Groups ) @<> ''
                    & DB ( '}ElementSecurity_Account', !Account, !}Groups ) @<> ''
                    & DB ( '}ElementSecurity_CostCenter', !CostCenter, !}Groups ) @<> '',
                        'READ',
                        Continue),
                continue);
```

In the rule above, all applicable dimensions with ElementSecurity had to be taken into consideration such as to avoid that CellSecurity can overrule ElementSecurity with less restrictive access. Yet if in

[2] We are assuming the cube contains numeric values only and hence will only apply our cell security workflow rules to leaf level elements

}CubeSecurityProperties, ‘CELLSECURITYMOSTRESTRICTIVE’ is set to ‘Yes’ for the cube(s) we want to secure via workflow, the rules can be simplified[3]:

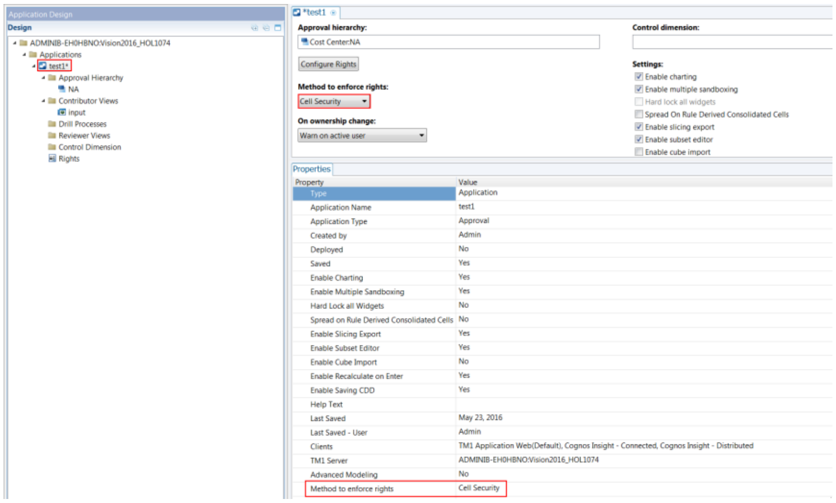
```
#Workflow Rule: will overrule non-READ access with READ access
[] = S:
      IF ( ISLEAF () = 1,
            IF ( ‘}ElementAttrbutes_Version, !Version, ‘P&L Lock’ ) @= ‘Y’,
                         ‘READ’,
                        Continue),
            continue);

#Rule for Rolling Versions/Scenarios such as Rolling Forecast, to prevent overwriting of ‘actualized’ periods. Common rule in models where
actuals are nor ruled in, but processed into a FCST or Plan scenario
[] = S:
      IF ( ISLEAF () = 1,
            IF ( NUMBR ( !Time Period) < DB('}ElementAttributes_Version', !Version, 'Actuals Through Date'),
                         ‘READ’,
                        Continue),
            continue);
```

4.2 Responsibility and Approval Workflow

4.2.1 Enforcing Security Rights in TM1 Applications Workflow

Depending on the security enforcement methods selected the TM1 Application Server will create various additional security objects and structures ‘on top of’ existing TM1 security. Where applicable, (existing) Element or Cell Security will be overruled or replaced by security metadata sourced from special TM1 Applications Workflow security objects or procedures. [4] As some of the below examples will illustrate, the subsequent workflow security and control regime built via Performance Modeler’s TM1 Applications can become complex. Implementation of TM1 Applications workflow should hence always go along with a thorough (re)-test of a) TM1 Security and (b) TM1 Security with TM1 Workflow. Effects of Workflow design on TM1 Security should be properly documented. It can be a good practice to use dedicated security groups for workflow (and to ensure via audit controls that this practice is followed). This way, both workflow and non-workflow related security can be maintained, tested & audited easier.

When designing a TM1 Applications via the Performance Modeler TM1 Applications modeling interface, the **Method to enforce rights** determines the method by which an approval or responsibility application is secured and controlled:



- ADMINIB-EH0HBNO:Vision2016_HOL1074
  - Applications
    - test1*
      - Approval Hierarchy
        - NA
      - Contributor Views
        - input
      - Drill Processes
      - Reviewer Views
      - Control Dimension
      - Rights

Approval hierarchy: Cost Center:NA

Control dimension:

Configure Rights

Method to enforce rights: Cell Security

On ownership change: Warn on active user

Settings:

☑ Enable charting
☑ Enable multiple sandboxing
☐ Hard lock all widgets
☐ Spread On Rule Derived Consolidated Cells
☑ Enable slicing export
☑ Enable subset editor
☐ Enable cube import

Properties

Property	Value
Type	Application
Application Name	test1
Application Type	Approval
Created by	Admin
Deployed	No
Saved	Yes
Enable Charting	Yes
Enable Multiple Sandboxing	Yes
Hard Lock all Widgets	No
Spread on Rule Derived Consolidated Cells	No
Enable Slicing Export	Yes
Enable Subset Editor	Yes
Enable Cube Import	No
Enable Recalculate on Enter	Yes
Enable Saving CDD	Yes
Help Text	
Last Saved	May 23, 2016
Last Saved - User	Admin
Clients	TM1 Application Web(Default), Cognos Insight - Connected, Cognos Insight - Distributed
TM1 Server	ADMINIB-EH0HBNO:Vision2016_HOL1074
Advanced Modeling	No
Method to enforce rights	Cell Security

[3] http://www.ibm.com/support/knowledgecenter/SS9RXT_10.2.2/com.ibm.swg.ba.cognos.tm1_cont_ug.10.2.2.doc/c_tm1_enforcing_rights.html & ‘Management of TM1 Security: An Introduction’

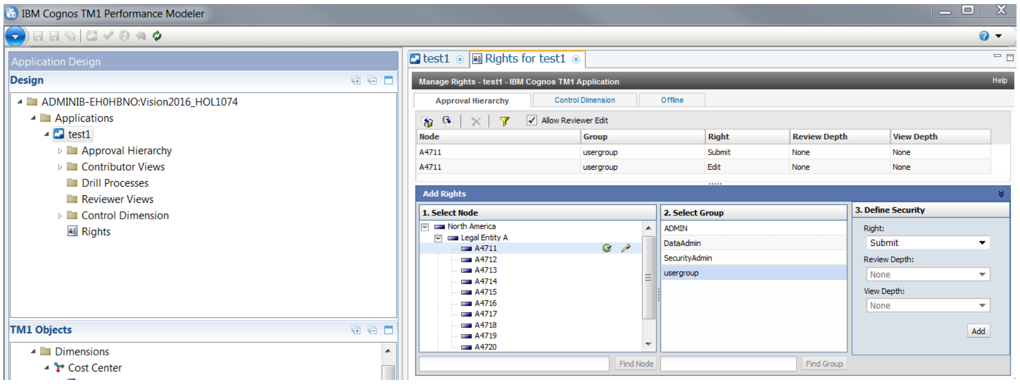
[4] See http://www.ibm.com/support/docview.wss?uid=swg21659476 for an overview and comparison to TM1 versions <10.2: Before TM1 version 10.2, a TM1 Applications approval hierarchy could not be shared across an application. Rights to view or edit a particular piece of data were enforced with element security on the approval hierarchy. In TM1 version 10.2, the architecture was changed so that rights may be enforced with cell security. This change meant the approval hierarchy dimension could be used in multiple applications. It also enabled multiple Applications to be deployed from the same cube.

4.2.2 CellSecurity

To share an Approval Hierarchy dimension across TM1 Applications, you need to use cell security to enforce rights. With cell security, a Control Dimension (such as a version dimension) is used to delineate the Applications. When Cell Security is used, the TM1 Application Server creates Cell Security cubes for all data cubes in the Application that contains the Approval Hierarchy dimension. If Cell Security cubes already exist, the TM1 Application Server extends their dimensionality to ensure that they include the Approval Hierarchy dimension and the Control Dimension if a control dimension is used.

Workflow Security Example:

i. Assigning & Applying Rights to the application (against Cube 'Operating Revenue And Expense FX'):

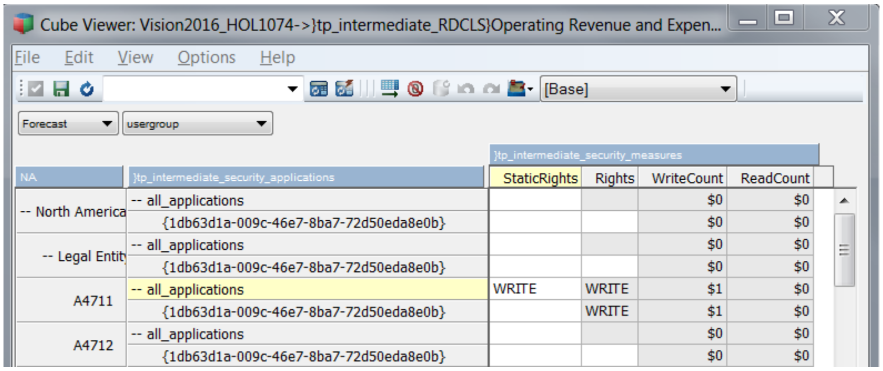


☑ Allow Reviewer Edit

Node	Group	Right	Review Depth	View Depth
A4711	usergroup	Submit	None	None
A4711	usergroup	Edit	None	None

Right: Submit
Review Depth: None
View Depth: None

ii. will result in the generation of a cube '}tp_intermediate_RDCLS}Operating Revenue And Expense FX' ('}tp_intermediate_RDCLS}<CubeName>') with corresponding security credentials in 'StaticRights' (in our example, Cost Center is the dimension with the approval hierarchy):


Cube Viewer: Vision2016_HOL1074->}tp_intermediate_RDCLS}Operating Revenue and Expen...
File
Edit
View
Options
Help
[Base]
Forecast
usergroup
}tp_intermediate_security_measures

NA	}tp_intermediate_security_applications	StaticRights	Rights	WriteCount	ReadCount
-- North America	-- all_applications			$0	$0
	{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}			$0	$0
-- Legal Entity	-- all_applications			$0	$0
	{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}			$0	$0
A4711	-- all_applications	WRITE	WRITE	$1	$0
	{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}		WRITE	$1	$0
A4712	-- all_applications			$0	$0
	{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}			$0	$0

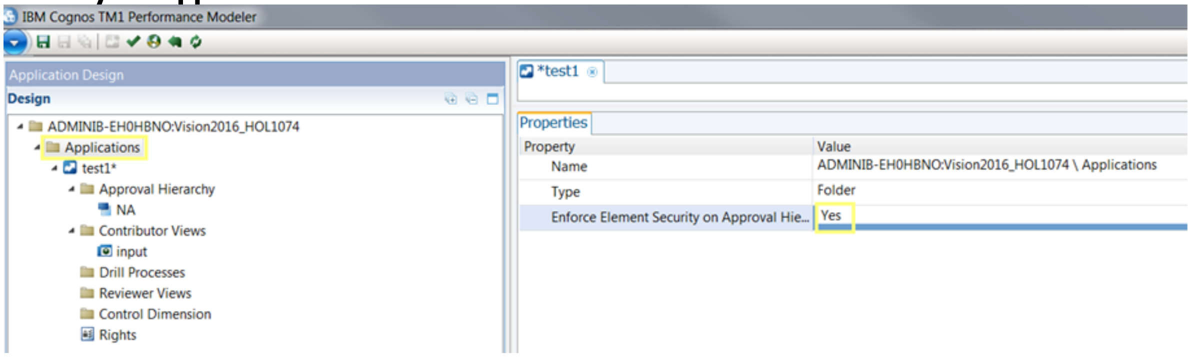
iii. which are referenced by the rule in the cell security cube '}CellSecurity_Operating Revenue And Expense FX' ('}CellSecurity_<CubeName>'):

```
#Region tp_planning_rule
[]=S:
IF(DB('}tp_intermediate_RDCLS}Operating Revenue and Expense FX', !Cost Center, !Version, !}Groups, 'all_applications', 'StaticRights' )@<>'',
DB('}tp_intermediate_RDCLS}Operating Revenue and Expense FX', !Cost Center, !Version, !}Groups, 'all_applications', 'StaticRights'),
CONTINUE);
#EndRegion
```

4.2.3 ElementSecurity

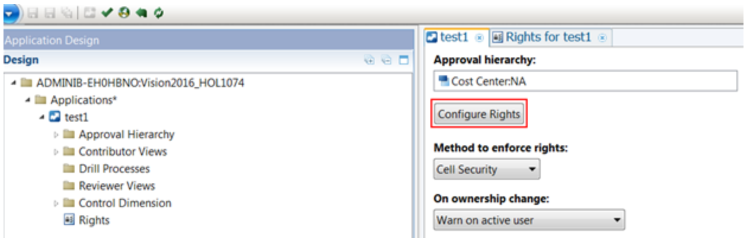
When rights are enforced with element security, the element security is populated on the Approval Hierarchy dimension using a TurboIntegrator process. Note that you cannot use a Control Dimension if Element Security is used to enforce the rights. The Control dimension is used to delineate security access across different applications using the same approval hierarchy via a CellSecurity rule that will reference a TM1 Applications Security object. If rights are enforced via ElementSecurity, this approach will not be used – Approval Workflow will be tied to Element Security and hence cannot be incorporating a control-dimension.

Furthermore, as of TM1 version 10.2.2 a TM1 Server-side setting is available to **Enforce Element Security on Approval Hierarchies**:



Property	Value
Name	ADMINIB-EH0HBNO:Vision2016_HOL1074 \ Applications
Type	Folder
Enforce Element Security on Approval Hie...	Yes

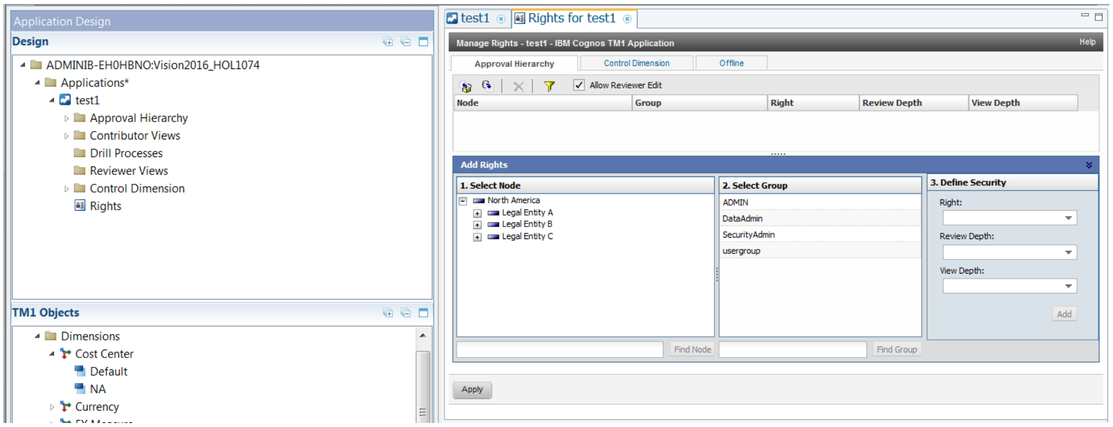
This parameter is a property of all the Approval or Responsibility Applications for a given TM1 server. **Enforce Element Security on Approval Hierarchies** defaults to **Yes** for both new and upgraded environments, hence ensuring Element Security consistency regardless of TM1 User Interface. Note that in the earlier releases 10.2 GA and 10.2 FP1, element security was not applied to the approval hierarchy dimension. In that case, if you used TM1 Architect, for example, you were able to see all the *elements* of the Approval Hierarchy in the subset editor, even though you can see the data for only the elements for which you have rights in the TM1 Application. The setting of this parameter is server-wide, i.e. the behavior is set for all applications, not by individual application. Note that the parameter **Enforce Element Security on Approval Hierarchies** with setting Yes forces element security to be aligned with security as defined by the 'Configure Rights' interface for the Approval or Responsibility Application:



Approval hierarchy: Cost Center:NA

Method to enforce rights: Cell Security

On ownership change: Warn on active user



Manage Rights - test1 - IBM Cognos TM1 Application

☑ Allow Reviewer Edit

Node	Group	Right	Review Depth	View Depth

1. Select Node: North America; Legal Entity A; Legal Entity B; Legal Entity C

2. Select Group: ADMIN; DataAdmin; SecurityAdmin; usergroup

3. Define Security: Right: / Review Depth: / View Depth:

It follows that if for an application, rights are enforced using Cell security, then Element Security is applied to the Approval Hierarchy dimension only if the **Enforce Element Security on Approval Hierarchies** option is set to Yes. When **Enforce Element Security on Approval Hierarchies** is yes, element security is applied using a rule that refers to a control cube maintained by the TM1 Application Server. This cube contains logic that computes the aggregate security across all Applications that use the same Approval Hierarchy dimension, and applies this security upstream to Elerment Security. In our example, the following rule would be placed into '}ElementSecurity_Cost Center.rux' (in our example, Cost Center is the dimension with the approval hierarchy):

```
#Region tp_planning_rule
[]=S:
 IF(DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost Center, !}Groups, 'all_applications', 'Rights')@<>'',
        DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost Center, !}Groups, 'all_applications', 'Rights' ),
        CONTINUE);
#EndRegion
```

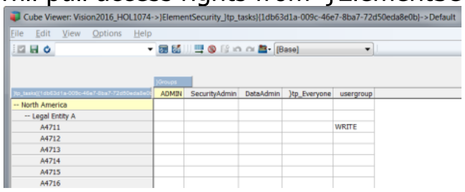
This rule means that wherever there is a 'Rights' entry in '}tp_intermediate_ElementSecurity}Cost Center' for a cost center and group, the rights from '}tp_intermediate_ElementSecurity}Cost Center' will be applied. Let's take a look at how '}tp_intermediate_ElementSecurity}Cost Center' manages/calculates 'Rights': We find that the below blue rules in }tp_intermediate_ElementSecurity}Cost Center determine the access for 'all applications' & 'rights'. As mentioned in the prior paragraph, they do so via computing the aggregate security across all Applications that use the same Approval Hierarchy dimension:

```
#Region rule_reference_shadow_element_security_{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}
['{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}','Rights']=S:IF(DB('}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-
72d50eda8e0b}',!Cost Center, !}Groups)@<>'',DB('}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}',!Cost Center,
!}Groups),CONTINUE);
#EndRegion
 ['WriteCount'] =
        IF(!}tp_intermediate_security_applications@<>'all_applications',
                IF(DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost Center, !}Groups,
                !}tp_intermediate_security_applications, 'Rights' ) @= 'WRITE',
                        1, 0 ),
                Continue);
['ReadCount'] =
        IF(!}tp_intermediate_security_applications@<>'all_applications',
                IF(DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost Center,!}Groups,
                !}tp_intermediate_security_applications, 'Rights' )@= 'READ',
                        1, 0 ),
                Continue);
['all_applications','WriteCount']=
        ConsolidateChildren('}tp_intermediate_security_applications');
['all_applications','ReadCount']=
        ConsolidateChildren('}tp_intermediate_security_applications');
['all_applications','Rights']=S:
        IF(DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost
        Center,!}Groups,!}tp_intermediate_security_applications,'WriteCount')>0, 'WRITE', CONTINUE);
['all_applications','Rights']=S:
        IF(DB('}tp_intermediate_ElementSecurity}Cost Center',!Cost
        Center,!}Groups,!}tp_intermediate_security_applications,'ReadCount')>0, 'READ', '');
```

The purple rules above are on the other hand referencing the element security on the approval hierarchy (the security set via 'Configure Rights'), i.e.

```
['{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}','Rights']=S:IF(DB('}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-
72d50eda8e0b}',!Cost Center, !}Groups)@<>'',DB('}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}',!Cost Center,
!}Groups),CONTINUE);
```

will pull access rights from '}ElementSecurity_}tp_tasks}{<TaskID>}.cub':



Cube Viewer: Vision2016_HOL1074->}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}->Default

}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0	ADMIN	SecurityAdmin	DataAdmin	}tp_Everyone	usergroup
-- North America					
-- Legal Entity A					
A4711					WRITE
A4712					
A4713					
A4714					
A4715					
A4716					

Note that because Element Security for the Cost Center dimension (*not* for the cost center approval hierarchy, which is in '}ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0b}' and configured via inputs in the Performance Modeler 'Configure Rights' UI) is driven using Rules, the TM1 Application Server must do a Security Refresh when the Rights are updated. This Security Refresh can take some time for a large TM1 Server and in environments with many users and/or security groups. Setting **Enforce Element Security on Approval Hierarchies** to Yes hence is only recommended for environments where

i. a security refresh is relatively quick (few users and/or groups & smaller models)

ii. or where an update of TM1 Applications security rights does not have to be performed during regular peak use hours

If neither (i) nor (ii) are granted & ElementSecurity on Approval Hierarchies needs to be enforced yet using Element Security to enforce rights is not applicable (because the approval hierarchy is to be used across multiple applications via use of a Control Dimension as a delineator), an option would be to synchronize }ElementSecurity with TM1 Applications security via a custom TI process which in our above example would process applicable security metadata from }ElementSecurity_}tp_tasks}{1db63d1a-009c-46e7-8ba7-72d50eda8e0b} and other }ElementSecurity_}tp_tasks}{<taskID>} to }ElementSecurity_Cost Center. This could be done by

i. Clearing security entries in '}ElementSecurity_Cost Center' for the workflow groups

ii. Copying data from applicable }ElementSecurity_}tp_tasks}{<taskID>} cubes: process one cube at a time and populate with values, only overwriting data from the processing of a prior cube if the element security is less restrictive. I.e. if }ElementSecurity_}tp_tasks}{<taskID1>} contained the entry WRITE and hence WRITE was processed to }ElementSecurity_Cost Center & if the value in }ElementSecurity_}tp_tasks}{<taskID2>} is READ, WRITE in }ElementSecurity_Cost Center is not to be overwritten with READ. Yet if }ElementSecurity_}tp_tasks}{<taskID1>} contained the entry READ and hence READ was processed to }ElementSecurity_Cost Center & if the value in }ElementSecurity_}tp_tasks}{<taskID2>} is WRITE, READ in }ElementSecurity_Cost Center is to be overwritten with WRITE. Via this logic, the }ElementSecurity_Cost Center will contain the aggregate of the security in the existing applications

4.2.4 Integrating (Non-)Approval Hierarchy Dimension Element Security into TM1 Applications Workflow

The IBM TechNotes
http://www.ibm.com/support/docview.wss?uid=swg21659499
& http://www.ibm.com/support/docview.wss?uid=swg21651554
describe how to enforce standard dimension Element Security restrictions within TM1 Applications. However, these TechNotes are written in the context of default TM1 security, where 'CELLSECURITYMOSTRESTRICTIVE' in }CubeSecurityProperties set to No or empty (default). With the default setting (= empty = No), CellSecurity may overrule ElementSecurity in both directions (i.e. allow overrule by making less or more restrictive). For example, Cell Security set to WRITE can override READ-level Element Security. This means that CellSecurity rules may have to take ElementSecurity restrictions into consideration. In the following example we are customizing the Workflow generated CellSecurity by a condition that will ensure that Read-Only accounts will always be READ only:

```
[] = S: IF(
        DB('}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups, 'all_applications',
        'StaticRights' )@<>'',
        If ( '}ElementSecurity_Account', !Account, !}Groups ) @='READ',
                'READ',
                DB( '}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups,
                'all_applications', 'StaticRights')),
        CONTINUE);
```

Or (modification of an auto-generated rule)

```
[] = S: IF(
            DB('}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups, 'all_applications',
            'StaticRights' )@<>'',
            If ( ‘}ElementSecurity_Account’, !Account, !}Groups ) @='READ',
                        'READ',
                        CONTINUE);
#Region <Description>
#Autogenerated ...
[] = S: IF(
            DB('}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups, 'all_applications',
            'StaticRights' )@<>'',
                        DB( '}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups,
                        'all_applications', 'StaticRights'),
                        CONTINUE);
#EndRegion
```

By setting the ‘CELLSECURITYMOSTRESTRICTIVE’ parameter to Yes, CellSecurity can only be more restrictive, i.e. READ Element Access can be overruled with NONE and WRITE Access with READ for example, but not the other way around. With ‘CELLSECURITYMOSTRESTRICTIVE’ set to Yes, combining Workflow with ElementSecurity can hence be simplified. With ‘CELLSECURITYMOSTRESTRICTIVE’ set to Yes for <CubeName>, the insertion of a rule or condition to check for READ-only access would not be necessary anymore, and the original rule could even be simplified to

```
[] = S:
  DB( '}tp_intermediate_RDCLS}<CubeName>', !ApprovalDim, !ControlDim , !}Groups, 'all_applications', 'StaticRights');
```

4.2.5 TM1 Workflow Management

There are three basic "layers" of security and workflow access control that are used by TM1 for restricting the data or cubes that a specific user can access: TM1 Security, Data Reservation, and Security Overlay.[5] The three layers of security may be implicitly used by ‘TM1 Applications’ (out-of-the-box workflow functionality as configured by TM1 Performance Modeller) yet can also be applied to devise ‘custom’ workflow (w/o the use of ‘TM1 Applications’):

TM1 Security:

Controls access rights. Is thereby the most fundamental layer. See Management of TM1 Security: An Introduction.

Data Reservation:

Controls (temporary) Ownership, i.e. who can write to a particular range of cells but applies only to specific users (not Groups) and is used to enforce the Ownership concept. See Using Data Reservations below for details. Data Reservation may only be used by user groups who have been granted the right to acquire and release data reservations

Security Overlay:

The Security Overlay is used to enforce the Submission concept to lock data. If a Security Overlay Cube is used (}SecurityOverlayGlobal_<CubeName>), the Security Overlay applies to *all* users in the TM1 server, i.e. a Security Overlay is not specified at the user or user group level. If one uses a CellSecurity rule to devise a security overlay, the Overlay may be defined at the group level.

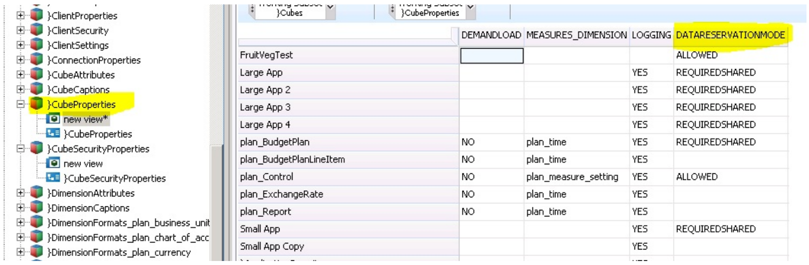
Keep in mind that Data Reservation or Security Overlay can never grant more permissive rights than TM1 security permits: they can only further constrain a user’s access**.** It follows that

[5] http://www.ibm.com/support/knowledgecenter/SS9RXT_10.2.2/com.ibm.swg.ba.cognos.tm1_cont_ug.10.2.2.doc/c_tm1_applications_security_rights_tm1_server.html

Security is the foundation. A Workflow implementation then – on top of the model's base TM1 Security model - leverages the functionalities of TM1 Security and/or Data Reservation and/or Security Overlay to (temporarily) facilitate workflow management and control.

The following table aligns the three basic layers with their corresponding security and workflow role and their typical use in TM1 applications as well as Custom Workflow:

Workflow Concept & Function	TM1 Server Functionalities	Details on functionalities, options and their use in TM1 Applications and Custom Workflow
Rights	TM1 Security via Element and/or Cell Security	TM1 Applications Workflow: When the administrator sets Rights for an Approval or Responsibility Application along the Approval Hierarchy and Control Dimension, these Rights are translated into either Element or Cell Security via security objects that are auto-generated, populated & applied to applicable Element and Cell Security. Non-workflow related (existing) Element and Cell Security may be impacted by TM1 Applications Workflow. To minimize the impact, it can be a good practice to separate workflow related groups and non-workflow groups. Also see section <Misc. Guidelines, Tips & Tricks> Custom Workflow: In a custom workflow regime, we recommend to use and build workflow-specific control cubes which may drive Element Security staging values and/or CellSecurity values. For Element Security, the use of an ElementSecurity_Staging_<DimensionName>.cub is recommended to allow for the use of workflow related security rules. The 'ruled' security access from the staging cube then is to be processed into the real }ElementSecurity_<DimensionName>.cub' via TI. In other words: 1.) Use a control cube to allow for workflow related security settings (such as applying a lock or opening a version/scenario etc.) 2.) Use a staging model for security (the cubes could be named }ElementSecurity_Staging_<DimensionName>.cub') to manage and model workflow related security and non-workflow related security. Note that you may and should use rules in the staging cubes: security will be applied to the real 'ElementSecurity_<DimensionName>.cub' cubes via Turbo-Integrator process, hence not requiring a security change to be followed by a 'SecurityRefresh' and thereby providing optimal performance. Note that the use of ElementSecurity_Staging cubes is also a proven practice for optimizing Security Management and Maintenance for regular non-workflow TM1 environments. Corresponding security management model accelerators are available per request.
Ownership	Data Reservation	Enabling and Enforcing Data Reservations: • When a cube is used in an Approval or Responsibility Workflow context, the **REQUIREDSHARED** mode of Data Reservation should be applied to the cube. This mode of Data Reservation requires that a user must have a Data Reservation before they can write to the cube. Note that the Data Reservation method in the **}CubeProperties** control cube applies to the whole cube. It follows that – with REQUIREDSHARED mode - even if a TM1 Application or a custom workflow solution is defined to work only on a particular slice of a cube (for example, a slice that is defined by a Control Dimension), a Data Reservation is required in order to write to *any* region of the cube. • **ALLOWED** mode on the other hand permits you to *optionally* take

<table>
<tr><th>Workflow Concept & Function</th><th>TM1 Server Functionalities</th><th>Details on functionalities, options and their use in TM1 Applications and Custom Workflow</th></tr>
<tr><td></td><td></td><td>Ownership if you want to have exclusive write access to all the cells in the scope of the Application.

}ClientProperties
}ClientSecurity
}ClientSettings
}ConnectionProperties
}CubeAttributes
}CubeCaptions
}CubeProperties
new view*
}CubeProperties
}CubeSecurityProperties
new view
}CubeSecurityProperties
}DimensionAttributes
}DimensionCaptions
}DimensionFormats_plan_business_unit
}DimensionFormats_plan_chart_of_acc
}DimensionFormats_plan_currency
}Cubes
}CubeProperties
<table>
<tr><th></th><th>DEMANDLOAD</th><th>MEASURES_DIMENSION</th><th>LOGGING</th><th>DATARESERVATIONMODE</th></tr>
<tr><td>FruitVegTest</td><td></td><td></td><td></td><td>ALLOWED</td></tr>
<tr><td>Large App</td><td></td><td></td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>Large App 2</td><td></td><td></td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>Large App 3</td><td></td><td></td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>Large App 4</td><td></td><td></td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>plan_BudgetPlan</td><td>NO</td><td>plan_time</td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>plan_BudgetPlanLineItem</td><td>NO</td><td>plan_time</td><td>YES</td><td></td></tr>
<tr><td>plan_Control</td><td>NO</td><td>plan_measure_setting</td><td>YES</td><td>ALLOWED</td></tr>
<tr><td>plan_ExchangeRate</td><td>NO</td><td>plan_time</td><td>YES</td><td></td></tr>
<tr><td>plan_Report</td><td>NO</td><td>plan_time</td><td>YES</td><td></td></tr>
<tr><td>Small App</td><td></td><td></td><td>YES</td><td>REQUIREDSHARED</td></tr>
<tr><td>Small App Copy</td><td></td><td></td><td>YES</td><td></td></tr>
</table>

<u>Built-in TM1 Applications Workflow:</u>
• For workflow types Approval and Responsibility, the <b>REQUIREDSHARED</b> mode will be set for the corresponding cube(s). The TM1 Application Server grants a Data Reservation to a user who takes Ownership of an Approval Hierarchy node or set of Nodes. A Data Reservation is specific to a particular User, not a Group. Only one user can have Ownership of a leaf node at any time. The Data Reservation granted by the TM1 Application Server is scoped to the relevant Approval Hierarchy nodes. If a Control Dimension is used, the Data Reservation is scoped to the writeable Control Dimension slices for the Application.
• TM1 Applications will use the <b>ALLOWED</b> mode of Data Reservation for ‘Central Applications’. Users in a Central application are able to write by default without taking Ownership subject to normal TM1 security.

<u>Custom Workflow</u>:
• In a custom workflow regime, the requirement for a user to take exclusive (temporary) ownership of one or more nodes or slice(s) should be addressed using data reservation.</td></tr>
<tr><td>Submit</td><td>Security Overlay or CellSecurity rule</td><td><b>Security Overlay</b>: Locking of data via a Security Overlay cube. Security overlay always applies to all users and groups (the security overlay cube doe s not include the }Groups dimension).

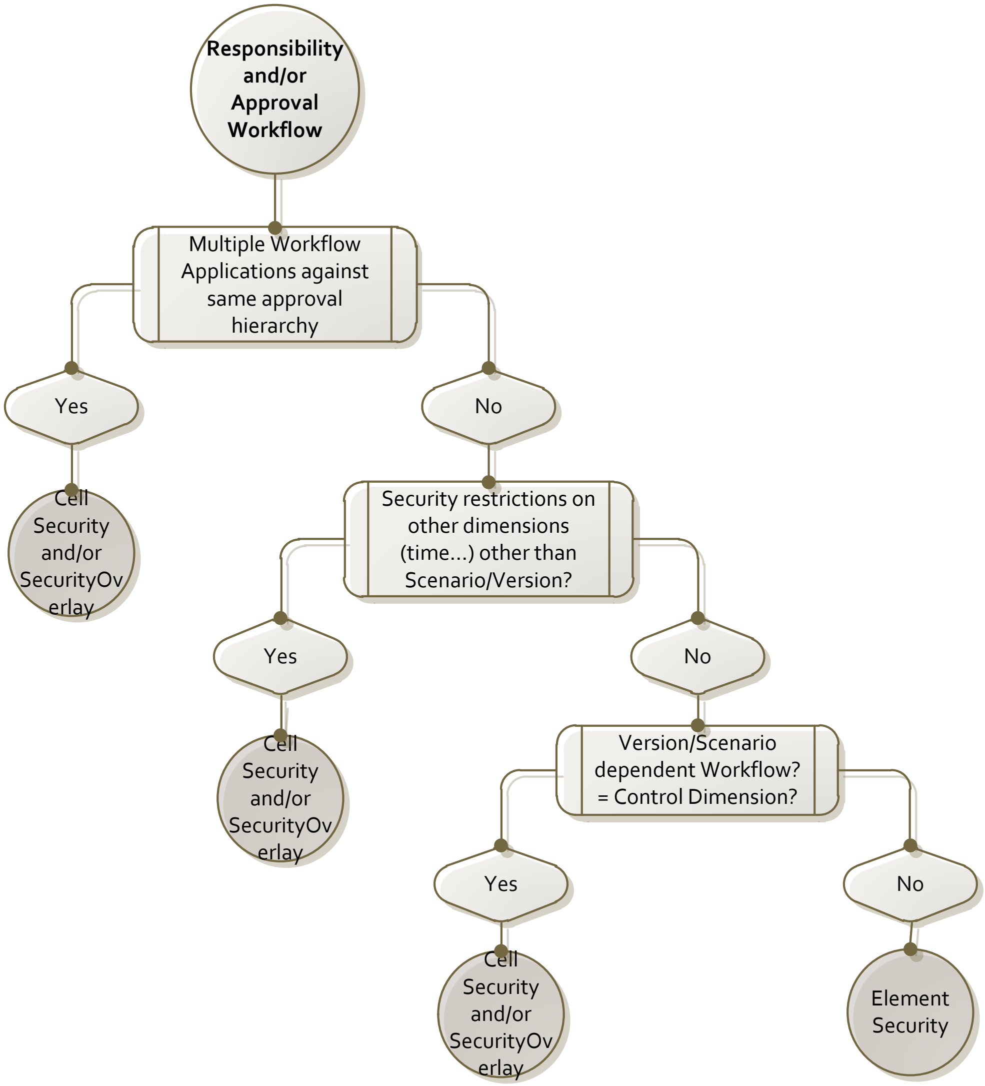
<b>CellSecurity rule</b>: a logical ‘Security Overlay’ can also be accomplished via a CellSecurity rule that – depending on a condition on a lookup cube - sets access to READ.

<u>Built-in TM1 Applications Workflow</u>: Use of Security Overlay Cubes
• The action of Submitting a node applies only to Approval applications.
• When a node is submitted, the slice of data that is identified by the Approval Hierarchy node and Control Dimension is locked via an entry in a Security Overlay Cube, preventing any further data entry.

<u>Custom Workflow</u>: Use of Security Overlay Cubes or Cell Security
• Security overlay may be used to submit a node (and prevent further inputs) or to ‘lock’ entire sections of a cube for workflow control purposes.
• Notes:
○ a custom workflow regime can often just as well leverage CellSecurity</td></tr>
</table>

Workflow Concept & Function	TM1 Server Functionalities	Details on functionalities, options and their use in TM1 Applications and Custom Workflow
		to achieve the same result (by creating a CellSecurity condition/rule that overrules other CellSecurity rules, i.e. creating an overlay via a rule). o advantage of using a Security Overlay Cube: always applies to all users and groups (a Security Overlay Cube does not include Security Groups). o disadvantage: adds yet another layer to security and hence can lead to additional complexity => If CellSecurity is in place and requires a similar dimensionality as SecurityOverlay, and if a Security Overlay can be achieved via CellSecurity rules , and if the # of user groups is not too small, it is an alternate practice to use CellSecurity (to reduce complexity for security management and auditing).

4.3 Workflow Cube Security Decision Support Matrix


Responsibility and/or Approval Workflow
Multiple Workflow Applications against same approval hierarchy
Yes
Cell Security and/or SecurityOverlay
No
Security restrictions on other dimensions (time...) other than Scenario/Version?
Yes
Cell Security and/or SecurityOverlay
No
Version/Scenario dependent Workflow? = Control Dimension?
Yes
Cell Security and/or SecurityOverlay
No
Element Security

5. Misc. Guidelines, Tips & Tricks

- Security should never be the afterthought of an implementation. The same applies to workflow management & control

- Security should be an integral consideration when designing the model:
  - Discuss & determine how the model is to be secured.
  - Determine how you want to manage security (adding users, groups, assigning rights to groups, changing permissions for groups, applying security when making hierarchy changes or adding hierarchies).
  - See Management of TM1 Security: An Introduction on how proven practices around TM1 security and the benefits of a security management model.
  - Evaluate how you could automate the retrieval and processing of security metadata in TM1. Can Cost Center, Legal Entity, etc. element/member security be retrieved from a DW? Can it be extracted from an LDAP directory?
  - Attempt to automate the processing of security wherever possible. This is especially important in volatile environments (larger number of users and groups, and frequently changing metadata and master data). Without automation, a 'security administrator' will have to spend countless hours managing security changes, which should be avoided.

- Start with Security. Once you have determined the desired security regime (what to secure and how and how to manage it), then look at Workflow. In other words, Security first and Workflow second:
  - Workflow should only ever be as complex & comprehensive as it needs to be.
  - Workflow hence should adhere to existing or new (yet proven) internal processes.
  - Design the simplest workflow you need. Do not over-engineer the workflow just because you can. Here's why: Workflow is meant to support a business process by guiding users (planners, analysts, managers) along the planning & analysis processes and by providing checks & balances along the way (submissions, approvals, rejections etc.). In its business support function, workflow shall never be making a proper process more complex or cumbersome that it needs to be (this can easily occur when workflow is designed to enforce too many unnecessary approval steps, i.e. an approval hierarchy with too many levels).
  - If in doubt, go live with the simpler workflow. It is better to start out simple.
  - Typically, the more contributors a particular application has, the more useful it becomes to work with Data Reservations.

- Always test Workflow security against input templates, the cube itself, CAFE and Perspectives. Test if Element Security is properly enforced both when viewing Cube Data and when using TM1 Applications (where applicable) (see prior notes on how CellSecurity could be configured to overrule ElementSecurity).

- As illustrated in some of the examples above, the security objects auto-generated & applied by Performance Modeller's TM1 Applications can add significant complexity to a security/workflow regime. Implementation of TM1 Applications workflow should hence always go along with a thorough (re)-test of a) TM1 Security and (b) TM1 Security with TM1 Workflow. Effects of Workflow design on TM1 Security should be properly documented. It can be a good practice to use dedicated security groups for workflow (and to ensure via audit controls that this practice is followed). This way, both workflow and non-workflow related security can be maintained, tested & audited easier.

- Do not shy away from custom workflow. Particularly for very large approval hierarchies and/or more complex workflow procedures (which may involve intra- or inter- model or other procedural dependencies), a custom workflow can be simpler to build, maintain and operate.

- Do not exclude using one or more entirely separate dimension for the Approval Hierarchy/Hierarchies. Workflow (Element) Security may at times be in conflict with non-workflow-related security. By using a separate dimension with the approval hierarchy, workflow and non-workflow security can at times be separated more clearly and may be easier to manage. With this approach, one can also use Element Security to enforce Workflow security, hence not requiring security refresh when changing rights and thereby providing fast performance even in large-scale environments. In other words: using ‘contribution’-specific dimensions as approval hierarchies can at times be a good method to separate ‘regular’ dimension element security from the workflow related implementation, hence simplifying security (higher transparency due to fewer overlay objects) and possibly achieving faster performance. Note that such ‘dependent’ approval hierarchy dimensions can be created and updated automatically following updates to the ‘master’ dimension.

- Note the consequences of changing the Default Cell Security to NONE (in the }CubeSecurityProperties cube): By Default, an empty value in a CellSecurity cube will result in access credentials according to Cube, Dimension & Element Security (and potential security overlays). If this default is changed to NONE, CellSecurity access needs to be defined for each applicable cell. A combination of CellSecurity Default = NONE and CELLSECURITYMOSTRESTRICTIVE = NO or empty for example would hence result in CellSecurity rules requiring many conditional checks against ElementSecurity credentials.

- TM1 provides the ability to define security at the most granular level plus provides multiple means to manage security, temporary security changes (overlays for example) and workflow related security (data reservations). In addition, different data reservation behaviours can be defined (in }CubeProperties) and the behaviour of security for individual cubes can be changed additionally (in }CubeSecurityProperties). While these many security and workflow capabilities of TM1 provide the ability to implement any type of security schema and workflow control, the capabilities can however become confusing if not applied consistently. => If and where possible, it is good practice to be and remain consistent in how security and workflow is applied. In other words: if and where possible, avoid using different default security configuration settings between cubes that otherwise share secured dimensions and elements (ElementSecurity).