Contents

[Introduction 2](#_Toc173277294)

[TM1 Applications 2](#_Toc173277295)

[Model Based Workflow 2](#_Toc173277296)

[End of Support for TM1 Applications 3](#_Toc173277297)

[Transition to Plans in Planning Analytics Workspace 3](#_Toc173277298)

[Extended Support 3](#_Toc173277299)

[Transition to Model Based Workflow 3](#_Toc173277300)

[TM1 Application Capabilities 4](#_Toc173277301)

[Approval Dimension 4](#_Toc173277302)

[Control Dimension 4](#_Toc173277303)

[Private Objects 5](#_Toc173277304)

[Model Security 5](#_Toc173277305)

[Locking 5](#_Toc173277306)

[Sandboxes 6](#_Toc173277307)

[Data Reservation 6](#_Toc173277308)

[Workflow Module 6](#_Toc173277309)

[Getting Started 6](#_Toc173277310)

[Workflow Module Cubes 7](#_Toc173277311)

[Workflow Module Dimensions 7](#_Toc173277312)

[Workflow Module Picklists 8](#_Toc173277313)

[Workflow Module Subsets 8](#_Toc173277314)

[Workflow Module Processes 9](#_Toc173277315)

[}WF\_Setup 9](#_Toc173277316)

[}WF\_CreateWorkflow 10](#_Toc173277317)

[}WF\_AssignNode 12](#_Toc173277318)

[}WF\_UpdateNodeState 12](#_Toc173277319)

[}WF\_LockUnlockWorkflow 13](#_Toc173277320)

[}WF\_DeleteWorkflow 13](#_Toc173277321)

[}WF\_RemoveNodeAssignment 14](#_Toc173277322)

[}WF\_RemoveAllNodeAssignments 14](#_Toc173277323)

[}WF\_ResetWorkflow 14](#_Toc173277324)

[Design Notes 14](#_Toc173277325)

# Introduction

This module is intended for IBM Planning Analytics customers that are currently using TM1 Applications to support workflow and Planning Analytics customers that do not currently implement workflow solutions. Workflow in the context of TM1 Applications and this module is used to provide context to how, when, and where users must contribute (input) and review data in cubes in a TM1 model. As support for TM1 Applications comes to an end, Planning Analytics customers should consider how workflow capabilities should be implemented in the modern Planning Analytics software stack.

Workflow in IBM Planning Analytics has traditionally been implemented with TM1 Applications or directly in the TM1 model. In either case a number of TM1 model objects and processes are involved to support the workflow.

## TM1 Applications

Also commonly referred to as TM1 Contributor, TM1 Applications is packaged with IBM Planning Analytics 2.0.9 and previous releases. TM1 Applications supports the deployment of workflow applications to a web portal (pmpsvc). The deployed applications leverage a combination of TM1 native cube views and Websheets, presented through TM1 Web (sometimes referred to as TM1 Application Web) to enable contribution and review of data. TM1 Applications simplifies and accelerates the process of deploying and managing workflow, but is rigid in its implementation. Workflow in TM1 Applications requires a single dimension to be used for responsibility assignment and forces a specific pattern of element or cell security that may not be compatible with the greater security requirements of the TM1 model. TM1 Applications was not universally adopted by IBM Planning Analytics customers with workflow requirements.

## Model Based Workflow

In many cases, workflow was implemented directly in the TM1 model used cubes to track the state of data being contributed, determine users access to slices of data, and lock and unlock slices of data based on current state of the workflow. The advantage of workflow being built into the TM1 model is that it can be customized based on the specific needs of the business. Workflow processes in IBM Planning Analytics should reflect the real business process of workflow. Model based workflow will often use only some of the same technical features and techniques as found in TM1 Applications. The disadvantage of model-based workflow is that it's custom built, which costs both time and money. Authoring Turbo Integrator processes, cube rules, and defining how workflow should be tracked in the model with dimensions and cubes is a complex task and requires familiarity with TM1 model development.

# End of Support for TM1 Applications

IBM Planning Analytics 2.0.9 will reach the end of mainstream support on April 30, 2025. The IBM Planning Analytics 2.1 release is available as an upgrade path. However, the 2.1 release does not include a number of legacy components such as Architect, TM1 Perspectives, TM1 Applications and Performance Modeler (see: <https://www.ibm.com/docs/en/planning-analytics/2.1.0?topic=configuration-upgrading-planning-analytics-local-21>). Planning Analytics customers currently using TM1 Applications have a few options to consider before the end of support.

## Transition to Plans in Planning Analytics Workspace

The modern web front of IBM Planning Analytics is Planning Analytics Workspace (commonly referred to as PAW or just Workspace). Workspace provides workflow functionality through the Plans feature (<https://www.ibm.com/docs/en/planning-analytics/2.1.0?topic=work-plans>). Plans is part of the Plans and Applications area of Workspace. Plans provides workflow functionality while Applications in Workspace act as a table of contents that provides simplified access to assets that include Workspace books, Workspace views, and Websheets. Support for Websheets in Workspace Plans is important because it means new interfaces do not need to be authored to support contribution (existing Websheets from TM1 Applications can be reused). The implementation of Plans in Workspace does not provide a direct migration from TM1 Applications. While Plans uses some of the same concepts as TM1 Applications, it also introduces new concepts such as tasks with dependencies on other tasks. Plans in Workspace was developed to be business owned and provides a framework of capabilities that can be selected to better represent real world business processes.

The Plans feature in Workspace has a dedicated roadmap that includes a mix of net new capabilities and integration of capabilities found in TM1 Applications.

## Extended Support

Extended Support is available for IBM Planning Analytics customers that want to continue support for TM1 Applications in the 2.0.9 release past April 30th, 2025. Extended Support is a paid option that provides access to IBM support. It is important to note that Extended Support does not include product enhancements. TM1 Applications has only received basic maintenance since 2016. IBM has no roadmap for TM1 Applications as all new workflow functionality is planned for the Workspace Plans and Applications feature. Requests for defect corrections are possible with Extended Support.

## Transition to Model Based Workflow

Transitioning to model-based workflow is the focus of the remainder of this document. Using a set of standard Turbo Integrator processes we can rapidly deploy a model-based workflow that applies the same concepts as TM1 Applications. This avoids the complex and time consuming task of developing cubes, dimensions, processes, and other TM1 objects to support workflow. This option does not require an additional IBM Planning Analytics licensing as it only relies on adding objects to existing TM1 models.

Model based workflow requires a user interface to manage workflows and allow contribution and review of data. Existing Websheets and cube views for contribution and review of data may be reused. User interfaces for management of model-based workflows can be authored using Workspace Books, Websheets, or a combination of the two. Note that Websheets can be used in TM1Web (Planning Analytics 2.1 includes TM1Web) and Planning Analytics Workspace.

A sample Websheet is included with this module to create and delete workflows, and assign user or groups to the approval dimensions. Workspace Books and Websheets use action buttons to execute the workflow processes provided in this module.

# TM1 Application Capabilities

Before making the transition from TM1 Application to model-based workflow it will help to understand exactly what features of the TM1 database were implemented by TM1 Applications. The workflow module does not replicate all functionality from TM1 Applications. The following list describes TM1 functionality that is used in the implementation of TM1 Applications.

## Approval Dimension

Also known as the responsibility dimension, members in this dimension are used to define unique slices of data to assign to user groups in the TM1 database. The members in the approval dimension are commonly referred to as nodes. This dimension must be selected as the workflow of being defined. A subset from the approval dimension is also selected to determine which alias is used. TM1 Applications validates this dimension has a simple structure. Multiple parents are not supported and there must be only a single root (top level) member). The dimension must represent a single consolidation path (rollup). The process to assign TM1 groups to be responsible for one or more members in the approval dimension is known as 'rights assignment'.

## Control Dimension

Members in the control dimension are used to determine unique slices of data in each application when multiple applications share an approval dimension and cubes. The members in the control dimension are used when the application is used to define slices where security and locking mechanisms are applied. A fundamental concept in TM1 Applications is slices of data must be unique to each application. The assumption that TM1 Applications makes is that once a slice of data is defined for use in the application, it should not be accessible in another application or elsewhere outside of the current application.

## Private Objects

TM1 database allows for public and private versions of cube views and subsets. Like sandboxes, private objects can only be viewed and manipulated by the owner of the private object. Private objects can not be shared between users. TM1 Applications generates private version of any views used in the workflow for the user when taking ownership of a node. The private view could be manipulated without impacting other users that were using the same view.

The workflow module is primarily based on Turbo Integrator processes. Turbo Integrator does not include functions to create or manipulate private views.

## Model Security

TM1 Applications automatically applies security in the TM1 model. If multiple applications are being deployed, where cubes are shared between applications then a cell security cube is created for each of these cubes. Both the members in the control dimension and approval dimension are used to assign cell security. TM1 Applications uses a set of cube rules on the cell security cube that it creates to automatically provide WRITE access to slices of data based on rights assignments. If a control dimension is not used because cubes are not shared between applications, then the applications can be deployed with element security (cell security is optional). An element security cube is created for the approval dimension. TM1 Applications also uses rules on the element security cube to ensure that groups that have been assigned rights have WRITE access to the members in the approval dimension they have rights assignment on.

The automatic generation of security cubes in the TM1 model may be seen as an advantage or disadvantage of TM1 Applications. The security objects generated by TM1 Applications is not comprehensive and security on members in other dimensions in the TM1 model may prevent WRITE access to slices of data. The implementation of security in TM1 Applications using rules also makes it challenging to override security generated by TM1 Applications.

TM1 Applications also requires the TM1 developer create user groups in the model that can be used for rights assignment. In many cases these groups need to be created specifically to reflect the required assignments against the approval dimension.

## Locking

TM1 Applications uses a security feature from TM1 named SecurityOverlayGlobal. This security feature is unique in TM1 in that it does not require groups. Like cell security, SecurityOverlayGlobal is implemented with a control cube associated to the model cube it's used with. Like }CellSecurity cubes the }SecurityOverlayGlobal cubes only need to include one or more dimensions from the model cube it is used to lock.

Instead of using the }Groups dimension the }SecurityOverlayGlobal cubes uses a dimension called }SecurityOverlay, which contains a single member. Intersections in the }SecurityOverlayGlobal cubes contain a 1 if the slice in the model cube represented by the intersection should be locked (0 or empty if unlocked). Locking a slice will only ever decrease access from WRITE to READ. If a user does not already have WRITE access to cells within the slice, then SecurityOverlayGlobal will have no net impact on access to those cells.

## Sandboxes

TM1 Applications uses TM1's sandbox feature in two different ways. A sandbox is automatically created for each user that opens a node. This automatic sandbox is somewhat abstracted in the TM1 Applications Web user interface. Users may or may not understand that they are working in a sandbox as the data they enter appears in blue. TM1 Applications includes both a Commit and Submit button. Both the Submit and Commit button act to commit data from the sandbox to the base. The submit button also locks the slice of data the user was contributing data to.

The Workflow processes provided in this module do not interact with Sandboxes. The implementation of any interfaces (Planning Analytics Workspace Books or Websheets) that use the Workflow processes may independently use the sandboxes feature.

## Data Reservation

Data Reservation is a powerful but complex feature in TM1. It allows for assignments of slices to specific users. Unlike all other security measures in TM1 (except global security overlays), groups are not used. Data Reservation is what supports the concept of 'bouncing' a user in TM1 Applications such that only one user may own a slice at the same time.

The Data Reservation feature in TM1 v11 does not support hierarchies and was deprecated in TM1 v12. The Workflow processes do not use the Data Reservation feature to support forward compatibility with TM1 v12.

# Workflow Module

The following list is a description of the workflow Turbo Integrator processes that are included with this module. The purpose and operation of each process is described. The set of processes is self-contained and will generate all other required objects in the TM1 model such as dimensions, subsets, cubes, and views that are used to support workflow capabilities.

## Getting Started

The workflow processes must be added to the TM1 model. With TM1 v11 the set of .pro files can be placed in the TM1 database directory and will be added to the model on next startup of the TM1 databases. The workflow processes can also be pulled into the TM1 database from a GIT repository (zip file containing the repo is included).

Once the workflow processes have been pulled into the TM1 database we have the }WF\_Setup process. This process will create all of the cubes, dimensions, subsets, and picklist cells as described below.

## Workflow Module Cubes

The following cubes are created when running the }WF\_Setup process. These cubes are required in the TM1 model before workflows can be deployed using the process }WF\_CreateWorkflow.

|  |  |  |
| --- | --- | --- |
| Cube Name | Dimensions | Description |
| }WF\_Control | }WF  }WF\_Control | This cube tracks data for each deployed workflow. |
| }WF\_Cubes | }Cubes  }WF | Used to track the contribution cubes used in each workflow. This cube does not contain workflow specific state data. It is only used in the workflow processes to assist with validation when multiple workflows contain the same cubes. |
| }WF\_Errors | }WF\_Error\_Lines  }WF\_Errors | This cube stores errors that are generated by the workflow processes, for all workflows. It includes workflows that fail to deploy. |
| }WF\_Picklists | }WF\_Picklists  }WF\_PickValue | Used to provide picklists that can be used in the workflow creation and management user interfaces. See description of picklist cells in Worklow Module Picklists documentation. |
| }Picklist\_}WF\_Picklists | }WF\_Picklists  }WF\_PickValue  }Picklists | Picklist cube for the }WF\_Picklist cubes. Used to define picklist content with rules. |

## Workflow Module Dimensions

The following dimensions are created when running the }WF\_Setup process. These dimensions are required in the TM1 model before workflows can be deployed using the }WF\_CreateWorkflow process.

|  |  |
| --- | --- |
| Dimension Name | Description |
| }WF\_Control | Members of this dimension are used to track details of each of the deployed workflow. This includes the name of the workflow, approval dimension, control dimension, details about the security and locking implementation. The comment index member is used to track the current position in the workflow specific cube that stores comments. |
| }WF | A member will be added to this dimension to track each deployed workflow |
| }WF\_State | Details about the state of the nodes (member in the approval dimension) for each deployed workflow. Includes the contributor and reviewer user (not group), state of the node, and last updated date of the node |
| }WF\_Comments | Used for tracking comments that are added when node state changes. For example, if the node state changes from open to submitted the user submitting may add a comment to explain the submission. The node state is logged along with the comment. |
| }WF\_Comment\_Lines | List of lines to track comments. New lines are added automatically as they are needed. There is no need to manually add members to this dimension. |
| }WF\_Errors | Used to track errors that might be identified when running any of the workflow processes. |
| }WF\_Error\_Lines | List of lines to track errors. New lines are added automatically as they are needed. There is no need to manually add members to this dimension. |
| }WF\_PickValue | Dimension with a single member to support string picklists. |
| }WF\_Picklists | Members represent picklist that can be used when authoring user interfaces in Workspace books or websheets. |

## Workflow Module Picklists

The following picklists are available in the }WF\_Picklists cube. These picklists cells are not directly referenced by the code in the Workflow processes. These picklists may be used in Planning Analytics Workspace Books or Websheets that are used to create, delete, and manage workflows.

|  |  |
| --- | --- |
| Picklist Name | Description |
| Approval Dimension | List of non-control dimensions that can be selected as the Approval dimension. |
| Approval Subset | List of available public subsets in the selected Approval dimension (dimension selected in the Approval dimension picklist cell). |
| Control Dimension | List of non-control dimensions that can be selected as the Control dimension. Cannot be the same dimension as the Approval dimension. |
| Control Member | List of members from the selected Control dimension (dimension selected) |
| Security | 'Cell', 'Element', or 'None'. This selection can be passed to the <pSecurity> parameter in the }WF\_CreateWorkflow process. |
| Security Mode | 'Client' or 'Group'. This selection can be passed to the <pSecurityMode> parameter in the }WF\_CreateWorkflow process. |
| Locking | 'Y' or 'N'. This selection can be passed to the <pLocking> parameter in the }WF\_CreateWorkflow process. |

## Workflow Module Subsets

The }WF\_Setup process generates a number of dynamic subsets that can be used in the workflow administration experience. These subsets are not used by the other workflow processes.

|  |  |  |
| --- | --- | --- |
| Subset Name | Dimension | Description |
| WF\_ApprovalDimensions | }Dimensions | Dynamic subset to list all model dimensions. Does not display hierarchies. |
| WF\_ContributionCubes | }Cubes | Dynamic subset to list all model cubes. |
| WF\_SecurityOverlayGlobalCubes | }Cubes | Dynamic subset to list all cubes in the model that have a corresponding SecurityOverlayGlobal cube. Can be used in the workflow management experience to track which cubes are used in workflows with locking enabled. |
| WF\_CellSecurityCubes | }Cubes | Dynamic subset to list all cubes in the model that have a corresponding Cell Security cube. Can be used in the workflow management experience to track which cubes are used in workflows with Cell based security. |

## Workflow Module Processes

### }WF\_Setup

This is the first process that must be run once the workflow processes have been added to the TM1 model. }WF\_Setup generates several dimensions, subsets, and cubes that are used to track the workflows that are deployed.

}WF\_Setup has two parameters:

|  |  |
| --- | --- |
| Name | Description |
| pReset | When set to 'Y' it will remove all workflows and regenerate all objects in the model used to track workflows. This should only be set to 'Y' when testing the workflow processes. |
| pWorkflowGroupPrefix | The Workflow processes will generate groups in the TM1 model if the <pSecurityMode> parameter is set to 'Client' when running the }WF\_CreateWorkflow process. This prefix will be added to any groups created by the workflow processes making it easy to identify which groups were generated by workflow. |

Subsets created by the }WF\_Setup process:

|  |  |  |
| --- | --- | --- |
| Name | Dimension | Description |
| }WF\_ApprovalDimensions | }Dimensions | List of dimensions that can be selected as Approval dimensions. This is a dynamic list that filters out control dimensions and members in the }Dimensions dimension that represent hierarchies. |
| }WF\_ContributionCubes | }Cubes | List of cubes that can be selected as input cubes for applications (dynamic list of model cubes only). |
| SecurityOverlayGlobalCubes | }Cubes | List of }SecurityOverlayGlobal cubes in the TM1 model. |

### }WF\_CreateWorkflow

This process is used to define and deploy a new workflow. Most of the Turbo Integrator code in the process is used to validate the selection of the Approval and Control dimensions, and selection of contribution cubes for the workflow being deployed is consistent with existing workflows. There are a few basic rules to consider when deploying a workflow:

1. Workflow names must be unique. Two or more workflows may not have the same name.
2. When two or more workflows contain the same contribution cube(s), unique slices in the cube(s) must be identified for each workflow. A unique slice cannot be shared between workflows.
3. When two or more workflows contain the same contribution cube(s) cell security must be used.
4. When two or more workflows contain the same contribution cube(s) a control dimension must be identified. A unique member in the control dimension must be assigned to each workflow.
5. All workflows that share contribution cubes must use the same Approval dimension and Control dimension.

Parameters in the }WF\_CreateWorkflow process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the new workflow. Must not match an existing workflow unless the <pReplace> parameter is set to 'Y'. |
| pDimension | Approval dimension |
| pHierarchy | Hierarchy to use from the Approval dimension |
| pSubset | Subset to use from the Approval dimension. This subset is used throughout the workflow processes to make sure the same alias is used in any views that are presented to users. |
| pCubes | Comma delimited list of cubes used for contribution in the workflow. |
| pReplace | Must be set to 'Y' if an existing workflow is being replaced with a workflow of the same name |
| pSecurity | Determines the security mechanism that will be applied. Options are None, Element, or Cell. Cell must be selected when any contribution cube is used in two or more workflows. |
| pLocking | There are two locking options, 'Any' and 'All'. When set to 'Any', a single contributor user that requests the node to be locked will lock the node. When set to 'All', all of the current contributors must state intent to lock or unlock before the node is actually locked or unlocked using GlobalSecurityOverlay. |
| pControlDimension | Control dimension. Required when multiple workflows will share cubes. |
| pControlHierarchy | Hierarchy in the Control dimension |
| pControlMember | Member in the Control dimension used to identify unique slices for the workflow being deployed. |

Cubes created by the process }WF\_CreateWorkflow:

|  |  |  |
| --- | --- | --- |
| Name | Dimensions | Description |
| }WF\_<pWorkflow>\_State | Approval dim.  }Clients/}Groups  }WF\_State | Cubes used to track the state of nodes in the workflow. <pWorkflow> stands for the name of the workflow. <pDimension> stands for the Approval dimension. |
| }WF\_<pWorkflow>\_Comments | }WF\_Comment\_Lines  }WF\_Comments | Cubes used to store comments that are added by contributors and reviewers as the state of the nodes change. <pWorkflow> stands for the name of the workflow. |
| }ElementSecurity\_<pDimension> | Approval dim.  }Groups | Created only when pSecurity=Element. Used to provide WRITE access to groups that contain the contributor and reviewer users. <pDimension> stands for the Approval dimension. |
| }CellSecurity\_<pCube> | Approval and Control dimensions | Created only when pSecurity=Cell. Used to provide WRITE access to groups that contain the contributor and reviewer users. Cell Security cubes are created only with the approval and control dimensions. <pCube> stands for a list of contribution cubes.  Cell security is documented here: <https://www.ibm.com/docs/en/planning-analytics/2.0.0?topic=scc-cellsecurity-cubename> |
| }SecurityOverlayGlobal\_<pCube> | Approval and Control dimensions | Created only when pLocking=Y. SecurityOverlayGlobal cubes are used to lock and unlock slices of data in the contribution cubes. SecurityOverlayGlobal are created with only the Approval and Control dimensions. <pCube> stands for a list of contribution cubes.  Documentation about SecurityOverlayGlobal is found here <https://www.ibm.com/docs/en/planning-analytics/2.0.0?topic=developers-security-overlay> |

### }WF\_AssignNode

This process is used to assign nodes (members from the Approval dimension) to specific users in the TM1 model. This process does the equivalent of the rights assignment feature in TM1 Applications.

There is a very significant difference between rights assignments in TM1 Applications and the }WF\_AssignNode process. Rights assignments uses existing TM1 user groups to assign nodes. The }WF\_AssignNode process will create groups and assign both the contributor and review user to the groups. When an assignment is made at a consolidated level the group will be generated for consolidated level and used to assign security at the consolidated level and for all the descendant members below the consolidation.

The }WF\_AssignNode process can be used to add multiple contributors and reviewers to the same node. The contributor and reviewer users are paired, but multiple contributors can share the same reviewer. All contributors and reviewers for a node are added to the same user groups to provide WRITE access to the node through element or cell security.

Node assignments can happen at the consolidated or leaf level. Assigning a contributor to a consolidated node will also assign all descendant members to the contributor.

Assigning a contributor to a node will not impact the lock status of slices defined by the node. Locking is not specific to users or groups. It's not safe to assume that a slice should be locked when an assignment happens because there may already be other assigned users currently contributing data to the slices.

Parameters of the }WF\_AssignNode process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to assign the node in |
| pContributor | User (member in the }Clients dimension) that will be assigned as a contributor to the node. |
| pReviewer | (Optional) User, member in the }Clients dimension, that is paired with the contributor to review data entered in the slices defined by the node. |
| pNode | The member in the Approval dimension used to assign the contributor and (optionally) the reviewer users. |
|  |  |

### }WF\_UpdateNodeState

This Turbo Integrator process is triggered by end users of the workflow. How and when the process is triggered depends on the implementation of the workflow. Examples of when this process should be called include taking ownership of a node or opening a node, submitting changes on a node, approving or rejecting changes made to the node, and finalizing the node. The workflow processes do not provide a predefined list of actions that must be implemented in the solution. However, each type of action that is defined in the implementation must determine if slices based on the node is locked or unlocked. There is no built-in assumption that a node opening action will unlock slices for the node.

Parameters in the }WF\_UpdateNodeState process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to assign the node in. |
| pNode | The member in the Approval dimension used to change the state of the node. |
| pLock | Determines if slices defined by the node (member in the Approval dimension) is locked or unlocked as part of the node state change. 1 = Locked and 0 = Unlocked. Locking or unlocking at a consolidated level will also lock or unlock the decedent members below the consolidated member in the Approval dimension. |
| pComments | Comments added by the user as the state of the node is changed. These comments might explain data changes on submission, or the reason a submission was rejected. |
| pState | Updated state of the node. The state value can be any string the workflow implementor decides to describe the state of the node. |

Notes:

1. The }WF\_LockUnlockNode TI process is called from }WF\_UpdateNodeState to adjust the lock state of the nodes. This process uses the Locking property of the workflow (Any or All) to determine if the node should be locked when the user indicates the node should be locked by setting pLock to 1.
2. The }WF\_AddComment TI process is called from }WF\_UpdateNodeState to add comments to the workflow comment cube. Comment data is stored as part of an audit trail for the workflow.

### }WF\_LockUnlockWorkflow

Locks or unlocks a complete workflow. When a control dimension is used this will unlock all slices in the global overlay security cubes specific to the member in the control dimension assigned to the workflow. When a control dimension is not used with the workflow the complete global security overall cube is locked or unlocked. The cube tracking contribution state for the workflow is also updated to reflect that all currently assigned nodes are locked or unlocked.

Parameters in the }WF\_LockUnlockWorkflow process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to assign the node in |
| pLock | Determines if all slices defined in the workflow are locked or unlocked. 1 = Locked and 0 = Unlocked. |

### }WF\_DeleteWorkflow

This process is used to remove an existing workflow. If element security is being used the }ElementSecurity cube for the Approval dimension will be removed.

If the workflow being deleted is the last workflow to use a contribution cube then the Cell security cube and security overlay global cubes will also be removed.

All user groups created by the workflow will be removed.

Parameters for the }WF\_DeleteWorkflow process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to delete |

### }WF\_RemoveNodeAssignment

Removes a contributor user from a node. Other users might still be assigned to the node. If the node had a security group assigned, then the contributor user is removed from that group. An error is logged if the user is not already assigned to the node.

Parameters for the }WF\_RemoveNodeAssignment process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to assign the node in |
| pNode | Member in the Approval hierarchy dimension that has been assigned to the contributor user. |
| pContributor | User to remove from the node |

### }WF\_RemoveAllNodeAssignments

Removes all contributor users from a node. Any group created for the node and of the child nodes are removed.

Parameters for the }WF\_RemoveAllNodeAssignments process:

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to assign the node in |
| pNode | Member in the Approval hierarchy dimension that has been assigned to the contributor user. |

### }WF\_ResetWorkflow

Resets an existing workflow by removing all contributor and reviewer assignments. Any groups that were generated by the workflow are deleted. All nodes are unlocked. Comments may optionally be cleared by this process.

|  |  |
| --- | --- |
| Name | Description |
| pWorkflow | Name of the workflow to reset |
| pResetComments | If set to 'Y' the comments cube for the workflow is removed |

## 

## Design Notes

I will add a few important design details about the workflow processes:

1. The workflow processes do not strongly define node states. The node state is updated with the }WF\_UpdateNodeState process. When you run this process, presumably from an action button in a PAW book or Websheet, you can use any string you want to update the state. If you want to emulate TM1 Applications you would use the states listed here (<https://www.ibm.com/docs/en/planning-analytics/2.0.0?topic=applications-cognos-tm1-workflow>) (e.g Available, Reserved, Locked, Incomplete, Ready). As you implement these processes you will need to determine which states should lock or unlock the node.
2. There are two security modes, pSecurityMode parameter in the }WF\_CreateWorkflow processs. The Group security mode uses existing groups in TM1 just like TM1 Applications. The Client security mode will allow you to name specific users (members in the }Clients dimension) as contributors and reviewers for each node. When using the Client security the security groups will be created and applied automatically when the }WF\_AssignNode process is used to make assignments.
3. Locking slices is enforced with the GlobalSecurityOverlay mechanism in TM1, just like TM1 Applications. However, there are two locking modes, ‘Any’ and ‘All’. The Any mode means that any user or group that attempts to lock the node when using the }WF\_UpdateNodeState process will cause the node to be locked for all contributors assigned to the node. The ‘All’ locking mode prevents the node from being locked until all users or groups assigned to the node have used the }WF\_UpdateNodeState process to lock the node.