

PL11233

Behaviors 201 for Padawan Learners

Ralf F. A. Walter
Autodesk, Inc.

Learning Objectives

- Understand how categories drive the initial behaviors of objects
- Understand how to use behavior change commands
- Understand how lifecycle changes affect other behaviors
- Interact with properties both manually and automatically

Description

Behaviors drive many aspects of objects in Vault. Understanding how they are configured and using them accordingly enables deterministic workflows. Furthermore, blocked workflows also become clear. Bringing all of the behaviors together through categories makes for a cohesive experience.

Your AU Experts



Ralf F. A. Walter

Ralf has a degree in computer information systems. Before his 20-year tenure at Autodesk, Inc., he was a software engineer with AT&T Network Systems and specialized in CAD productivity applications for AutoCAD users. At Autodesk, he has been in many development roles for software products such as AutoCAD, AutoCAD Mechanical, Mechanical Desktop, Inventor, Simulation, and, currently, Vault. This is his second year speaking at Autodesk University.



Irvin Hayes, Jr.

Irvin is a Product Manager on the Autodesk Vault team based in Novi, Michigan. He has worked at Autodesk for ten years starting in product support and as a user experience designer. Irvin is a Microsoft® Certified Professional and has been working in the information technology field for more than 22 years. He helps partners, consulting and sales develop Vault deployment plans in enterprise environments and system requirements. You can find multiple classes Irvin has presented at Autodesk University, on a wide range of Vault topics. Irvin is a technology geek and loves sharing with the community on Twitter (@ihayesjr) and Flipboard Magazine.



Table of Contents

Learning Objectives.....	1
Description.....	1
Your AU Experts	1
Introduction	3
Behaviors Summarized	3
Relationship Gathering	4
Behaviors Administration Settings.....	5
Understand how categories drive the initial behaviors of objects.....	6
Understand how to use behavior change commands	12
Change Category Command	12
Change State Command.....	14
Change Revision Command	17
Understand how lifecycle changes affect other behaviors	19
Criteria Checks	19
Triggering Behavior Changes	20
Interact with properties both manually and automatically.....	22
Manual Property Edits	22
Automatic Property Edits.....	24
Introduction to Properties Administration	25



Introduction

The “Behaviors 101 for Younglings” course covered the basics of Vault behaviors including categories, lifecycles, revisions and properties. That material serves as a prerequisite to this course and this is a prerequisite to “Behaviors 301 for the Jedi Master”. It is assumed there is already a basic understanding of those topics as the primary goal for this course includes the use of behavior commands with some insight to behaviors administration.

Behaviors Summarized

Behaviors, in a nutshell, help to describe something meaningful about objects. They are more than just metadata, however, in that they generally interact with each other in some way when the user has performed some kind of action. Furthermore, some behaviors

Revisions

A revision or revision number is an identifying alpha-numeric value that represents the iteration of an object corresponding to a design milestone or release event. They are the primary key to understanding whether you are working with the latest or correct record. In Vault a revision scheme denotes the progressive values that can be assigned as revision:

- Revision Scheme can be numeric, alphabetic or a mixture, traditionally it is from 0-1 or A-Z with a number of variations
- Revision schemes can be employed to denote major and minor iteration changes

Revisions in Vault have three parts with some kind of delimiter separating a primary, secondary and tertiary decoration (letters and/or numbers). For example, a part file with a revision of “8.3.2” could mean that the part is on its eighth major release cycle, been reviewed three times (for this particular revision) and was “tweaked” (quick-changed) twice. It should be noted that secondary and tertiary values are not always displayed. In Vault the tertiary value is not shown unless it was changed (also known as a revision ‘bump’) and similarly for the secondary value. Bumping a revision always resets the fields after it so then in the example shown a secondary bump from “8.3.2” will result in “8.4” and a primary bump would result in “9”.

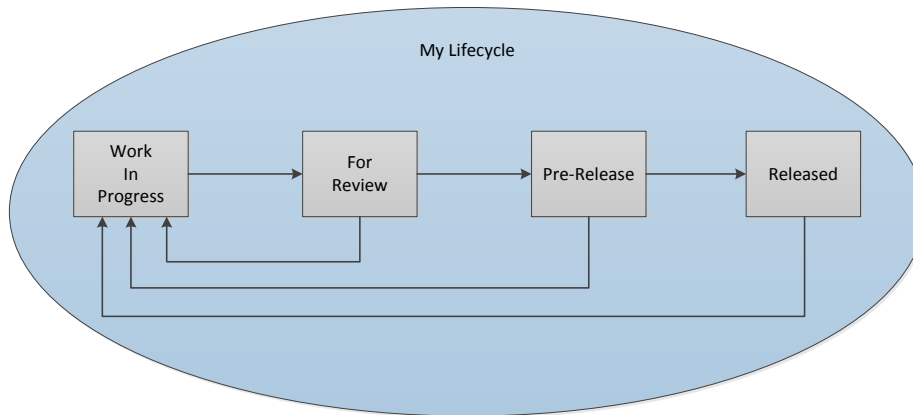
Lifecycles

A lifecycle describes the process or stages an object passes through from innovation to production or construction. Another way to describe a lifecycle is that it is the intended flow of work for a release or revision. The checkpoints throughout that journey are commonly referred to as lifecycle states.

A lifecycle state is a reflection of an object status within a discreet process. In Vault it’s typically about managing design data through to manufacture or build:

- It describes the current design status
- A state can contain security settings to control access to an object
- When applied to a file, the state controls who can read, modify and delete a file
- It also controls whether files can be purged





The figure above describes a lifecycle containing four states and the intended flow of work through those states. The movement from one state to another is called a transition and, in this example, the lifecycle is configured such that some states can transition to other states but not all states necessarily transition to any other state

Properties

Properties are text-based strings, numeric values, booleans or dates making up non-visual object identifiers enabling users to describe object characteristics, search for and identify objects.

In other words properties are, simply put, information. Some properties hold on to Vault-specific data and cannot be changed (intrinsic). Other properties may reflect file property information even write to file properties. The read and write of properties to and from files in Vault is called property mapping. Not all properties are mapped and can simply serve as information associated to the object.

Categories

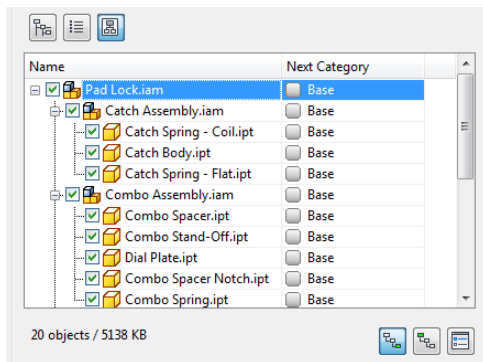
Behaviors are assigned to and made available to objects via the notion of categories.

Categories, however, are not actually behaviors but instead serve to facilitate initial assignment and availability of revisions, lifecycles and properties. Furthermore, they make any number of behaviors available (if configured as such).

Relationship Gathering

The commands (covered later in this document) to interact with revisions, lifecycles and categories utilize a common user interface element referred to as the Relationship Gatherer (other commands do as well but this document will just cover the behavior-related commands). As the name implies its function is to gather the relationships of objects. For example, consider an assembly that has subassemblies and parts all of which are located in different places in Vault. In order to interact with them all at once relationship gathering is necessary to gather all of the children (or parents) of the selected object(s). Relationship gathering can also gather other kinds of related objects like attachments, library files, etc. The relationship gathering UI element may vary slightly based on the type of object selected (not all options apply to all object types) and also the command used but will mostly look and function the same:

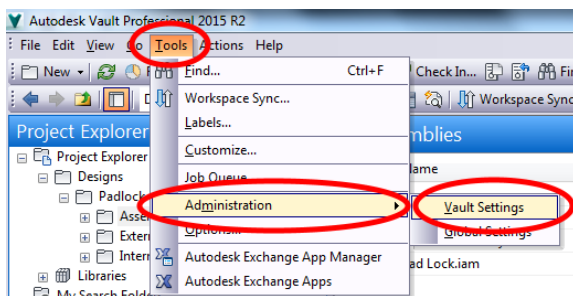




Note: Some objects may be unchecked and disabled. When that occurs it is usually due to some sort of restriction regarding the object in question. An example of this is that the object is inaccessible due to security reasons (the user does not have read access to the object).

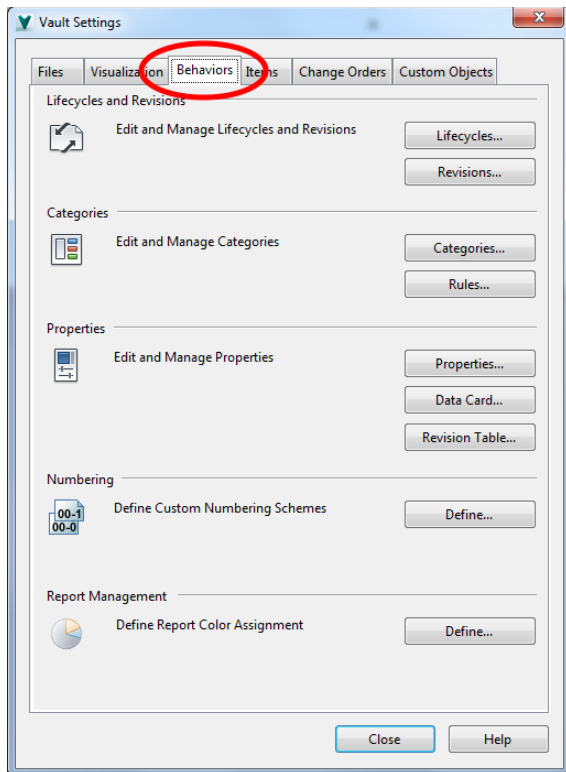
Behaviors Administration Settings

This document will make reference to accessing the Behaviors settings in Vault Settings dialog. To access the settings use the “Tools” menu, then “Administration” and “Vault Settings”:



The Vault Settings dialog will then appear and the Behaviors tab is where this document will make reference to:

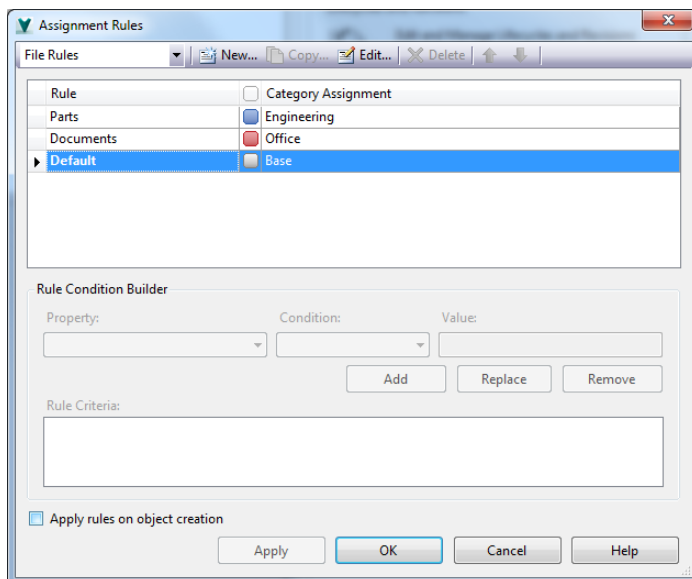
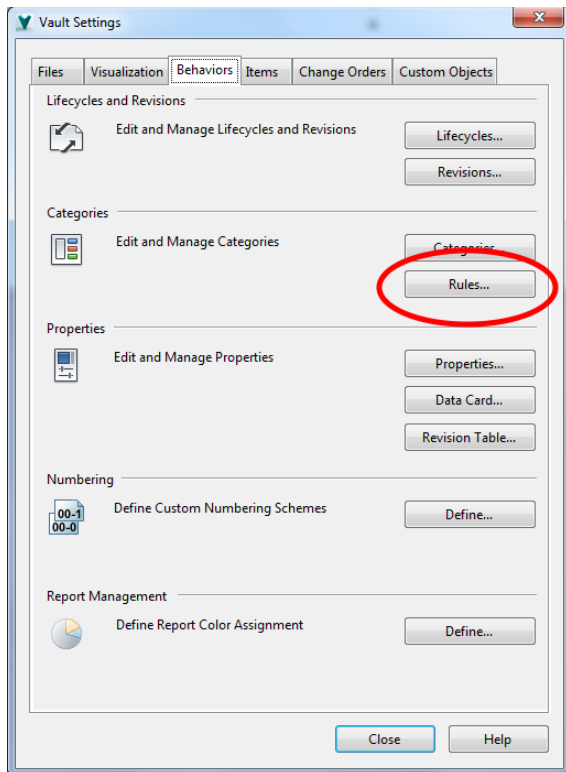




Understand how categories drive the initial behaviors of objects

When objects are initially created in Vault (either initial check-in of files or explicit creation of objects such as folders items or custom entities) they are automatically assigned to categories via category assignment rules. It is accessible via the Categories section of the Behaviors setting by clicking the "Rules..." button:



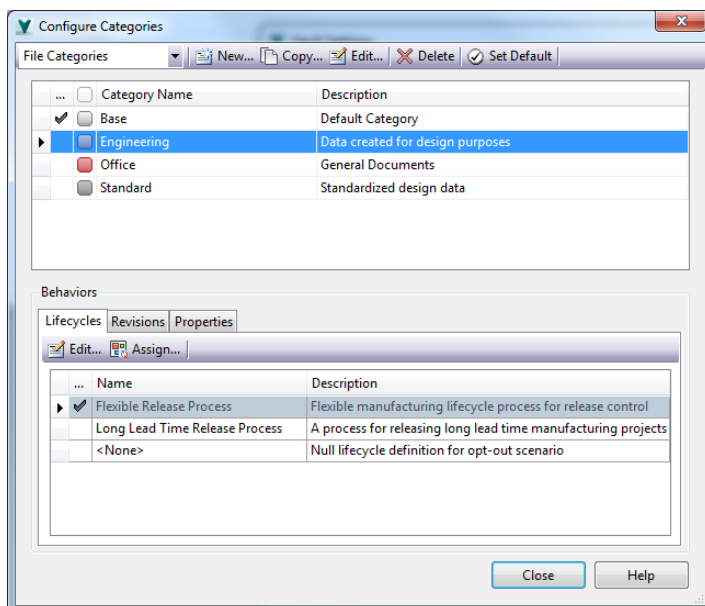
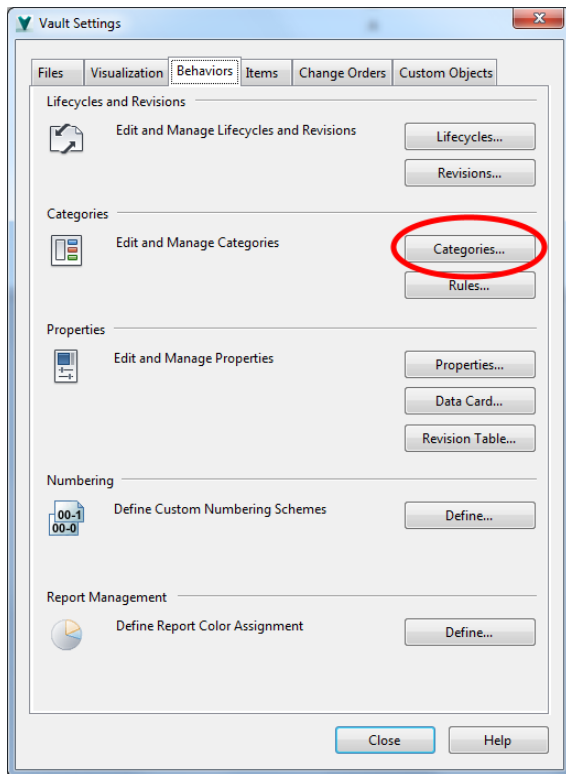


The category assignment rules will then be listed and can be edited. Rules are checked in the order shown until a successful rule is found and that category is what is assigned.

Note: There is always a “Default” category rule and that rule cannot be deleted. The “Default” rule assigns the object to the default category for that object type.

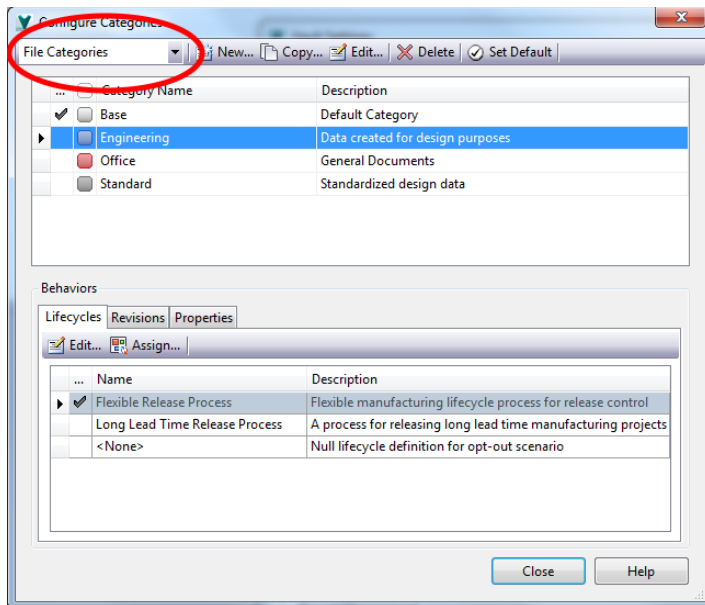


If “Apply rules on object creation” (applicable to files only) is not checked (it is checked by default for new vaults) then a category will be assigned based on the object type’s default category accessible via clicking the “Categories...” button in the Behaviors settings:

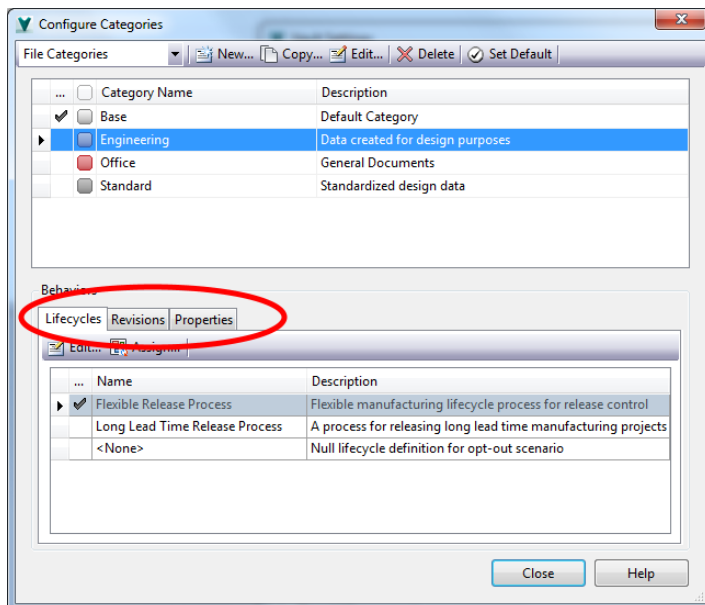


It is important to note which object type is being worked with. For example for files then the dropdown list should reflect “File Categories”:



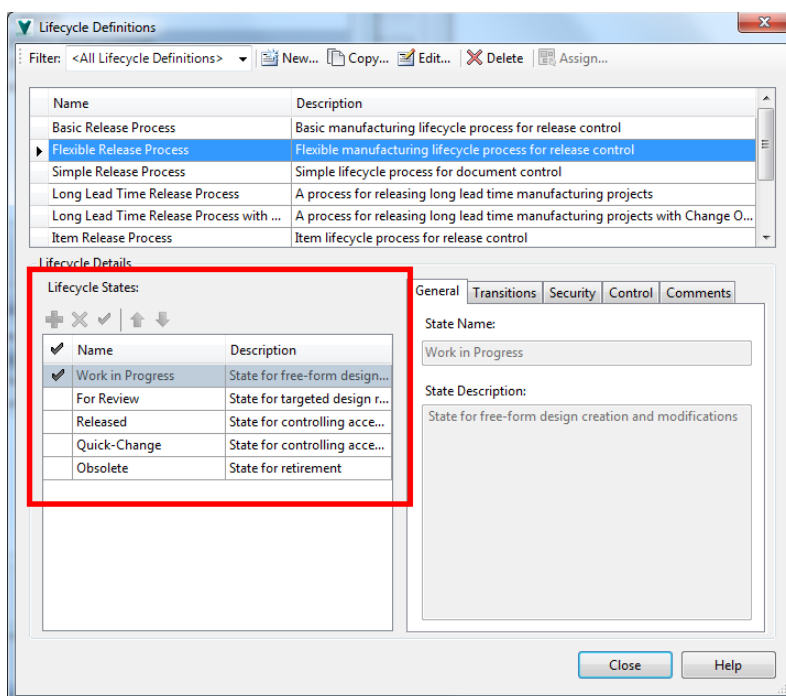
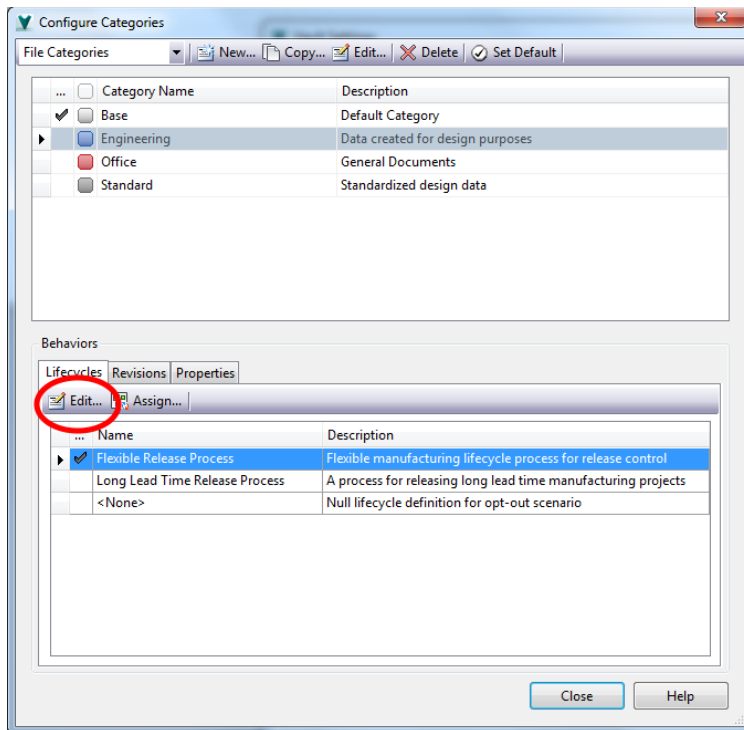


The initial behaviors that the new object will be primed with are shown in the respective tabs:



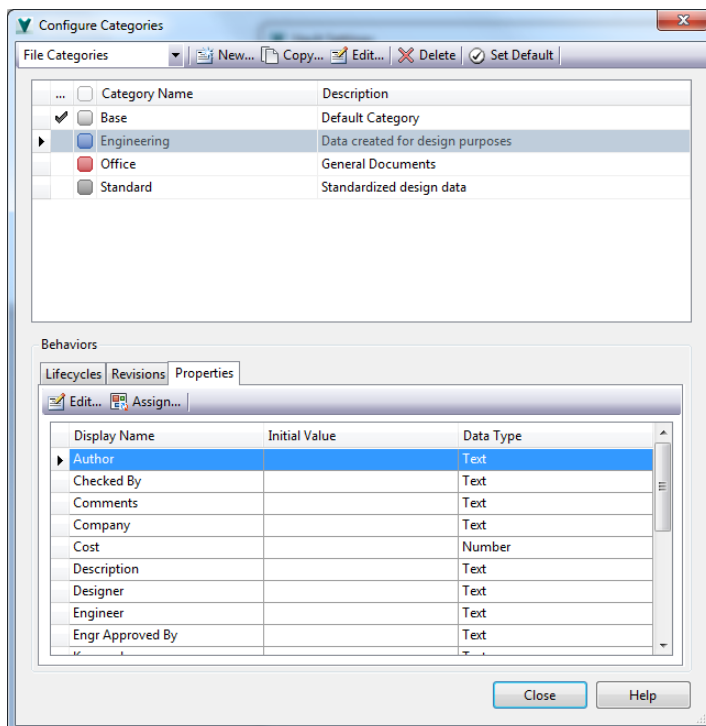
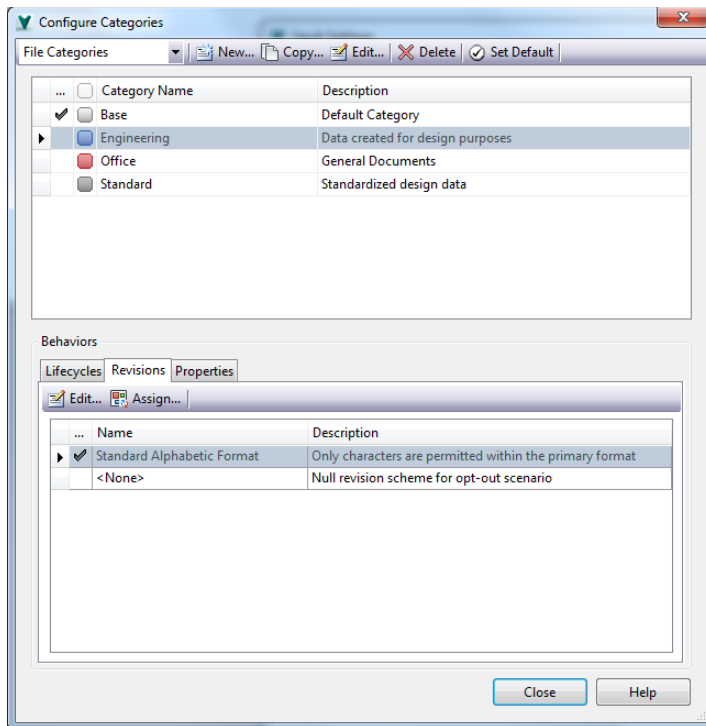
In the figure above there are three lifecycles listed. The one that will be used for the object is the lifecycle that is checked. Note: The "<None>" lifecycle is a means for the administrator to allow for objects to be able to not participate in lifecycles. The "lifecycle" assignment will use the lifecycle definition's first lifecycle state to determine the initial state of the object ("Work In Progress, for example). To view the states for the lifecycle click the "Edit..." button (which is for editing the list of lifecycle definitions), navigating to the lifecycle in the list and then clicking the "Edit..." button for the lifecycle definition itself:





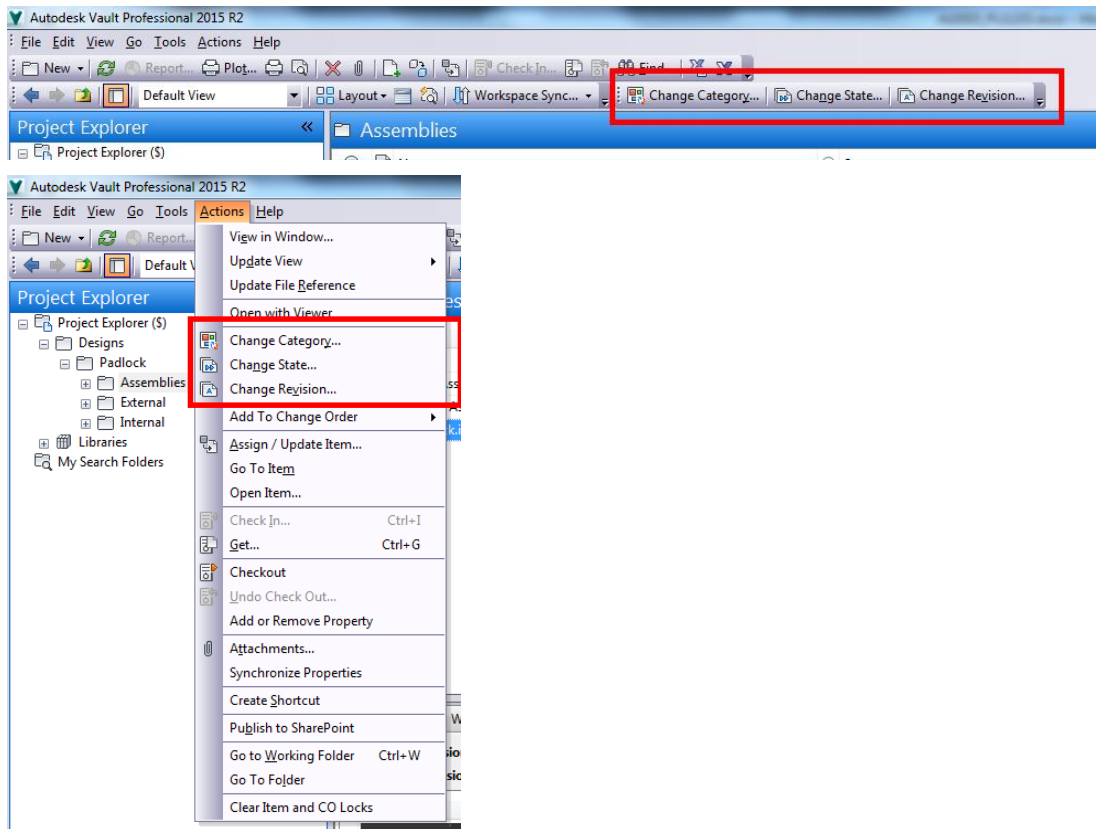
Revision and property availability follow a similar pattern to lifecycles:





Understand how to use behavior change commands

The behavior commands are (by default) grouped together in either the behaviors toolbar or in the Actions menu:



Executing a command from either location accomplishes the same thing.

Objects in the relationship gatherer element that will be operated on will be shown as checked and available (to be unchecked).

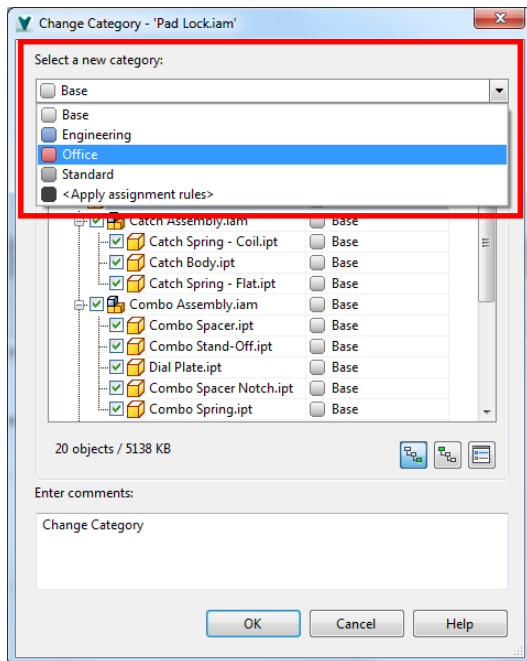
In the event of a failure with any of these commands a restriction may be shown indicating that the operation cannot continue and information about what the first failed object and reason will be displayed to the user.

An optional comment may be included in these operations and, if configured by the administrator, there may be a default. There is no enforcement of entering a comment.

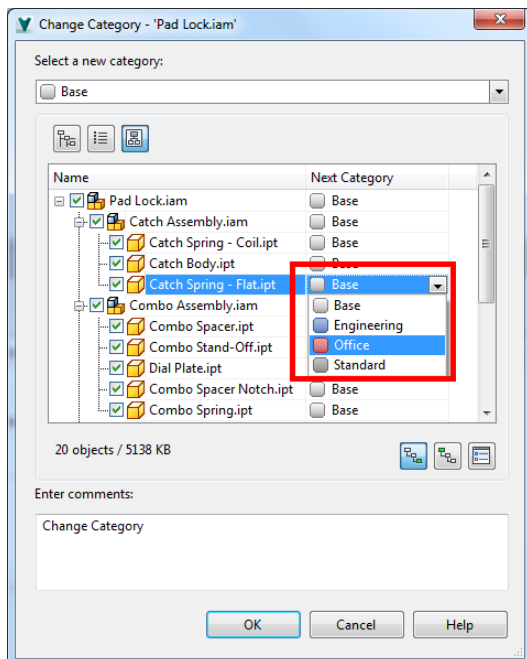
Change Category Command

The dialog for the change category command will display the current category of the objects in the drop list at the top of the dialog (when the category varies across objects then "<Click here to select new category>" will be shown instead). The category of all checked objects can be changed by using the drop list.





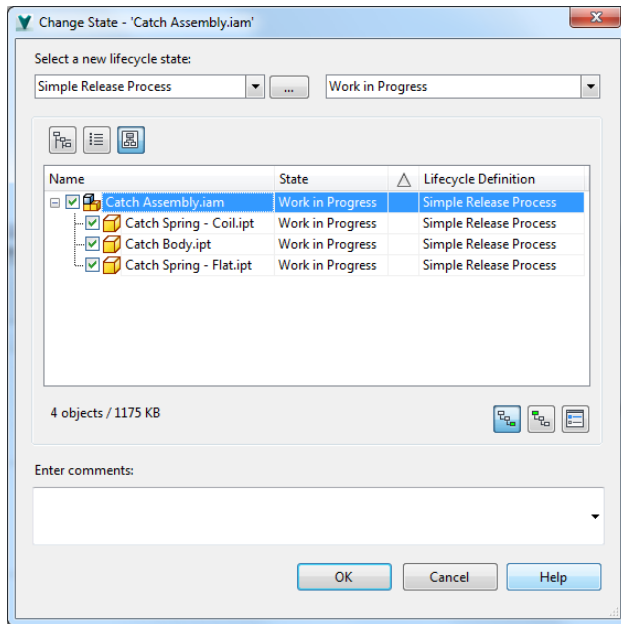
Categories may also be changed one at a time by click in the “Next Category” column for those object rows.



The Change Category command will not change the properties of the object. It may, however, assign the object(s) to a lifecycle or revision if either is not currently assigned a respective behavior. Properties that exist for the new category that did not with the previous will be created.

Change State Command

The change state dialog will show two drop lists at the top of the dialog:

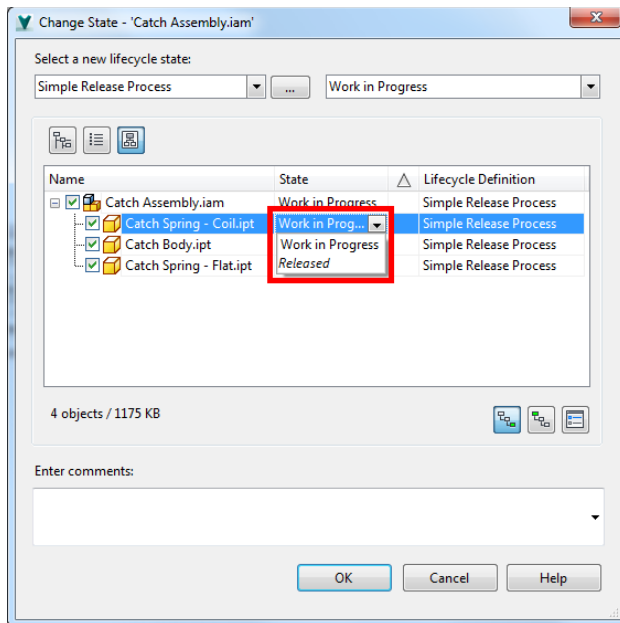


The upper left list reflects the definition (unless it varies) and then the state in the right.

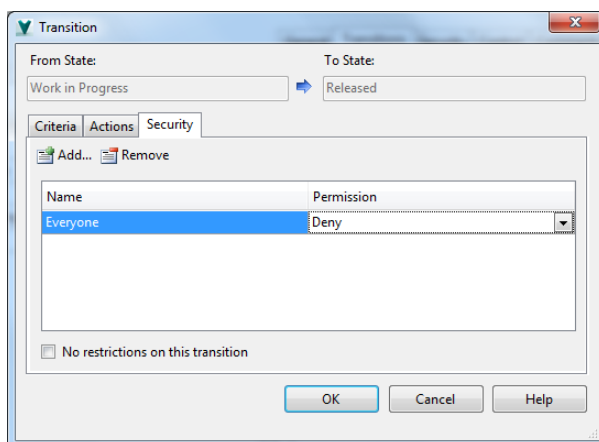
Changing the Lifecycle State

Selecting a state from the drop list in the upper right of the dialog will invoke a state change for all checked objects. States may be changed individually by selecting in the state column for the respective object row:





The available states are based on administration settings for state transitions. So the states shown are the available “To States”. The administration settings are accessible from within the Behaviors settings and clicking down into “Lifecycles...”, editing the lifecycle definition that is in use, clicking on the appropriate transition and viewing the Security tab:

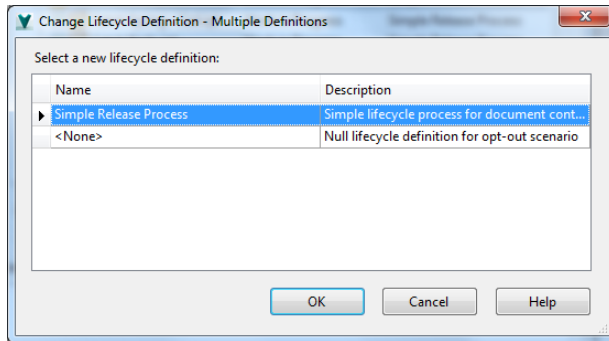


In the dialog box above the administrator has configured the state transition from “Work in Progress” to “Released” to be denied to “Everyone”. Therefore the “Released” state will not be available in the list of available states if the current state is “Work in Progress”. This is often configured on a per-user or user group basis.



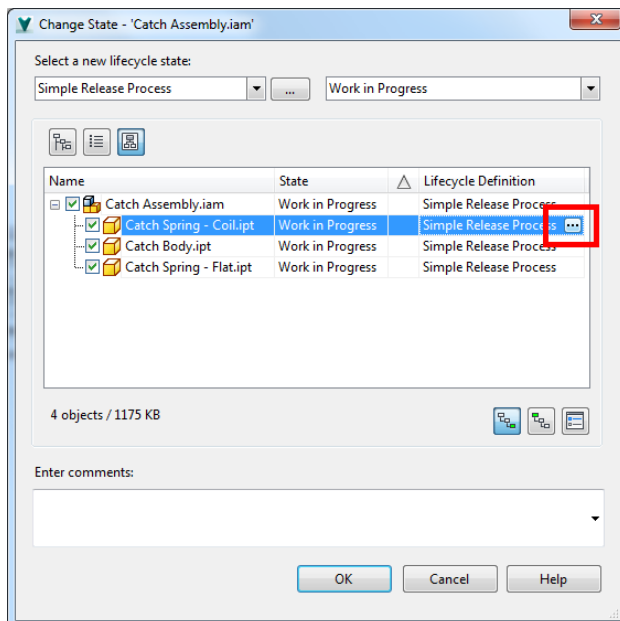
Changing the Lifecycle Definition

The drop list of lifecycle definitions may be used to change to another lifecycle definition. However, the only definitions shown here are those that are currently in use by the objects in the relationship gatherer. Clicking the “...” button at the top of the dialog allows the user to change to a different definition:



Note: The available definitions are based on the category.

Like states, definitions may be changed on a per-object basis by click in the Lifecycle Definition column for the respective row of the object and then clicking on the “...” button:



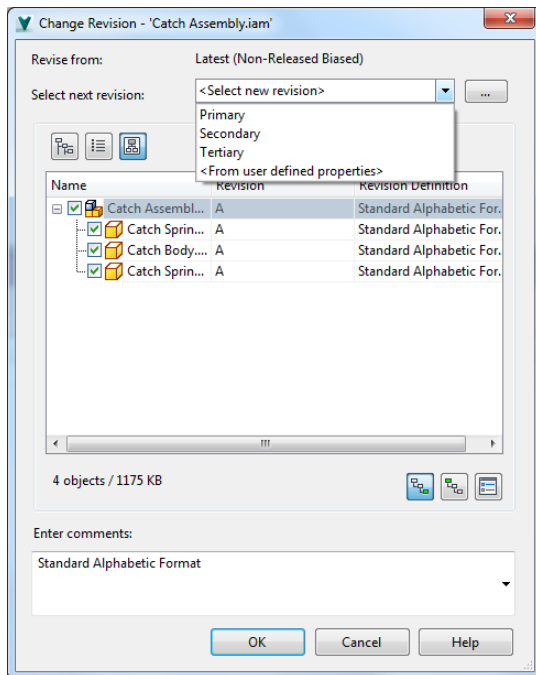
Changing an object’s lifecycle definition will change the state to the default state of the target definition.



Change Revision Command

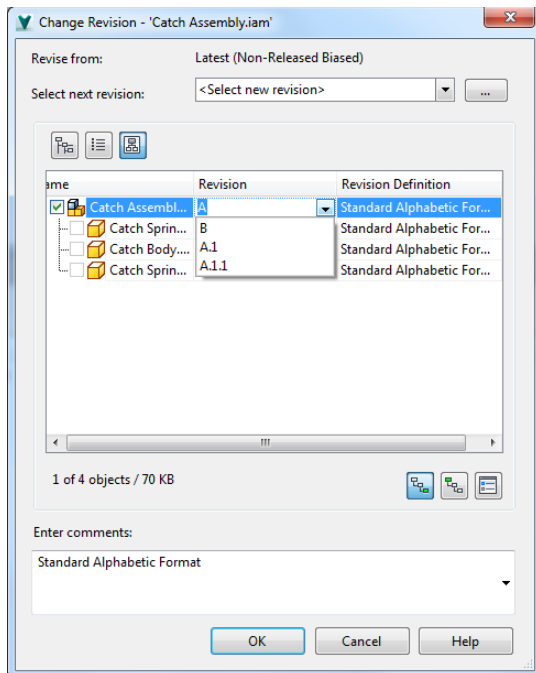
Changing the Revision

Change Revision presents to the user a drop list with options to bump the primary, secondary or tertiary revision. Additionally the user may change the revision based on user defined properties (UDP).



Note: Revising based on a UDP requires that the UDP character adheres to the current revision scheme.

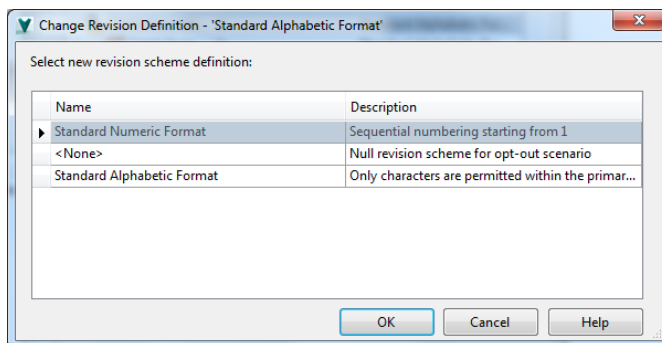
Selecting a revision from the drop list at the top of the dialog will bump the revision for all checked objects. To bump individual object revisions click in the Revision column for the respective row of the object:



Note: Changing the revision via the column does not provide a means to change “From user defined property”.

Changing the Revision Definition (Scheme)

The revision definition (also known as scheme) scheme may be changed for all checked objects by clicking the “...” button at the top right of the dialog box. A list of revision schemes are presented then to the user:



The available revision schemes are based on those configured for the category of the object.

Like the Revision, Revision Definition (scheme) may be changed for individual objects via the column of the same name in the row of the respective object.

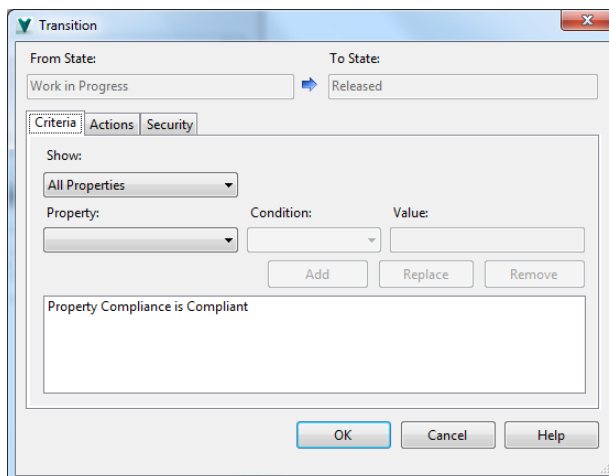
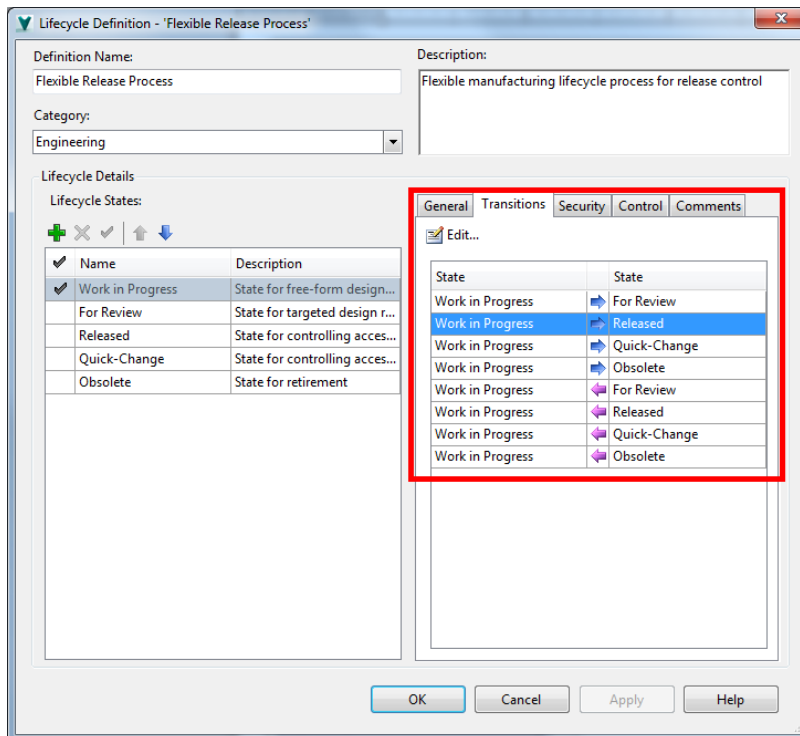


Understand how lifecycle changes affect other behaviors

Changing lifecycle states often do much more than that. They may rely on certain criteria or trigger other behavior changes.

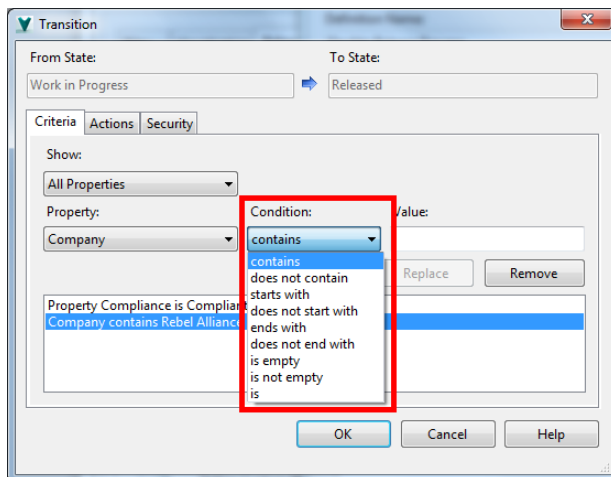
Criteria Checks

Criteria checks are properties-based and performed at the state transition level. If criteria are not met then the state transition will be disallowed (and displayed in the form of an error). View the administration settings of any lifecycle and then in the Transitions tab select any transition that goes to the Released state and then click the Edit button:



In the example shown the Work in Progress to Released state was chosen. State transitions to the Released state tend to utilize the “Property Compliance is Compliant” criteria in order to allow the transition. Property Compliance is, in itself, a system property in Vault. The value of “Compliant” (or something else) is automatically set by the system when property compliance is checked (at check-in or during a property file re-index). “Compliant” means that property policies and property equivalence (due to read and/or write mappings) are met. Non-compliant properties can cause a lifecycle change to fail.

Like Property Compliance, other property criteria can be checked for and by making use of the Condition drop list:



Property Criteria that fails will cause a lifecycle change to fail as well.

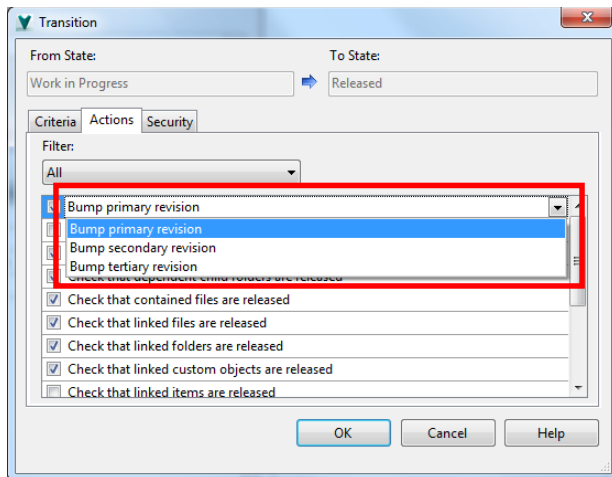
Triggering Behavior Changes

Lifecycle changes can cause other behaviors to react by way of lifecycle transition actions. What this means is that when a lifecycle change is successful something will occur. These are the actions that “do” something. There are other lifecycle transition actions that will “check” on certain things instead, like “Check that dependent child files are released” or other “Check that...” actions. The behaviors that are triggered as the result of lifecycle state changes are revision bumping and synchronizing properties using job server.

Bump Revision

The primary, secondary or tertiary revision (if selected) may be bump as the result of a state transition.





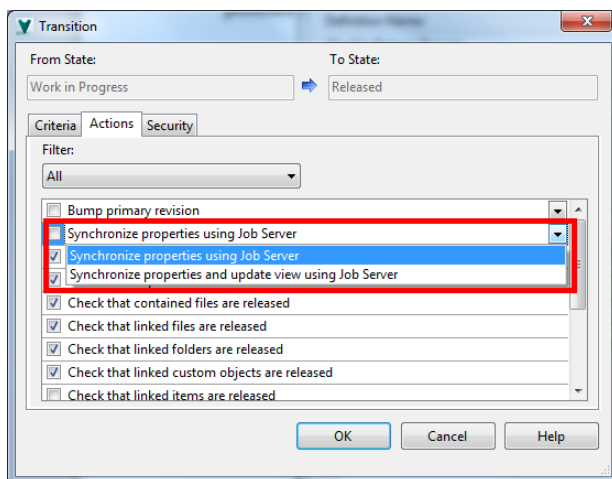
When the state transition occurs then the portion chosen (primary, secondary or tertiary) of the revision will be bumped.

Synchronizing Properties

When a transition occurs file versions are changed which does cause a check-out/check-in. However, properties are not synchronized (updated) during this process unless the administration option to synchronize those properties is checked. There are two:

- Synchronize properties using Job Server
- Synchronize properties and update view using Job Server

Synchronize properties does not occur at the time of the lifecycle change. The work is spun-off and handled by the job processor so the results will not be immediate.

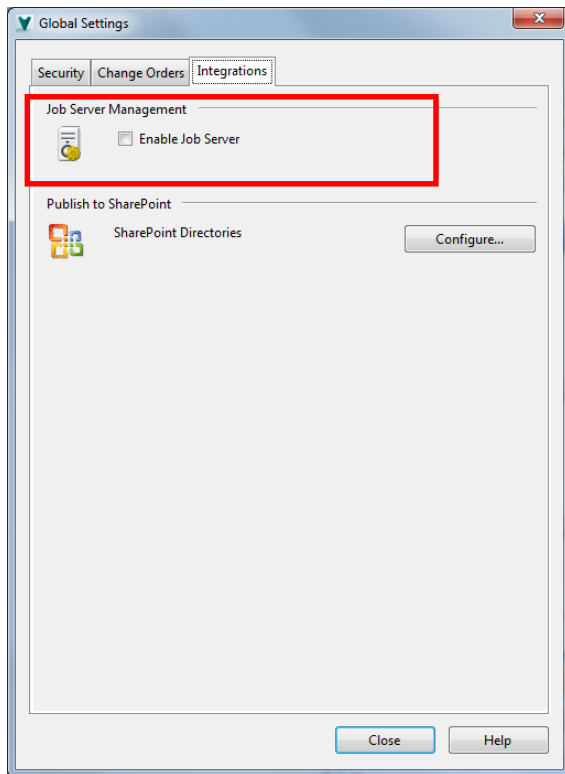


One option synchronizes just the properties and the other will also update the view. This class topic does not cover file visualization so we'll just stick to the properties synchronization. Only property file write-back will occur during this process (not property-read).

A requirement that must be met first in order to achieve properties synchronization is that a job server must be configured and the job server user must have read & write access to the vault



objects that will be affected as a result of properties synchronization due to a lifecycle state change. Enabling and setting up the job server is a function of the administrator. Enabling the job server can be found in the administration global settings:



Because the properties synchronization is a post-process of the lifecycle state change any failures that may occur during synchronization will not affect the success or failure of the state change itself.

Note: A manual property synchronization is available via the “Actions” menu.

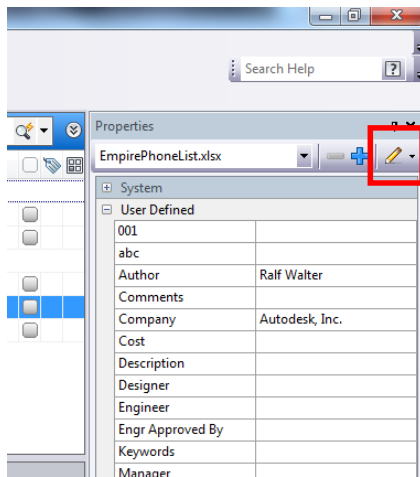
Interact with properties both manually and automatically

In previous section we touched lightly on some concepts of properties including what they are and also how lifecycle changes can cause them to become updated (synchronized). Let’s look a bit deeper into manually interacting with properties and also what causes them to change automatically.

Manual Property Edits

Manual property edit in Vault are accomplished via the Properties window which is typically located on the right-hand side of the screen within Vault by default:

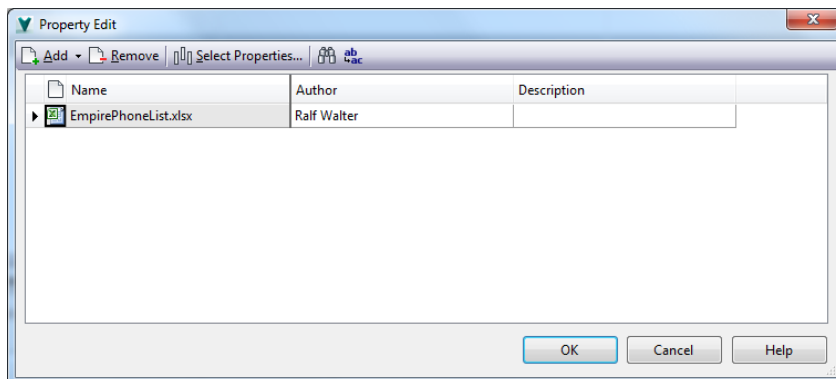




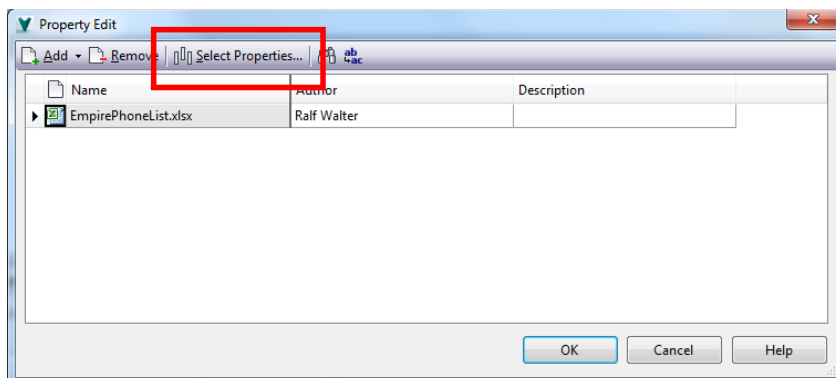
To edit simply click the pencil button in the Properties window (outlined in red above).

The dropdown on the button allows the user to enter edit mode with previously-selected properties or with currently-selected properties. If no properties have been previously selected then the “Author” and “Description” properties will be shown to start with.

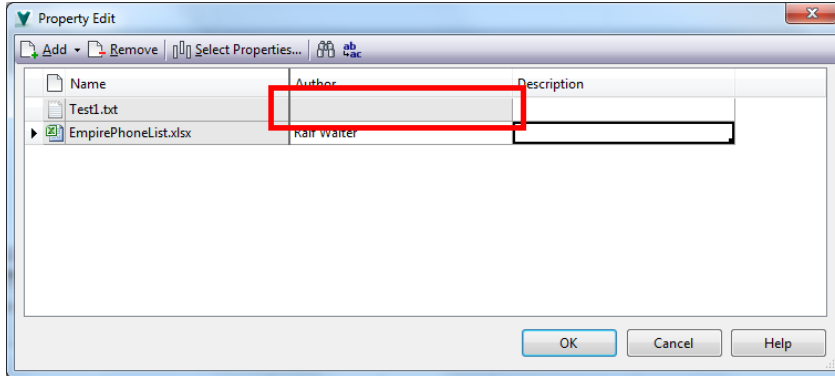
The Property Edit dialog will then be shown:



To add property columns simply use the “Select Properties...” button at the top of the dialog:



Properties for multiple objects may also be edited. Property fields will be disabled for objects that do not have those properties (although they may be enabled for others) or for system properties:



It is by design to allow the user to select non-editable properties as it is often useful to display other information that may assist with property editing.

As long as the properties are editable simply click (or double-click in some cases) in the field to edit the properties that you wish to edit. The field will then react accordingly based on the type of property that is being edited:

- Text – The user may enter free text. In some cases a list of values may be configured by the administrator to force the user to adhere to only those selections.
- Number – The user may only enter number values. Like “Text”- a list of allowable values may be configured by the administrator.
- Boolean – A true or false value will be available
- Date – A calendar tool will be shown to allow the user to select a date

Changes to the object(s) will not occur until the OK button has been clicked. Once the OK button is clicked the objects will be updated. It is important to note that, for files, when properties are manually changed the files are automatically checked out, the changes are applied and then the files are checked back in.

A manual property edit may put properties out of compliance. For example if a text property requires a value the user may still clear out the value and the edit will be successful. Recall that property compliance comes into play for lifecycle changes and may affect whether or not a lifecycle change is successful.

Finally, once properties are updated a dialog will appear to display the results of the property edits including successful edits and failures.

In the event that a property write mapping is configured for the Vault property that has been changed the file will be checked out, the mapped property in the file will be updated and then the file will be checked back in. This may then cause Vault properties to become automatically updated because of the checked-in file, which is covered next.

Automatic Property Edits

Automatic property edits occur via a mechanism shown as *property mapping*. In a nutshell, property mapping creates a relationship between Vault properties and file properties. What this means is that Vault properties may be updated from file properties (known as *read-mapping*) and file properties may



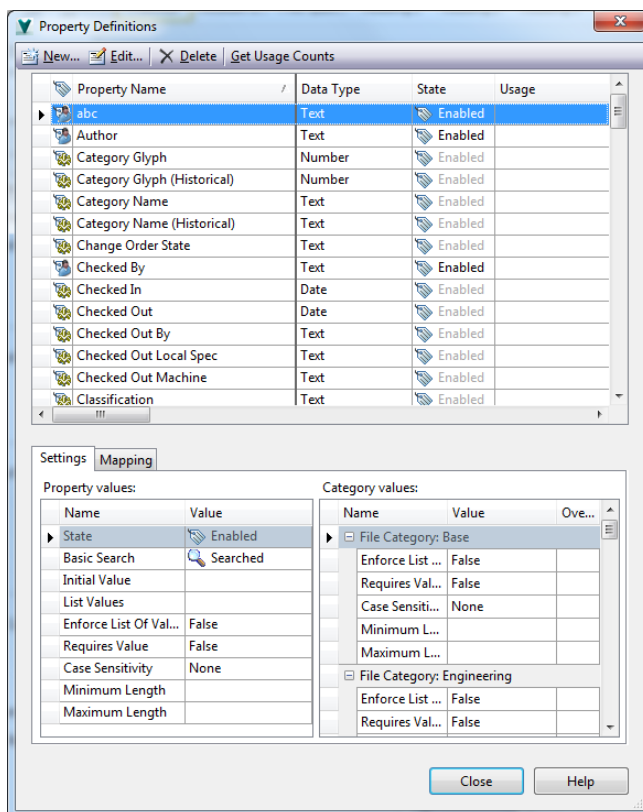
be updated from Vault properties (known as *write-mapping*). Read-mappings only occur when files are explicitly checked in or as the result of a lifecycle state change that also synchronizes properties. In some cases there could be a read and write mapping. The administrator sets up property mapping and more information on that can be found in the *Introduction to Properties Administration* section of this document.

Read-mapping means that Vault properties mapped to be read from file properties will become automatically changed as a result of a manual check-in. A key thing to keep in mind here is that property policies and compliance will be enforced here. At the time of check-in Vault properties that are read-mapped from file properties may cause some Vault properties to become non-compliant and therefor may cause lifecycle changes invalid (if configured as such) and there may also be action necessary to make those properties become compliant.

Introduction to Properties Administration

This is a supplemental section that describes, at a high level, some of the aspects of properties administration mainly with respect to mapping and compliance.

The administration dialog is available from the same tab in Vault Settings as the other behaviors:



The properties are listed at the top and then there are two tabs below: "Settings" and "Mapping". Administrators may edit these settings by editing the property selected (not covered in this document).



The “Settings” tab

The “Settings” tab (shown above) shows basic property information including “Property values” and “Category values”.

- “Property values” contain both general values and also policy values.
 - General values
 - “State” – This is simply whether or not the property is enabled or disabled. A disabled property will not be available to objects when they are assigned to a category.
 - “Basic Search” – This indicates whether or not the property is automatically included in basic search (“Tools” menu, then “Find”).
 - “Initial Value” – This primes the property with an initial value. This could be configured to set the value via read the property from a file as well.
 - “List Values” – These values are the values that will be available to the user. Also see “Enforce List Of Values”.
 - Policy values are values that are enforced for things like lifecycle changes. If a property policy is not met then it could cause a lifecycle change to fail. This is also known as *property compliance*.
 - “Enforce List Of Values” – If the property value is not one of the values in the list then the property is deemed non-compliant.
 - “Requires Value” – Just as it implies – the value cannot be blank.
 - “Case Sensitivity” - The property value must adhere to the format chosen.
 - “Minimum Length” – The number of characters must be at least this many.
 - “Maximum Length” – The number of characters must not exceed this many.
- “Category values” shows which categories the property is available to and also shows the policies that may be overridden on a per-category basis.

The “Mapping” tab

“Mapping” information describes how Vault properties will either read or write from file properties:

