AUTODESK UNIVERSITY 2015

PL11664 - Behaviors 301 for the Jedi Master

Irvin Hayes Jr, Autodesk, Inc., Product Manager - Primary Speaker
Ralf Walter, Autodesk, Inc., Principal Engineer - Co-Speaker

Learning Objectives

- Learn how to create and manage Properties
- Learn how to create and manage Revision Schemes
- Learn how to create and manage Lifecycles
- Learn how to bring it all together with Categories

Description

This class is essential for those who want to become a Jedi Master in creating and administering Vault Behaviors. You should have previous training in Behaviors 101 and 201 before attending this class. The main focus of this class is to cover Vault Behavior administration and become a master in behaviors. This class the third of a series of three Vault Behavior classes focusing on creation and administration of categories, lifecycles, revisions, and properties. After this class, you will have a strong presence in the force of Vault Behaviors.

Your AU Experts

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.

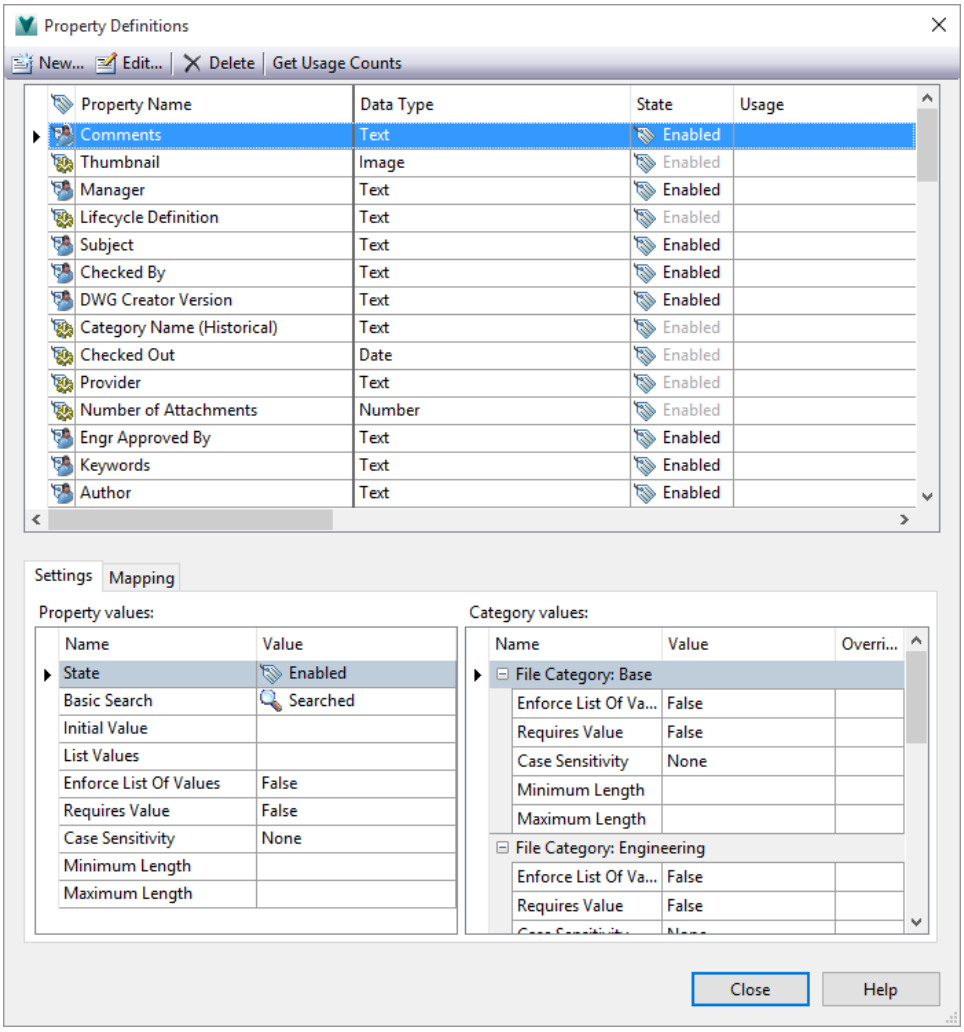
Ralf F. A. Walter has a degree in computer information systems. Before his 20-year tenure at Autodesk, Inc., Ralf was a software engineer with AT&T Network Systems and specialized in CAD productivity applications for AutoCAD users. At Autodesk, he has been in many development roles for software products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

Contents

Property Administration

Properties are text based strings, numeric values or dates making up non-visual object identifiers, enabling users to describe object characteristics, search on and identify objects. Properties are associated with files, items, change orders and reference designators. Vault has two types of property definitions: system-defined properties and user-defined properties (UDPs). System-defined properties are a global set of properties that are applied to files in Vault. System-defined properties cannot be deleted but can be renamed and has some mapping options. User-defined properties are custom properties created by an administrator that can be used to provide additional information on vault entities.

Property Administration can only be performed by users with the Administrator role. To get to the property definitions, select Tools -> Administration -> Vault Settings -> Behaviors tab and click the Properties button.



Common Terms

Before working with properties, it is important to become familiar with the terms, data types and attributes associated with the Vault properties system.

The following table contains terms commonly used with properties.

Term	Definition
Associations	Attribute that determines whether the property gets associated with a file, folder, item, or change order.
Compliant	The status of a property that meets all property policies and equivalence evaluations.
Classification	There are essentially 2 main classifications Standard and Custom. These classifications refer to the classification of the CSP (file properties). A classification of Standard means that this file property is a “standard” property for these types of files (Inventor ipt). A classification of Custom means that the file property is a “custom” property and may only get associated with some of these file types.
Data Type	The type of data accepted for the property value. This type can be text, number, boolean, or date.
Database Property	Any property in the database, either user defined or system.
Entity	An entity is the system class with which a file can get associated. Entities are files, folders, items, or change orders.
Equivalent	The status of a mapped property when its value matches the source value.
File Property	A property associated with a file.
Mapped Property	A property from which the propriety being defined gets its value. For example, a UDP can get its value from several different file properties. A file property can get its value from a system property.
Mapping	A set of relationships between the property and the property from which it receives its value. There can be multiple mappings for a given property definition.
Master	The property from which a mapped property gets its value. The master property writes its value to the subordinate property.
Non-compliant	The status of a property when it has failed to meet one or more property policies or its equivalence evaluation.
Non-equivalent	The status of a mapped property when its value does not match the source value.

Property Definition	All attributes and constraints about the property including its name, data type, initial value, mapping, minimum and maximum values, case values, in-use value, and basic search value.
Property Name	The name used in the GUI (graphical user interface) to identify the property.
Property Policy	Depending on the data type, the property policy specifies certain constraints that get met. The constraints may include a value range, a value type, or a value format must be met. For example, a property policy might be described as follows: the property must have a value, and that value must be in the range of 1 to 10. When a property fails to meet its property policies, it is considered non-compliant.
Property Value	The literal content of a property attribute for a specific file version.
Override	Determines whether the property value gets overridden by the policy defined by its category.
Subordinate	The mapped property that receives its value from the master property.
System-Defined Property	A property in the database created by the system, which gets assigned to a file.
User-Defined Property (UDP)	A property in the database created by an administrator. The property can be applied to a file when it gets added to a vault.

Content Property Providers

Several Content Property Providers already exists and are listed in the table below. The Content Property Provider indexes a file and provides the file's File Properties or Content Source Properties. These File Properties are used to create the property mappings as described in the Mappings section in this document. Some Content Property Providers also provide property write back capability that allows clients to update these File Properties directly in the file.

Provider	Supported File Extensions	Write Back Support
AutoCAD	.dwg,.dwt	Y
AutoCAD Electrical	.wdp	Y
Inventor DWG	.dwg	Y
Inventor	.ipt,.iam,.idw,.ipn,.ide	Y
Office 97-2003 Documents	.doc,.dot,.xls,.xlt,.ppt,.pot	Y
Office Documents	.docx,.docm,.dotx,.dotm,.xltx,.xltm,.xlsx,.xlsm,.xlsb,.potx,.pptx,.pptm,.potm	Y
AutoCAD C3D	.c3ddata	N
Sheet Set Data	.dss,.dsu,.dsh	N
Revit Models	.rvt,.rte	Y
Revit Families	.rfa	N
Outlook	.msg	N

Other Files (IFilter)	.*	N

The Autodesk Vault server also utilizes a technology created by Microsoft called iFilters. iFilters are small applications that exist on the server containing a set of instructions on how to open and extract property data from files. There are many third-party iFilters available for other file formats. An Internet search for iFilter returns some of the filters that other software developers have published.

Property Definitions Dialog

The Property Definitions dialog displays all system and user-defined properties that are in the vault. It displays the data type, the proper state and the Usage count that is blank until you select the Get Usage Counts button in the menu bar. Selecting the Get Usage Counts button gets the number of times objects use that property inside of the vault.

Property Name	Data Type	State	Usage
Effectivity	Text	Enabled	0
User Status	Text	Enabled	0
Author	Text	Enabled	2619
Mfg Approved By	Text	Enabled	0
Property Compliance (Historical)	Number	Enabled	
Checked Out By	Text	Enabled	
Version	Number	Enabled	
Checked Out	Date	Enabled	
Part Number	Text	Enabled	2644
Number of Attachments	Number	Enabled	
Thumbnail	Image	Enabled	
Rev Number	Text	Enabled	0
Material	Text	Enabled	2281
Category Name (Historical)	Text	Enabled	

The Settings tab shows all of the settings associated with the selected property in the top grid.

State – Indicates whether or not the property gets enabled for indexing and visible to the user.

- **Enabled** the property visible to the user is indexed
- **Disabled** the property is not visible to the user is not indexed

Basic Search – Indicates if a string property gets searched when using the basic search feature.

Searched – the property gets included in the basic search.

- **Not Searched**: the property gets excluded from the basic search.
- **Not Allowed**: the property is not a string value and, therefore, cannot get included in the basic search.

Initial Value – The Initial Value is applied once when the property gets initially associated with an object. The initial value gets applied in the absence of a mapped value.

The initial association occurs in three circumstances:

1. Object gets created (ex: adding a file or creating an item)
2. Assignment to a category that automatically adds the property
3. Manually adding a property to an object

There are two types of Initial Value: static and mapped. The static value is a fixed value and may be any value that is valid for the selected data type. An initially mapped value copies the value from a file or BOM property. Where there are mappings and an initially mapped value defined, the initial mapping gets used if the mappings do not result in an actual value. See Mappings for more details.

List Values – Properties of type Text and Number may provide a list of values for user selection and searching. The administrator may add or remove values from the list at any time.

Removal of a value from the list does not remove the value from any property where that value gets applied. When specifying the value of this property, the user may choose from the list of values. Enter values that are not on the list are allowed. If this property is mapped to read a value from a file or BOM, the imported value is not required to be on the list.

Enforce List of Values – When enabled, this option ensures that the property value is a value that matches one of the values in the List Values.

Requires Value – When enabled, this option ensures that the property has an actual value assigned.

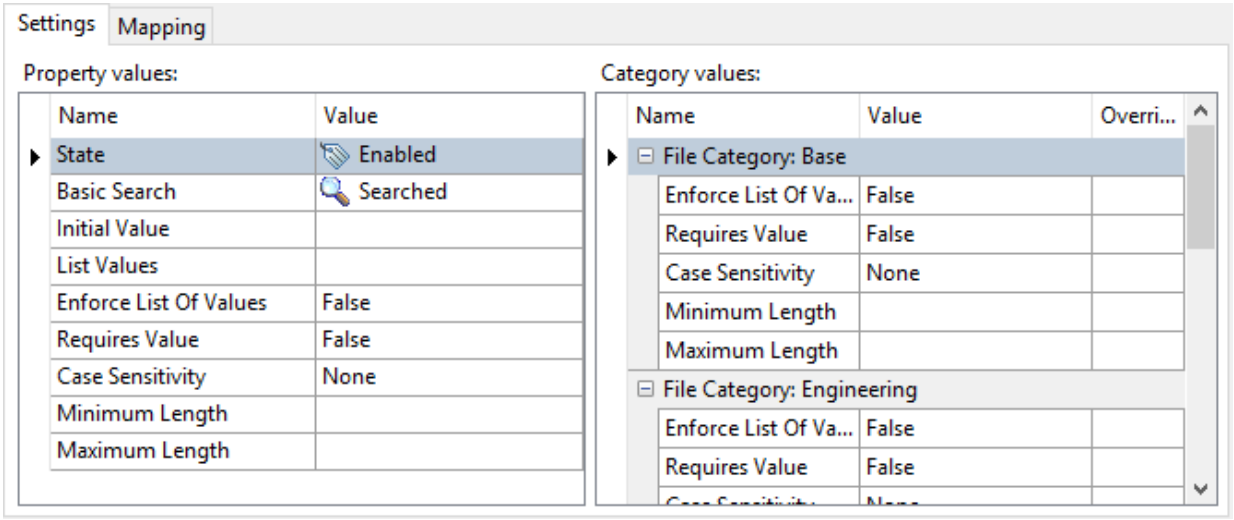
Case Sensitivity – When enabled, this option ensures that the property value matches the specified Case Sensitivity criteria.

- None
- UPPER CASE
- lower case
- Name Case
- Sentence case

Minimum Length – When enabled, this option ensures that the property value has the specified minimum length.

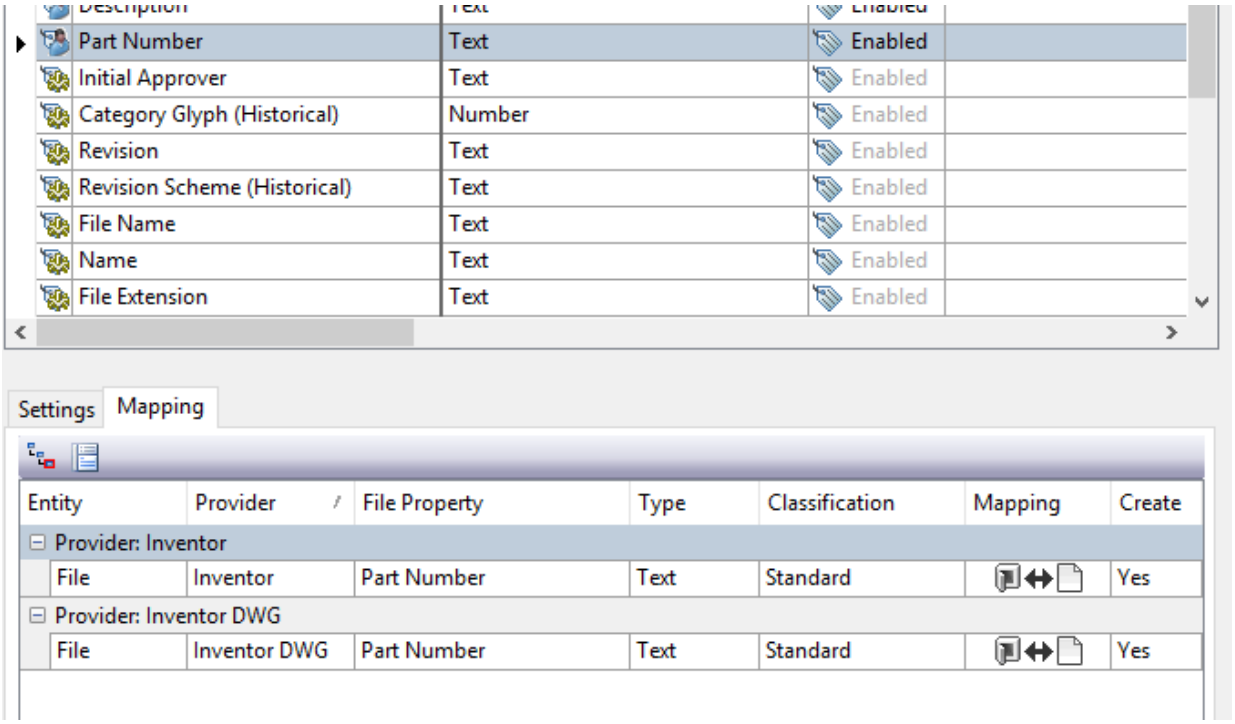
Maximum Length – When enabled, this option ensures that the property value does not exceed the specified minimum length.

The policy values under the *Property Values* column (left side of the dialog) gets applied to all instances of this property except where the category override applies. The *Category Values* allow overrides by category. In the example below, the Requires Value policy has been configured only to be applied when it gets associated with the Engineering Category.



Mappings

The Mapping tab shows all of the mappings associated with the selected property. Property mapping associates a system or user-defined property to a file property. You map file properties to a user-defined properties (UDP) so that values from the file properties get written to the UDPs. Properties can also get mapped so that UDP values get written back to the file property.



This dialog opens in a Read-only mode, so you are unable to make any changes to the selected property or created a new property until you click the Edit or the New button in the menu bar respectively.

Classification

There are essentially two main classifications Standard and Custom. There is third classification None, used when either of the main two are not determined. These classifications refer to the classification of the CSP (file properties). A classification of Standard means that this file property is a standard property for these types of files (Inventor ipt). A classification of Custom means that the file property is a custom property and may only get associated with some of these file types.

Mapping Priority

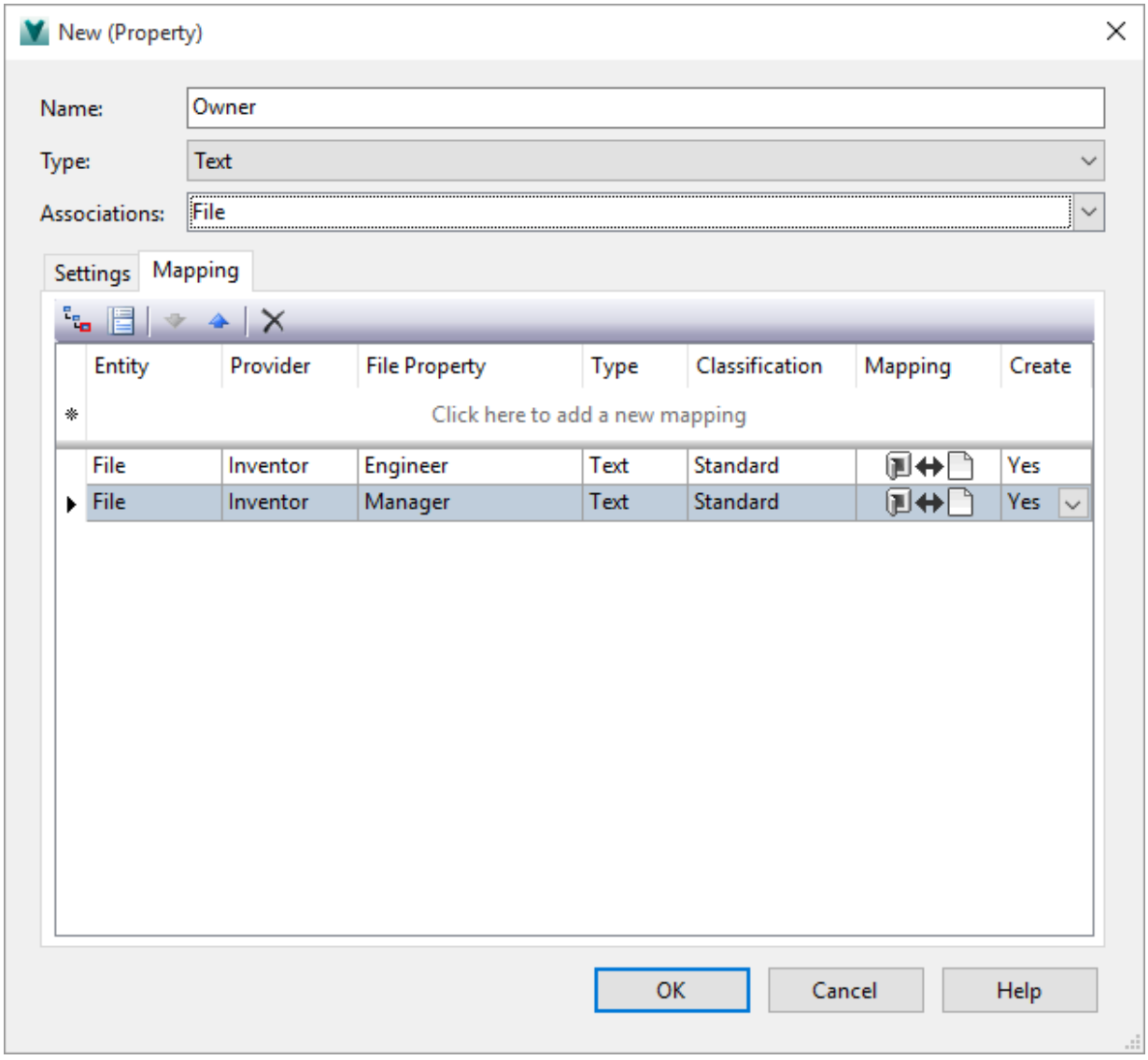
Mapping priority determines the order in which properties get written to a vault object. If more than one property writes to a particular object, the highest ranking property takes precedence.

Note: When both the file and item UDP's get mapped to the same file property, the item UDP takes precedence.

When a file gets added, the provider gets determined, and only the Read mappings for that specific provider are applied and grouped by Provider. When there is more than one mapping associated with a provider, the mappings are processed in priority order until a value is determined.

	Read Only
	Write Only
	Read-Write (default)

The image below shows example mappings. The mappings shown in the top section are the Read mappings Read and Write bi-directional mappings get shown as one row. The read Mappings get listed in priority order. In the image below, the Inventor Provider has two Read Mappings associated with *Owner*. When an Inventor file gets checked in, the Engineer Property in the Inventor file is first inspected. If this property has a value, the value is assigned to *Owner*. If this property does not exist or has no value assigned, the Manager property is then inspected and used if it has a value. If the Manager property results in no value (or there are no Read Mappings), the Initial value gets used. You can modify the mapping priorities by using the up and down arrow buttons in the menu bar.



Mapping Across Data Types

There are four property types: Text, Number, Boolean and Date. The following matrix defines valid property mappings.

	Source Property (File or BOM)				
UDP		Text	Number	Boolean	Date
	Text	Yes	Yes[2]	Yes[1]	Yes[1]
	Number	Yes[2]	Yes	Yes	No
	Boolean	Yes[1]	Yes	Yes	No
	Date	Yes[2]	No	No	Yes

Whenever a mapping gets created between two different property types, there is the possibility of incompatibility. The onus is on the user to input valid values. If an invalid value gets entered in most cases, the equivalence flags the property as nonequivalent because an invalid value results in an empty value getting assigned to the UDP, which gets the same as the source value.

See the exceptions listed below.

1. Mapping Boolean with Text: The supported valid text values are: Yes/No, True/False and 1/0. These values are localized. A string like 'Autodesk' entered in a Text property cannot get transferred to a Boolean property. This property mapping would be flagged as not equivalent.
2. Mapping Text with Number or Text with Date: Works well when all clients and the server are in the same language-locale. With mixed locales, values may convert in a manner that is not intuitive and may produce an undesirable result. Therefore, mapping Text with Number or Text with Date is only recommended when the server and all clients are working in the same locale.

Create Option

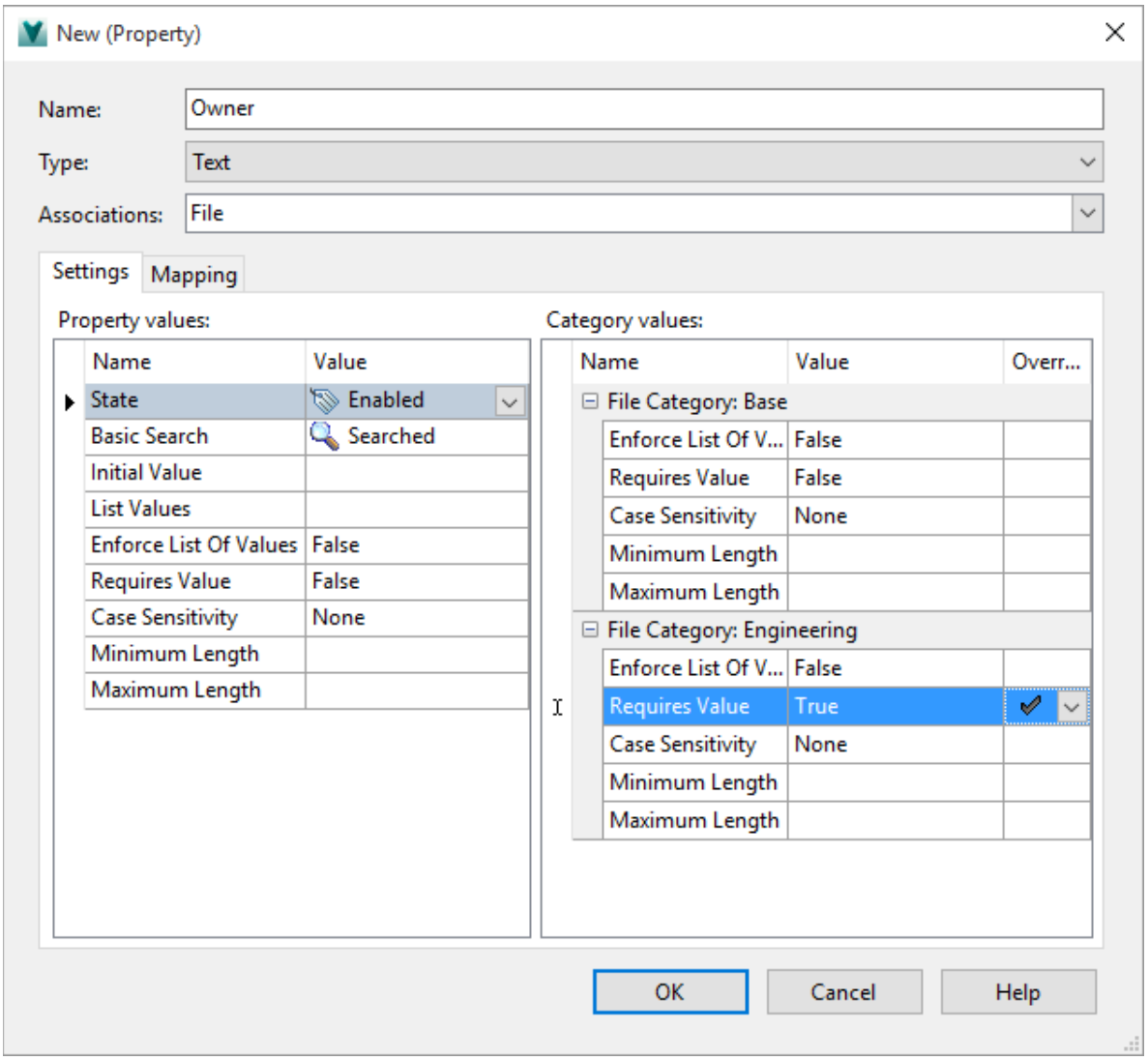
There is one other factor that comes into play with Write Mappings, and that is the Create Option (right-hand side of mapping definition). If the file property does not exist and Create is true, the file property gets created. If the file property does not exist and Create is false, the value does not get written to the file – if there are multiple write mappings, the next mapping in priority order gets processed.

Initial Value as a Mapping

The Initial Value defined in Settings has two types of Initial Value: static and mapped. When read mappings and an Initial value are defined, the initial mapping does not get used if the read mappings do not result in an actual value. If the Initial mapped value does have an actual value, then the initial static value gets used if one is defined.

Property Compliance

The Policy is one of the two tenants of Property Compliance. The other is Property Equivalence, which deals with Mappings discussed in the Mapping section. When these policy settings are configured the property value on the object is evaluated to ensure that it is not in violation of this policy. If in violation, a warning symbol adjacent to this property value is displayed. When a value is in violation of policy, the default configuration for lifecycle transitions does not allow a file or item to be released.

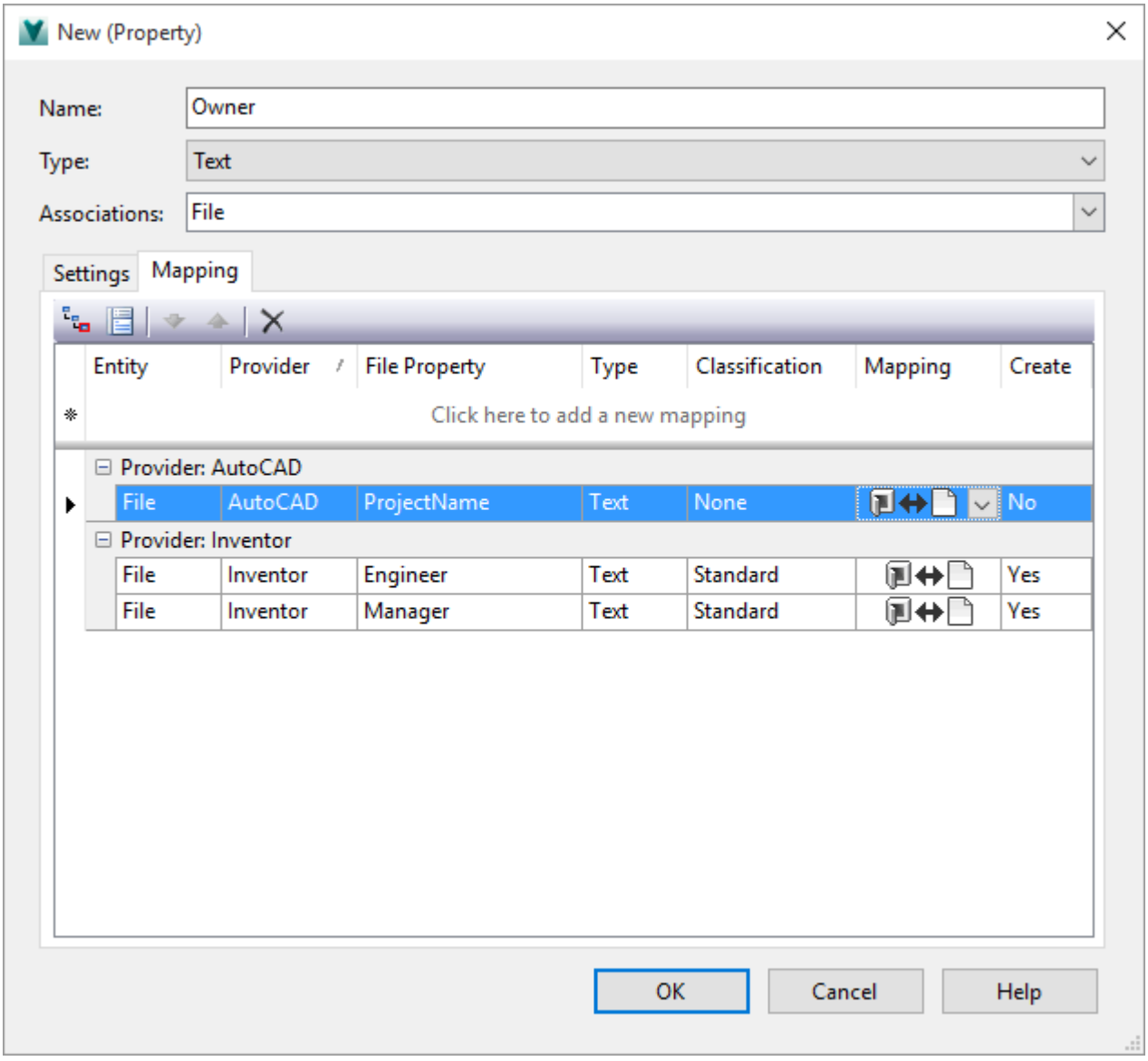


An Administrator can configure Policy for Enforce List of Values, Requires Value, Case Sensitivity, Minimum Length and Maximum Length. These policy settings can also be defined for each Category if required.

When these policy settings are configured the property value on the object gets evaluated to ensure that it is not in violation of this policy. If in violation, a warning symbol adjacent to this property value gets displayed.

Equivalence

Only file objects evaluate Equivalence. Equivalence ensures that the evaluated Read and Write mapping for the UDP and File Property match. Once a mapping is created, equivalence always gets evaluated for the file.



In the above example, if we have an AUTOCAD File, Equivalence ensures that the UDP *Owner* and the file property Project Name have the same value. If in violation, a warning symbol adjacent to this property value gets displayed. When the Inventor file gets evaluated for equivalence, the Engineer file property in the Inventor file is inspected first. If this property has a value, the value is compared with *Owner*. If the Engineer file property does not exist or has no value assigned, the Manager property is

then inspected and evaluated for equivalence. If the Manager property results in no value then no value gets used for equivalence evaluation for the file property – in this case, UDP *Owner* must have no value (or empty) to get classed as equivalent. Equivalence also gets evaluated for Write Mappings.

Property Compliance gets evaluated on all supported objects when the changes to the object are committed to Vault. For example, for a file, this is when it gets added or checked into Vault.

Pending Evaluation

When a Property Definition gets modified by the Administrator that could affect Compliance (e.g. policy setting, mapping), all objects that get associated with this Property Definition are queued for a Property Compliance check. The evaluation gets executed on the Vault Server in the background. When an object gets queued for Property Compliance, the Pending icon displayed.

Property Compliance Icons

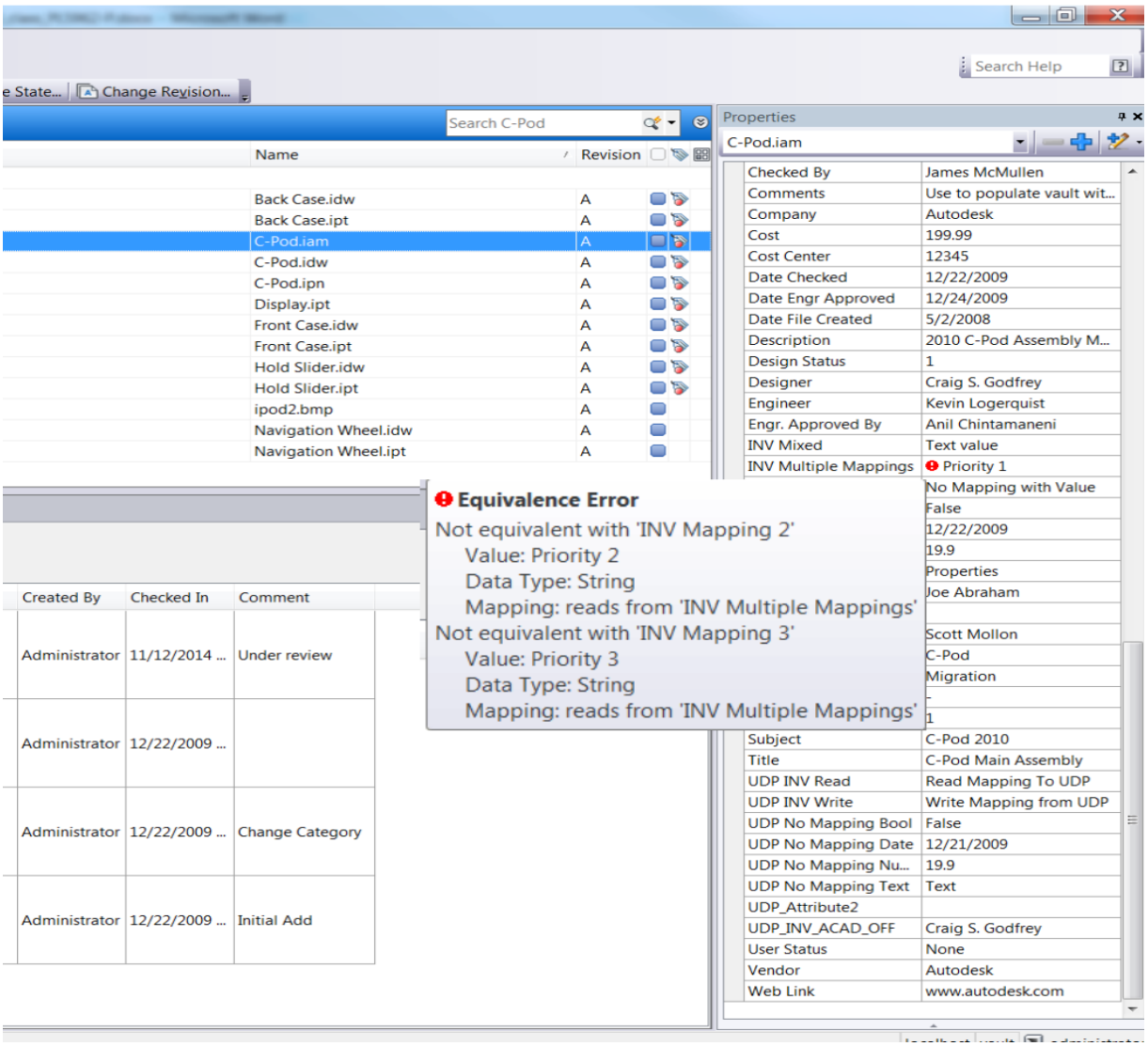
The main view in the Vault client has a Property Compliance column that lists various status icons. The Icons in this column indicate whether a property associated with the file is compliant, non-compliant, pending, or has failed the equivalence evaluation.

No icon	Compliant
	Non-compliant (Policy failure, Equivalence failure or both)
	Pending - Object (e.g. File/Item) is queued for a Property Compliance check
	Evaluation failed

In the image below you can see the Non-Compliant icon where properties are non-compliant. In the main grid, the Non-Compliant icon is displayed for C-Pod.iam. The Properties grid shows the actual properties that are non-compliant. If you hover the mouse over the property, it displays a detailed reason as to why it is non-compliant. The image below indicates that it is a Write Mapping to Inventor file properties “Inv Mapping 2” and “Inv Mapping 3” that are non-equivalent.

A synchronize of Properties for C-Pod.iam resolves the non-equivalence.

Synchronize Properties (under Actions menu in Vault Explorer) executes a write back of properties to the file – a new version of the file gets created.

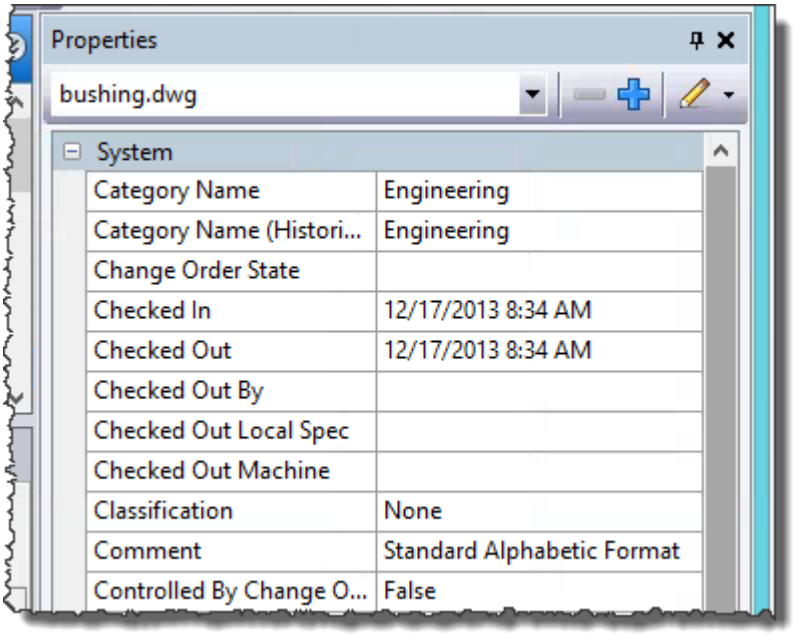


To resolve compliance violations, edit the UDP value to be compliant. For read-only mappings, edit the property in the file using the authoring application or change the policy configuration. To resolve equivalence failures for write mappings, you can edit the property in the Vault client, or you can use the Synchronize Properties command.

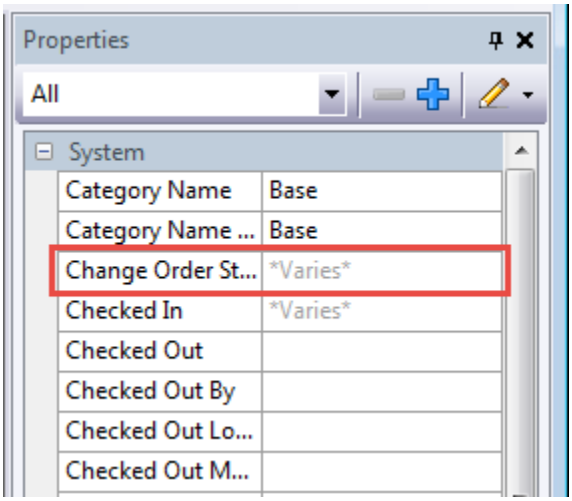
Most Read Mapping equivalence failures usually result from the modification (add, delete or reorder) of a UDP's read mapping. When Read mappings are modified, all files associated with the UDP get queued for a Property Compliance check. After the Property Compliance check, some files may become non-compliant. Read Mapping equivalence failures can get resolved by executing a Property File Re-Index.

Property Grid

The Properties grid appears in the right pane of the Vault client and lists all properties for a selected file. Properties are separated based on whether they are system-defined or user-defined. It displays the name and value of each property associated with the selected file. The grid can be configured to show only the properties you want.



If multiple files are selected and the values for the property is not the same, the property grid shows *Varies* for the value. You can edit this property to set all of the selected files with the same value.



Indexing Properties

When a file gets added to a vault, the file and all of its file properties (file size, date created, and so on) are added to or updated in the vault. New properties added to the vault gets defined before storing the value. Depending on the type of files stored, each vault may have a different collection of properties that it maintains. These properties and values are indexed by the vault for quick retrieval during searches.

Re-indexing scans the selected vault database and extracts and indexes properties from the files in the vault database. The extraction and indexing process uses the latest available iFilters/Content Source Property Providers for the files. Only properties currently set to Enabled and properties that have read mappings are re-indexed. During re-indexing, existing property (e.g., user-defined property) values get updated.

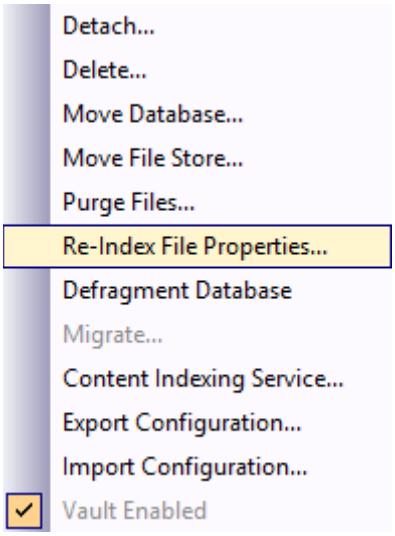
Reasons for re-indexing

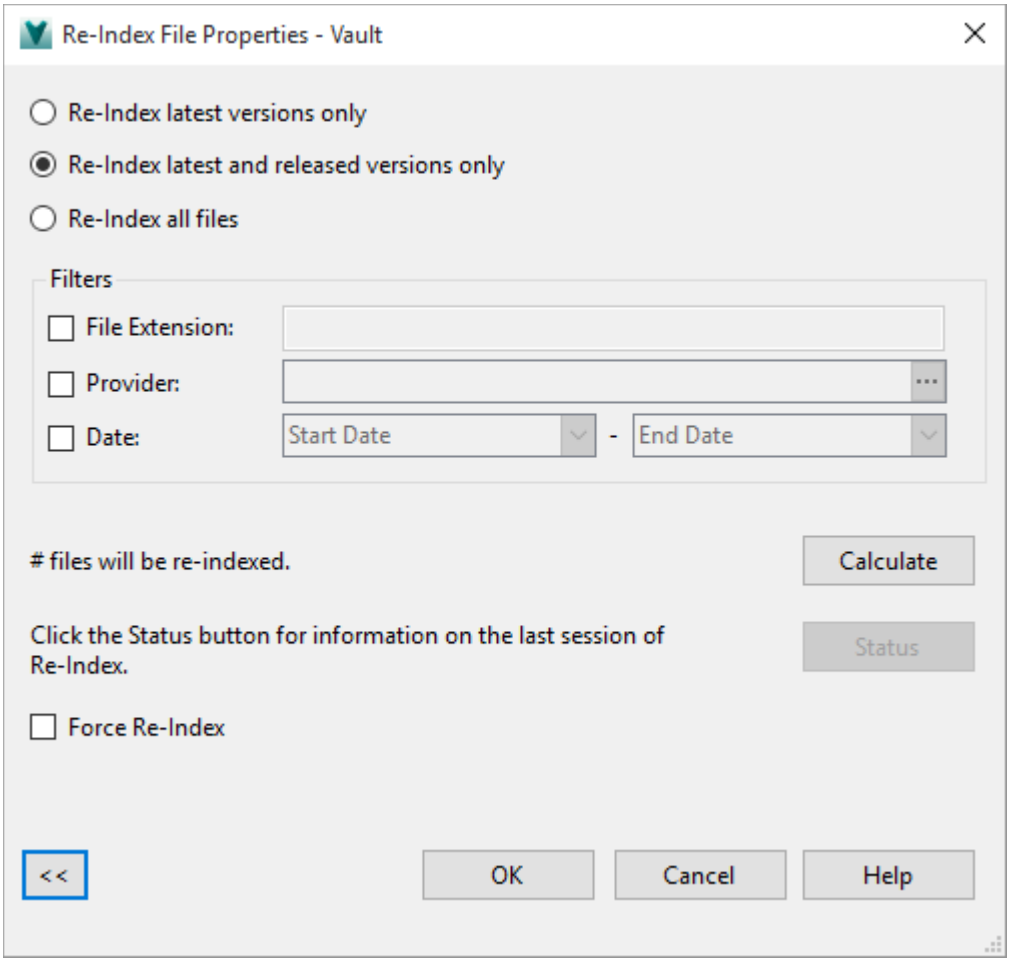
New iFilters or Content Source Property Providers get added to the server that extract more relevant information. When Property definitions get set to Enabled or Read-mappings get modified.

The following are some of the automation property definitions in Vault and set to Enabled when the Vault Server gets installed:

- Author
- Comments
- Keywords
- Thumbnail
- Material
- Part Number
- DWG Creator Name
- DWG Creator Version

Right clicking on a vault from within ADMS Console gives you the ability to Re-Index File Properties:





Filters

You can use filters to limit the number of files that are re-indexed. Following is a list of filters that can be used to refine the re-indexing process:

- **File Extension** – use to re-index files of a particular type.
  Usage: Input file extensions with or without leading dots. Separate each extension by a comma or a semi-colon (with or without spaces).
- **Provider** – use to re-index files based on the iFilter or Content Source Property Provider.
  Usage: Select provider(s) from the pre-defined list of providers.
- **Date** – use to re-index files that get checked in within a specified period.
  Usage: Select the start and end dates to define the date range. The reindexing process indexes files that were checked in during this time frame.

Note: All three filters get used in combination with each other. Use the OR operator between File Extension and Provider filters. Use AND between Date and other filters.

When the Force Re-Index option gets selected, the Vault re-indexes the files whether or not the file requires a re-index. It executes a physical re-index in the vault which means it retrieves the file from file store, index the file and update the File Properties table with the indexed properties. Use the Force option sparingly.

One reason an administrator would use the Force option is when an IFilter (or a Custom Content Provider) gets installed. Vault does not detect that an IFilter after it gets installed and needs to be instructed to physically re-index these files. When indexing the file, the newly installed IFilter gets used to extract the new File Properties. The Force option should always be used in conjunction with the File Extension filter to limit the number of files. You do not want to index all files in a vault accidently.

Another reason an administrator may use the Force option is when an Index Block Attribute gets created in the Vault Server Console. Block Attributes are only extracted (indexed) from the file after they get configured console. So the only way to extract the newly configured Block Attribute for currently checked in files is to execute a FORCE re-index.

Indexing and Server Performance

The re-indexing process is assigned Below Normal priority on the server, allowing users to continue accessing the Autodesk Data Management Server even while files are being re-indexed. If the vault is backed up during the re-indexing process, the re-indexing is paused for the duration of the backup and then automatically restarted once the backup has finished. If a vault is being re-indexed at the same a vault is getting restored, the re-indexing operation stops and all properties re-indexed to that point are replaced by the properties of the restored version.

Note: To run the re-indexing process at a higher priority on the server, use the command line. While running the re-indexing process as a high priority is faster, the priority cannot be changed once the process begins.

When migrating a vault, all property definitions that get set to Enabled remain Enabled. Once migrated, the administrator can disable those property definitions that are not needed. Any new property definition added to the vault following migration is set to Enabled by default. Re-indexing processes files from newest to oldest. If the server gets suddenly rebooted or loses power, re-indexing starts

automatically on the last file processed when the server is brought back online. When re-indexing properties in a replicated environment, the re-index operation processes files located on the server where it started.

Indexing Block Attributes

When adding AutoCAD-based files to the vault, you can map block attribute properties to vault properties. Properties in AutoCAD blocks get indexed by entering the block name with the following process.

Launch the Autodesk Data Management Server Console.

Select Tools -> Index Block Attributes.

In the Index Block Attributes dialog box, click New.

Note: If you prefer to use the prompt instead of an attribute tag, you can choose to do so. To use the prompt instead of the attribute tag when it is available, enable the *Extract attribute prompt when available* checkbox. If this check box is turned on and there is no prompt, the attribute tag is used.

Re-Index File Properties from the Command Line

Run Connectivity.ADMSConsole.exe from a command line with the following switches:

Command	Details
-Oreindexfileproperties	The operation being performed (required).
-Ndatabasename	The name of the database to be re-indexed (required).
-VUusername	Vault administrator account user name (required).
-VPpassword	Vault administrator account password (required).
-TIP	Re-indexes only leading versions of files (optional). This is the default setting. If -TIP is used (or not specified), only the latest (tip) versions are re-indexed.
-ALL	Re-indexes all versions of files (optional).
-TIPREL	Re-indexes only leading and released versions of files (optional). By default, only the latest versions (tip) are re-indexed. If -TIPREL is used only the latest versions, the released versions, and pending released versions are re-indexed.
-EXTextensions	File extensions filter (optional). Enter values separated by ',' or ';' without spaces. With or without '*' or '.'
PROVIDERcontentsourceproviders	Content source providers filter (optional). Enter values separated by ',' or ';' without spaces. Use the following system name for each provider: • AutoCAD – AutoCAD_PropertyProvider • AutoCADElectrical – AutoCADElectrical_PropertyProvider • InventorDWG – InventorDwg_PropertyProvider • Inventor – Inventor_PropertyProvider • Revit– Revit_PropertyProvider • Office 97-2003 Documents – Office_PropertyProvider • Office Documents – OfficeOpenXml_PropertyProvider • All Files – IFilter_PropertyProvider • AutoCAD C3D – AutoCADC3D_PropertyProvider • Sheet Set Data – SSM_PropertyProvider

-FROMstartdate	Start date filter (optional) Use mm/dd/yyyy format
-TOenddate	End date filter (optional) Use mm/dd/yyyy format
-FORCE	Force all files or latest files to be cracked open and re-indexed (optional).
-STOP	Terminates the re-indexing operation (optional). Once stopped, re-indexing cannot be started from the point at which it left off.
-NORMAL	Bumps the process priority from "Below Normal" to "Normal" (optional). The priority cannot be changed once the re-indexing operation is initiated. If the -NORMAL switch is used, re-indexing will complete sooner; however, server performance will degrade due to the higher priority of the operation.

Examples

Connectivity.ADMSConsole.exe -Oreindexfileproperties -NVault -VUadministrator -VPadmin

Connectivity.ADMSConsole.exe -Oreindexfileproperties -NVault -VUadministrator -VPadmin -STOP

Connectivity.ADMSConsole.exe -Oreindexfileproperties -Nvault -VUadministrator -VPadmin -TIP -FORCE -EXTipt,iam -PROVIDERAutoCADElectrical,OfficeOpenXml -FROM5/10/2011 -TO5/20/2011

Search Indexes (Lucene)

Search indexes are properties that are extracted from the vault database and placed into a more efficient system for searching. These indexes get used when a user performs a basic or advanced search in any of the Vault clients. Instead of searching whole property values this search system indexes tokens.

Tokens are individual chunks of a property value that allow the system to locate files based on pieces of information. This type of searching allows for quicker search results and reduces the impact to the system when searching vaults containing thousands of property values.

Fine tuning searches

By far one of the most important but often misunderstood areas of Vault functionality is searching. In addition to the basics of search tokens and search properties detailed in Searching Vault, we can also manipulate the range of the Vault searches by tweaking the web.config file by modifying the Lucene search slop factor. The slop factor determines how many positions may occur between any two terms in your search phrase and still be considered a match, from an exact match of your search string to some combinations of this string.

The slop is, in fact, an edit-distance, where the units correspond to moves of terms in the query phrase out of position. For example, to switch the order of two words requires two moves (the first move places the words atop one another), so to permit re-orderings of phrases, the slop must be at least two.

In simple terms, the slop factor decides how out of order search tokens are permitted to be before they get excluded from search results. By default, more exact matches are given precedence but the total number of search results can be directly affected by this value.

How to control the slop factor

From the Web.config file (C:\Program Files\Autodesk\ADMS <edition> 20XX\Server\Web\Services) search for slop factor – you should then find the following 2 lines:

```
<!-- slop factor provided to Lucene search -->
<add key="SearchSlopFactor" value="10" />
```

Modifying this value on the ADMS server changes the results returned by a given search and depending on what it is you are trying to find, can return a more refined or open set of search results.

The slop value by default is set to “10”, you can, however, set this as low as “0” for exact matches only or increase this range to whatever you desire. It is important though to keep in mind how this affects the user search experience, too low and users may become frustrated at not finding the correct data, too high and users will get too many results.

Example 1: Say I am searching for file “A-055401-321.ipt”, this would be represented by six (6) search tokens:

If I search for file name contains A-055* with slop factor = 10 (Default) I could, for example, receive any number of additional results similar to the following that meet the slop factor settings:

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt
- B-055401-321-A.ipt
- B-321-055401-A.ipt

If I now search for file name contains A-055* with Slop Factor = 6 the same results yields because the tokens are still within the edit distance set by the slop factor.

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt
- B-055401-321-A.ipt
- B-321-055401-A.ipt

However, search for file name contains A-055* with Slop Factor = 4 reduces this to 5, the tokens in B-055401-321-A.ipt are spread too far apart to meet the slop restriction or edit distance. Starting from the first dash, we would have to move "A" 5 positions to match A-055*.

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt
- B-321-055401-A.ipt

Search for file name contains A-055* with Slop Factor = 2 now returns only 4 results (Note again this is the minimum value for the system to return search tokens out of order e.g. file name contains –A055*). B-321-055401-A.ipt gets omitted as the "A" would need to move 3 positions to match search phrase A-055*.

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt

Finally, if we search for file name contains A-055* with Slop Factor = 0 only exact token matches are found

- A-055401-321.ipt
- A-055401-321-B.ipt

Example 2: What happens if we repeat this process searching for A055* (removing the dash and reducing tokens)

If I search for file name contains A055* with slop factor = 5 I would, for example, receive the following, as we have effectively reduced the number of positions required to move the tokens for a match the slop factor does not need to be as high to return 6 results:

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt
- B-055401-321-A.ipt
- B-321-055401-A.ipt

If I search for file name contains A055* with slop factor = 3 I would, for example, receive 5 results, once again the edit distance is shorter than it was in the previous example but the results are the same with B-055401-321-A.ipt getting excluded:

- A-055401-321.ipt
- A-055401-321-B.ipt
- A-321-055401.ipt
- A-321-055401-B.ipt
- B-321-055401-A.ipt

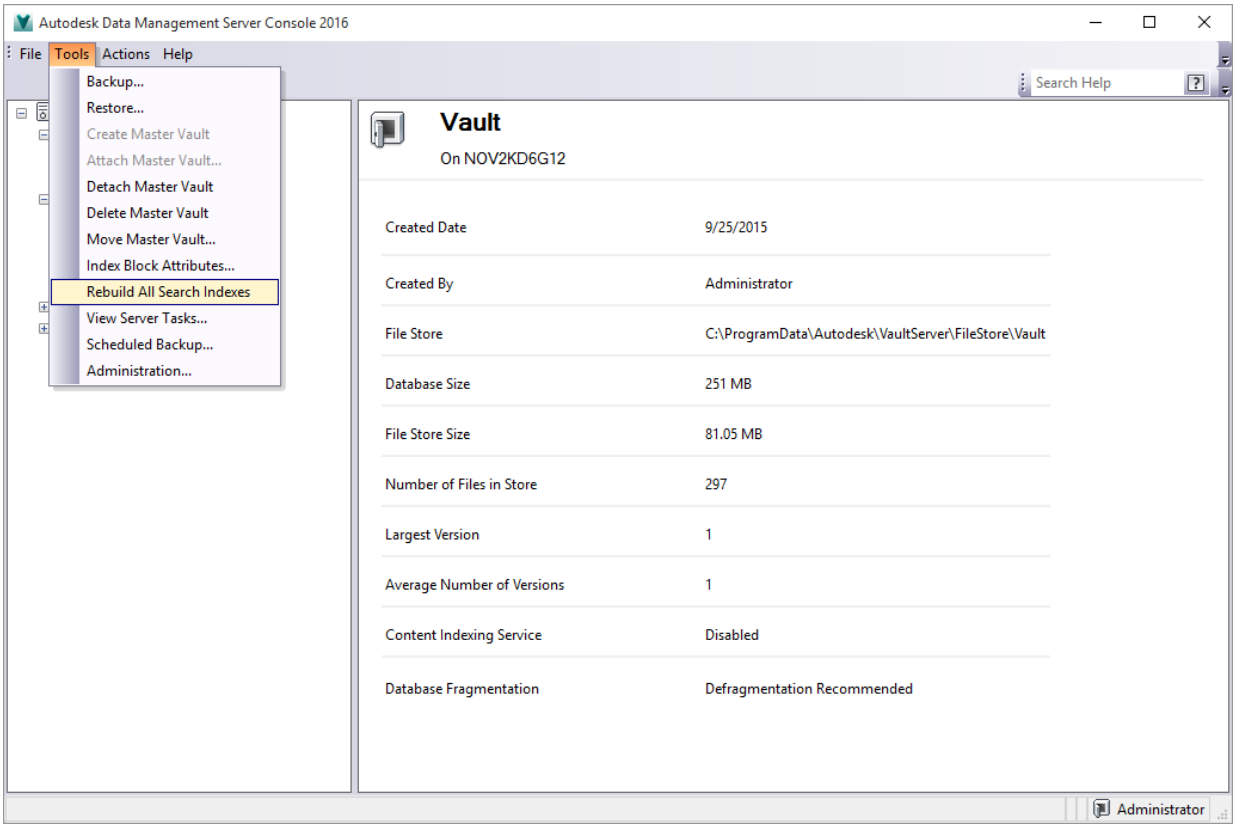
If I search for file name contains A055* with slop factor = 1 there are 2 results.

- A-055401-321.ipt
- A-055401-321-B.ipt

However now with the "-" character removed, my search for file name contains A055* with slop factor = 0 yields NO results – there are no exact matches in this case.

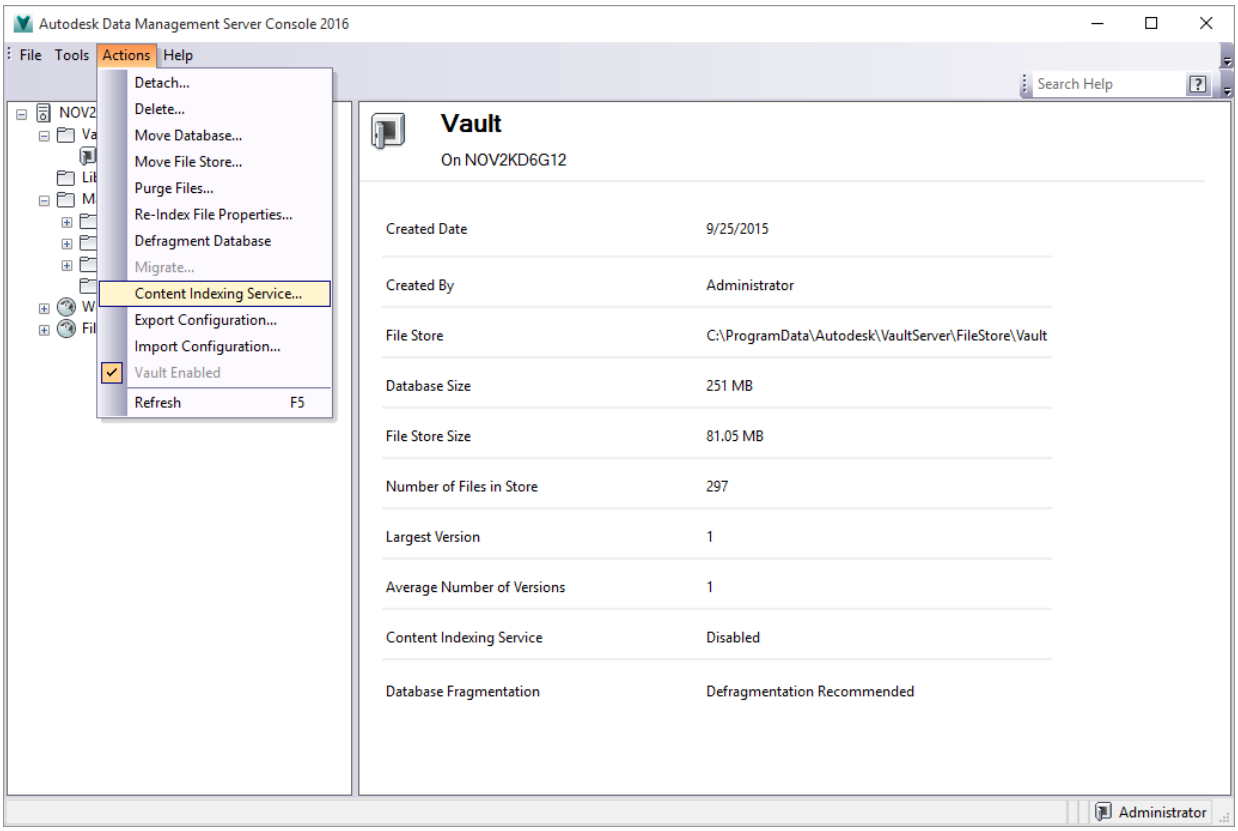
Rebuilding the Search Indexes

It is rare that you have to rebuild the search indexes. However, if they ever become corrupt, you can rebuild them by clicking Rebuild All Search Indexes in the Tools menu of the ADMS console.



Full Content Indexing

Extracts the full-text contents of existing files in the vault, allowing full content searching. This process only runs if full-content searching is enabled. If full-content searching is enabled and you attach a vault, this process gets initiated. If anything interrupts this process, it continues from where it left off at once it restarts.



Lifecycle Administration

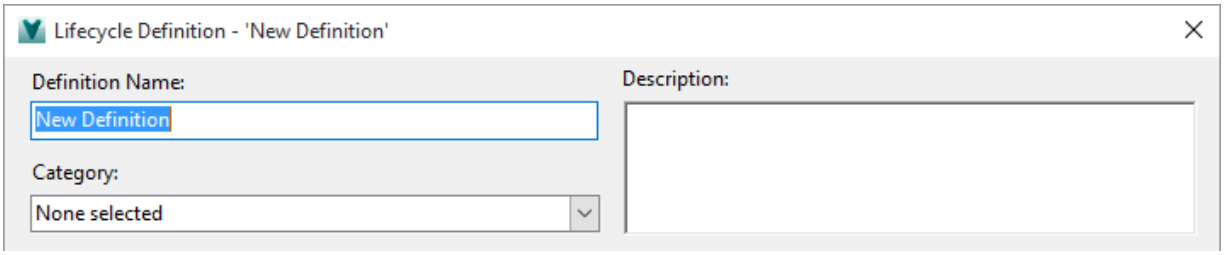
A lifecycle describes the process or stages an object passes through from innovation to production or construction. Lifecycles definitions are created by administrators in the Behaviors tab in the Vault Setting dialog.

Vault contains server predefined lifecycles that can be used or reconfigured to meet your needs. These are:

- Basic Release Process
- Flexible Release Process
- Simple Release Process
- Long Lead Time Release Process
- Long Lead Time Release Process with Change Order
- Item Release Process
- <None>

There is a special out-of-the-box lifecycle definition called *<None>*. It is always available to all objects and cannot be edited, copied, deleted, and it has no states. Its purpose is to allow objects to use other category behaviors (i.e., revisions) without having to participate in lifecycles hence it gets referred to as the "opt-out" scenario.

Only users with the *Document Manager Level 1* role assigned to them can use the Change State command to change the lifecycle state of an object.



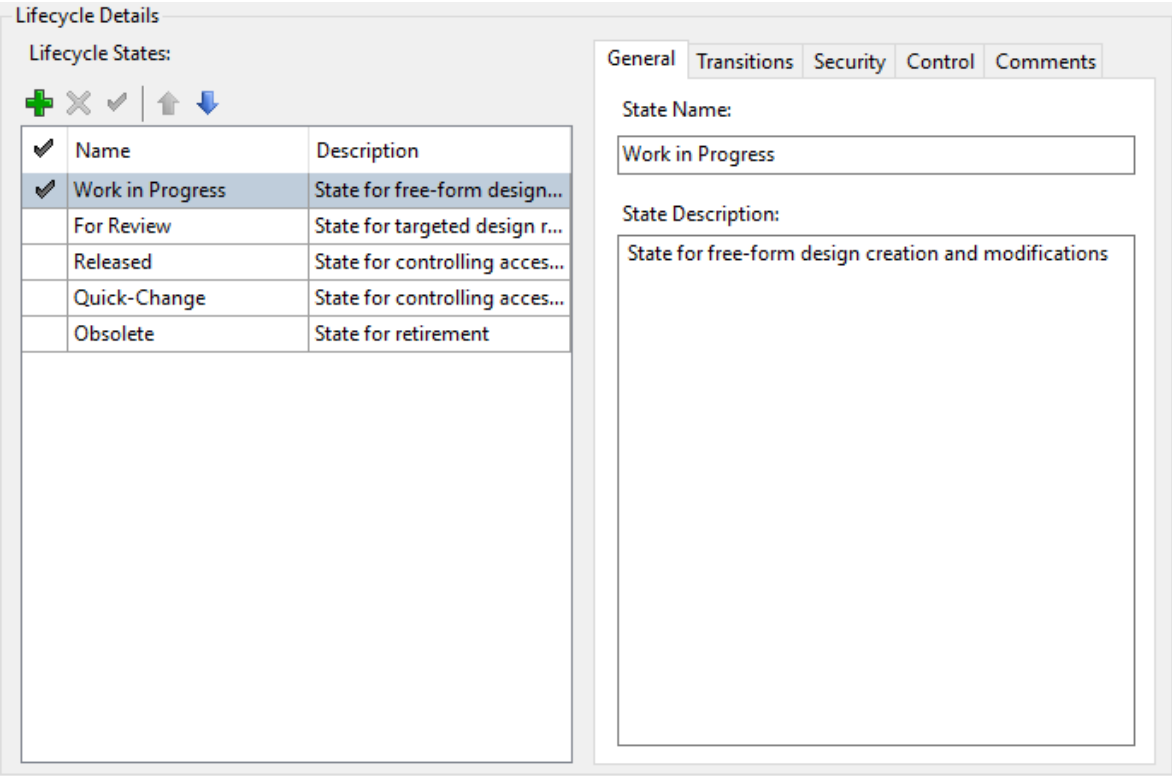
A definition name is required, must be unique and cannot contain special characters. The definition name and description are limited to 255 characters (i.e. \ / : * ? " < > |). The category drop-down allows users to select the category for which this definition is available. If the definition is not assigned to a Category, it is not available to be used on an object.

Lifecycle States

A lifecycle state is a reflection of an object status within a discreet process, typically when managing design data through to manufacture or build.

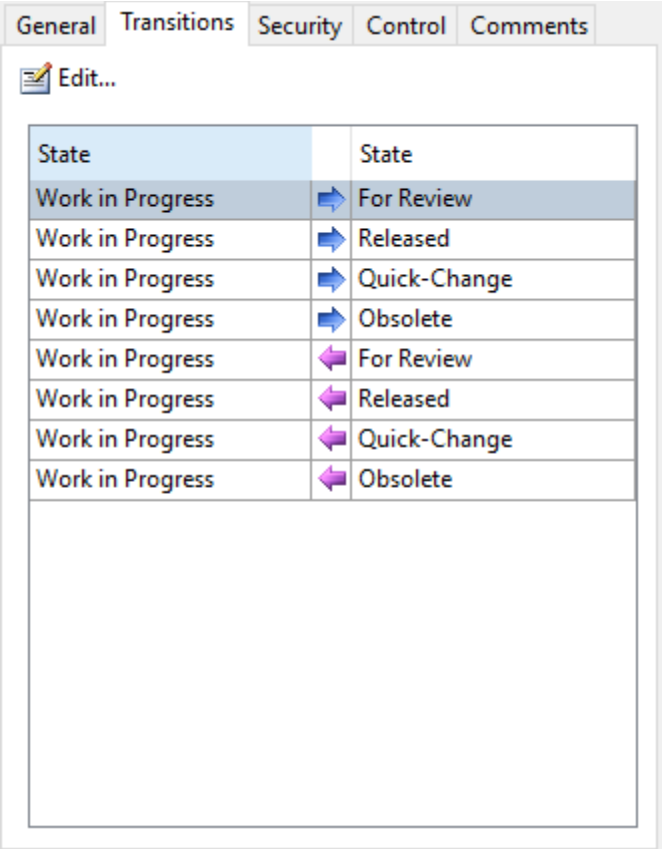
Some examples of state names that can be applied to describe the status:

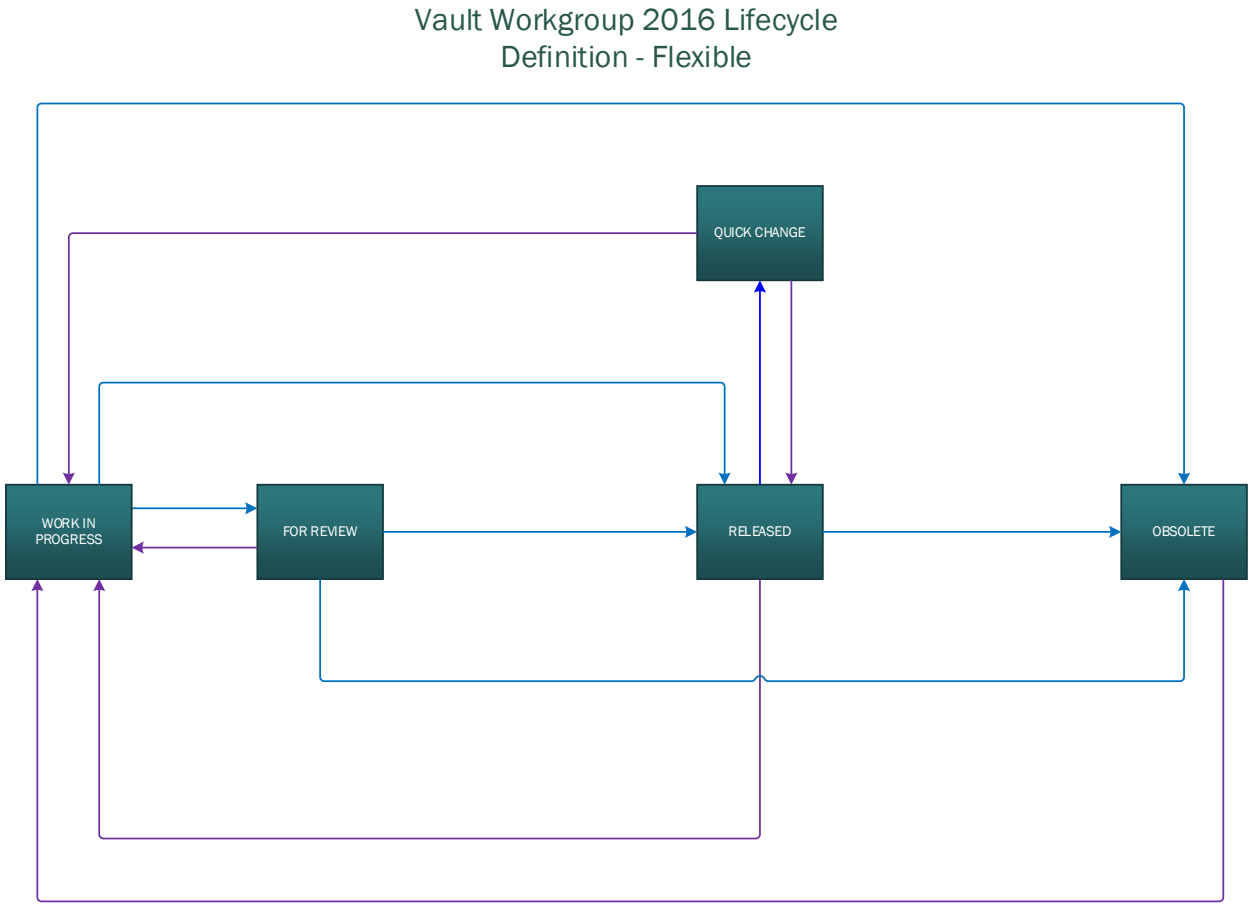
- Work In Progress
- For Review
- Released
- Quick Change
- Obsolete



Lifecycle Transitions

Transitions are what allows or denies an object to move from one state to another. A transition's security setting defines who can move the object from one state to the next and the paths available for each transition.



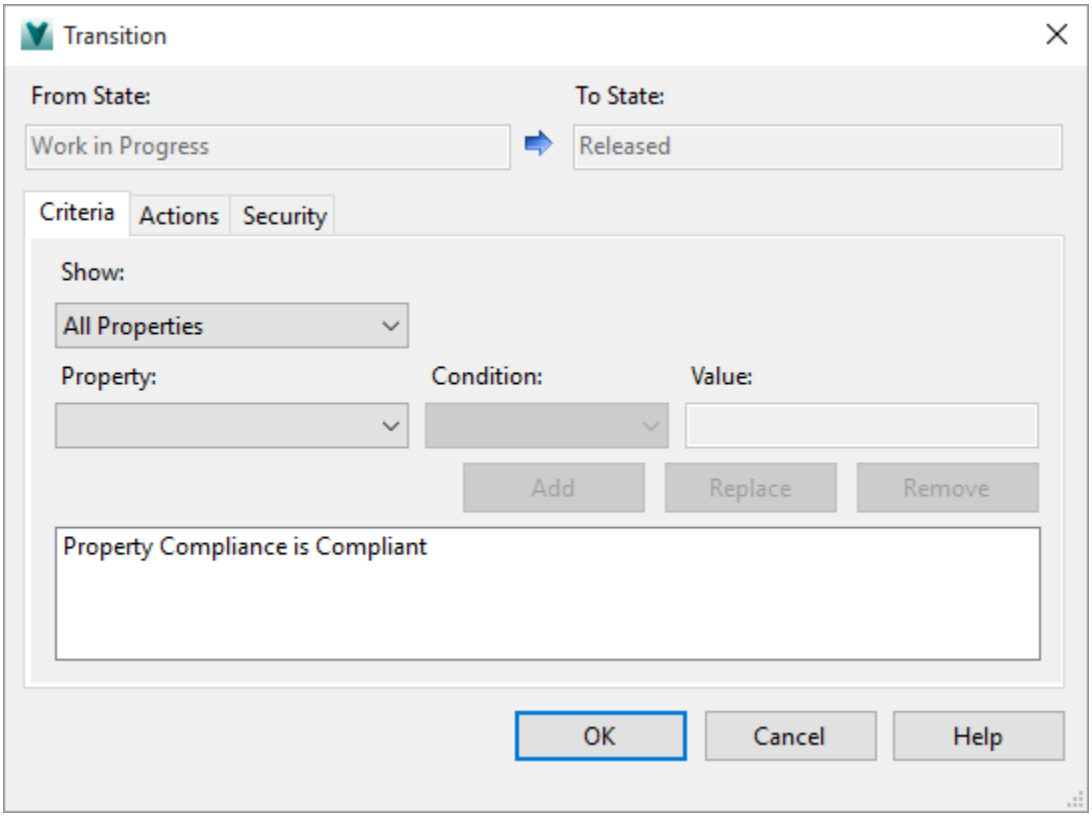

Vault Workgroup 2016 Lifecycle
Definition - Flexible
QUICK CHANGE
WORK IN PROGRESS
FOR REVIEW
RELEASED
OBSOLETE

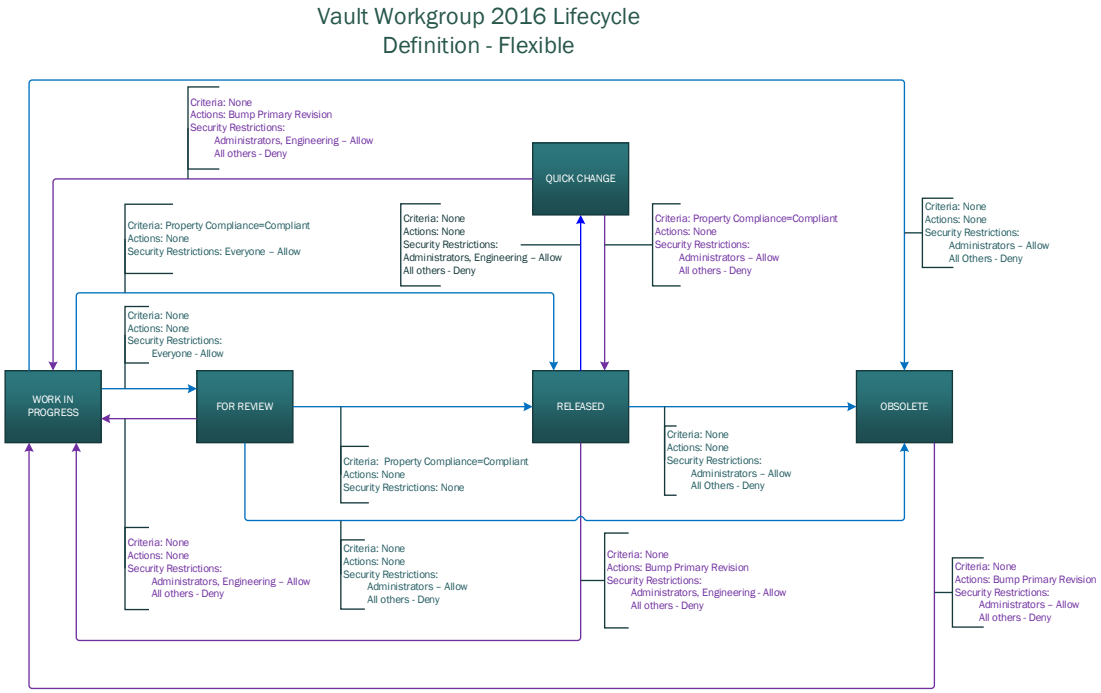
Transition Criteria

Criteria allow the administrator to configure a transition such that it does or does not get allowed based on properties of the object that a state change is intended. The Show drop-down is a filter to pare down the list of properties to make it easier to find object-type-specific properties (file, folder, item). Then a property for the object is selected, a condition and value. The property criteria can then be added using the Add button. Several criteria may be used and they may also be replaced or removed.

Note: The Value field is not case-sensitive, and changes made get committed when the OK button gets clicked. If the property configured is not associated with the object type that the state change occurs for then the criteria get ignored.

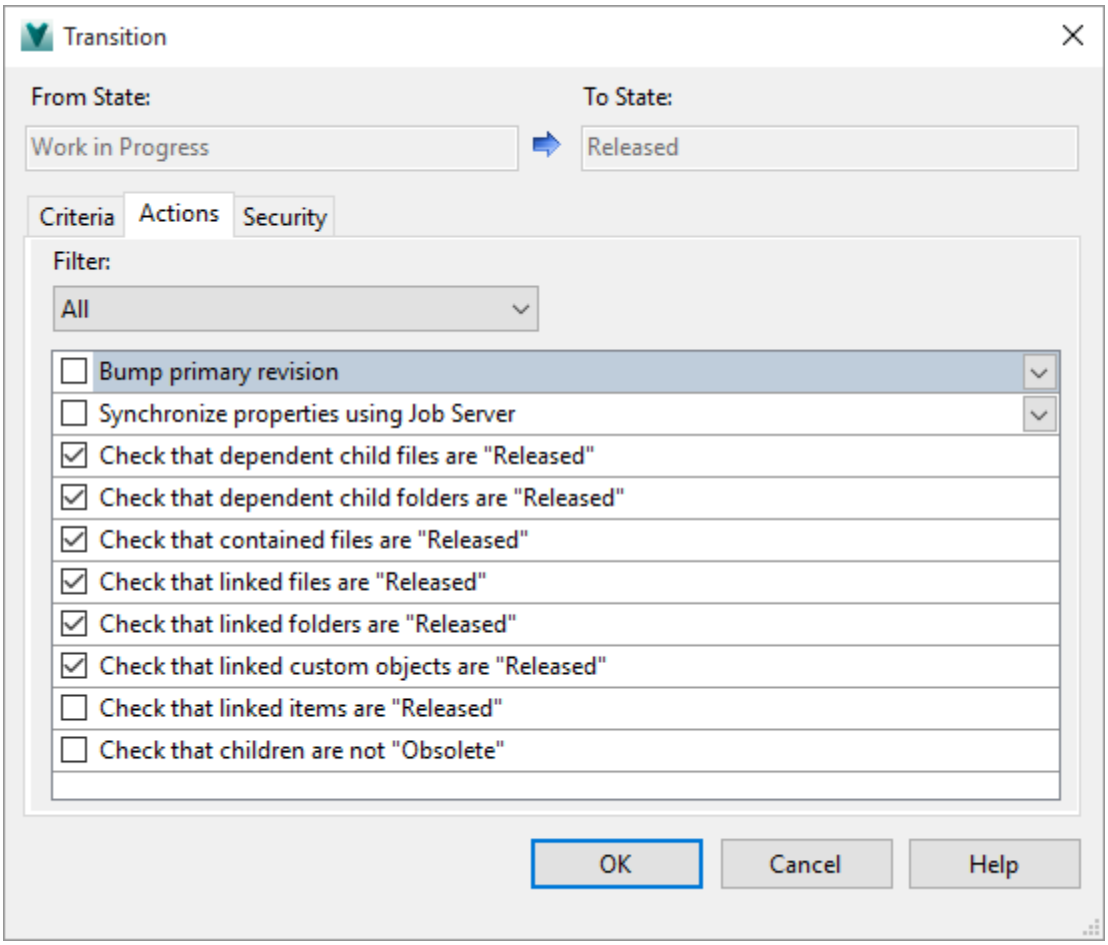
It is possible that the properties of objects are not up to date in that at some later time that object is configured to have a new or different properties. For example, a file is added to the vault and at the time the file was added the properties configured for files consist of some properties. At some later time, the administrator configures the system such that files should have a new property. The new property gets effective only for newly created files, thus leaving the existing files without the property. If transition criteria are used that makes use of the new property, then the criteria check fails for the files that do not have the new property.




Vault Workgroup 2016 Lifecycle
Definition - Flexible
Criteria: None
Actions: Bump Primary Revision
Security Restrictions:
Administrators, Engineering – Allow
All others - Deny
QUICK CHANGE
Criteria: Property Compliance=Compliant
Actions: None
Security Restrictions: Everyone – Allow
Criteria: None
Actions: None
Security Restrictions:
Administrators, Engineering – Allow
All others - Deny
Criteria: Property Compliance=Compliant
Actions: None
Security Restrictions:
Administrators – Allow
All others - Deny
Criteria: None
Actions: None
Security Restrictions:
Administrators – Allow
All Others - Deny
Criteria: None
Actions: None
Security Restrictions:
Everyone - Allow
WORK IN PROGRESS
FOR REVIEW
RELEASED
OBSOLETE
Criteria: Property Compliance=Compliant
Actions: None
Security Restrictions: None
Criteria: None
Actions: None
Security Restrictions:
Administrators – Allow
All Others - Deny
Criteria: None
Actions: None
Security Restrictions:
Administrators, Engineering – Allow
All others - Deny
Criteria: None
Actions: None
Security Restrictions:
Administrators – Allow
All others - Deny
Criteria: None
Actions: Bump Primary Revision
Security Restrictions:
Administrators, Engineering - Allow
All others - Deny
Criteria: None
Actions: Bump Primary Revision
Security Restrictions:
Administrators – Allow
All others - Deny

Transition Actions

Actions allow the administrator to force certain things to occur when a state change is successful, restrict the state change based on the state of related objects or check that objects are up to date.



Actions intended for certain object types result in no action if the state change is for another object type. For example, Check that dependent child files are released has no effect on a state change transition occurring for an item. Use the Filter drop-down to show actions specific to the selected object. To enable an action simply check the checkbox to the left of it.

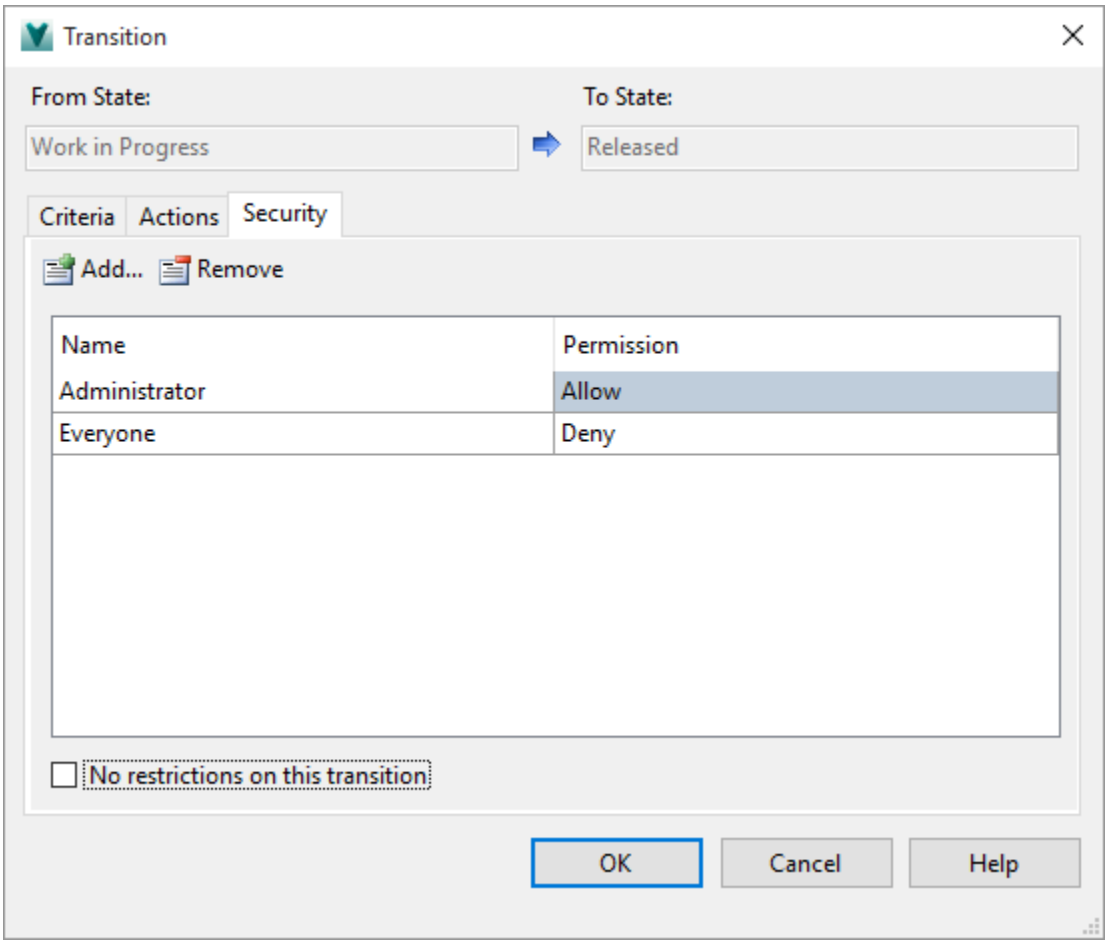
Release Restrictions

Check that dependent child files are released	Verifies that state-dependent children are in a released state.
Check that dependent child folders are released	Verifies that the sub-folders of a project folder are in a released state.
Check that contained files are released	Verifies that all files contained in a project folder are in a released state.

Check that linked files are released	Verifies that all linked files in a project folder or custom object are in a released state.
Check that linked folders are released	Verifies that all linked folders in a project folder or custom object are in a released state.
Check that linked custom objects are released	Verifies that all linked custom objects in a project or custom object are in a released state.
Check that linked items are released	Verifies that all linked items in a project folder or custom object are in a released state.
Check that children are not obsolete	Verifies that children objects are not in an obsolete state. The parent is not released if any child items are obsolete.
Check that child items are released	Verifies that children items are in a released state.
Check that associated item file links are up to date	Verifies that all associated item file links are current. Click the ellipses (...) button to specify only certain item file links: • Primary File Links • Secondary File Links • Standard Component Links • Primary Subcomponent File Links • Secondary Subcomponent File Links • Design Documents
Check that associated item file links are released	Verifies that all associated item file links are in a released state. Click the ellipses (...) button to specify only certain item file links: • Primary File Links • Secondary File Links • Standard Component Links • Primary Subcomponent File Links • Secondary Subcomponent File Links • Design Documents

Transition Security

Transition security determines who can transition an object from one state to the next. The administrator adds users or groups to the security tab and set whether to allow or deny the users or group of users to perform the transition. You can disable a transition by denying everyone the permission to make the transition.

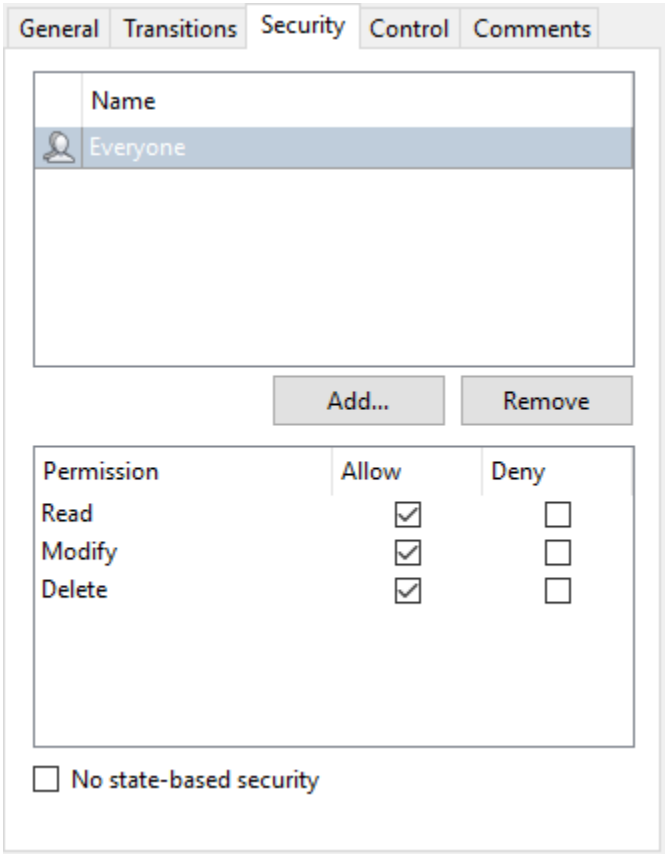


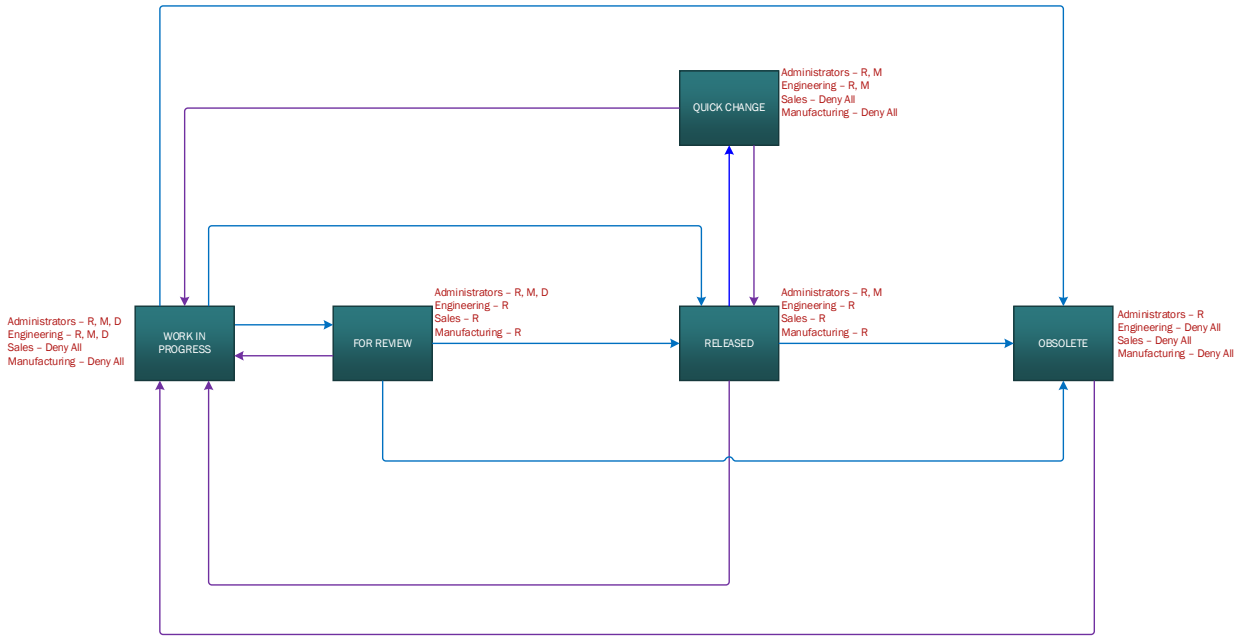
Lifecycle Security

A state can contain security settings to control access to an object. The security controls who can read, modify and delete a file. It also controls whether files can get purged. The Security tab is used to define the security that gets applied to objects when entering the state. Modify the access control list (ACL) accordingly to set the security of the object.

Note: Making security changes does not affect objects that are already in the state. A state change must occur for the security to be applied.

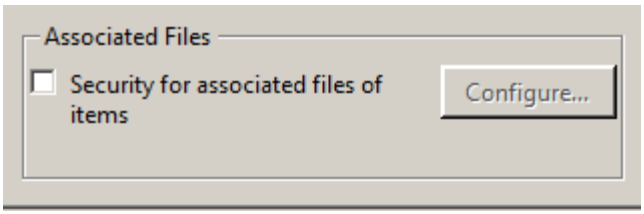
The *No state-based security* checkbox clears out the ACL. If the administrator clears out all users and groups and clicks Apply or OK the result ends up being as if the *No state-based security* is checked.





Applying Security to (Item-related) Associated Files

When entering a state for an item the security of the Item, associated files may be changed as well. To configure the security for Item associated files check the *Security for associated files of items* option and click the Configure button.



The Security for Associated Files of Items dialog has the following options:

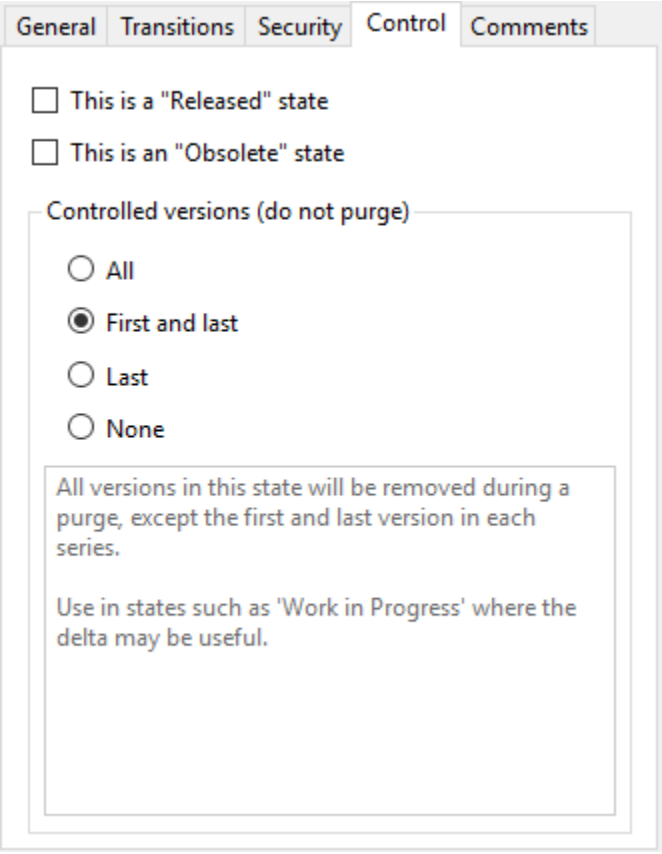
- Apply item security to associated files
  When selected, the Access Control List settings for the item for this state are also applied to the associated file.
- Apply custom security to associated files
  When selected, administrators can set an Access Control List that is different from the one applied to the item for that state.
- Clear security override from associated files
  When selected, if there is a current override Access Control List on the associated file, the security override is removed when the item enters this state.

Lifecycle Control

The Control tab contains the option to declare the state an Obsolete or Release State. These selections help determine if the object in these states gets purged or not. The system always retains at least one version of a file with a released state.

Several options for files NOT to purge

- **All** - No versions in this state get removed when a purge gets performed. This option is recommended for states where not many versions get created or where each version is critical.
- **First and Last** - All versions in this state get removed during a purge, except the first and last version in reach series. Use this option for states where the changes between the first and last file are useful.
- **Last** - All versions in this state get removed during a purge, except the last version in each series. This option gets recommended for states where a record that the file was in the state is important.
- **None** - No version in this state gets retained after the purge gets performed.

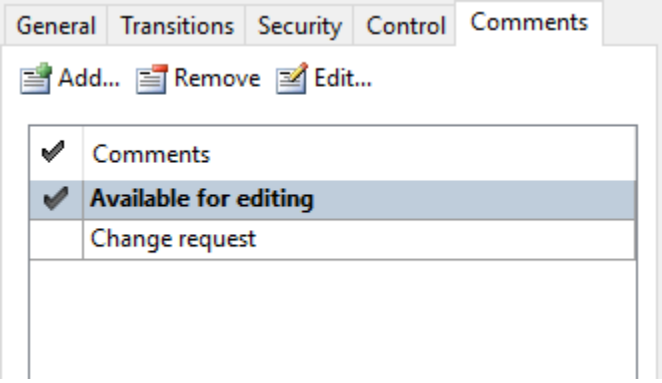


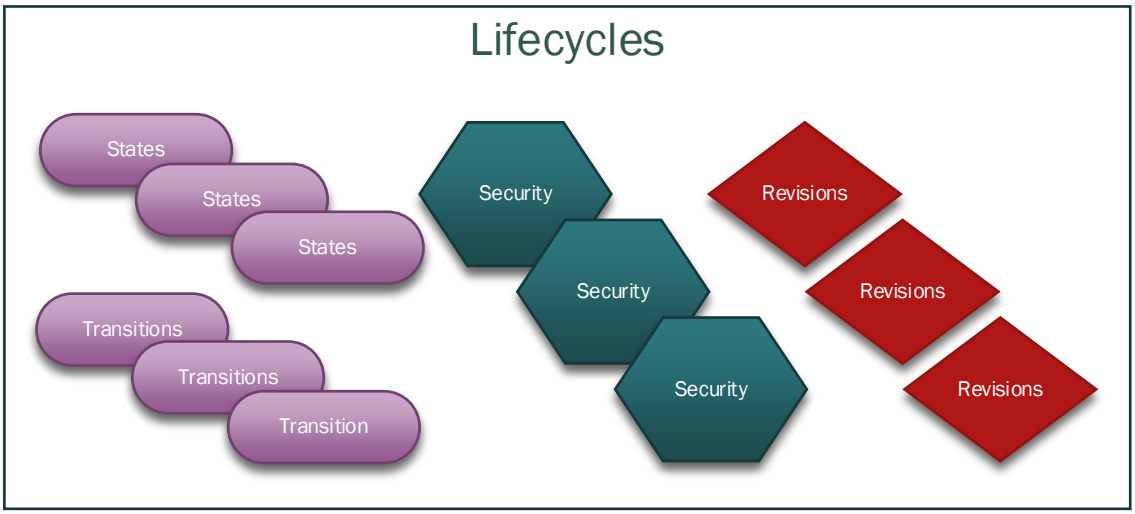
Each version of the file is marked purgeable as it gets created. The good side to this method is that Vault can now purge more versions. For example, if a user creates 20 versions while in WIP we can purge all but the last one (depending on the settings). The purge can also happen much quicker because there are

no calculations (based on relationships) needed. A version is purgeable, or it is not, it is that simple. The downside to this is that a user could set a certain state to purge all versions within that state.

Lifecycle Comments

The Comments tab allows the administrator to create multiple pre-defined comments for each a state. The comments are visible to users when they executed the Change State command and selected by clicking the drop-down error in the Change State dialog.





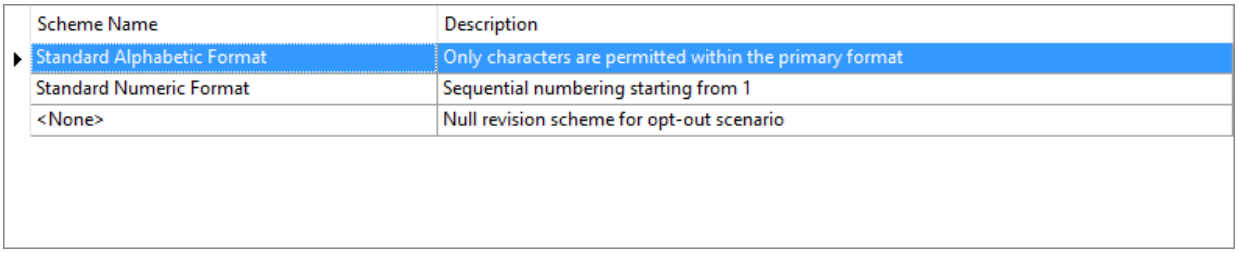
Revision Administration

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. Revisions are the primary key to understanding whether you are working with the latest or correct record.

A revision scheme denotes the progressive values that get assigned to the file. The Revision Scheme can be a numeric, alphabetic or a mixture; traditionally it is from 0-1 or A-Z with some variations. Revision schemes denote major and minor iteration changes.

Revision Management

Vault comes with two pre-defined revisions that are available for you to use. The Standard Alphabetic Format assigns an alpha character as the primary scheme and numeric values for the secondary and tertiary schemes. The Standard Numeric Format assigns a numeric character as the primary, secondary and tertiary schemes.

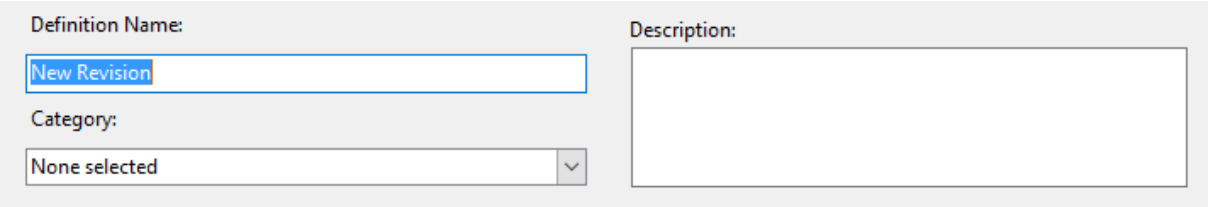


Import Revision

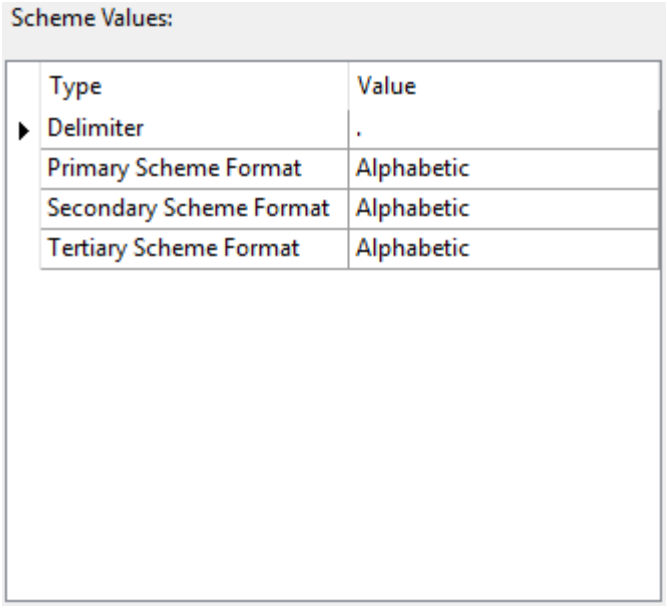
An administrator can import a pre-defined revision scheme. A scheme format gets imported from a text file with each revision character on a separate line in the text file. Once imported, you can use the new scheme in a revision definition.

Creating a New Revision

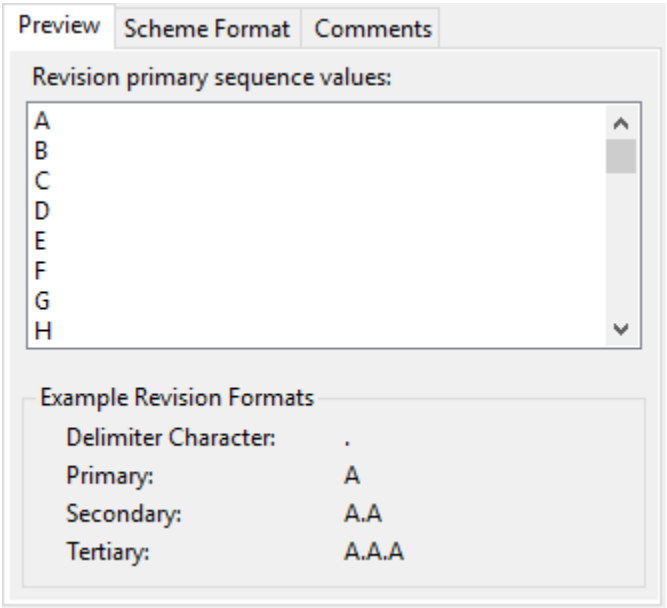
Click the New button in the Revision Schemes Definition dialog. A definition name is required, must be unique and cannot contain special characters. The definition name and description are limited to 255 character (i.e. \ / : * ? " < > |). The category drop-down allows users to select the category for which this definition is available. If the definition is not assigned to a category, it is not available to be used on an object.



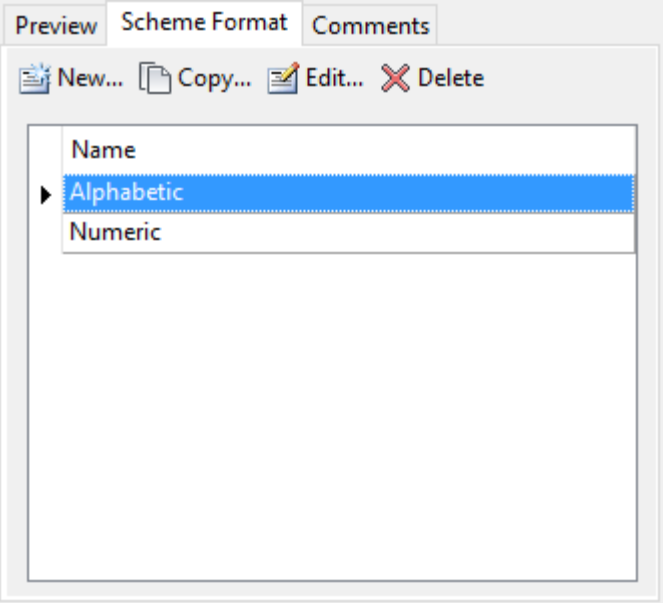
Each part of the scheme gets set in the Scheme Values section of the definition. The delimiter is a character that gets used between each part of the scheme.



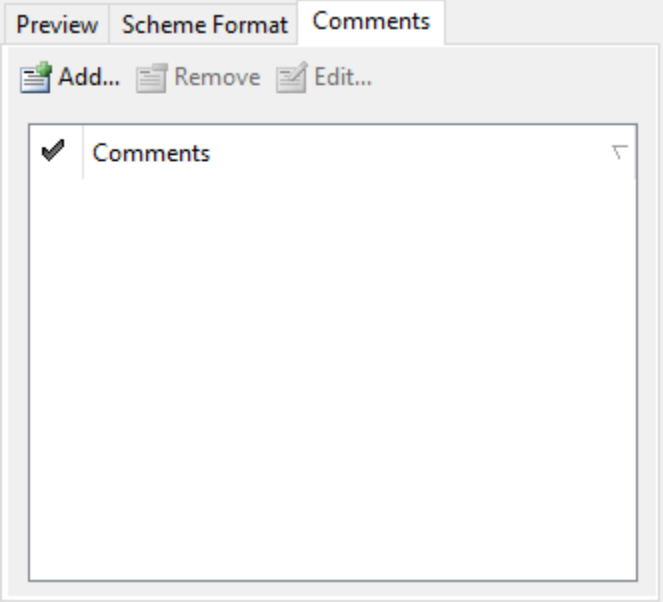
The Preview tab shows an example of the selected scheme.



The Scheme Format tab shows all the schemes that are available to use when creating a new revision scheme. A scheme format gets imported from a text file with each revision character on a separate line in the text file. Once imported, you can use the new scheme in a revision definition.



The Comments tab enables you to add multiple comments to each revision definition. These comments can get selected when the revision scheme gets applied to a file.



Category Administration

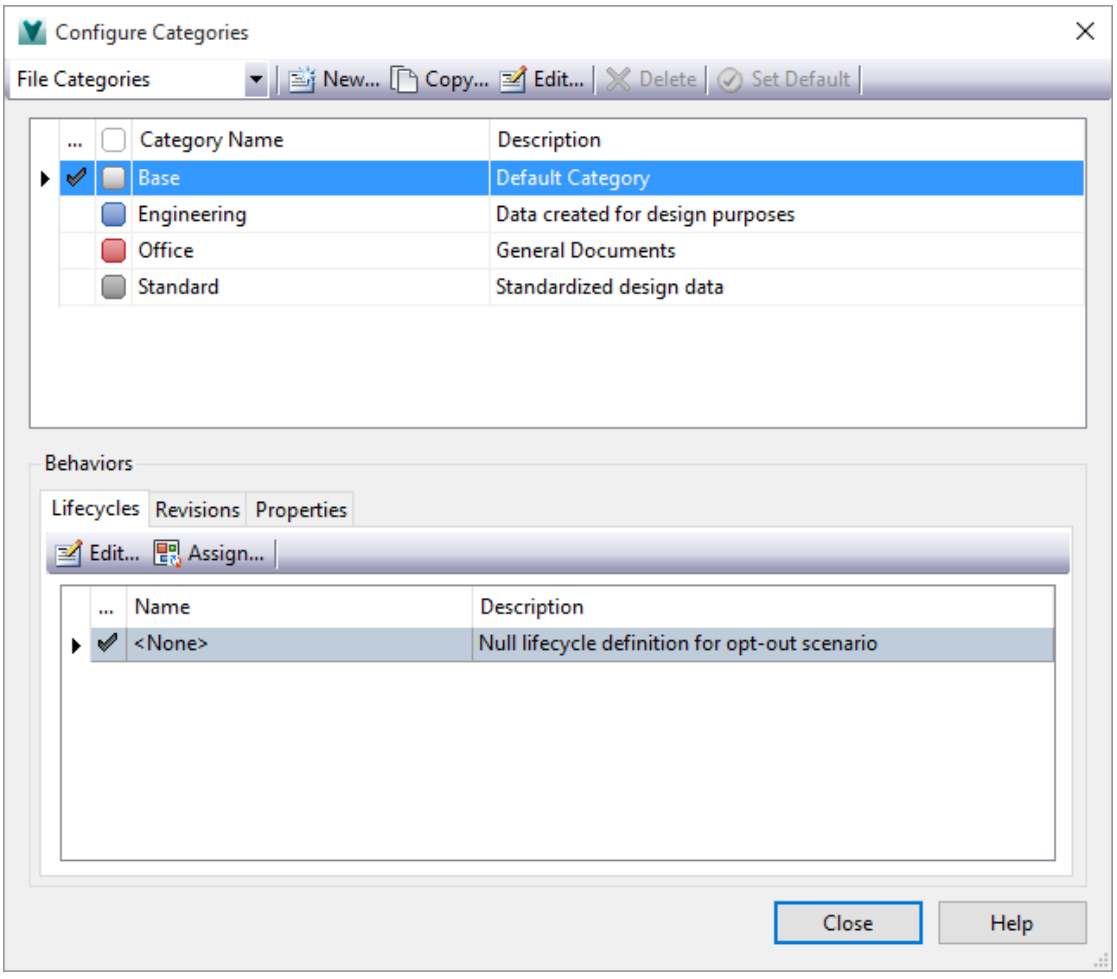
Category Definition

A Category is a container that brings all this together. Categories automatically determine an object's behavior and characteristics based on the category that gets assigned to it. As an example, when file get checked into the Vault, a revision scheme, lifecycle definition, file security, user-defined properties, and property policies can be assigned based on configurable criteria. Using categories provide consistent behaviors for all objects in Vault.

A user with the Document Manger Level 1 or above can assign a category to any file. A file may remain uncategorized in a Base or *<None>* category to take in default behaviors. There are several categories that come with Vault out- of-the-box, including a default or null category called Base Engineering Office Standard. Administrators can create, copy, edit and delete categories in the Configure Categories dialog box.

Note: instead of deleting a category, you can change it to unavailable. Changing the category to unavailable removes it from selection during the Assign process, but not delete from the Vault.

As a general rule, keep Categories separate for different system types. In other words, keep Inventor separate from AutoCAD, separate from MS Office files.



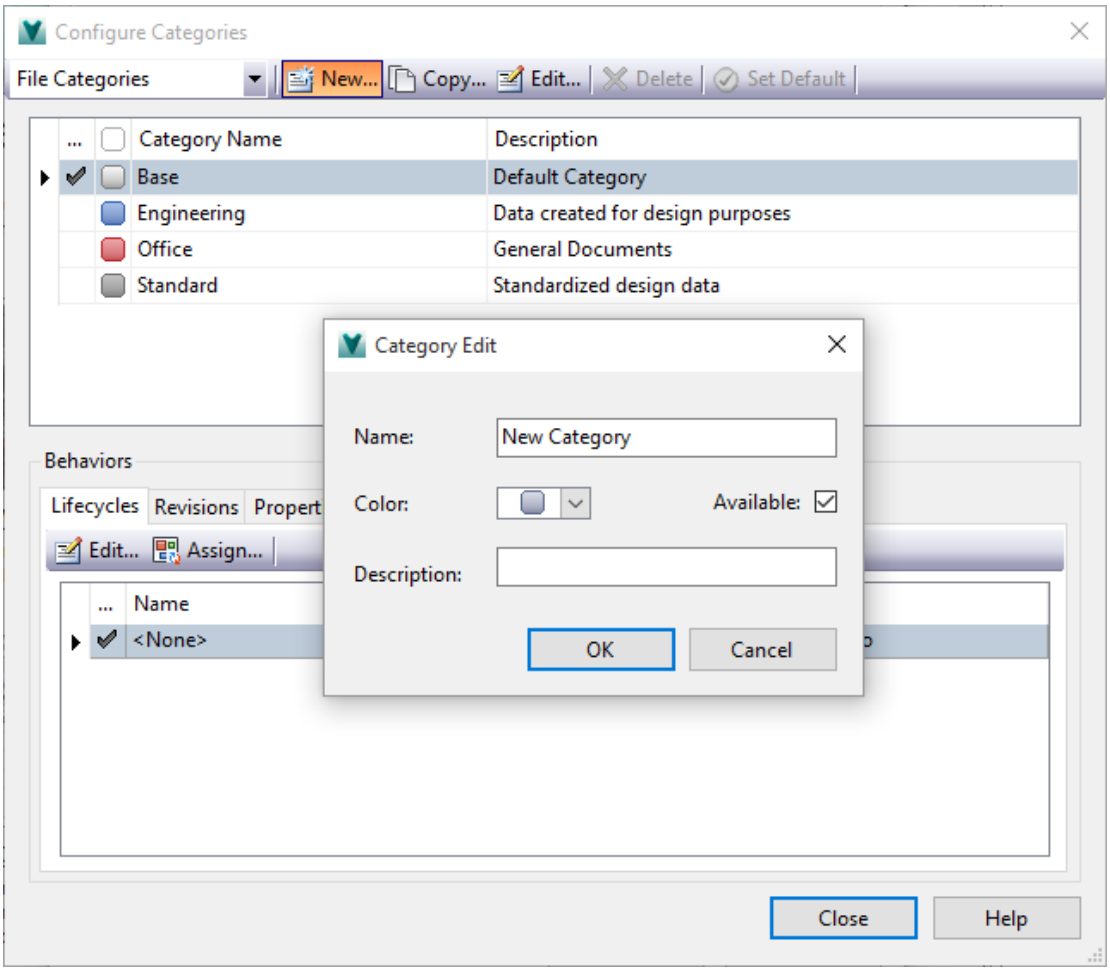
Category Creation

Rules govern which categories get assigned to a file, folder, custom object, or item. These get referred to as assignment rules. Assignment rules are used to automate category assignment to files, folders, custom objects, and items in a vault. Rules determine category assignment that the administrator sets up in the Rule Condition Builder. Use this builder to set conditions based on properties of the object.

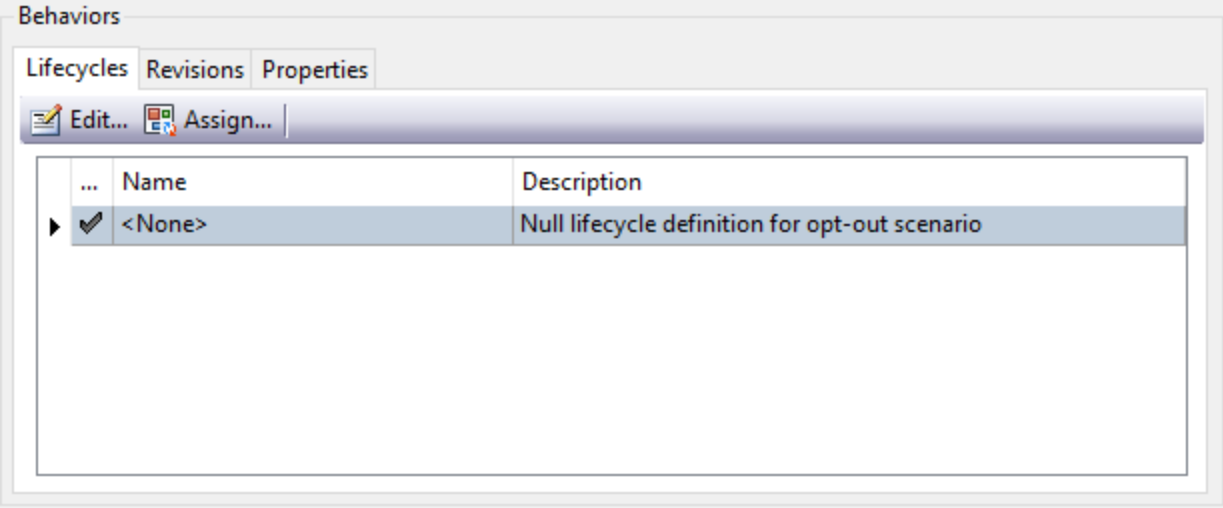
Several categories that come out- of-the-box:

- Engineering
- Office
- Standard

To create a new category, click the New button in the Configure Categories dialog box. Enter a name for the Category, select a color and give it a description. If you are ready for the category to get used, leave the Available checkbox checked.

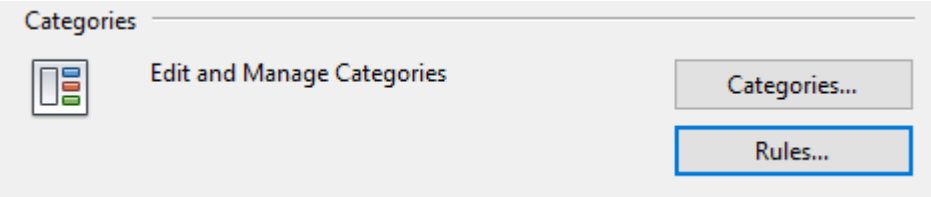


Next you have to assign the Lifecycles, Revisions and Properties associated with the Category. Click the Assign button in each of the corresponding tabs in the Behaviors section of the Configure Categories dialog.

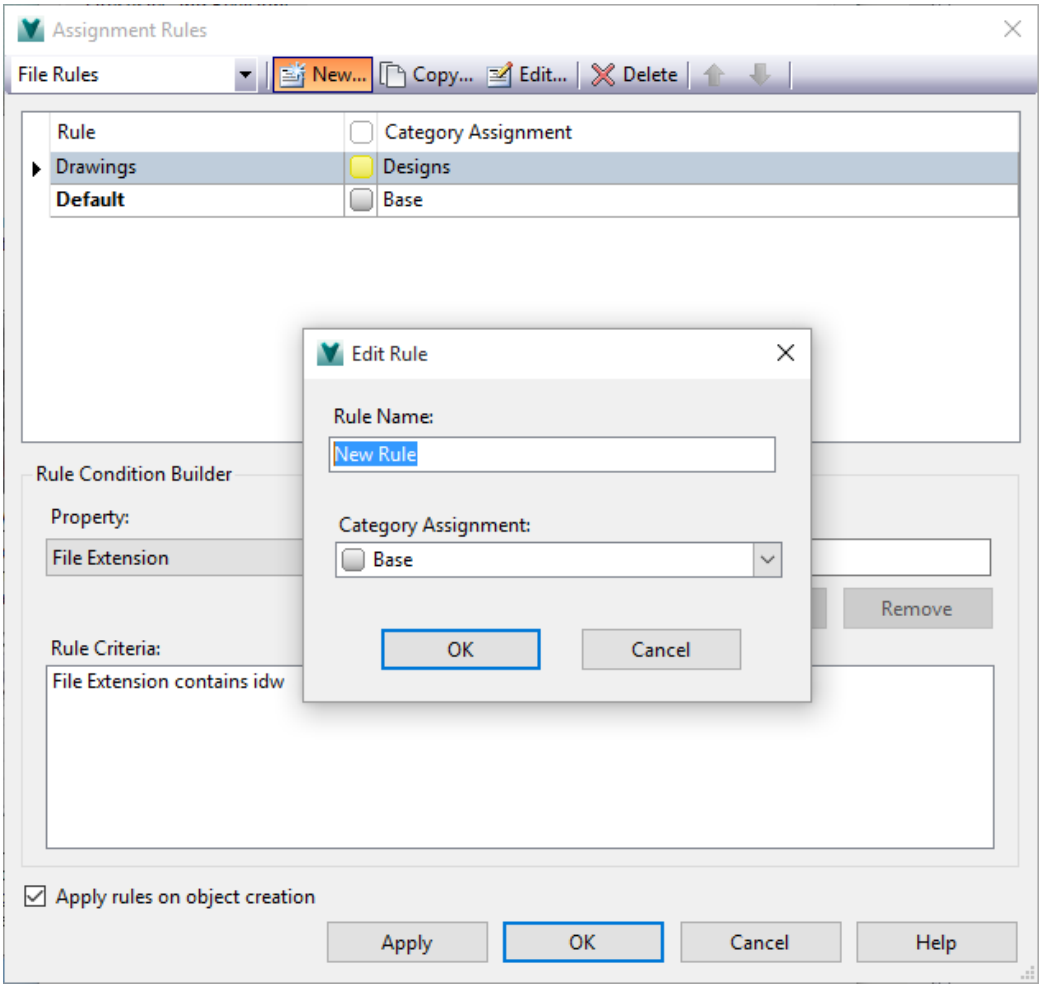


Category Assignment Rules

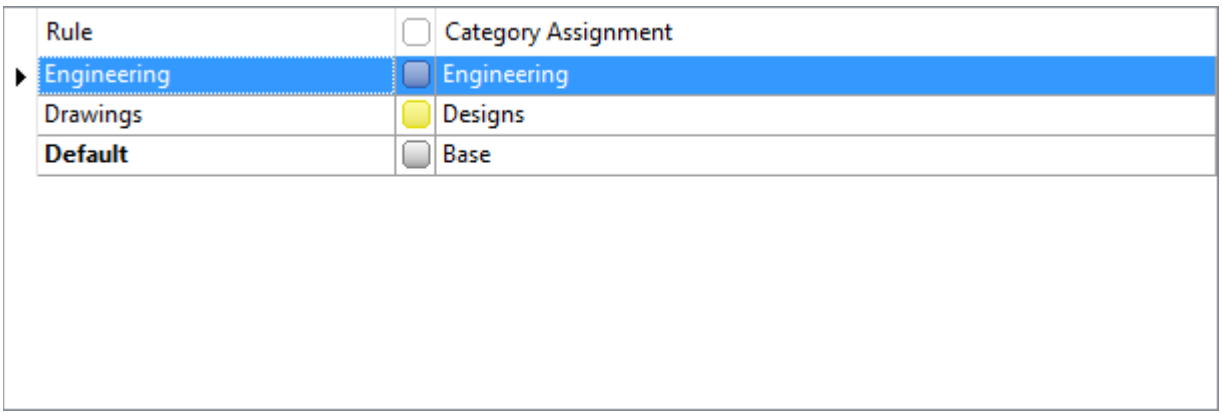
Rules are used to automate assigning files, custom objects, and items to categories in the vault. To create a Rule, select the Rules button in the Behaviors tab of the Vault Settings dialog box.



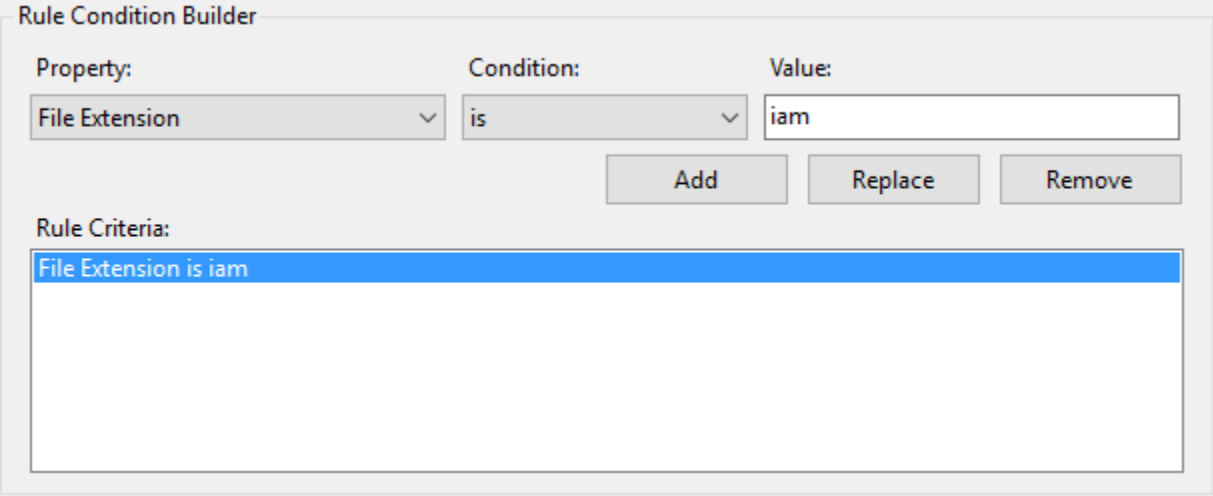
Rules can get defined for Folders, Files, Custom Objects, and Items. In the Assignment Rules dialog, select the object you want to create the Rule for, then click the New button to create a new rule. You can use the Copy, Edit or Delete button to copy, edit or delete existing rules respectively.



Enter the Rule name and select which Category gets assigned when the Rule gets applied. When multiple Rules exists, Rules are tested in the order in which they appear in the dialog. Reorder the rules by selecting the Up or Down arrow buttons on the menu bar.



Rules are automatically applied based on the conditions defined for each rule. Use the Rule Condition Builder section to define the criteria that need to be meet when the rule gets evaluated. If multiple criteria are defined, all criteria must be met then the category gets applied to the object.



Rules can be applied automatically when a file or custom object gets created. For this to occur, the Apply Rules Automatically on Creation checkbox must be selected.