



# BOM Management with Autodesk® PLM 360: Configuration Management

Hagay Dvir – Autodesk  
Martin Gasevski - Autodesk

**PL2570** Learn how to implement configuration management and release processes in Autodesk PLM 360 cloud-based product lifecycle management software. The class covers the concepts of change management and effectivity and dives into the CMII app. We demonstrate the basic workflows and concepts.

## Learning Objectives

At the end of this class, you will be able to:

- Describe how change management and release processes are managed with Autodesk PLM 360
- Learn how to apply CMII practices in Autodesk PLM 360
- Manage effectivity with Autodesk PLM 360
- Customize the configuration management process to your needs

## About the Speaker

*With degrees in Mechanical Engineering, Computer Science, and Business Administration, I've spent the past 17 years working on software solutions for manufacturers in PLM related areas such as Manufacturing Process Planning, System Engineering, and BOM Management. Over the past 2 years I've been on the Autodesk PLM 360 Product Management team with specific focus on engineering business processes in general and BOM functionality in particular.*

*My goal as a product manager is to deliver functionality that solves our customer's business needs. As such, I am always eager to learn more about the business challenges our customers face, and the needs they have from the Autodesk PLM 360 solution.*

*Feel free to contact me at any time at [Hagay.dvir@autodesk.com](mailto:Hagay.dvir@autodesk.com)*

## Introduction

### A little history

Configuration Management (CM) was introduced by the Air Force in 1962 as means to ensure

- Documentation delivered with product matched the product delivered
- Ability of suppliers to create additional identical products

Concepts of CM were widely adopted by industry and are at the core of

- System Engineering (SE)
- Capability Maturity Model Integration (CMMI)
- ISO 9000
- Information Technology Infrastructure Library (ITIL)
- And many others...

Military standard replaced in 1994 by 'best commercial practices' and ANSI GEIA-STD-649, the DoD sponsored alternative

### What is 'Configuration Management'

As defined by my favorite resource, [Wikipedia](#), Configuration Management (CM) is “a **systems engineering process for establishing and maintaining consistency of a product's performance, functional and physical attributes with its requirements, design and operational information throughout its life.**”

Unfortunately, unless you already know what Configuration Management means, the above will not help to explain it. But further down the Wikipedia page the article goes on to explain that “**CM emphasizes the functional relation between parts, subsystems, and systems for effectively controlling system change. It helps to verify that proposed changes are systematically considered to minimize adverse effects. Changes to the system are proposed, evaluated, and implemented using a standardized, systematic approach that ensures consistency, and proposed changes are evaluated in terms of their anticipated impact on the entire system. CM verifies that changes are carried out as prescribed and that documentation of items and systems reflects their true configuration. A complete CM program includes provisions for the storing, tracking, and updating of all system information on a component, subsystem, and system basis.**” [*Systems Engineering Fundamentals*]. Defense Acquisition University Press. January 2001. Retrieved 2012-03-25.]

Even more clearly, the following passage from later in the same article summarizes what CM is all about “**CM is the practice of handling changes systematically so that a system maintains its integrity over time. CM implements the policies, procedures, techniques, and tools that are required to manage, evaluate proposed changes, track the status of changes, and to maintain**

an inventory of system and support documents as the system changes. CM programs and plans provide technical and administrative direction to the development and implementation of the procedures, functions, services, tools, processes, and resources required to successfully develop and support a complex system. During system development, CM allows program management to track requirements throughout the life cycle through acceptance and operations and maintenance. As changes are inevitably made to the requirements and design, they must be approved and documented, creating an accurate record of the system status. Ideally the CM process is applied throughout the system lifecycle.”

## Why is it important?

If there is one thing you can rely on, it is that things change. There are many reasons why things change, and we will not list it here, but it is important to understand that it is virtually impossible to make identical products over long durations of time. This is not to say that you cannot make products that fulfill the same requirements, just that they are not going to be identical. Understanding that, CM is aimed to answer questions like how did this product change over time? What did it look like as any point in the past? What did we build for this customer last time? And if we wanted to build another one of these, what are going to be the differences between it and what we delivered last time?

## So what does it mean?

In simple words, it is about

- Managing the product Bill Of Materials (BOM)
- Managing the product Change Process

PLM 360 provides many built in capabilities and templates to support these tasks. The purpose of this class is to review and introduce these capabilities

## Items, Revisions, and Effectivity

### Items and Revisions

The basic currency in PLM 360 is an Item. When items go through changes, when they are ‘ready’ for consumption by a business process, they get frozen in a Revision. There are many different business processes that consume ‘ready’ items – design review, prototyping, production, and more. When an item is frozen, a Revision of the item is created and shows the state of the item at that point in time. The revisions of an item form a Linked List

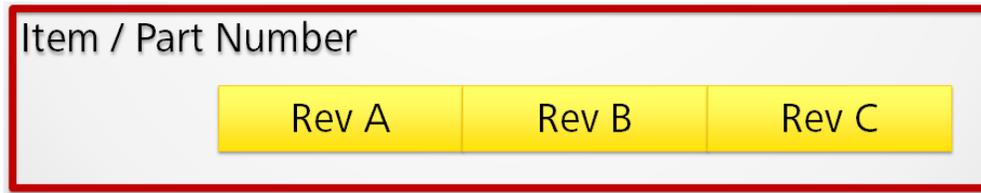


Figure 1 - Item Revisions form a Linked List

## The Lifecycle Map

Lifecycle states in PLM 360 allow our users to manage the ‘readiness’ of an item for a given purpose. The lifecycle states represent maturity of the items or sometimes gateway readiness. What each state means is totally up to the customer (there are no assumed behaviors associated with the lifecycle states other than those defined in the lifecycle map itself), and the number of states and transitions between them are also customer specific. Out of the box the lifecycle map looks something like this

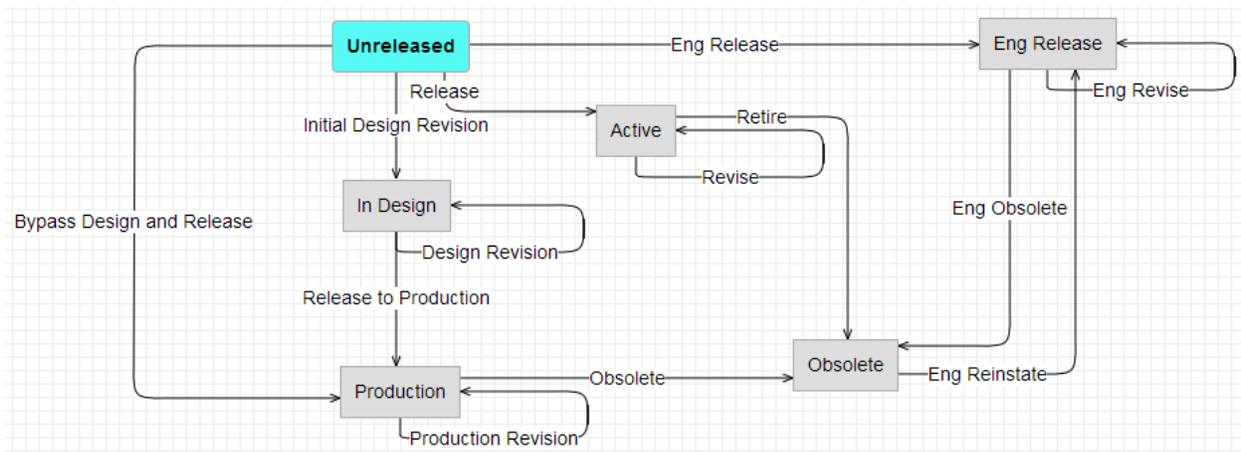


Figure 2 - Example of a Lifecycle map

As you can see in the map, if the customer’s business processes allows for an item to iterate within a lifecycle state, a transition from that lifecycle state to itself is necessary. In the example above, you can see an iteration loop for the ‘In Design’ state and for the ‘Production’ state. That means that you can release newer revisions of the same item into the same lifecycle state.

## The Working Revision

But if the revisions of an item are Locked, then how can we make changes to it? The trick is that in addition to the set of Revisions, every item also has a ‘Working’ revision. The Working revision is always available for editing (except when it is temporarily locked as part of a release process as we will see later in the presentation). When the working revision represents what you want to release, you freeze it into the next revision in the chain

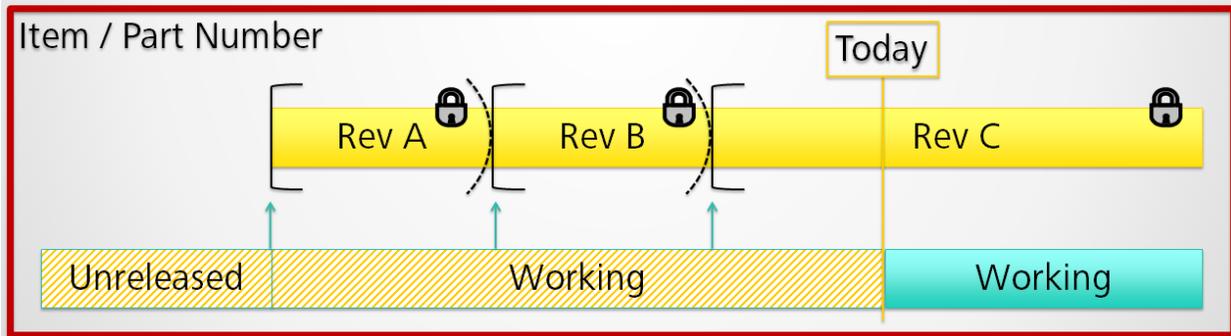


Figure 3 - The Working 'revision'

It is not very meaningful to ask 'what did the working revision looked like last week'? If you really want to know the answer to that, you can always have a look at the Change Log for the item. But you can't just configure the item to show what it looked like before.

## Effectivity

Normally, when freezing a revision, it immediately goes into effect. Unless otherwise specified, the Effective or Effectivity date of an item is identical to the Release date of the item. However, sometimes it is important to make a revision effective only at a point in time in the future. For example, when an assembly will stop using an item supplied by one manufacturer and start using a comparable item supplied by another manufacturer that requires some additional changes in tooling. You want to approve the change immediately so that manufacturing can order the new tooling and procurement can order the new parts, but you don't want the assembly personnel to get confused with the new design until its time. In that case, the user can specify an effectivity date that is later than the release date.

Once a new revision of the item has been released with the new effective date, this is when the prior revision of the item is no longer effective. PLM 360 does not store an 'End Effectivity' date for an item. Instead, it is assumed to always be one day prior to the effective date of the next revision in the linked list of revisions. This ensures that there is never a gap in effectivity for an item.

## Revision Pinning

Different revisions of a part should normally be fully forward and backward compatible (aka the 'No change to Fit, Form, or Function'). So ideally the BOM should always consume the latest revision of any item. But in the real world there are sometimes reasons to consume a prior revision of an item. When that happens, you may pin a prior revision of an item to its' parent. This is done in the BOM on the relationship between the Parent and the Child.

But if that is true, you may wonder what is the purpose of Effectivity? That is a great question! Effectivity on the Item level is really not that important as long as the 'no change to Fit, Form, or

Function' is maintained. However, it is a lot more interesting to set an Effectivity date on an Assembly that consumes different parts, some of which may only be available for us in the future, but that otherwise will not have a Fit, Form, or Function changes between the revisions of the assembly.

## BOMs, BOM Relationships, and BOM Configuration

### Basic BOM functionality

PLM 360 defines a BOM as a hierarchical set of relationships of the type 'Is Made Of' between a Parent item and one or more Child items. The BOM is a property of the Parent Item, and therefore is only one-level deep (parent-child). When a Parent item is 'made of' other Parent items, it creates a multi-level hierarchical BOM.

#	Descriptor	Revision	Lifecycle	Quantity	Units
0.0	600-0001-000 - SBOM Project 621 Head [REV:C]				
▶ 1.1	680-0000-000 - SBOM Mech Upper Bushing [REV:B]	B	F	2.0	
▲ 1.2	683-0000-000 - SBOM PIPING Pneumatic Assembly, Project 621 [REV:B]	B	F	1.0	EA
2.1	165-0000-000 - PIPE Threaded Ends Schedule 40S - 2 [REV:A]	A	F	3.1	IN
2.2	162-0000-000 - FITTING Threaded Tee Malleable Iron Class 150 , [REV:A]	A	Production	1.0	EA
2.3	162-0001-000 - FITTING Coupling Class 125 4" to 1" , [REV:A]	A	Production	1.0	EA
2.4	162-0002-000 - FITTING Reducing Coupling Class 125 4" to 1/2" , [REV:A]	A	on	1.0	EA
2.5	165-0001-000 - PIPE Threaded Ends 1/2" Schedule 40S 1.5" [REV:A]	A	on	1.5	IN
2.6	163-0000-000 - VALVE BALL 1/2" Female-Female Threaded , [REV:A]	A	on	1.0	EA
2.7	165-0002-000 - PIPE Threaded Ends 1/2" - Schedule 40S - 4 [REV:A]	A	Production	4.0	IN

Figure 4 - Multi Level Hierarchical BOM

BOM Relationships in PLM 360 are the property of the Parent. If the relationship between a parent and any of its children must change, it necessitates making changes to the Parent item (but NOT to the child items). There are several build-in properties of the relationship in addition to as many custom properties as you want to add. The built in properties include:

- Line Number
- Quantity
- Revision Pinning
- Sourcing

Line numbers are simply an identifier of the line item and are used for sorting the BOM. Unless otherwise specified by the user, they are sequential numbers.

Quantity is how much of the child item will be included in the parent item. For discrete items, this is simply an integer. But it can may also represent quarts of oil or ounces of glue and therefore it can be any number.

Revision pinning allows for freezing the relationship between a Parent Item Revision and a Child Item Revision. For example, it allows a parent to continue to use revision A of a child item even if the child item went through several revisions since then. Unless a revision pin is used, the BOM always shows the Effective revision on the 'As Of' date selected

## BOM Configuration

Once the BOM is set up, it is interesting to ask 'what did the BOM look like on this or that date' and 'what will this BOM look like when we approve these change orders'. PLM 360 provides two main mechanisms to configure the BOM.

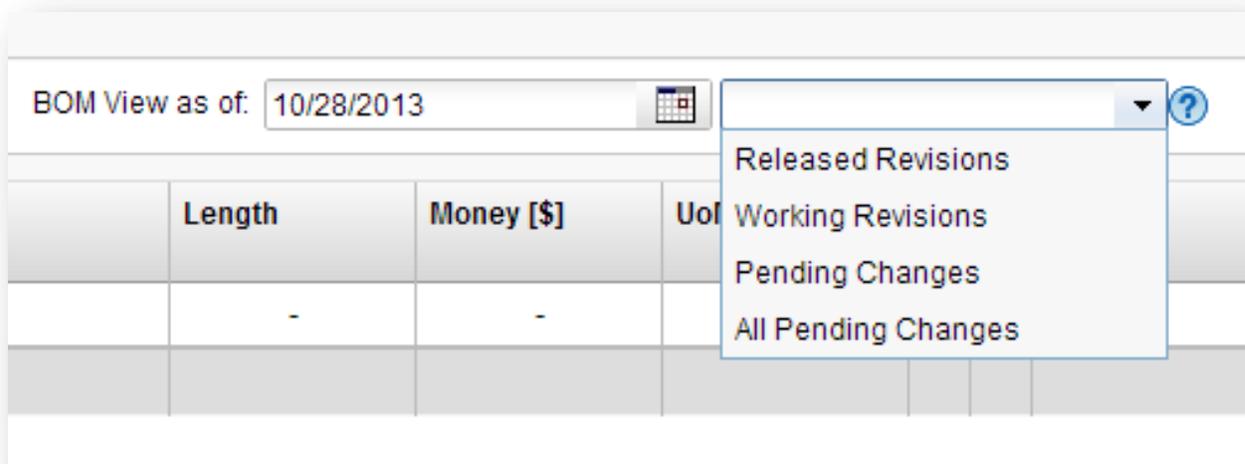


Figure 5 - BOM Configuration Options

The first mechanism is the 'As Of Date'. This configuration recursively selects the effective revision of each item down the BOM tree for the date selected. It is useful in two ways – finding out what the BOM looked like in the past, for example for warranty purposes, and finding out what the BOM will look like in the future, for example for procurement purposes.

The second mechanism is the 'Revision Configuration' seen on the image above as a drop down on the right side of the image. There are currently four built in selection rules.

- Released Revisions – simply chooses the effective released revision recursively. This is useful e.g. when you are a production engineer trying to build a product. This will tell you exactly what the approved configuration of the product looks like

All the other selection rules are only relevant when the 'As Of Date' is today or later.

- Working Revisions – unless an item is Pinned, it chooses the working revision of all items recursively. Useful during the design phase of the product as a planning tool.
- Pending Changes – chooses the working revision of an item IF the working revision is associated to the same Change Order as the current (top) item. Useful in finding out what the BOM would look like if the current change order would be approved.
- All Pending Changes – chooses the working revision of an item IF the working revision is associated with any change order. Useful in finding out what the BOM would look like if all open change orders in the system are approved

## Change Management in PLM 360

### Revision Control

Items in PLM 360 may be subject to revision control. All revision controlled items can only be revised through a Workflow associated with a Revisioning item. This is consistent with CMII requirements. You can think of it as a Change Order (and, in fact, the revisioning item is in the Change Orders workspace in the base tenant).

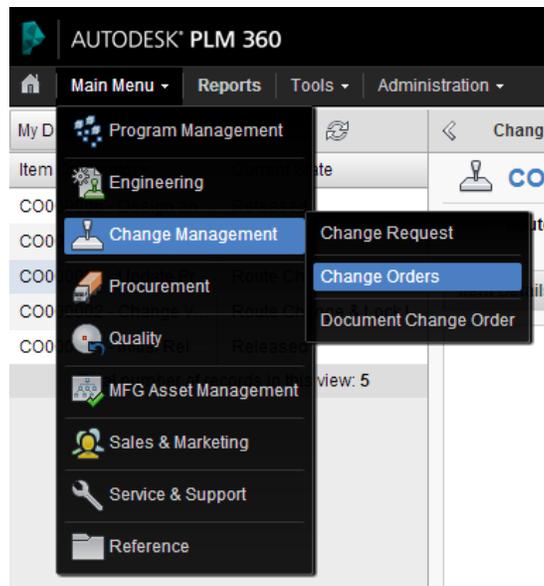


Figure 6 - The Change Orders workspace

The revisioning item (or change order) has a fully customizable workflow associated with it. An example of this workflow is seen in the next image.

There are a couple of noteworthy tags in the workflow that are important to understand. In particular the Lock State and the Managed State. Once the Locked State is reached, the

change order itself, as well as any associated items ('Affected Items') can no longer change. This is useful to ensure that when a Change Control Board (CCB) approves a change, they will all be approving exactly the same state of the item, and will be comfortable knowing that this is what will get released.

Once all approvals are complete and the Managed State is reached, the new revisions are locked.

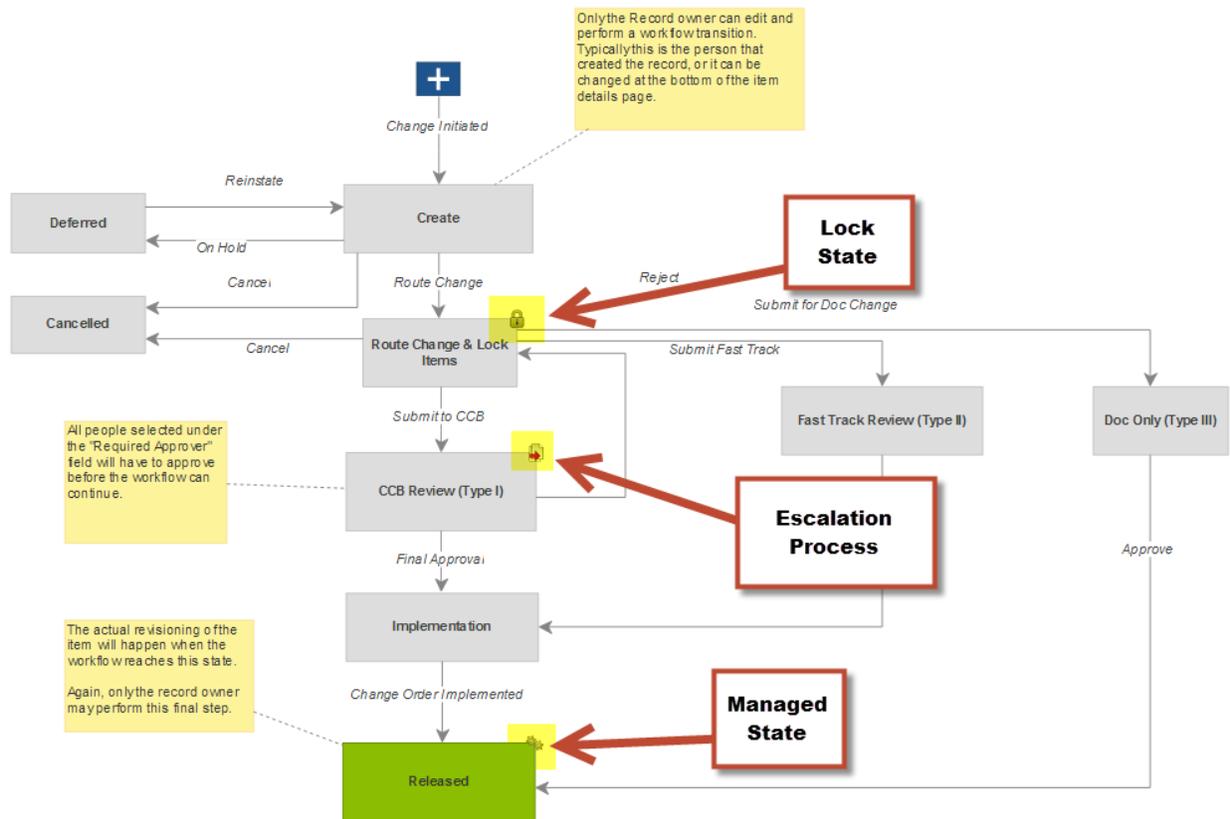


Figure 7 - Workflow associated with the Change Order workspace

## Assess Changes

When you are asked to approve a Change Order, you will likely want to know what will be the effect of the Change Order on your BOM?! You can find this out directly from the Managed Items (sometimes called Affected Items) tab on the change order. When you click on the 'Compare' action, the system opens the item's BOM tab and immediately puts it in the 'Pending Changes' configuration with an 'As-Of Date' as specified by the Effectivity entry in the tab

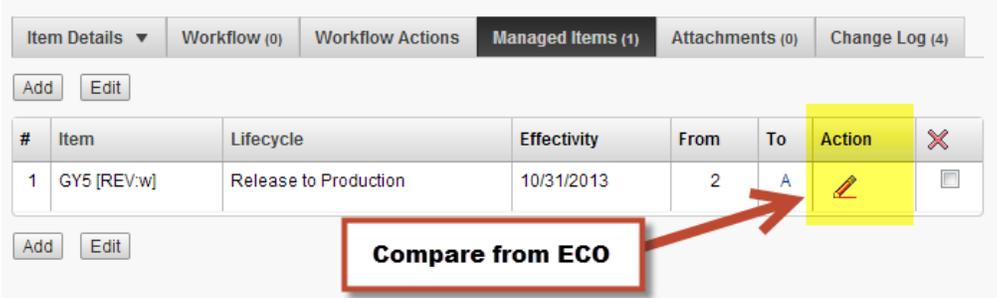


Figure 8 - Compare from ECO