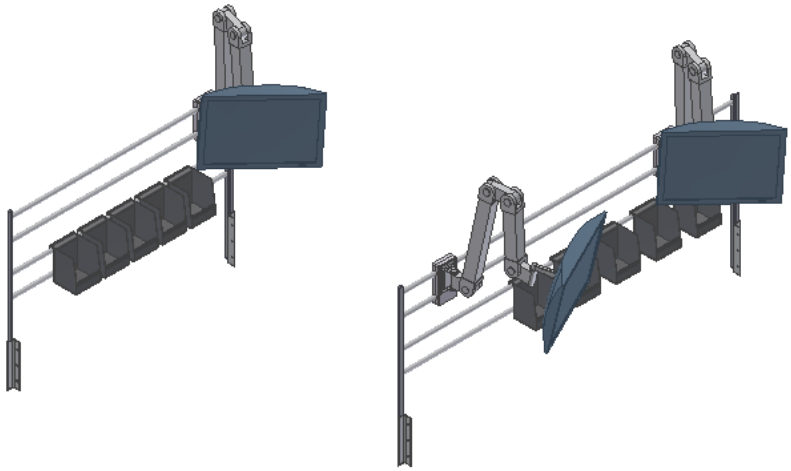
KETIVAMA
AUTODESK MANUFACTURING ACADEMY

Configurable Design

1-C | 1:00 - 2:15 pm
Instructor: Nicole Morris
Create intelligent design configurations with Inventor and AutoCAD using iLogic, iParts/ iAssemblies and Dynamic Blocks. Protect your company's design intent by adding relevant rules and conditions. Decrease design times as you increase efficiency and accuracy by capturing and focusing design know-how in the configurations provided by various Autodesk design tools.

Contents



Understanding Assembly Configuration

In today's market place, it is common for manufacturing companies to look for new ways to innovate. From green design, to lean engineering, to reviewing process to eliminate waste. One such way to eliminate waste is to reuse existing work where possible.

The concept of assembly configurations is not a new idea. For years companies have created a master assembly list. Often the company's sales team would work with customers to determine the options needed for a specific job. The sales team would then send this information back to engineering to create a package of drawings that fit the customers need. This workflow tends to be more common today in the United States today.

We will show you how Autodesk Inventor and even AutoCAD can be used to create Assembly Configurations. This will prevent rework, by creating variations quickly, allowing engineers to spend their time improving quality of the product. This technique also helps with file management. By maintaining a main master design and creating many different variations from the original.

First you will see how you can use Inventor's iAssembly module to create a master assembly and several different variations on your original assembly. We will talk about the capabilities and best practices for using these configurations and documentation for your variations. We will also show how you can use Inventor's iLogic to control the iAssembly interface for a non-Inventor user.

Finally we will show a similar concept using AutoCAD's Dynamic Block capabilities. By creating a master block with many variations nested in that block, you can maintain several different designs within one. Inventor and AutoCAD Exercises are provided on ama.ketivtech.com. ILogic Exercises are available with the iLogic subscription download.

Building Your Master Assembly

To build your master assembly in Inventor or in AutoCAD, it is best to first understand what options you will offer the customer.
Below is an example or orders from 3 companies.

Company	Bench Length	# Compartments	Monitor Left	Monitor Right
A	1500	3	No	Yes
B	3000	5	Yes	Yes
C	3500	6	Yes	Yes

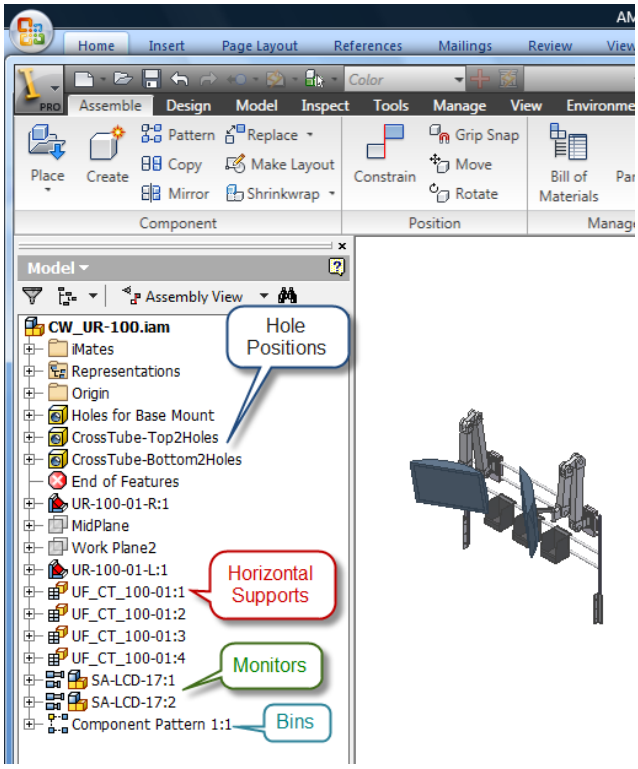
Configurations in Autodesk Inventor

While configurations can be created at a feature level and part level, this course is focused on Assembly configurations. If you have not used iParts or iFeatures, I highly recommend looking into using these tools to help create different variations of a single part or feature.
In this section we will discuss Autodesk Inventor's iAssembly functions and get an understanding of how to approach configurations with your Assemblies.
One basic principal to understand is that the iAssembly function controls parameters that are available at the assembly level. Some of these values include constraint parameters, suppression values, iPart versions and assembly features. In short, information contained in the top level assembly can be controlled by the iAssembly function.
When an assembly becomes an iAssembly, a "factory" is then generated which contains all the different configurations. Once a factory is generated it can be edited and appended. The factory generates members that are kept in an automatically generated subfolder in the folder where the factory resides. Each member of the factory will have a version in this file. In the case of the iassembly each member is a subassembly file. The common parts will need to be included within the project but not in the member subfolder.
While iAssemblies that have a finite number of members can reside in a library, an iassembly where new members will be added as part of the workflow is recommended not to be kept in a read-only library. Since both the factory and the members will need to be modified. If you are using iParts or iAssemblies with a vault environment, it is recommended that the files not reside in a library folder.

Parameters available at the assembly level include:

Tab	Configurable Items	Tab	Configurable Items
Components	• Include/Exclude • Grounding Status • Adaptive Status • Table Replace	Exclusion	• Components • Constraints • Assembly Features • Work Features • iMates • Representations • Component Pattern
Parameters	• Constraints • Assembly Features • Work Features • iMates • Component Pattern	iMates	• Offset Value • Include/Exclude • Matching Name • Sequence Number
Properties	• Summary • Project • Physical – Weld Bead Material • Custom	BOM	• BOM Structure • BOM Qty
		Other	• User-Defined field

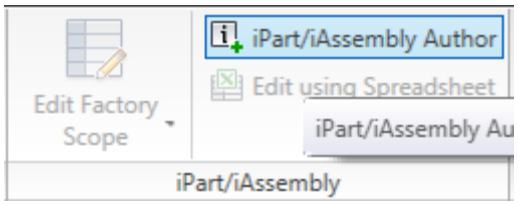
It is not difficult to create an iAssembly, but it helps to plan your assembly based on how your configuration will change.
Let us take a look at our example assembly.
The horizontal supports of the assembly will change lengths according to how long the assembly gets. So we will use an iPart family with standard sizes, since iPart members are interchangeable without losing constraints.
The two monitors are flexible subassembly occurrences that may or may not be included. We will use Include/Exclude option for these.
The number and offset of container bins will vary so we will use a component pattern whose number and offset value are parameters in the assembly.

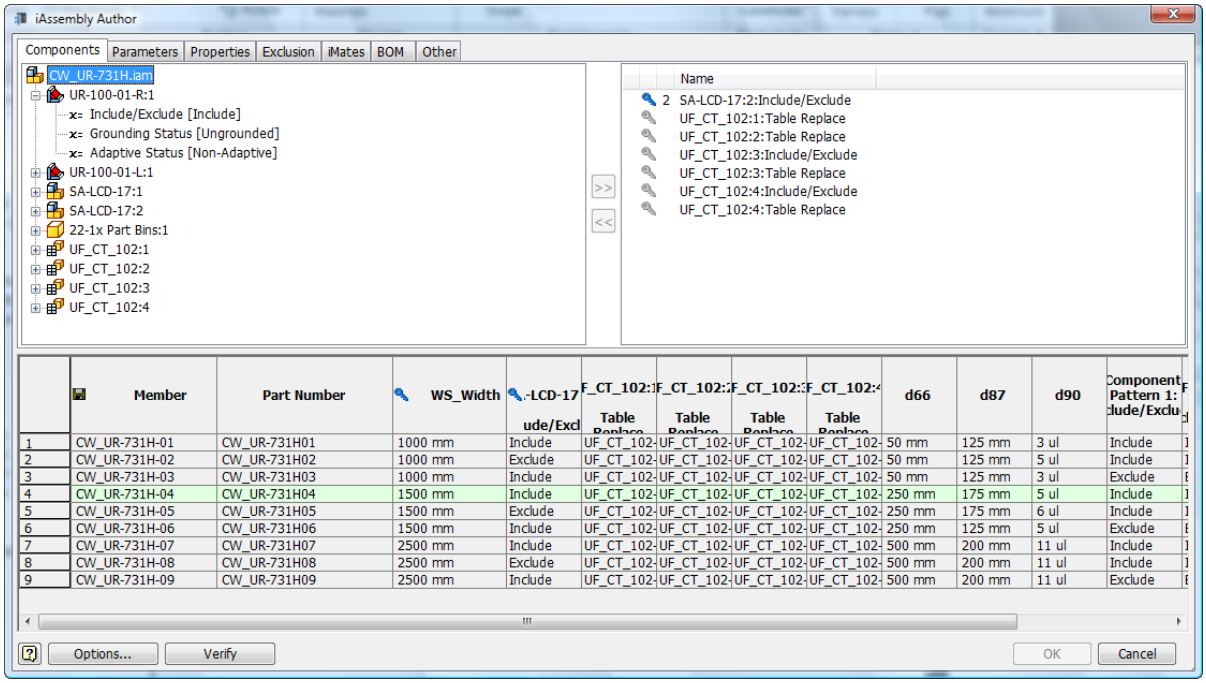


Tip: Before creating the iAssembly it makes sense to rename parameters so that when you see them on the list you will be able to understand what value they represent. Also, they will automatically be added to the list of columns.

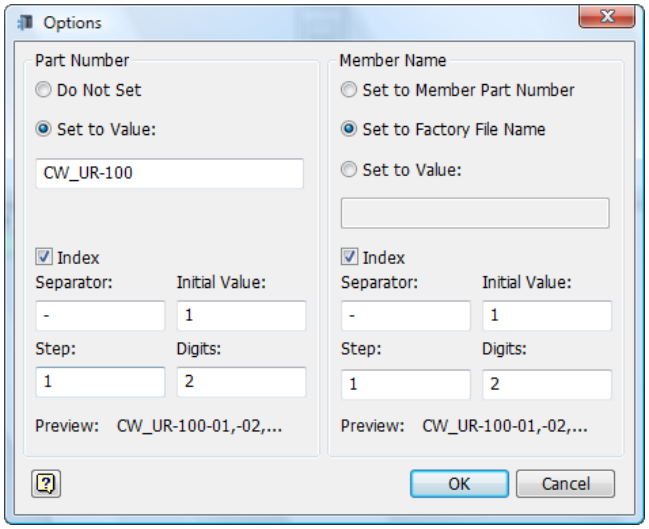
General Procedure for Creating an iAssembly:

1. Create Master Assembly
2. Go To Assembly Tab>iPart/iAssembly Panel and select iPart/iAssembly Author
3. Add attributes for each change your assembly will undergo
4. Add rows for each variation
5. Change the values of each row accordingly
6. Verify Name of members
7. Add Keys if necessary
8. Select Ok.





Naming Options



The left two columns are for Member Name, which is the filename, and Part Number, which will show in the Bill of Materials. The names do not have to match except that it makes it easier to find later. Select Options to control these names in a sequence or specific value.

Keys

Keys can help you filter a large list of configurations. For example your assembly could have 9 different configurations but 3 of them are 1000mm long, 3 of them are 1500mm long and 3 are 2500mm long. You would set the WS_Width as a Key column. Now the configurations will be grouped by their length.

Configuration Table Cell Colors

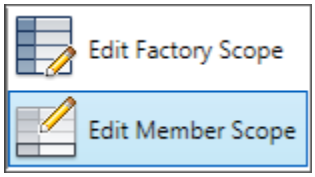
- Light Green – Active Row
- Dark Blue – Custom Parameter
- Mango – Excel formula is driving the value
- Yellow – Syntax error with the cell's value

Change current Member

Once you have created your assembly you will see a Table icon in your browser. You can double click on the different members of your iAssembly to see the different configurations.

Add Scope of Work

You can edit the iAssembly members or all of the members at once by selecting the proper scope. Select Edit Factory Scope (Default) so that changes make to the assembly affect all members.
Select Edit Member Scope to modify only the active member. You can now make changes to the assembly that will automatically be captured in the active member.

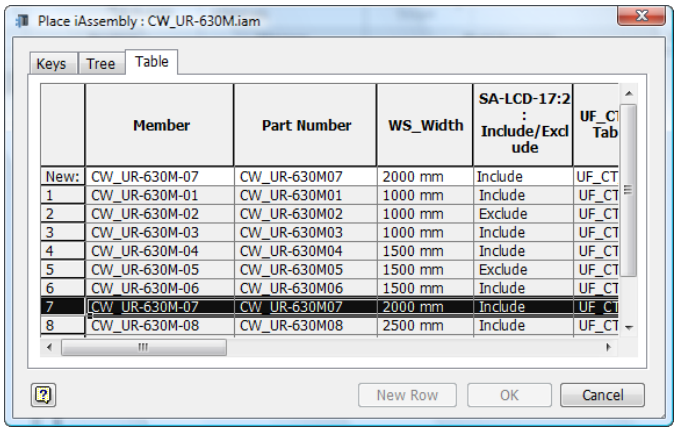


Editing in spreadsheet

You right click the table in the browser to enable editing just as you can in an ipart. For more information on options for editing in spreadsheet see help.

See Exercise – 1-C Configurable Design Exercise 1.pdf

iAssembly Usage

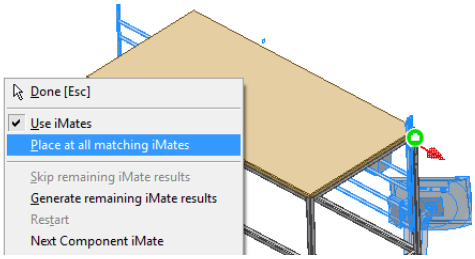


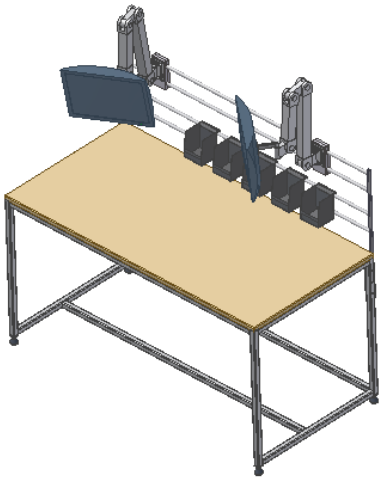
When you insert an iAssembly into another assembly you will have the option to select which member to include. Use Keys, Tree or Table View.
Each time a member is used a member is created in the project in a subfolder named for the Factory.
Each member will have the Member name given in the original iAssembly Author dialog box.
In the Table View, at the top of the list is a row entitled New. You can change the values in this row and select New Row on the bottom of the dialog. This will create a new member in the factory.

Too automatically place the assembly. You can also use iMates that were placed in the original factory. Imates of the same name must exist in both the original factory and in the destination assembly.

While in the insert dialog, right click and check Use iMates, right click again and select Place at all Matching iMates.

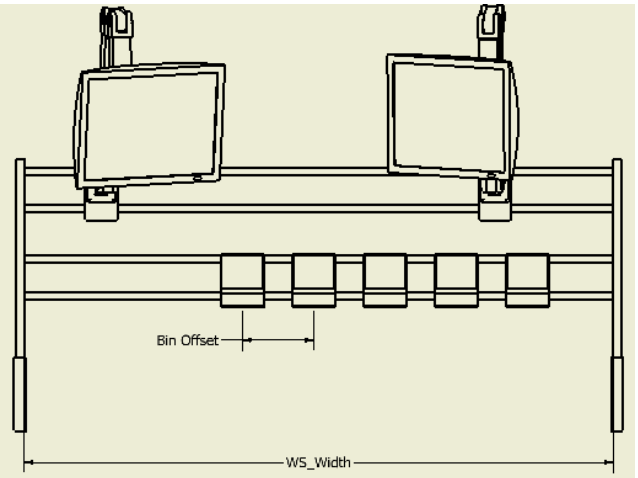
See Exercise – 1-C Configurable Design Exercise 2.pdf





iAssembly Documentation

You can create a specification sheet of an iAssembly factory. In an .idw, create a view you the main factory. The view creation dialog, Model State Tab, will allow you to select which member you want to show in the view.



TABLE

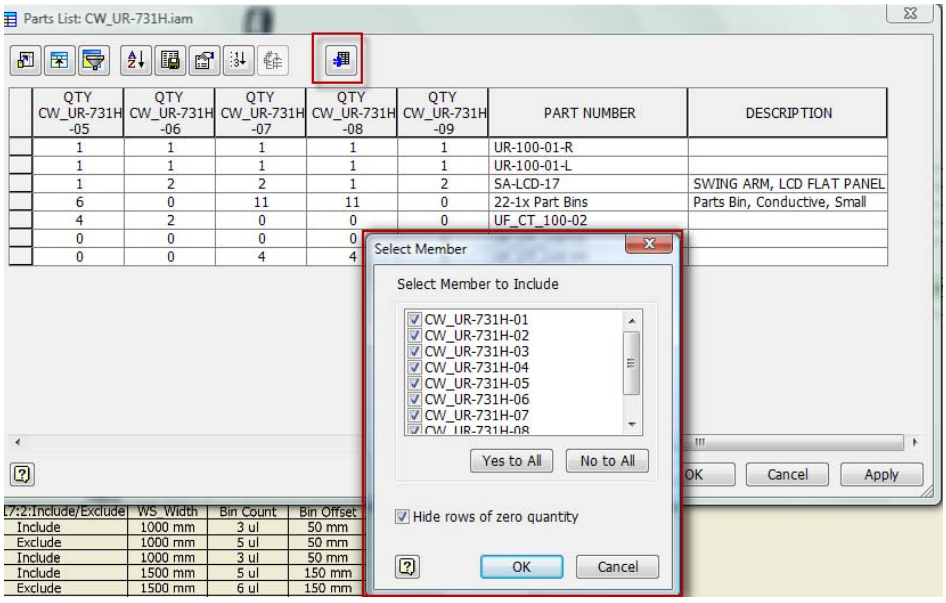
Member	SA-LCD-17:2:Include/Exclude	WS_Width	Bin Count	Bin Offset
CW_UR-731H-01	Include	1000 mm	3 ul	50 mm
CW_UR-731H-02	Exclude	1000 mm	5 ul	50 mm
CW_UR-731H-03	Include	1000 mm	3 ul	50 mm
CW_UR-731H-04	Include	1500 mm	5 ul	150 mm
CW_UR-731H-05	Exclude	1500 mm	6 ul	150 mm
CW_UR-731H-06	Include	1500 mm	5 ul	150 mm
CW_UR-731H-07	Include	2500 mm	11 ul	500 mm
CW_UR-731H-08	Exclude	2500 mm	11 ul	150 mm
CW_UR-731H-09	Include	2500 mm	11 ul	150 mm

Create a Table for iAssembly factory parameters.

Create a Table for this view and select the Column Chooser icon. Here you can select which parameters to include in a table. In dimensions, you can suppress the actual value and type in the Parameter name.

Create Parts List For iAssembly factory members

When you create a parts list for an iAssembly factory you can show quantities for each member of the factory. After placing the parts list, edit it. Select the Select Members button at the top. Select which members to include in the parts list.



See Exercise – 1-C Configurable Design Exercise 3.pdf

iLogic Rules in iAssembly

iLogic is an addin available for subscription customers only. You can download iLogic from your subscription site. It comes with installation files for both 32 bit and 64 bit versions as well as samples. The iLogic download comes with 3 easy to understand tutorials that will help you understand how to set up simple rules in Inventor. This is a great stepping stone if you are interested in learning VBA.net in the future. If you already have programming experience in VB or VBA, you will find this tool very easy to use.

What is ILogic?

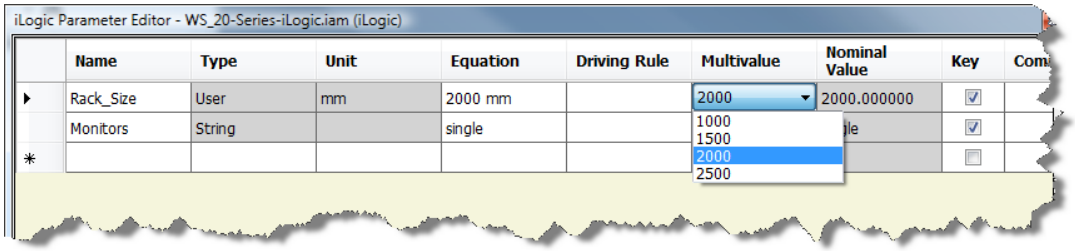
"*iLogic extends the computational capabilities within Inventor to include rules. These rules work along with the parameter update mechanism of Inventor, and allow you to include much more sophisticated design intent into your models. Traditional parametric modeling involves driving geometry with dimensional parameters. These parameter values can be input directly by the user, or can result from fixed equations involving other parameters or even linked spreadsheet values. Using rules in a parametric model allows for conditionally-defined equations. These "conditional equations" are not limited only to the parameters, but can involve all aspects of the design. Equations or relationships can be defined between the parameters, properties, attributes, features, components, or any other aspect of the design. Defining the relationships between all objects in a design makes it possible to update the model completely, correctly (according to the rules), and automatically when input parameter values are changed. A rules-enriched model is therefore far superior to a simple parametric model." ~ excerpt from Tutorial 1.*
After learning iLogic you may find many applications where you can streamline your process and build in more intelligence.

ILogic with iAssembly

For this example, we have used ilogic in our iAssembly to facilitate combinations of functions such as changes nested parameters, changes in iFactory members.
In the datasets, is a sample called *WS_20-Series-iLogic.iam*. If you have the iLogic add-on installed, you will be able to go to the Parameter flyout and select iLogic Parameters. Change the size of the rack and note that the nested parameter of the table updates concurrently with the rack, which would normally take 2 steps. You could even create a dialog interface within Inventor to further simply this tool.

```
'set parameter of table to match rack_size parameter
if Rack_Size=1000 then
Parameter("CW-BF-400:1", "WS_Width")=1000
Elseif Rack_Size=1500 then
Parameter("CW-BF-400:1", "WS_Width")=1500
Elseif Rack_Size=2000 then
Parameter("CW-BF-400:1", "WS_Width")=2000
Elseif Rack_Size=2500 then
Parameter("CW-BF-400:1", "WS_Width")=2500
End If

'select one of the possible iassembly members for the length
If Parameter("CW-BF-400:1", "WS_Width")=1000 then
i = iPart.FindRow("Rack_iLogic", "WS_Width", "=", 1000)
Elself Parameter("CW-BF-400:1", "WS_Width")=1500 then
i = iPart.FindRow("Rack_iLogic", "WS_Width", "=", 1500)
Elself Parameter("CW-BF-400:1", "WS_Width")=2000 then
i = iPart.FindRow("Rack_iLogic", "WS_Width", "=", 2000)
Elself Parameter("CW-BF-400:1", "WS_Width")=2500 then
i = iPart.FindRow("Rack_iLogic", "WS_Width", "=", 2500)
End If
```



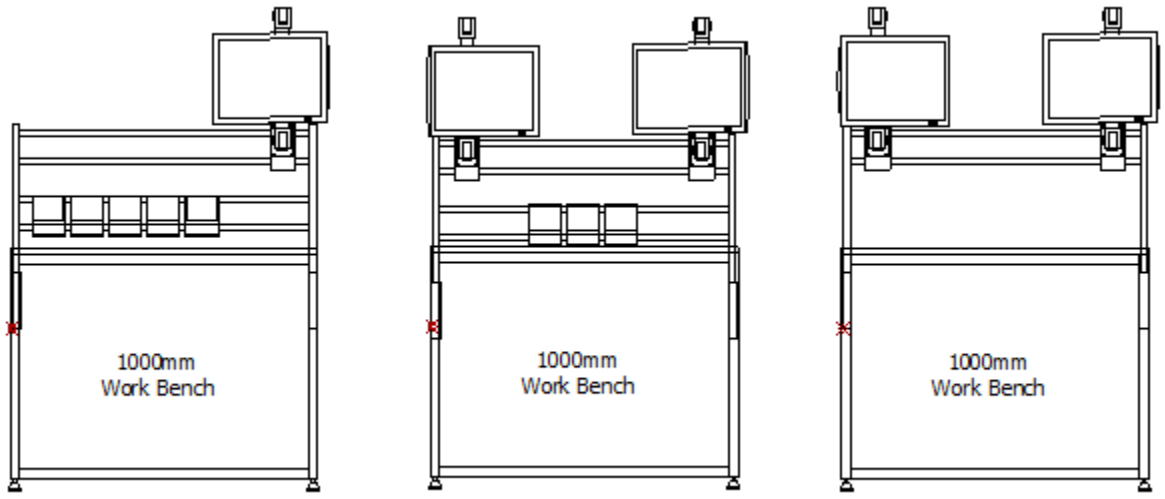
Vault Considerations

When using a Vault environment, it is recommended that you do not keep your factory (iassembly or ipart) or their members in a library folder. When the ifactory is not in the library folder, you can check out the factory and if you make changes to the members it will prompt you to check the members out as needed. However, if your factory or members are in the library folders it adds a level of complication. You will need to have editing rights to the factory and the members.

Configurations in AutoCAD Using Dynamic Blocks

You can use Dynamic blocks in AutoCAD to represent different configurations of an assembly or part.

- Create different views for each different option.
- Create Individual blocks from each of these views.
- Create a single block.
- Use Block Editor to Control Visibility States.



See Exercise – 1-C Configurable Design Exercise 4.pdf