Subject: IronCAD Hyper-Interoperability Explained from the Basics

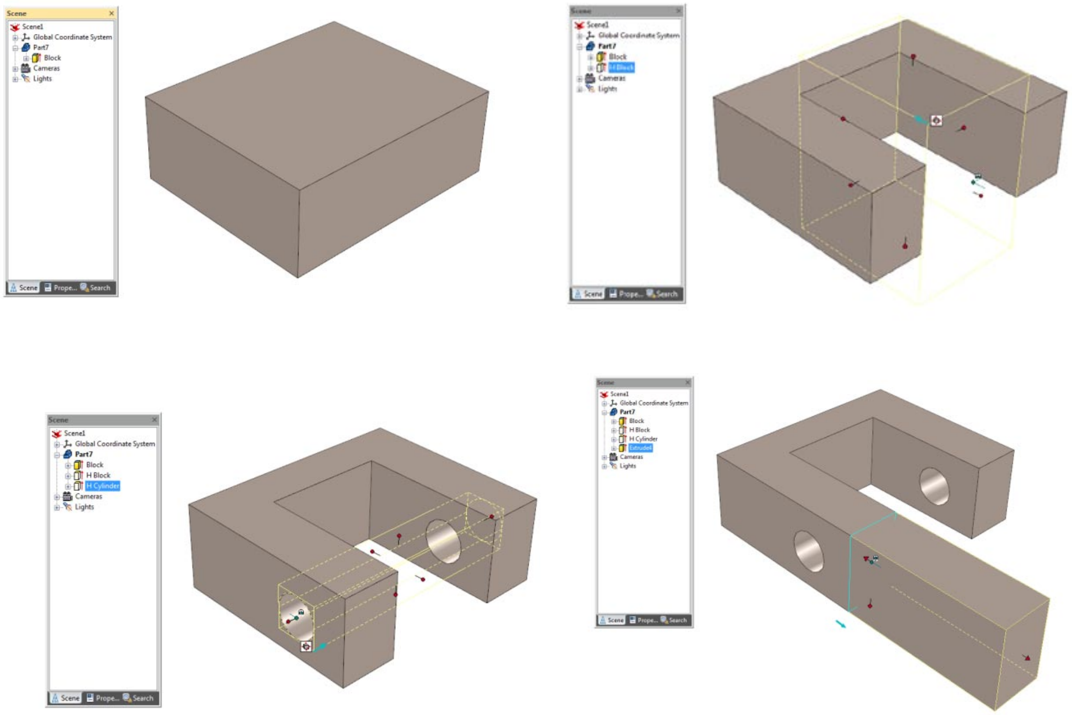
Hi IronCAD Community Members,

Recently in the MCAD industry news articles, you may have been noticing a recent surge in articles about history-based and direct modeling capabilities. Since IronCAD's inception in 1998, IronCAD has had the ability to support History Based Features and Direct Modeling in a single part structure that has intelligent logic. However, since we were years ahead of the market (you could say around 8-10 years), it has been difficult to convey the concept in a very logical and clear illustrating way. This article is intended to give a general overview of IronCAD's design technology that will hopefully help you understand some of the benefits experienced in IronCAD that can hinder traditional modeling systems and can help you understand where IronCAD is positioned in the industry today among the new coming technologies.

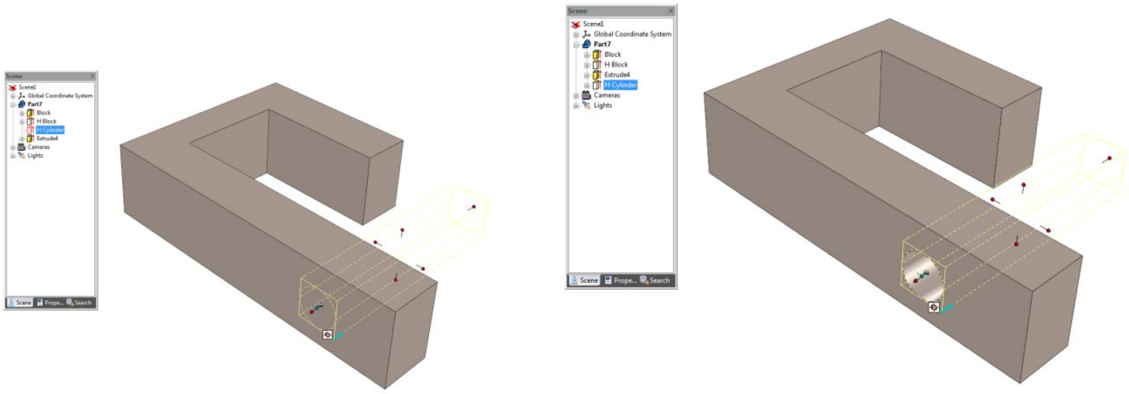
Let's start with a little bit of the history behind IronCAD. Back in the late 90's, the focus on 3D Modeling was primarily parametric history-based applications with a smaller subset on direct modeling systems. Many of the users of these technologies raised common road-blocks found in both paradigms. Among these road-blocks in history-based systems were limitations with dynamic changes in the conceptual stages, ability to work with imported data for design and for other application specific processes (such as FEA), getting unexpected results due to history-dependency or other relations, and difficulties making unpredictable changes late in the production design change in parametric systems. While direct modeling users discovered issues with ability to build and re-use feature information as well as building certain geometrical rules that can be easily updated and modified (refer to my P.S. comment at the end to explain one subset of this). From this feedback, IronCAD was born. It was the first modeling system on the market that supported History-based feature modeling and Direct Modeling mixed in a single part structure.

Although it was revolutionary at the time, it was easily misunderstood by press in the industry as a direct modeling system. We believe the reason for this was that IronCAD made the technology very seamless to the user resulting in IronCAD users not really knowing that modifications being performed were a technical hurdle in the other modeling systems individually. To better illustrate this, let me give a very simple example.

In this example, let me show a simple modeling example that a user may think is logical but is a very real road-block in parametric history-based systems (especially in conceptual design and late stage modifications). First I'll start with a simple extrusion called a "Block". Next, I will make a cut in the side of the block using an "H-Block" removal extrude. I'll follow this with adding a simple hole on the other side of the block that cuts into the "H-Block" removal area through the part. Last, I will extend one of the legs that was generated from the "H-Block" with another extrude feature. At this point, we are working along without any hiccups. Now, we want to move the hole we created out onto the leg segment we just created in the history order. "Where did it go!!" or "What! I can't do this?" would be the phrase from users on the parametric history systems. As all parametric system users know, this is due to the history order. The leg segment does not exist at the time the Hole was added. So moving the hole is like moving it into space. Now this was where IronCAD added intelligence into the modeling system. When you attempt this in IronCAD, it logically understands there is geometry there and it automatically manages the structure to achieve the logical result (as a History Reorder – no need for direct modeling for this). IronCAD calls this history logic "Dynamic History" or "Non-Rigid History System". Simple example, but this can be very frustrating to history-based users in the design process when attempting to make changes when the feature trees start expanding and especially when multiple users are working on a single part design.

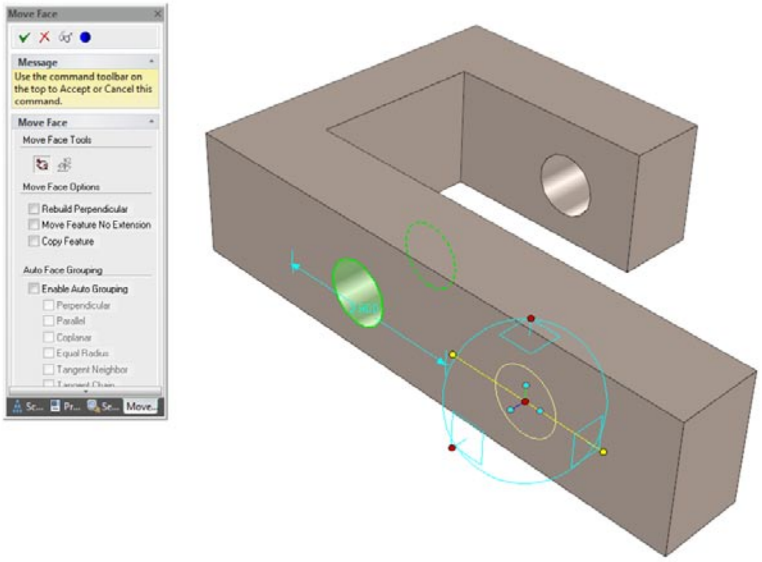


Build up of the model in a history based order



Result of hole move (Left: History Dependent. Right – IronCAD Dynamic History)

Now let’s take this simple example a step further. If you noticed as we created the example above, the original hole resulted in affecting two leg segments (thru hole). In our final step where we wanted to move the hole, what if we wanted to move it such that the one leg segment was not affected or left intact on the original leg. This is where direct “face” editing comes into the picture (IronCAD refers to is as Direct “Face” Modeling (DFM). You simply select faces you want to modify instead of features. In this case, the user would select to the face of the hole on one segment and move it using the “Move DFM” command. At this point, IronCAD would automatically determine what features need to be transformed to allow for this type of modification.


Move Face

Message

Use the command toolbar on the top to Accept or Cancel this command.

Move Face

Move Face Tools

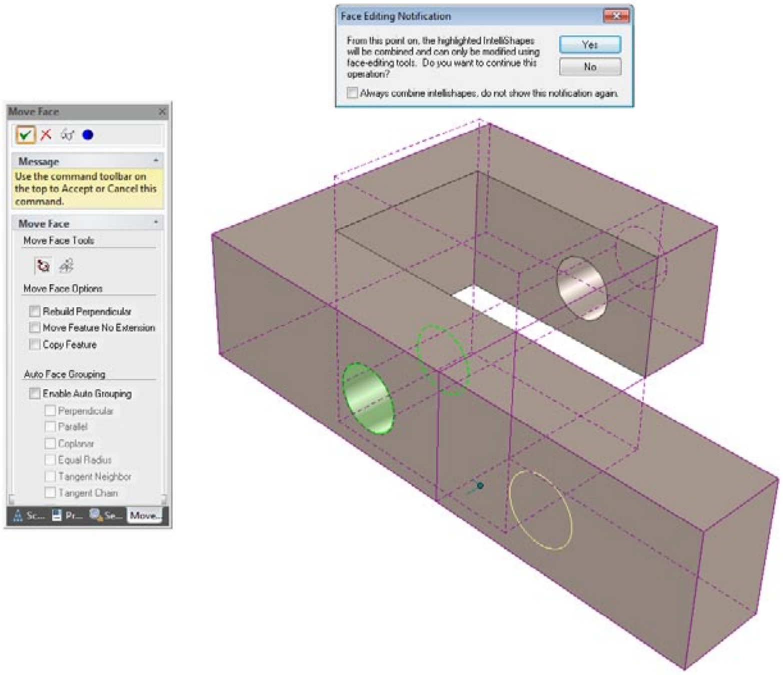
Move Face Options

□ Rebuild Perpendicular
□ Move Feature No Extension
□ Copy Feature

Auto Face Grouping

□ Enable Auto Grouping
□ Perpendicular
□ Parallel
□ Coplanar
□ Equal Radius
□ Tangent Neighbor
□ Tangent Chain

Selecting and Moving One Segment of the Hole



Face Editing Notification

From this point on, the highlighted IntelliShapes will be combined and can only be modified using face-editing tools. Do you want to continue this operation?

□ Always combine intellishapes, do not show this notification again.

Message

Use the command toolbar on the top to Accept or Cancel this command.

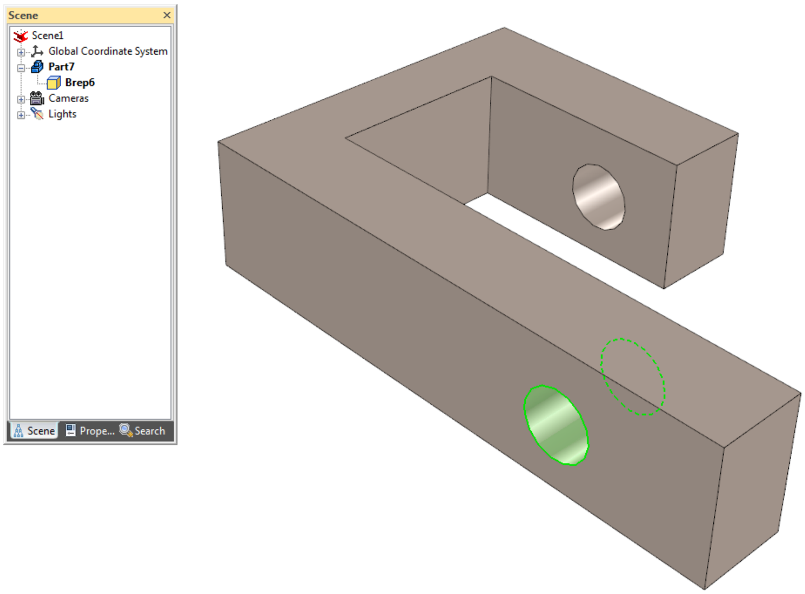
Move Face Options

□ Rebuild Perpendicular
□ Move Feature No Extension
□ Copy Feature

Auto Face Grouping

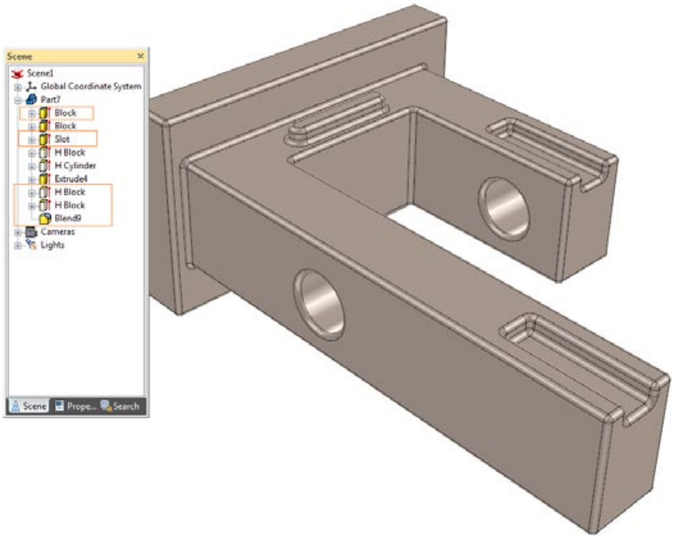
□ Enable Auto Grouping
□ Perpendicular
□ Parallel
□ Coplanar
□ Equal Radius
□ Tangent Neighbor
□ Tangent Chain

Notification of Combine Process to Achieve Direct Editing



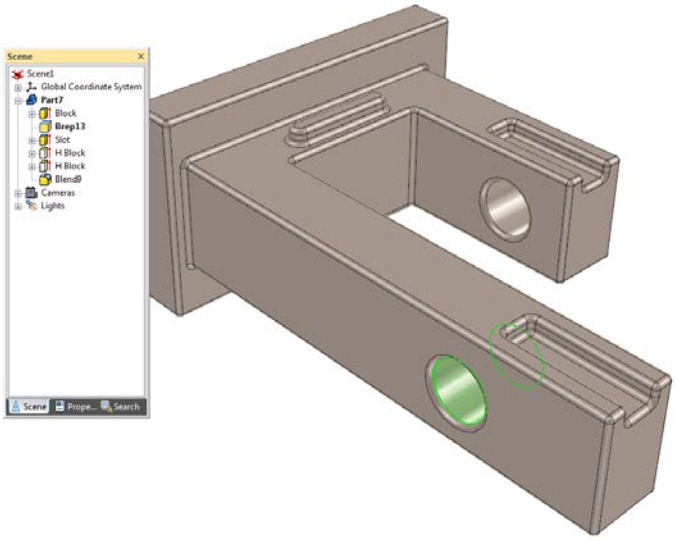
Result of Move Direct Face Modification

As you can see from the result, the structure automatically combines the features needed to support such a modification. There is no need for the user to determine what features are needed to support this. In this example, it is not clear exactly what is happening at this stage since it is such a small example. So first let’s add some geometry to the model before the Hole “Direct Face Modeling” command to better illustrate what is happening.



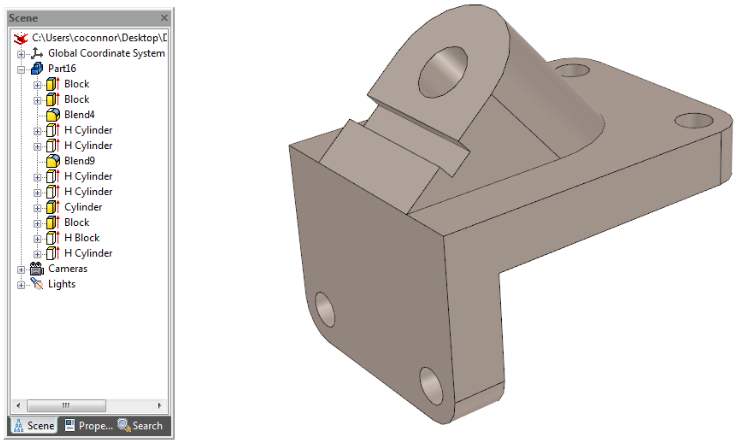
Added Base Block as First Feature, Slot After our original Block and before H-Block, and Two Cuts and Blends Near End of Structure

As you can see by the red highlighted items, I added a feature early in the design and a couple of features late in the design. Now, let’s make the same Hole Direct Face Modification at this point. Notice that the original base block we added is not modified, the same features needed to modify originally are still used and placed in the history order after the base block, and the feature not affected by the modification are left intact as original history based features. This is ability to intelligently determine the features affected by a direct modification and inserting a direct modification feature into a single history base part (mixed) is the patented technology that IronCAD originally developed in 1998 (Refer to the P.S. example at the end on why the order matters in a mixed history-based and direct editing model). Makes you wonder what the big buzz about programs attempting to add direct editing to their History-based systems is now doesn’t it ☺

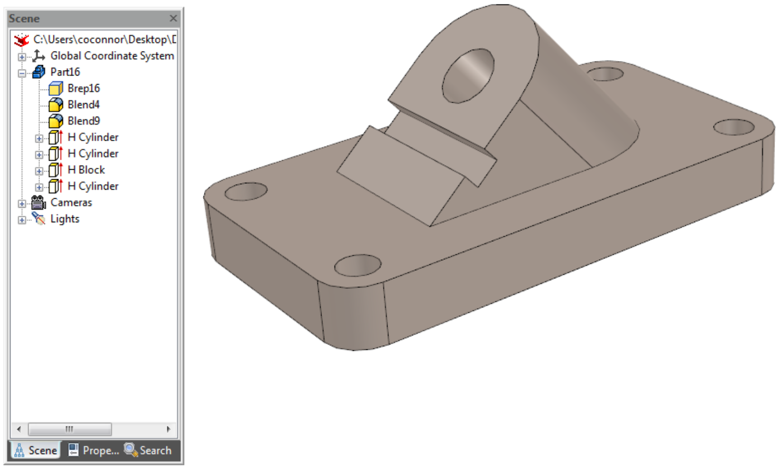


Result of the Move Face with Mixed History Results of Features and Direct Edited Features

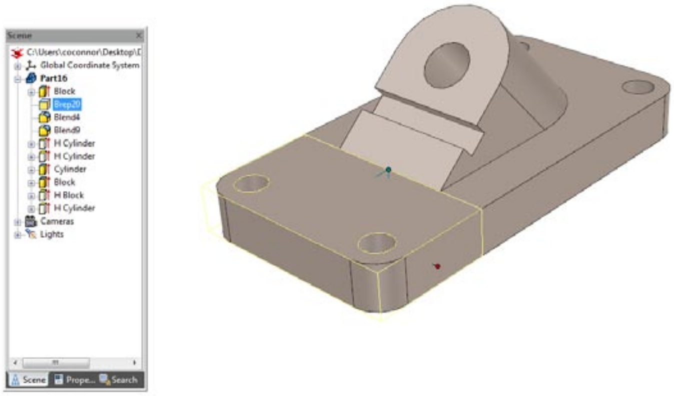
Basically what is happening is IronCAD gives the users the power of both design methodologies in a single part without the need for a user to figure out what modeling technique is needed for a given situation. Simply make the change and the system will do it for you (while informing you what is happening). Let me expand on this point a bit more where this really comes into play.



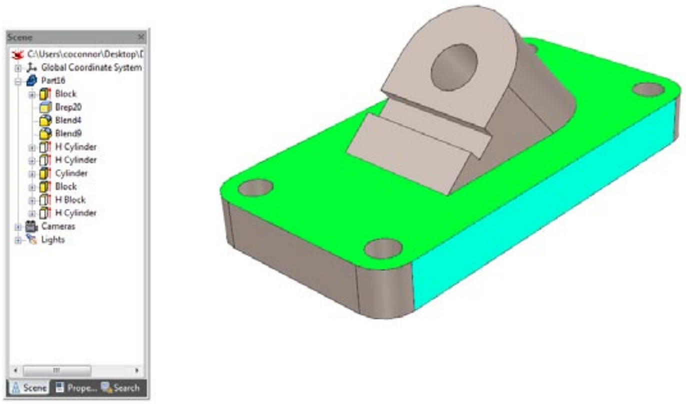
Let's start with the example above where we built the model using history base features. There is not really in any particular order but I did make it a point to make the bottom leg a separate feature for this next illustration. Say that our design has changed in that the leg at the bottom needs to be rotated to be parallel with the top block so that we now have a flat mounting block. Of course, we could alter the history and make this modification since I know how it was built, but let say for the sake of this example the best way to achieve the modification is a direct modification. So we simply select the faces needed for the modification and rotate it to the desired location (sounds easy…really it is…).



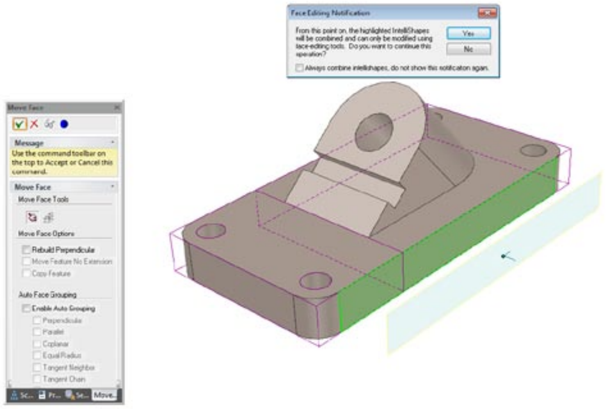
Notice again, that IronCAD automatically determined what is needed for the change and combined them for the user and left the features that are not impacted by the change. Where else can this type of automatic determination make it easy for the user? Take a look at this example:



In this example, we end up where there is a Direct Modeling Feature result in the mix of history features. The interesting result is that the result causes a merge of the face results (notice the green and blue face below):



Now we want to move one of these faces that include both a direct edit feature and history feature. To an IronCAD user, there is really no issue. You simply select the face and perform the Direct Face Modification. Again, IronCAD will intelligently make the determination of how to handle the features and perform the modification.



From this point on, the highlighted IntelliShapes will be combined and can only be modified using face-editing tools. Do you want to continue this operation?

□ Always combine intellishapes, do not show this notification again.

Use the command toolbar on the top to Accept or Cancel this command.

□ Rebuild Perpendicular
□ Move Feature No Extension
□ Copy Feature
□ Enable Auto Grouping
□ Perpendicular
□ Parallel
□ Coplanar
□ Equal Radius
□ Tangent Neighbor
□ Tangent Chain

Automatic Handling of a Move Command on the Merge Face of Direct and History Based Features

Getting back to the inception of IronCAD, the goal was to make both modeling paradigms available to the user without the overhead of trying to figure out which is right for a particular task and what portions of the features are what. Users are tasked to design and not learn a design philosophy of a modeling system and that is what IronCAD set out to accomplish. The flow is build your feature based model and modify them as features or move faces when you want, and the system will guide the process along. This process is the same if it was a native model built in IronCAD or imported data. For imported data, IronCAD can use the full power of the direct editing to modify the model that contains no history features at all (or again a mix if the user adds features to the import and wants to modify in one, the other, or both conditions).

Now I must admit that a lot of cool features have developed in the market since 1998 in the area of direct editing and IronCAD is lacking a few of these cool features. However, our development has a high focus on customer demands and to be honest, not a lot of customers have requested these newer items at a high frequency. However, we do see the benefit in a few of the newer development items and IronCAD is actively developing to incorporate these newer items in future versions. We believe the reason we don't see the requests as much is that IronCAD is a Hybrid system of Direct Editing and History-based Features while supporting the intelligent modeling rules mixed in a single part. Many of the modeling system hyping the technology are starting as one or the other and attempting to merge the result. Therefore, it was more logical to add commands found in one mode into the other mode to support a consistent capability. For example: Direct Modeling Pattern Dimension. While this feature is cool and useful (especially on imported parts), it was not as critical to IronCAD users. Users would simply build the patterns with the history features and make direct edits where necessary. It was not overly common to want to define features as direct editing features and build a pattern driving direct edit commands. Mainly the direct modifications rarely affected patterns and that IronCAD has an advance Direct Editing option that can limit the effect to a feature instead of a global definition (we can discuss this in an advance session). However, we do feel this is useful for imported parts and I would expect this to become reality soon.

Overall, I hope this better explains the concept of IronCAD and how and why the product came to be. Clearly there are reasons why the industry is so active in attempting to achieve the single part history and non-history design capabilities. Simply put, there are cases that are simply painful to resolve in one or the other mode. IronCAD achieved this feat but simply was overlooked or misunderstood by the industry at large. In summary, let me bullet the main concepts that truly make IronCAD unique and the leader in the industry for innovative design:

- "Dynamic" History Structure Ordering that automatically alters the feature order to achieve logical changes to the model the user is attempting. Rigid History-Based modeling systems, require the users to understand the feature tree structure and manage it to make changes. It becomes even more difficult for these users to accomplish changes in the structure when many dependencies are related to modified features.
- "Automatic Grouping" of features needed to perform direct face modification. The system understands what it will take to perform a direct face modification and clearly informs you what features will be combined to achieve the result.
- "Mixed History and Non-History Feature Environment". Having either a history-based mode and a non-history mode will not always achieve the correct results in design. It requires a true mixed

feature tree that uses history in the mix of the both types of features not just a grouping of one or the other (Refer to the P.S. Section below). Requirements in design sometimes require a sense of feature order and without this, users would be required to added additional geometry to achieve the results.

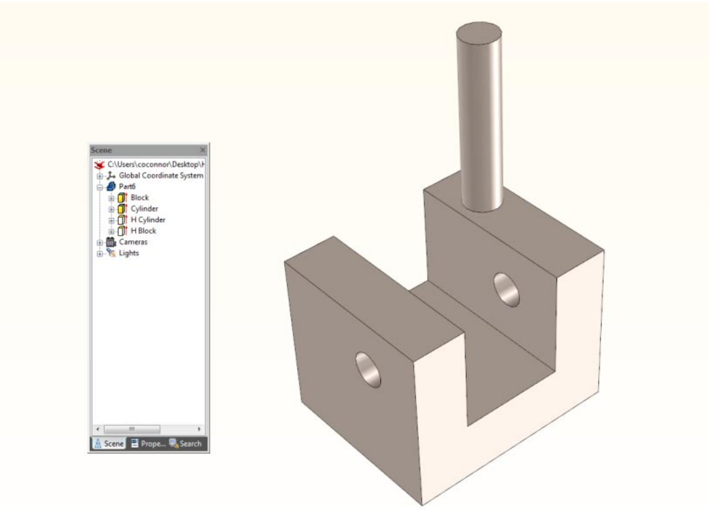
- Note: "Recreating Features". I didn't cover it in this article but it is worth a mention. IronCAD also supports the ability to select faces from imported data or direct face modification features and convert them into features. While this is not always possible to achieve, it is useful in cases to recreate features for advance modification control (such as blends or recreating the ability to define the sketch).

Since those of you reading this article are IronCAD users, you have long understood the advantages of the product and have experienced the productivity it offers by reducing critical design time adjusting history dependencies that others combat often. Now that you have a better understanding of the core technology of IronCAD, keep an eye out for technology reviews online and voice your knowledge. Any exposure you offer on IronCAD helps our product grow and extends our ability to achieve more capabilities to benefit all users. If you have any questions, feel free to contact me or the IronCAD team.

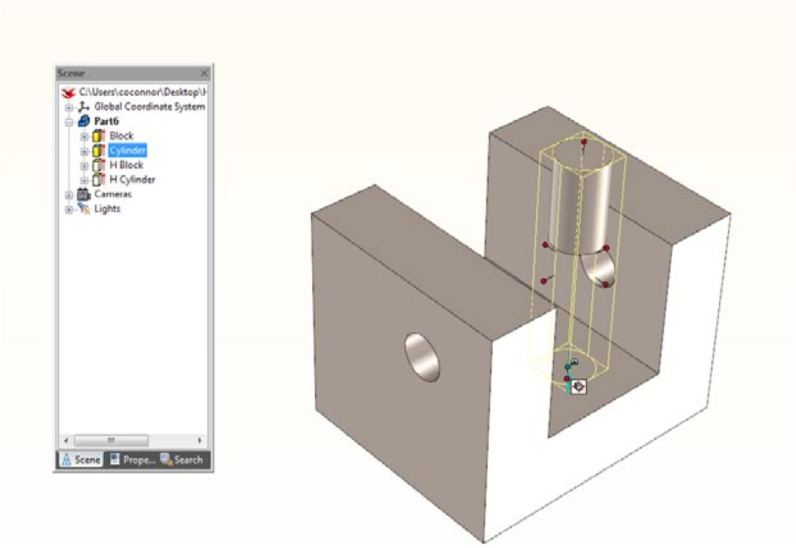
Thanks

Cary O'Connor

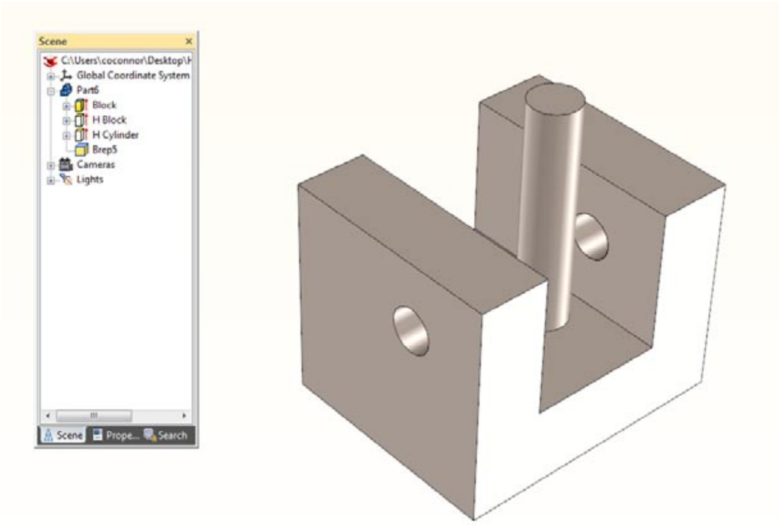
P. S. Here is an example why a hybrid system is needed for some cases. When I say hybrid, I mean that it uses both history-based features and direct features while leveraging history order rules.



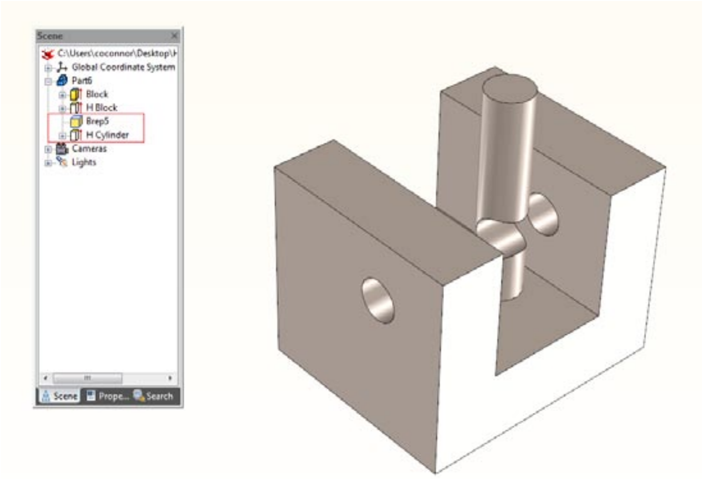
Simple Block, Cylinder, H-Block, and H-Hole example in that order (simple feature example to illustrate a point. Consider the Cylinder a group of faces for example). The goal is to move the cylinder to the inside bottom face of the H-Block. However there is a rule that the H-Hole need to cut through all geometry in that area.



If you move the hole as a history-based move, the result is that the H-Block and H-Hole cut the Cylinder.



If you do a Direct Editing alone, the H-Hole will not affect the Cylinder (same issue for History-based modeling systems that support direct editing as a feature command since the new feature would be added at the end of the history).



Therefore, you need a mix of History order and Direct Editing to achieve the end result.