

MYTH BUSTER: REVIT & IFC, PART 3

I've given a lot of thought as to how this trilogy of articles should end. At first I thought I'd use the Coordination View 2.0 EXPRESS-G Wallpaper for IFC2x3 (http://www.mdr-advies.nl/wp-content/uploads/CoordinationView_V2-0_WallPaper_IFC2X3_Version-1.pdf) as a stepping stone, to give an "IFC for Dummies" type of solution. But that turned out to be quite lengthy and rather boring. But the link is included above for obvious reasons—it's useful!

So I went back a little and asked myself, "Why IFC?" And with that question, the inevitable follow-up immediately appeared: "Why not IFC?" I realized that in the previous articles I failed to explain why, in my humble opinion, we should care about this somewhat crooked, badly implemented, severe headache-inducing format. Therefore, instead of explaining the fundamental technicalities of IFC that can be explored in other places, this final installment will explore the Why versus Why Not IFC discussion and begin with the latter.

WHY NOT IFC?

When everybody uses Autodesk® Revit®

Well, in this case it's a no-brainer. Share your Revit files. Period.

Loss of data

This is an easier way to share data, instead of hacking your way through IFC and praying your stuff doesn't get lost in the process. Even when you manage to export your geometry through IFC, there's always loss of data.

Loss of intelligence during import

Even if you only need to send over geometry, not data, importing that into Revit makes it utterly useless. All constraints are lost, component parametrics are gone, and so on.

Availability of bidirectional links

When not everyone uses Revit on a given project (which is quite often when you're not located in the U.S.), why not simply use a bidirectional linking between proprietary software standards? This can be easier and better to control.

WHY IFC?

Which may be asked as: why would you want to use IFC? Or more accurately, why would you want that option available?

Consider what Sir Arthur Conan Doyle wrote: “Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth.”

Let’s put that to the test, shall we?

When everybody uses Revit

Okay, simple math: Autodesk has around 7,000 employees. The world has about 7.2 billion people. What are the chances someone not working for Autodesk has a brilliant idea and creates the software you need tomorrow? There will always be better solutions. They might not be viable for you overall, but there’s no reason to shut down access to them. Revit is a wonderful generic tool for which I see a brilliant future with all kinds of additions for specific niche-markets like AutoCAD has.

What about today? DDS is MEP-engineering software that can do energy calculations (not simulations!) inside your model. Rhino lets you model things on the fly that could (arguably) take ages in Revit—if you ever get them done. Tekla lays waste to the Revit structural rebar functionality any day of the week, without breaking a sweat. Solibri is a wonderful solution for clash detection and, most importantly, rule-based model checking and reporting—something Revit (or Navisworks) doesn’t do.

Should I go on?

Only problem for all this magnificent software: It’s either based on the IFC format or requires model exchange through IFC.

The fact of the matter is there will always be better software. Any one may be the software for you, but that doesn’t mean your partners, associates, etc. need to have the choice to use it forced upon them.

Loss of data

On July 1, 2013, there was an extra release for Autodesk software. The Sourceforge Open Source IFC Exporter released a new version that incorporated the option to map custom Revit parameters (shared or non-shared, family or project, even calculated values) to their IFC counterparts.

On September 28, 2013, I had the honor of showing the second step in a complete and controllable data export from Revit.

During my RTC Europe class, I showed the new option to create a custom IFC Property Set and map any given Revit parameter into IFC, fully customizable.

IFC does not automatically lead to loss of data. That idea is simply not true. Incomplete implementation in Revit does lead to loss of data. As of now, you are able to put any piece of data into an IFC, even calculated values. And we didn’t change the IFC format. We just improved implementation in Revit.

October 2013

Loss of intelligence during import

“Just because something hasn’t happened yet doesn’t mean it’s impossible.”

I don’t know who originally had this insight and I doubt it to be the movie line I remember, but it is true.

In the same RTC Europe class I showed a native Revit family being exported to IFC, altered, and re-imported—while keeping all parametric constraints and even adjusting to the alterations made directly in the IFC file. It is possible, but not with the current importer.

Availability of bidirectional links

I’ve been using Revit for almost 10 years now. For 4-5 years I didn’t know about IFC at all. Before a year ago this was my strongest argument against IFC: Why would you need a generic “language,” with all its limitations (of which there are), weaker implementation and use of parametric capabilities (because it has) and lag-times in development (by nature)?

IFC will never be able to do what native software does, simply because it’s responsive in nature. Development budgets are a fraction of what Autodesk, Trimble, or Nemetschek get to spend on their software.

So why not keep it simple and use a bidirectional link? Revit has it with a lot of other Autodesk software and third-party applications.

But how long have we waited for them? And are they even truly bidirectional? Do they exactly do what you want them to do? In my humble opinion the answer to these questions is not very inspiring. Even bidirectional links tend to have major limitations and severe constraints in flexibility. And most important: The link itself is also proprietary software. No way of adjusting it to your needs.

WHAT IF MORE PEOPLE STOPPED TALKING ABOUT IT AND JUST GOT INVOLVED?

The improbable

First off, it’s a matter of simple logic. As Sir Arthur suggested, if we can successfully check off all likely options to NOT want to use IFC, doesn’t that mean the improbable is true? That IFC is in fact a valuable tool in our toolkit and we should be able to use it when necessary?

Mind you, I’m not saying you should use IFC. If you’re on a Revit-only project, it would be insane. Just send and receive RVTs. The fact is this will only be viable for limited parts of the building lifecycle. If you’re lucky, the entire design team is on Revit. But manufacturers have other software; facility managers, too.

As for the design team, what if the best structural engineer is not [on Revit], or the MEP guy that has loads of experience in similar projects? Would you settle for second-best just because someone uses the same brand of tools you do? I wouldn’t.

IT'S THE "I" THAT MATTERS. IF WE'RE COLLABORATING, THE INFORMATION IS MOST ALL I CARE ABOUT. I JUST WANT YOUR DATA—IN THE LEANEST, SMARTEST, AND MOST EFFECTIVE WAY POSSIBLE.

Open Access

Which brings us to the biggest advantage of IFC (or any open format): it's open! Which means it's adjustable to your needs. Now don't go "that's too difficult / expensive." That's bullocks.

The first external enhancement to the Open Source Exporter was created by Tom Pesman, a self-employed API specialist.

Parametric IFC?

Implemented by Jon Mirtschin, also self-employed, IFC parametrics is available. Agreed, Jon is a well-known IFC guru but still, the point holds.

Custom property sets and parameters?

Wim Tas, owner of a small MEP engineering firm, has added functionalities toward this end and he's not even a true API specialist.

All these developments were done for free by altruistic, volunteer efforts as part of the creation of the Dutch Revit Standards. And, of course, with the invaluable help of the Autodesk Open Source IFC Exporter team, Angel Velez in particular.

But what if these guys actually got paid? What if this project wasn't a Saturday night side-job? What if more people stopped talking about it and just got involved—either by themselves or by leveraging experts to get what is needed? The entire IFC format is Open Source, the source code is out there, which basically means you can do whatever you want and create whatever you need—even on a project-by-project basis.

BIM: IT'S THE "I" THAT MATTERS

In my humble opinion the entire debate on BIM is focusing on the wrong part of the acronym. I don't care if "Building" is a noun or a verb. Nor do I care whether it's "Model" or "Modeling," nor am I commenting on whether one should say "BIM Model."

It's the "I" that matters. If we're collaborating, the Information is most all I care about. I just want your data—in the leanest, smartest, and most effective way possible.

A Revit model will do, conversely a lot of times. A Revit model will give me heaps of data I don't want, geometry I don't care about, and a file format I can't use. Now we can all keep on being really anal about it or we can simply use the tools available to us, one of them being IFC.

A FINAL NOTE

As stated at the beginning of the article, I really enjoyed writing this series and for now, this is the last of the series. But if you're interested in more tips and tricks on IFC, let me know. I might be able to persuade the guys at AUGI to let me write about it again. Lord knows there is still a lot to talk about regarding IFC.



Martijn de Riet is a self-employed BIM Consultant from the Netherlands, working with Revit since version 5.1. Martijn has a bachelor degree in Building Science. After his studies, he started his own engineering firm working for contractors, architects and private clients. Starting in 2007 his company transformed into a full-time BIM consultancy service. At the moment Martijn's clients vary from mid-sized architectural firms to the largest dutch General Contractor and MEP Engineering firms, with a focus on specific corporate solutions, design and implementations of Revit and BIM workflows. Martijn is a member of the Dutch Revit User Group and currently working on creating a Master Template and Component Library. He provides lectures for companies, technical universities, seminars, etc. on a regular basis.