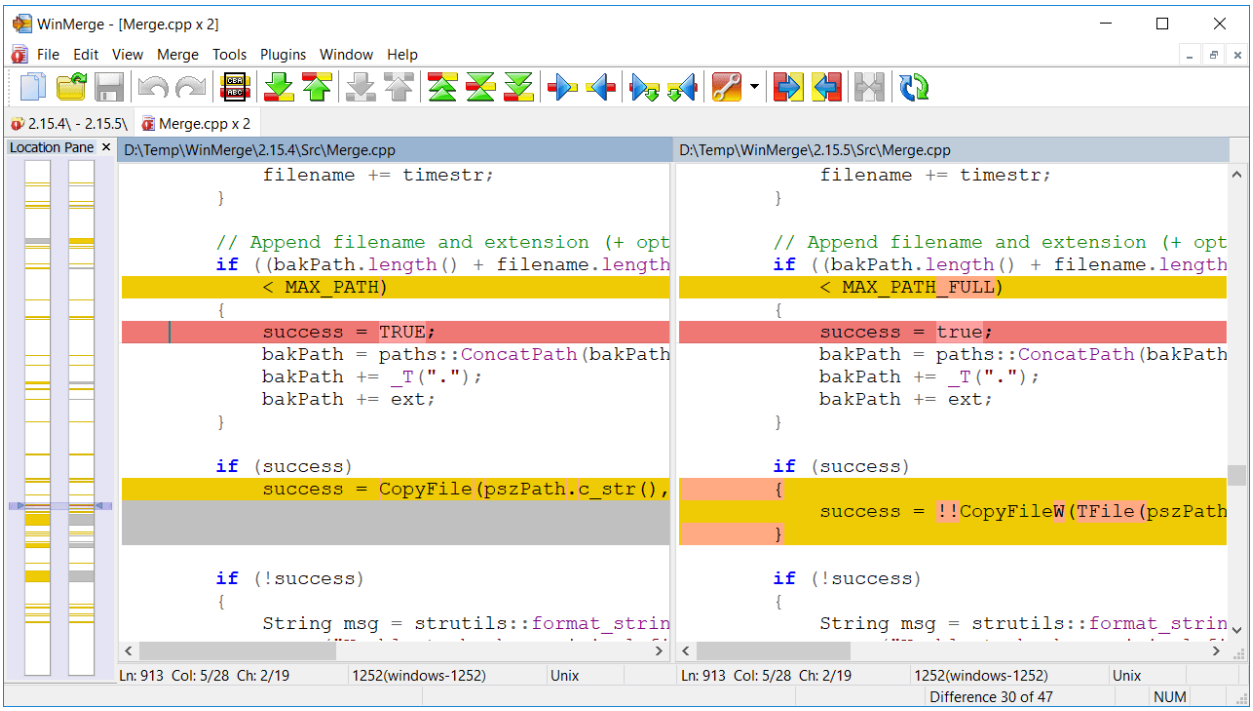
Merging Rules With… Other Rules?

Introducing the irAuthor Merge Extension

What is a merge tool?

While InRule is designed to empower business users to author their own rules, there will inevitably be folks from the IT side of the organization that interact with business rules at some point – be it for data integration or rule authoring purposes. One of the tools that software developers and IT professionals often develop a strong affinity for is an application that displays a line-by-line comparison between two different files, allowing them to see precisely what changed. Being able to know precisely what you're changing in an environment (be it configuration files or code) is important for reducing the potential for errors to slip by. Combining the "Diff" tool with a code repository also allows for things like partial merges, where only the specific changes you want are included. This combination of comparing and selectively including changes in an output result is the primary purpose of a Merge tool.

Since many folks have become accustomed to using it in other aspects of their job, it's only natural that we'd receive requests for a similar tool that can work with a Rule Application. While that seems like it should be a relatively trivial thing to accomplish, it is substantially more complex that it would at first appear.



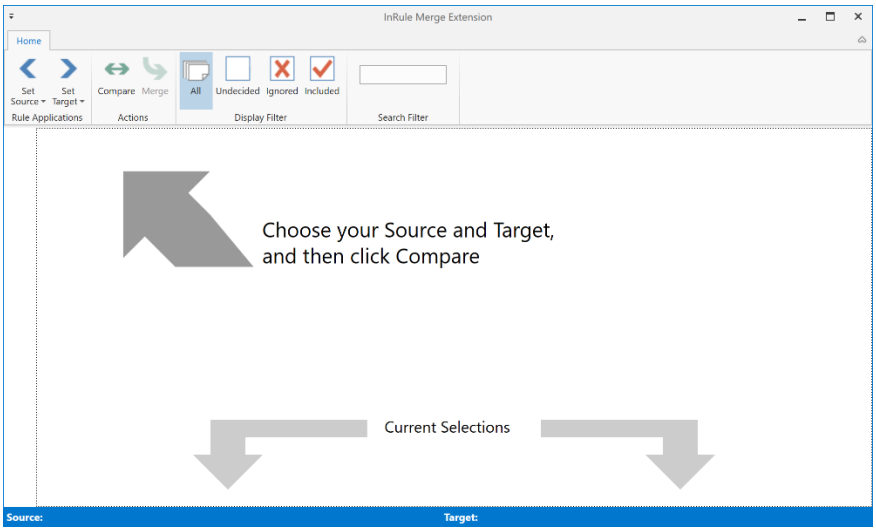
Why not just use my favorite merge tool?

Software code, configuration files, and most of the file types that common comparison tools operate against are generally fundamentally text files – they simply are interpreted in different ways by the consumer of the file. Because text is a very simple data type, it's a straightforward task to build a comparison between two different versions of a file. Rule Applications, on the other hand, are substantially more complex.

Things start off simple - ruleappx files are compressed archives of the ruleapp file, so can be decompressed into a folder structure containing a ruleapp file (which is stored as XML). Unfortunately, the resulting XML file is not a simple thing to deal with. Without going into too much detail, the XML used to store a ruleapp is a an extremely large, dynamically generated, flattened collection of hierarchical objects that are compiled into the Rule Application based on lookup keys that may change between environments. As a result, performing a text-based comparison of the XML will provide an extensive, difficult-to-parse list of changes with lots of false positives, and little to indicate what each changed GUID was referring to. Once you have a set of differences, merging them together in XML (and ensuring that you have all dependencies included) would be farily impractical.

I'm now sufficiently bummed out. What's this post about, then?

After implementing a handful of one-off utilities for customers based on their specific needs (which our ROAD team < https://www.inrule.com/professional-services/custom-services/> is great at), we identified a set of functionality that would cover the majority of merge-related use cases that our customers had encountered. We pulled a few bits and pieces from previous tools, added in an entirely new difference engine and UI layer, and are now quite pleased to announce the availability of the InRule irAuthor Merge Extension! Unlike other irAuthor extensions (offered as unsupported source code), the Merge Extension is offered as a new category of Managed Extension – with available binaries and installation material.

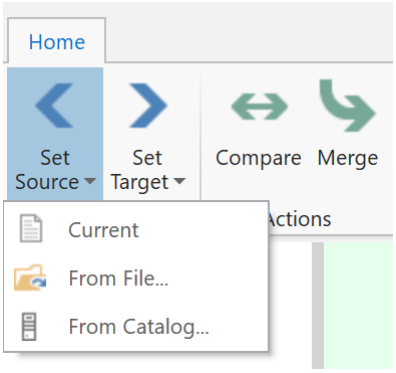


Awesome! What does the extension do?

This may come as a surprise, but… it loads and compares two Rule Apps, shows you the differences, and then allows you to merge them together. Let's dive into each of those functions

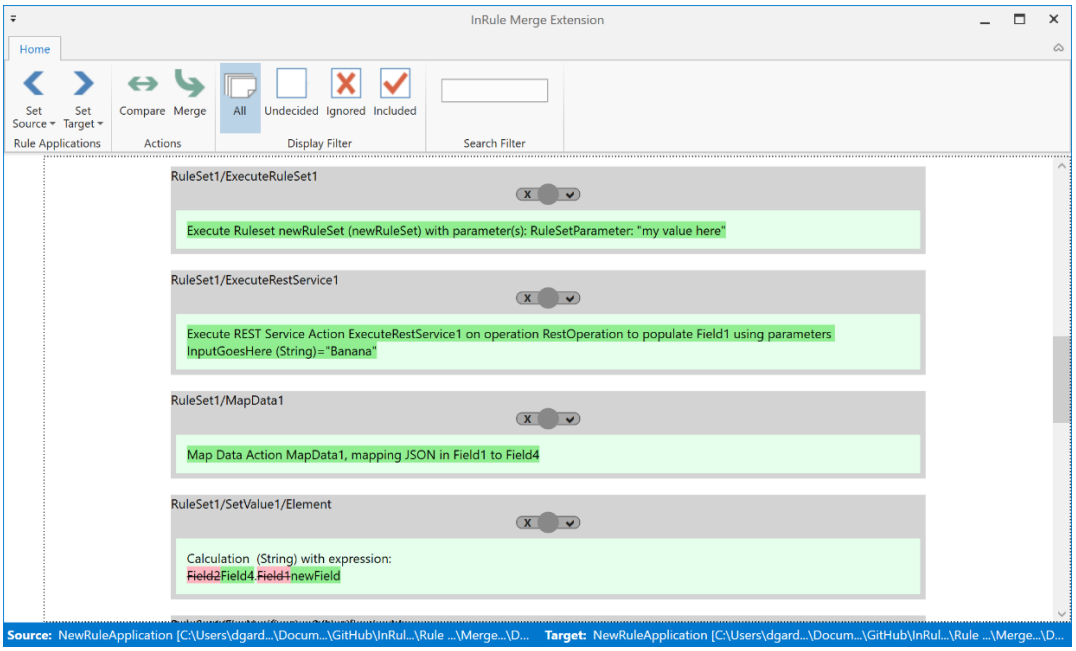
Load

Both the Source (the version that has changes you want included) and the Target (the old version you want to update) Rule Applications can be loaded from a .ruleapp or ruleappx file, from any irCatalog instance you have configured in irAuthor, or can be loaded from the currently open Rule Application.

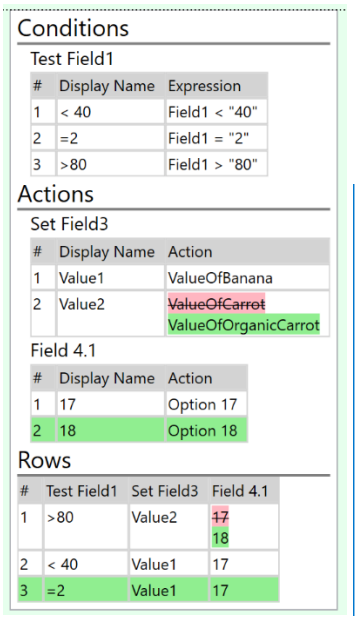


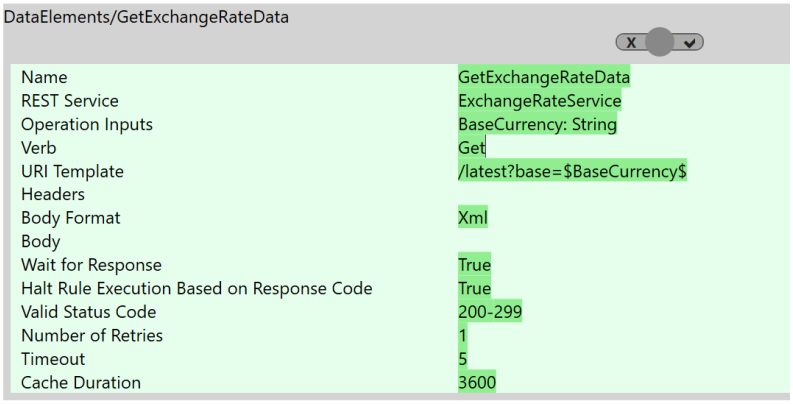
Compare

Once both Rule Apps are loaded, click the "Compare button, and the comparison engine will "walk the tree" of the two Rule Applications, hierarchically determining what is different between them. Each difference is then displayed in a box with a description specific to each object type, with color and font indicating items that were added to or removed from the Source Rule App.



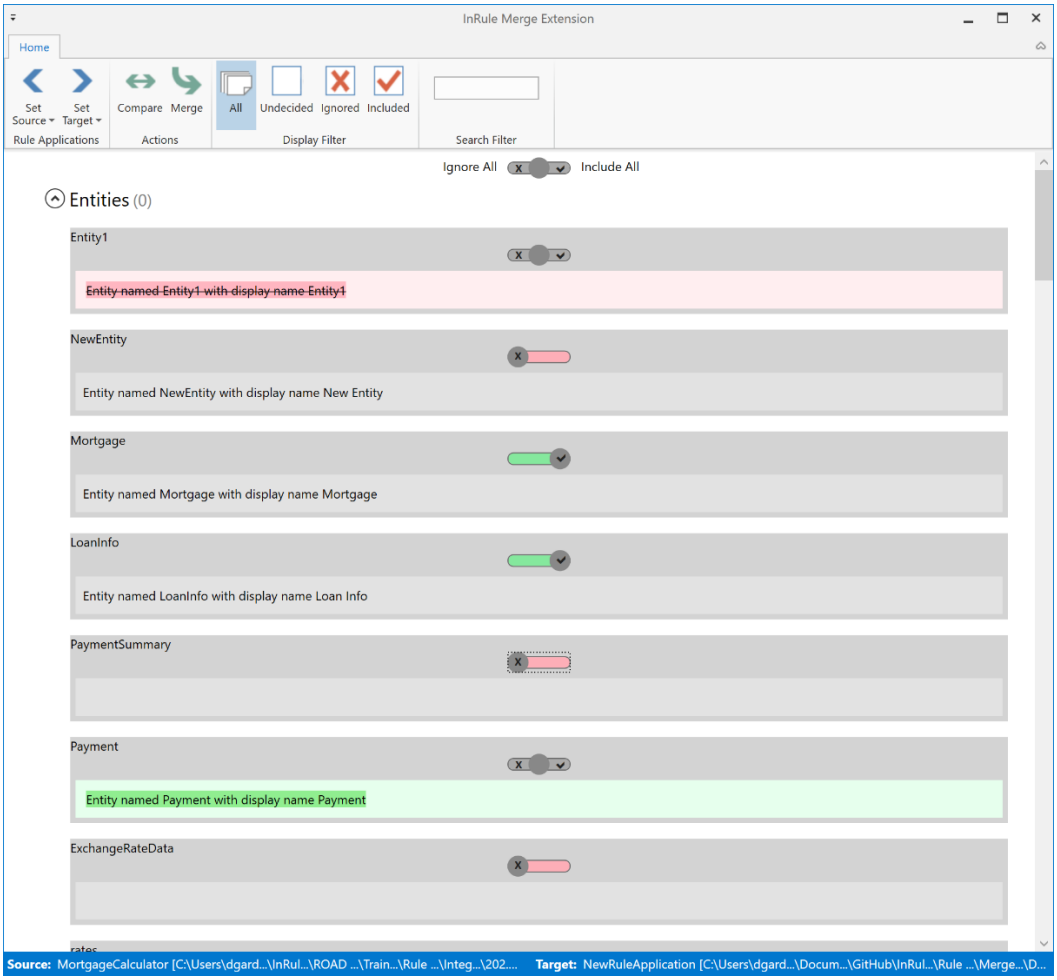
We spent a lot of time focusing on how to make the interface easy to understand given the wide variety of different object structures in InRule. To that end, we made custom display information for over 100 different object types, trying to show all relevant information for each in a concise, easy-to-understand structure. For some data types, this can be as simple a sentence of text, more complex objects may take the form of a table of data; for Decision Tables, it takes the form of a series of tables – all with the end objective of ensuring you can tell what each change contains.





Select

Once you've reviewed a change, it's time to decide if you want it included in the final output of the merge or not with the slider in the center. As you work through the list of differences, choosing to include or exclude the change will update the comparison box to show what the final value will be at the end of the merge, allowing you to easily identify items that you've yet to make a decision about. If an item is not explicitly included (IE if you ignore the slider), it will not be included in the merge.



Merge

At long last, you're ready to make magic happen! Once you've selected the items you want to include, clicking on the "Merge" button in the ribbon will apply the selected changes from your Source application and move them to a working copy of the Target application. After a successful merge, a new instance of irAuthor will open with the final, merged application, where you can validate everything looks good and save the merged Rule Application wherever you need to.

I'm so excited to use it! How do I get it?

Since this is a Managed Extension, you don't need to download the code and compile it yourself – simply head on over to our Extension Repository <https://github.com/InRule/irAuthor-Extensions/tree/master/MergeExtension> and follow the instructions in the "Installation" section.

We can't wait to hear how you use the extension – comment below and let us know! Find a bug? No problem, submit a report to support@inrule.com and we'll take a look.

What's next?

This extension has a range of potential applications:

- Reviewing pending changes before checking in
- Performing code reviews of previous check-ins
- Validating the updates that will be deployed before Promoting a Rule App
- Merging updates made in two different copies of the same Rule Application

That last potential application might have caused a few ears to perk up – the ability to merge two copies of a rule application together is one of the prerequisites to enabling another widely-requested piece of functionality: Branching. While we are not currently in the process of implementing branching, it is on our long-term roadmap to build out; getting the Merge extension in the wild was the first step in that direction.

Happy Merging!