

8.2.30 LitSavant Conformity Engine

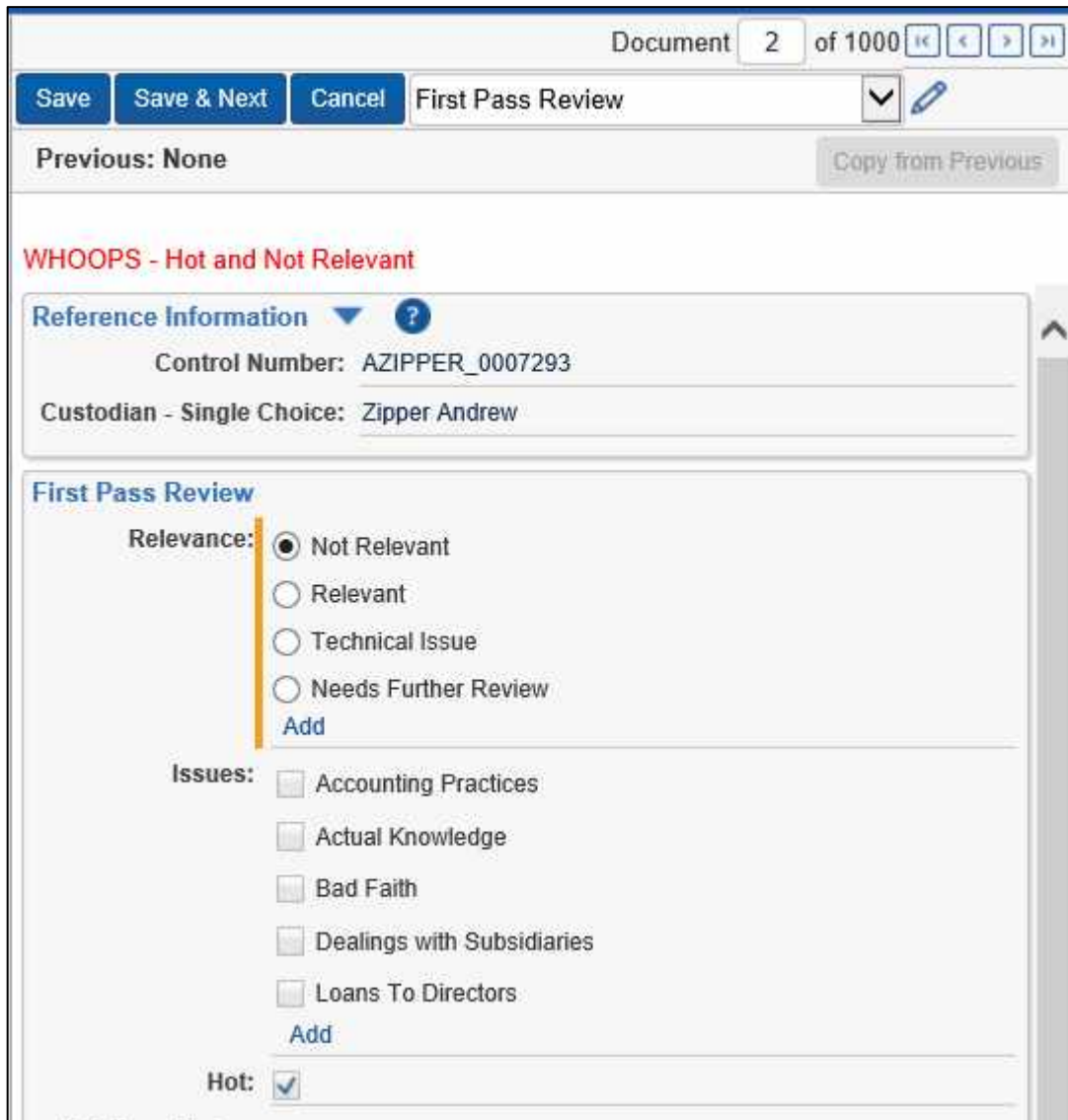
The LitSavant Conformity Engine is an application that works within Relativity and RelativityOne, which enables Project Managers to design and implement validation rules and other logical processes which are then enforced during data entry.

This works by allowing the user to specify what they want the software to do and the conditions that need to be fulfilled in order for the software to do it.

Here are 5 easy examples illustrating how these rules can be used and why they are very useful in increasing accuracy and reducing costs:

Example 1 – Alert the user when they get it wrong

This is the commonest scenario and results in an onscreen message when the user codes something in a way that doesn't meet predefined rules.



Example 2 – Log mistakes (when users get it wrong)

When a user makes a mistake (such as coding a document both “Hot” and also “Not Relevant” as per the example above), in addition to alerting them so that they correct the error, we can also log the mistake. The mistakes can then be analysed to identify recurring themes for corrective action etc.

New Mistake		All Mistakes		
0 Selected Item(s)				
#	<input type="checkbox"/> Name			Mistake Type
1	<input type="checkbox"/> MSTK1005	Edit		Hot and Not Relevant
2	<input type="checkbox"/> MSTK1002	Edit		Missing Privilege Decision
3	<input type="checkbox"/> MSTK1001	Edit		Missing Issue
				Mistake Made By
				Dingle, Mark
				Dingle, Mark
				Dingle, Mark

Example 3 – Send an email when hot documents are found

This example is pretty self-explanatory - when a user identifies and codes a document as hot for the first time, the software can send a customised notification email to specified users. The email could include information about the user who tagged the document as hot, any comments that they made and any additional coding from the document.

Example 4 – Update the “Last Coded By” field

It’s not uncommon to want to know who reviewed a document during a particular review round. The software can update a “Coded By” field – saving the user from having to enter this information.

Example 5 – Update a “Family Privilege” field based on “Privilege” coding

Relativity propagation is a pretty blunt instrument in that the last coding decision on a family member is applied to all members of the family. If we want to know whether a family contains a privileged (or part privileged) document, standard Relativity propagation doesn’t really do the job.

Instead we can create a custom action to use “Intelligent” propagation. With this approach, users code the “Privilege Status” of each document and the software then updates a field for all members of each family based on that coding to indicate if they are members of a Privileged, Part Privileged or Not Privileged family.

Detail

The examples above illustrate the 5 functions that the software can perform as part of the coding process. These functions are:

- Show an onscreen message
- Create a new instance of an object
- Send an email
- Update a field
- Execute a Relativity script

One or more of these functions can be triggered when the specified conditions are met. Each of these functions is customisable and can be triggered by one or a combination of conditions. The conditions themselves are also fully customisable.

Whilst all of the examples above relate to coding documents, these functions are available on all custom objects as well.

Each of the functions above could be created by a programmer in code and deployed via an Event Handler. The innovation in the LitSavant Conformity Engine is that Relativity’s standard interface is used to enter the rules – no programming knowledge is required and so the rules can be created, tested and deployed in minutes. And because the LitSavant Conformity Engine saves your rules into your Relativity

Chapter 8 – Supplier & Software Details

database when it is time to upgrade Relativity, you simply update the application as part of the upgrade process. If you were using Event Handlers you would need your programmers to manually rewrite and redeploy each of your Event Handlers in all of your databases.

The LitSavant Conformity Engine supports Relativity Server versions 10.3 and above (up to and including the current version) and RelativityOne. It is licenced on an annual basis and is available to anyone with their own Relativity environment.

Owned/Supplied by :	LitSavant
Used by :	Various UK, European and US organisations with their own Relativity environment