INSURANCE TECHNOLOGIES

Wizard Performance 2.17

# FIRELIGHT BASE

FireLight®

Platform

**WIZARD PERFORMANCE 2.17**

**Document Version: 1.x**

**Published: February 20, 2020**

## Insurance Technologies, LLC



**Insurance Technologies, LLC**
Two South Cascade Avenue
Colorado Springs, CO 80903
USA
Phone: 719.442.6400
FAX: 719.442.0600
Internet E-Mail: info@insurancetechnologies.com
Website: http://www.insurancetechnologies.com

# Table of Contents

# iConnect Design Approach 218717 - Wizard Performance 2.17

Enhancement to research and improve the performance of wizards. This includes opening and loading a wizard, editing fields in wizards and saving wizards. This will be a multi-release approach with several changes being introduced in April 2.17.

• Reduce the number of queries.

• Eliminate double save/initial save.

• Explore caching rules and forms definition.

• Improve field update speed in wizards.

Impacts:

- Opening a new wizard activity.
- Open new wizards and forms activity.
- Updating wizard fields.
- Saving new wizard activity.
- Saving wizard and forms activity.

## 1 Research wizard performance issues

Research the performance issues in the wizard. Run analysis in QA Time (It Networking) or Profiler.

## 2 Reduce number of queries

Recommend reducing the number of queries.

1. For each form/wizard in an application, the FormLibrary table is queried for the timestamp. Modify this to use a single bulk query for all forms in the application instead of individual queries.

2. Optimize GetApplicationRules - for forms with a primary rule set, duplicate rule definitions only for those that actually contain [FORM_*].

3. Evaluate optimization of de/serialization of rule, wizard, form definitions, form data, etc. Replace DataContractJsonSerializer with a faster option.

https://michaelscodingspot.com/the-battle-of-c-to-json-serializers-in-net-core-3/

***Acceptance Criteria***

- Reduce time to create or load activities.

# 3 Eliminate double save/initial save

When a new activity is created and in some cases when it is opened the related application forms (and possibly other records) are deleted and inserted into to the database two times. For activities with many forms, this adds unnecessary overhead to the load process, and it is inconsistent with allowing the user to choose when to save an activity. Modify the system to not save this data at all during activity creation/load, and instead write only when saving the activity. Also, modify it to write to the database only if it is different from the previously queried data.

***Acceptance Criteria***

- Reduce time to create or load activities.

# 4 Explore caching rules and forms definitions

Explore caching rule, wizard, and form definitions. These can be large and storing in the cache manager and loading from there if found could reduce time querying and deserializing from the database.

However, we cannot affect a common test scenario of changing a rule, wizard, or form, then reloading the activity and immediately see the changes. We cannot remove the cache item when something is updated because there are multiple presentation servers, each with its own cache. Investigate whether querying for the timestamp to invalidate from the cache would continue to justify caching. The query should be done in bulk, not using a separate query for each rule set/wizard/form.

Also, cache the results of GetOrganizationAssembly to minimize DB lookups and instantiation of the provider class.

***Acceptance Criteria***

- Reduce time to create or load activities.
- Caching will only reduce load times in cases where the data has already been cached. Items are removed from the cache after 20 minutes and will be reloaded from the database when not in the cache.

# 5 Improve field update speed in wizards

Updating fields in wizards takes several seconds.

***Acceptance Criteria***

- Reduce time to move from field to field in wizards.