

# iConnect 136499- Enhance FL to support Rules that can drive Packages

---

## Project Overview

Enhance FL to support Rules that can drive Packages.

## Features/Requirements

In existing system, we have a rule node (setformstate) by which we can show/hide any specified form in an application. Currently we don't have any rule node by which we can show/hide whole package. This new enhancement supports this new functionality. Just like "setformstate/ setformstatebyname" node, we need to create new rule node "setpackagestatebyname" which can show/hide whole package in an application.

## Use Cases / Workflow Changes

The SalesAgreement will now allow the admin to set a package with Duty Type of "Required" or "Conditional". Only conditional packages will be affected by this new Rule, Required Packages will always be Load. Rules that drive packages should be put in the required packages, because rules will not be run in excluded packages.

## Shared Changes

This project contains one/more changes in shared project.

- PackageDuty is a New Column in SalesAgreementPackage (Defaults to Required).
- GetSalesAgreementPackagesByRoleCodes and GetSalesAgreementPackages now get only required packages by default. But will also pull Conditional Packages Associated to application Id. Which is based on ApplicationPackage Table
- ApplicationPackage and ApplicationForms Will be updated when rule runs.
- New Interface(s) RegisterRemoveAddPackages, RemoveAddPackages, ReInit, and ReInitializeFinished for rules to callback and add packages, and reinitialize rules.
- Changes to Rules Engine Will allow for call back to add new forms and reinitialize the rules.
- We need to add a new property "PackageName" in *FormItem* class so we can have package name with every form.
- We need to create a new rule node "setpackagestatebyname" to set state of package. Under this rule, we need to modify "IsExcluded" property of every form for specified package.

## Admin

Forms Will Implement New rule where necessary.

Sales Agreement need to add Duty Type to Conditional Where Applicable.

## App Changes

We need to create new method “GetPackageName” in DBManager class to retrieve “form guid” & “package name” list for an application and used in EditApplicationModel class to set “package name” in “FormItem” list before passing it to ServerTriggerHandler, which can be used while executing “setpackagestatebyname” rule.

## Integration Changes

There is no change in Integration Process.

## UI Mock Ups

Selling Org

Insurance Technologies

Product Name

Test Conditional Packages

More Info

Start

10/05/2016

End

10/05/2056

Time Zone

UTC - Universal Coordinate Time

Replace Product

Select Replaced Product

Last Updated

10/10/2016 10:50 PM UTC

State Approvals

Back Office Script

Options

State	Approved
Alabama	<input checked="" type="checkbox"/>
Alaska	<input checked="" type="checkbox"/>
Arizona	<input checked="" type="checkbox"/>
Arkansas	<input checked="" type="checkbox"/>
California	<input checked="" type="checkbox"/>
Colorado	<input checked="" type="checkbox"/>

☒ Check/Uncheck All

Available Packages

AA Application - Package 1  
Ajay\_Test\_PKG1 Application - Package 1  
Ajay\_Test\_PKG2 Application - Package 1  
Ajay\_Test\_PKG3 Application - Package 1  
BR Pack Application - Package 1  
BR Pack2 Application - Package 1  
BR Test Application - Package 1  
COL Pack Application - Package 1  
Console Package Application - Package 1  
Copy of IT Package Application - Package 1  
Copy of Multi Signers

Assigned Packages

Test Conditional Package Application - Package 1  
Package Duty Required  
  
Conditional Package Application - Package 1  
Package Duty Conditional

Save Sales Agreement

New Package Duty in Sales Agreement Page.

## How to Enable and Use This Feature

To use this rule node first, we need to make sales agreement of that package for that product and jurisdiction.

Create a new rule in design form using `<setpackagestatebyname />`

```
< setpackagestatebyname name="PackageName">
```

```
  <const value="true/false" />
```

```
</ setpackagestatebyname >
```

**Note:** When “setpackagestatebyname” is set to true. By default we exclude any optional forms. You can add any optional forms by selecting “Manage Optional Forms” in the “Other Actions” menu.

## Areas Impacted

System Area	Yes	Comment
<b>Admin Tool</b>		
- <b>Form Library</b>		
- <b>Design Forms</b>	Yes	There is not direct change in Design. New rule node will need to be added to the system to set form's "isExcluded" property to add forms of a package in application at run time. Most of the changes will be implemented in Shared projects.
- <b>Profile Administration</b>	Yes	SalesAgreement. New Package duty Added.
- <b>Reports</b>		
- <b>Deployment</b>	Yes	SalesAgreementsPackages need to account for PackageDuty.
<b>FireLight App</b>		
- <b>New Application</b>	Yes	Only Retrieve Require Forms when
- <b>Edit Application</b>	Yes	We need add and remove packages in the editapplication model, and re register items to the trigger and
- <b>Signature Process</b>		
- <b>Review Queue</b>	Yes	Get Sales Agreement could impact review queue.
- <b>Manual Review</b>		
- <b>User Preferences</b>		
- <b>Inbound Integration</b>	Yes	Batch and any service creating/editing application from TXLife (i.e. NewBusinessService) will be affected.
- <b>Outbound Integration</b>	Yes	Should only pull forms from packages included in the app, instead of all packages from sales agreement.
- <b>PDF Generation</b>		
- <b>Email System</b>		

<b>FireLight Console</b>	Yes	Rules Triggers need to be updated and New Variables need to be created to store Database packages offline.
- <b>Windows</b>		
- <b>iOS</b>		
<b>Other Systems</b>		
- <b>DTCC Integration</b>		
- <b>Commission Netting</b>		
- <b>Activity Reporting</b>		