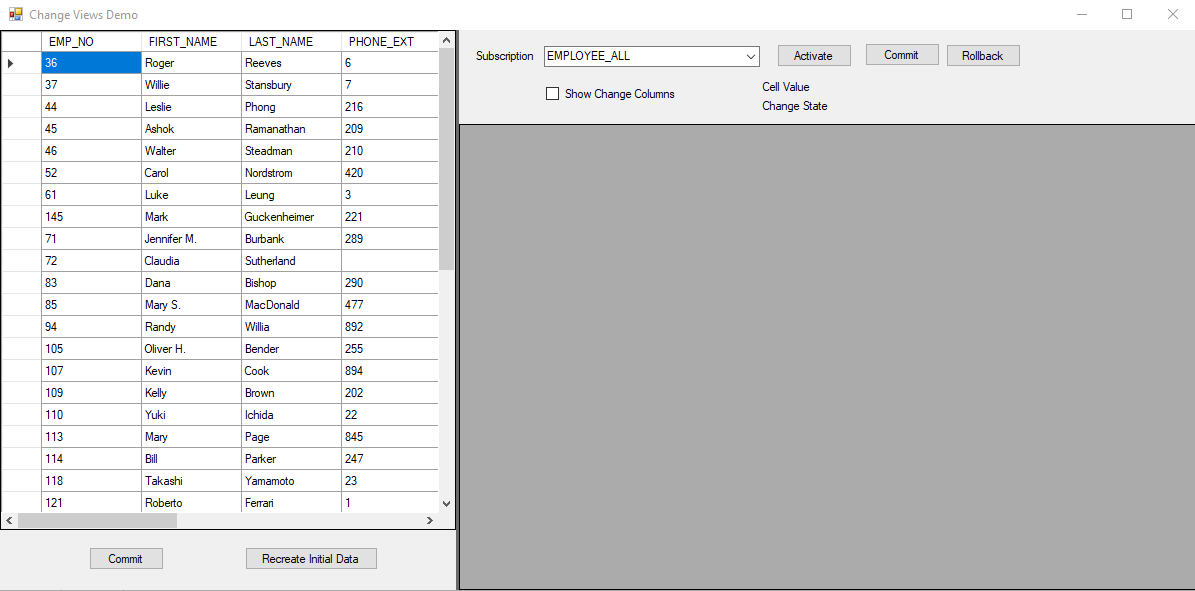
# ChangeView Demo Readme

This is just a quick overview of the change view demo. Change views require InterBase XE7 or higher. The full documentation on the new classes and methods introduced for change views are documented in the main documentation.

Upon startup it will check for the existence of the Demo database which is placed in the same location as the executable. If it isn’t there it will create the DB, build all the structures needed including the subscriptions, and finally populate the data. After priming the subscriptions, a few insert updates and deletes are made.

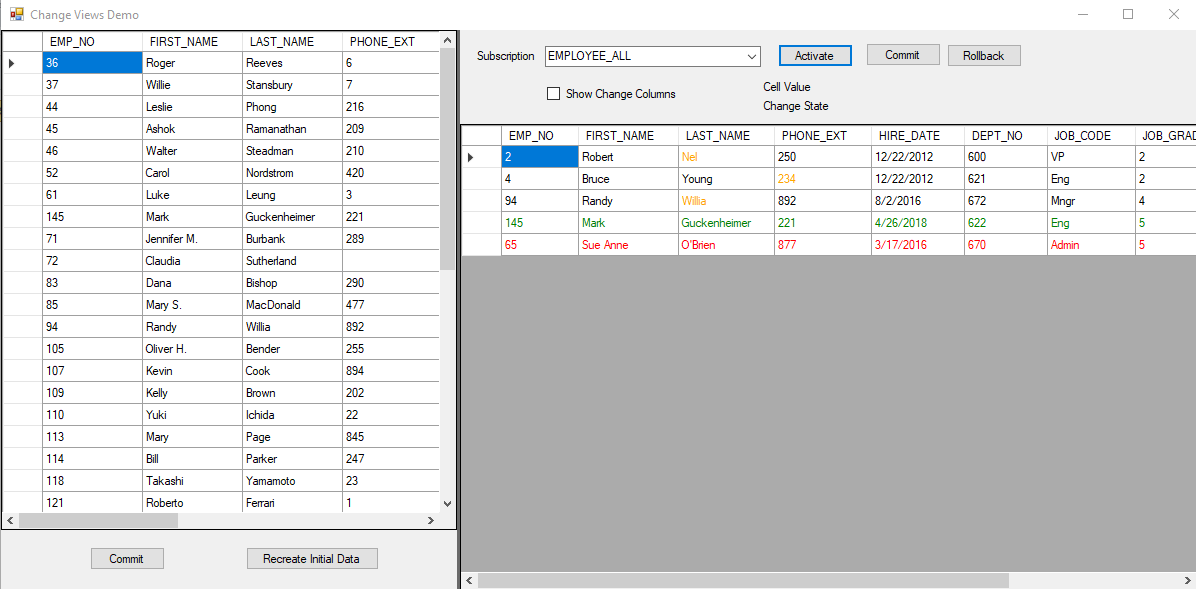


The left hand side is the full select \*. From there you can make changes, insert new records or delete records. You need to commit any changes before you will see them on the right hand side. Recreate data will empty the table and then do the populate/prime/changes sequence as if the DB was being created from scratch.

The right hand side is the change view. Note that both sides use the same IBConnection component, they are run in separate transactions. ChangeViews are always done in a snapshot connection. The drop down combobox lists all the Subscriptions. Activate button activates that subscription and will populate the grid underneath.

The activate button looks similar to a normal filling out a Datagrid except for two things. For the DataAdater you are going to create a IBDataAdapter. Then instead of calling Fill on it passing the table, we call FillWithChangeState. IBDataAdapter has a corresponding FillWithChangeState to match the signatures of the normal Fill.





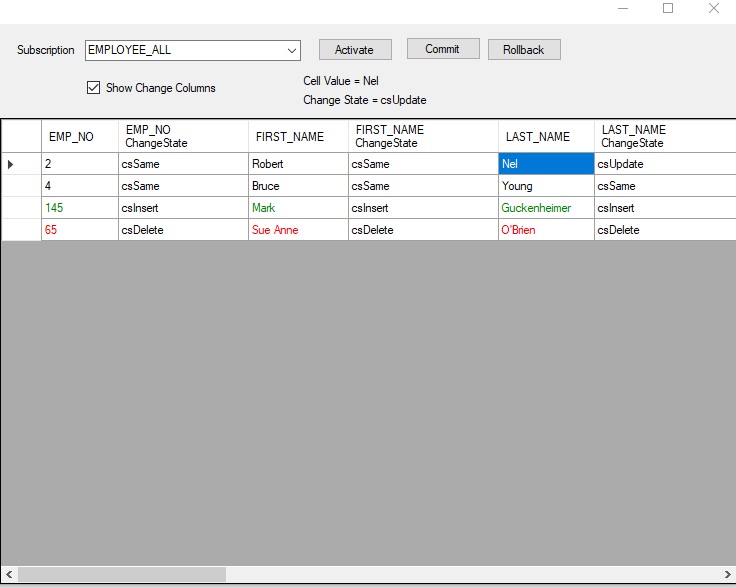
The color coding is done like this



To get at the change state you look at the underlying IBTable that is the grids datasource. IBDataRow has a method called ChangeState that accepts the column index you want and returns the IBChangeState for that column.

Commit and Rollback just commit or rollback the snapshot transaction.

Final couple of things in the demo is the check box for show change columns and then 2 labels to show the selected cell and its change state.



The check box fires code that calls the new IBTable method ChangeStateColumns. When true it will add a column to the rows for each real column called “<realname> ChangeState” , when false (the default) it deletes those rows.

The labels are fairly straight forward. Once again we work from the underlying IBTable/IBDataRow level to get at the change state.

