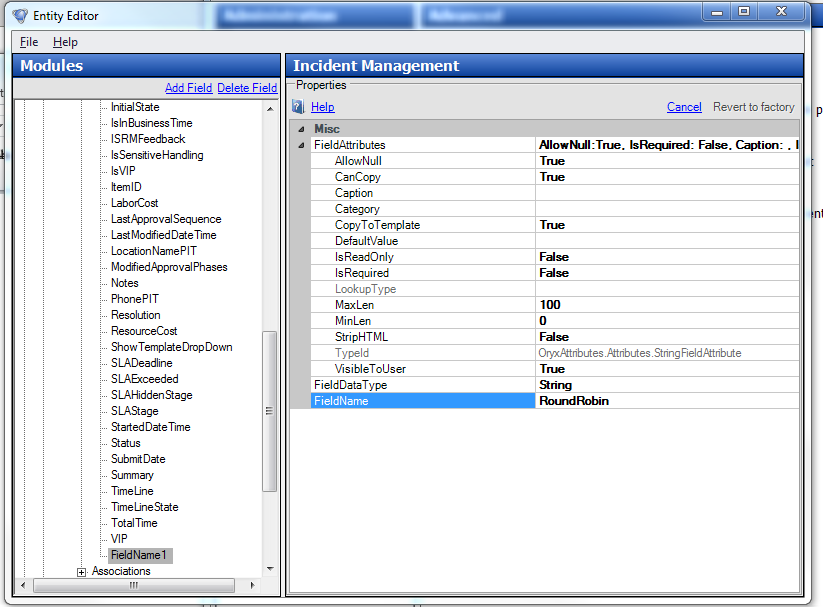
Round Robin Auto Assignment

About

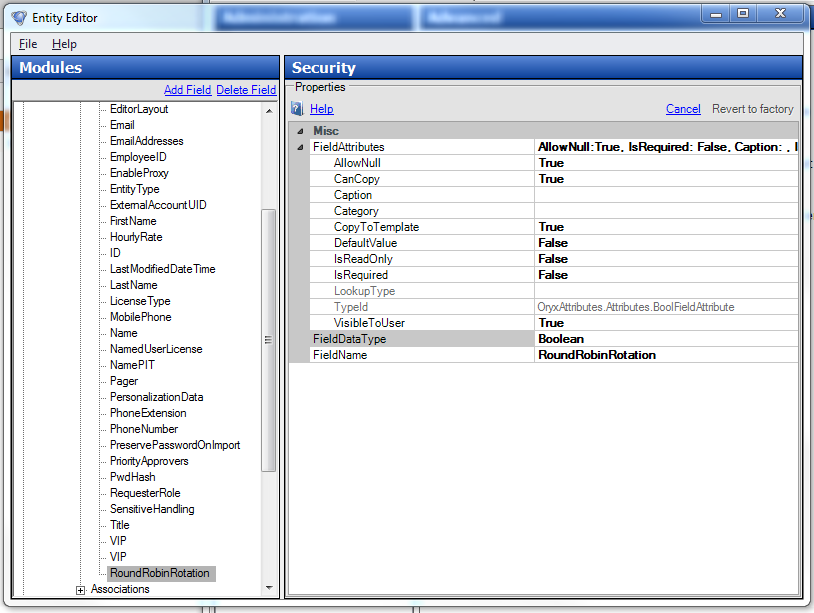
1. This is a solution for integrating a round robin system into ChangeGear. It consists of the following:
   1. 2 custom UDF fields - one string and one a boolean. The string field could eventually be converted into a boolean field if need be.
   2. Run Code (see round\_robin.cs)
   3. A stored procedure (see round\_robin.sql)
   4. 3-4 users should be used for testing this. They should have the RoundRobinRotation UDF field checked for their user profile in order to be assigned a ticket.
2. Two customizations could be removed from this solution which are…
   1. The round robin system will only assign tickets to users who are in the team that is passed to the stored procedure. The team name can be specified in the run code for this solution, but the team used in this package is called “Round Robin Team”
   2. By default, the ticket will be assigned to the requester **if the requester is part of the team that is passed to the stored procedure.**
3. This has been tested in versions 6.0.6466 and 6.0.6478 of ChangeGear

Implementation

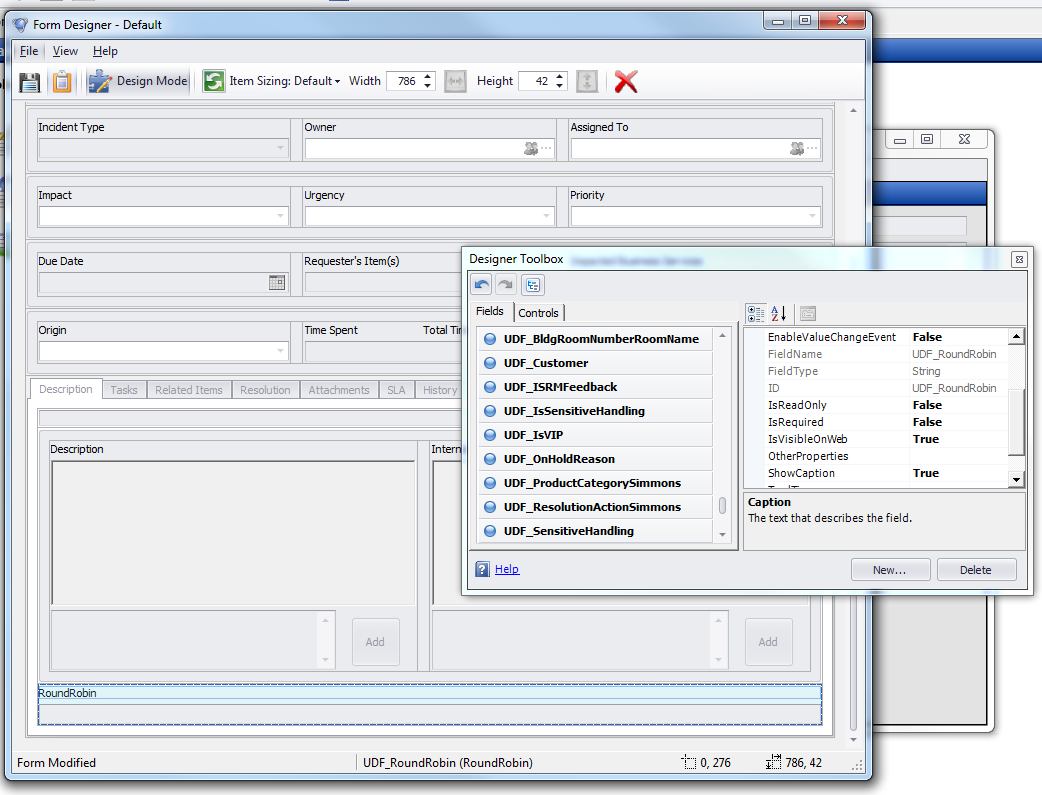
1. Go to the Entity Editor in ChangeGear
2. Under Incident Management > Incident Request > Model > Fields > Create a **String** field called “RoundRobin” (without the quotes). See screenshot below for more details.



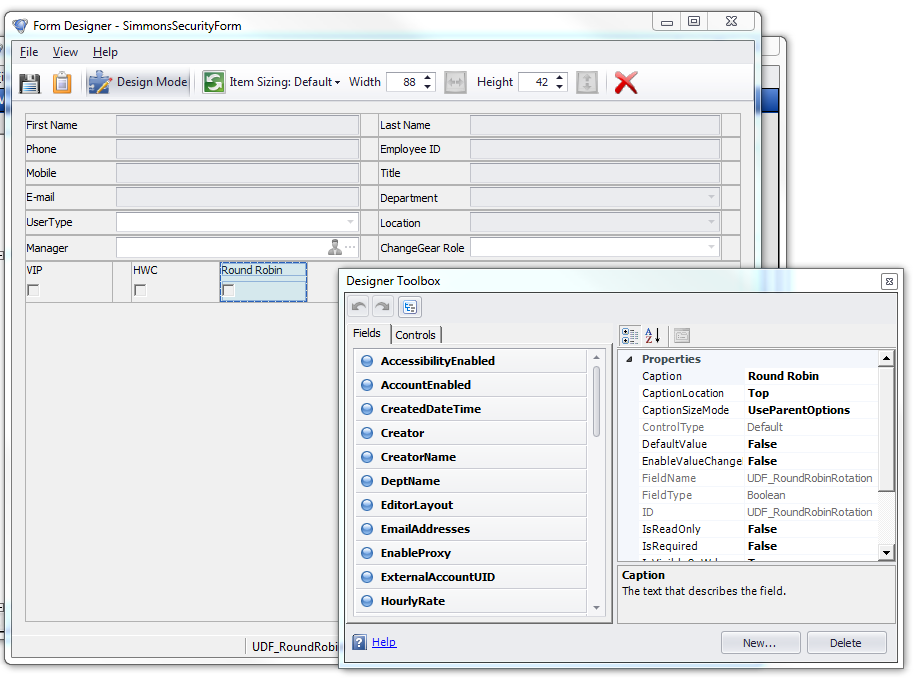
1. Under Security > Model > Fields > Create Boolean field called “RoundRobinRotation” (without the quotes). See screenshot below for more details



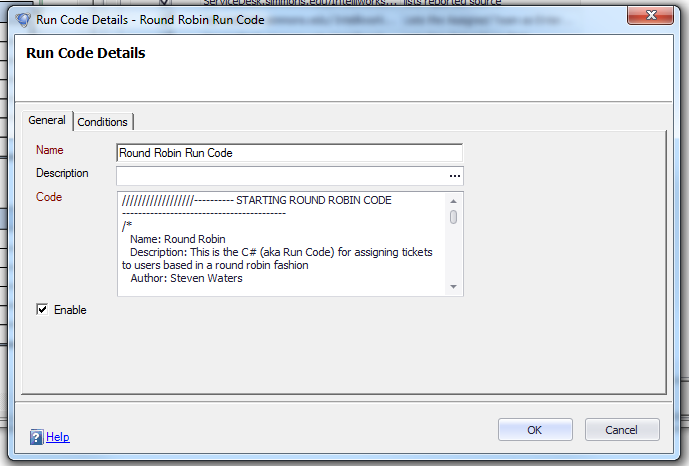
1. Add the string field that was just created to the active End User and Staff forms for the Incident Management (Incident Request) module.
   1. Set the **IsVisibleOnWeb** property for this field to **true** for now.
   2. You can keep the other properties default for this field
   3. This field doesn’t need a user friendly caption because it won’t be shown to the user in the long run.



1. Add the Boolean field that was just created to your active security forms.
   1. This field can go below the main field.
   2. Give it a user friendly caption such as “Round Robin” or “Round Robin Rotation”
   3. This field will should be visible to your users so they can opt out of the round robin rotation system



1. In SQL management studio, Connect to the ChangeGear database and start a new query for
2. Copy in the contents of the round\_robin.sql file into the new query window
3. Change the part of the query that says “USE [ChangeGear]” (which should be on line 7 as of this writing) by changing ChangeGear to the name of your ChangeGear database
4. Execute the query by hitting F5 (or by pressing on the execute button)
   1. This will create a stored procedure called “**GetNextAssignee**“
5. Enable Run Code automations if they are not enabled already
6. Go to the active Incident workflow
7. Click on the Submit action and add a new Run Code automation to it
8. Give the Run Code automation a title and description
9. Open up the provided round\_robin.cs file
10. Copy the contents of this file (along with any changes) into the code field for the Run Code automation



**Note:** The below line can be changed so it references a different team instead of “Round Robin Team” :

*query.Append(string.Format(@"EXEC GetNextAssignee @Team='Round Robin Team', @RequesterID=" + currentRequester));*

1. Save the Run Code automation and save your changes to the workflow
2. **Restart the ChangeGear service** and then reset IIS
   1. Restarting ChangeGear is necessary so the UDF fields can be written to
3. After the ChangeGear service has been restarted and ChangeGear is back up, create a new Incident ticket (do not submit it yet)
4. After filling out the required fields and before submitting it, look for the RoundRobin UDF field that was added to the Incident Request forms earlier
5. In this field, type in “yes” in the RoundRobin UDF field
6. Submit the ticket
7. Add this point, you can hide the RoundRobin UDF field for each of your active forms by setting it’s **IsVisibleOnWeb** to false

Testing

1. Create 3-4 dummy users or choose 3-4 users for testing purposes
2. Create a team called “Round Robin Team” (without the quotes)
3. Add these users to the team that was just created. In this case, it should be “Round Robin Team”
4. Now check the RoundRobinRotation UDF field for these users
   1. You can leave this field uncheck for one of these users or create or choose an additional user who shouldn’t be in the round robin rotation
5. Now log these users in. Running the below query will show you the OID of the users who will be picked for ticket assignment:
   1. “Round Robin Rotation Team” should be replaced with the name of the team that will be used in the Round Robin Rotation system

SELECT aa.Person AS OID,

Row\_number()

OVER (

ORDER BY cc.LoggedIn DESC) AS RowN, cc.LoggedIn

FROM VCG\_PersonToTeam\_Grid\_View aa

JOIN CG\_Person bb

ON aa.Person = bb.OID

JOIN #temp1 cc

ON aa.Person = cc.UserID

WHERE bb.UDF\_RoundRobinRotation = 1

AND aa.Teamname = 'Round Robin Rotation Team'

AND aa.GCRecord IS NULL

GROUP BY aa.Person,

cc.LoggedIn

1. Now submit several incident tickets when the AssignTo field is blank
2. At this point, you should see users being assigned a ticket automatically. Each user should be assigned tickets one by one in a round robin fashion.

Troubleshooting

If not tickets have been assigned to a user by the round robin system, then the following could have gone wrong:

1. Check that users are signed and are part of the round robin rotation by running the query referenced above
2. Make sure users have the RoundRobinRotation UDF field checked for them
3. Make sure the users are part of the team that is specified in the run code and in the SQL script (which should only be hard coded in comments)
4. Make sure at one incident record has a value of “yes” written into the RoundRobin UDF field. This can be done by doing a select from the CG\_IncidentRequest table.
5. Check the spellings of the UDF field references in the run code and the store procedure and confirm that they match what was created in the Entity Editor
6. Make sure the RoundRobinRotation field is a Boolean type field and that the RoundRobin field is a string field
7. Check the log files for compiling errors in case there’s a typo in the run code. Checking the log files will tell you if there’s a syntax error.
8. Note the log file will contain several info lines for debugging purposes
9. Restart the ChangeGear service if you’re not seeing “yes” written into the RoundRobin UDF field. The ChangeGear service needs to be restarted at least once after adding these UDF fields to the forms.
10. The stored procedure /SQL script behind this solution contains commented out SELECT statements. Run those to see what data is being used for the round robin ticket assignment
11. Running this line in the SQL statement will show you what is being returned to the run code**: exec GetNextAssignee @Team='Round Robin Team', @RequesterID=1**
    1. If it’s not returning a numeric value then that means the run code can’t assign a user to the ticket