# Adamma V2 description

Adamma v2 is a programs mainly used to track the SLA (number, rate) for each team. Analysis the On hand bugs, work load, productivities between teams and team members under any period.

## Functionalities guidance in Adamma2

### Common functionalities

1. Query the bugs from TFS with customized team. Access the details data for each bugs with the customized fields you set here.
2. Query the bugs from TFS with imported queries.
3. Query the bugs for the several teams for an overviews of teams SLA.
4. Analysis the Dev & Test productivities for each team with different filter.
5. Query the bug from TFS, Product studio (Supported database: DAXSE, AXSE, AX6) through bug id.

### Functionalities according to your Role:

If you are a **Manager**, you are able to

1. Query the monthly resolved bugs (number, rate) for any team. Setup your query like [this](#_Query_monthly_(any).
2. Track all the active SLA breakers (number, rate) for any team. Follow the setups [here](#_Track_all_the).
3. Compare the total active/Resolved/Any status bugs between the teams you customized [here](#_Setup_your_Mode).
4. Easy to review each guys productivities during any period for any purpose.(e.g. PE review) Setup your query like [this](#_Query_the_total).

If you are a **Dev\Test Team Leader**, you are able to

1. Track new in coming bugs like [this](#_Query_new_in).
2. Track SLA For each bug with any required status in your team like [this](#_Track_SLA_For).
3. Track all the active SLA breaker in your team. Follow the setups [here](#_Track_all_the).
4. Analysis the work load between team members. You need setup your filter like [this](#_We_can_also).
5. Easy to review each members productivities during any period for any purpose.(e.g. PE review) Setup your query like [this](#_Query_the_total).

If you are a **Dev\Test**, you are able to

1. Track each bugs status & SLA which bugs assigned to you. Be aware of the SLA threat! Look at [this](#_Track_the_dangerous).
2. Review yours productivities during this year, month, and week like [this](#_If_you_want).
3. Easy to access the details data in your bugs in Adamma. Customize you fields you want to track like [this](#_Custom_your_required).
4. Easy to query the bug from TFS, Product studio(Supported database: DAXSE, AXSE, AX6) through bug id. Use “Control + G” or “F2”

If you are a **PM**, you are able to

1. You are able to track each state active bugs which PM fields assigned to you. Setup your query like [this](#_Track_all_my).

If you are a **Code/Test spec reviewer**, you are able to

1. Build a list for the bugs waiting for code review. Setup review priority for them.
2. Easy to access the Code flow/Review bugs.

If you are **TAE member/EE or bug hunter**, you are able to

1. Track all the bug status opened by you. You build a query like [this](#_Track_the_active).
2. Easy to access the details data in your bugs in Adamma. Customize you fields you want to track like [this](#_Custom_your_required).

## Asynchronous mode

Adamma provided three asynchronous mode,

1. Asynchronous the Team/ Query data.
2. Asynchronous the chart groups for teams.
3. Asynchronous signal bugs details queried below the grid view. You can turn off this Asynchronous through Adamma -> Setup -> Adamma Setting -> Pre-Load signal bugs fields. Thus, such data for each correspondent bug in grid view will be queried together with the Asynchronous mode one. It’s strongly recommend that turn on this Asynchronous mode when the results in target data are excessive. Or it will cause a performance loss with the Asynchronous mode one.

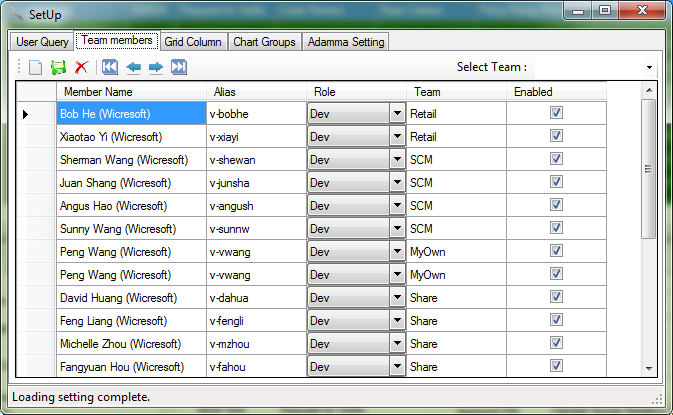
# Start to use your Adamma

## Create your team

User can create the Team from Adamma -> Setup ->Team members.

To each fields in the grid,

1. Member Name: User must input the member’s name with the same name in the TFS like below,
2. Alias: This field is optional for extend, since I current can’t get a access to read the database which link the name & alias.
3. Role: Include the three Enumerable value : Dev, Test, PM. Adamma will query the bugs due to the member and their role.
4. Team: Case must be specify here.

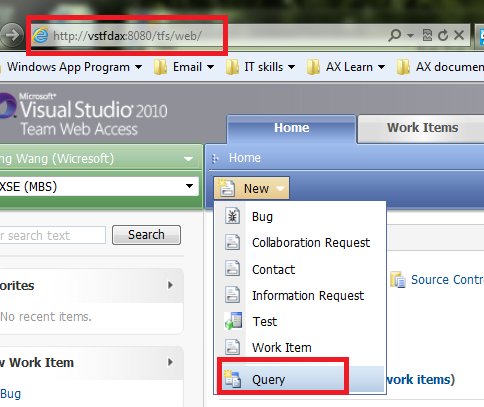


## Add your own queries in Adamma

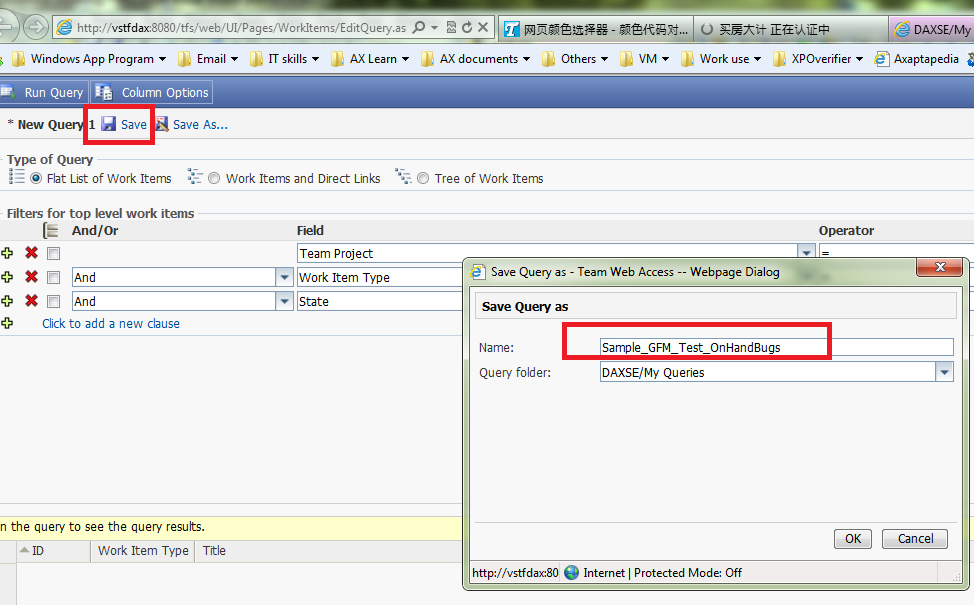
### How to create/Save your query in TFS

Firstly, you need to create your own query like below,

For some complex/customized query which due to personal use. You can create you query at <http://vstfdax:8080/tfs/web/>

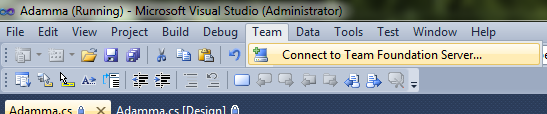


Add specific conditions, and save your query with name e.g. “Sample\_GFM\_Test\_OnHandBugs”. For the help document how to create your own query please refer <http://msdn.microsoft.com/library/ee523998.aspx#Queries>

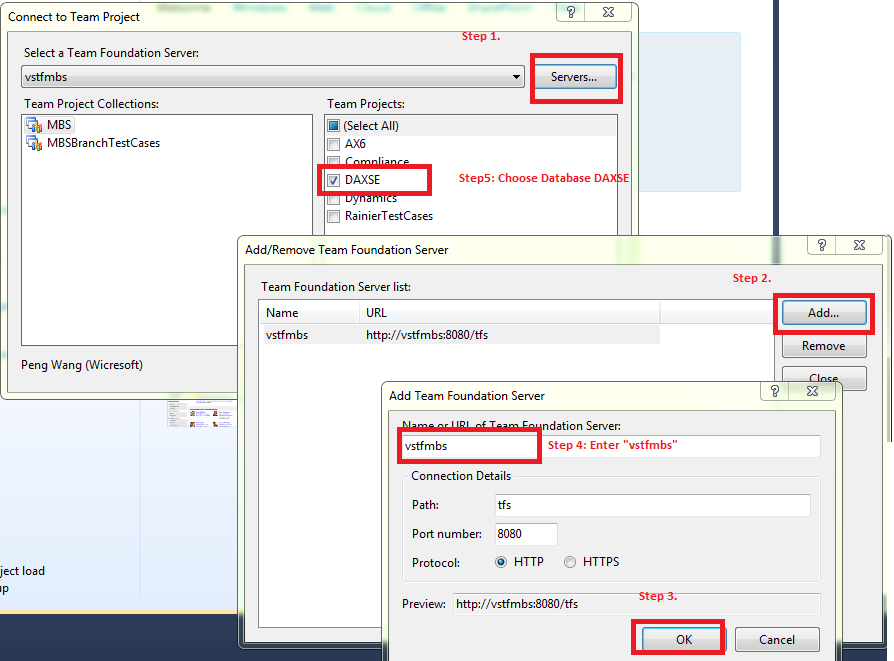


**Secondly, you need to save your created query with visual studio.**

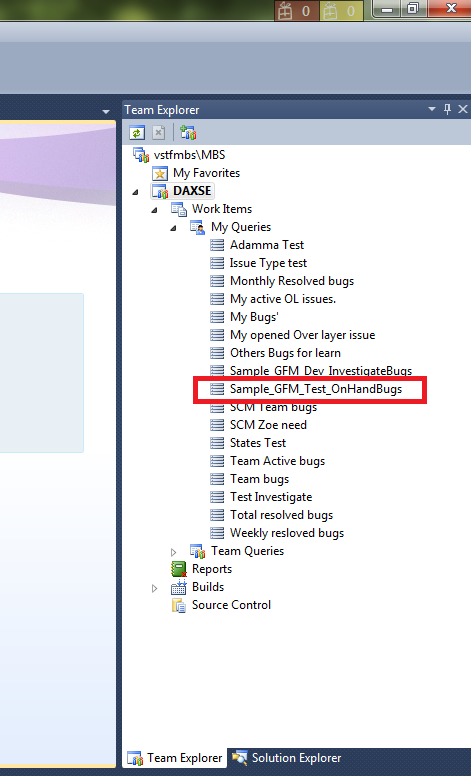
Connect the TFS through “Team-> Connect to Team foundation server”



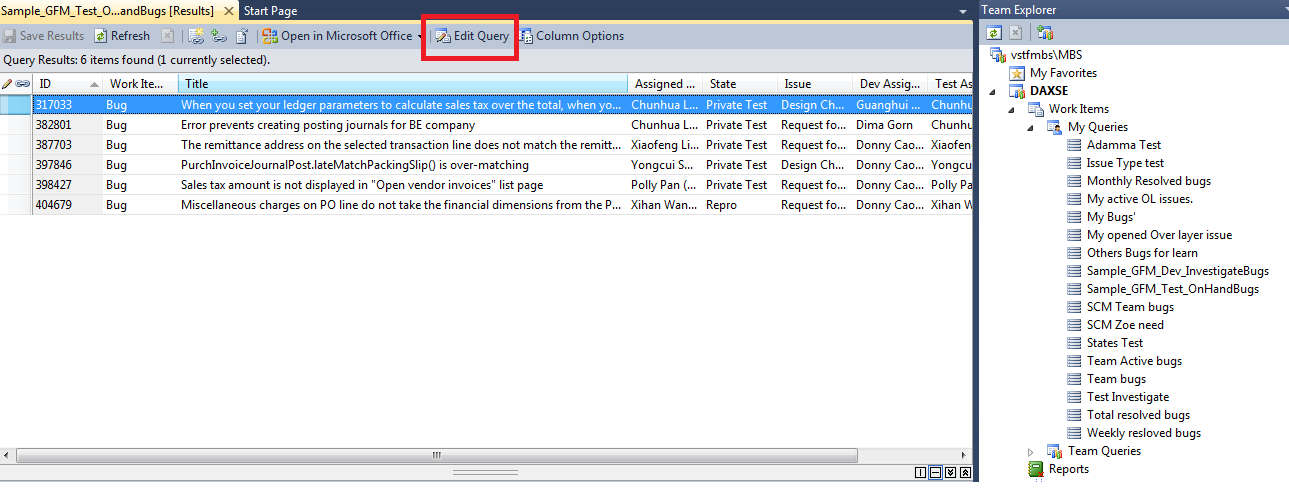
Following the four steps in the bellow picture,



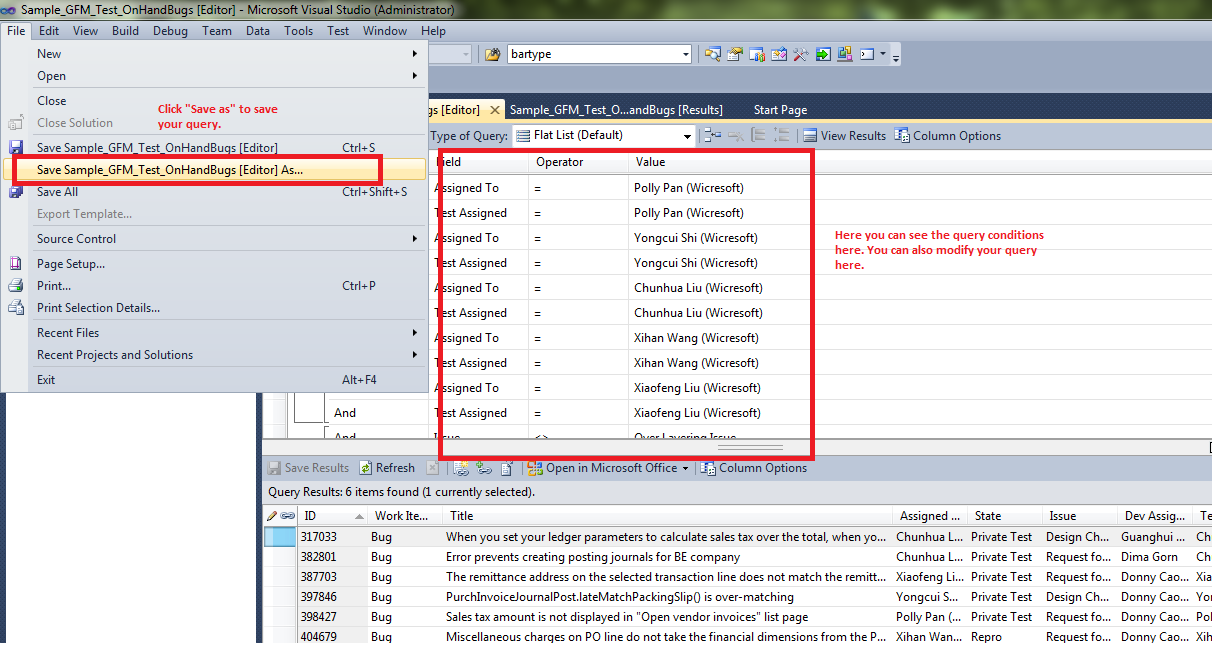
Now, in the left of visual studio. You can see the items from the tfs server you just connected.



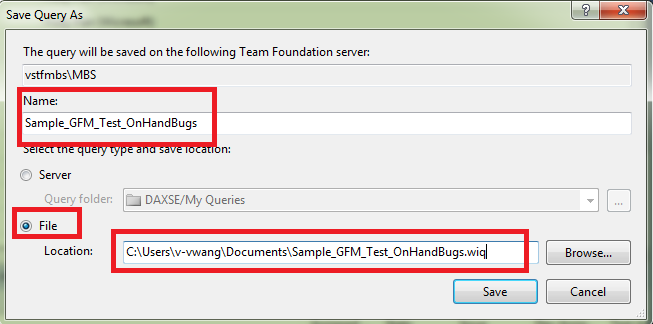
Double click this node. Click edit query, you are able to see the query editor now.



Now click “Save as” in File -> save “…”as,

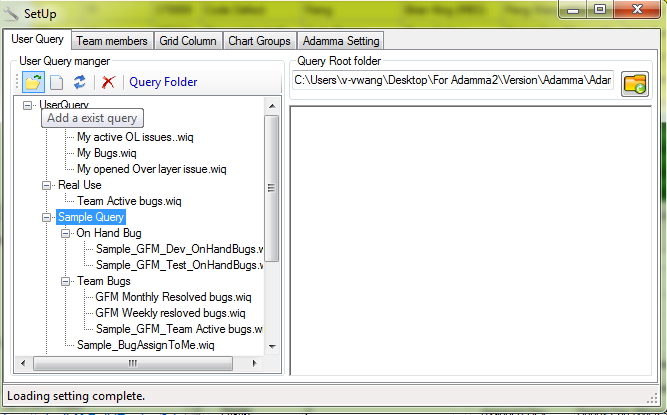


You can save the query as “\*.wiq” file now.



### Import your \*.wiq query to Adamma

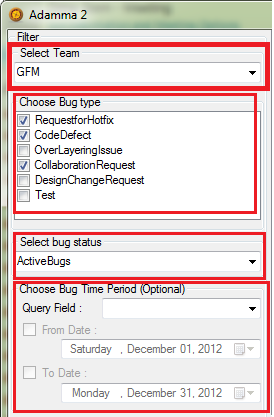
In Adamma, opened setup -> Use Query, select the base folder you want to import your folder. For example we want



## Filter in Adamma

Adamma 2, provided two kinds of filter. One for the query before filter which define the sql used to query the result for gridview/Chart team group. To he

### Filter before query

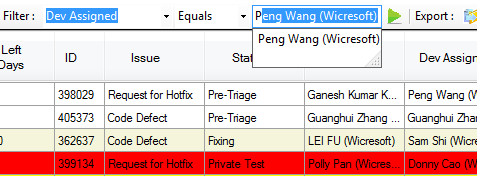


The filter in Adamma control the expression sql generated for quest data from TFS.

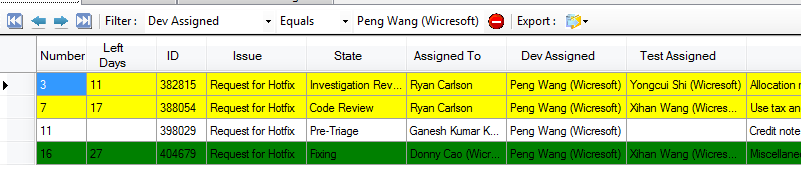
1. **Team**, this could get all the **enabled** team members from one team.
2. **BugType**, current suppotted bug type here. RFH, CD, OL, CR, DCR, Test issue.
3. **BugStatus**, here contains all the each basic status of bug. Like “triage , repro, investigate, fixing, codereview, etc..” And there also provided two types of composite bugs status, they are:
   1. **Active bugs**: The bug status are not closed or resolved.
   2. **ResolvedColsedBug**: The bug status are in closed or resolved
4. **Bug Time period**, this filter provided each date query for each fields with typeof “DateTime”. The fields like : CreatedDate, ETADate, ResolvedDate, etc.

Note, if you leave empty to team, nothing will be fetched, if you leave empty to other fields like “Bug type”, “Bug Status”, “Time period”, which means all the bug with different type, status, Time will be in considered.

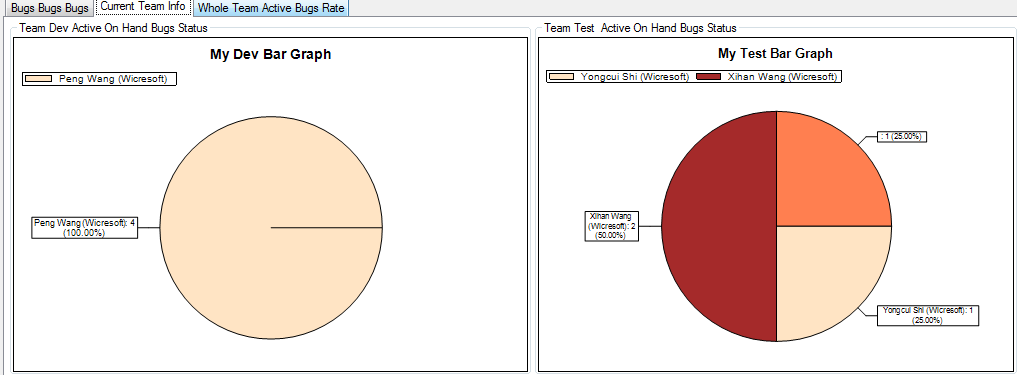
### Filter after query



After query from TFS to grid view. User also can use the filter which above to the grid to do some extended query. E.g “Dev Assigned Equals Peng Wang (Wicresoft)”, the results will like below,

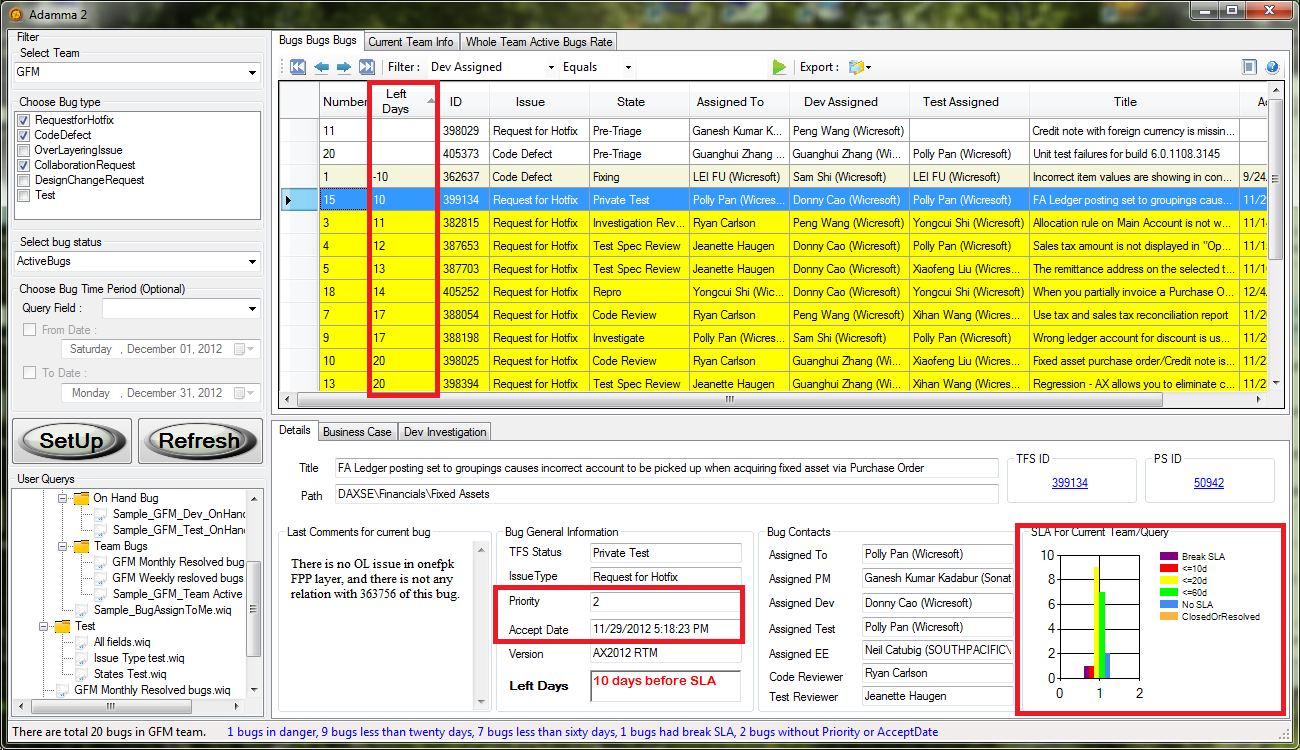


The Pie of DEV & Test will update accordingly.

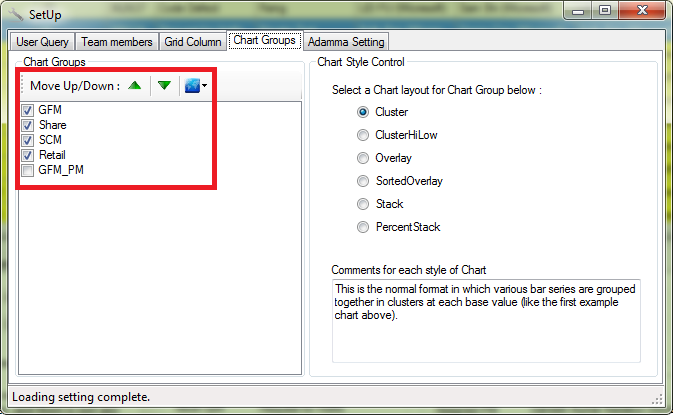


## Team(s) SLA track

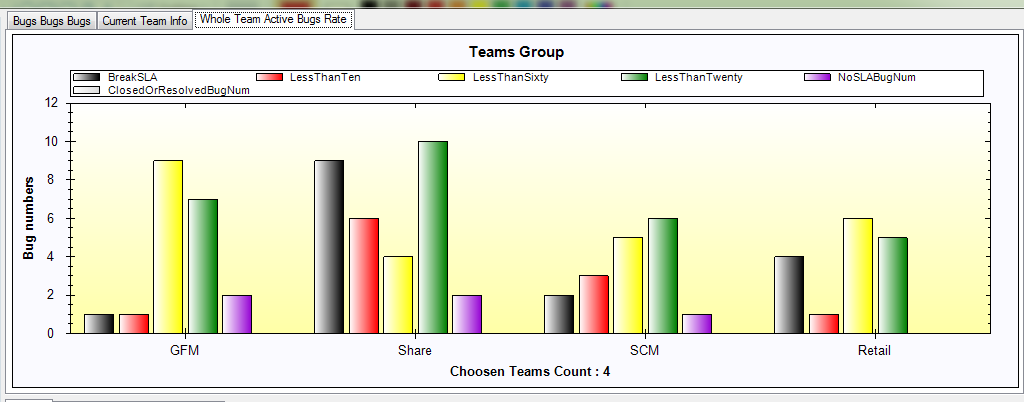
You can track SLA for each bugs in current team, the each status rate of bugs in current team. Results like below,



And Adamma also enables you track the overview of SLA rate for any teams you defined in Setup->Chart Groups.



The result for team groups chart will be like,



Note: The filter for chart groups is from the [filter before query](#_Filter_before_query).

# Customize your Adamma

With the customized setup you can get a better BI results as you want from the main windows of Adamma.

## Manage your query

First, you need to import your customized query. Please refer the [Import your \*.wiq query to Adamma](#_Import_your_*.wiq). You can create your folder to manage your different query for each purpose. The query tree list is totally matched with the physical location. You can also place the wiq query fields to the “Query Folder”.

## Best practice to manage your team members

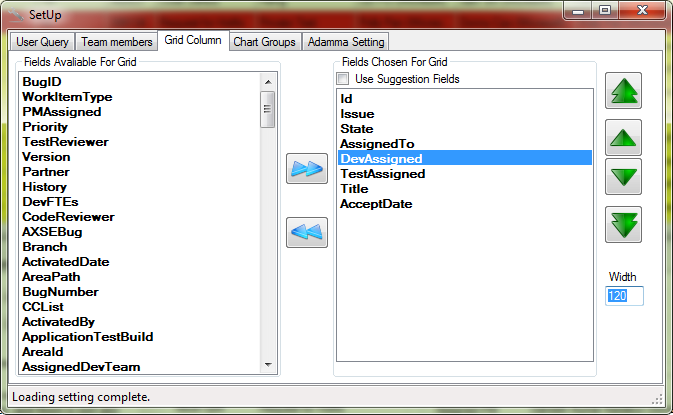
Here are some best practices to manage your team members use Adamma. For the details of how to create your team please refer [here](#_Create_your_team).

1. Disable the Team members which are unused for current team now. For example if someone transfer to other teams. When you query the active bugs, this guy’s result should not contain. If you want to see the input (s)he does. You can enabled his(her) account.
2. If you are a Dev/Test leader, you may only want to create the team with only role of Dev/Test under your manager to avoid dirty data.
3. You can create a team with your PMs,(e.g. GFM PMs) so you can access how many bugs are currently exists in Financial/GDL area.

## Custom your required fields and width in main grid view

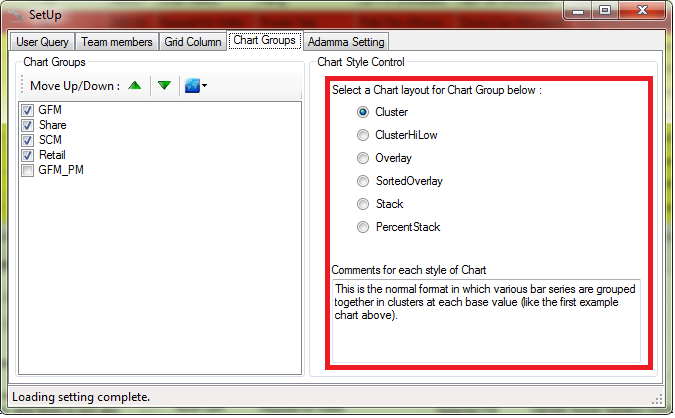
Adamma enables user to customized the fields(width) they want to see in the grid view of main window. 

From the setup below,



## Setup your Mode for chart group for SLA overview between Teams

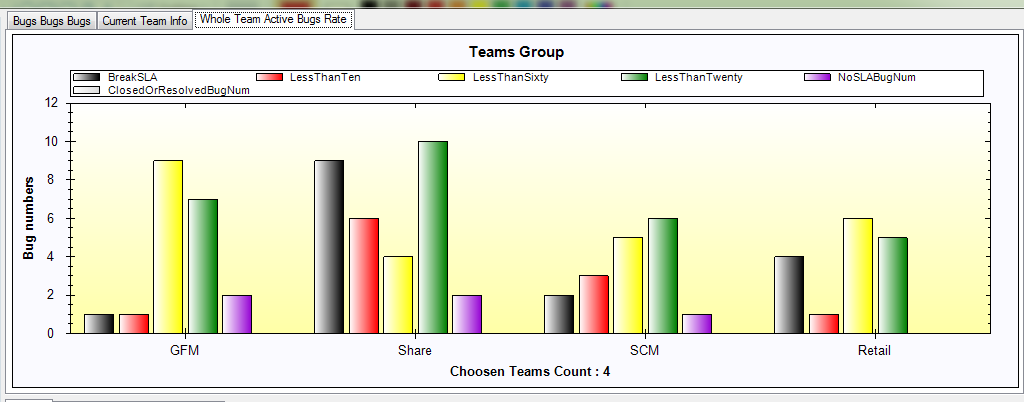
For how to track SLA between teams please refer [here](#_Team(s)_SLA_track). You can also customized the mode to display the target data with different mode defined below,



With the different mode, you are able to see the result like this.

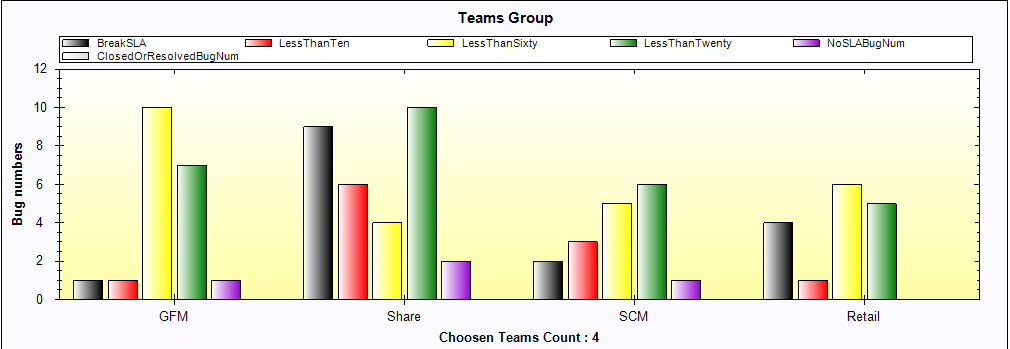
Cluster,

This is the normal format in which various bar series are grouped together in clusters at each base value.



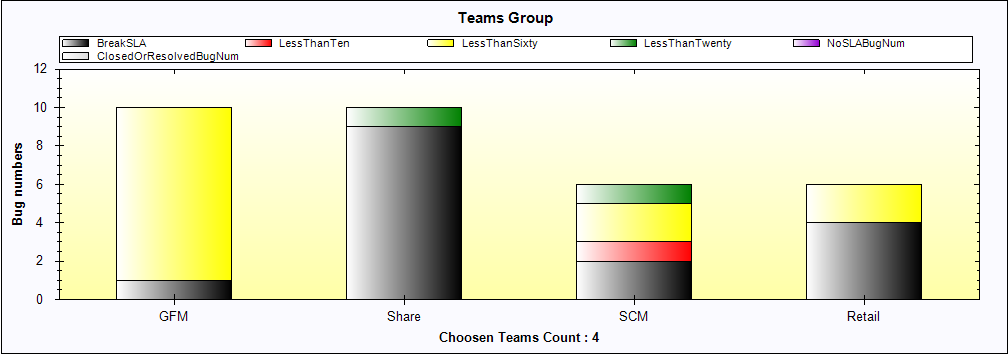
ClusteHiLow,

This format draws a hi-low (bars have a top and bottom that are user defined) in a cluster format, so multiple high-low bars can be grouped together at each base value.



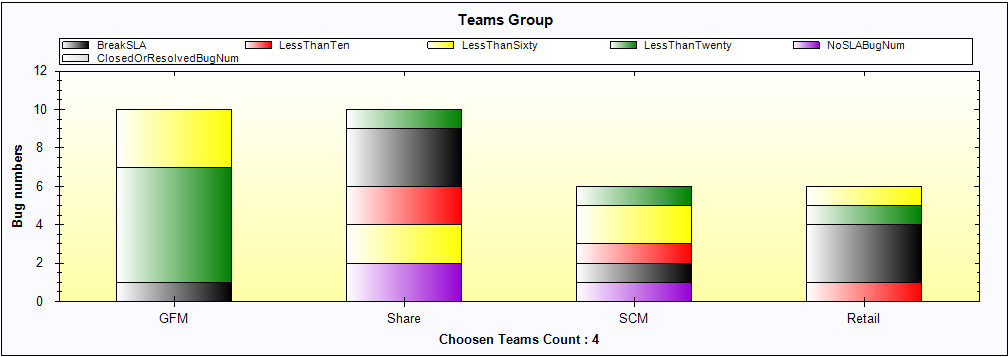
Overlay,

In this format, the bars are drawn on top of each other, with the first BarItem drawn at the back, and the last BarItem drawn at the front.



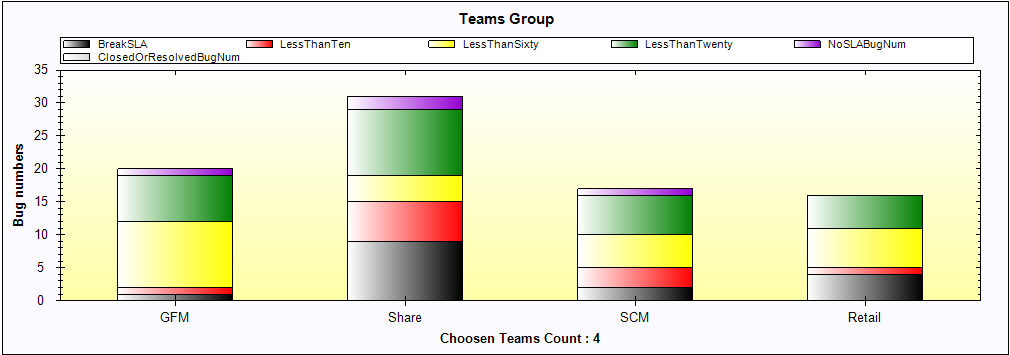
SortedOverlay**,**

This is similar to Overlay, but the bars are sorted on value, and the highest value is drawn at the back, and the lowest value is drawn at the front.



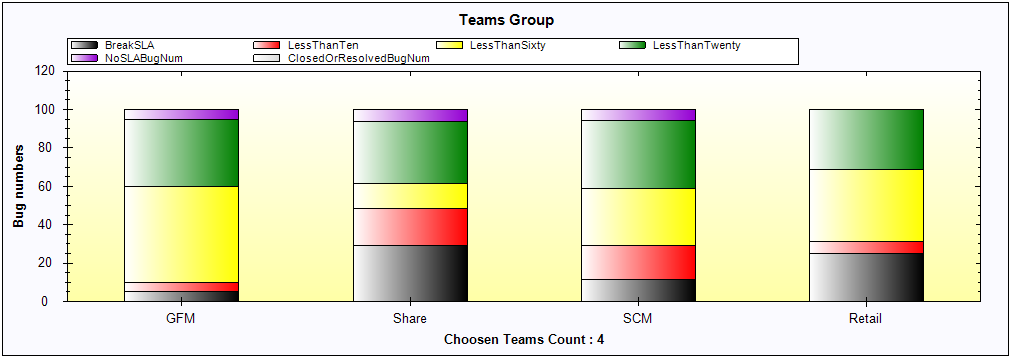
Stack,

The bars are stacked on top of each other, accumulating in value.

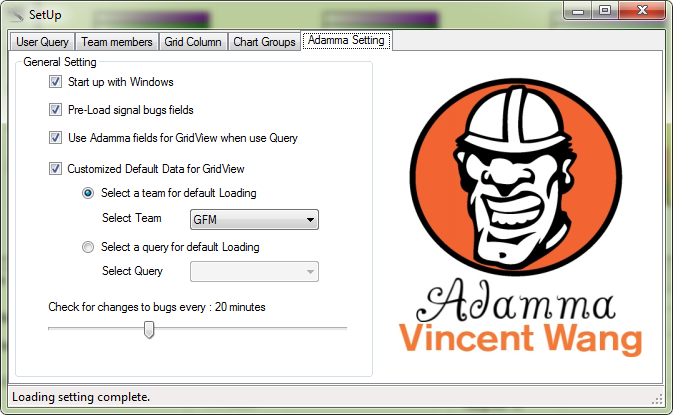


PercentStack,

The bars are stacked on top of each other, and plotted as a percentile, with the total height always being 100%.



## Other Adamma setting

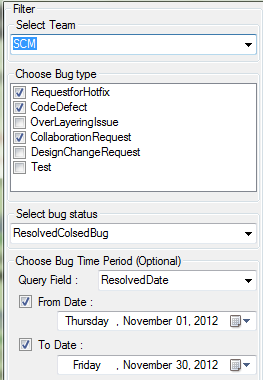


1. Pre-Load signal bugs fields, this is related to asynchronous mode please read the comments from [here](#_Asynchronous_mode).
2. You can choose a default team/Query for load when Adamma start.

# BI in Adamma

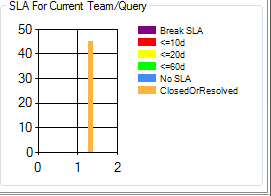
## Query monthly (any period) resolved bugs in your team & between teams

If you want to query the monthly resolved bugs in SCM team in November, you need to build your filter like below,

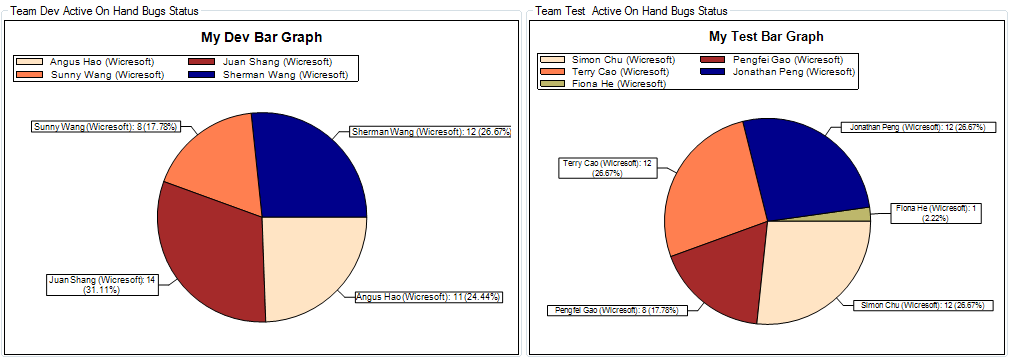


The result for SCM team will like, now we can see there are 45 bugs was resolved by SCM team in November.

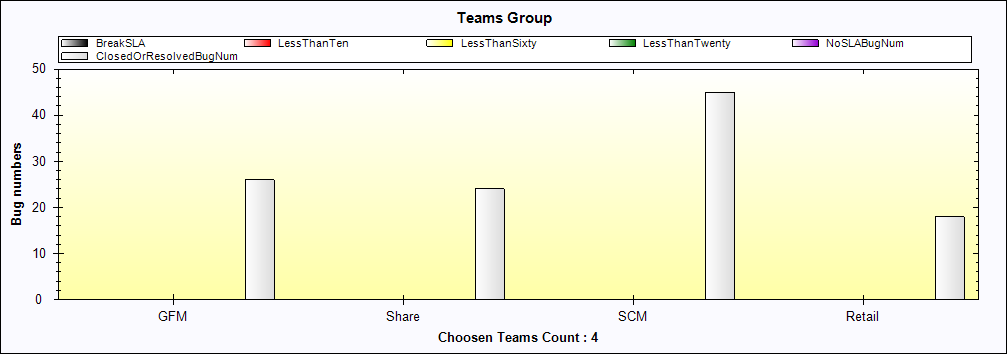




### We can also see the contribute rate from each member (Dev & Test) was here.



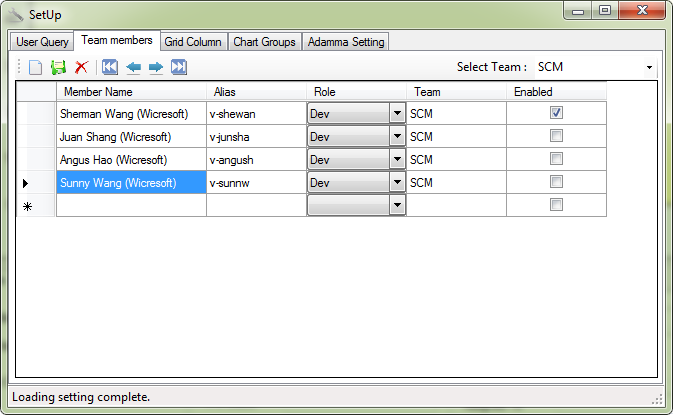
### The monthly resolved bugs between teams will be like,



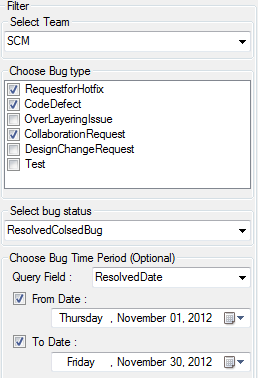
## Query the total resolved bugs (any type) for any team member

### If you want to see any one’s (like Sherman in SCM team) productivity during this year,

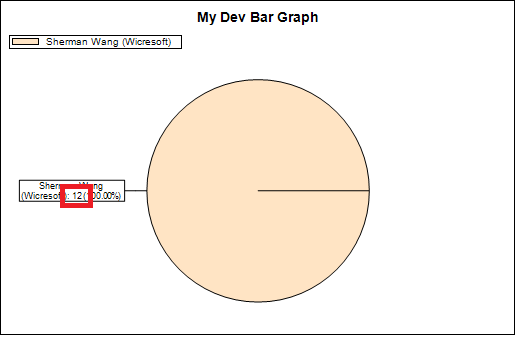
You need first disable other team members,



Build the filter like this,



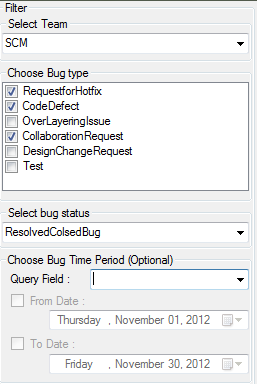
You can see the result here,



And here,



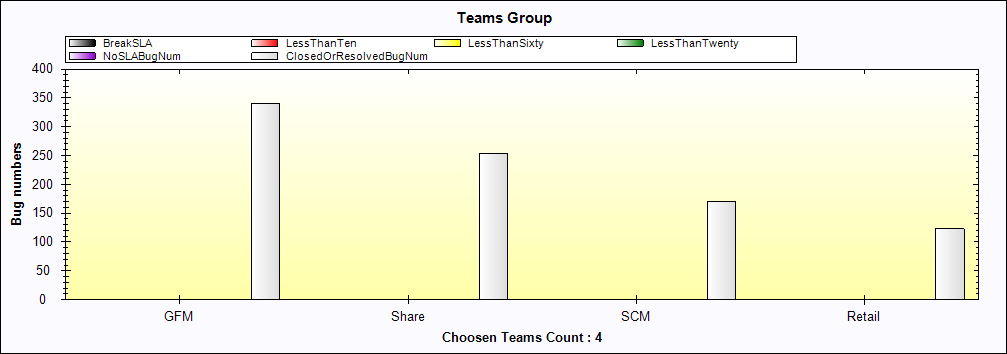
### **If you want to see the totally resolved bugs provided by Sherman**, you can leave the time period to empty,



You will see the result here, 171 was totally resolved by Sherman,

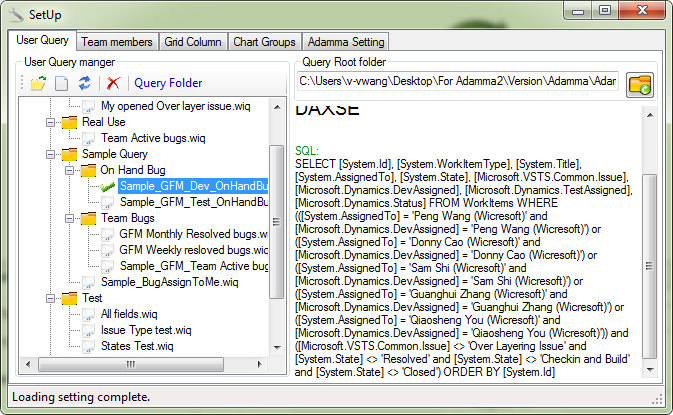


PS: We can also see the Sherman’s productivities are really weird from here….



## Query the Dev or Test on hand working bugs

To this topic, you need to build your customized query like this, please refer the sample “Sample\_GFM\_Dev\_OnHandBugs.wiq” in Adamma



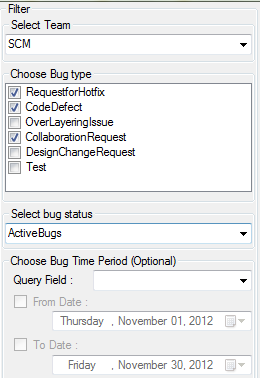
You will get the current working bugs in GFM team as below from this query,



For how to create query and import to Adamma, please refer [here](#_Import_your_*.wiq).

## Track all the active SLA breakers (number, rate) for any team

Build your query like this,



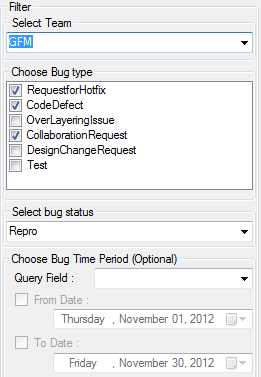
Add your advance filter,



For the numbers and rate, please refer the “Dev Bar Graph”

## Query new in coming bugs

You can easily track your team’s bugs incoming like this,



## Track the dangerous bugs which near SLA

You can track your team’s dangerous bugs like this,

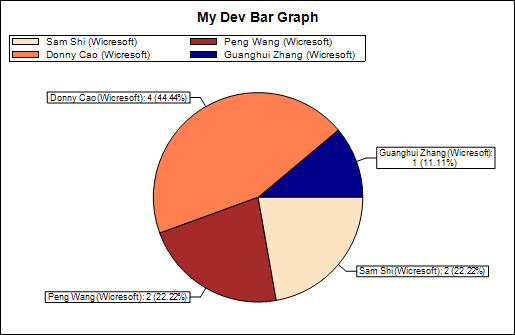


We can see there is one bug with 10 days left.

We can also use the [advance filter](#_Filter_after_query) like this after queried above result,

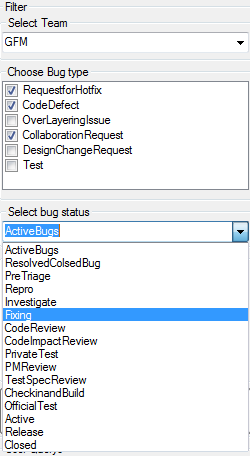


And here is the dangerous rate,



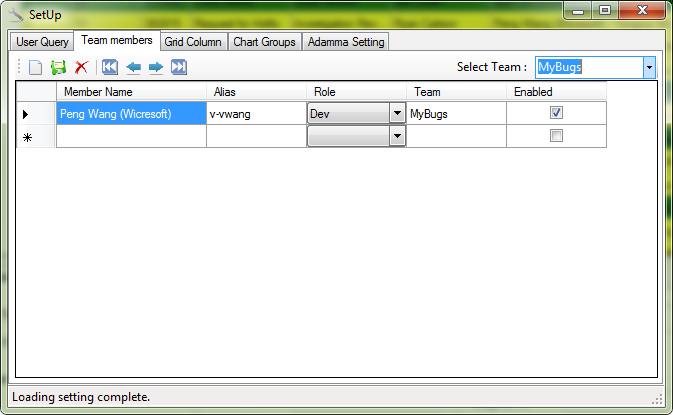
## Track SLA For each bug with any required status in your team

Setup query like this,

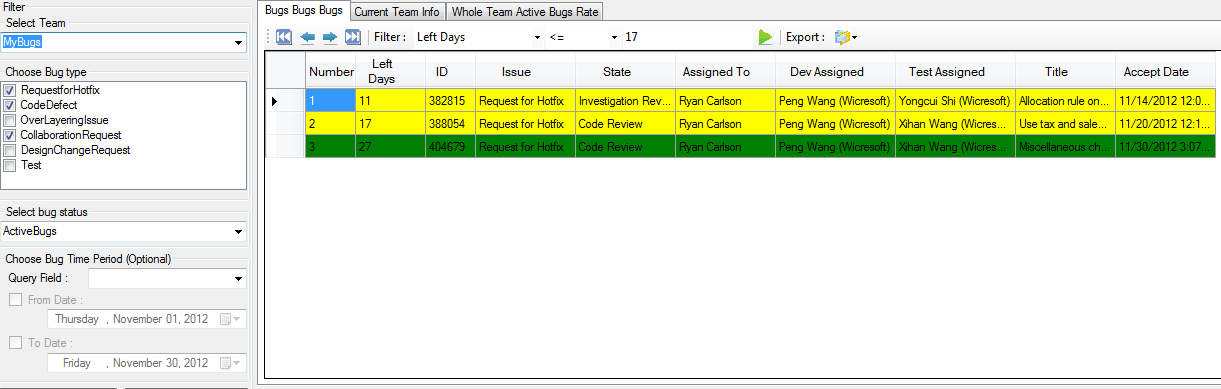


## Track all my current on hand bugs

Here I suggested create a team with your name only first.

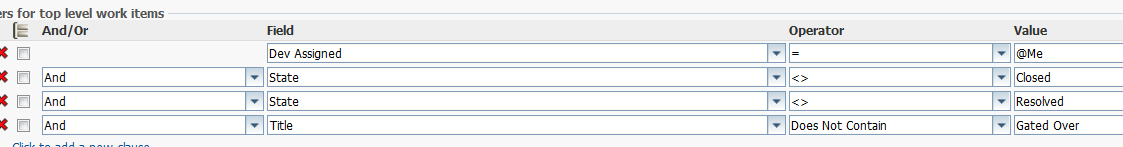


Query all the active bugs will get your all on hand bugs,

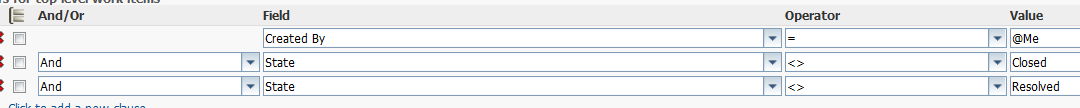


### You can also use the sample query “Sample\_ActiveBugs\_DevAssignedToMe” or “Sample\_ActiveBugs\_TestAssignedToMe”

Query like below,



## Track the active bugs opened by me

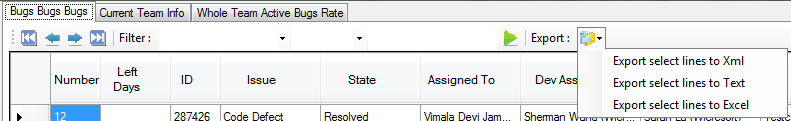


Please refer the Sample query “Sample\_ActiveBugsOpenedByMe”

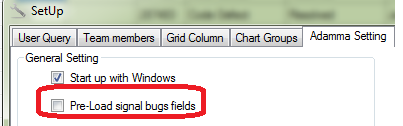
# Others

## Export the data queried from Adamma,

Adamma supports the three styles(txt, excel, xml) to export the data from Adamma.



If you don’t want to involved the invisible fields to the exported file, please disable the setting from below,

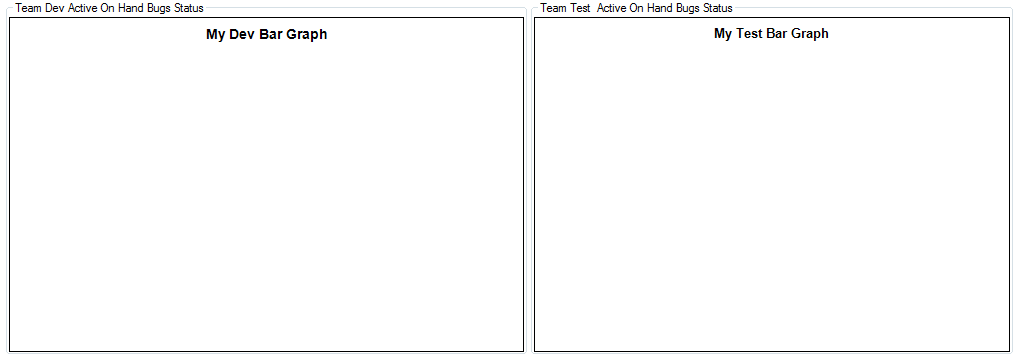


## Shot key description

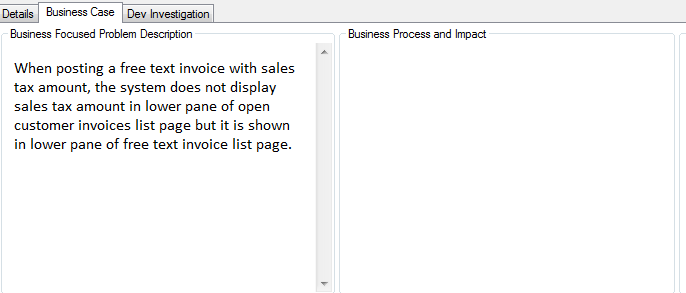
|  |  |
| --- | --- |
| Shot cut | Function |
| F5 | Call asynchronous mode one and two, start to query the data in grid view and teams group chart. |
| F2, Ctrl + G, Win +k, Alt + K | Call the GoTo form to specific bug id to access the bug in database DAXSE,AXSE,AX6. |
| F1 | Open this help document. |
| Esc | Exist each windows (Main, Goto, Setup.) in Adamma. |
| F3 | Start setup. |

## Known issues.

1. When the numbers of bugs queried was too much. The chart of “Dev/Test Bar Graph” was unable to display correctly like below, (by desgin)



1. To the fields in “Business case” and “Dev investigation”, sometimes can’t display correctly if the text inside is complex. Issue like below, (control issue)



## Extended functionalities under current framework

With more complex query, you are able to customized the code to build any chart you want. E.g. get the monthly resolved bugs during each month.

You need do the coding following.

1. Add your asynchronous worker at Adamma.Adamma().
2. Instantiation the class “ExpressionController”, Adamma will help automatically update the expression from the filter on main window.
3. Extend the enum: QueryType, Add your query to the constructor of sql class “QuerySQLGenerator”. Add your query branch to QuerySQLGenerator\ ConstructSQLFromTypeAndExpression.
4. Transfer the sql text to method ExecuteTFSQuery. QueryTeamFoundationServerViaWiq(). Get source data with a style of datatable.
5. Create a datatable in class GlobalValue to accept data. Send this source data to class ZedGraphController. Create the chart with style of “BarChart ,CurveChart,PieChart,StackBarChart”.