Pro-action- Master Data Application

Contents

[1 Introduction 4](#_Toc462847420)

[1.1 Objective 4](#_Toc462847421)

[1.2 References 4](#_Toc462847422)

[2 Analysis 5](#_Toc462847423)

[2.1 Master Data Application: 5](#_Toc462847424)

[2.2 Steps to load the application locally using postgres sql: 5](#_Toc462847425)

[2.3 Steps to load the application locally using Microsoft Sql Server: 5](#_Toc462847426)

[2.4 Current data model: 6](#_Toc462847427)

[3 Current Layout of each menu and future requirements: 6](#_Toc462847428)

[3.1 Dashboard- 6](#_Toc462847429)

[3.2 Organization: 7](#_Toc462847430)

[3.2.1 Company Codes 7](#_Toc462847431)

[3.2.2 SBU’s. 8](#_Toc462847432)

[3.2.3 BU’s 9](#_Toc462847433)

[3.2.4 Sub BU’s 10](#_Toc462847434)

[3.2.5 Department 11](#_Toc462847435)

[3.3 Clients: 11](#_Toc462847436)

[3.3.1 PSA Client Details 11](#_Toc462847437)

[3.3.2 Contracts 12](#_Toc462847438)

[3.3.3 Client Offices: 13](#_Toc462847439)

[3.3.4 Client Health Check Reports: 14](#_Toc462847440)

[3.4 Projects: 14](#_Toc462847441)

[3.4.1 PSA Projects 14](#_Toc462847442)

[3.4.2 PSA Activities 16](#_Toc462847443)

[3.4.3 PSA Billable Types 17](#_Toc462847444)

[3.4.4 Jira Projects 17](#_Toc462847445)

[3.5 Members: 18](#_Toc462847446)

[3.5.1 PSA Members 18](#_Toc462847447)

[3.5.2 Administrators 19](#_Toc462847448)

[3.5.3 Jira Project Members 20](#_Toc462847449)

[3.5.4 Jira Tempo teams 21](#_Toc462847450)

[3.5.5 Jira Users 21](#_Toc462847451)

[3.5.6 PSA Teams 21](#_Toc462847452)

[3.5.7 Skills Assessments – 22](#_Toc462847453)

[3.5.8 Team member Hierarchy Distribution 23](#_Toc462847454)

[3.5.9 Team skill level Distribution 24](#_Toc462847455)

[3.6 Lookup Tables: 25](#_Toc462847456)

[3.6.1 Jira Issue Type- 25](#_Toc462847457)

[3.6.2 Statuses- 25](#_Toc462847458)

[3.6.3 Currencies- 25](#_Toc462847459)

[3.6.4 Assessment Statuses- 26](#_Toc462847460)

[3.6.5 Countries- 26](#_Toc462847461)

[3.6.6 Exchange Rates- 27](#_Toc462847462)

[3.6.7 Industries- 27](#_Toc462847463)

[3.6.8 Jira Instances- 27](#_Toc462847464)

[3.6.9 Kpis- 28](#_Toc462847465)

[3.6.10 Member levels: 28](#_Toc462847466)

[3.6.11 PSA Locations 29](#_Toc462847467)

[3.6.12 Skill Categories 29](#_Toc462847468)

[3.6.13 Skill Levels 29](#_Toc462847469)

[3.6.14 Skill Sub Categories 30](#_Toc462847470)

[3.6.15 Skills 30](#_Toc462847471)

[3.6.16 Unit of measures 30](#_Toc462847472)

[3.7 Schemes: 31](#_Toc462847473)

[3.7.1 KPI Targets 31](#_Toc462847474)

[3.7.2 Target Schemes 32](#_Toc462847475)

[3.7.3 Billing Schemes 33](#_Toc462847476)

[3.7.4 Issue Type Billing Rates 33](#_Toc462847477)

[3.7.5 Member Billing Rates 34](#_Toc462847478)

[3.7.6 Planned Revenues 34](#_Toc462847479)

[3.7.7 PSA Cost Actuals 35](#_Toc462847480)

[3.7.8 PSA Revenue Actuals 35](#_Toc462847481)

[3.7.9 Team Billing rates 36](#_Toc462847482)

[4 Eazy BI Reporting 36](#_Toc462847483)

[5 PSA Jira-Tempo Timesheet Integration 38](#_Toc462847484)

[5.1 Process: 38](#_Toc462847485)

[5.2 Reporting: 39](#_Toc462847486)

[6 Proposed Technical Implementation (TBD) 39](#_Toc462847487)

[6.1 Tools used: 39](#_Toc462847488)

[6.2 Data model: 40](#_Toc462847489)

[6.3 Data Migration: 40](#_Toc462847490)

[6.4 PSA Information: 41](#_Toc462847491)

[6.5 Jira Integration: 42](#_Toc462847492)

[6.6 Deployment: 44](#_Toc462847493)

[7 Ruby & Rails to PHP Migration: 44](#_Toc462847494)

# Introduction

This document contains a basic understanding of the functional and technical aspects involved in Pro action- Master data application. The application is currently built using Ruby on Rails Web framework with Postgres as RDBMS. The application is not fully functional but this documents aims to detail the current layout and any gaps to address the current issues.

## Objective

The objective of this document is to identify the scope of the application that is currently functional, identify the gaps and design for the non-functional areas /areas of improvement.

## References

|  |  |
| --- | --- |
| Applications | Links |
| Master Data Application | <https://proaction-global.cgi.com/>admin/ |
| Jira Application | <https://proaction-global.cgi.com/>jira/ |

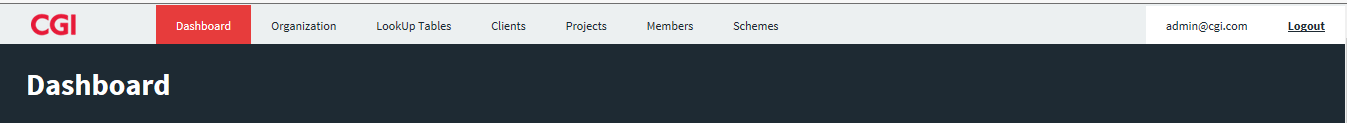
# Analysis

The master data application would pull and display the data from PSA and Jira Applications. Currently the data is synced on adhoc basis via XLS/CSV/XML

Tools used: Ruby on Rails Web framework , Post gres (SQL Server to be used in Production)

## Master Data Application:

The applications has following tabs:



The web framework is built using Ruby and Rails application using postgres sql as a database. The current version of application is deployed to cloud. The application open to: <http://ptw-dev.cloudapp.net/admin>

## Steps to load the application locally using postgres sql:

* Download Postgres SQL and install it.
* Restore the file- proadmin-db-08-10.sql
  + Create an empty database in postgres sql e.g. Proaction
  + Use the command below to restore:
    - psql (connection options here) database
    - database=# \i /path to proadmin-db-08-10.sql

E.g. NewTestDB-# \i E:/db-rbl-restore-20120511\_Dump-20120514.sql

* + - Check using \dt+ to see the tables loaded or use the tool PgAdminIII to see the list of the tables loaded.
* Download the installer for Ruby on rails and install it
* Copy the current code base folder under C:\Sites\proadmin
* Changed the database.yml to use postgres sql . Refer the sample file-
* Install node.js and restart the computer
* On the command prompt for Ruby & Rails Application- use rails server to start and load the application
* The application loads in <http://localhost:3000/>

## Steps to load the application locally using Microsoft Sql Server:

* Install Sql Server 2012 ( I have installed free Sql server Express edition)
* Restore the db backup from Master data live server-
  + Use the Restore Database from SSMS GUI ( right click on database)
  + Select the .bak file provided in the Source under Devices
  + Under destination, select the database. If new one, put the new database name.
  + Say ok and restore will be done.
* Download the installer for Ruby on rails and install it
* Copy the current code base folder under C:\Sites\proadmin
* Install node.js and restart the computer
* Changed the database.yml to use Microsoft sql server. Refer the sample file-
* On the command prompt for Ruby & Rails Application- use rails server to start and load the application
* The application loads in <http://localhost:3000/>

## Current data model:

* Relationships at the application layer
* Types of tables:
  + PSA Information tables
  + Lookup Tables
  + Jira tables
  + Client tables
* See the data model below (zoom out).



* The master data was loaded using the excel file imports done sometime ago or manually entered.

# Current Layout of each menu and future requirements:

## Dashboard-

Currently shows following details:

* Recently added contracts
* Actual PSA Revenue
* Recent Customers
* Recent Projects

Organization:

### Company Codes

Displays CGI Company Details

There is no relevant data in production. Not sure what the company codes are?

Displays the company details.

* Start date
* End date
* PSA Company code
* Name
* Description
* Head of Unit
* Address currently not displayed.

Filter /Search Conditions:

* Name
* CGI Company code

Action Buttons:

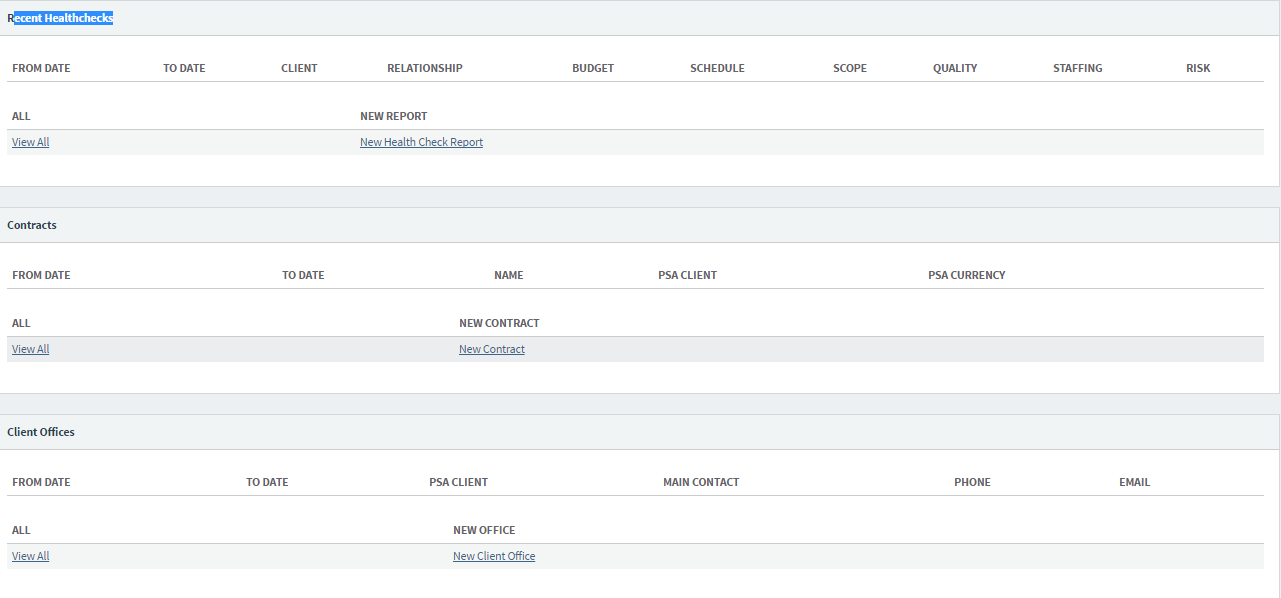
* New CGI Company code
* Update CGI Company Code via Edit

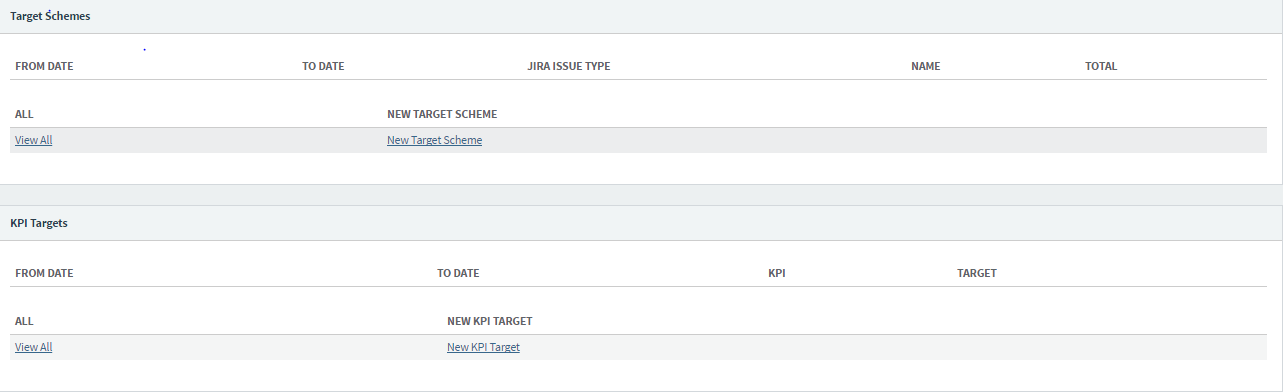
Data Source: Manual

When a particular company is viewed, there is a facility to given to add or view the following at the company level.

* Recent Health checks-[Refer Section 3.3.4](#_Health_Check_Reports:)
* Contracts- [Refer Section 3.3.2](#_Contracts)
* Client Offices-- [Refer Section 3.3.3](#_Client_Offices:)
* Target Schemes- [Refer Section 3.7.2](#_Target_Schemes)
* KPI Targets- - [Refer Section 3.7.1](#_KPI_Targets)

For each of these sub menus, there is an ADD button (e.g. New KPI Target) to add the values /data for that Company code. If these menus are used, then only they are added at company level. If they are directly added via their individual screens in master data, the values are added at generic level and not at the individual sbu or bu level





### SBU’s.

Displays SBU Details

* Start date
* End date
* CGI Company code – Drop down enabled. [Uses CGI Company Code values- Section 3.2.1](#_Company_Codes:_-) Currently it allows null but needs to be mandatory to be filled in
* PSA SBU Code
* Name
* Description
* Address <currently not displayed>
* Head of Unit

Filter Conditions/Search Criteria:

* Name
* CGI Company Code- Drop down enabled to list CGI Company codes to select.
* PSA SBU Code- This needs to be added along with a drop down list.
* Created At
* Updated At

Action Buttons:

* New SBU
* Update SBU via Edit
* Import SBU- This gives functionality to import SBU via a file. The file format should be as below i.e. it should match the headers in SQL Table.



The import functionality currently is only for new records. It cannot update old records with this functionality. Moreover it ensures, if the file is imported twice, it does not create any new records.

When a particular company is viewed, there is a facility to given to add or view the following at the SBU level.

* Recent Health checks
* Contracts
* Client Offices
* Target Schemes
* KPI Targets

### BU’s

Displays BU Details

* Start date
* End date
* SBU - Drop down list enabled/ View should show the link to SBU page. Uses SBU Codes from [Section 3.2.3](#_SBU_Details)
* BU Code
* Name
* Description
* Address <currently not displayed>
* Head of Unit

Filter/ search criteria:

* BU Code- This needs to be added along with a drop down list.
* Name
* SBU Code- Drop down enabled to list SBU codes to select.

Action Buttons:

* New BU
* Update BU via Edit
* Import BU- This functionality may need to be added

When a particular company is viewed, there is a facility to given to add or view the following at the BU level.

* Recent Health checks
* Contracts
* Client Offices
* Target Schemes
* KPI Targets

### Sub BU’s

Displays SUB BU Details

* Start date
* End date
* BU- Drop down list enabled/ View should show the link to BU page. Uses BU Codes from [Section 3.2.3](#_BU_Details)
* PSA SUB BU Code
* Name
* Description
* Address <currently not displayed>
* Head of Unit

Search/filter Conditions:

* Name
* BU- Drop down enabled to list BU codes to select.
* Sub BU Code- This new search needs to be added

Action Buttons:

* New Sub BU
* Update Sub BU via Edit
* Import Sub BU- This functionality may need to be added

When a particular company is viewed, there is a facility to given to add or view the following at the BU level.

* Recent Health checks
* Contracts
* Client Offices
* Target Schemes
* KPI Targets

.

### Department

Displays Department Details

* Start date
* End date
* PSA Department Code
* SUB BU Code- Link to corresponding sub-BU page- [Section 3.2.4-Sub BU](#_SUB_BU_Details) and on edit page drop down enabled for Sub BU Code.
* Name
* Description
* Location- Link to PSA Locations under Lookup Menu- [Section 3.6.9 PSA Locations](#_PSA_Locations)
* Head of Unit

Currently Department code is appearing twice on the screen.

Search/Filter Condition:

* Name
* SUB BU- Drop down enabled to list Sub BU codes to select
* Department Code- This new search needs to be added with a drop down of existing department code.

Action Buttons:

* New Department
* Update Department via Edit
* Import Department- This functionality may need to be added

When a particular company is viewed, there is a facility to given to add or view the following at the BU level.

* Recent Health checks
* Contracts
* Client Offices
* Target Schemes
* KPI Targets

## Clients:

### PSA Client Details

It displays the client details as follows:

* Client id
* Name
* Created At
* Updated At
* PSA Client
* Description
* Address
* Website
* Phone
* Facebook
* Industry – Link to Industry from the Lookup menu. On Edit, drop down enabled to select the industry the client belongs to.
* Start date
* End date
* Client Number
* Primary BU – Link to the corresponding BU page. On Edit, drop down enabled to select the BU the client belongs to.

Filter/search condition:

* Name
* Industry- drop down enabled to select the industry to search
* Client\_number – to be added
* Created at

Action Buttons:

* New PSA Client

Data Source: PSA

Following Lookup tables are being used:

* [Industry](#_Industries-)
* [BU](#_BU_Details)

### Contracts

Displays contract information as follows:

* Start date
* End date
* External Contract ID
* PSA Client- Link to PSA client table. On Edit, drop down enables to select the PSA client.
* Name
* Description
* Currency- Link to the PSA currency table. On Edit, drop down enables to select the currency.

Search/filter condition:

* Name
* PSA Currency – Need to check if it is needed.
* CONTRACTABLE TYPE – This field is populated when the contracts are added and determines which level it has been added for. It has values as Company, SBU, BU, Sub-bu, and department.
* CONTRACTABLE- This field is populated with the value of the type selected in the CONTRACTABLE TYPE field.

E.g. If the CONTRACTABLE TYPE is sbu, then CONTRACTABLE will be the corresponding SBU.

To be added:

* PSA Client

Data Source: Manual ( Can be from PSA- To check)

Following Lookup tables are being used:

* + [PSA Statuses](#_Statuses-)
  + [Department](#_Department)

Action Buttons:

* + New Contract
  + Update Contract via Edit

### Client Offices:

This menu is currently un-available from the main title bar. The client office page is accessed via the **New client office link** available under Company, SBU, BU, Sub-bu and Department.

We think the same functionality of client offices need to extend at the Client level i.e. to view the client offices when a particular client is selected.

* From date
* To date
* Name
* Main Contact
* PSA Client- Link to the PSA client on view and drop down of the available clients to select.
* Address- This address would be the registered address only.
* Email
* Phone

Action Buttons:

* + New Client Office
  + Update Client Office

Data Source:

* Manual
* Proposed: TBD

Filter Conditions on:

* + Name
  + From Date
  + To Date
  + PSA Client- drop down of the available clients to select.
  + Contactable Type- This field is populated when the client officea are added and determines which level it has been added for. It has values as Company, SBU, BU, Sub-bu, and department.
  + Contactable This field is populated with the value of the type selected in the Contactable TYPE field.

E.g. If the Contactable TYPE is sbu, then Contactable will be the corresponding SBU.

This field is populated with the value of the type selected in the Contactable TYPE field.

### Client Health Check Reports:

This menu is currently un-available from the main title bar. The client office page is accessed via the New client office link available under Company, SBU, BU, Sub-bu and Department.

* From date
* To date
* Name
* PSA Client- Link to the PSA client on view and drop down of the available clients to select.
* Client Relationship Status- Drop down using [Statuses](#_Statuses-)
* Client Relationship Status Description
* Budget Finance Status- Drop down using [Statuses](#_Statuses-)
* Budget Finance Status Description
* Schedule status- Drop down using [Statuses](#_Statuses-)
* Schedule status Description
* Scope status- Drop down using [Statuses](#_Statuses-)
* Scope status Description
* Quality status- Drop down using [Statuses](#_Statuses-)
* Quality status Description
* Staffing status- Drop down using [Statuses](#_Statuses-)
* Staffing status Description
* Risk status- Drop down using [Statuses](#_Statuses-)
* Risk status Description

Action Buttons:

* + New Client Health Check Report
  + Update Client Health Check Report

Data Source:

* Manual
* Proposed: TBD

No separate page opens for client health check report to search/filter

## Projects:

### PSA Projects

Displays Project Details

* Department- Drop down enabled to select the available department on Edit. Refer [Section Department](#_Department_Details)
* Name
* PSA Project- On edit, this shouldn’t be editable I guess. This is the project ID Field.
* Description
* PSA Member- Member associated with a project. Currently we can have only 1 member associated with the project.

A separate table will be needed to store the PSA members associated with the project in the same way as Jira projects associated to PSA members are currently stored.

* PSA Status – Drop down enables to select the status from [PSA Statuses](#_Statuses-) lookup on Edit.
* Jira Projects- This is not shown currently on view.
* PSA Client id- This new field needs to be added to link to the project to the corresponding contract

On selection of a particular project for editing, it also displays Jira projects. The associated Jira project is not displayed at the summary level information or at the view level above

Jira projects need to be displayed on the view screen.- To be asked if needed

The Jira projects are displayed via- jira\_projects tables and selected ones are stored in jira\_project\_psa\_projects.

When you psa project, it displays the current jira projects associated with it. If they are deselected and new ones are selected, the older entries are removed from jira\_project\_psa\_projects table.

How a Jira project is linked to the PSA project:

• Add new jira project via Jira projects menu in master data.

• Link that project to PSA project via the project menu:

* + Edit the project
  + Select the new jira project

**Filter Conditions on:**

* + PSA Project ID - This needs to be added with a drop down of PSA Project ID
  + PSA Member name- This needs to be removed.
  + PSA Status- Drop down enabled to list the PSA Status and select
  + Created date
  + Update date
  + Name
  + Department - Drop down enabled to list the Department and select .

**Data Source**:

* + Currently manual or backend upload
  + Proposed -PSA

The Jira projects are displayed via jira\_projects tables via Jira Integration. The list is currently displayed as checkboxes. Some other component that allows members contained in a box can be used.

Following Lookup tables are being used:

* + [PSA Statuses](#_Statuses-)
  + [Department](#_Department)

Action Buttons:

* + New PSA Project
  + Update PSA project via Edit

### PSA Activities

Describes the PSA activities associated with the project.

At summary level it shows:

* External PSA Activity ID
* PSA Project – hyperlink for the psa project above
* Name
* Description

On selecting a particular PSA Activity, following info is displayed:

* ID
* PSA Activity
* Name
* Description
* Status – This is empty or ‘A’ for most of the projects. I think this relates to General Status/Summary Status that is available on the activity list sheet available from PSA.

Refer Activity sheet under Proposed Technical Implementation.

* PSA project- On Edit, a drop down to list all the associated [PSA projects](#_PSA_Projects)
* Created by
* Updated by
* PSA Billable Type – Link to Lookup table for [Billable Types](#_PSA_Billable_Types)- Section 3.4.3 and in edit mode a drop down to select the billable type.

Filter Conditions on:

* Name
* PSA Project- Drop Down enabled for the projects
* PSA Billable Type- Drop Down enabled for the PSA Billable Types
* Created AT
* PSA activity ID- This filter needs to be added with drop down for existing PSA Activity id.

Data Source:

* Currently manual or backend upload. Each Jira project sends information in the following format that is updated to master data manually.



* Proposed: PSA

PSA needs to send the billable type information as well. Based on how the project activity is currently setup in the PSA, the number of rows needs to repeat for each billable type. For e,g. if a particular activity allows all 3 billable types, then 3 entries have to be created.

Lookup Tables used:

* [Billable Types](#_PSA_Billable_Types)

Action Buttons:

* + New PSA Activity
  + Update PSA Activity via Edit

### PSA Billable Types

Describes the PSA Billable types: This act as a lookup table and used in PSA Activities. This table needs to be moved to Lookup menu.

At summary level:

* Name
* Description

On viewing/editing a particular billable type:

* PSA Billable Type
* Name
* Description
* Created At
* Updated At

Filter Conditions on:

* Name
* Created At

Data Source: Manual

Action Buttons:

* + New PSA Billable Types
  + Update PSA Billable Types on Edit Screen

### Jira Projects

At summary level:

* Name
* Project Category – Not a drop down currently
* Jira Instance
* Jira Project ID
* API End Point

On viewing/editing a particular Jira project:

* Name
* Jira Project
* Description
* Jira Instance- Link to Lookup table- [Jira Instances](#_Jira_Instances-). Edit screen shows a drop down of available instances
* Project category
* Project Lead
* URL

Action Buttons:

* + New Jira project
  + Update Jira project on Edit Screen
  + Update Projects- This button pulls the projects added in Jira, Refer [Jira Integration](#_Jira_Integration:).

This functionality is currently switched off in Production.

Filter Conditions on:

* Name
* Project Category
* Created At
* Jira instance- Drop down menu of Jira Instances.

Data Source:

* Current : TBD
* Proposed- Jira Integration

Lookup Tables used: [Jira Instances](#_Jira_Instances-).

## Members:

### PSA Members

Displays member level information:

At summary level:

* PSA Member
* First Name
* Last Name
* Department
* Created At

On viewing/editing a particular member:

* ID
* PSA Member
* First Name
* Last Name
* Department- Link to the table- [Department](#_Department_Details). View should show a link to the corresponding department and edit shows a drop down of available departments.
* Personal Status – What values to put?
* Employee Status- What values to put?
* Location – Drop down currently displays numbers. Need to map to PSA location names. View should show link to PSA location and on edit a drop down from available PSA location. Uses the lookup- [PSA Locations](#_PSA_Locations)
* Created At
* Updated At
* Member level- Link to the lookup table – Member level. View should show a link to the corresponding member level and edit shows a drop down of available member levels.

Uses the lookup- [Member level](#_Member_levels:).

* From date
* To date
* Rate card cost
* Real Member Cost- This is currently being used for reporting but as a number and no currency information.
* Currency- Proposal is to add currency as a drop down referring the Currency lookup so the user can select the appropriate currency while putting the member cost.
* Target Member Cost
* Jira User-
  + This is not available during EDIT/ Need to investigate how this information is populated, Can this populated via Jira Users table using member email?
  + During automatic upload, a function needs to written to populate the Jira user using the member email and Jira users table.
  + When a user adds PSA member manually, Jira User can be selected as drop down from the available Jira Users

Filter Conditions on:

* PSA Member
* Last Name
* First Name
* Department- drop down list of available departments to select
* Location- drop down list of available locations to select
* Created Date

Data Source:

* Current - Manual
* Proposed - PSA

Action Buttons:

* + New PSA member
  + Update PSA member

Lookup Tables used:

* [Department](#_Department_Details)
* [Member Level](#_Member_levels:)
* [PSA Locations](#_PSA_Locations)

### Administrators

Displays users of this application

* ID
* Email
* Current Sign in At
* Last Sign in At
* Sign in Count
* Created At
* Updated At
* Gauth Enabled

On edit, it asks for

* Email
* Password
* Password Confirmation

The email address has to be validated to use the cgi or cgi federal email address.

Currently all users added to the application are administrators. We need to think of what kind of different users would be needed.

In our opinion, we can have 3 different types of users:

* Admin user- - read write access to all menus
* Master user- write access for selected menus- lookups and few menus that are not automatically updated and read access to all
* Users- only read access to all menus

### Jira Project Members

Displays Jira Project Members with role and email

* Jira Project – Not possible to add/edit.
* Jira Project Role- No drop down for Jira Project roles
* Member email – Currently no validation in place to check if a valid PSA member. Need a drop down for members.

On edit Screen,

* Jira Project Role- No drop down currently. But good to have.
* Member email- No drop down currently. But good to have

Not possible to modify or add Jira project details.

Data source: This table is currently updated with the information from Jira every 15 minutes. There is crono job configured in \config\ cronotab.rb that calls -\app\jobs\jira\_permissions\_job.rb every 15 minutes to update this table in Master data.

Filter Conditions on:

* Member Email
* Jira Project Role
* Created At
* Updated at

Action Buttons:

* + New Jira Project member
  + Update Permissions – This button updates the permissions/ roles for the Jira team members. The information is fetched from Jira.

Lookup Tables used: None

### Jira Tempo teams

Displays Tempo team and description

* Name
* Description

Data source:

* Current – Empty table
* Proposed - Jira Integration

Filter Conditions on:

* Name

Action Buttons:

* + New Jira Tempo Team
  + Update Teams – This pulls up the team from Jira.

Lookup Tables used: None

### Jira Users

Currently un-used---replaced by crowd

Displays Jira instances and uses

* Jira Instance – Drop down of Jira Instances on Edit Screen.
* Jira User
* Email
* User name

Data source:

* Current – Almost empty table. Only 1 user.
* Proposed - Jira Integration

Filter Conditions on:

* Email
* Username
* Created At
* Jira Instance- drop down list of available Jira Instances to select

Action Buttons:

* + New Jira User
  + Update Jira User via Edit

Lookup Tables used: [Jira Instances](#_Jira_Instances-)

### PSA Teams

Currently un-used. This will be a derived table from PSA projects and contracts.

Displays PSA Teams

* PSA Team
* Name
* Description
* Contract – Links to the Contract screen. Drop down of contracts on the edit screen.
* Address - Distributed teams possible

Data source:

* Current – Empty table
* Proposed – to be populated using from psa\_projects , contracts using a trigger whenever an entry is inserted /updated in psa\_projects/ or a function routine to update the table when a new PSA project entry is created.
  + Psa\_team\_id- Projects. Psa\_project\_id
  + Name - Projects.name
  + Description – Projects.description
  + Contract\_id – Get contract id using the Projects.psa\_client and linking to the contracts table

Filter Conditions on:

* Name
* Created At
* Contract- drop down list of available Contracts to select

Action Buttons:

* + New PSA Team
  + Update PSA Team Via Edit

Lookup Tables used: [Contracts](#_Contracts)

### Skills Assessments –

Currently un-used.

This displays skills for each member. Only 1 skill per member is displayed at one go.

At summary level

* PSA Member –Drop down of the PSA Members table
* Skills- Link to skills page on the lookup menu. On Edit, drop down of the available skills to select.
* Skill Level - Link to skill level page on the lookup menu On Edit, drop down of the available skill level to select.
* Assessment Status- Link to Assessment Statuses page on the lookup menu- On Edit, drop down of the available assessment status to select. Need more information on the various status values and its source
* Name – What is this field for?
* Description– What is this field for?
* Assessor- Drop down of the PSA Members table
* Assessed on
* From Date
* To date

Data source:

* Current – Empty table
* Proposed – TBD

Filter Conditions on:

* PSA Member
* Skill- - drop down list of available skills to select
* Skill level- drop down list of available skill levels to select

Action Buttons:

* + New Skill Assessment
  + Update Skill assessment via Edit

Lookup Tables used:

* [PSA Member](#_PSA_Members)
* [Skill](#_Skills)
* [Skill Level](#_Skill_Levels)
* [Assessment Status](#_Assessment_Statuses-)

### Team member Hierarchy Distribution

Currently un-used

This describes the member level hierarchy of the team and its percentage e.g. how many consultants within a team. But at a time, it displays only the percentage of 1 particular member level.

At summary level

* Jira Tempo Team- Link to Jira Tempo Team. On edit, drop down of the available Jira Teams.
* Member Level- Link to Member levels. On edit, drop down of the available member levels to select.
* Percentage
* From date
* To date

At view/edit level, start and end date are added.

Data Source:

* Current Status: Empty
* Proposed:

This information needs to be derived from PSA members that are assigned to Jira Tempo team and then find the percentage level for each designation. The member level information is stored in the PSA Members table. Jira user id is on the PSA Member table.

There is a table – dbo.jira\_user\_jira\_tempo\_teams that stores jira users and the corresponding jira tempo team id. This table is not currently populated and needs to be populated via Jira Integration and then do the linkage to get the information.

Need to investigate how this information can be extracted,

Filter /Search Criteria:

Jira Tempo Team- Drop down list for the team.

Action Buttons:

* + New Team member Hierarchy Distribution
  + Update Team member Hierarchy Distribution via Edit

Lookup Tables used:

* [Jira Tempo Teams](#_Jira_Tempo_teams)
* [Member level](#_Member_levels:)

### Team skill level Distribution

Currently un-used

This describes the team level split based on their skill level and its percentage e.g. how many members at a particular skill within a team.

At summary level

* Jira Tempo Team- Link to Jira Tempo Team. On edit, drop down of the available Jira Teams.
* Skill Level -Link to Skill levels. On edit, drop down of the available skill levels to select.
* Percentage
* From date
* To date

Data Source:

* Current Status : Empty
* Proposed:

The skill level information can be obtained from skill assessments for a PSA Member and then then find the percentage level for each skill. Jira user id is on the PSA Member table.

There is a table – dbo.jira\_user\_jira\_tempo\_teams that stores jira users and the corresponding jira tempo team id. This table is not currently populated and needs to be populated via Jira Integration and then do the linkage to get the information.

There needs to be a validation that the total percentage distribution should not be more than 100%

Filter /Search Criteria:

Jira Tempo Team- Drop down list for the team.

Action Buttons:

* + New Team skill level Distribution
  + Update Team skill level Distribution via Edit

Lookup Tables used:

* [Jira Tempo Teams](#_Jira_Tempo_teams)
* [Skill level](#_Skill_Levels)

## Lookup Tables:

### Jira Issue Type-

Displays the Jira issue types.

* Name
* Description

.

Filter /Search Criteria:

* Name:

Data source

:

* Manual

Action Buttons:

* + New Jira Issue Type
  + Update Jira Issue Type on Edit

### Statuses-

Displays the PSA status. Static table.

* Name
* Description

Filter /Search Criteria:

* Name
* PSA Status

Action Buttons:

* New PSA Status
* Update PSA Status on Edit

Data Source: Manual

### Currencies-

Displays the available currencies.

* From date
* To Date
* PSA Currency
* Name
* Description
* ISO currency code
* Currency Symbol

Filter /Search Criteria:

* Name
* Created at

Data Source: Manual or one time load from PSA

Action Buttons:

* New PSA Currency
* Update PSA Currency on Edit

### Assessment Statuses-

Displays the assessment status of skill of a particular member

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name

Data Source: Manual

Up to 3 entries currently added.

Action Buttons:

* New Assessment Status
* Update Assessment Status on Edit

### Countries-

Displays the countries

* From Date
* To date
* Name
* Description
* ISO Country Code
* UN Country Code

Filter /Search Criteria:

* Name
* Created At

Data Source: Table is populated. Just needs to be copied.

.

Action Buttons:

* New Country
* Update Country on Edit

It is not used as direct lookup within the master data but is populated for future use.

### Exchange Rates-

Displays the exchange rates between the currencies

* From Date
* To date
* From Currency – Lookup to PSA currencies. On Edit, drop down for available PSA currencies to select
* To Currency – Lookup to PSA currencies. On Edit, drop down for available PSA currencies to select
* Rate

Filter /Search Criteria:

* From Currency

Data Source: Table is populated. Just needs to be copied. But need to think from where the data could be updated as rates keeps on changing.

Lookup Tables used:

* [PSA Currency](#_Currencies-)

Action Buttons:

* New Exchange Rate
* Update Exchange Rate on Edit

It is not used as direct lookup within the master data but is populated for future use.

### Industries-

Displays the industries which the clients are associated with,

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Data Source: Manual

Action Buttons:

* New Industry
* Update Industry on Edit

### Jira Instances-

Displays the available Jira Instances.

* Name
* Description
* url

Filter /Search Criteria:

* Name
* Created At

Data Source: Manual

Action Buttons:

* New Jira Instance
* Update Jira Instance on Edit

### Kpis-

List the various KPI’s used for Incidents and Problems.

* From date
* To date
* Name
* Description
* Jira Issue type- Drop down menu for [Jira Issue types- Section 3.6.1](#_Jira_Issue_Type-) during Edit and Link to the Jira Issue type in View mode.

Filter /Search Criteria:

* Name
* Jira Issue type- Drop down to be enabled for [Jira Issue types- Section 3.6.1](#_Jira_Issue_Type-)

Data Source: Manual

Action Buttons:

* New KPI
* Update KPI on Edit

### Member levels:

List the member level (PSA Designation) for members in CGI

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Action Buttons:

* New Member Level
* Update Member Level on Edit

Data Source: Manual

Currently upto 7 levels have been added.

### PSA Locations

List the various CGI office locations

* From Date
* To date
* Code
* Description

Filter /Search Criteria:

* Code

Action Buttons:

* New PSA Location
* Update PSA Location on Edit

Data Source: One time load from PSA and then after can be manual.

### Skill Categories

List the various skill categories a member can have.

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Action Buttons:

* New Skill Category
* Update Skill Category on Edit

Data Source: Need to be determined. Date can be fetched from PSA RM Tool.

Currently this information is sourced from Internet

### Skill Levels

List the various skill levels a member can have.

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Action Buttons:

* New Skill Level
* Update Skill Level on Edit

Data source: Manual- or can be mapped to PSA RM skill levels.

Currently up to 7 levels have been added.

### Skill Sub Categories

List the various skill categories a member can have.

* From Date
* To date
* Skill category- Link to [skill Category- section 3.6.9](#_Skill_Categories) on View and on Edit –drop down menu for skill Categories
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Action Buttons:

* New Skill Sub Category
* Update Skill Sub Category on Edit

Data Source: Need to be determined. Data can be fetched from PSA RM Tool.

Currently this information is sourced from Internet.

### Skills

List the various skills a member can have and what category & sub category the member belongs to.

* From Date
* To date
* Code
* Name
* Description
* Skill Category- Link to [skill Category- section 3.6.9](#_Skill_Categories) on View and on Edit –drop down menu for skill Categories.
* Skill Sub Category- - Link to [skill sub Category- section 3.6.11](#_Skill_Sub_Categories) on View and on Edit –drop down menu for skill sub Categories

Filter /Search Criteria:

* Name
* Code
* Created At

Action Buttons:

* New Skill
* Update Skill on Edit

Data source: Manual- or can be mapped to PSA RM skills.

Currently this information is sourced from Internet. If these have to be used, all members will need to map to these skills.

### Unit of measures

List the unit of measures for any measurement e.g.KPI

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Created At

Action Buttons:

* New Unit of Measure
* Update Unit of Measure on Edit

Data source: Manual

## Schemes:

### KPI Targets

List of KPI Targets for each KPI listed in [KPI-Section 3.6.6](#_Kpis-). This field is used in Eazy BI Reporting on Jira to compare the target and actual KPI for a particular project.

* From Date
* To date
* KPI- Link to [KPI](#_Kpis-) on View and Drop down of KPI on edit. This currently shows as Metric 1 , metric 2 etc. It needs to show the descriptive metrics.
* Target-Drop down list of values
* Unit of measure- Link to Unit of measure- [Section 3.6.13](#_Unit_of_measures) on View and drop down on edit.
* Targetable Type-This field is populated when the KPI target metric is added and determines which level it has been added for. It has values as Company, SBU, BU, Sub-bu, and department.
* Targetable- This field is populated with the value of the type selected in the Targetable type field.

E.g. If the targetable type is sbu, then tagrtetable will be the corresponding SBU.

Filter /Search Criteria:

* KPI- Drop down to be enabled
* Target
* Targetable Type
* Targetable

Action Buttons:

* New KPI Target
* Update KPI Target

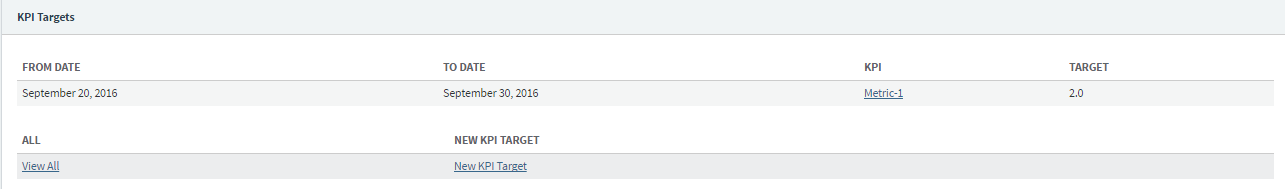
Data source: Manual

The values have to be entered via the menu available when you view a particular company code or SBU e.g.

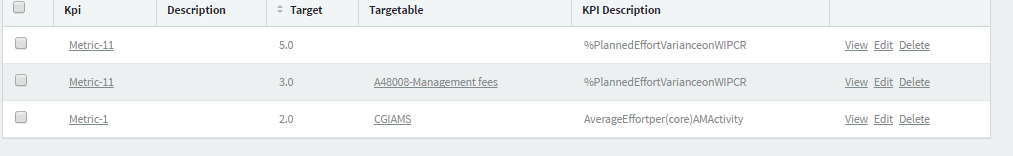
* Select a SBU to view



* Then add the KPI Target for that SBU using the New KPI Target.



* KPI Target gets added for a SBU. See the above screenshot.
* It can also be viewed on the KPI Target screen under



* Similarly you can add KPI for any company, sbu, bu, sub bu etc.

The concept behind this design is that the same table is used to hold KPI targets at various levels. It has been termed as Polymorphic association. The targetable type and targetable will differentiate the value stored for same KPI.

### Target Schemes

<what is this-TBD>

* From Date
* To date
* Jira Issue Type- Link to [Jira Issue types](#_Jira_Issue_Type-) on View and Drop down of Jira Issue type on edit.
* Name
* Description
* Target Direct Cost
* Target InDirect Cost
* Total1
* Management Overhead
* Total2
* Other Overhead
* Total3
* Risk – What values this field should hold?
* Total4
* Margin
* Total

Filter /Search Criteria:

* Name
* Schemable Type- This field is populated when the target scheme is added and determines which level it has been added for. It has values as Company, SBU, BU, Sub-bu, and department.
* Schemable- This field is populated with the value of the type selected in the Targetable type field.

E.g. If the Schemable Type is SBU, then Schemable will be the corresponding SBU.

Action Buttons:

* New Target Scheme
* Update Target Scheme.

Data source: Manual

Proposed: From where this data would come from?

### Billing Schemes

<what is this-TBD>

* From Date
* To date
* Name
* Description

Filter /Search Criteria:

* Name
* Billable Type- This field should populated when the billing scheme is added and determines which level it has been added for. It has values as Company, SBU, BU, Sub-BU, and department. Currently this functionality is unavailable.
* Billable - This field should be populated with the value of the type selected in the Billable type field. Currently this functionality is unavailable. This is not on the search menu as well.

E.g. If the Billable type is SBU, then Billable will be the corresponding SBU.

Action Buttons:

* New Billing Scheme
* Update Billing Scheme.

Data source: Manual

Proposed: From where this data would come from?

### Issue Type Billing Rates

This determines the [billing scheme](#_Billing_Schemes) applicable for a particular Issue type.

* From Date
* To date
* Issue Type - Link to [Jira Issue types](#_Jira_Issue_Type-) on View and Drop down of Jira Issue type on edit.
* Billing Scheme-- Link to [billing scheme](#_Billing_Schemes)  on View and Drop down of Jira Issue type on edit.
* Rate

Filter /Search Criteria:

* Billing Scheme

Action Buttons:

* New Issue type Billing Rate
* Update Issue type Billing Rate

Data source: Manual

Proposed: From where this data would come from?

### Member Billing Rates

This determines the [billing scheme](#_Billing_Schemes) applicable for a particular member.

* From Date
* To date
* Issue Type - Link to PSA Members on View and Drop down of [PSA Members](#_PSA_Members) on edit
* Billing Scheme-- Link to [billing scheme](#_Billing_Schemes)  on View and Drop down of Jira Issue type on edit.
* Rate

Filter /Search Criteria:

* PSA Member –Drop down enabled with the list of members for users to select
* Billing Scheme- Drop down enabled with the list of billing schemes for users to select.

Action Buttons:

* New Member Billing Rate
* Update Member Billing Rate

Data source: Manual

Proposed: From where this data would come from?

### Planned Revenues

These are the planned revenues per contract.

* Contract - Link to [Contracts](#_Contracts) on View and Drop down of [Contracts](#_Contracts) on edit
* Planned Revenue
* Description
* From date
* To Date
* Contingency
* Estimated revenue
* Forecast
* Original Budget
* Original Profitability committed
* Original Signed Value
* Provision Value
* Revised Budget
* Revised Signed Value
* Total Anticipated Value
* Value Pending Signature

Filter /Search Criteria:

* Contract –Drop down enabled with the list of contract for users to select
* From Date
* To Date

Action Buttons:

* New Planned Revenue
* Update Planned Revenue

Data source: Manual

Proposed: From where this data would come from?

### PSA Cost Actuals

This displays the PSA Actual Cost associated with a contract

* Contract – Drop down list of available contracts for the user to select on Edit and link to the contract on View.
* PSA Cost Actual
* From Date
* To date
* Monthly Cost Value

Filter /Search Criteria:

* Contract –Drop down enabled with the list of contract for users to select
* From Date
* To Date

Data source:

* Manual
* Proposed: From where this data would come from?

### PSA Revenue Actuals

This displays the PSA Revenue Actuals associated with a contract

* Contract – Drop down list of available contracts for the user to select on Edit and link to the contract on View.
* PSA Revenue Actual
* From Date
* To date
* Monthly Revenue Value
* Monthly Provision Booked

Filter /Search Criteria:

* Contract –Drop down enabled with the list of contract for users to select
* From Date
* To Date

Data source:

* Manual
* Proposed: From where this data would come from?

Action Buttons:

* New PSA Revenue Actual
* Update PSA Revenue Actual on Edit

### Team Billing rates

This displays the tempo team billing scheme the team is associated with.

* From Date
* To date
* Jira Tempo team- Drop down list of available [Tempo team](#_Team_Billing_rates) to select on Edit and link to the tempo team on view
* Billing Scheme- Drop down list of available [billing schemes](#_Billing_Schemes) to select on Edit and link to the billing schemes on view
* Rate

Filter /Search Criteria:

* Jira Tempo Team- Drop down of available [Tempo team](#_Team_Billing_rates) to select
* Billing Scheme- Drop down list of available [billing schemes](#_Billing_Schemes) to select

Action Buttons:

* New Team Billing Rate
* Update Team Billing Rate

Data source: Manual

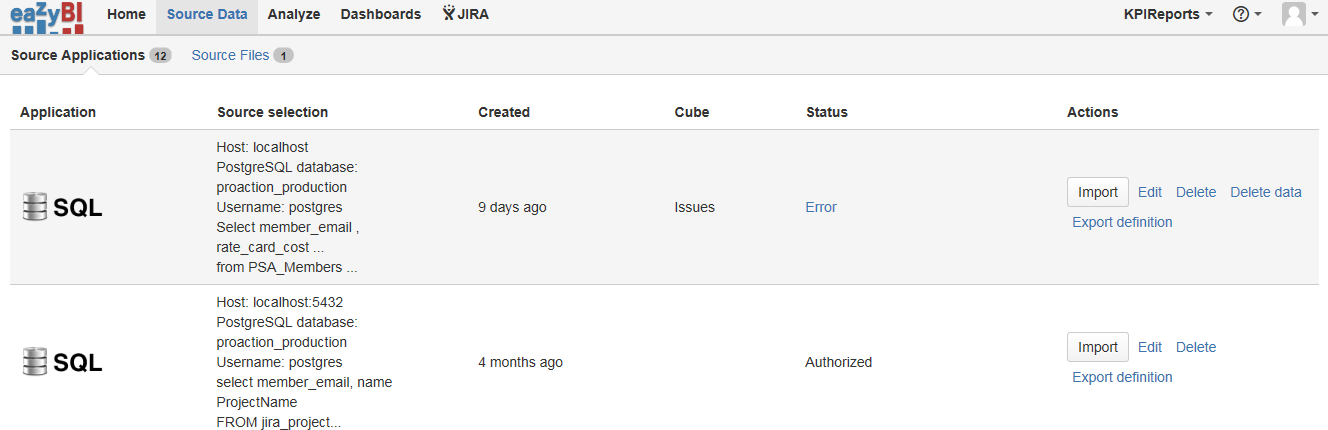
Proposed: From where this data would come from?

# Eazy BI Reporting

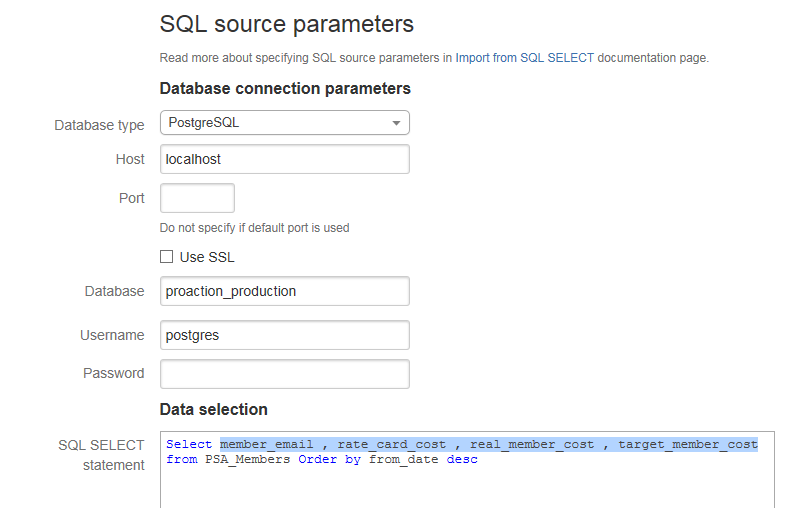
Master data is being used for BI Reporting. These reports are available under KPI Reports in Jira.

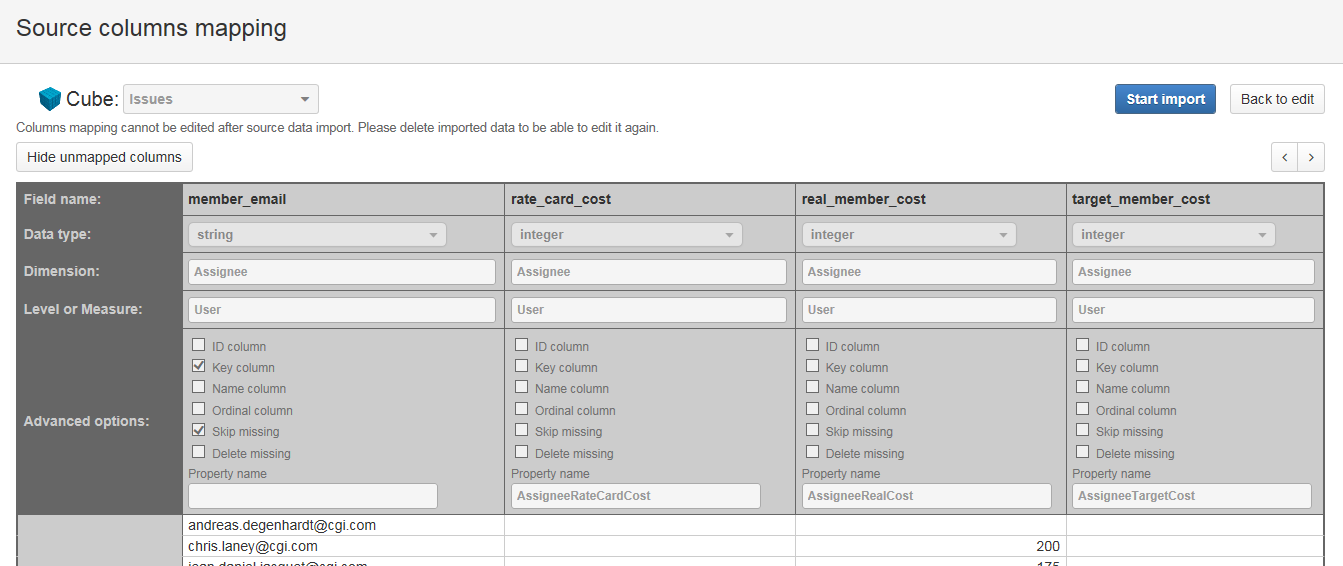
The Source data tab is used to make the database connection and queries.

E.g.



Sample report configuration:





Following tables and the fields are being used currently in various KPI Reports

|  |  |
| --- | --- |
| Master data tables | Fields |
| PSA\_Members | member\_email , rate\_card\_cost , real\_member\_cost , target\_member\_cost |
| jira\_project\_members  jira\_projects | member\_email, name |
| Kpis, kpi\_targets | Jira\_issue\_types.name, target |
| Jira\_psa\_worklog\_errors |  |

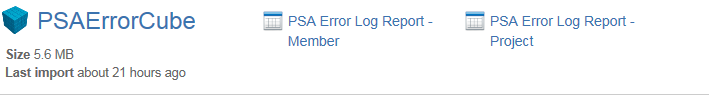
# PSA Jira-Tempo Timesheet Integration

## Process:

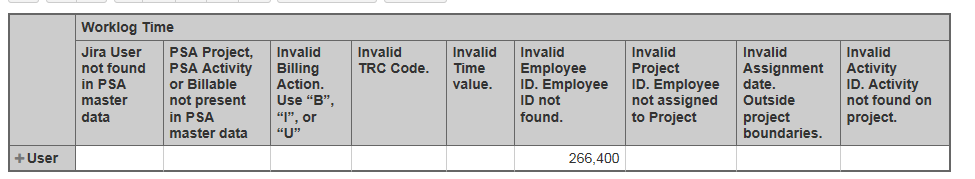
* Members log time against a particular Jira issue or Ticket. This ticket has the PSA project and activity the time should be logged to in PSA.
* After around 15 minutes, this time is imported in PSA. ◾Time bookings are periodically uploaded from JIRA to the PSA Interface Server via the PSA Time Web Service.
* When a member fills his PSA timesheet, member has to use the PSA Import functionality available to import the time in PSA.
* Once the import button is pressed, the time is pulled from Jira to PSA.
* During this import functionality , following tables in Master data database are populated:
  + jira\_psa\_worklog\_counters- This tables hold the last datetime when the import happened. Next time when the import happens, it uses this date as a start datetime for the import.
  + jira\_psa\_worklog\_error\_codes- It is the reference table for all possible errors that can happen during the import.
  + jira\_psa\_worklog\_errors - This table lists the rows that could not be imported to PSA and an error code referencing the above table to indicate the cause of the import failure.
  + jira\_psa\_worklog\_transfer- This table lists the rows which are successfully transferred to PSA
* The tables are not being truncated as of now. To keep the table size minimum, we can think of deleting entries after a particular time frame.
* About Job: TBD

## Reporting:

There are Eazy BI reports (refer screenshot below) which enable a member to see the errors which led to the failure of PSA import. The reports are as below. Members need to take the necessary action to get them resolved.



Sample PSA Error log report- member:



# Proposed Technical Implementation (TBD)

## Tools used:

Database Server:

Microsoft SQL Server Standard Edition –

Other pre-requisites: https://msdn.microsoft.com/en-us/library/ms143506 (v=sql.110).aspx

Please note the choice also depends upon the database server that may be used for Jira Production.

Application Front End: PHP

Database deployed to Azure Cloud- Details: TBD

## Data model:

* Relationships are at application Level. Need to be changed to database level.
  + Added them in the local database. Need to be replicated.
    - Proposed Model: ( Few tables un-used)



* Psa\_client\_ id needs to be changed to int in the Contracts tables.
* Link jira\_projects and jira\_project\_members
* Most of the text fields are at nvarchar (4000) which needs to be reduced as so much is not necessary.
* Non –clustered indexes to be added on the fields used for filter/search criteria.
* Created/Updated date needs to defaulted to current date/system date
* Partition huge table like PSA Members, projects etc

## Data Migration:

* Tables that are currently being used and data would need to be migrated.
* Following tables would need data migration; they have data in it.
  + assessment\_statuses
  + bus
  + business\_processes - only 3 entries. Not used.
  + cgi\_company\_codes
  + contracts
  + countries -- not used in master data application but data is populated.
  + crono\_jobs
  + departments
  + exchange\_rates- not used in master data application but data is populated.
  + industries
  + jira\_instances
  + jira\_issue\_complexities -- not used in master data application but data is populated.
  + jira\_issue\_types
  + jira\_project\_members
  + jira\_project\_psa\_projects- bridge table
  + jira\_projects
  + jira\_psa\_worklog\_counters
  + jira\_psa\_worklog\_error\_codes
  + jira\_psa\_worklog\_errors
  + jira\_psa\_worklog\_transfer
  + kpis
  + member\_levels
  + psa\_activities
  + psa\_billable\_types
  + psa\_clients
  + psa\_currencies
  + psa\_locations
  + psa\_members
  + psa\_projects
  + psa\_statuses
  + revision\_reasons -- only 4 entries. Not used
  + sbus
  + schema\_migrations – what is this table for?
  + skill\_categories
  + skill\_levels
  + skill\_sub\_categories
  + skills
  + sub\_bus
  + users

## PSA Information:

PSA Information is currently obtained via excel and either manually added or uploaded. Some of the data was uploaded through backend.

**Proposed process:**

Following tables would need data from PSA Interfaces.



**Proposal** : Import through the PHP crontab automated jobs

* Need to check how the data can be obtained from PSA. Interface to be created.
  + Excel sheets
  + Secure process to copy the files need to be identified
* The excel sheets should contain changed or updated data.
* The job would look for the files and will upload them to the database tables. The dependencies as indicated in the section below needs to be followed. If any of the relationships are not satisfied the entry should be captured in the log and added later.
* Frequency will need to be decided ( Once in a day)

The sample sheets are received as follows:

* Department -
* Activity List -
* Member list-

****

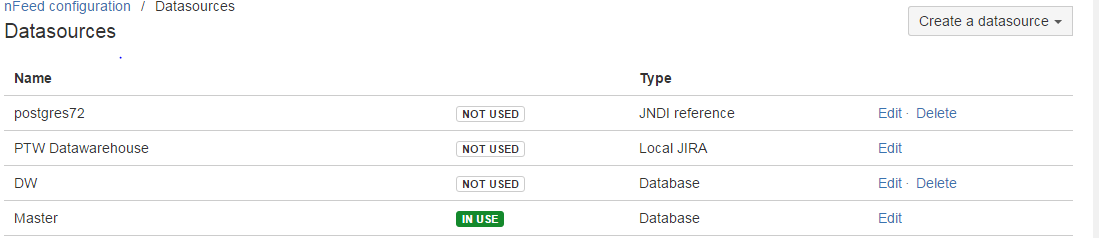
**Order in which the tables have to be populated based on their dependencies**

* Organization details
* Lookup/ Reference Tables – PSA Billable type to move to Lookup menu
* Data tables
  1. PSA Clients
  2. Contracts
  3. PSA Members
  4. Jira Projects
  5. PSA Projects
  6. Jira\_project\_psa\_projects
  7. PSA Activities
  8. PSA Teams- Derived table
  9. Jira Project members
  10. Jira Tempo Teams
  11. Jira users
  12. Jira\_users\_jira\_tempo\_teams

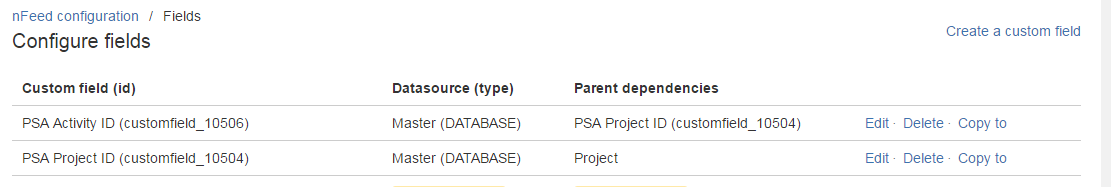
## Jira Integration:

**From Master data to Jira:**

* The Jira information is sent via an interface developed using Nfeed- a REST API client for Jira.
  + Currently PSA Project Code and PSA Activity Code is sent from PSA to Jira via nfeed.
  + This PSA project code and PSA activity code is available for use in Jira.
* The nFeed configuration is done as below:
  + <http://ptw-dev.cloudapp.net/jira/nfeed/admin/pages/datasources>. The Master is setup to connect to the Master database. Currently the project and the database is setup on the same server.

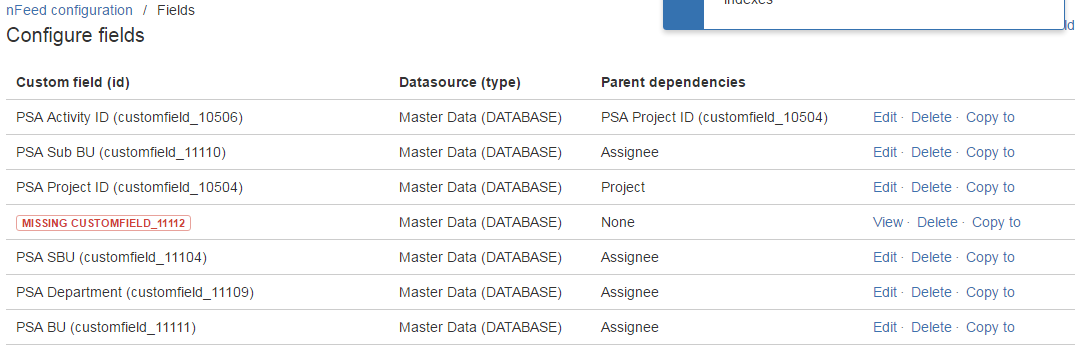


Field Configuration:



Edit to get the SQL query to fetch data from Master.

Apart from PSA Project ID and PSA activity ID, there are additional fields in the build envirionment as below. These fields are org related fields and are used in POC done at France – Oracle BI Solution to be used for reporting at the org level. These is under development phase. (Input from Jean- Daniel)



* Steps to add a custom field:
  + Create a custom field-
    - The add custom field
    - On select a field type, goto advanced tab
    - Select nfeed
    - Enter a name and description. This appears on the sce screen where the custom field appears.
    - Create
  + Configure the custom field, add the tables and query, dependency field if any etc and save it
  + The field is available for use.

**From Jira to Master data:**

The data from Jira is updated in master data using an API. The program that is currently used is jira\_api.rb which has all the details.

The config file to be used is proadmin.env where the configuration details are stored and is to used when starting the master data application.

On git bash, following command have to be used- source proadmin.env; bundle exec rails server

When starting the master data application

**Jira projects**: There is an **Update Projects link** on Jira Projects screen which is used to pull the data from i.e. project information from Jira. It internally uses the API connection to get the information.

**Jira Tempo Teams:** There is an **Update Teams link** on Jira Tempo Team screen which is used to pull the data from i.e. project information from Jira. It internally uses the API connection to get the code.

**Jira\_project\_members:**

This table is currently updated with the information from Jira every 15 minutes. There is crono job configured in \config\ cronotab.rb that calls -\app\jobs\jira\_permissions\_job.rb every 15 minutes to update this table in Master data. It also used the API but the process is automated.

Currently whenever the job runs, the entire table-is refreshed.

In the new design the Jira project members will have dependencies on the Jira projects table, so it cannot be refreshed entirely. The new entries need to be inserted and old entries to be updated.

We can have all Jira updates to be done via job that runs 15 minutes. The following order have to be maintained:

* + Jira Projects and Jira Users
  + Jira Project Members
  + Jira Tempo Teams

## Deployment:

The Jira instance and the corresponding master data is setup in the same cloud environment.

Azure Subscription Name: ProAction Team Workbench

Virtual Network: ptw

**Environments and urls:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Dev | Build | Live |
| Jira | <http://ptw-dev.cloudapp.net/jira/> | <http://ptw-build2.cloudapp.net/jira/> | <https://proaction-global.cgi.com/>jira/ |
| Master data | <http://ptw-dev.cloudapp.net/admin/> | <http://ptw-build2.cloudapp.net/admin/> | <https://proaction-global.cgi.com/>admin/ |

Proposed Environment: TBD

1. Dev:
2. Build:
3. Production:

# Ruby & Rails to PHP Migration:

There is no tool available to migrate the Ruby and Rails application to PHP directly. So we would need to rewrite the code using a PHP Framework.

About the PHP Framework to be used:

Laravel is the top 1st framework in PHP framework as per the survey of 2015.

Laravel is yet another brilliant PHP Framework that’s equipped with tons of interesting features including RESTful routing, native PHP or light weight tempting engine and many more. Built using several Symfony components, Laravel offers you web application an amazing foundation of reliable and well-tested code. Some other interesting features of Laravel include a powerful queue library, an amazing ORM, painless routing and a simple authentication.

* Restful routing that manages all the client/server routes easily and connects resources smoothly.
* Inherent Database Version control
* A lightweight Blade Templating Engine that is powerfully driven by template inheritance of files having .blade.php extension. This comes under the View part of the robust MVC architecture of Laravel.
* Composer – An amazing tool that lets you manage your application’s third party packages easily.
* Comes bundled with Eloquent – An ORM that offers a beautifully simple Active Record implementation to work with your databases and is relatively faster than all other PHP frameworks
* Built-in unit testing and simply readable impressive syntax
* Larger Community catering to thousands of programming geeks and application developers
* Intelligently designed to offer incredible flexibility to developers that help them create each and everything from small sites to giant enterprise applications.

Layout Design

--------------

* HTML: HTML5
* CSS: CSS3, Bootstrap (for mobile friendly design)
* JS:  jQuery (UI interaction), datatables, EasyUI, Highcharts
* Template : <https://themeforest.net/category/site-templates/admin-templates?page=3>
  + - : (Clip Two Template - Bootstrap with Angular JS)

Appendix:

About Schema migration table- This table is being used as version control for database. If there is any change in the database a date stamp entry is created indicating a change in the database.