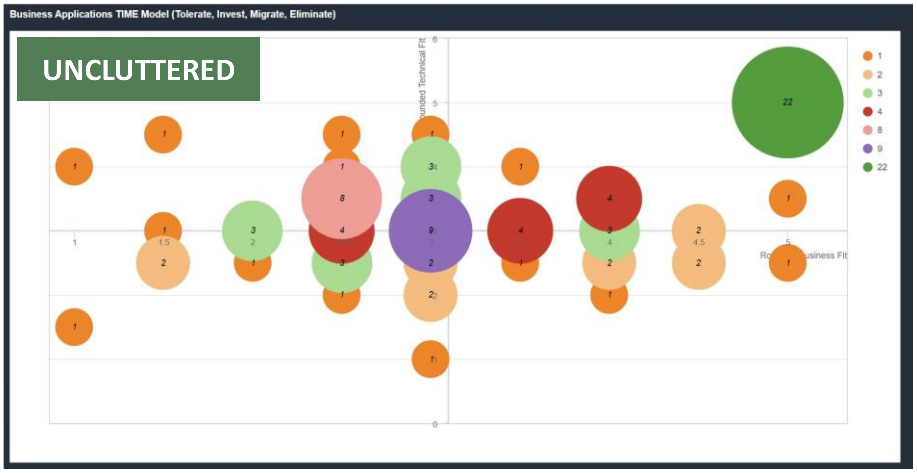
LABNAF 6.0 **TIME MODEL GENERATION**

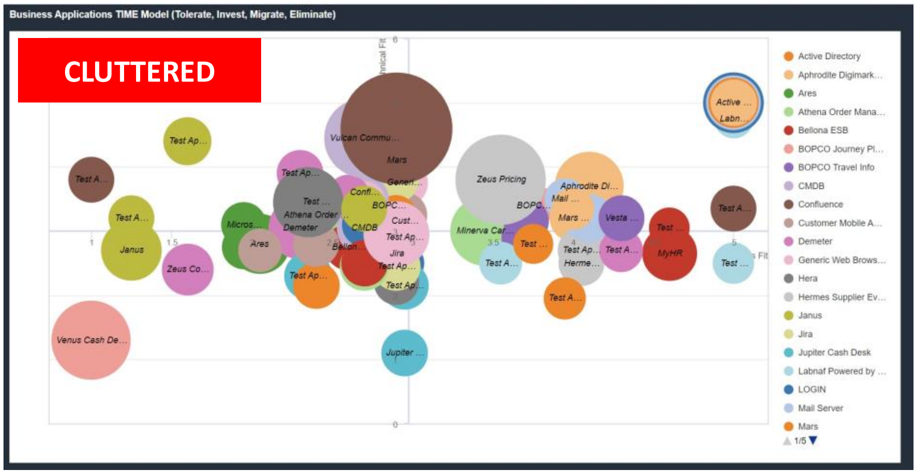
CONFIGURATION GUIDE

CONTENTS

OVERVIEW

Application TIME Models





Legacy applications frequently demand more attention than what the budget allows. The Gartner TIME framework (Tolerate, Invest, Migrate, Eliminate) stands as a renowned industry method for strategizing and fine-tuning application portfolio enhancements.

This strategy provides a comprehensive avenue to optimize both the business and technological aspects of an application portfolio, ensuring each application aligns well with the organization's requirements.

To facilitate this, application leaders often resort to the TIME categorization, visualized as a bubble chart for their applications. Clicking on these bubbles reveals in-depth details about the respective applications. This visualization aids in prioritizing portfolio improvements by considering factors such as business and technological suitability, associated risks, and costs.

Labnaf's approach delves deeper by evaluating the business and technological fit grounded on specific criteria. This includes, but is not limited to, aspects like business satisfaction, potential, technical maintainability, and scalability. The flexibility of this system allows for criteria to be seamlessly added, deleted, renamed, or weighted as needed, with the resulting fits being recalculated in real-time.

Uncluttering TIME models

As the quantity of applications in our portfolio expands, the clarity of the TIME models becomes compromised, rendering them less functional and more chaotic. In the TIME model depicted at the bottom-right, each bubble symbolizes an individual application. Even with just a few scores of applications displayed on the second chart, it's evident how readability is quickly challenged due to overlapping bubbles.

Conversely, in the top-right TIME model, every bubble stands for a cluster of applications. By clicking on any given bubble, you can delve deeper into that group. To maintain clarity, Labnaf organizes applications with analogous business and technological alignments into specific groups. This level of organization can be readily adjusted to ensure optimal visibility of the TIME models.

Should there be a need for multiple TIME models, perhaps domain-specific, distinct grouping criteria can be designated for each, contingent on the number of applications associated with each domain.

Implementation Overview

The uncluttered TIME model calculation and generation is implemented using a low code Labnaf Power Script that you can run either once, or that you can schedule following your preferences. Labnaf Power Scripts are part of the Labnaf PowerShell environment.

By default, the process runs during the day. It periodically recalculates the business fit and technical fit, refreshes the uncluttered TIME models, and finally stops in the evening.

The detailed application evaluation criteria, like business satisfaction, potential, technical maintainability, and scalability, can be easily and quickly adapted and extended. Same for the calculation of the business fit and technical fit.

Example: Business Fit = = Bus_Data*25/100 + Bus_Needs*30/100 + Bus_Potential*15/100 + Bus_Satisfaction*30/100

You can generate multiple TIME models, for different application categories, and following different grouping criteria that can be easily configured.

The solution comes with Prolaborate widget specifications: TIME Model bubble chart, landscape chart and report.

PREREQUISITES FOR USING THE LABNAF TIME MODEL GENERATION

INSTALLED SOFTWARE

The following software should be installed on your server:

- Sparx Enterprise Architect (Corporate Edition minimum)
- Prolaborate
- Labnaf PowerShell version 6.03 or later.

The following software should be installed on your workstation:

- Labnaf AddIn For Sparx EA version 6.03 or later.

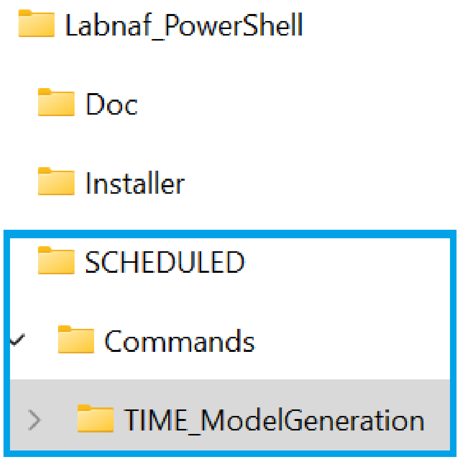
REFERENCE DOCUMENTATION

Labnaf PowerShell installation documentation.

Labnaf PowerShell User Guide.

Labnaf PowerShell Reference Guide.

LABNAF POWERSHELL PACKAGE - TIME MODEL GENERATION CONTENT



The Labnaf PowerShell package encompasses a structured hierarchy of folders dedicated to TIME Model generation:

- **SCHEDULED** Folder: This holds the commands and parameters necessary to set scheduled intervals for TIME Model generation, such as from 8:00 to 20:00.

- **Commands** Folder: It contains a specific command to initiate a one-time generation of all TIME models.

- **TIME_ModelGeneration** Folder: Within this folder, you will find:
  - The comprehensive TIME models generation script.
  - Parameters that specify the range of applications for the upcoming TIME models, the TIME model packages that should be populated in the model repository, the property names to utilize, and the rounding specifics for the Business Fit and Technology Fit values.
  - A sample Prolaborate bubble chart configuration applicable to TIME Models.
  - A test command that provides the flexibility to generate TIME Models either once or in a recurring manner, like a restart every 5 minutes.
  - Documentation, including the present document.

STEP 1: CONFIGURE THE REPOSITORY

DEFAULT APPLICATION EVALUATION PROPERTIES AND CALCULATIONS - OVERVIEW

The Labnaf sample and startup repositories features some default application evaluation properties and calculations.

The following two properties are calculated based on the detailed properties bearing a name starting with “z_” (like “zoom into detailed properties”). In case the properties are being calculated outside of the repository, they can be entered or imported directly in the repository.

- Eval_Business_Fit
- Eval_Technical_Fit

Here are the detailed properties used to calculate Eval_Business_Fit and Eval_Technical_Fit:

- z_Bus_Needs
- z_Bus_Criticality
- z_Bus_Data
- z_Bus_Potential
- z_Bus_Satisfaction

- z_Tech_Skills_Availability
- z_Tech_Maintainability
- z_Tech_Provider_Support
- z_Tech_Archi_Alignmt
- z_Tech_Stability
- z_Tech_Security
- z_Tech_Scalability

The following calculations are provided as default.

Eval_Business_Fit

=z_Bus_Criticality*0/100
+z_Bus_Data*25/100
+z_Bus_Needs*30/100
+z_Bus_Potential*15/100
+z_Bus_Satisfaction*30/100

Eval_Technical_Fit

=z_Tech_Skills_Availability*0.15
+z_Tech_Maintainability*0.15
+z_Tech_Provider_Support*0.15
+z_Tech_Archi_Alignmt*0.10
+z_Tech_Stability*0.15
+z_Tech_Security*0.20
+z_Tech_Scalability*0.10

RENAME THE DEFAULT PROPERTY NAMES (IF NEEDED)

You will find, in the **TIME_ModelGeneration** folder, a set of configuration and script files used by TIME Model Generation solution.

Within this folder, the sub-folder '**RenameProperties'** contains a command to

- copy all files with the extensions .TXT, .LPSC (Labnaf Power Script file), and .XML,
- and then rename the property names within all the file copies.

The command file '**_RenameProperties.cmd**' specifies the original and updated property names.

So, this is the file where you can adapt the property names.

To rename properties, please proceed as follows:

- Backup the **TIME_ModelGeneration** folder
- Go to the **TIME_ModelGeneration\RenameProperties** folder.
- Edit the **_RenameProperties.cmd** to specify how each property should be renamed.
- Run the command **CopyFilesAndRenameProperties.cmd**

Once the file copies have been updated, you can replace the original files with the copies.

- Goto to the **TIME_ModelGeneration\RenameProperties\Files** folder
- Copy the translated files back to their original folders.

CREATE THE EVALUATION PROPERTY TYPES (TAGGED VALUE TYPES) IN THE REPOSITORY

Any inexistent property is automatically created as soon as a property value is imported.

If you are **only** importing the application evaluation property values, i.e. you are never editing this values other in Sparx EA or in Prolaborate, then you don't need to define, neither these properties, nor their types. Indeed, these properties will be automatically created or updated when their values are imported.

The tagged value definition file (.XML) is one of the files that get automatically updated by the "renaming properties" command.

You can either create tagged value types by hand, or you can import the tagged value type definition file (.XML) as described in the Sparx EA documentation.

To import the tagged value definition file (after property type names have been renamed, if necessary):

Select the option **Settings > Transfer > Import Reference Data**

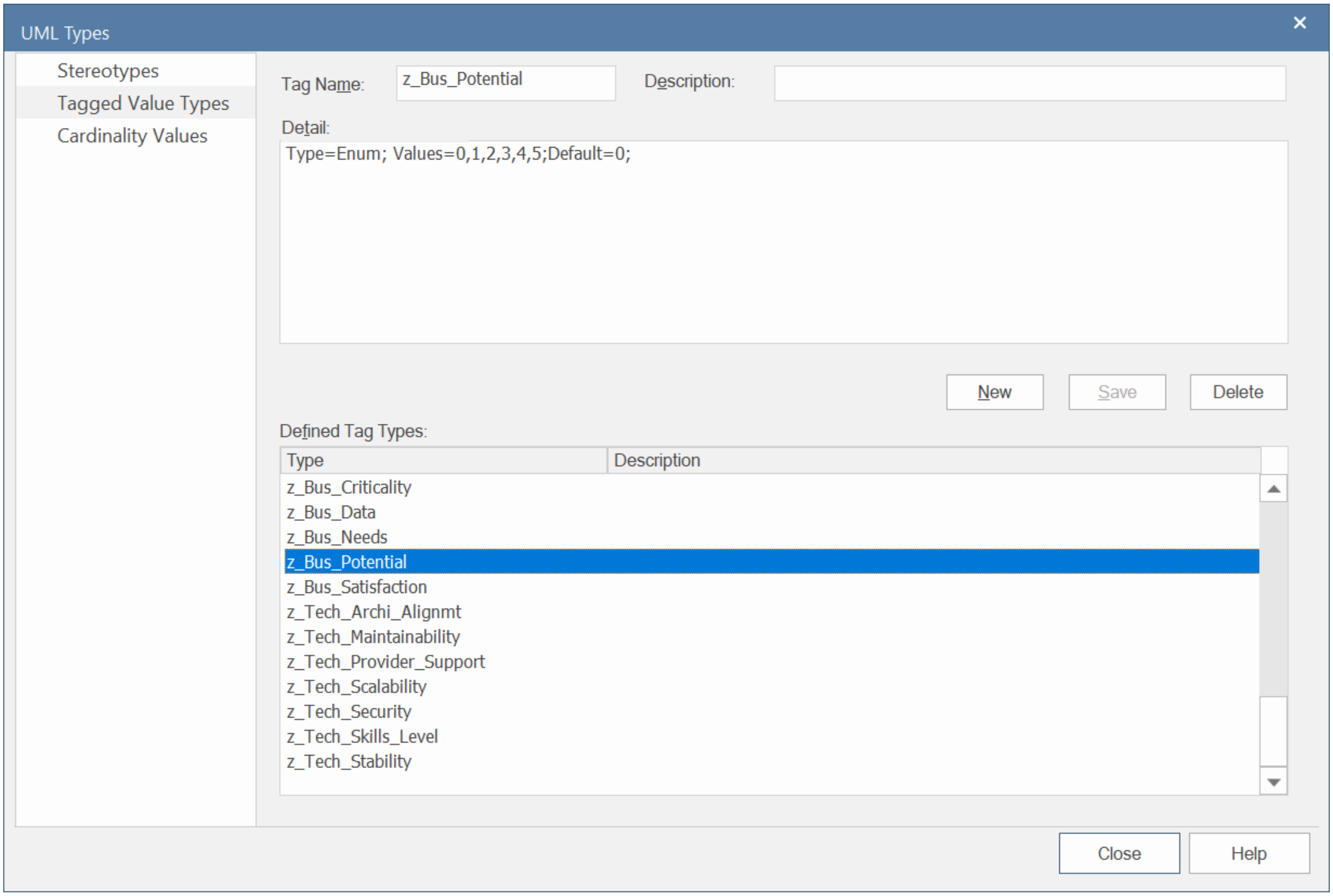
Select the file **'TIME_Model_TaggedValueTypes.xml'**

Select '**Tagged Value Types**'.

Press '**Import'**.

Once the tagged value types have been set, any element property named as a tagged value type will get the constraints defined in the tagged value type.

You can visualize, add, delete and rename tagged value types using this user interface.



Tag Name: z_Bus_Potential

Description:

Detail:

```
Type=Enum; Values=0,1,2,3,4,5;Default=0;
```

Defined Tag Types:

Type	Description
z_Bus_Criticality	
z_Bus_Data	
z_Bus_Needs	
z_Bus_Potential	
z_Bus_Satisfaction	
z_Tech_Archi_Alignmt	
z_Tech_Maintainability	
z_Tech_Provider_Support	
z_Tech_Scalability	
z_Tech_Security	
z_Tech_Skills_Level	
z_Tech_Stability	

Here are the predefined format details (like “Enum”) defined in Sparx EA.

If your initial startup repository version is version 6.0.3 or above: These tagged value types are already defined. You can rename them, delete some, or add more, if you wish, using the above user interface.

CREATE APPLICATION EVALUATION PROPERTIES

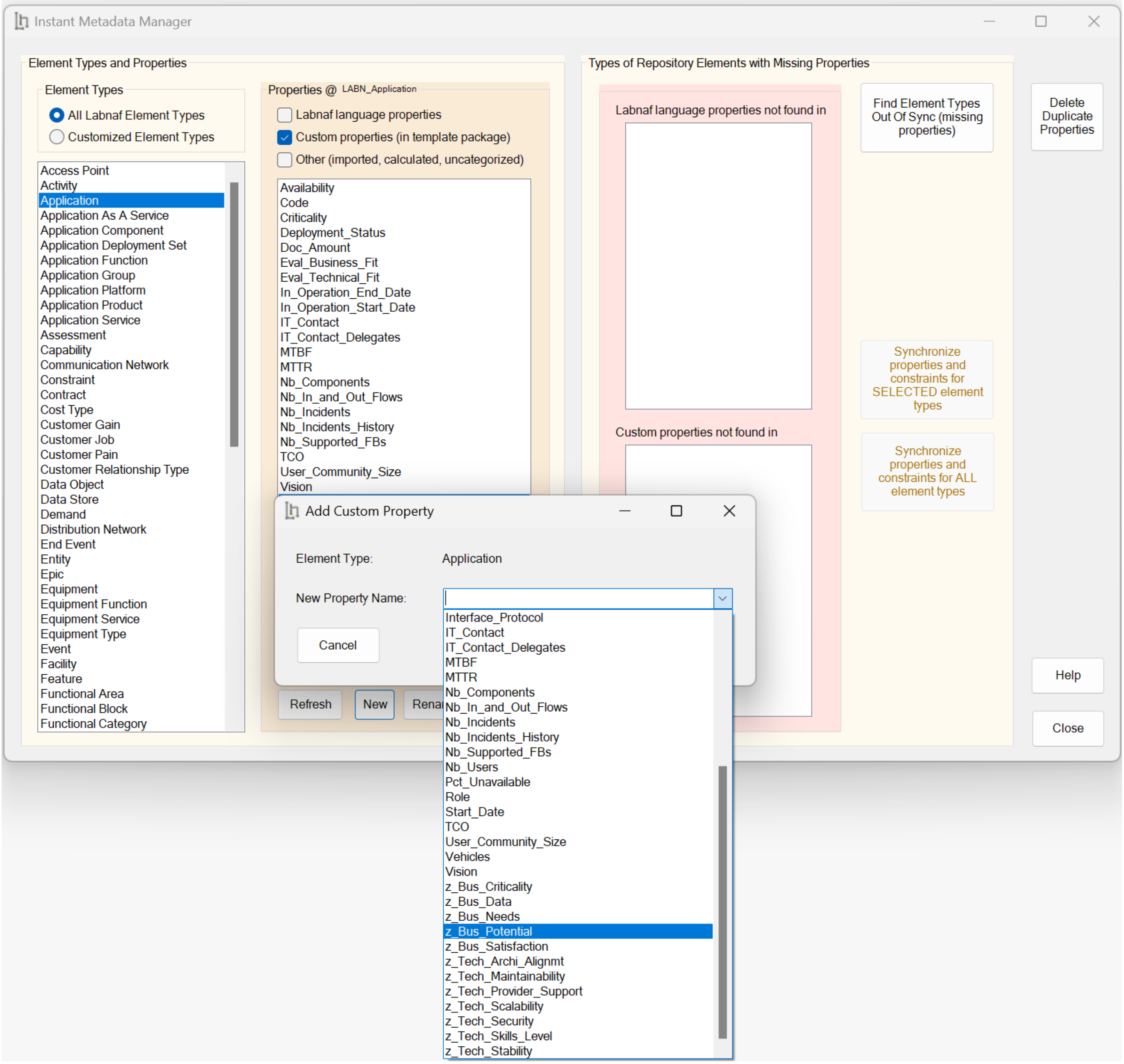
See also: **Any inexistent property is automatically created as soon as a property value is imported.**

Once the tagged value types have been defined, you can create the custom properties (tagged values).

To do so, use the Labnaf Instant Metadata Manager.

You will notice that, when you add a new property, you can select its name from the drop-down list.

The dropped-down list is populated from the list of tagged value types. If you select one of these named types, a custom property, with that name and type/format, will be added to the selected element type i.e. in this case 'Application'.



Element Types:
● All Labnaf Element Types
○ Customized Element Types

Properties @ LABN_Application:
□ Labnaf language properties
☑ Custom properties (in template package)
□ Other (imported, calculated, uncategorized)

Add Custom Property

Element Type: Application

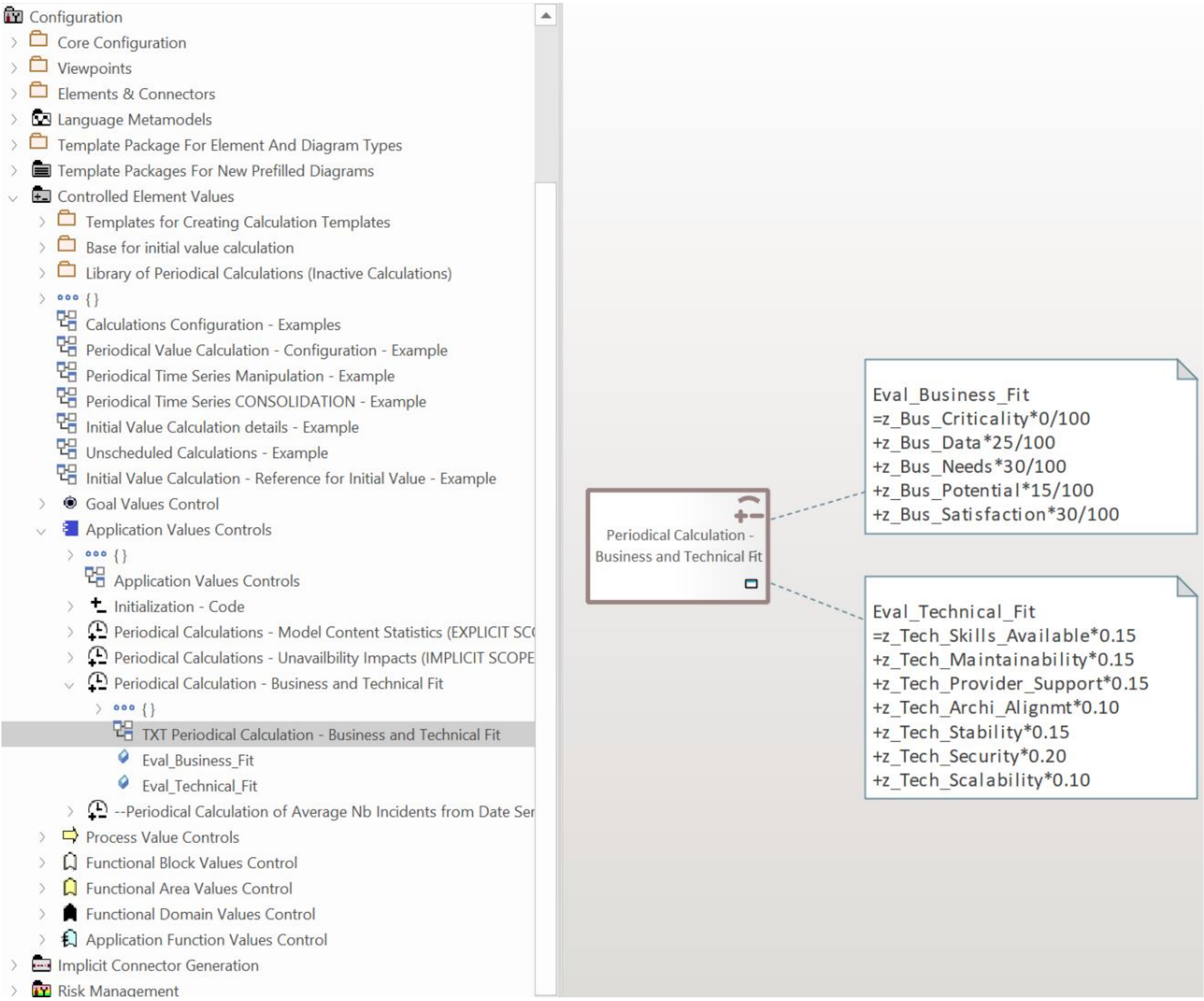
New Property Name: (selected: z_Bus_Potential)

If your initial startup repository version is version 6.0.3 or above: These custom properties are already defined. You can rename them, if you wish, using the above user interface.

CREATE VALUE CALCULATIONS

You can learn about configuring and testing the many value calculations capabilities by reading the Calculation documentation.

In the present case, the configuration is quite simple. You can see below the default configuration for the calculation of the Business Fit and Technical Fit as provided with the startup and sample repositories (version 6.03 or above).



- Configuration
  - Core Configuration
  - Viewpoints
  - Elements & Connectors
  - Language Metamodels
  - Template Package For Element And Diagram Types
  - Template Packages For New Prefilled Diagrams
  - Controlled Element Values
    - Templates for Creating Calculation Templates
    - Base for initial value calculation
    - Library of Periodical Calculations (Inactive Calculations)
    - { }
    - Calculations Configuration - Examples
    - Periodical Value Calculation - Configuration - Example
    - Periodical Time Series Manipulation - Example
    - Periodical Time Series CONSOLIDATION - Example
    - Initial Value Calculation details - Example
    - Unscheduled Calculations - Example
    - Initial Value Calculation - Reference for Initial Value - Example
    - Goal Values Control
    - Application Values Controls
      - { }
      - Application Values Controls
      - Initialization - Code
      - Periodical Calculations - Model Content Statistics (EXPLICIT SC(
      - Periodical Calculations - Unavailbility Impacts (IMPLICIT SCOPE
      - Periodical Calculation - Business and Technical Fit
        - { }
        - TXT Periodical Calculation - Business and Technical Fit
        - Eval_Business_Fit
        - Eval_Technical_Fit
      - --Periodical Calculation of Average Nb Incidents from Date Ser
    - Process Value Controls
    - Functional Block Values Control
    - Functional Area Values Control
    - Functional Domain Values Control
    - Application Function Values Control
  - Implicit Connector Generation
  - Risk Management

Periodical Calculation - Business and Technical Fit

```
Eval_Business_Fit
=z_Bus_Criticality*0/100
+z_Bus_Data*25/100
+z_Bus_Needs*30/100
+z_Bus_Potential*15/100
+z_Bus_Satisfaction*30/100
```

```
Eval_Technical_Fit
=z_Tech_Skills_Available*0.15
+z_Tech_Maintainability*0.15
+z_Tech_Provider_Support*0.15
+z_Tech_Archi_Alignmt*0.10
+z_Tech_Stability*0.15
+z_Tech_Security*0.20
+z_Tech_Scalability*0.10
```

If your initial startup repository version is version 6.0.3 or above: These calculations are already defined. You can rename them and change the calculations, if you wish.

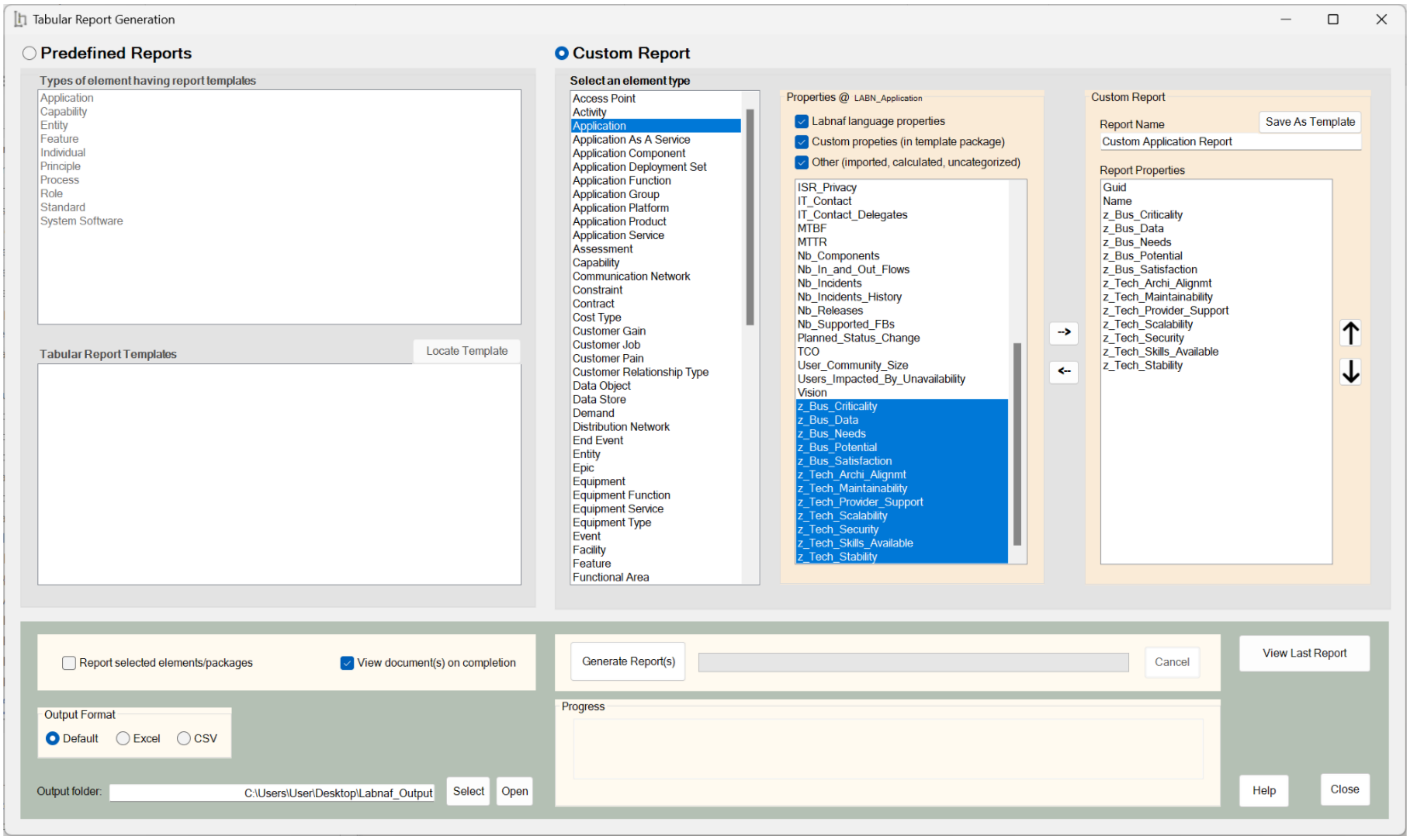
If you don’t want to create these calculation from scratch, you can

- copy the element from the startup or sample repository (full structure for duplication)
- paste into a package to your own repository
- move the ‘**Periodical Calculation’** element into the application value controls element.
- and rename the properties as required.

Once the calculations are configured, the Labnaf PowerShell will run these periodically following the configured schedule.

SET APPLICATION EVALUATION VALUES

The Labnaf report generation features that you need here are very easy to use. You can learn more by reading the ['Tabular Report Generation' guidance](#).



Tabular Report Generation

○ Predefined Reports

Types of element having report templates

Application
Capability
Entity
Feature
Individual
Principle
Process
Role
Standard
System Software

Tabular Report Templates — Locate Template

● Custom Report

Select an element type

Access Point
Activity
Application
Application As A Service
Application Component
Application Deployment Set
Application Function
Application Group
Application Platform
Application Product
Application Service
Assessment
Capability
Communication Network
Constraint
Contract
Cost Type
Customer Gain
Customer Job
Customer Pain
Customer Relationship Type
Data Object
Data Store
Demand
Distribution Network
End Event
Entity
Epic
Equipment
Equipment Function
Equipment Service
Equipment Type
Event
Facility
Feature
Functional Area

Properties @ LABN_Application

☑ Labnaf language properties
☑ Custom propeties (in template package)
☑ Other (imported, calculated, uncategorized)

ISR_Privacy
IT_Contact
IT_Contact_Delegates
MTBF
MTTR
Nb_Components
Nb_In_and_Out_Flows
Nb_Incidents
Nb_Incidents_History
Nb_Releases
Nb_Supported_FBs
Planned_Status_Change
TCO
User_Community_Size
Users_Impacted_By_Unavailability
Vision
z_Bus_Criticality
z_Bus_Data
z_Bus_Needs
z_Bus_Potential
z_Bus_Satisfaction
z_Tech_Archi_Alignmt
z_Tech_Maintainability
z_Tech_Provider_Support
z_Tech_Scalability
z_Tech_Security
z_Tech_Skills_Available
z_Tech_Stability

Custom Report

Report Name: Custom Application Report — Save As Template

Report Properties

Guid
Name
z_Bus_Criticality
z_Bus_Data
z_Bus_Needs
z_Bus_Potential
z_Bus_Satisfaction
z_Tech_Archi_Alignmt
z_Tech_Maintainability
z_Tech_Provider_Support
z_Tech_Scalability
z_Tech_Security
z_Tech_Skills_Available
z_Tech_Stability

☐ Report selected elements/packages ☑ View document(s) on completion

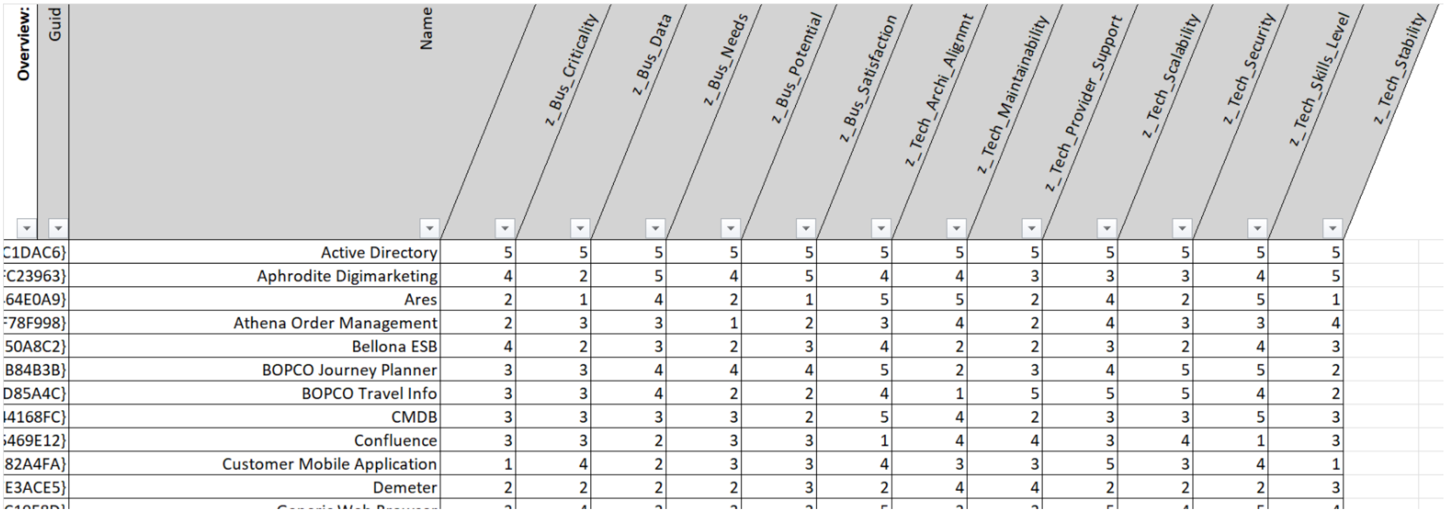
Output Format: ● Default ○ Excel ○ CSV

Output folder: C:\Users\User\Desktop\Labnaf_Output — Select — Open

Generate Report(s) — Cancel — View Last Report

Progress

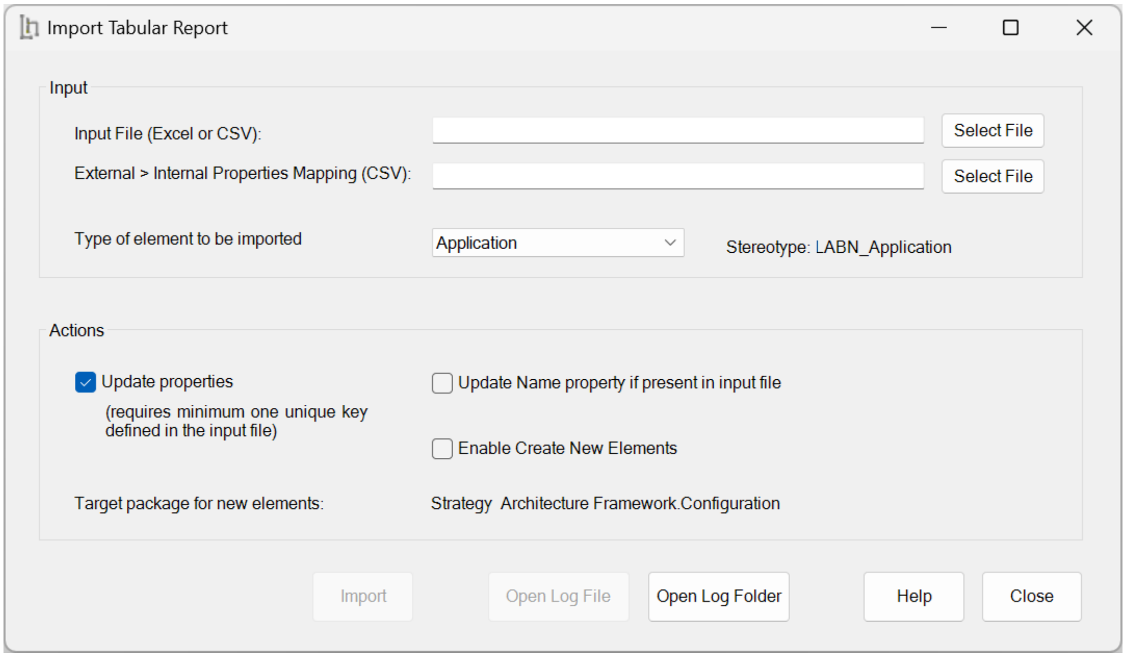
Help — Close



Overview:

Guid	Name	z_Bus_Criticality	z_Bus_Data	z_Bus_Needs	z_Bus_Potential	z_Bus_Satisfaction	z_Tech_Archi_Alignmt	z_Tech_Maintainability	z_Tech_Provider_Support	z_Tech_Scalability	z_Tech_Security	z_Tech_Skills_Level	z_Tech_Stability
C1DAC6}	Active Directory	5	5	5	5	5	5	5	5	5	5	5	5
C23963}	Aphrodite Digimarketing	4	2	5	4	5	4	4	3	3	3	4	5
64E0A9}	Ares	2	1	4	2	1	5	5	2	4	2	5	1
78F998}	Athena Order Management	2	3	3	1	2	3	4	2	4	3	3	4
50A8C2}	Bellona ESB	4	2	3	2	3	4	2	2	3	2	4	3
B84B3B}	BOPCO Journey Planner	3	3	4	4	4	5	2	3	4	5	5	2
D85A4C}	BOPCO Travel Info	3	3	4	2	2	4	1	5	5	5	4	2
4168FC}	CMDB	3	3	3	3	2	5	4	2	3	3	5	3
469E12}	Confluence	3	3	2	3	3	1	4	4	3	4	1	3
82A4FA}	Customer Mobile Application	1	4	2	3	3	4	3	3	5	3	4	1
E3ACE5}	Demeter	2	2	2	2	3	2	4	4	2	2	2	3

Once the report is generated, you can remove the first grouping column if you wish, then enter new values and finally use the Labnaf "[Import Tabular Report](#)" feature to load the updated information.



Import Tabular Report

Input

Input File (Excel or CSV):

External > Internal Properties Mapping (CSV):

Type of element to be imported: Application

Stereotype: LABN_Application

Actions

☑ Update properties (requires minimum one unique key defined in the input file)

☐ Update Name property if present in input file

☐ Enable Create New Elements

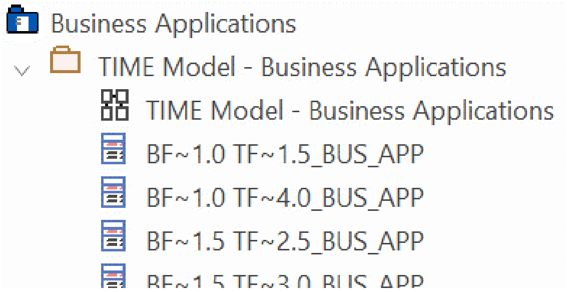
Target package for new elements: Strategy Architecture Framework.Configuration

CREATE TIME MODEL PACKAGES TO HOST APPLICATION GROUPS

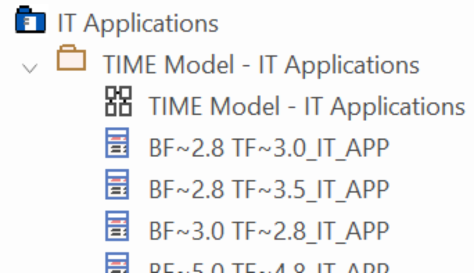
In the repository, create one TIME Model package per required TIME Model chart.

As we will see further in this document, the GUIDs of each application catalog package and its related TIME model package will need to be declared in the script that updates TIME models.

The content of each TIME model package is presented using its own Prolaborate bubble chart.



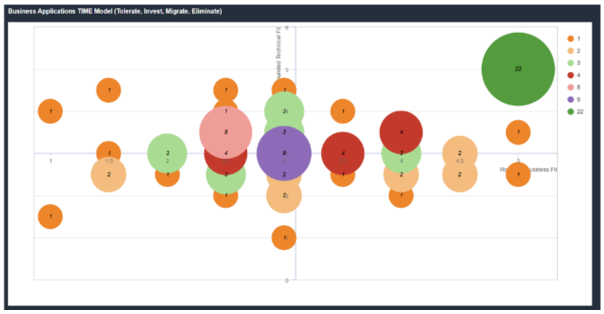
Business Applications
- TIME Model - Business Applications
  - TIME Model - Business Applications
  - BF~1.0 TF~1.5_BUS_APP
  - BF~1.0 TF~4.0_BUS_APP
  - BF~1.5 TF~2.5_BUS_APP
  - BF~1.5 TF~3.0_BUS_APP

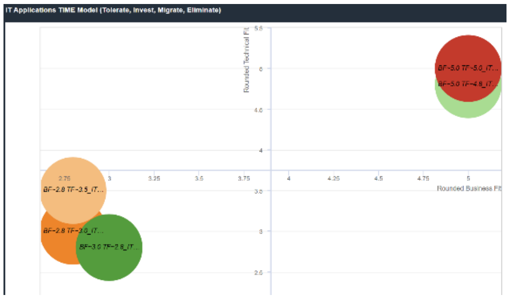


IT Applications
- TIME Model - IT Applications
  - TIME Model - IT Applications
  - BF~2.8 TF~3.0_IT_APP
  - BF~2.8 TF~3.5_IT_APP
  - BF~3.0 TF~2.8_IT_APP
  - BF~5.0 TF~4.8_IT_APP

Bubble can show the number applications per group …

… or bubble can show the average Business Fit and Technical Fit





STEP 2: CONFIGURE THE LABNAF POWERSHELL

CONFIGURE THE LABNAF POWER SCRIPT PERFORMING TIME MODEL GENERATION

In the ‘**TIME_ModelGeneration’** folder, you will find the TIME model generation script called **‘Update_TIME_Models.lpsc’.**

You can adjust the content of the script to define

- what TIME Models you want to generate
- what level of uncluttering you want to reach.

By default, the script file contains values that are applicable only to the Labnaf Sample repository.

As described below, **you definitely need to adapt this script to your needs.**

Below is the default content of that script file…

- **The to- be-set variable values are highlighted in red.**
- After setting the variable values, the TIME Model data generation commands (highlighted in blue) remain consistent across all application group sets.

```
// -- WARNING: FIELD NAME SETTINGS
// To avoid weird field name generation, please make sure that, in the script,
// each declared field name does not contain another declared field name.

// --- APPLICATION FIELD NAMES
set FIELD_Eval_Business_Fit=Eval_Business_Fit
set FIELD_Eval_Technical_Fit=Eval_Technical_Fit

// --- GENERATED APPLICATION GROUP FIELD NAMES
set FIELD_Rounded_Business_Fit=Rounded_Business_Fit
set FIELD_Rounded_Technical_Fit=Rounded_Technical_Fit
set FIELD_NumberOfApplicationsInGroup=NumberOfApplicationsInGroup
set FIELD_Application_Names=Application_Names

/* Calculate Business Fit and Technical Fit */
CalculateTaggedValues "Application Values Controls" %FIELD_Eval_Business_Fit%
CalculateTaggedValues "Application Values Controls" %FIELD_Eval_Technical_Fit%

// FOR Business and IT apps
// 1. Create new lists of app groups based on app business and tech fit
// 2. Delete existing TIME application groups from the repository
// 3. Import new list of TIME application groups

// --- SET 1: GROUPS OF BUSINESS APPLICATIONS ---
set GROUP_ROUNDING=0.5
set APPGROUP_NAME_SUFFIX=_BUS_APP
set CSV_ListOf_APP_GROUPS=%TIMEMODEL_DIR%\Data_Generated\TimeApplicationGroups_Business.csv
set PackageGUID_APPS={A7207EB5-94E0-473c-9DEE-BD285BE81BE6}
set PackageGUID_APP_GROUPS={1D6B2641-8EAE-4377-8A13-54FDE632873F}
SqlExportToCsv %TIMEMODEL_DIR%\SQL_BuildListOfTimeAppGroups.txt %CSV_ListOf_APP_GROUPS%
DeleteSelectedElements %TIMEMODEL_DIR%\SQL_SelectTimeApplicationGroups.txt
ImportTabularReport %CSV_ListOf_APP_GROUPS% - Class EnableCreate %PackageGUID_APP_GROUPS%

// --- SET 2: GROUPS OF IT APPLICATIONS ---
set GROUP_ROUNDING=0.25
set APPGROUP_NAME_SUFFIX=_IT_APP
set CSV_ListOf_APP_GROUPS=%TIMEMODEL_DIR%\Data_Generated\TimeApplicationGroups_IT.csv
set PackageGUID_APPS={0451D0DD-CB83-412b-A4EB-F2B722C356DF}
set PackageGUID_APP_GROUPS={7CD754BD-68CE-4ac6-B5D6-980EDC9A6B83}
SqlExportToCsv %TIMEMODEL_DIR%\SQL_BuildListOfTimeAppGroups.txt %CSV_ListOf_APP_GROUPS%
DeleteSelectedElements %TIMEMODEL_DIR%\SQL_SelectTimeApplicationGroups.txt
ImportTabularReport %CSV_ListOf_APP_GROUPS% - Class EnableCreate %PackageGUID_APP_GROUPS%
```

Most variable names are self-explanatory, with the following exceptions:

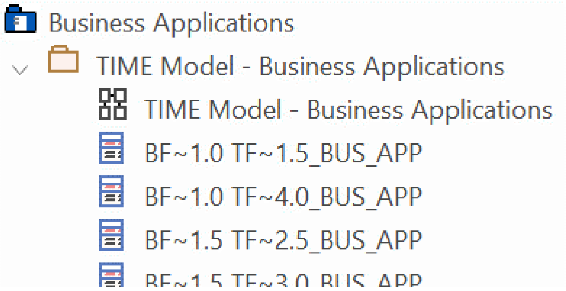
- The variable “**APPGROUP_NAME_SUFFIX”** is used to avoid the same application group name is generated for different sets of application groups. Indeed, each application group name must be unique.
- The variable “**CSV_ListOf_APP_GROUPS”** is the name of an intermediary CSV file that contains the list of generated application groups before they get imported in the repository. You could use the same intermediary file for all application group sets, but, for testing purposes, it is useful to see what the process is generating.

TEST THE CONFIGURATION

Update the connection string in the **TestRepository.EAP** file by using any text editor like NotePad.

You can then test the configuration and the generation of TIME Models by clicking on the **TIME_ModelGeneration\Test.cmd**.

If everything is properly configured, this will populate the TIME Model packages in your repository.



```
Business Applications
  TIME Model - Business Applications
    TIME Model - Business Applications
    BF~1.0 TF~1.5_BUS_APP
    BF~1.0 TF~4.0_BUS_APP
    BF~1.5 TF~2.5_BUS_APP
    BF~1.5 TF~3.0_BUS_APP
```

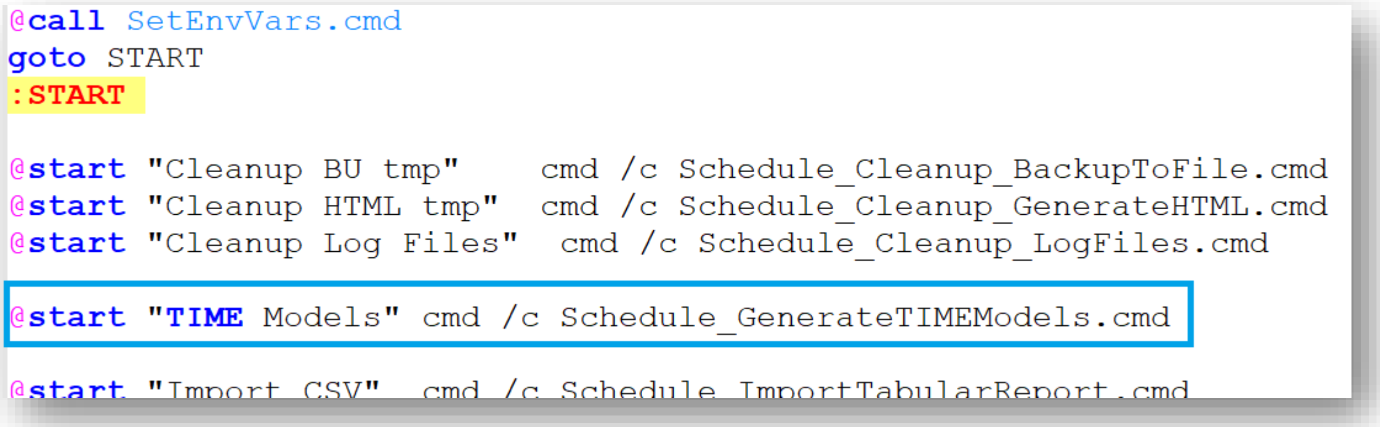
In the **SCHEDULED\Command** folder, you will find the command **GenerateTIMEModels.cmd**. This is the PRODUCTION command called by the scheduler. You can click on this command to generate TIME models just once.

SCHEDULING TIME MODEL GENERATION

The **SCHEDULED** folder contains the following commands:

_ScheduleAllMultipleThreads.cmd

This is the master scheduling command which, by default, includes TIME model (re-)generation.



```
@call SetEnvVars.cmd
goto START
:START

@start "Cleanup BU tmp"    cmd /c Schedule_Cleanup_BackupToFile.cmd
@start "Cleanup HTML tmp"  cmd /c Schedule_Cleanup_GenerateHTML.cmd
@start "Cleanup Log Files"  cmd /c Schedule_Cleanup_LogFiles.cmd

@start "TIME Models" cmd /c Schedule_GenerateTIMEModels.cmd

@start "Import CSV"  cmd /c Schedule_ImportTabularReport.cmd
```

If you prefer, you can of course replace this command with another (non-Labnaf) scheduling solution.

Schedule_GenerateTIMEModels.cmd

This command schedules the TIME model generation only.

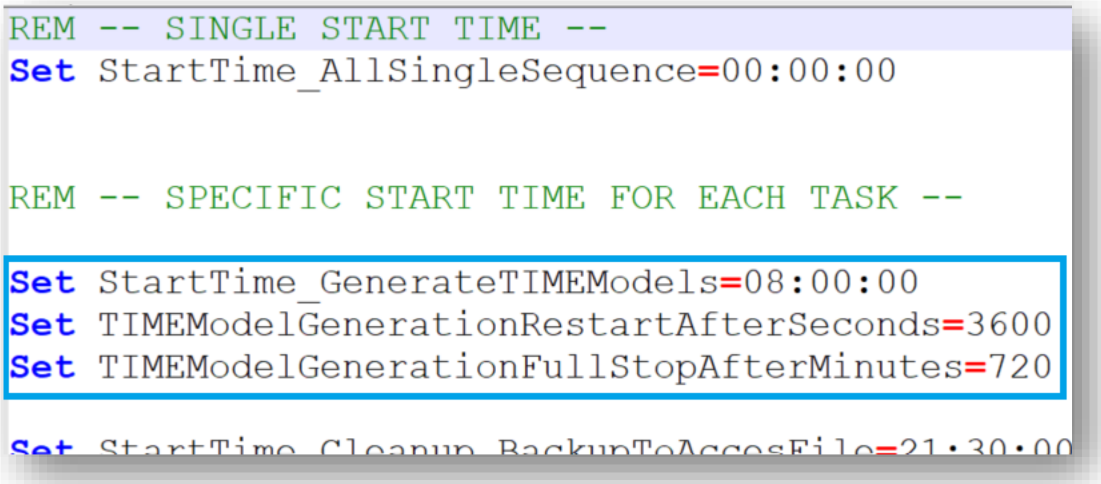
SetTimes.cmd

Adjust this command to customize your scheduling preferences if the default settings don't meet your needs.

- **StartTime_GenerateTIMEModels**: Defines when TIME model generation must start every day.
- **TIMEModelGenerationRestartAfterSeconds** : Defines after how many seconds the TIME model generation must restart.

- **TIMEModelGenerationFullStopAfterMinutes**: Defines when the TIME Model generation must stop restarting. This is expressed in minutes following the **StartTime_GenerateTIMEModels**:.

  So by default, TIME Model generation starts every morning at 8:00. During the day, it will restart every hour (3600 seconds), and it will stop restarting after 12 hours (720 minutes) i.e. at 8:00 + 12 = 20:00.



```
REM -- SINGLE START TIME --
Set StartTime_AllSingleSequence=00:00:00


REM -- SPECIFIC START TIME FOR EACH TASK --

Set StartTime_GenerateTIMEModels=08:00:00
Set TIMEModelGenerationRestartAfterSeconds=3600
Set TIMEModelGenerationFullStopAfterMinutes=720

Set StartTime_Cleanup_BackupToAccesFile=21:30:00
```

STEP 3: CONFIGURE PROLABORATE FOR TIME MODEL VISUALIZATION

In the following Prolaborate chart creation windows, the content of the Prolaborate Chart **Query** field is provided in a set of files located in the Labnaf PowerShell **TIME_ModelGeneration\Prolaborate** folder.

The default property names used in these SQL statements can be automatically renamed following your preferences.

To learn more about renaming default property names, please read the section Rename the Default Property Names.

CREATE A TIME MODEL BUBBLE CHART

Please, proceed as described in the following Prolaborate chart creation windows.

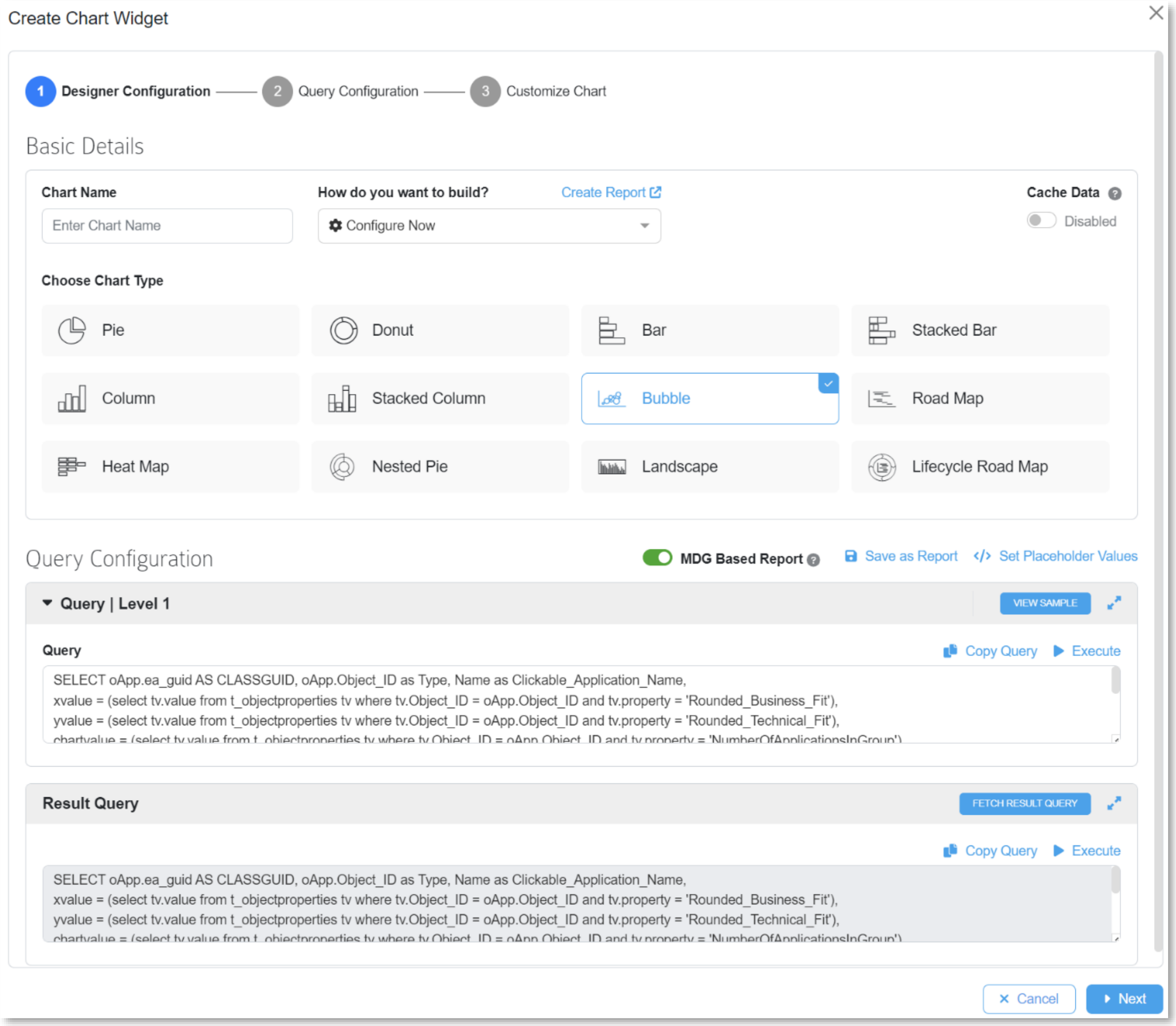


Chart type: Bubble

MDG Based Report: enabled

Query:

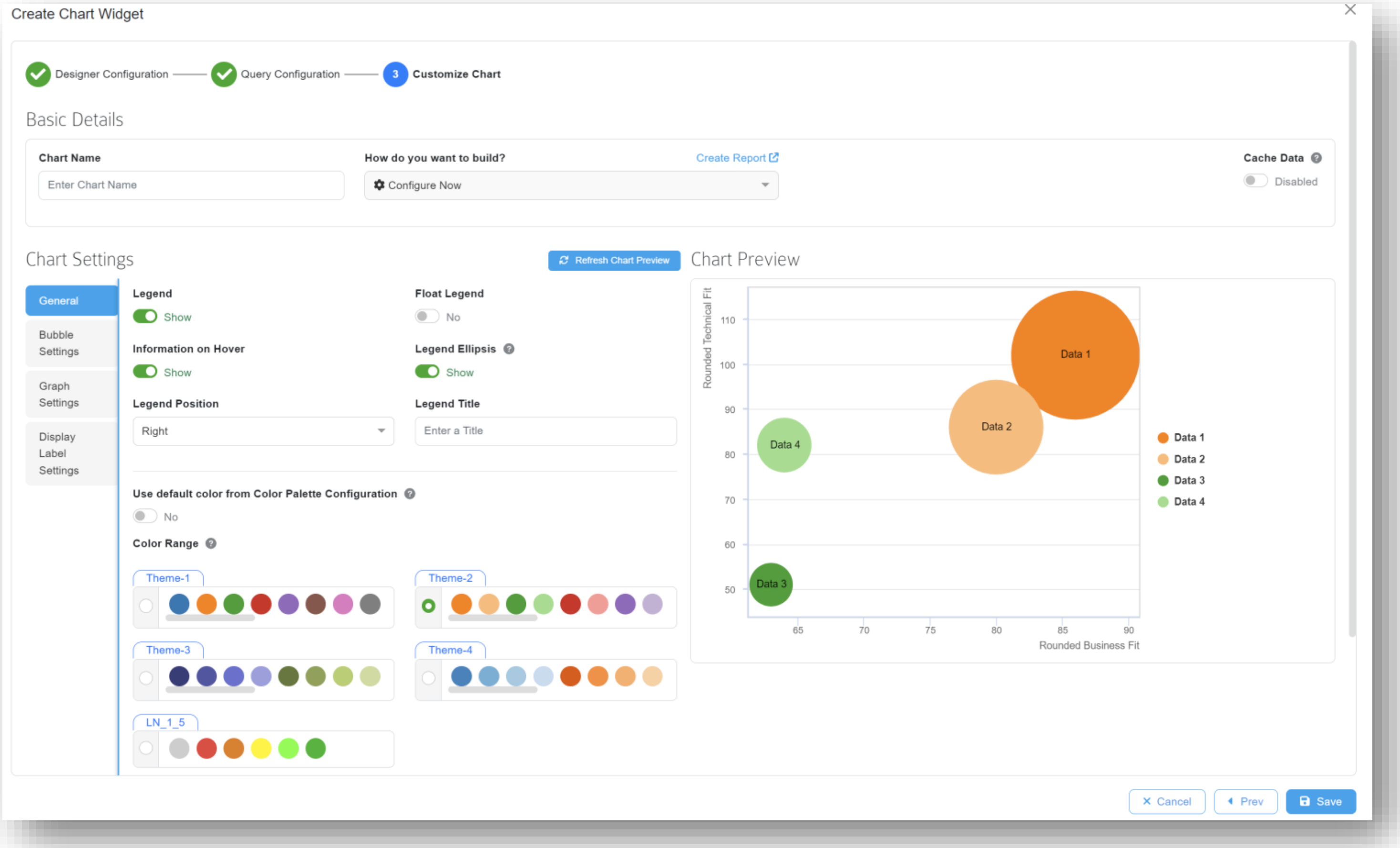
```
SELECT oApp.ea_guid AS CLASSGUID, oApp.Object_ID as Type, Name as Clickable_Application_Name,
xvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Business_Fit'),
yvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Technical_Fit'),
chartvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'NumberOfApplicationsInGroup')
```

Result Query:

```
SELECT oApp.ea_guid AS CLASSGUID, oApp.Object_ID as Type, Name as Clickable_Application_Name,
xvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Business_Fit'),
yvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Technical_Fit'),
chartvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'NumberOfApplicationsInGroup')
```

Query details:

```
SELECT oApp.ea_guid AS CLASSGUID, oApp.Object_ID as Type, Name as Clickable_Application_Name,
xvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Business_Fit'),
yvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Rounded_Technical_Fit'),
chartvalue = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property='NumberOfApplicationsInGroup'),
series = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'NumberOfApplicationsInGroup')
FROM t_object oApp
WHERE Object_Type = 'Class'
AND Package_ID = 6094
ORDER BY Name
```



Create Chart Widget

✓ Designer Configuration — ✓ Query Configuration — 3 Customize Chart

Basic Details

Chart Name: Enter Chart Name

How do you want to build?: Configure Now

Create Report

Cache Data: Disabled

Chart Settings

Refresh Chart Preview

General | Bubble Settings | Graph Settings | Display Label Settings

Legend: Show

Float Legend: No

Information on Hover: Show

Legend Ellipsis: Show

Legend Position: Right

Legend Title: Enter a Title

Use default color from Color Palette Configuration: No

Color Range

○ Theme-1

● Theme-2

○ Theme-3

○ Theme-4

○ LN_1_5

Chart Preview

✕ Cancel | ◂ Prev | Save

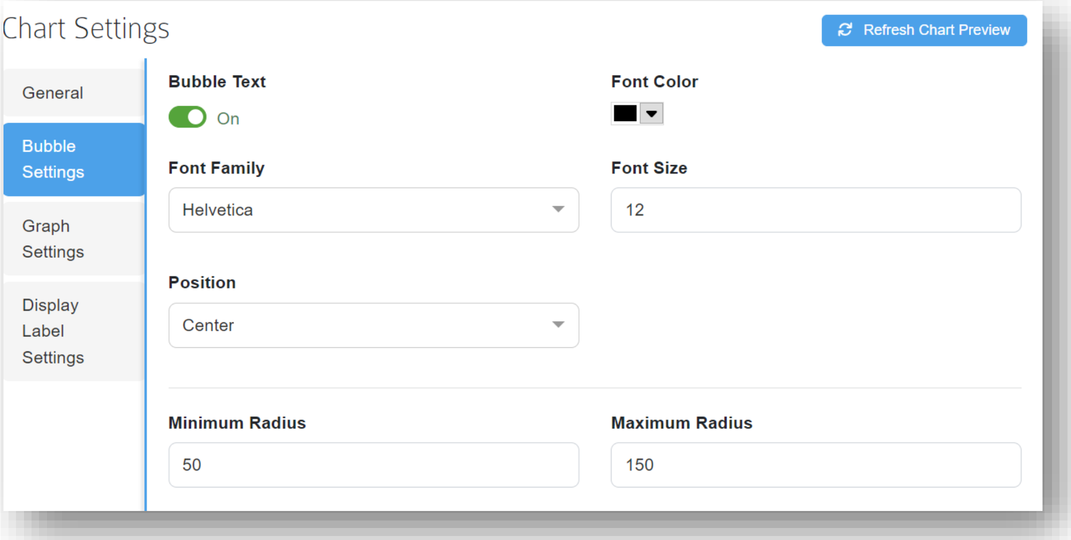


Chart Settings

Refresh Chart Preview

General | Bubble Settings | Graph Settings | Display Label Settings

Bubble Text: On

Font Color: ■

Font Family: Helvetica

Font Size: 12

Position: Center

Minimum Radius: 50

Maximum Radius: 150

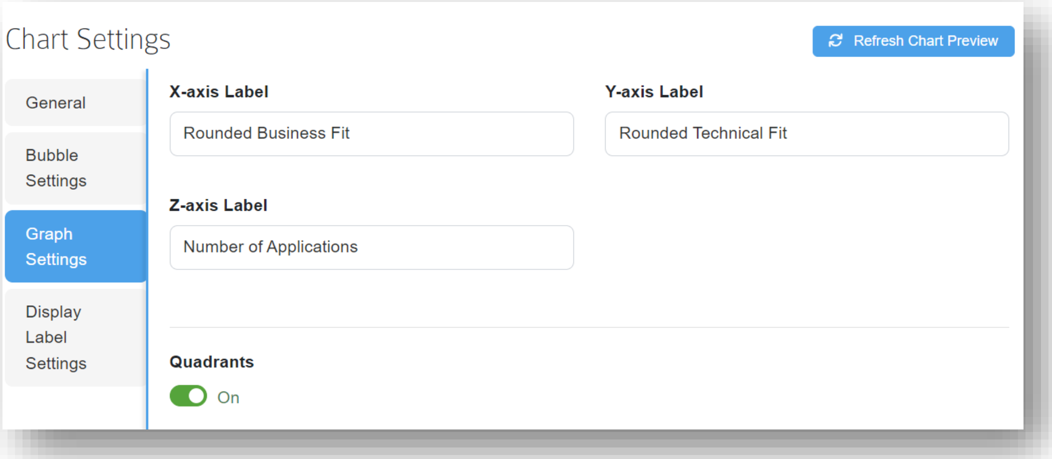


Chart Settings

X-axis Label
Rounded Business Fit

Y-axis Label
Rounded Technical Fit

Z-axis Label
Number of Applications

Quadrants
On

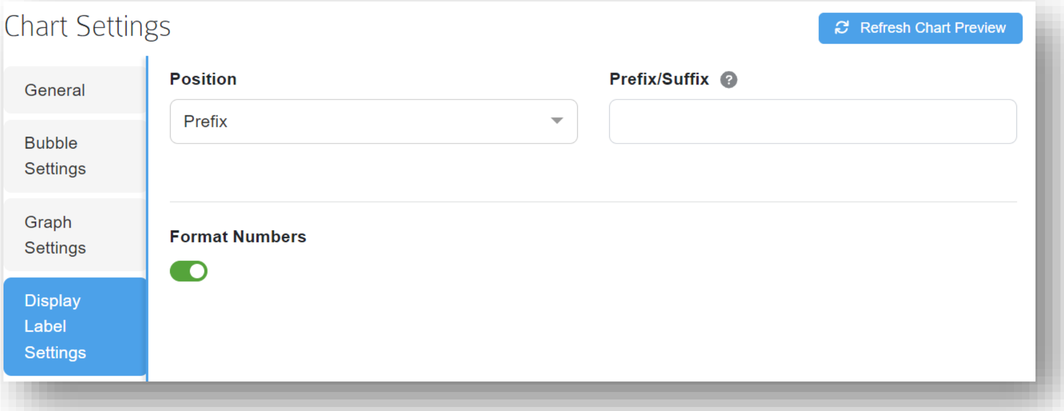


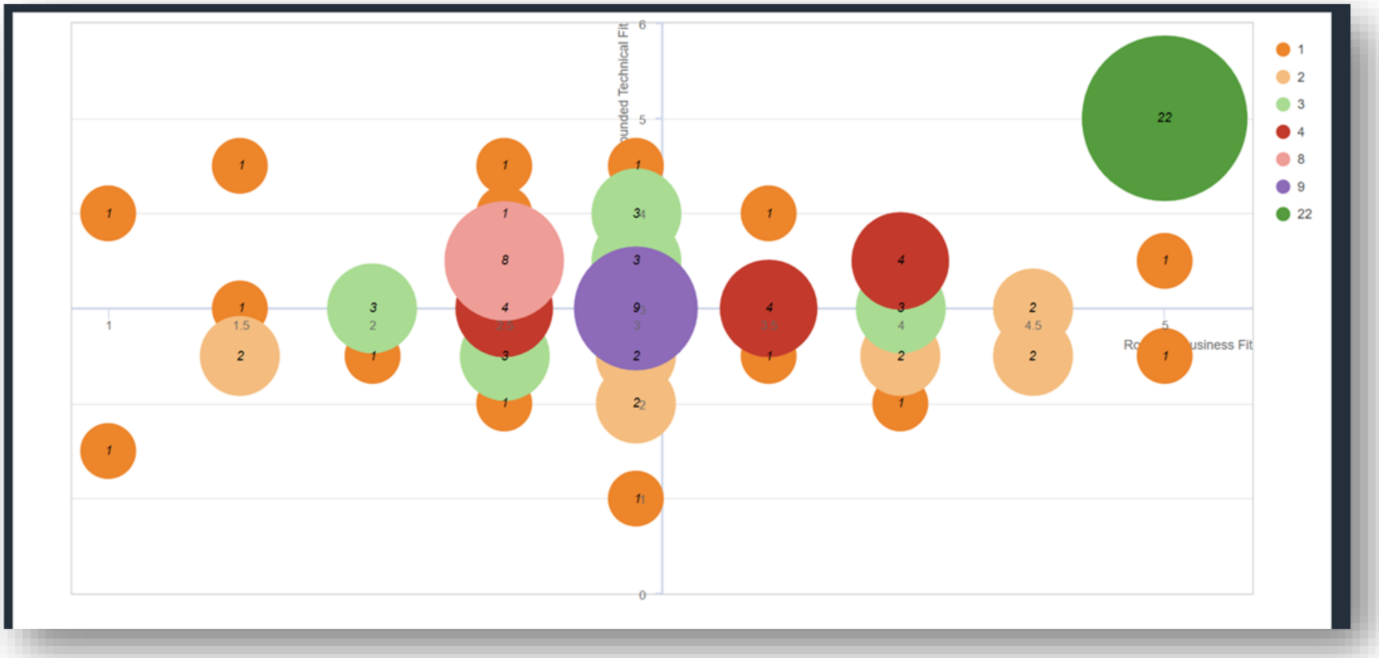
Chart Settings

Position
Prefix

Prefix/Suffix

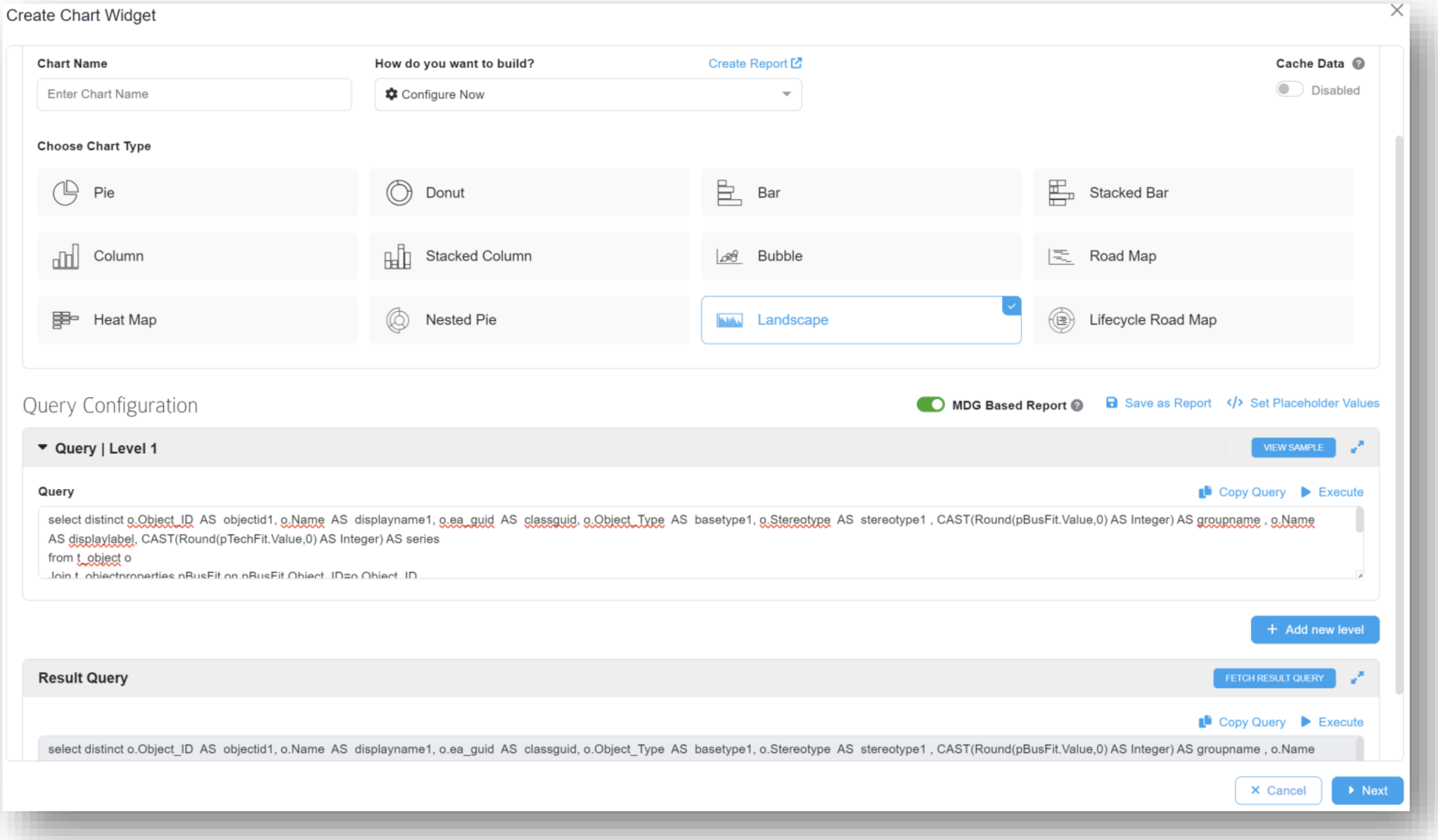
Format Numbers

The resulting chart should appear as follows:



CREATE A TIME MODEL LANDSCAPE CHART

Please, proceed as described in the following Prolaborate chart creation windows.



Create Chart Widget

Chart Name: Enter Chart Name

How do you want to build? Configure Now

Cache Data: Disabled

Choose Chart Type: Landscape

Query Configuration

MDG Based Report: enabled

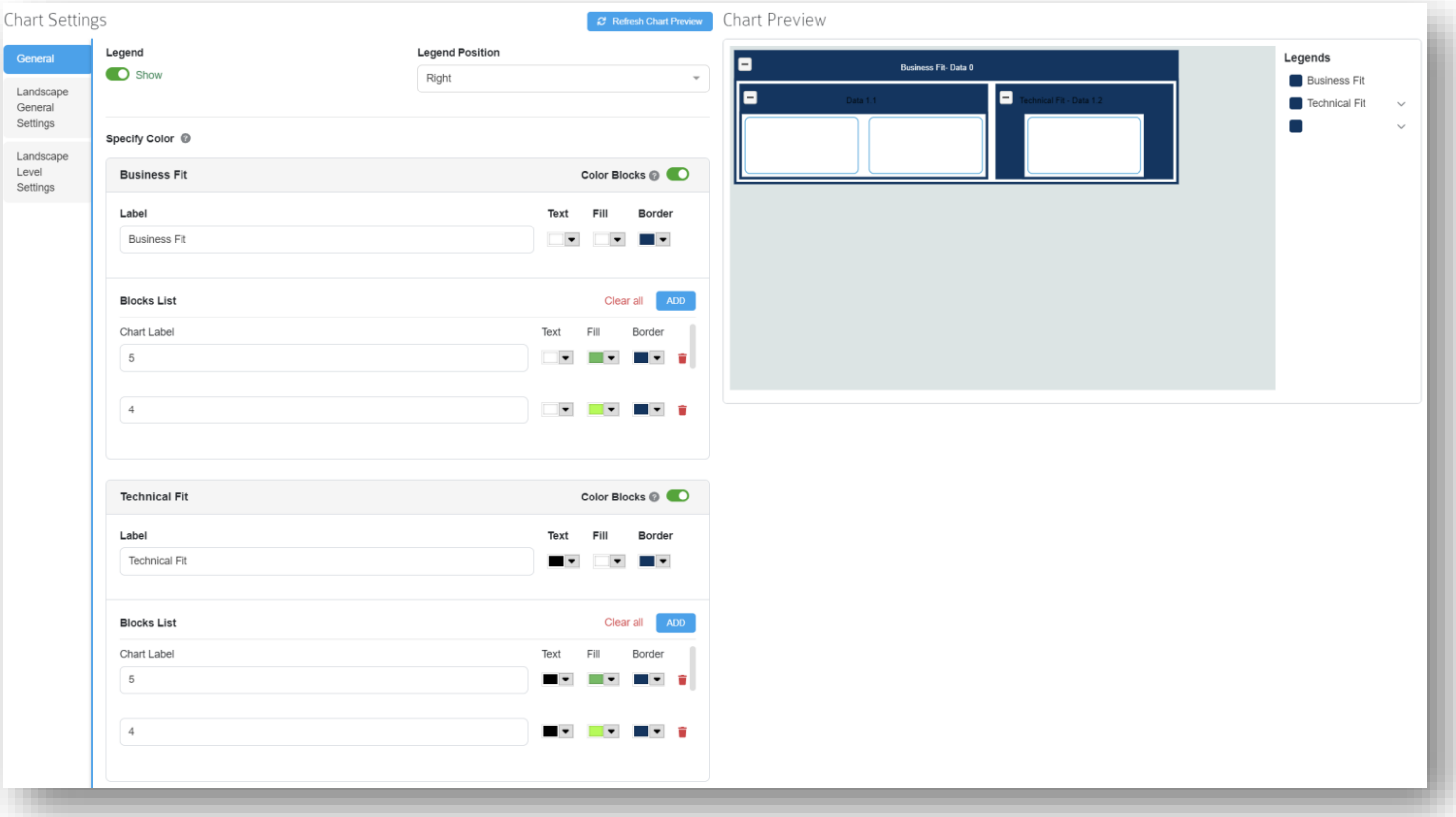
Query | Level 1

Query

```
select distinct o.Object_ID  AS  objectid1, o.Name  AS  displayname1, o.ea_guid  AS  classguid, o.Object_Type  AS  basetype1, o.Stereotype  AS  stereotype1 , CAST(Round(pBusFit.Value,0) AS Integer) AS groupname , o.Name
AS displaylabel, CAST(Round(pTechFit.Value,0) AS Integer) AS series
from t_object o
Join t_objectproperties pBusFit on pBusFit.Object_ID=o.Object_ID
```

Result Query

```
select distinct o.Object_ID  AS  objectid1, o.Name  AS  displayname1, o.ea_guid  AS  classguid, o.Object_Type  AS  basetype1, o.Stereotype  AS  stereotype1 , CAST(Round(pBusFit.Value,0) AS Integer) AS groupname , o.Name
```



Legend: Show

Legend Position: Right

Specify Color

Business Fit Color Blocks

Label	Text	Fill	Border
Business Fit	⬜	⬜	🟦

Blocks List

Chart Label	Text	Fill	Border
5	⬜	🟩	🟦
4	⬜	🟩	🟦

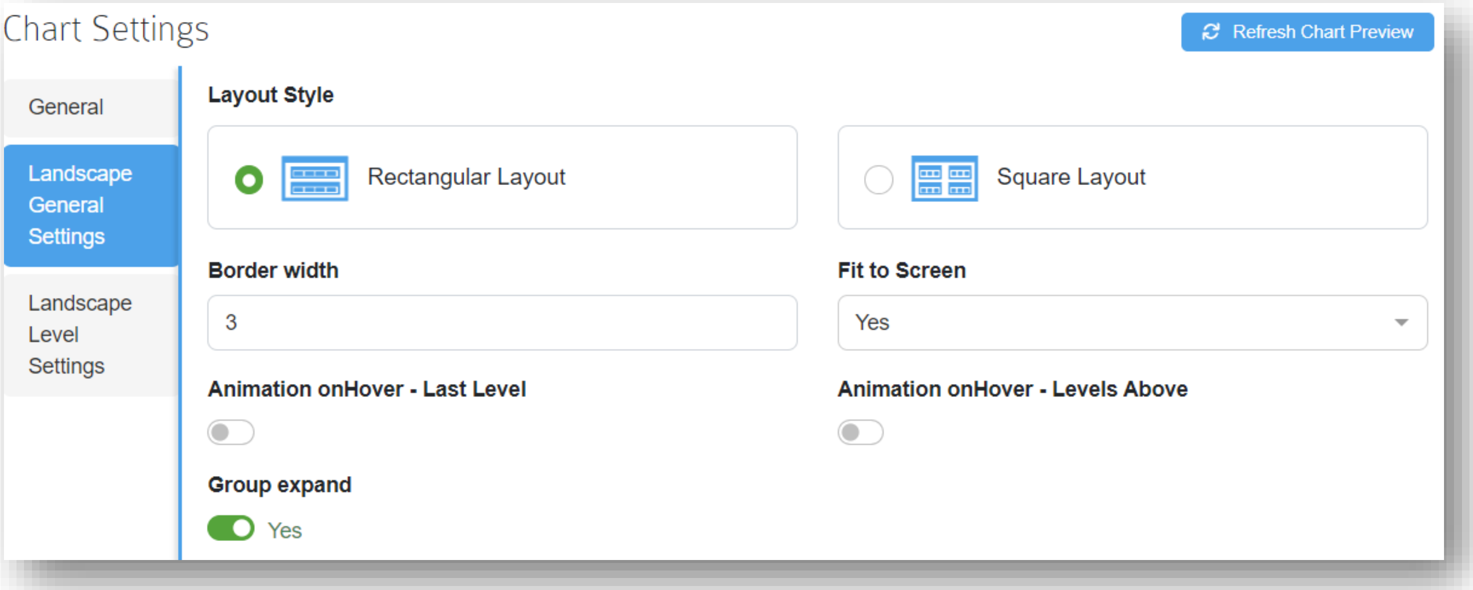
Technical Fit Color Blocks

Label	Text	Fill	Border
Technical Fit	⬛	⬜	🟦

Blocks List

Chart Label	Text	Fill	Border
5	⬛	🟩	🟦
4	⬛	🟩	🟦

Legends
- 🟦 Business Fit
- 🟦 Technical Fit
- 🟦



Layout Style

● Rectangular Layout ○ Square Layout

Border width: 3

Fit to Screen: Yes

Animation onHover - Last Level: off

Animation onHover - Levels Above: off

Group expand: Yes

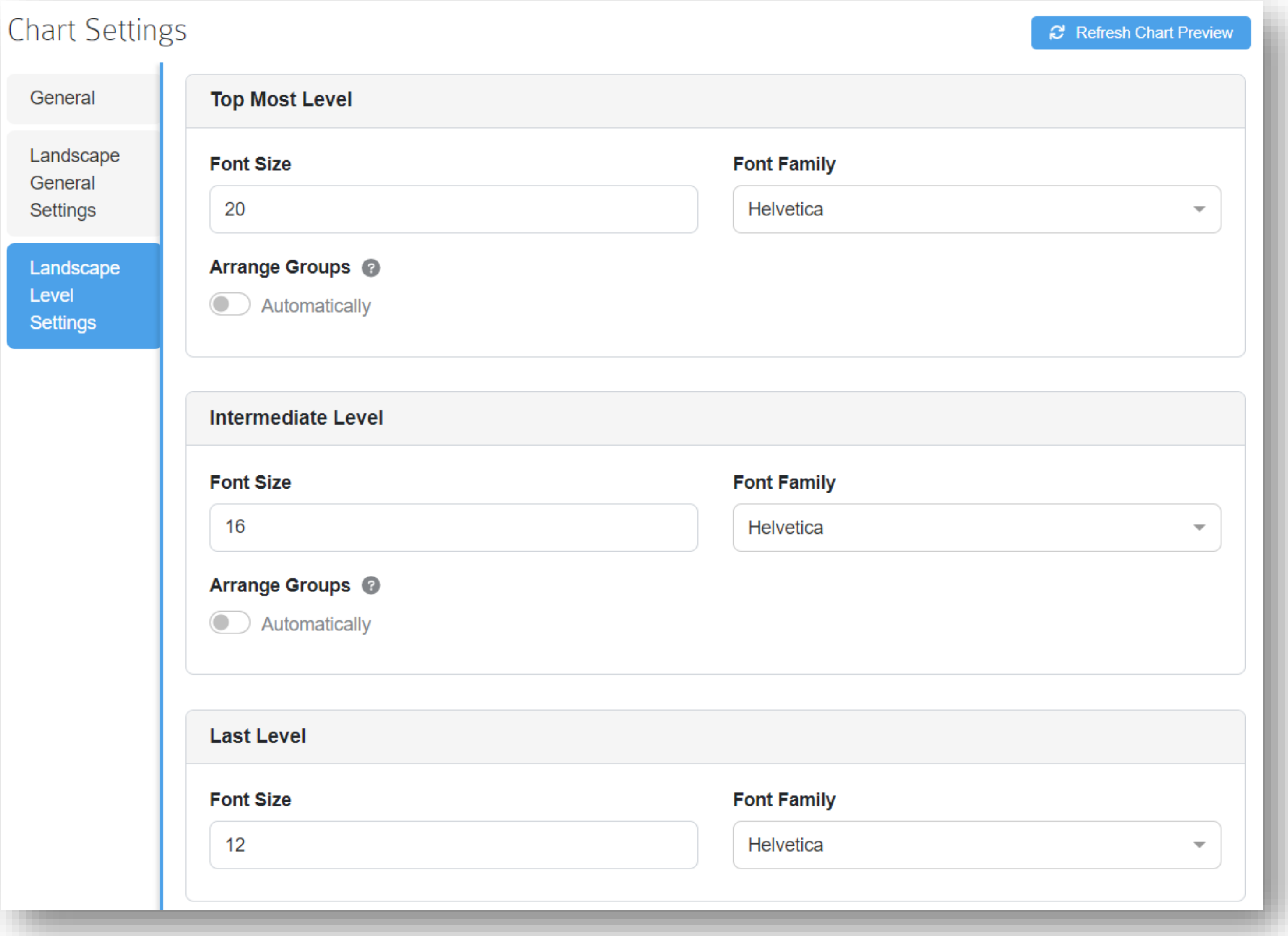


Chart Settings

Top Most Level

Font Size: 20

Font Family: Helvetica

Arrange Groups ?

○ Automatically

Intermediate Level

Font Size: 16

Font Family: Helvetica

Arrange Groups ?

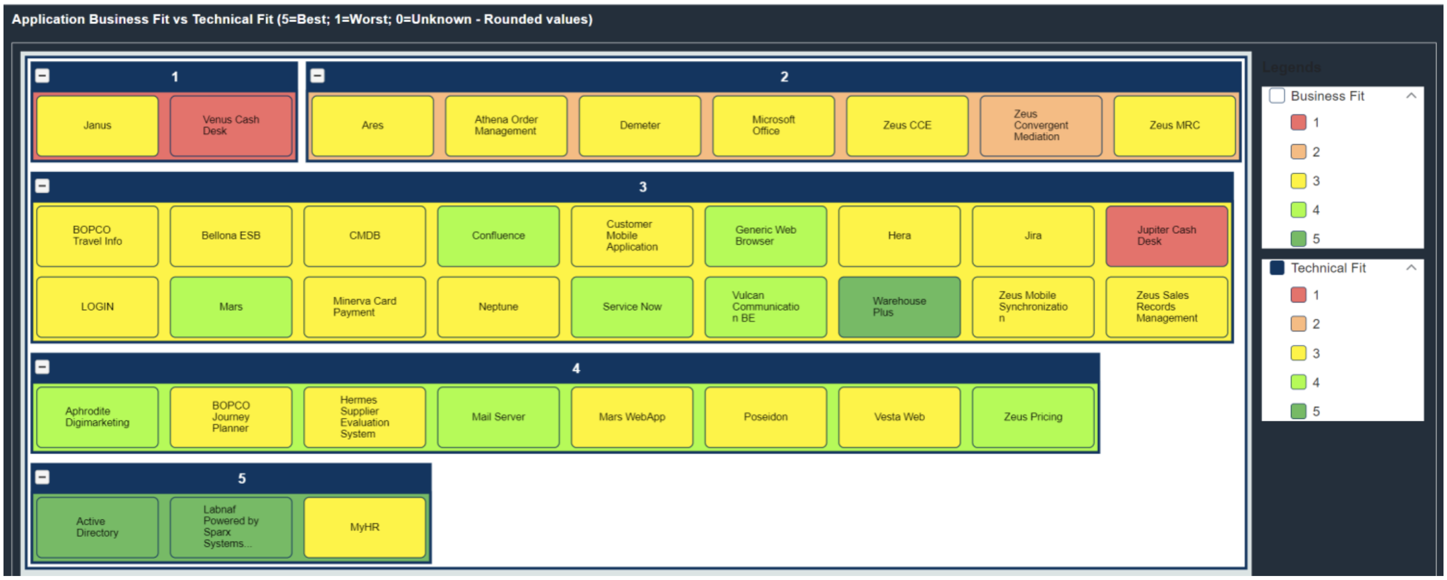
○ Automatically

Last Level

Font Size: 12

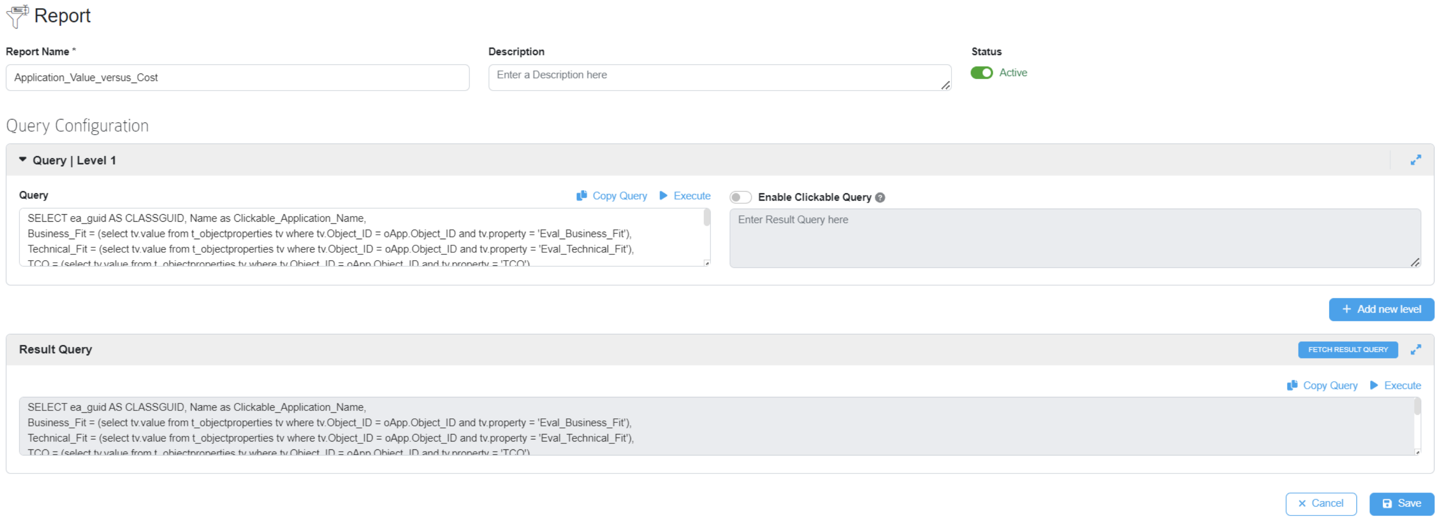
Font Family: Helvetica

The resulting chart should appear as follows:



CREATE A TIME MODEL REPORT

Please, proceed as described in the following Prolaborate report creation window.



Report

Report Name *: Application_Value_versus_Cost

Description: Enter a Description here

Status: Active

Query Configuration

Query | Level 1

Query

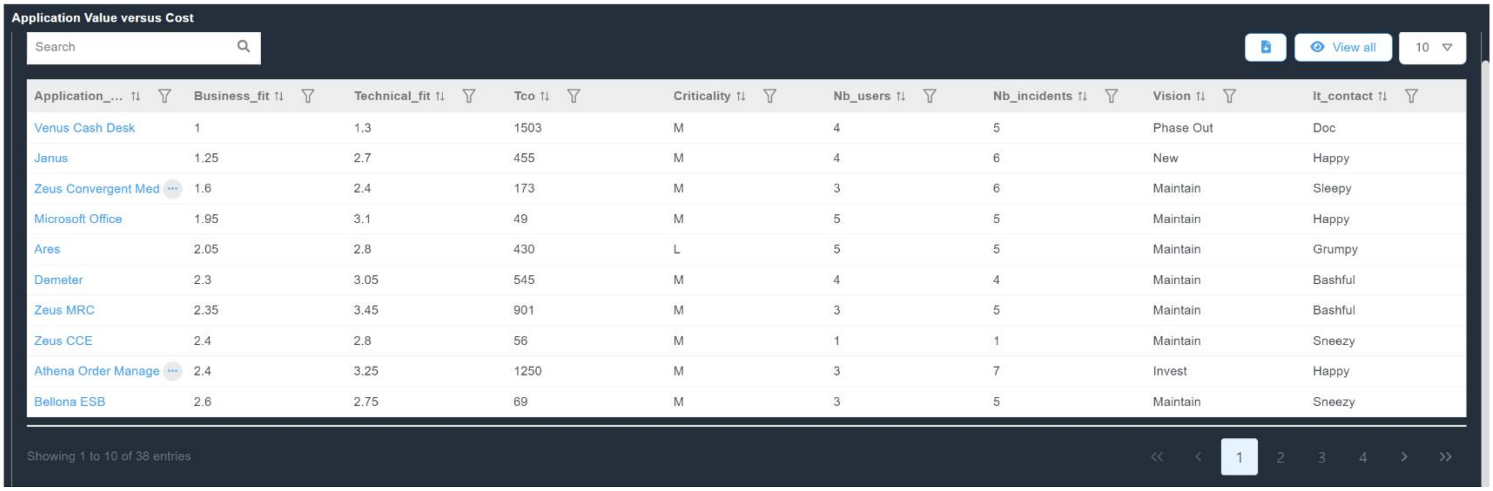
```
SELECT ea_guid AS CLASSGUID, Name as Clickable_Application_Name,
Business_Fit = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Eval_Business_Fit'),
Technical_Fit = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Eval_Technical_Fit'),
TCO = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'TCO')
```

Result Query

```
SELECT ea_guid AS CLASSGUID, Name as Clickable_Application_Name,
Business_Fit = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Eval_Business_Fit'),
Technical_Fit = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'Eval_Technical_Fit'),
TCO = (select tv.value from t_objectproperties tv where tv.Object_ID = oApp.Object_ID and tv.property = 'TCO')
```

Then add the report to a dashboard.

The resulting report should appear as follows:



Application Value versus Cost

Application_...	Business_fit	Technical_fit	Tco	Criticality	Nb_users	Nb_incidents	Vision	It_contact
Venus Cash Desk	1	1.3	1503	M	4	5	Phase Out	Doc
Janus	1.25	2.7	455	M	4	6	New	Happy
Zeus Convergent Med ···	1.6	2.4	173	M	3	6	Maintain	Sleepy
Microsoft Office	1.95	3.1	49	M	5	5	Maintain	Happy
Ares	2.05	2.8	430	L	5	5	Maintain	Grumpy
Demeter	2.3	3.05	545	M	4	4	Maintain	Bashful
Zeus MRC	2.35	3.45	901	M	3	5	Maintain	Bashful
Zeus CCE	2.4	2.8	56	M	1	1	Maintain	Sneezy
Athena Order Manage ···	2.4	3.25	1250	M	3	7	Invest	Happy
Bellona ESB	2.6	2.75	69	M	3	5	Maintain	Sneezy

Showing 1 to 10 of 38 entries