



A Language Built in its Native Framework

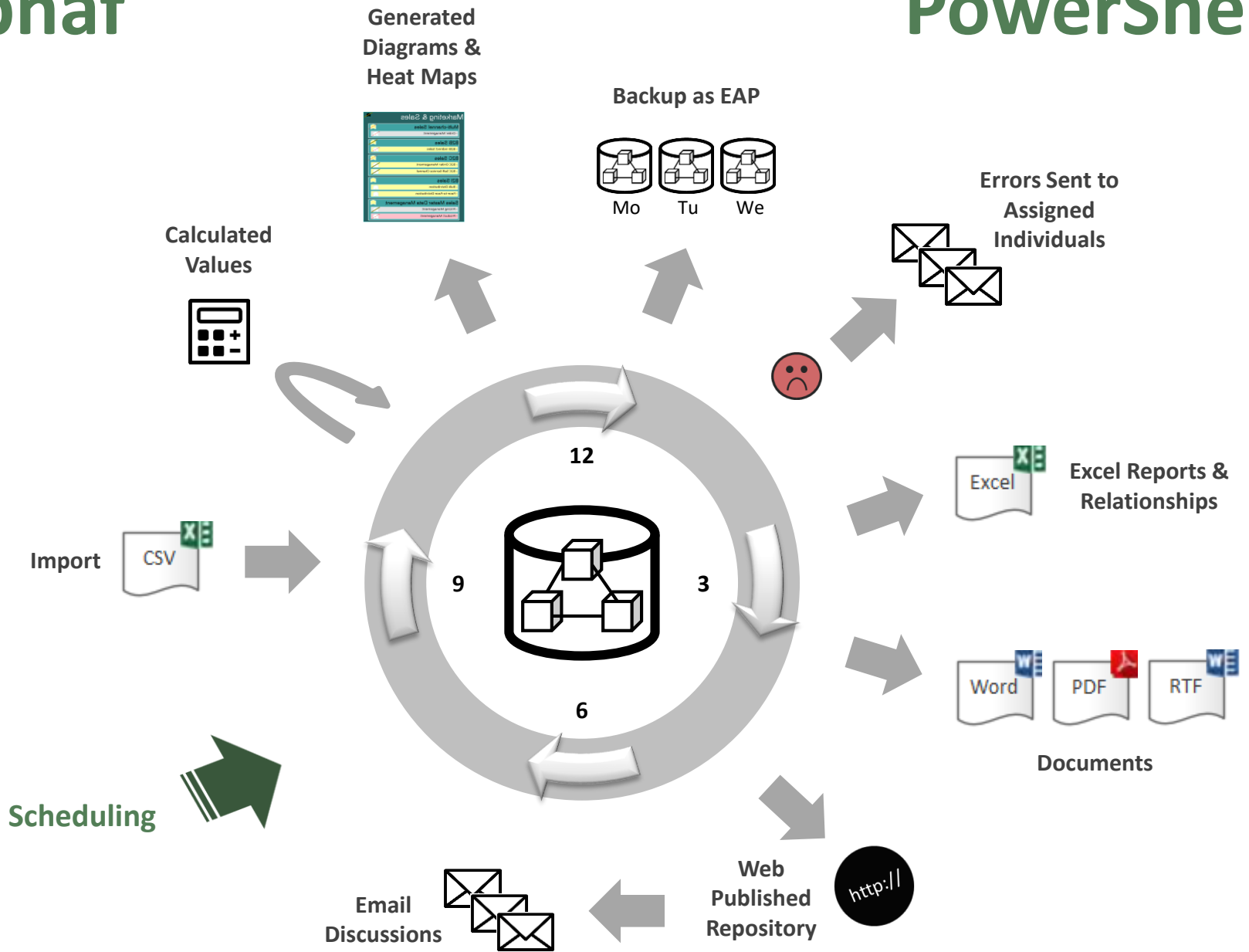
# The Labnaf Strategy & Architecture Framework

## Labnaf PowerShell

### User Guide

# Labnaf

# PowerShell

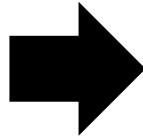


# The Labnaf PowerShell provides **command line** and **scheduled** access to the repository content

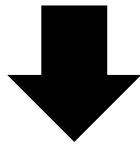
- Import From Csv
- Calculate Values
- **Validate** and send emails to assigned individuals
- Generate Diagrams
- Generate Tabular Report (Excel), **DOC** (Word, RTF, PDF), Html
- Backup To Access File
- Schedule Command (not only Labnaf PowerShell commands)

# Running commands manually (on the command line)

Inps



Inps {command name}



```
Usage : Inps Command [arguments]
```

```
Available Commands:
```

```
BackupToAccessFile
```

```
CalculateTaggedValues
```

```
GenerateDiagrams
```

```
GenerateDoc
```

```
GenerateHTML
```

```
GenerateTabularReports
```

```
ImportCsv
```

```
ScheduleCommand
```

```
Validate
```

**Example: if you type « Inps GenerateTabularReports » you get the following info.**

```
Command: GenerateTabularReports
```

```
Description: Generate spreadsheets from a model repository based on configuration stored in that same repository.
```

```
Usage : Inps GenerateTabularReports [arguments]
```

```
Arguments:
```

```
SourceRepoPathName: Path name of the source model repository (EAP file).
```

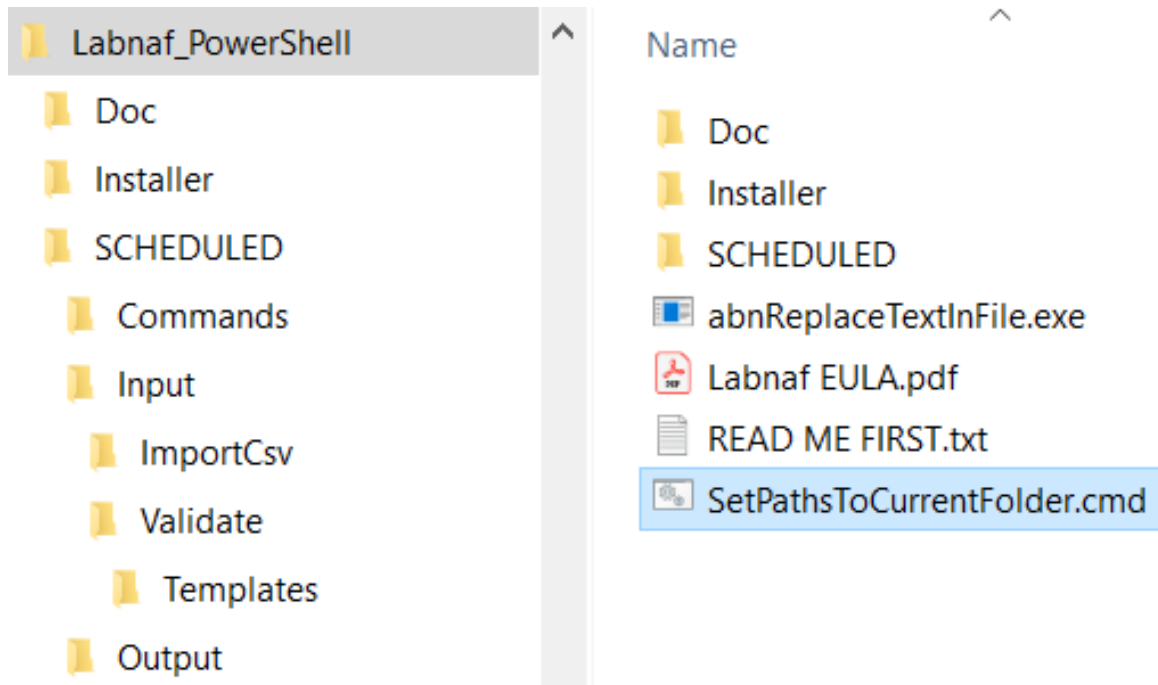
```
OutputDirectoryPath: Directory path name where the spreadsheets must be generated. The name of each spreadsheet file is the name of the template report.
```

```
[ElementPrototypeName]: The name of a specific element prototype name for which all embedded tabular report templates must be applied.
```

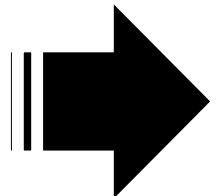
```
[TabularReportTemplateName]: The name of a specific tabular report template to be applied.
```

# Automatic configuration

- Copy the Labnaf\_PowerShell folder anywhere you want on your file system
- Double-click on “SetPathsToCurrentFolder.cmd”



*This updates the Labnaf PowerShell configuration files following the “Labnaf\_PowerShell” folder location.*



Preconfigured  
batches calling  
commands with  
predefined  
settings

Folder structure:

- SCHEDULED
- Commands
- Input
- ImportCsv
- Validate
- Templates
- Output

File list:

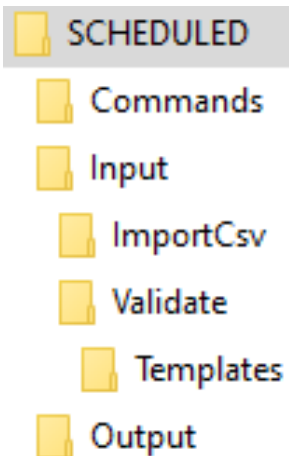
Name
_Demo_With_Access (limited to Access capabilities).cmd
_Demo_With_SqlServer.cmd
BackupToAccessFile.cmd
CalculateTaggedValues.cmd
Command Prompt
GenerateDiagrams.cmd
GenerateDoc.cmd
GenerateHTML.cmd
GenerateTabularReports.cmd
ImportCsv.cmd
RunAll.cmd
<b>SetPaths.cmd</b>
ShowListOfAvailableCommands.cmd
Validate.cmd

The paths in this command  
were reset automatically

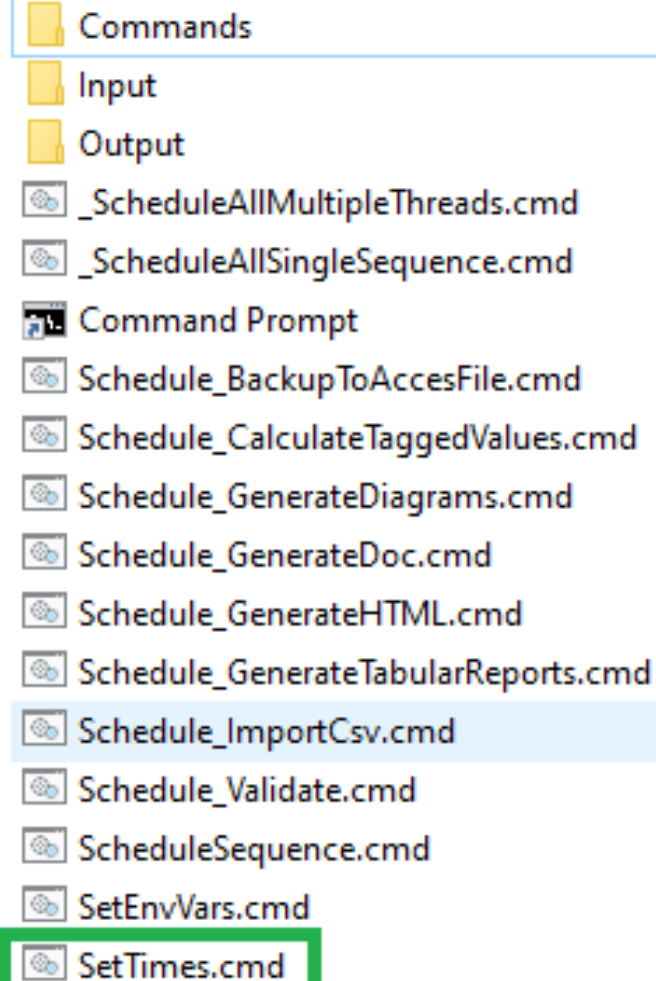
```
SetPaths.cmd x
```

```
1 set LABNAF_POWERSHELL=C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe
2
3 set SCHEDULED_DIR=C:\Users\User\Desktop\Labnaf_PowerShell\SCHEDULED
4 set COMMANDS_DIR=%SCHEDULED_DIR%\Commands
5
6 set INPUT_DIR=%SCHEDULED_DIR%\Input
7 set OUTPUT_DIR=C:\Users\User\Desktop\Labnaf_PowerShell\SCHEDULED\Output
8
9 set REPOSITORY=%INPUT_DIR%\Repository.eap
```

# Preconfigured Command Scheduling



Name



```
SetTimes.cmd
1  REM -- SINGLE START TIME --
2  Set StartTime_AllSingleSequence=00:00:00
3
4
5  REM -- SPECIFIC START TIME FOR EACH TASK --
6
7  Set StartTime_Cleanup_BackupToAccessFile=22:00:00
8  Set StartTime_Cleanup_GenerateHTML=22:00:05
9
10 Set StartTime_ImportCsv=22:30:00
11
12 Set StartTime_CalculateTaggedValues=23:00:00
13 Set StartTime_GenerateDiagrams=23:30:00
14
15 Set StartTime_BackupToAccessFile=00:00:00
16 Set StartTime_Validate=01:00:00
17
18 Set StartTime_GenerateTabularReports=02:00:00
19 Set StartTime_GenerateDoc=02:30:00
20 Set StartTime_GenerateHTML=03:00:00
21
22
23 REM -----
24
25 set SCHEDULED_MINUTES_UNTIL_RESTART=1440
```

# Import CSV

## Information to be imported

	A	B	C	D	E
1	guid	application_owner	application_owner_delegates	it_responsible_service	legal_owner
2	{D303A068-2CAA-438d-9E81-287EE9777F1D}	<a href="mailto:homer.simpson@labnaf.local">homer.simpson@labnaf.local</a>		Microsoft development	Labnaf
3	{305AA65E-A3F8-435b-81EC-C22EB7DF01C4}	<a href="mailto:marge.simpson@labnaf.local">marge.simpson@labnaf.local</a>	<a href="mailto:lisa.simpson@labnaf.local">lisa.simpson@labnaf.local</a>	Enterprise Architecture	Labnaf
4	{07F7FA8B-A01C-4aed-B5C2-80C9D62BD3FF}	<a href="mailto:bart.simpson@labnaf.local">bart.simpson@labnaf.local</a>		SAP development	Labnaf

## CSV vs Repository Column Mapping

	A	B
1	Input_Column_Names	Target_Column_Names
2	guid	ea_guid
3	application_owner	IT_Contact
4	application_owner_delegates	IT_Contact_Delegates
5	it_responsible_service	IT_ResponsibleService
6	legal_owner	Legal_Owner

## To manually start the import

```
Command: ImportCsv
```

```
Description: Import properties and tagged values from a CSV file.
```

```
Usage : Inps ImportCsv [arguments]
```

```
Arguments:
```

```
RepoPathName: Path name of the model repository (EAP file).
```

```
SourceCsvFile: A CSV file containing the data that needs to be imported.
```

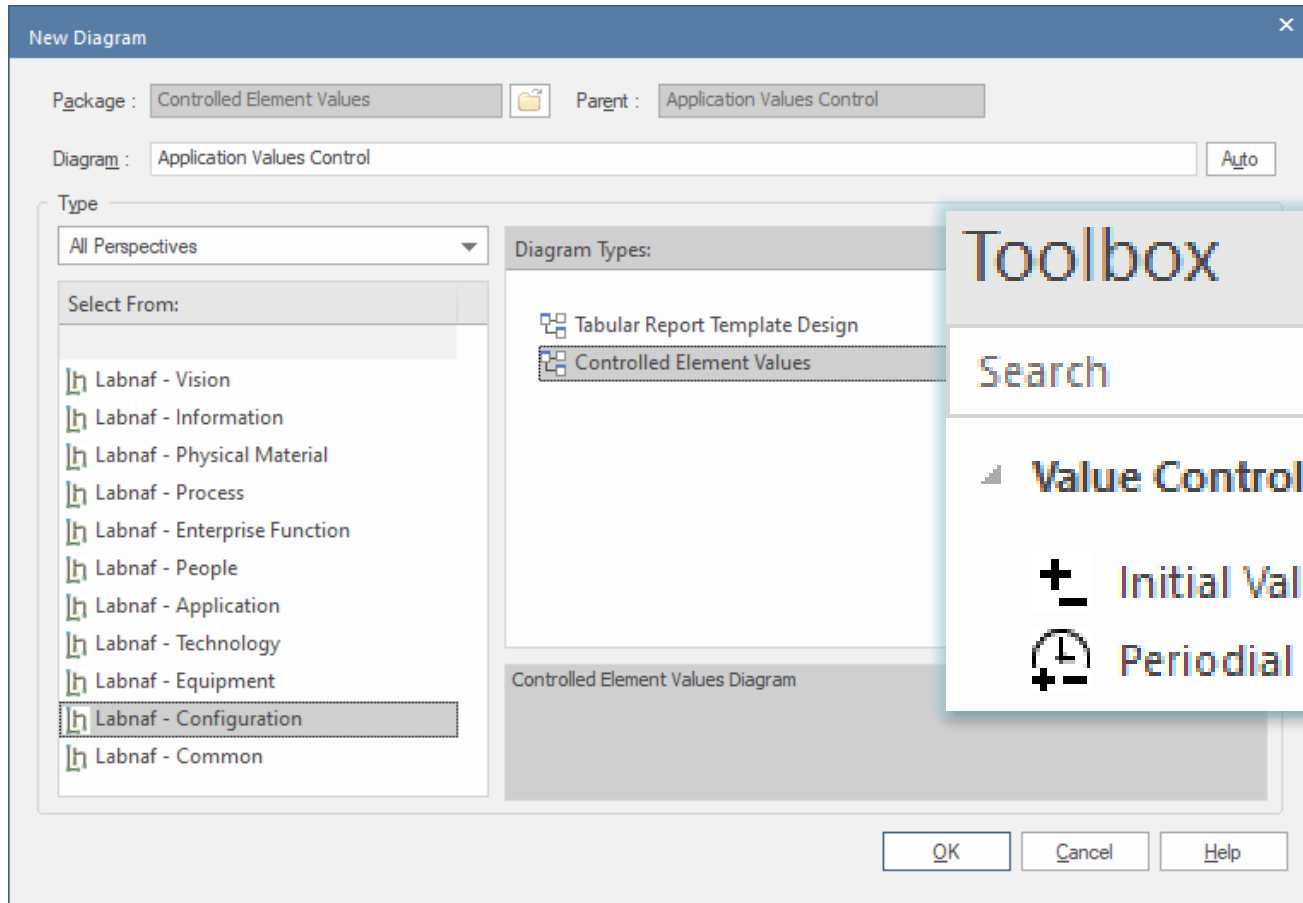
```
ColumnMappingFile: A CSV file containing the mapping between the input and output column names.
```

```
ElementStereotypeName: The stereotype of the elements that need to be updated..
```



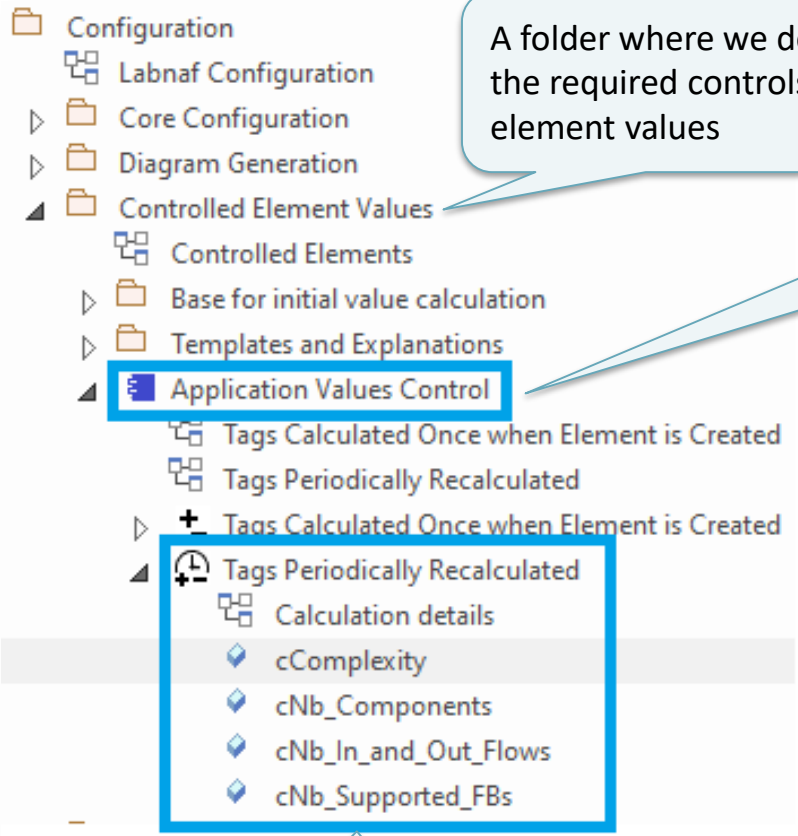
# Calculate Tagged Values

## Model your value calculations



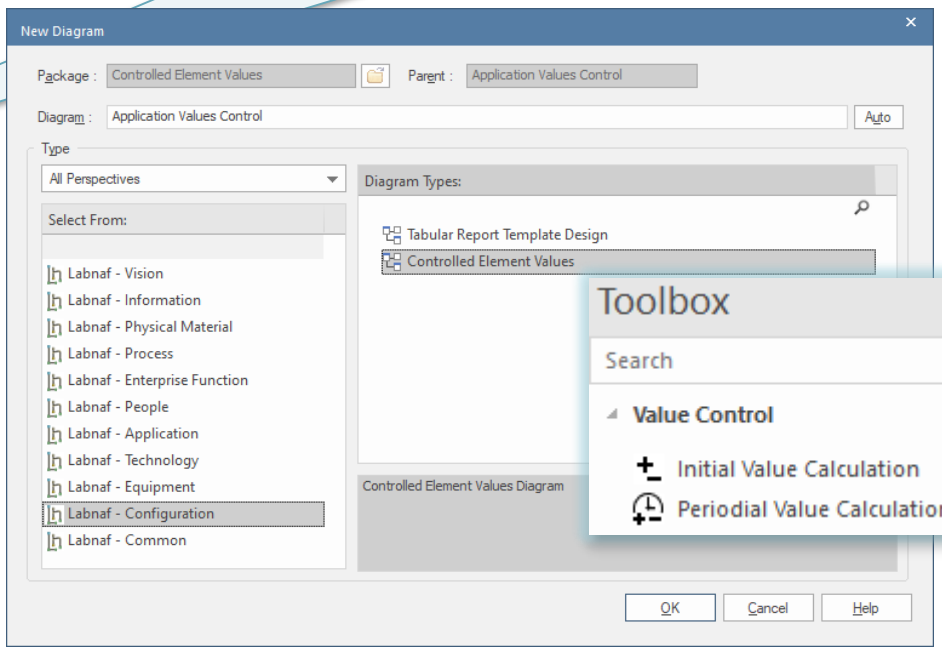
# Calculate Tagged Values (cont.)

## Model your value calculations



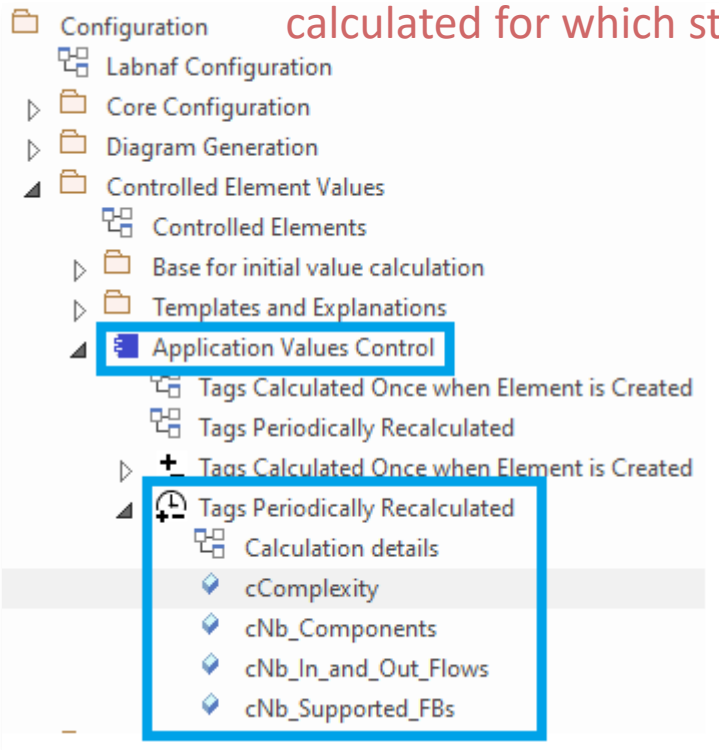
A folder where we define the required controls on element values

An element prototype for grouping all the required value controls. So we can see that the embedded value control definitions are for elements of that specific type and stereotype.



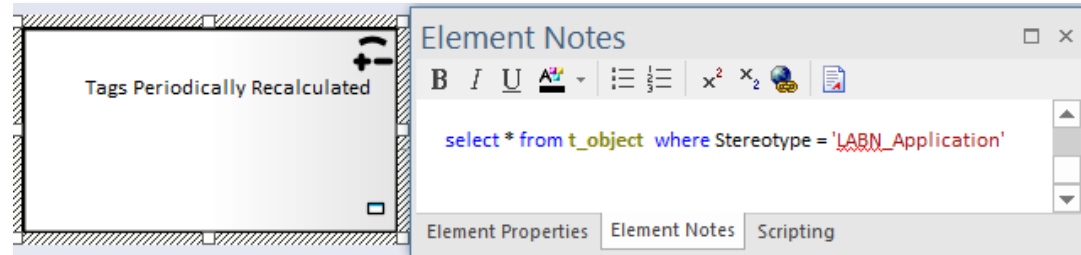
The tagged values that must be automatically calculated

# 1. Structure: What tagged values need to be calculated for which stereotype?

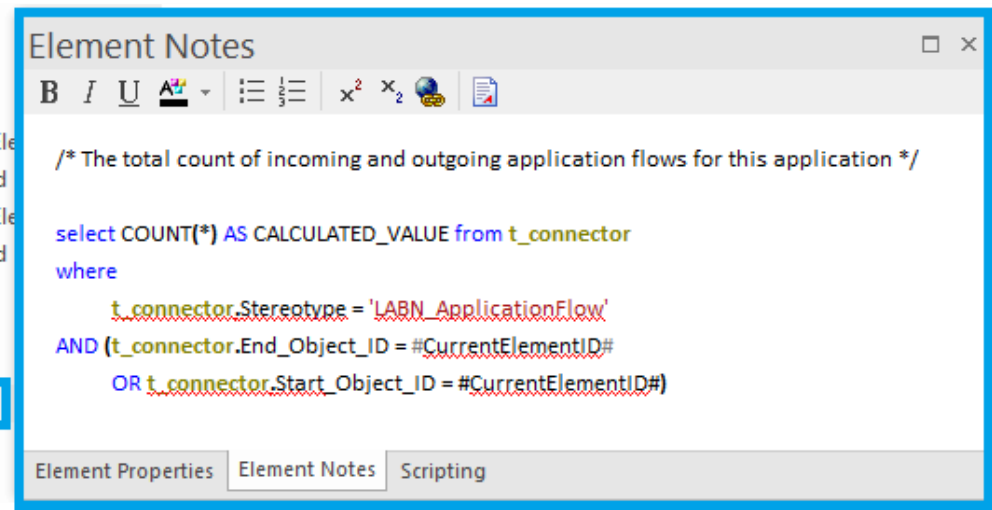
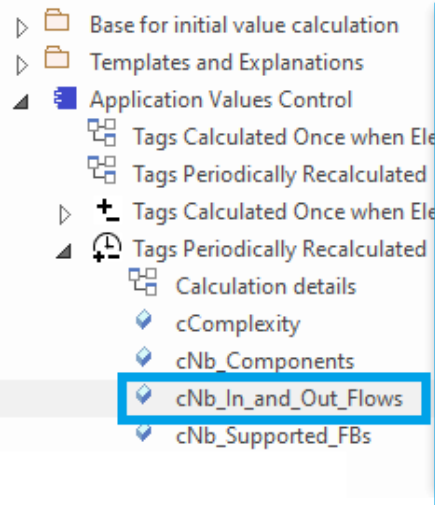


# Calculate Tagged Values (cont.)

## 2. Scope: Which elements need to be updated?



## 3. Calculation: How shall we calculate the value?



# Calculate Tagged Values (cont.)

## 4. To manually start calculation:

Command: `CalculateTaggedValues`

Description: Calculate values for some defined tags and elements. The elements to be selected, the tags to be updated and the calculation formulas are all defined in the model repository.

Usage : `lnps CalculateTaggedValues [arguments]`

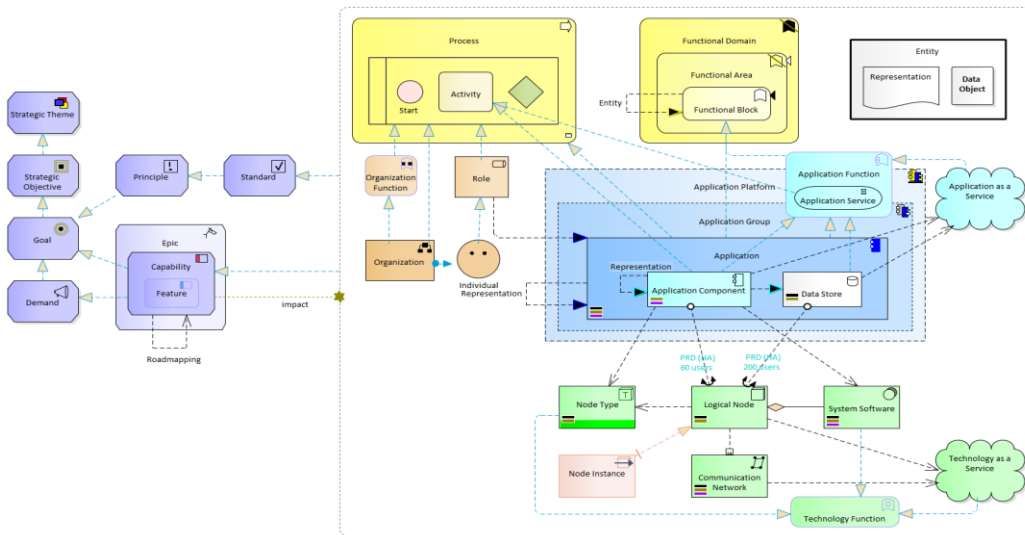
Arguments:

`RepoPathName`: Repository path name (EAP file).

`[ElementPrototypeName]`: A specific element stereotype for which tagged values must be calculated.

`[TagName]`: The name of a specific tagged value that must be calculated.

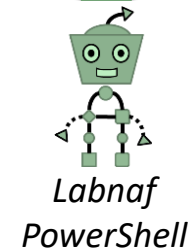
The Language Metamodel is used both for documentation & automatic model validation



While Modeling

Existing Invalid Connectors

Prevent creation of invalid connectors



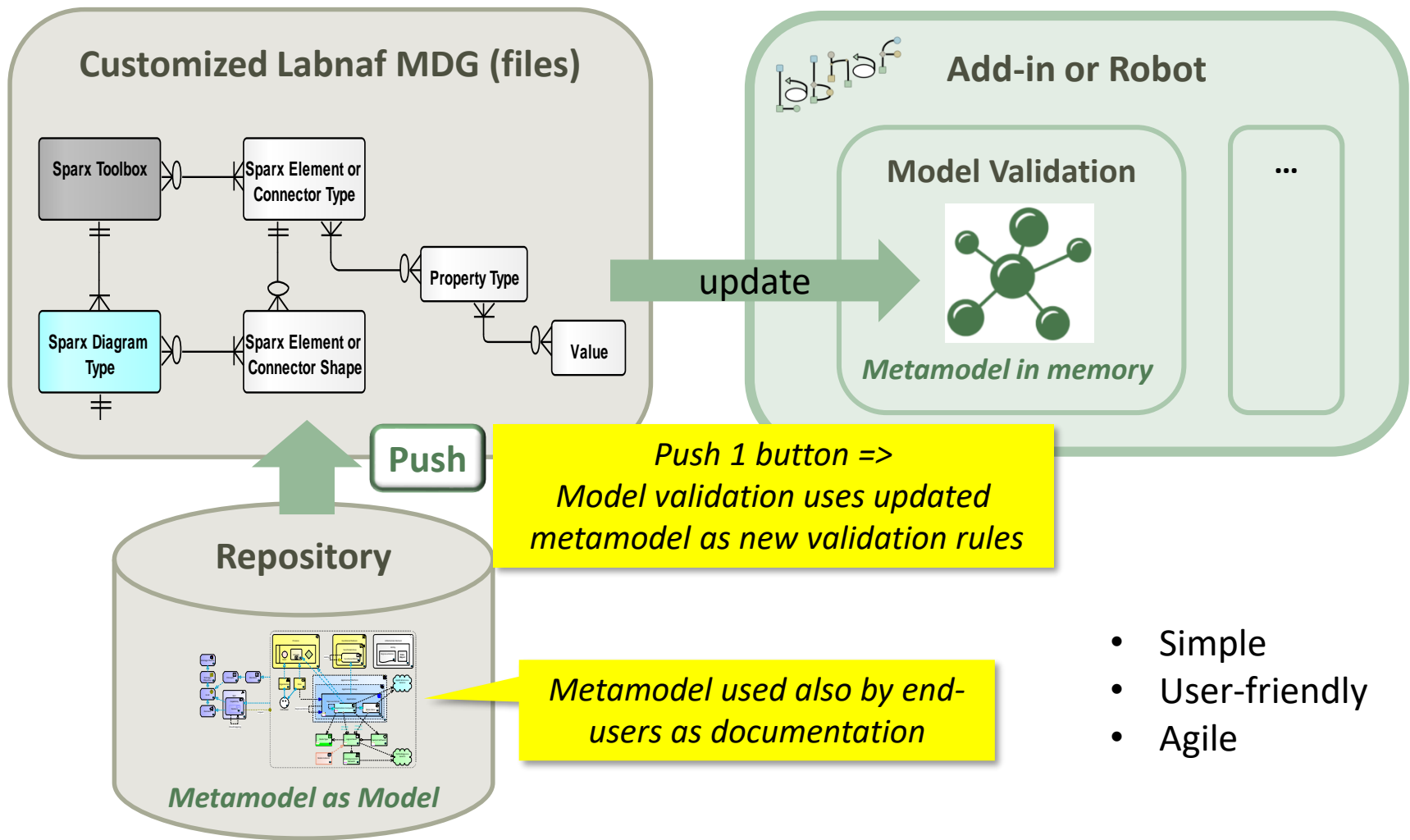
Send Error Emails to Relevant Recipients

## Why do we need periodical validation?

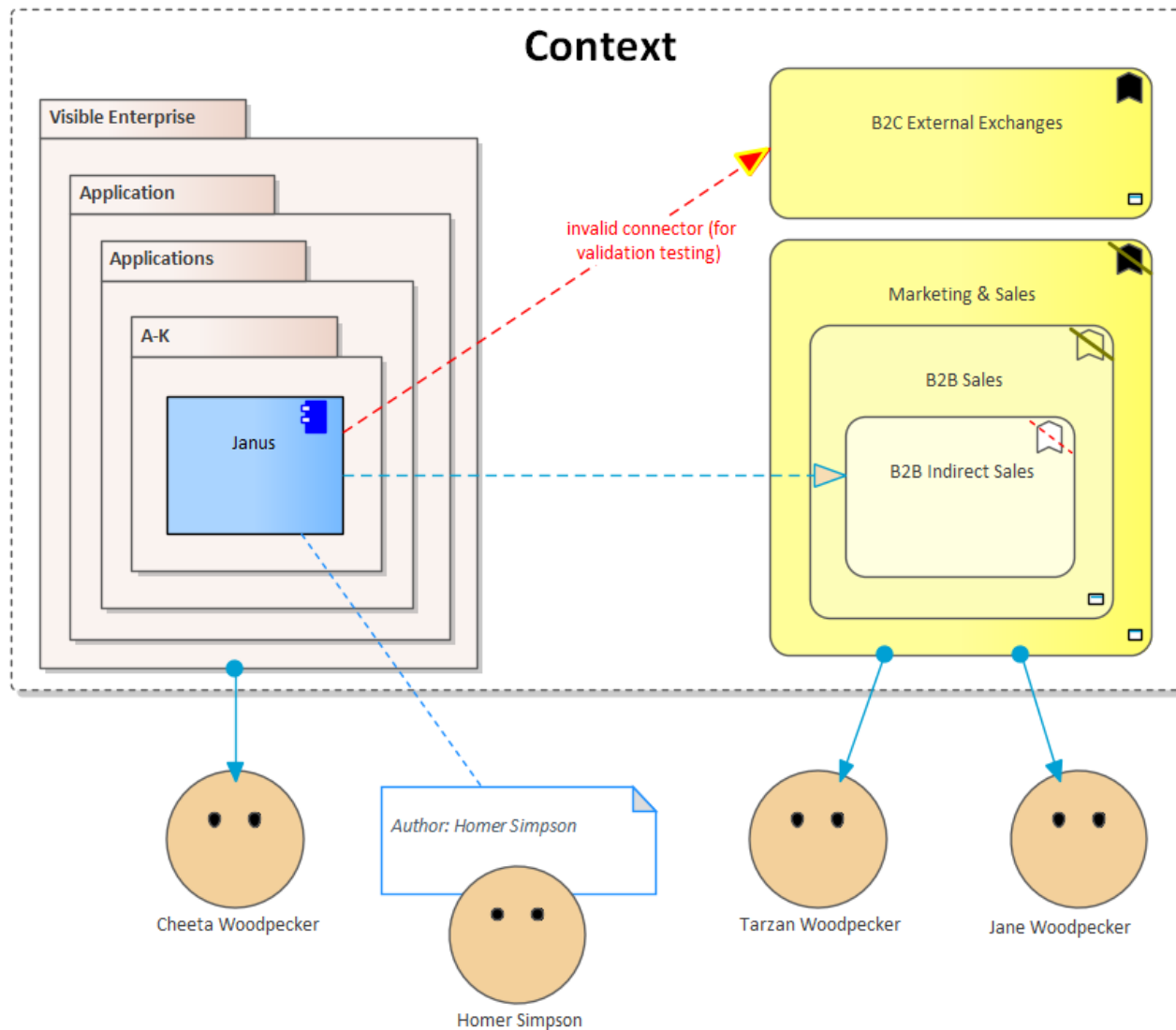
How could we have errors if we prevent users from entering errors?

- At the beginning, informal (invalid) models can be imported and their language can be transformed e.g. from ArchiMate to Labnaf.
- Then, every time you update the metamodel to adapt to your enterprise specificities, some existing model repository content becomes invalid... according to your new rules.

The default metamodel can be very easily updated:  
One click on a connection in your production repository.



# Error message routing is based on architecture management assignments



Error message distribution depends on

- recent authors
- individuals assigned to packages
- individuals assigned to domains
- default recipients
- and rules that you can use to combine these different combinations

The catalog of individuals includes email addresses



# Sample error message sent to an assigned individual

PAPERCUT Log Rules Options Exit

**Welcome to Papercut!**  
2019-10-15 21:24:50 (19.3KB)

**From Model Validation Robot: Invalid objects.**  
2019-10-16 21:37:19 (2.1KB)

**From Model Validation Robot: Invalid objects.**  
2019-10-29 14:43:02 (1.1KB)

From: model.validation@labnaf.local

To: tarzan.woodpecker@Labnaf.local

Date: 2019-10-16 21:37:19 +02:00

Subject: **From Model Validation Robot: Invalid objects.**

Message Headers Body Sections Raw

The [connector validation robot](#) identified some inconsistent content.  
Following our current knowledge, these problem(s) occur in a domain or package where you are personally involved in.  
Please make sure that the following [model repository](#) items get corrected either by you or by your team.

Number of errors: 4

[Labnaf Application Component](#) (LABN\_ApplicationComponent) => [LABN\\_Triggering](#) => [Labnaf Activity](#) (LABN\_Activity)

[Labnaf Gateway](#) (LABN\_Gateway) => [UML:Dependency](#) => [Labnaf Activity](#) (LABN\_Activity)

[Labnaf Application Component](#) (LABN\_ApplicationComponent) => [UML:Aggregation](#) => [Labnaf Application Component](#) (LABN\_ApplicationComponent)

[UML Component](#) (UML:Component) => [LABN\\_Realization](#) => [Labnaf Application Component](#) (LABN\_ApplicationComponent)

# Validation rules can be further customized

```
<ValidationConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" >
  <SelectElementsInScope>
    SELECT * FROM t_object o
    WHERE stereotype like 'LABN_Application'
    AND o.Package_ID IN
    (SELECT p1.Package_ID FROM t_package p1
    LEFT JOIN t_package p2 ON p2.Package_ID = p1.Parent_ID
    WHERE p2.ea_guid = '{3E299811-1C55-43aa-841A-B1BC9869814A}')
    ORDER BY Name
  </SelectElementsInScope>
  <Sender>model.validation@labnafdemo.com</Sender>
  <SendTo>
    <FirstAvailableAlternativeOnly>true</FirstAvailableAlternativeOnly>
    <AuthorDuringMonthsAfterElementCreated>120</AuthorDuringMonthsAfterElementCreated>
    <PeopleAssignedToPackage>true</PeopleAssignedToPackage>
    <PeopleAssignedToDomain>true</PeopleAssignedToDomain>
    <DefaultEmailAddresses>lisa.simpson@labnaf.local</DefaultEmailAddresses>
  </SendTo>
  <PackageGuids>
    <Individuals>{C1EDF8B0-C5D3-453d-966A-D27D0CB2E8D5}</Individuals>
    <EnterpriseFunctions>{5FFB72A5-D3E1-4d74-96FE-E60967ABCE0D}</EnterpriseFunctions>
  </PackageGuids>
  <PublishedRepositoryWebSiteUrl>http://localhost/guidance</PublishedRepositoryWebSiteUrl>
  <DocumentationReferences>
    <GuidanceWebSiteUrl>http://www.Labnaf.one/guidance</GuidanceWebSiteUrl>
    <DiagramGuids>
      <ConnectorValidation>{269E2D0C-3B9E-4d85-915A-87905EB7271F}</ConnectorValidation>
      <ModelRepository>{EF41E336-AC6B-4407-88D9-3ECC41725132}</ModelRepository>
    </DiagramGuids>
  </DocumentationReferences>
</ValidationConfiguration>
```

If you want to be specific about the elements to be validated. By default all Labnaf elements are validated.

Error messages are sent from this email address.

Who will receive the error messages.

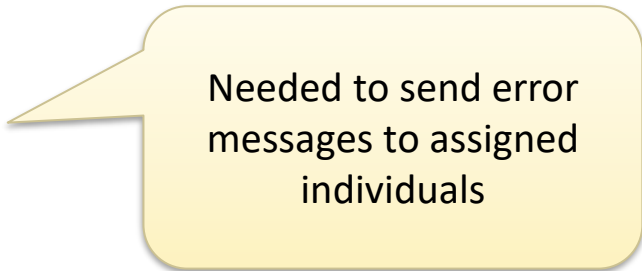
Assigned individuals with their emails are listed in a package

Individuals can be assigned to enterprise functions which are stored in a package.

Error messages contains urls to invalid elements. HTML publication should be scheduled as well

# Smtp Server configuration is straightforward

```
<?xml version="1.0" encoding="utf-8"?>
<SmtpServerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <Host>127.0.0.1</Host>
  <ClientPort>25</ClientPort>
  <EnableSSL>>false</EnableSSL>
  <UserName>alain@labnafdemo.com</UserName>
  <Password></Password>
</SmtpServerConfiguration>
```



Needed to send error messages to assigned individuals

## To manually start validation:

```
C:\A\LT\SparxDev\VSPProjects\Alain\LabNaf\PowerShell\bin\Release\Dotfuscated>lnps validate
Command: Validate
Description: Validate model repository.
Usage : lnps Validate [arguments]
Arguments:
  RepoPathName: Repository path name (EAP file).
  ValidationConfigurationFile: Path name of the model validation configuration file.
  SmtpServerConfigurationFile: Path name of the SMTP Server configuration file.
```

## Enterprise Function Taxonomy & Applications Supporting Level 1



List of applications supporting the domain.

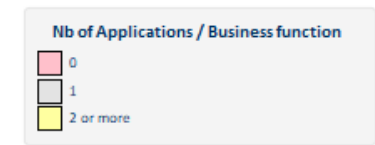
'Application Name'
Ares
Athena Cash Desk
Customer Mobile Application
Demeter
Hera
Janus
Jupiter Cash Desk
LOGIN B2B
Neptune
Venus Cash Desk
Zeus Pricing
Zeus Sales Records Management

**Legend for Business Functions**

Differentiator = ?	
= Y	
= N	

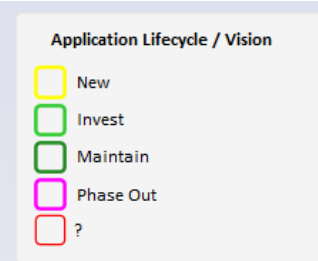
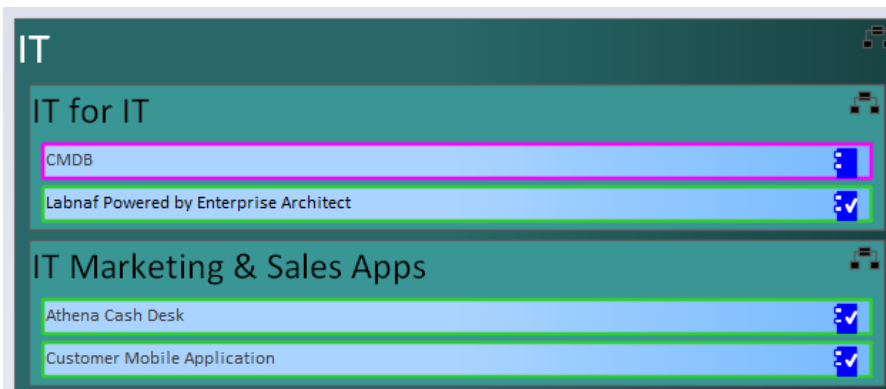
Externalized = Y	◀ Entirely externalized
= P	◀ Partially externalized
= N	Not externalized (default)



Other diagram(s) for this enterprise function:

[FULSales](#)

## Applications managed by organizations



Other diagram(s) for this organization:

[Configuration](#)

# Sample Diagram Templates

## Enterprise Function Taxonomy & Applications Supporting Level 1

The screenshot shows a software configuration tool interface. On the left is a tree view with the following structure:

- Configuration
  - Labnaf Configuration
  - Core Configuration
  - Diagram Generation
    - Diagram Generation
    - Periodical Diagram Generation - Elements or Packages In Scope
    - Diagram Templates
      - Applications Supporting Functional Domain
        - FAL - Applications Supporting Functional Domain** (highlighted)
        - List of applications supporting the domain.

The main area displays a diagram template titled "Functional Domain". It contains a "Functional Area" element with a "Functional Block" sub-element. To the right of the diagram is a table titled "List of applications supporting the domain." with a header "Application Name".

On the far right, there is a legend for business functions:

**Legend for Business Functions**

Differentiator = ?	
= Y	
= N	

Externalized = Y	◀ Entirely externalized
= P	◀ Partially externalized
= N	Not externalized (default)

**Nb of Applications / Business function**

0	
1	
2 or more	

Other diagram(s) for this enterprise function:

[Configuration](#)

## Applications managed by organizations

The screenshot shows a software configuration tool interface. On the left is a tree view with the following structure:

- Applications Supporting Organization
  - OBO - Applications Owned By Organization** (highlighted)
- Organization1
  - Organization2
    - Organization3
      - Organization4
        - Application1
        - Application2
        - Application3
        - Application4
        - Application5

The main area displays a diagram template titled "Applications managed by organizations". It shows a hierarchical structure of organizations:

- Organization1
  - Organization2
    - Organization3
      - Organization4
        - Application1
        - Application2
        - Application3
        - Application4
        - Application5

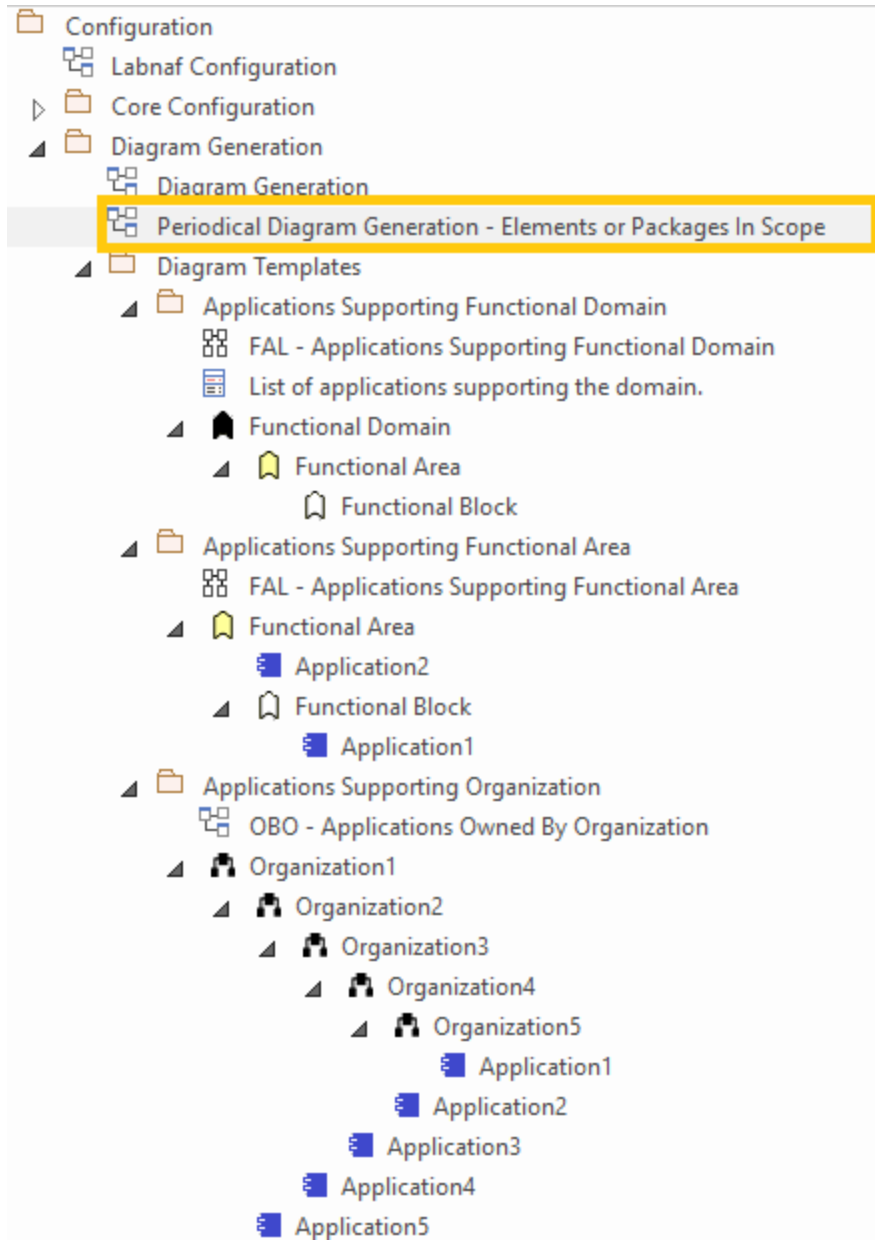
On the right, there is a legend for application lifecycle/vision:

**Application Lifecycle / Vision**

- New (Yellow square)
- Invest (Green square)
- Maintain (Light Green square)
- Phase Out (Pink square)
- ? (Red square)

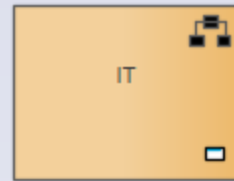
Other diagram(s) for this organization:

[Configuration](#)



## Enterprise Functions

- B2C External Exchanges
- B2B External Exchanges
- Information Management
- Internal Exchange
- Strategy, Legal Affairs, Risk & Compliance
- Traveler Communication
- Marketing & Sales
- Traffic Management
- Station Management
- Bus Route Planning
- Bus Maintenance
- Security
- HR & Corporate services
- Finance
- Supply Chain
- Information Technology



## To manually start diagram generation:

Command: `GenerateDiagrams`

Description: Generate diagrams in a model repository.

Usage : `Inps GenerateDiagrams [arguments]`

Arguments:

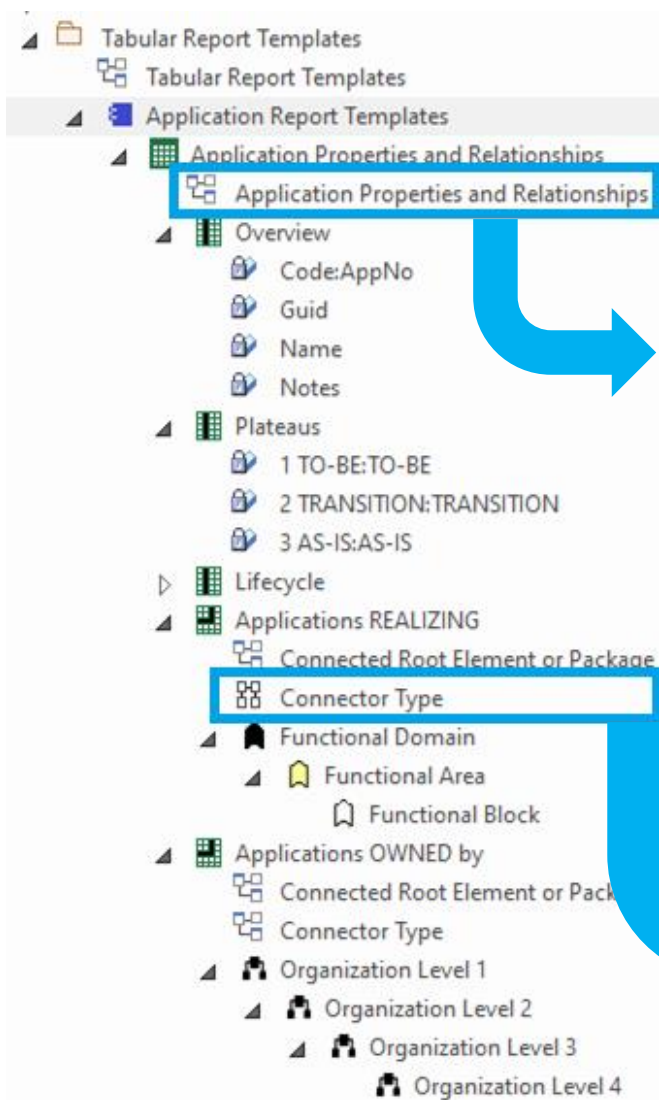
`RepoPathName`: Path name (EAP file) of the repository where the diagrams must be generated.

`GenerationScopeDiagramGUID`: A diagram containing organizations elements and/or a package of enterprise functions for which diagram generation is required.



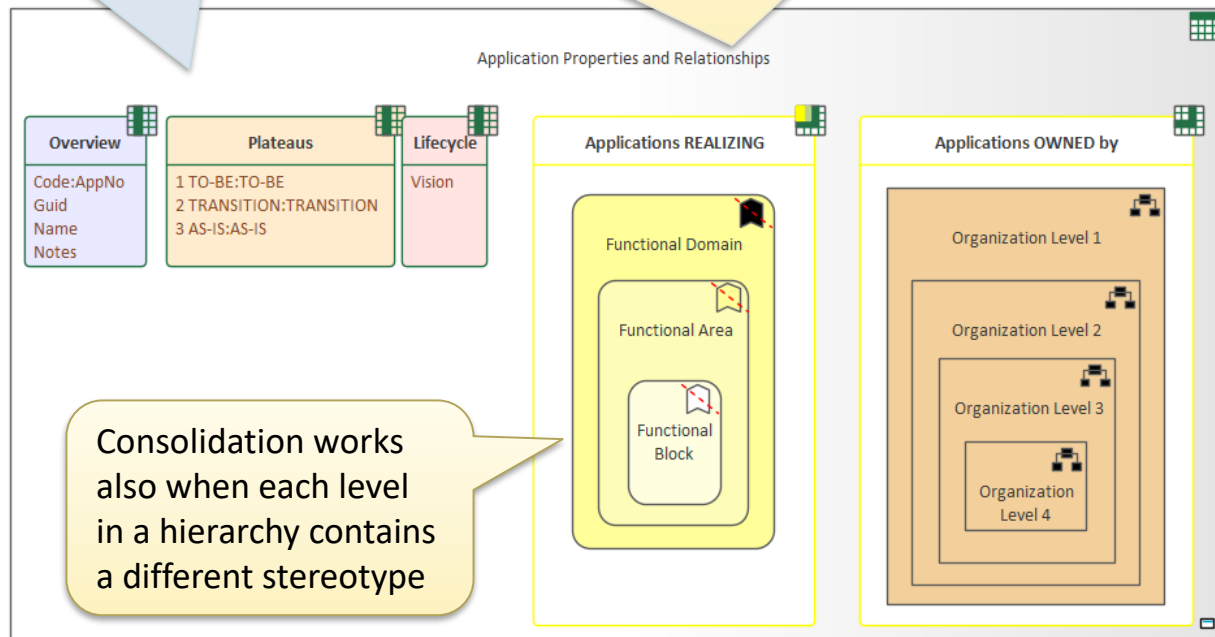


## Model your tabular report

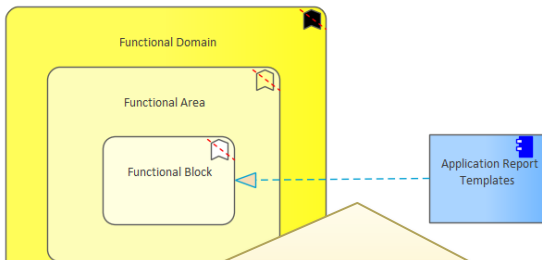


Model element properties and/or tagged values with optional renaming and colored groupings

Model specific connections in specific direction to specific types of elements. Model automatic connection **consolidation** into parent element relationships



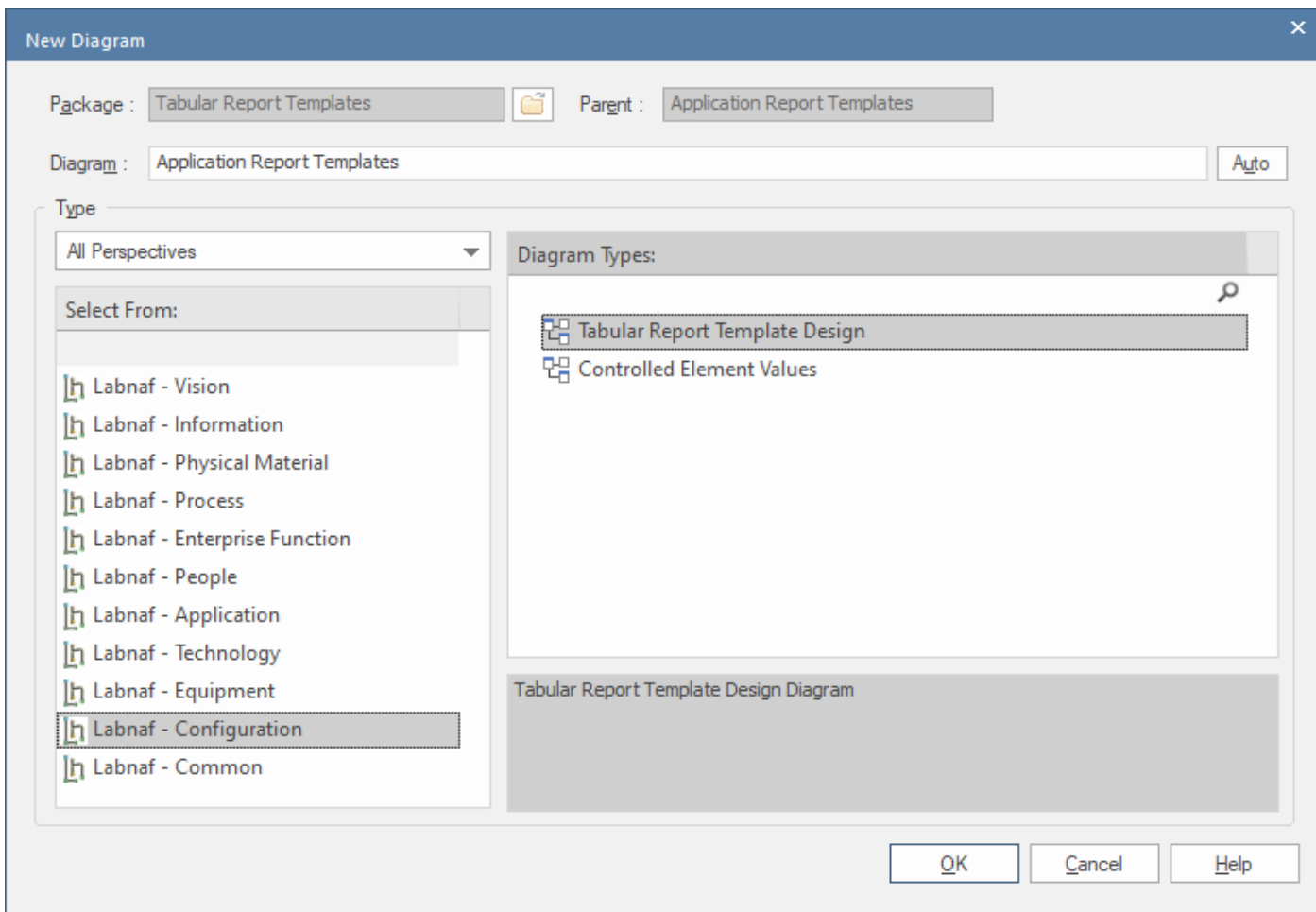
Consolidation works also when each level in a hierarchy contains a different stereotype



Model leaf element **connector** to be reported and consolidated

## Model your tabular report

A tabular report can contain tagged values, properties and connections to any kind of element.



## Toolbox

Search

### Tabular Report

- Tabular Report Template
- Tabular Properties Template
- Tabular Connections Template

### Vision

- Strategic Theme
- Strategic Objective
- Goal
- Standard
- Principle
- Demand
- Epic
- Capability
- Feature
- Story

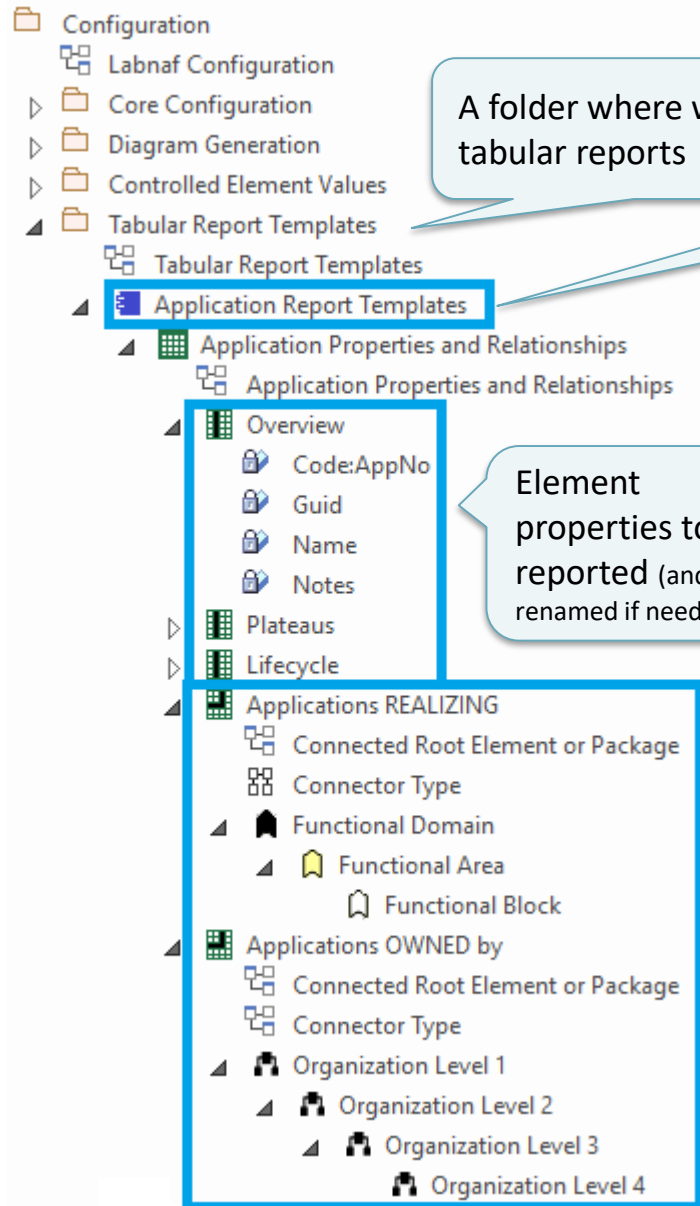
### Information

- Information Domain
- Entity
- Representation
- Data Object

### Process

- Process
- Event
- Junction
- Swimlane
- Activity
- Start Event
- Intermediate Event
- End Event
- Gateway

## Model your tabular report



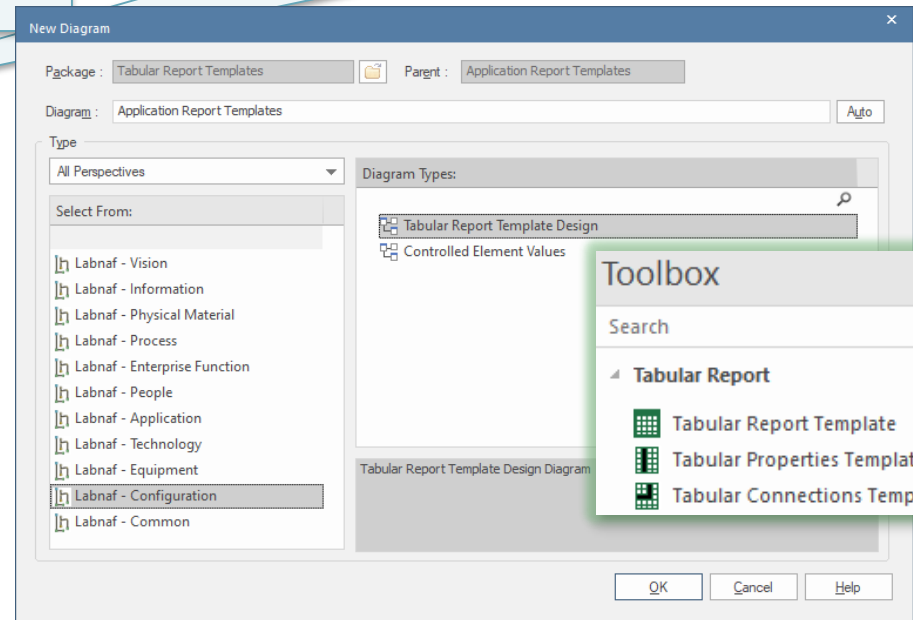
A folder where we define tabular reports

Element properties to be reported (and renamed if needed)

Applications REALIZING

- Connected Root Element or Package
- Connector Type
- Functional Domain
  - Functional Area
  - Functional Block
- Applications OWNED by
  - Connected Root Element or Package
  - Connector Type
  - Organization Level 1
    - Organization Level 2
      - Organization Level 3
      - Organization Level 4

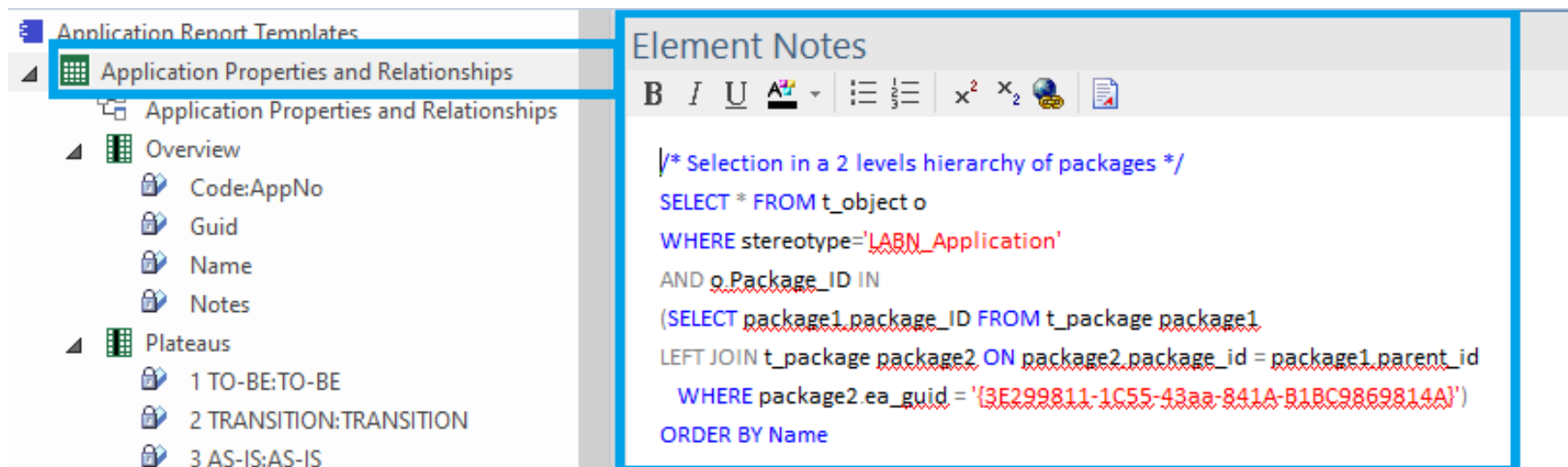
An element prototype for grouping tabular reports  
So we can see that the embedded tabular report definitions are for elements of that specific type and stereotype.



Element connections to be reported and consolidated at several levels of detail.  
Connected element types and stereotypes can be different at each level.

Reported collection of elements (e.g. applications) selected following any kind of rule

## Define the set of elements to be reported



The screenshot shows a software interface with a tree view on the left and a text editor on the right. The tree view is titled 'Application Report Templates' and contains a folder 'Application Properties and Relationships'. Under this folder, there are two sub-folders: 'Overview' and 'Plateaus'. The 'Overview' folder contains four items: 'Code:AppNo', 'Guid', 'Name', and 'Notes'. The 'Plateaus' folder contains three items: '1 TO-BE:TO-BE', '2 TRANSITION:TRANSITION', and '3 AS-IS:AS-IS'. The text editor on the right is titled 'Element Notes' and contains the following SQL query:

```
/* Selection in a 2 levels hierarchy of packages */  
SELECT * FROM t_object o  
WHERE stereotype='LABN_Application'  
AND o.Package_ID IN  
(SELECT package1_package_ID FROM t_package package1  
LEFT JOIN t_package package2 ON package2_package_id = package1_parent_id  
WHERE package2_ea_guid = '{3E299811-1C55-43aa-841A-B1BC9869814A}')  
ORDER BY Name
```

The "Notes" field of the Tabular Report Template contains some SQL SELECT statement.

**That SQL statements selects the elements that need to be included in the report.**

With professional database engines, that SELECT statement can reach a level of sophistication that goes way beyond users' requirements.

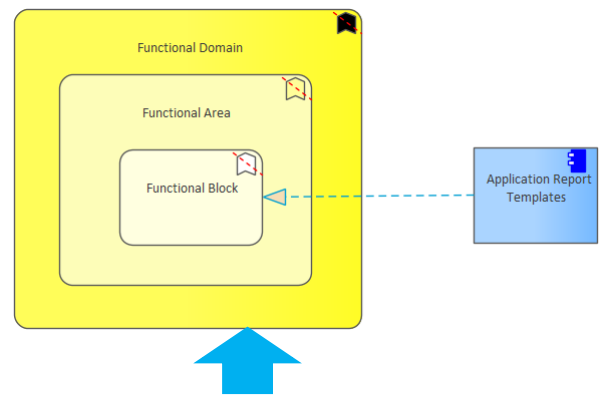
Access databases, on the other hand, have some limitations, but it is still usually sufficient to implement most use cases. Access databases are anyway not designed for running on professional database servers.

# Generate Tabular Report (cont.)

*If you want to limit the scope of the reported connections...*

**Put on a diagram the set of root elements to be selected at the other connection end.**  
The diagram can contain packages and elements.

## Enterprise Function Levels



The screenshot shows a software interface with two main panels. On the left is a tree view titled 'Application Report Templates'. The tree is expanded to show 'Applications REALIZING', which is further expanded to 'Connected Root Element or Package'. Below this, 'Functional Domain' is expanded to show 'Functional Area' and 'Functional Block'. On the right is a large panel titled 'Enterprise Functions' containing a list of 18 items, each with a radio button. The items are: B2C External Exchanges, B2B External Exchanges, Information Management, Internal Exchange, Strategy, Legal Affairs, Risk & Compliance, Traveler Communication, Marketing & Sales, Traffic Management, Station Management, Bus Route Planning, Bus Maintenance, Security, HR & Corporate services, Finance, Supply Chain, and Information Technology. A blue border highlights the 'Enterprise Functions' panel.

## To manually start tabular report generation:

```
Command: GenerateTabularReports
```

```
Description: Generate spreadsheets from a model repository based on configuration stored in that same repository.
```

```
Usage : lnps GenerateTabularReports [arguments]
```

```
Arguments:
```

```
    SourceRepoPathName: Path name of the source model repository (EAP file).
```

```
    OutputDirectoryPath: Directory path name where the spreadsheets must be generated. The name of each spreadsheet file is the name of the template report.
```

```
    [ElementPrototypeName]: The name of a specific element prototype name for which all embedded tabular report templates must be applied.
```

```
    [TabularReportTemplateName]: The name of a specific tabular report template to be applied.
```

By default, all report templates will be applied.

But you can also be specific.

# Generate Doc

(Word, RTF, PDF)

## To manually start document generation:

Command: `GenerateDoc`

Description: Generate a Word, RTF or PDF document from a model repository package.

Usage : `lnps GenerateDoc [arguments]`

Arguments:

`SourceRepoPathName`: Path name of the source model repository (EAP file).

`OutputPath`: Path name of the document file to be generated.

The file extension specified will determine the format of the generated document - for example, RTF, PDF

`PackageGuid`: The GUID of the package or master document to run the report on.

`TemplateName`: The document report template to use; if the PackageGUID has a stereotype of MasterDocument, the template is not required.

# Generate Html

## To manually start HTML generation:

Command: `GenerateHTML`

Description: Generate an HTML web site from a model repository package.

Usage : `Inps GenerateHTML [arguments]`

Arguments:

`SourceRepoPathName`: Path name of the source model repository (EAP file).

`OutputPath`: The path of the file system folder where the HTML pages must be generated.

`SourcePackageGUID`: The GUID of the repository package for which HTML must be generated.

`[WebSiteTemplateName]`: The optional name of a web style template used for HTML generation (default=Sparx EA default template).

On the web site, you can email a stable link to the current page by clicking on the little envelope.





# Backup To Access File

To manually start the backup to an Access file:

```
Command: BackupToAccessFile
```

```
Description: Backup a DBMS or Access repository to an Access Repository.
```

```
Usage : lnps BackupToAccessFile [arguments]
```

```
Arguments:
```

```
SourceRepoPathName: Path name of the source repository (EAP file).
```

```
DestEapPathName: Path name of the destination Access repository (EAP file).
```

```
LogFilePath: Path name of the log file name.
```

SourceRepoPathName (EAP) must point to a DBMS repository

# Schedule Command

To schedule a **nightly** command **starting at midnight**:

- **InitialStartTime = 00:00:00** Don't schedule 2 commands starting exactly at the same time
- **PeriodAsMinutes = 1440** There are 1440 minutes in a day

## To manually start the scheduler:

```
Command: ScheduleCommand
```

```
Description: Schedule a task to run periodically starting at a specific time.
```

```
Usage : Inps ScheduleCommand [arguments]
```

```
Arguments:
```

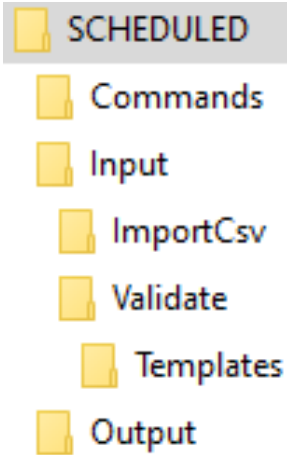
```
  CommandPathName: Path name of the command that needs to be periodically started (.cmd or .bat).
```

```
  InitialStartTime: The initial start time for the task (HH:MM:SS).
```

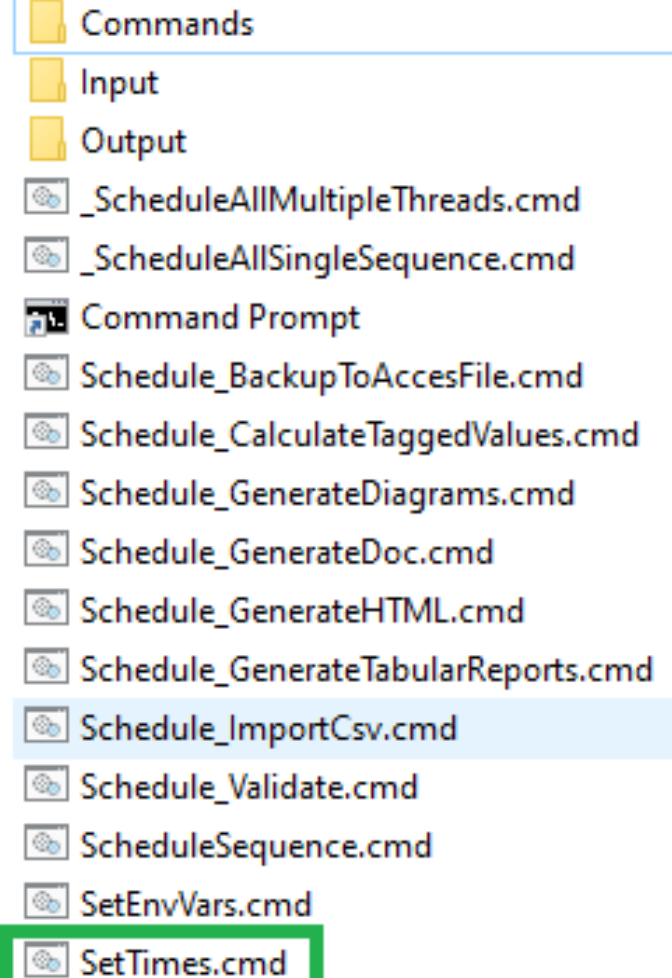
```
  [PeriodAsMinutes]: The length of a period expressed in minutes.
```

# Final Recommendation

Use the preconfigured batches and settings



Name



```
SetTimes.cmd
1  REM -- SINGLE START TIME --
2  Set StartTime_AllSingleSequence=00:00:00
3
4
5  REM -- SPECIFIC START TIME FOR EACH TASK --
6
7  Set StartTime_Cleanup_BackupToAccessFile=22:00:00
8  Set StartTime_Cleanup_GenerateHTML=22:00:05
9
10 Set StartTime_ImportCsv=22:30:00
11
12 Set StartTime_CalculateTaggedValues=23:00:00
13 Set StartTime_GenerateDiagrams=23:30:00
14
15 Set StartTime_BackupToAccessFile=00:00:00
16 Set StartTime_Validate=01:00:00
17
18 Set StartTime_GenerateTabularReports=02:00:00
19 Set StartTime_GenerateDoc=02:30:00
20 Set StartTime_GenerateHTML=03:00:00
21
22
23 REM -----
24
25 set SCHEDULED_MINUTES_UNTIL_RESTART=1440
```

# Labnaf PowerShell

## Command Compatibility Matrix

Power Shell Commands	Sql Server	Pro Cloud Server	Access
ImportFromCsv	X	X	X
BackupToAccessFile	X	X	
GenerateTabularReports	X	X	X
CalculateValues	X	X	
GenerateDoc (Word, RTF, PDF)	X		
GenerateHtml	X		
GenerateDiagrams	X	X	X
Validate	X	X	X
ScheduleTask	X	X	X