The Labnaf Strategy & Architecture Framework

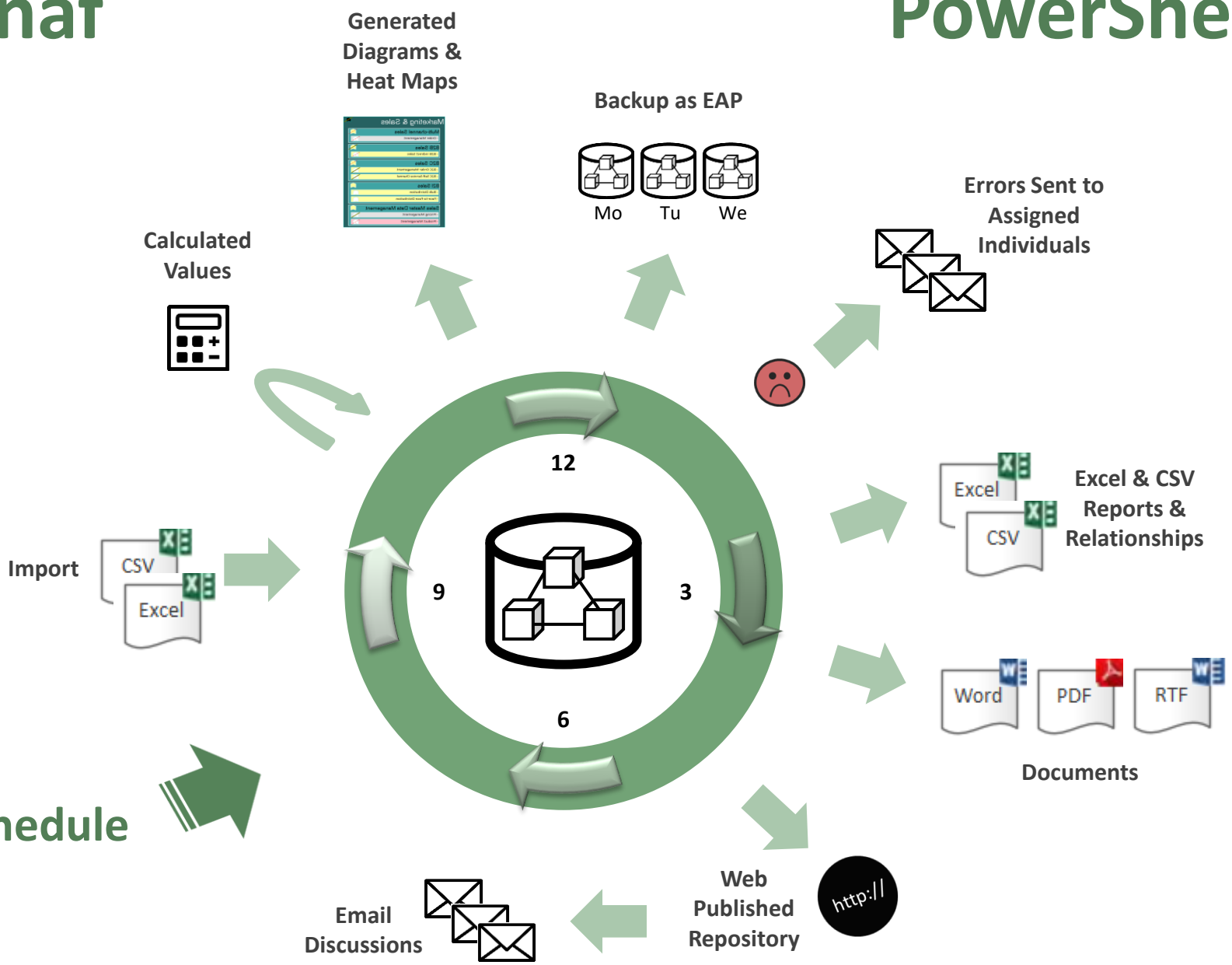
Labnaf PowerShell

User Guide

© 2014-2021 Alain De Preter - All Rights Reserved

Labnaf

PowerShell



© 2014-2021 Alain De Preter - All rights reserved

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

- Import Tabular Report (Excel, CSV)
- Calculate Values
- Validate and send emails to assigned individuals
- Generate Diagrams
- Generate Tabular Report (Excel, CSV), Doc (Word, RTF, PDF), Html
- Auto Connectors Generate / Delete
- Backup To Access File
- Schedule Command (not only Labnaf PowerShell commands)

© 2014-2021 Alain De Preter - All Rights Reserved

Running commands on the command line

lnps

lnps {command name}

```
Usage : lnps Command [arguments]

Available Commands:

   AutoConnectorsDelete

   AutoConnectorsGenerate

   BackupToAccessFile

   CalculateTaggedValues

   GenerateDiagrams

   GenerateDoc

   GenerateHTML

   GenerateTabularReports

   ImportTabularReport

   ScheduleCommand

   Validate
```

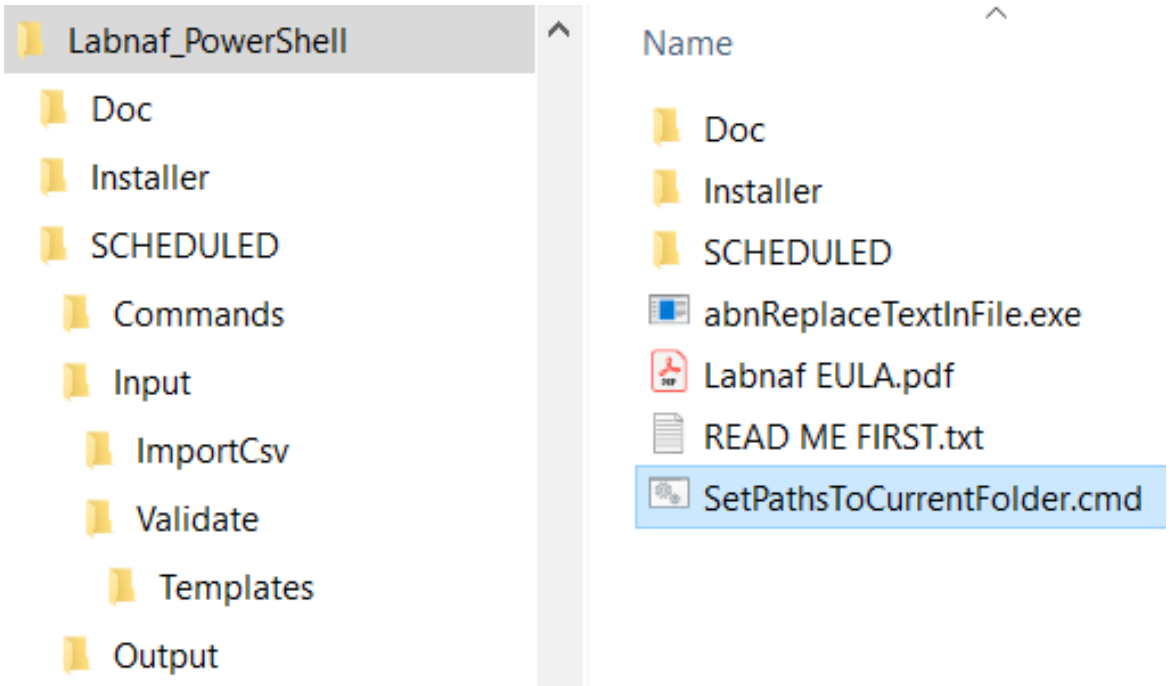
Example: if you type « lnps GenerateTabularReports » you get the following info.

```
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.
```


© 2014-2021 Alain De Preter - All rights reserved

Automatic configuration

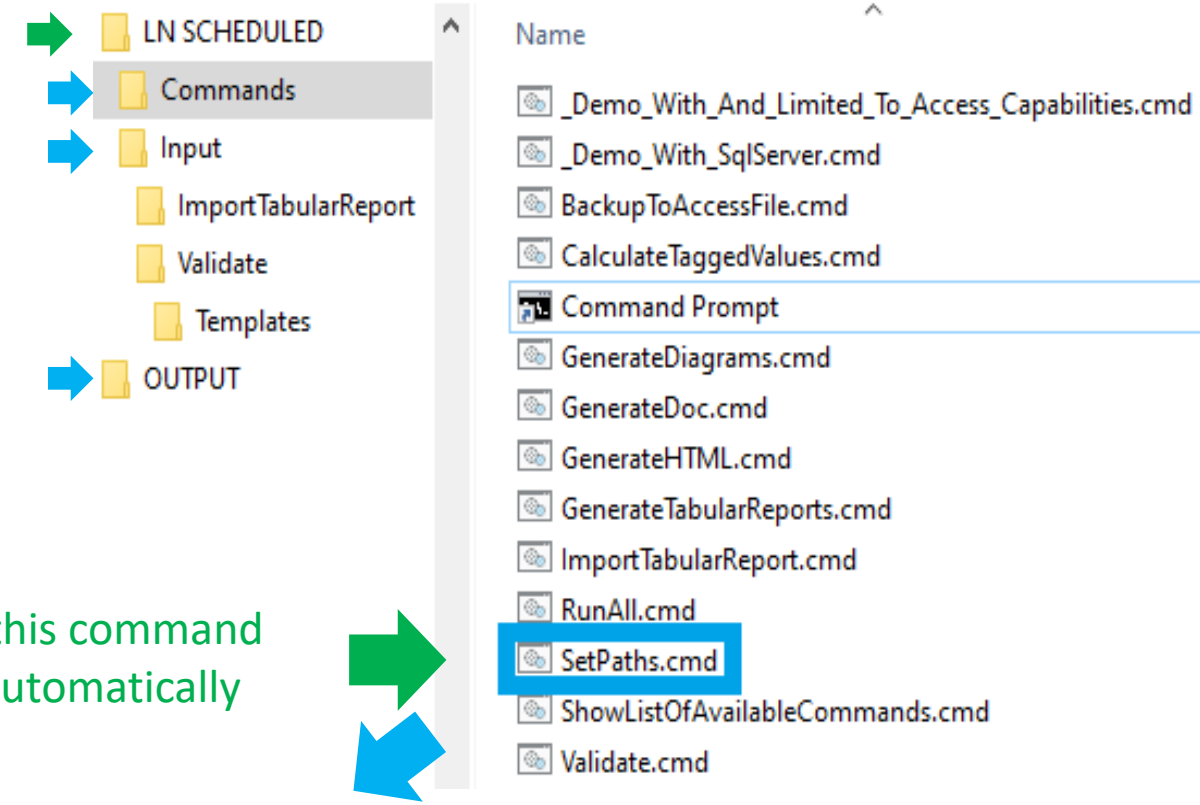
- Copy the Labnaf_PowerShell folder anywhere you want on your file system
- Double-click on “SetPathsToCurrentFolder.cmd”



This updates the Labnaf PowerShell configuration files following the “Labnaf_PowerShell” folder location.

© 2014-2021 Alain De Preter - All Rights Reserved

Preconfigured batches calling commands with predefined settings



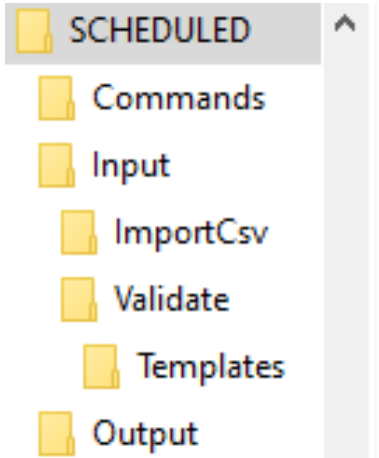
The paths in this command were reset automatically

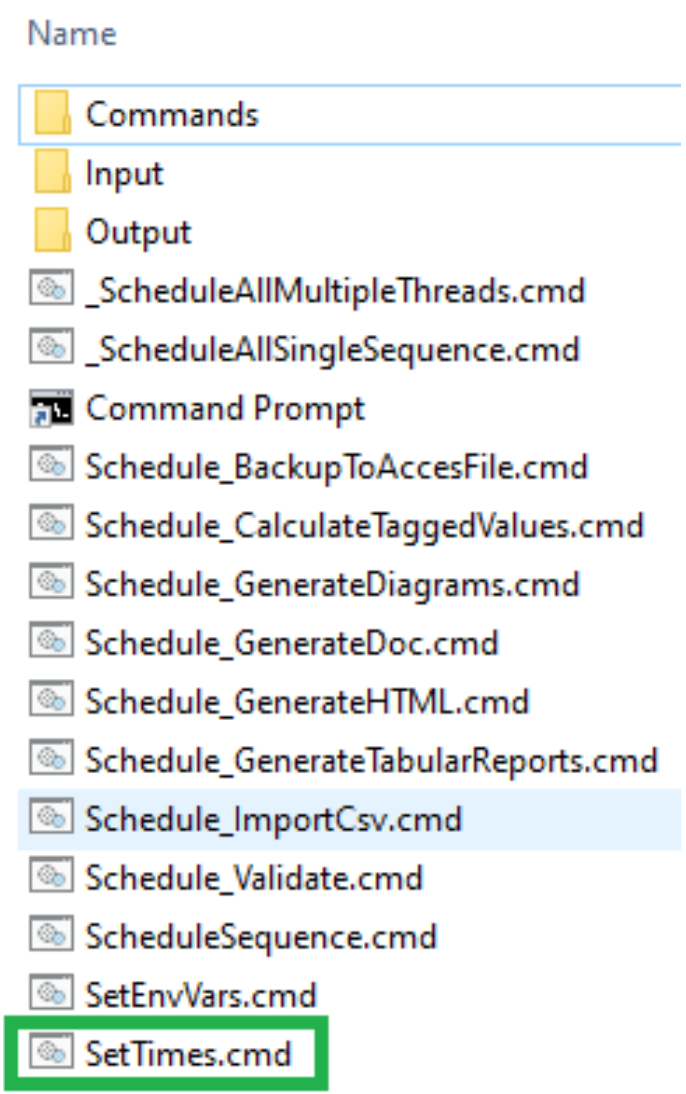
SetPaths.cmd

```
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
```


© 2014-2021 Alain De Preter - All Rights Reserved

Preconfigured Command Scheduling





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_BackupToAccesFile=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
```


© 2014-2021 Alain De Preter - All Rights Reserved

ImportTabularReport

File to be imported can be .CSV or .XLSX (Excel)

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

OPTIONAL repository column mappings are stored in a .CSV file

	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

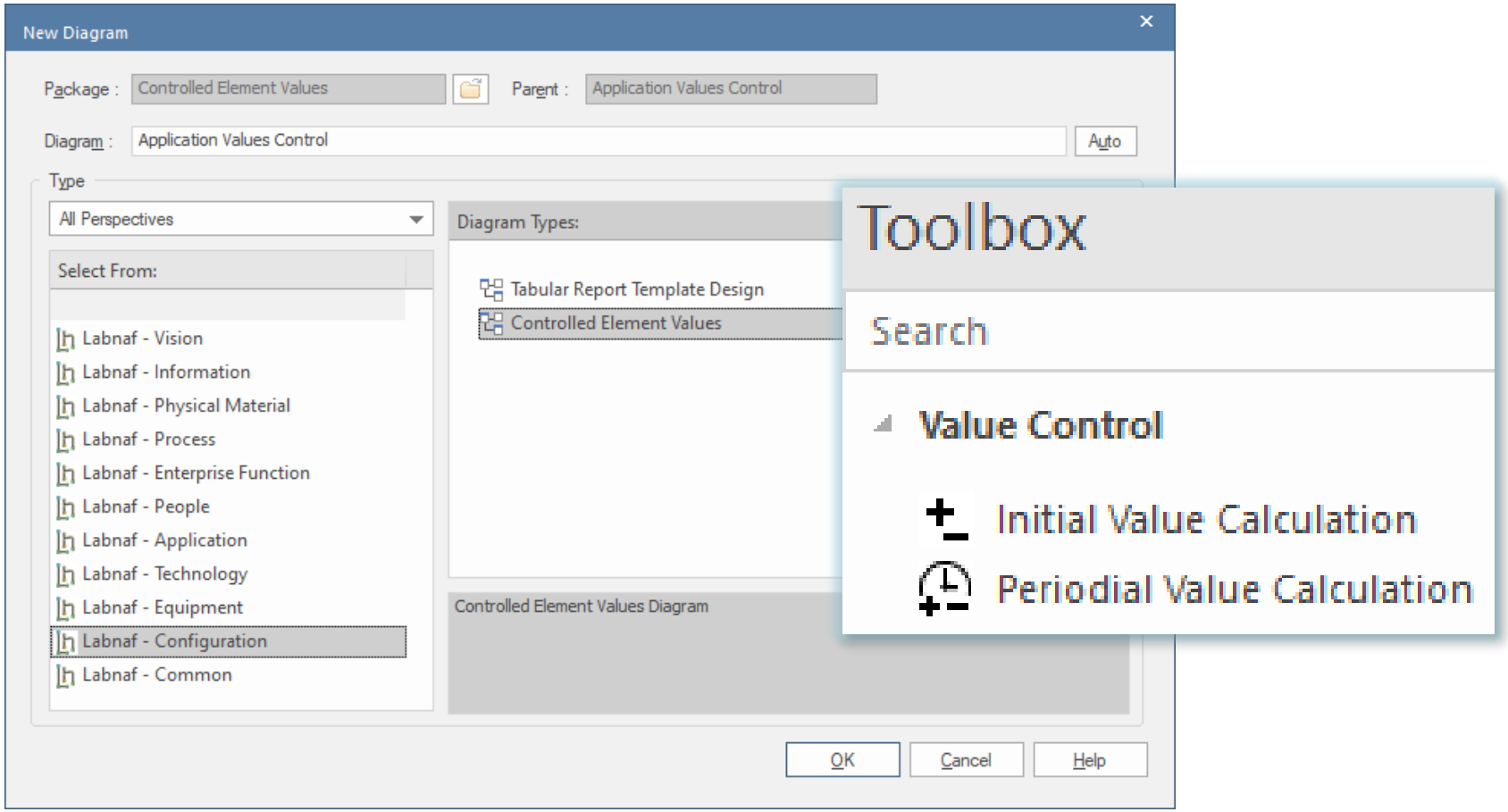
To start the import

```
C:\A\LT\SparxDev\VSProjects\Alain\Labnaf\PowerShell\bin\Debug>lnps ImportTabularReport

Command: ImportTabularReport

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

Usage : lnps ImportTabularReport [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 namesor '-' if all input and output column names are the same.

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


© 2014-2021 Alain De Preter - All rights reserved

Calculate Tagged Values

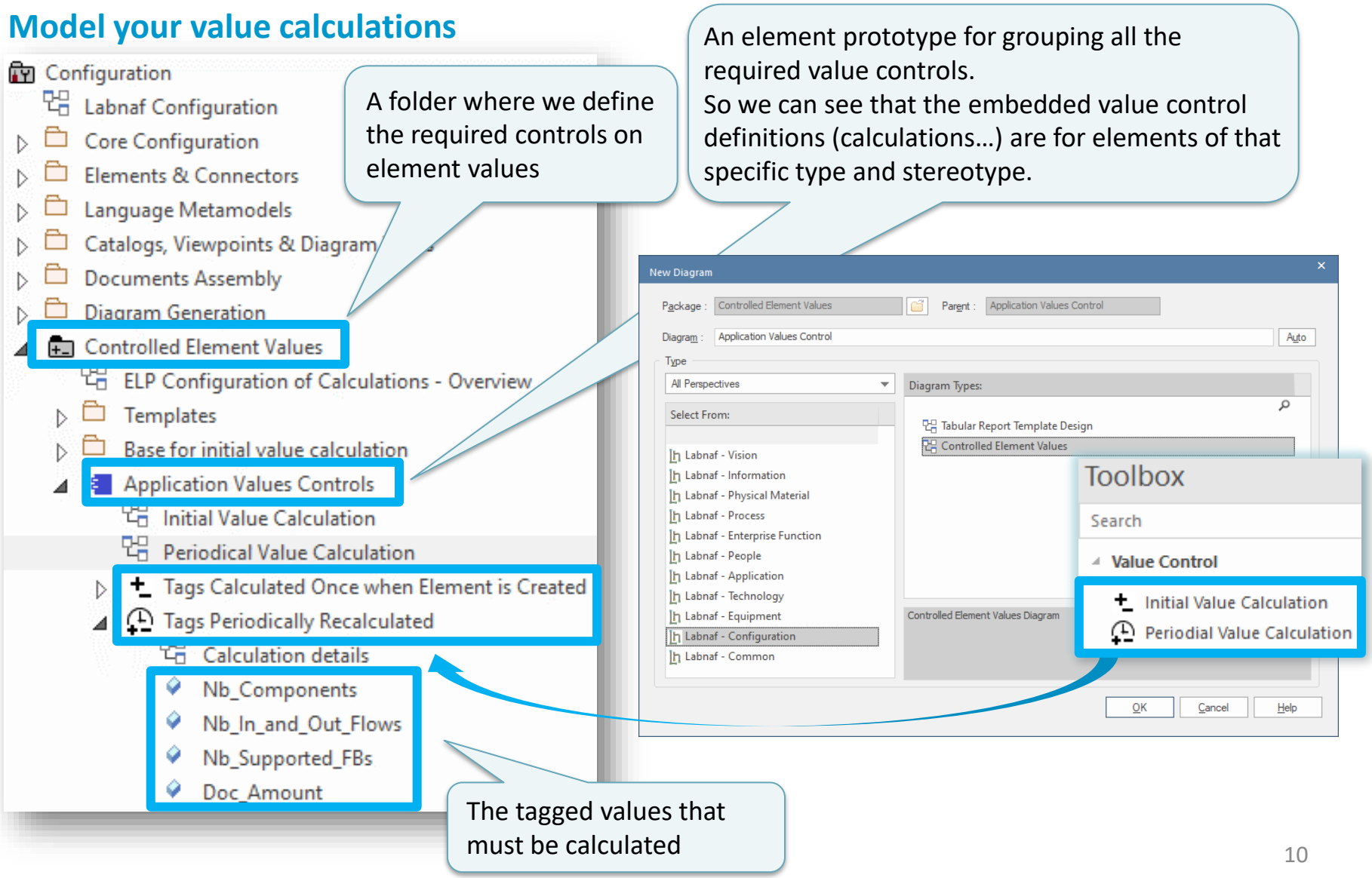
Model your value calculations



© 2014-2021 Alain De Preter - All rights reserved

Calculate Tagged Values (cont.)

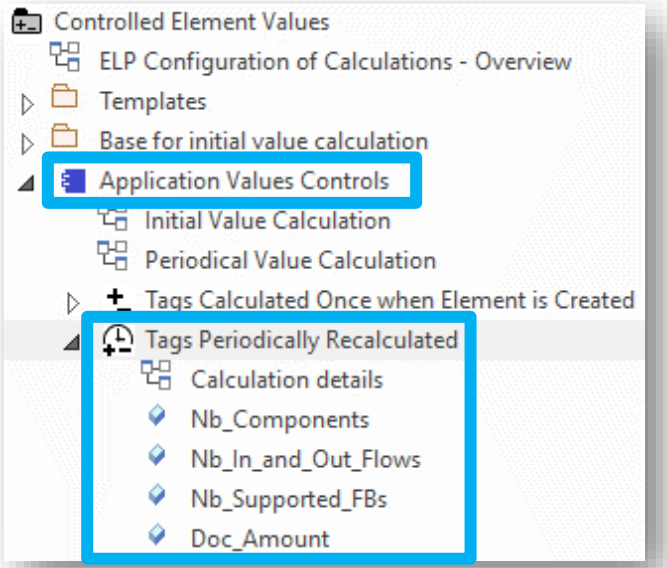
Model your value calculations



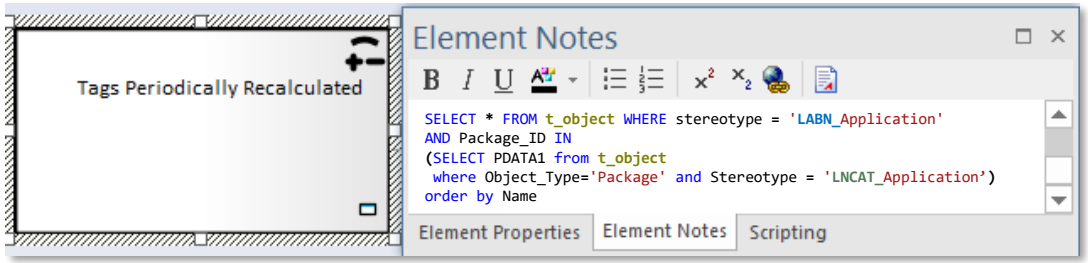
© 2014-2021 Alain De Preter - All rights reserved

Calculate Tagged Values (cont.)

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

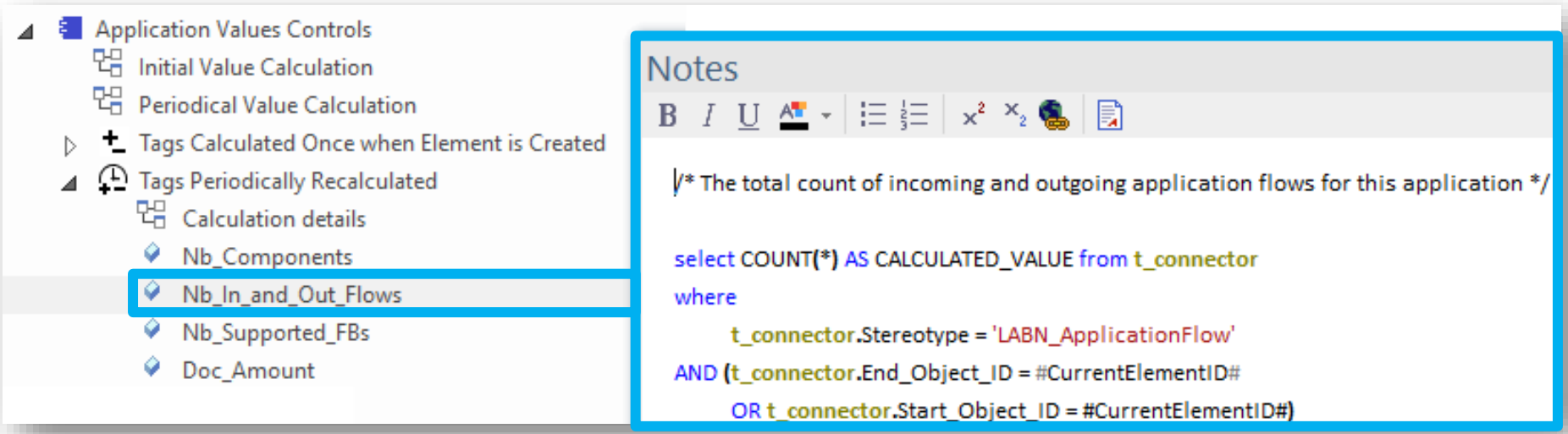


2. Scope (optional): Which elements need to be updated?



By default, all elements with the same stereotype "LABN_xxx" as the element prototype are selected from the related catalog with stereotype "LNCAT_xxx".

3. Calculation: How shall we calculate the value?



© 2014-2021 Alain De Preter - All rights reserved

Calculate Tagged Values (cont.)

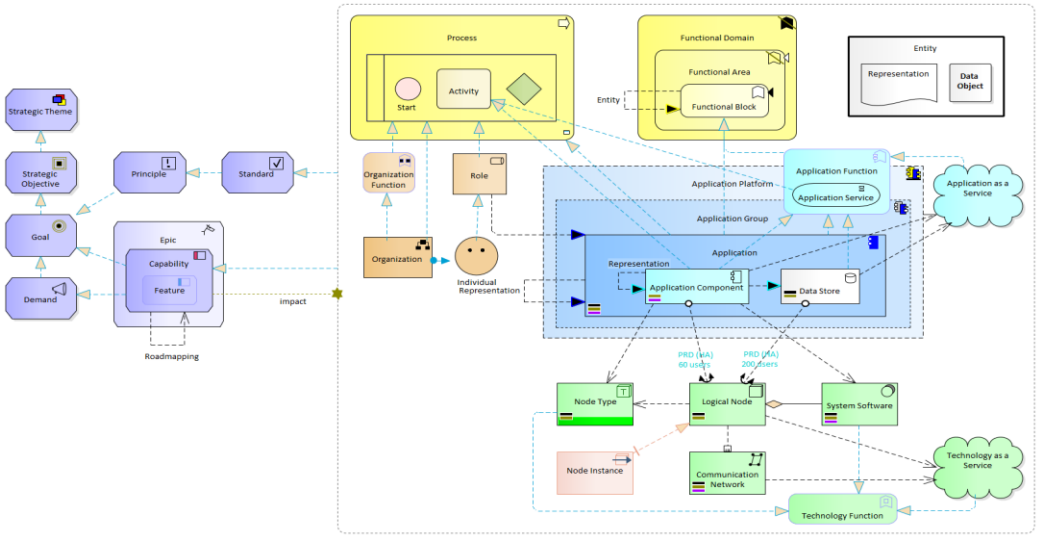
4. To start calculation:

```
Command: CalculateTaggedValues

Description: Calclute 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 aspecific tagged value that must be calculated.
```


© 2014-2021 Alain De Preter - All rights reserved

Validate

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



While Modeling

Existing Invalid Connectors

Prevent creation of invalid connectors

Labnaf PowerShell

Send Error Emails to Relevant Recipients

© 2014-2021 Alain De Preter - All rights reserved

Why do we need periodical validation?

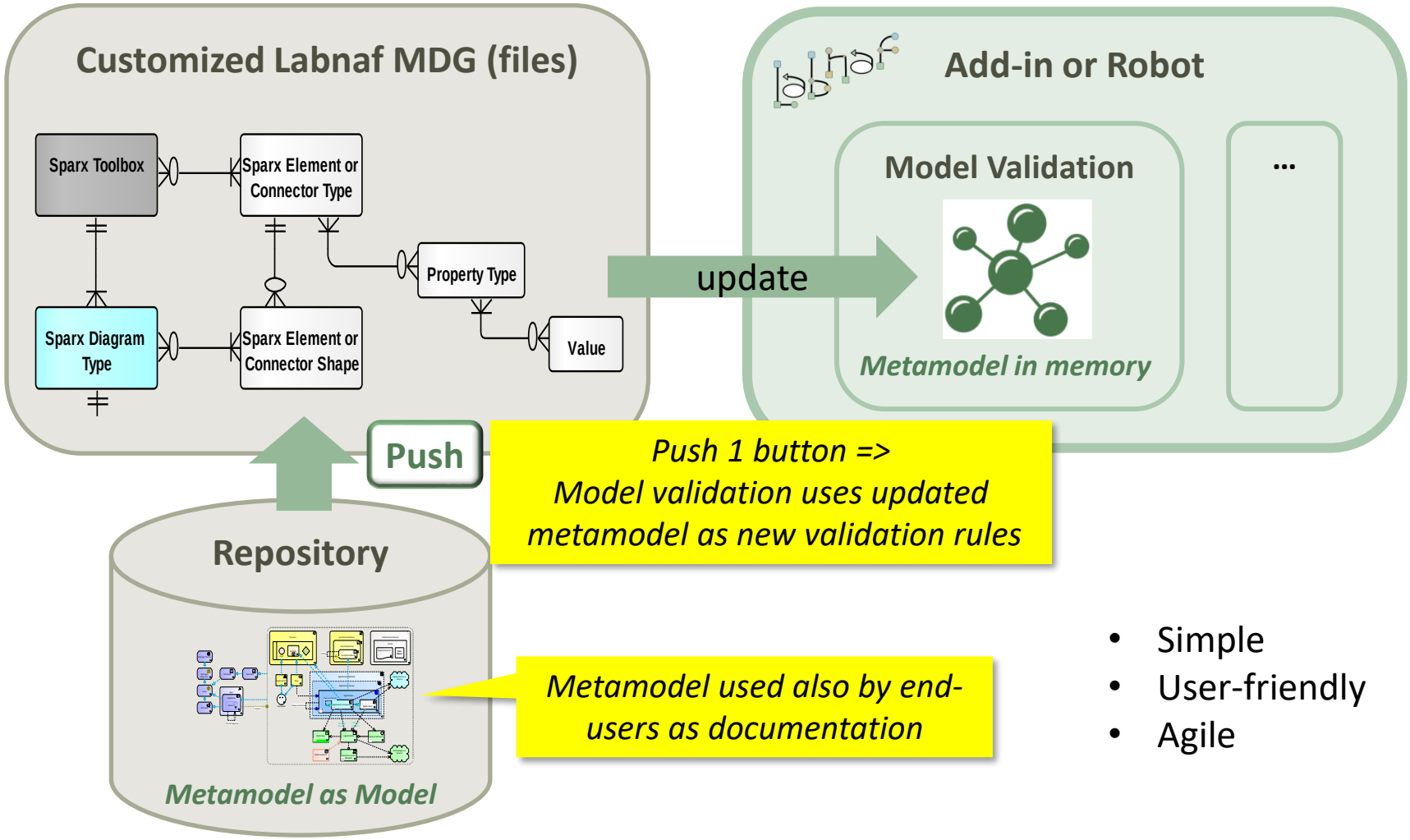
How could we have errors if we prevent users from entering errors?

- 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.

© 2014-2021 Alain De Preter - All rights reserved

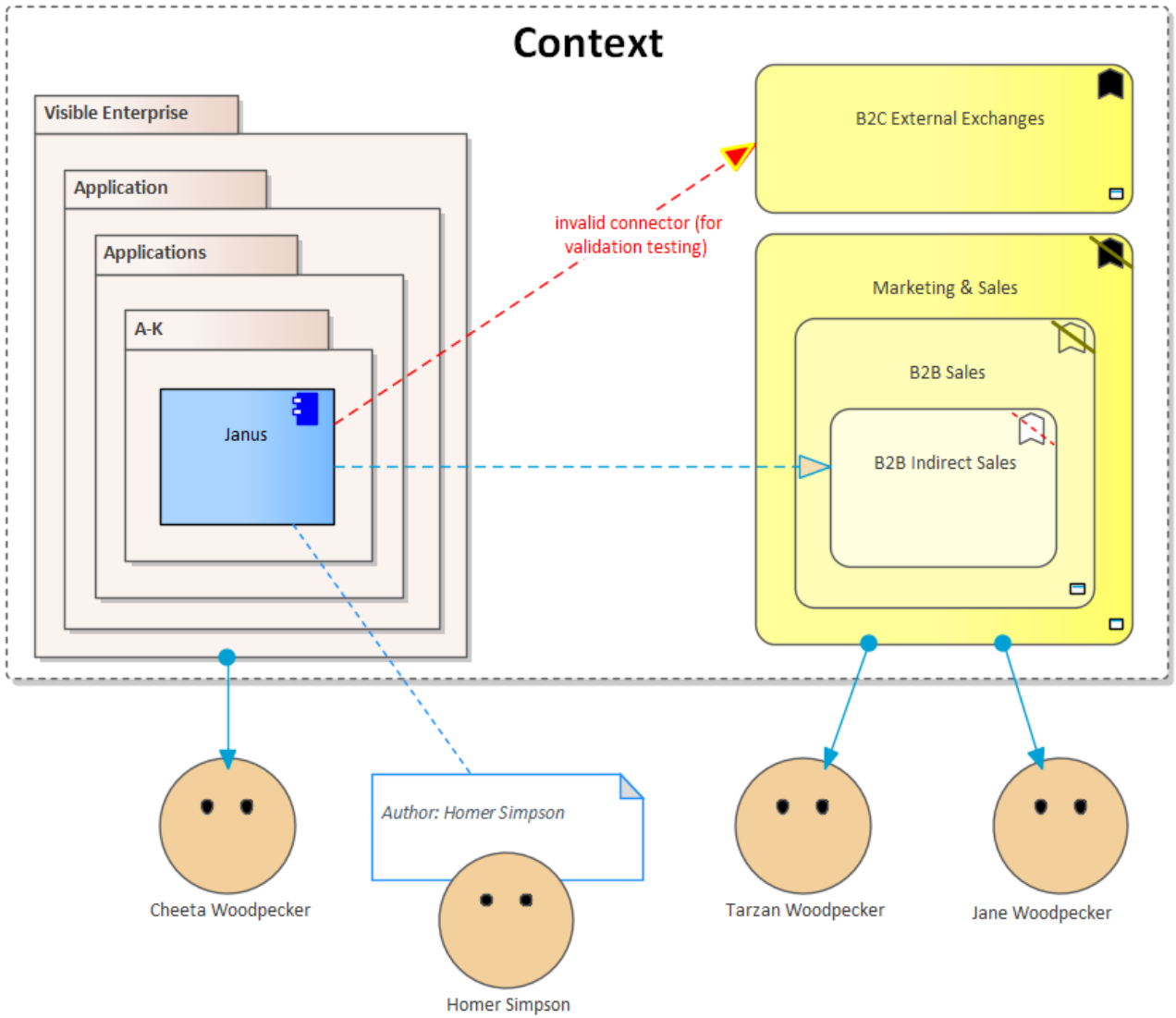
The default metamodel can be **very** easily updated: One click on a connection in your production repository.



- Simple
- User-friendly
- Agile

© 2014-2021 Alain De Preter - All rights reserved

Error message routing is based on architecture management assignments



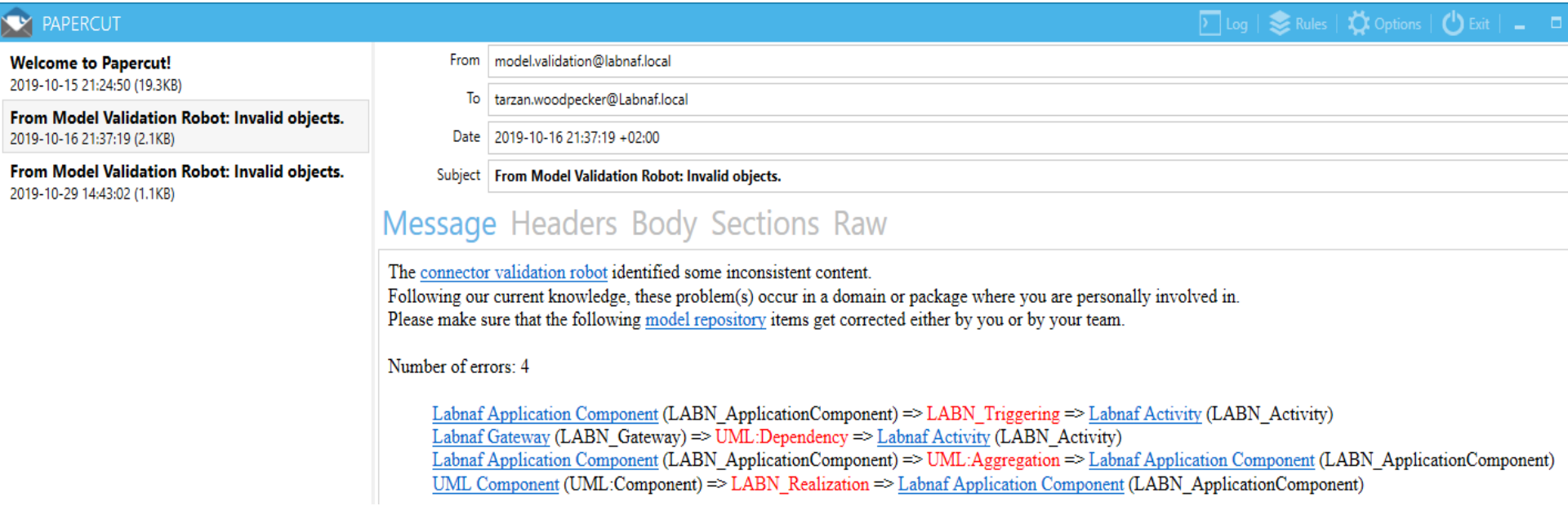
Error message distribution depends on

- 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**

© 2014-2021 Alain De Preter - All rights reserved

Sample error message sent to an assigned individual



© 2014-2021 Alain De Preter - All rights reserved

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 WHERE stereotype like 'LABN_%' AND Package_ID IN (SELECT PDATA1
from t_object where Object_Type='Package' and Stereotype like 'LNCAT_%') ORDER BY Name
  </SelectElementsInScope>
  <Sender>model.validation@labnafdemo.com</Sender>
  <SendTo>
    <FirstAvailableAlternativeOnly>true</FirstAvailableAlternativeOnly>
    <PeopleAssignedToDomain>true</PeopleAssignedToDomain>
    <AuthorDuringMonthsAfterElementCreated>120</AuthorDuringMonthsAfterElementCreated>
    <PeopleAssignedToPackage>true</PeopleAssignedToPackage>
    <DefaultEmailAddresses>lisa.simpson@labnaf.local</DefaultEmailAddresses>
  </SendTo>
  <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.

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

© 2014-2021 Alain De Preter - All rights reserved

Smtp Server configuration is straightforward

```
<?xml version="1.0" encoding="utf-8"?>
<SmtpServerConfiguration xmlns:xsi="http://w
  <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 start validation:

```
C:\A\LT\SparxDev\VSProjects\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.
```


© 2014-2021 Alain De Preter - All rights reserved

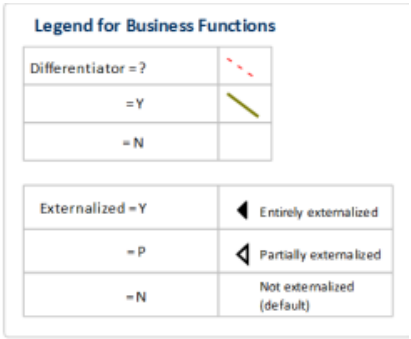
Generate Diagrams

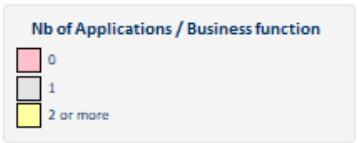
Sample Results

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

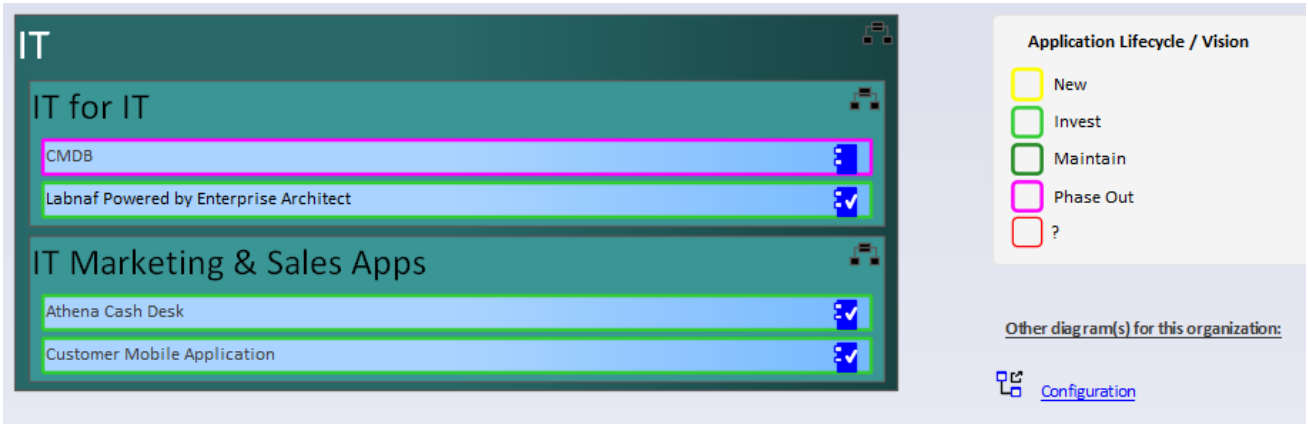




Other diagram(s) for this enterprise function:

FULSales

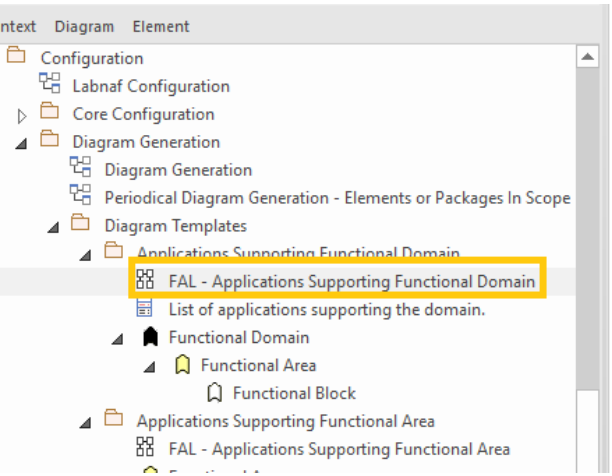
Applications managed by organizations

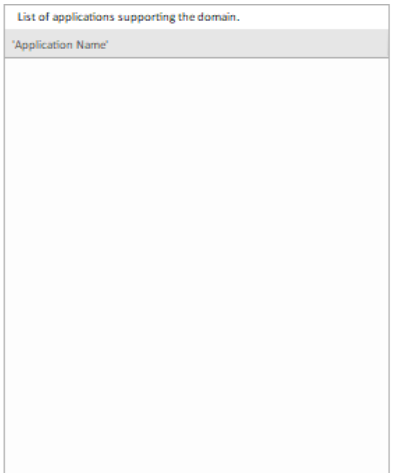


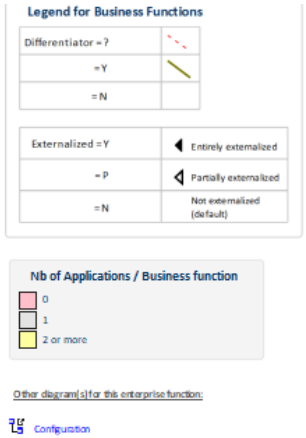
© 2014-2021 Alain De Preter - All rights reserved

Sample Diagram Templates

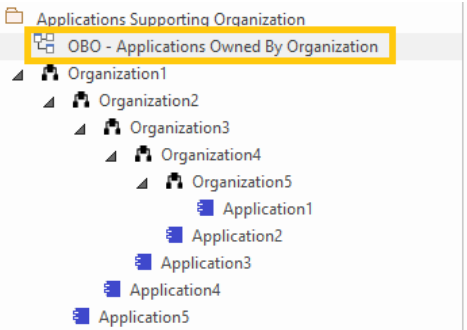
Enterprise Function Taxonomy & Applications Supporting Level 1

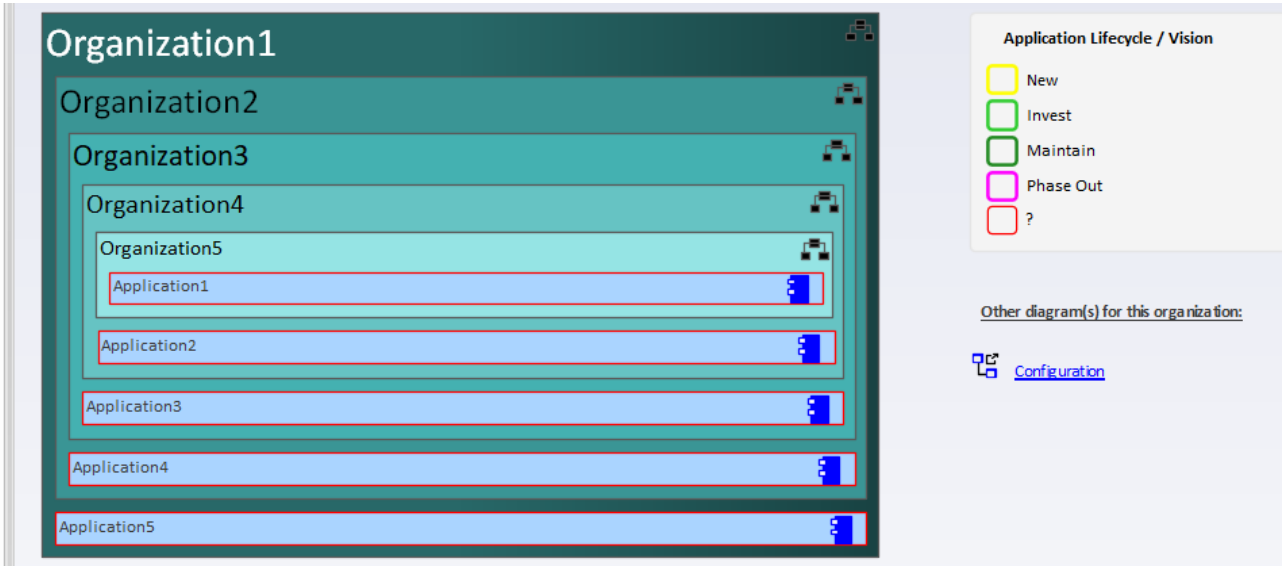






Applications managed by organizations

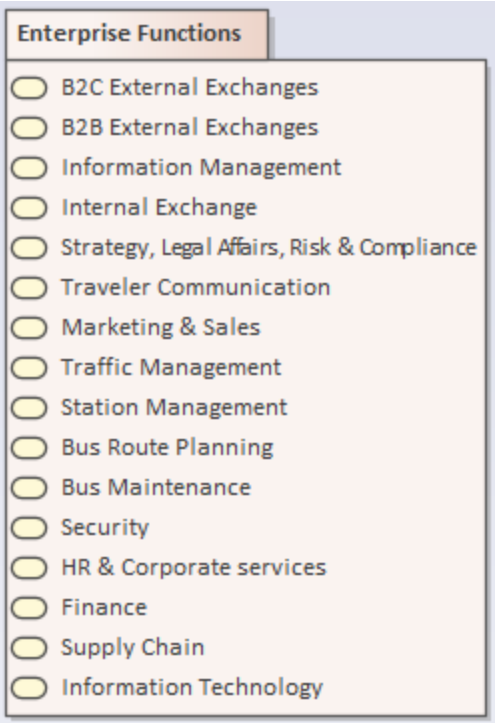




© 2014-2021 Alain De Preter - All rights reserved

Items in Scope for Generation

- 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
        - List of applications supporting the domain.
        - Functional Domain
          - Functional Area
            - Functional Block
      - Applications Supporting Functional Area
        - FAL - Applications Supporting Functional Area
        - Functional Area
          - Application2
          - Functional Block
            - Application1
      - Applications Supporting Organization
        - OBO - Applications Owned By Organization
        - Organization1
          - Organization2
            - Organization3
              - Organization4
                - Organization5
                  - Application1
                - Application2
              - Application3
            - Application4
          - Application5



This is optional

By default, all elements are selected from the related catalog

© 2014-2021 Alain De Preter - All rights reserved

Generate Diagrams (cont.)

To start diagram generation:

```
Command: GenerateDiagrams

Description: Generate diagrams in a model repository.

Usage : lnps 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.
```


© 2014-2021 Alain De Preter - All rights reserved

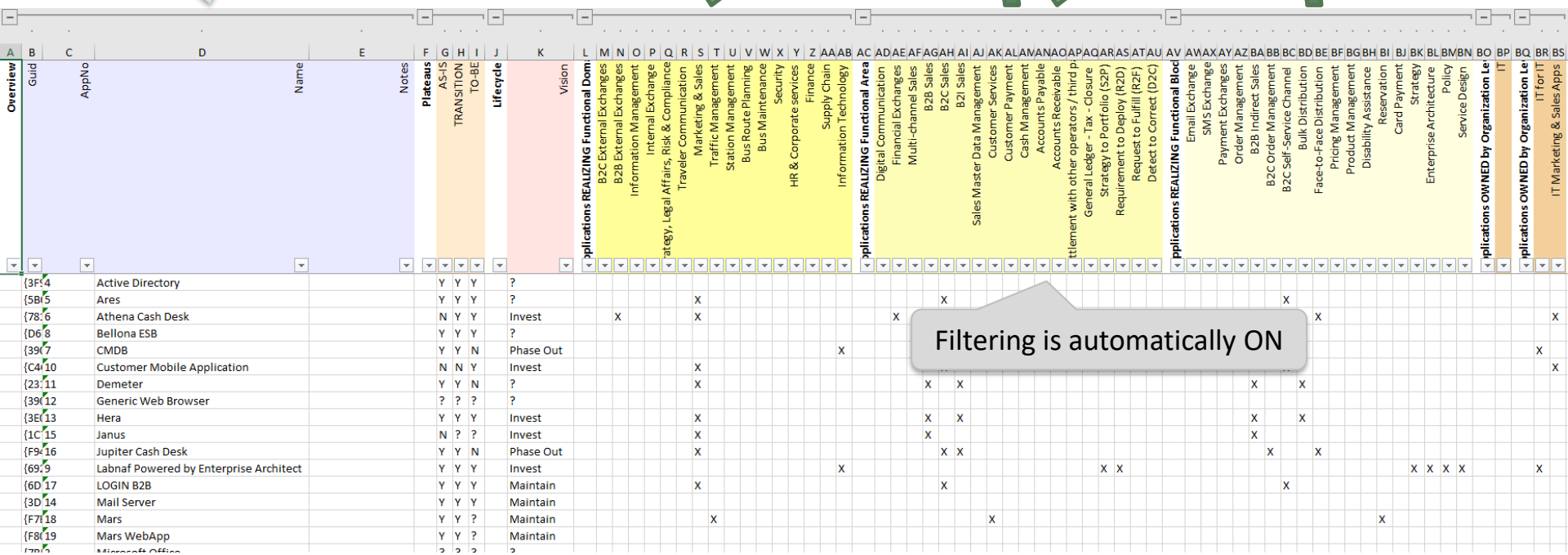
Generate Tabular Reports

Sample Result

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

Element properties and/or tagged values with optional renaming and colored groupings

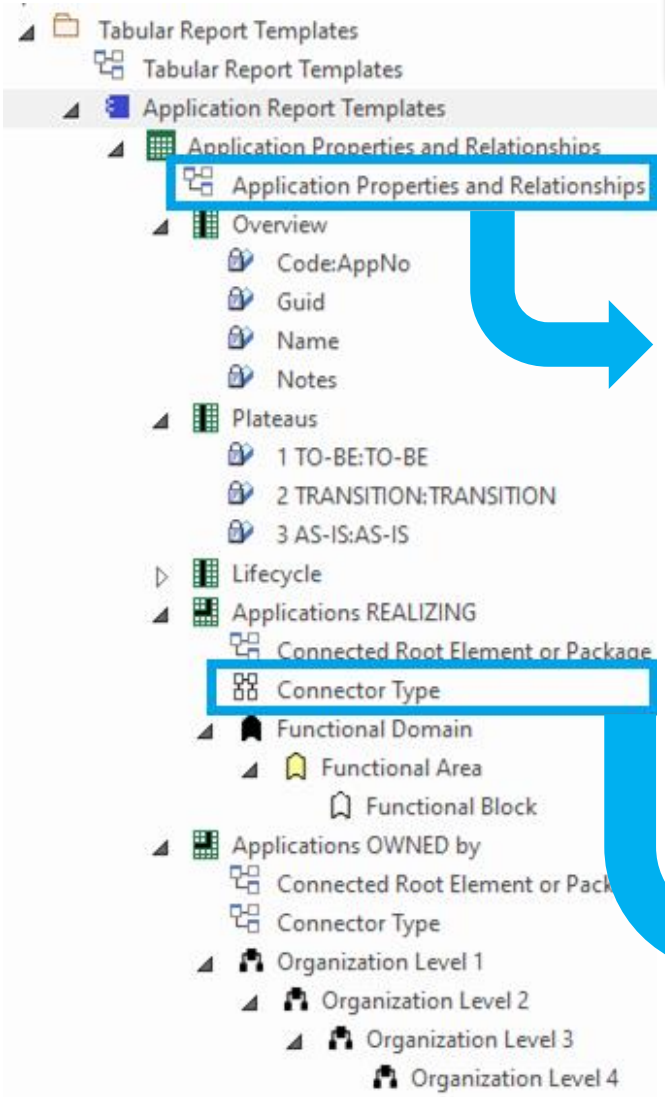
Specific connections in specific direction to specific types of elements.
Automatic connection **consolidation** into parent element relationships.



© 2014-2021 Alain De Preter - All rights reserved

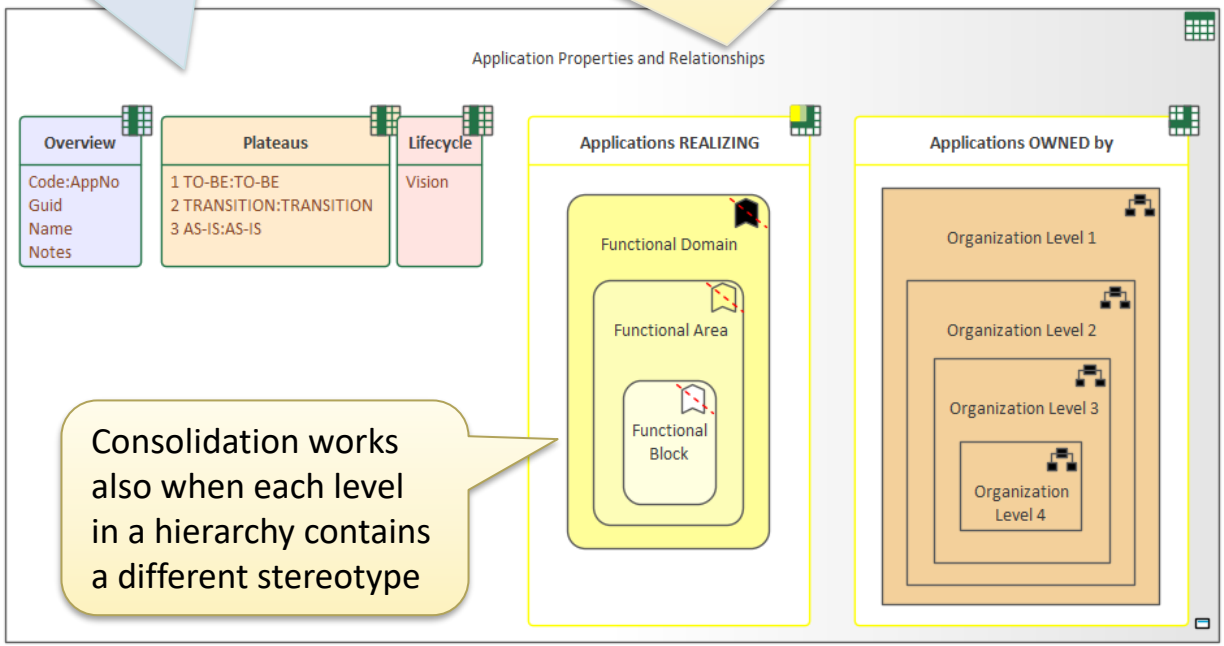
Generate Tabular Report (cont.)

Model your tabular report

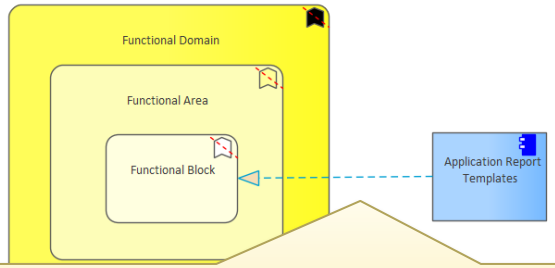


Model element properties and/or tagged values with optional renaming and colored groupings

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



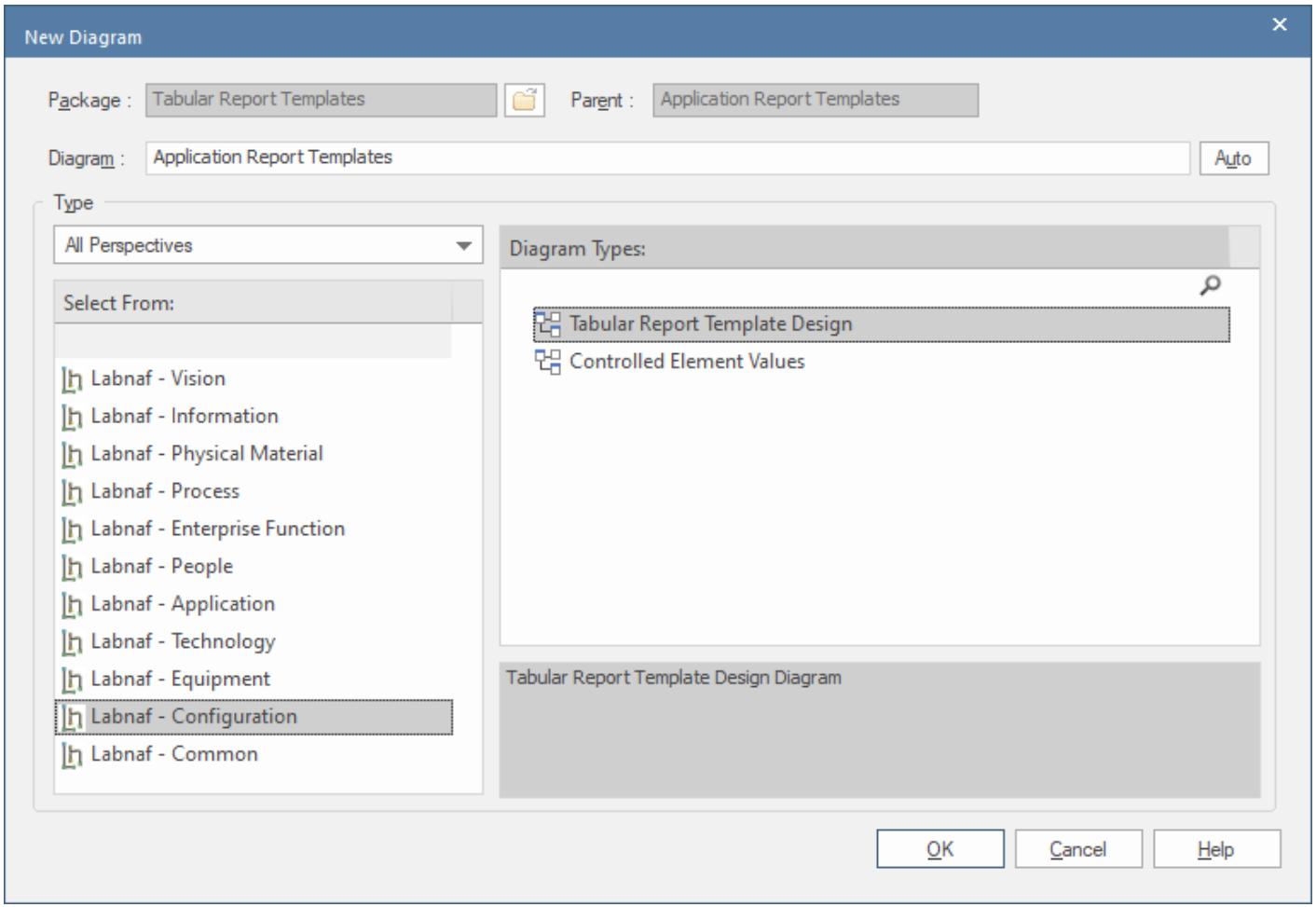
Model leaf element **connector** to be reported and consolidated

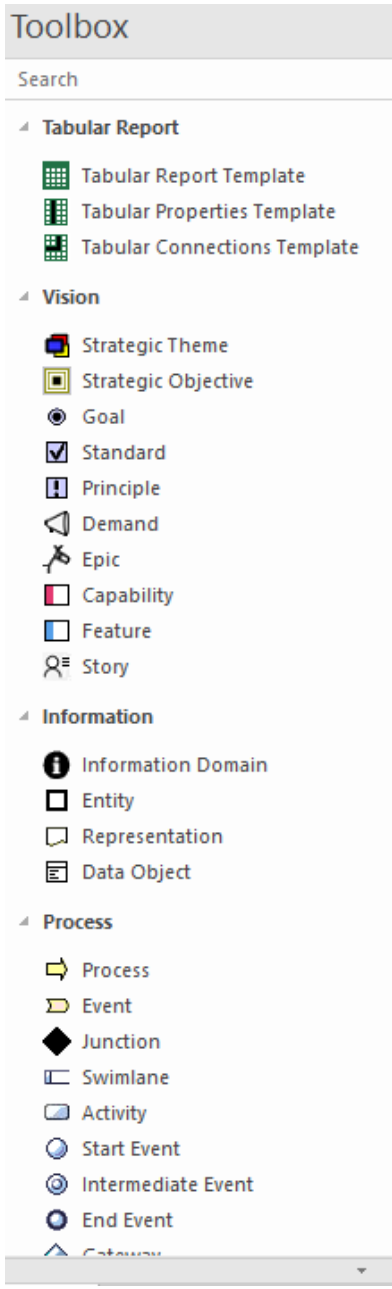
© 2014-2021 Alain De Preter - All rights reserved

Generate Tabular Report (cont.)

Model your tabular report

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

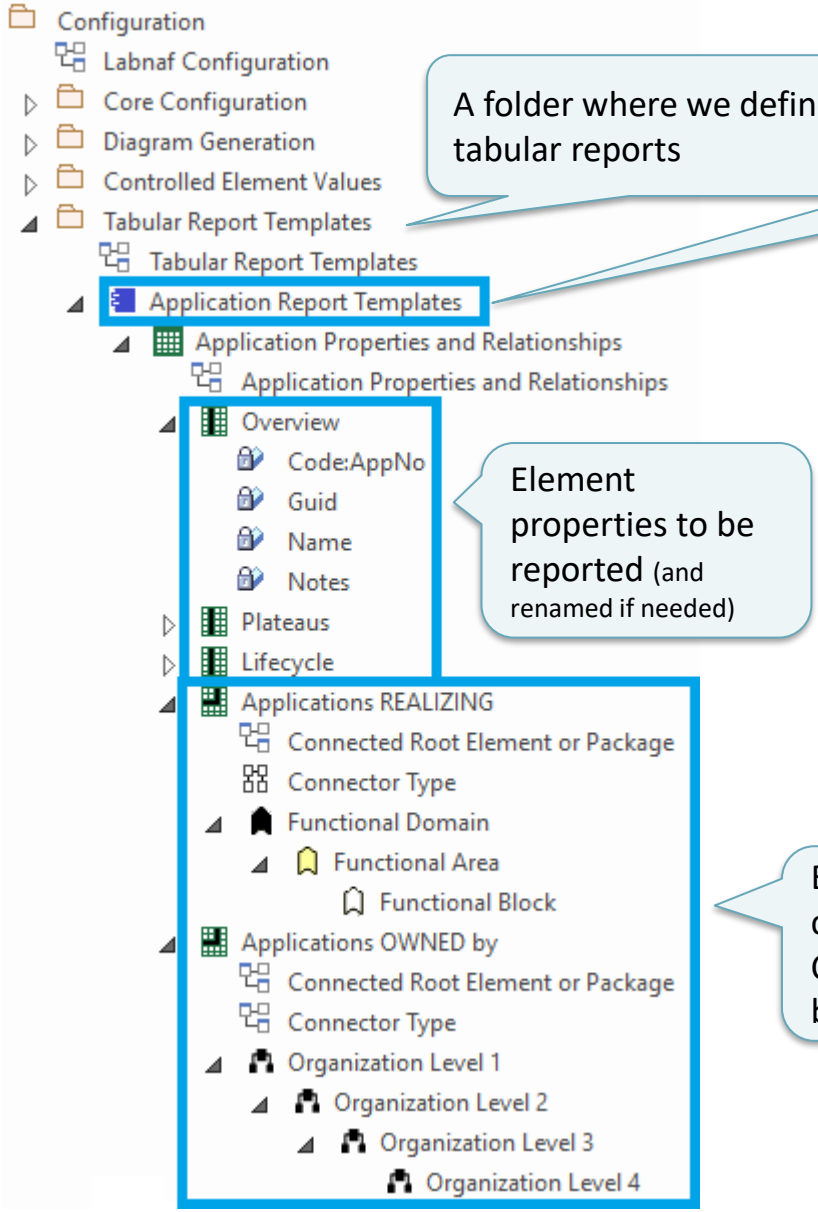




© 2014-2021 Alain De Preter - All rights reserved

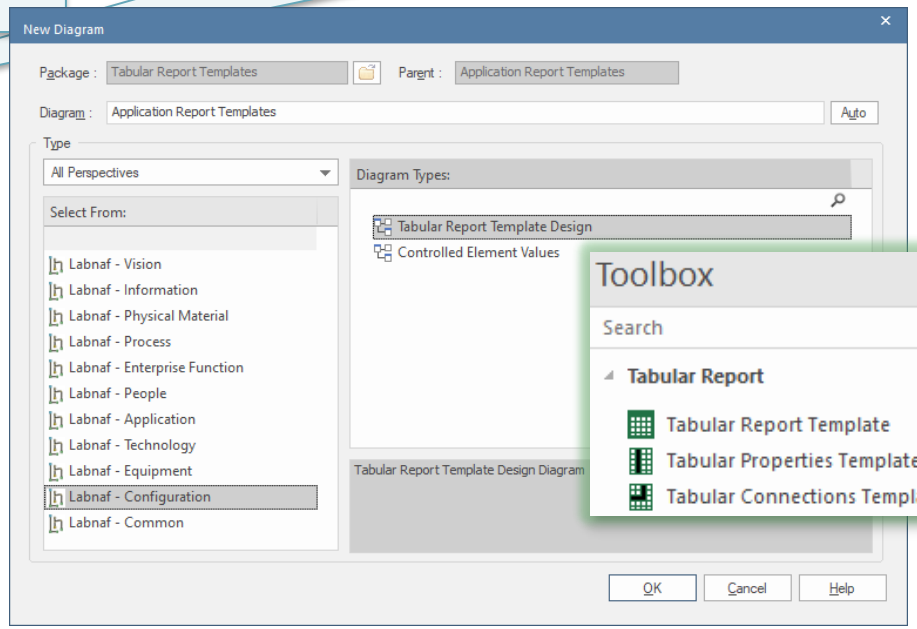
Generate Tabular Report (cont.)

Model your tabular report



A folder where we define tabular reports

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 properties to be reported (and renamed if needed)

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

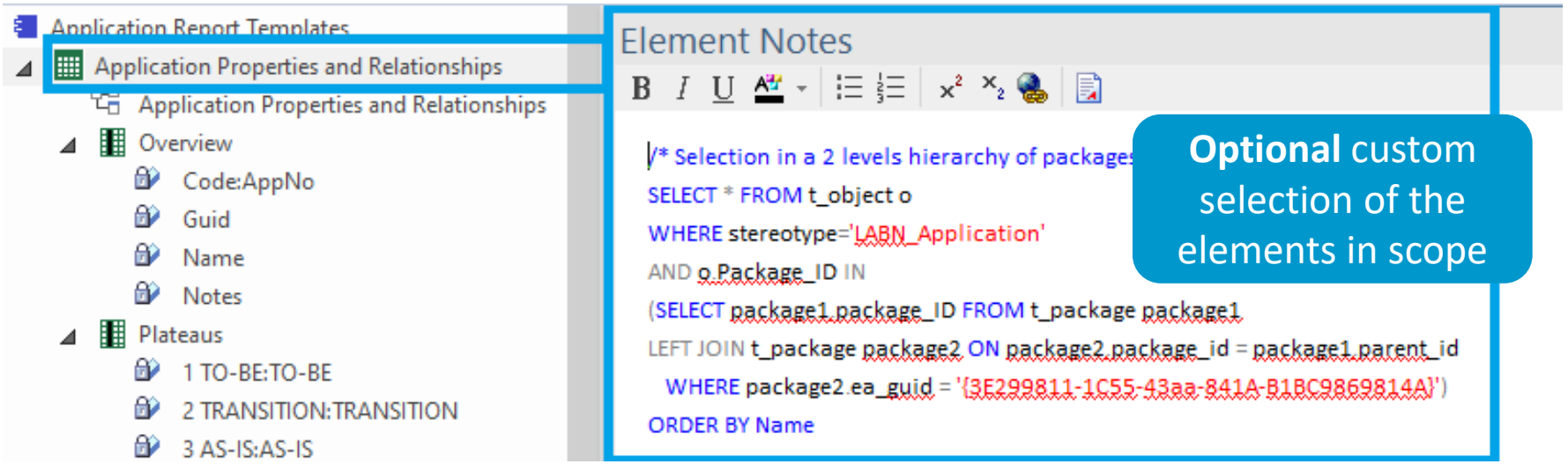
© 2014-2021 Alain De Preter - All rights reserved

Generate Tabular Report (cont.)

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

Define the set of elements to be reported

By default, **all elements** with the same stereotype “LABN_xxx” as the element prototype are selected from the related catalog with stereotype “LNCAT_xxx”.



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.

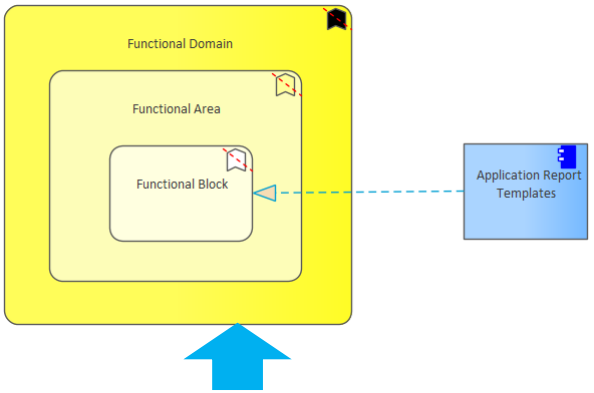
© 2014-2021 Alain De Preter - All rights reserved

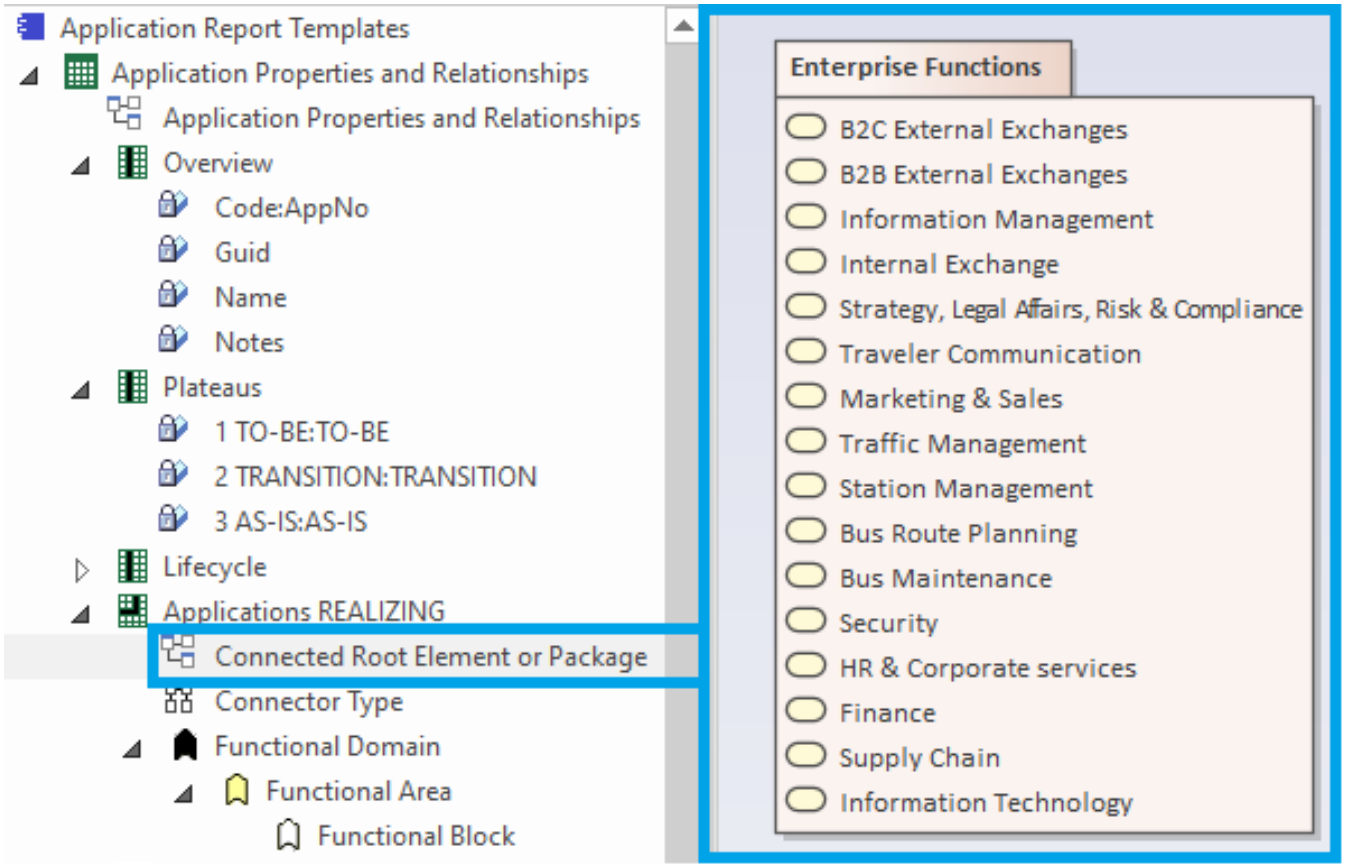
Generate Tabular Report (cont.)

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





© 2014-2021 Alain De Preter - All rights reserved

Generate Tabular Report (cont.)

To 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.

When a report template name ends with '.CSV' a CSV file is generated instead of Excel.

© 2014-2021 Alain De Preter - All rights reserved

Generate Doc
(Word, RTF, PDF)

To 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 MasterDocume
nt, the template is not required.
```


© 2014-2021 Alain De Preter - All rights reserved

Generate Html

To start HTML generation:

```
Command: GenerateHTML

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

Usage : lnps 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.

© 2014-2021 Alain De Preter - All rights reserved

Auto Connectors - Generate

To start connector generation:

```
Command: AutoConnectorsGenerate

Description: Generate connectors for child elements following defined element stereotype hierarchies.

Usage : lnps AutoConnectorsGenerate [arguments]

Arguments:

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

The generated connectors are aggregations.

Benefits of generated connectors

- Normalizes the way elements are related in a repository i.e. based on connectors
- Eases Prolaborate chart definitions: Prolaborate relies on connectors not on hierarchies

Sample Hierarchies for which aggregations are generated

- Functional Domain.Functional Area.Functional Block.Functional Category.Functional Service
- Application Platform.Application Group.Application.ApplicationComponent or Data Store
- Organization.Organization.Organization…

© 2014-2021 Alain De Preter - All rights reserved

Auto Connectors - Delete

To start deletion of generated connectors:

```
Command: AutoConnectorsDelete

Description: Delete generated connectors for child elements following defined element stereotype hierarchies.

Usage : lnps AutoConnectorsDelete [arguments]

Arguments:

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

The generated connectors are aggregations.

© 2014-2021 Alain De Preter - All rights reserved

Backup To Access File

To 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

© 2014-2021 Alain De Preter - All rights reserved

Schedule Command

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

- **InitialStartTime = 00:00:00**
- **PeriodAsMinutes = 1440**

Don't schedule 2 commands starting exactly at the same time

There are 1440 minutes in a day

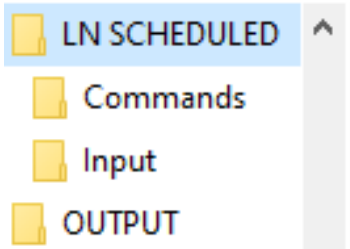
To start the schedular:

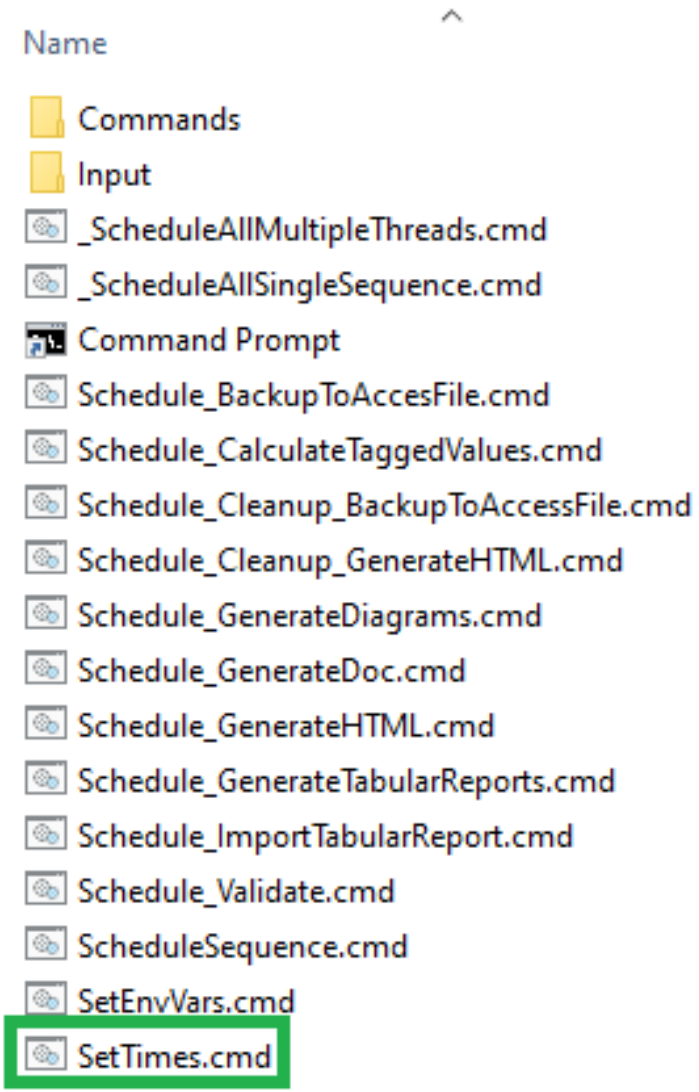
```
Command: ScheduleCommand

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

Usage : lnps 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.
```


© 2014-2021 Alain De Preter - All rights reserved

Final Recommendation

Use the preconfigured batches and settings





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_BackupToAccesFile=22:00:00
8  Set StartTime_Cleanup_GenerateHTML=22:00:05
9
10 Set StartTime_ImportTabularReport=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
```


© 2014-2021 Alain De Preter - All Rights Reserved

Labnaf PowerShell
Command Compatibility Matrix

Power Shell Commands	Sql Server	Pro Cloud Server	Access
ImportTabularReport (CSV, Excel)	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

© 2014-2021 Alain De Preter - All Rights Reserved