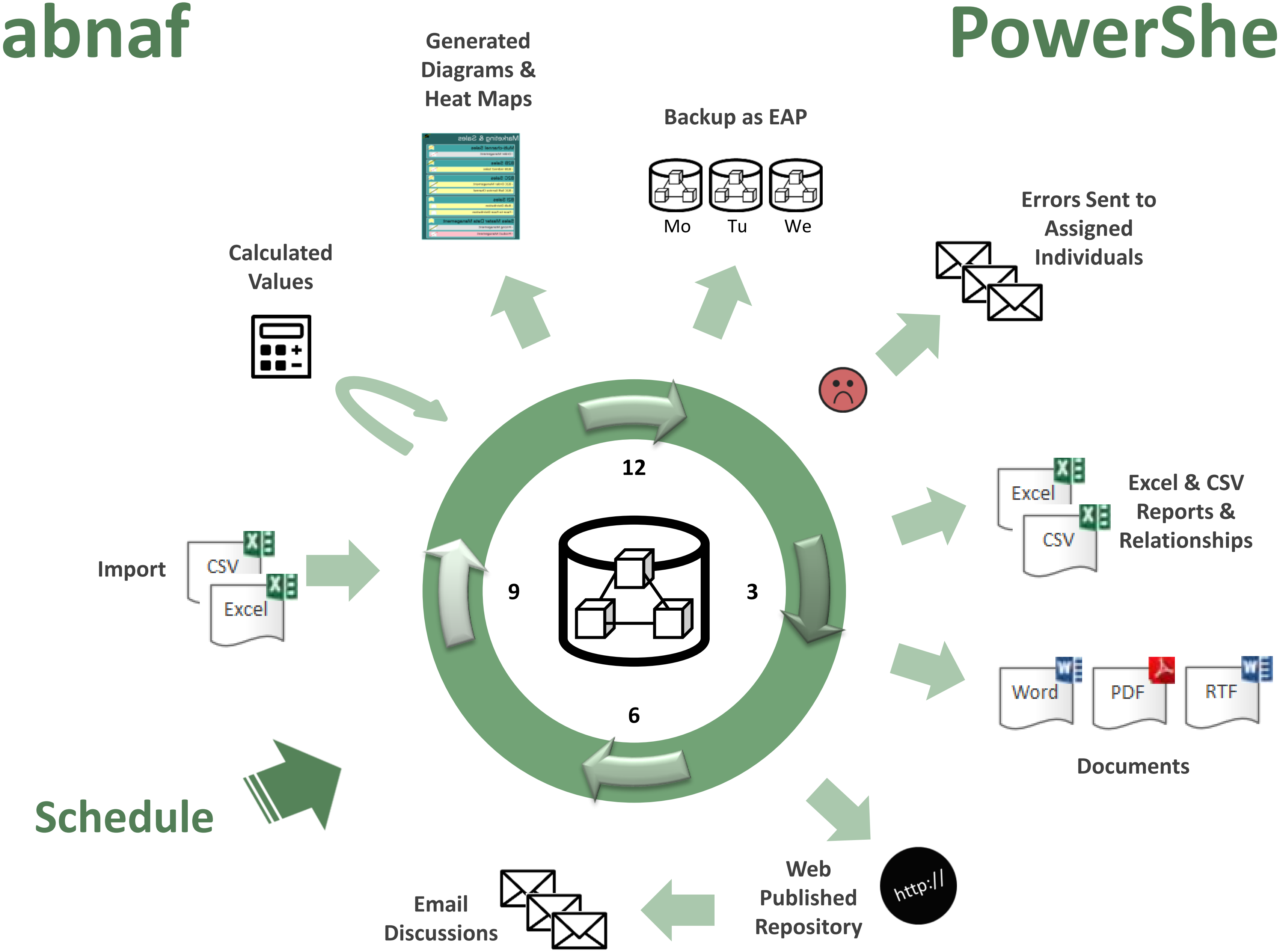
The Labnaf Strategy & Architecture Framework

Labnaf PowerShell

User Guide

Labnaf PowerShell


Generated Diagrams & Heat Maps

Backup as EAP

Errors Sent to Assigned Individuals

Calculated Values

Import

Excel & CSV Reports & Relationships

Documents

Schedule

Web Published Repository

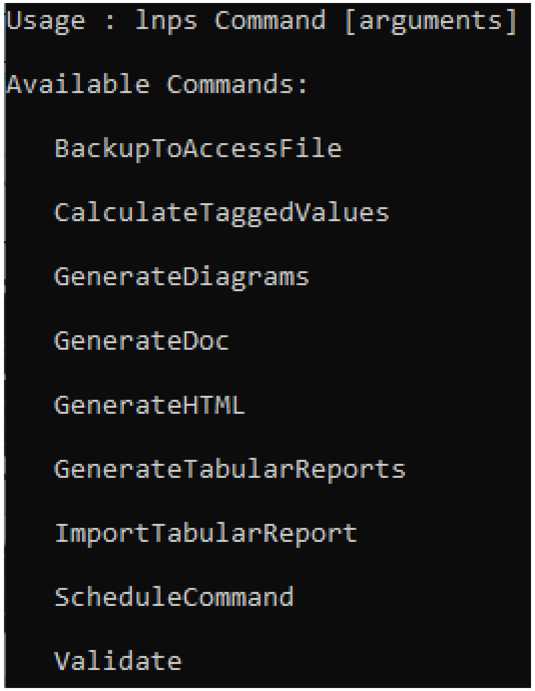
Email Discussions

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
- Backup To Access File
- Schedule Command (not only Labnaf PowerShell commands)

Running commands on the command line

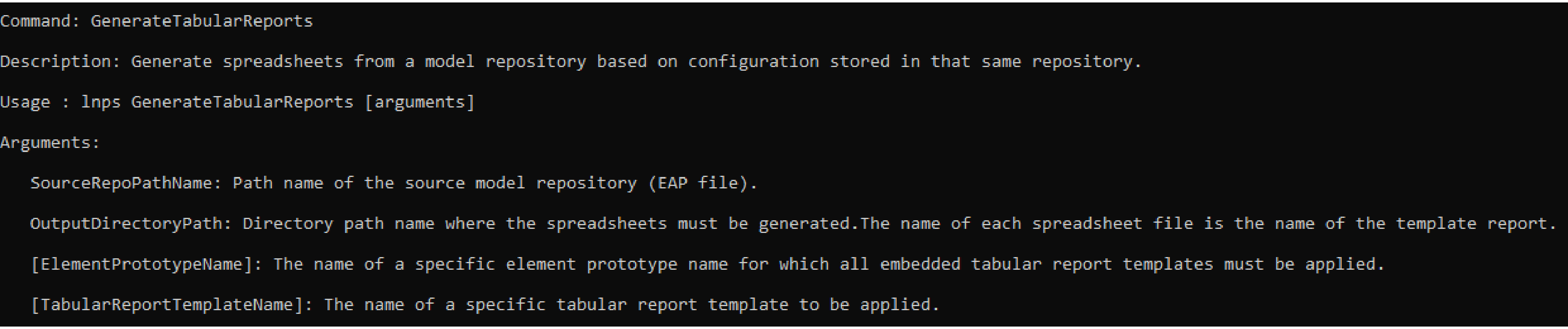
lnps



```
Usage : lnps Command [arguments]

Available Commands:

   BackupToAccessFile

   CalculateTaggedValues

   GenerateDiagrams

   GenerateDoc

   GenerateHTML

   GenerateTabularReports

   ImportTabularReport

   ScheduleCommand

   Validate
```

lnps {command name}

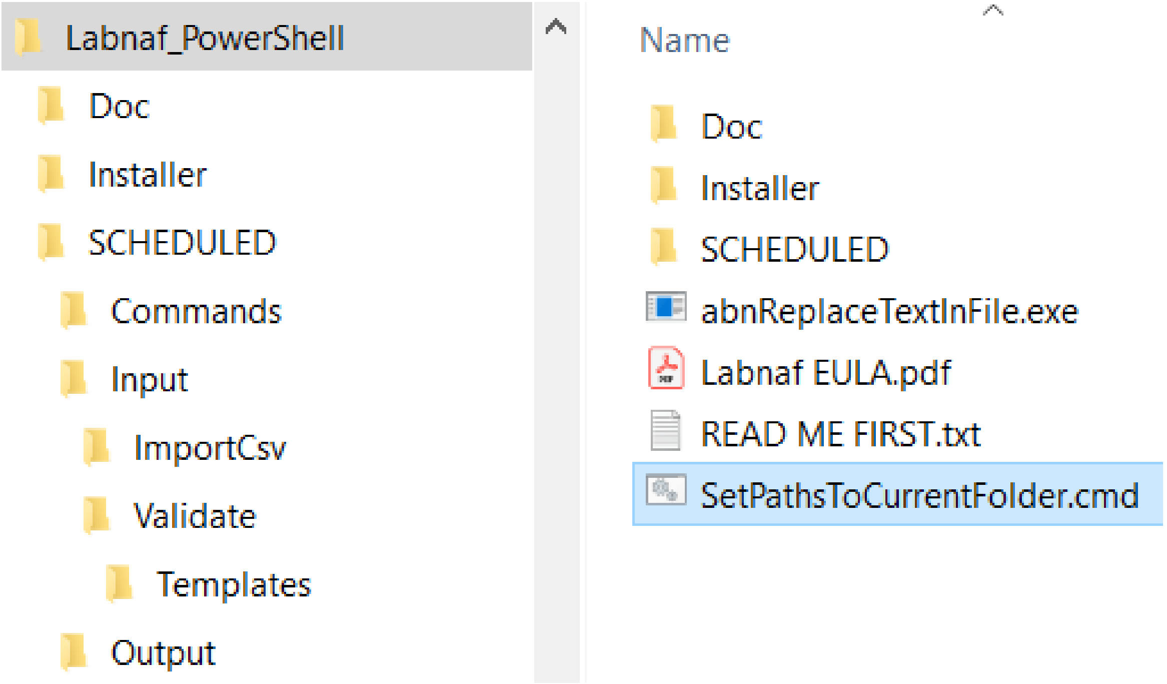
Example: if you type « lnps GenerateTabularReports » you get the following info.



```
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.
```

Automatic configuration

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



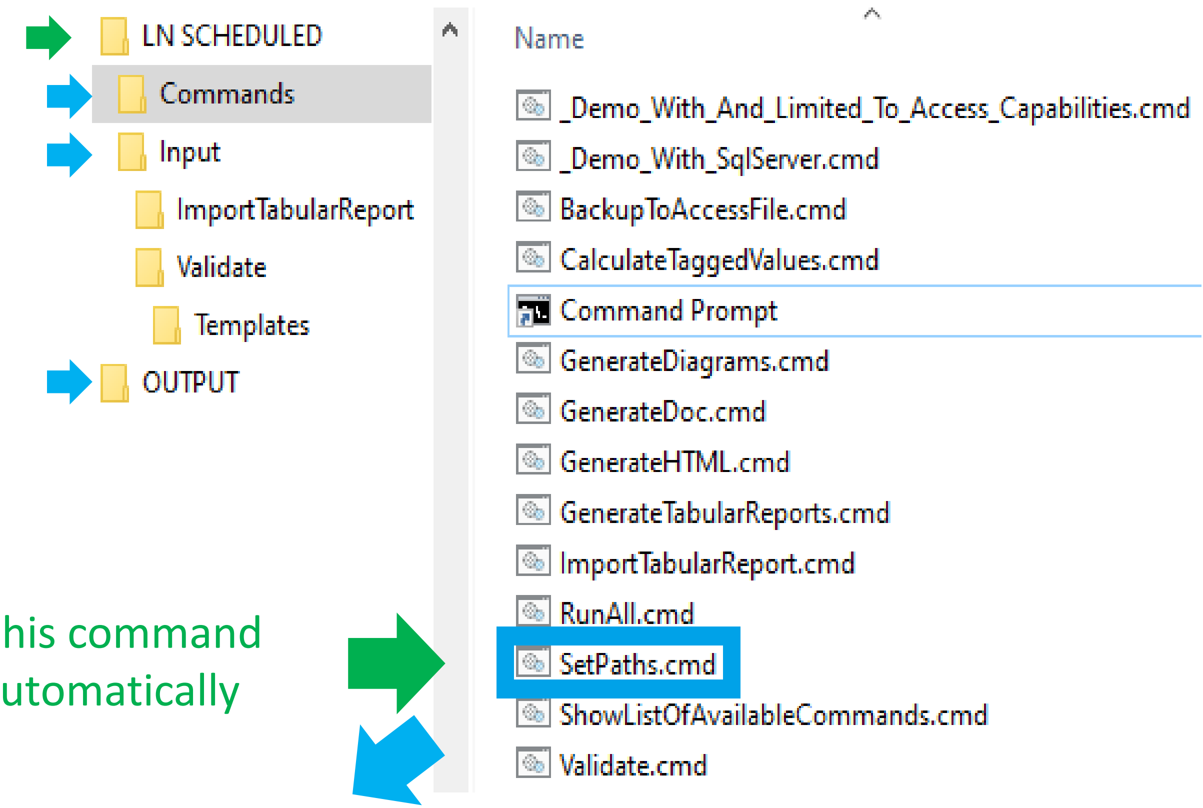
- Labnaf_PowerShell
  - Doc
  - Installer
  - SCHEDULED
    - Commands
    - Input
      - ImportCsv
      - Validate
        - Templates
    - Output

Name

- Doc
- Installer
- SCHEDULED
- abnReplaceTextInFile.exe
- Labnaf EULA.pdf
- READ ME FIRST.txt
- SetPathsToCurrentFolder.cmd

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

Preconfigured batches calling commands with predefined settings

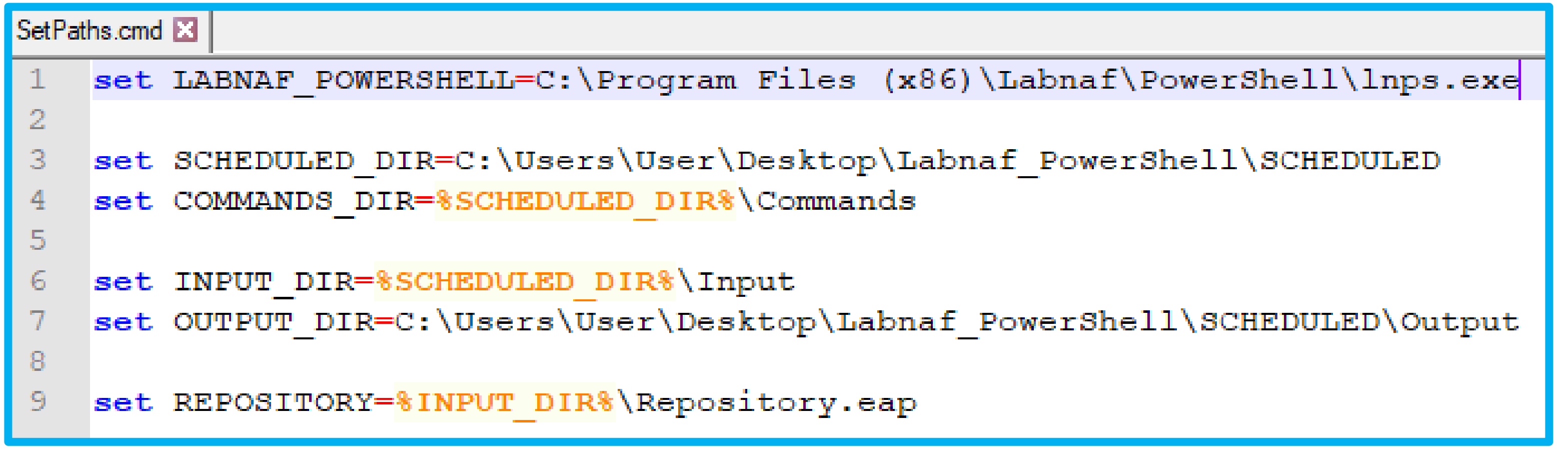


LN SCHEDULED
Commands
Input
ImportTabularReport
Validate
Templates
OUTPUT

Name

- _Demo_With_And_Limited_To_Access_Capabilities.cmd
- _Demo_With_SqlServer.cmd
- BackupToAccessFile.cmd
- CalculateTaggedValues.cmd
- Command Prompt
- GenerateDiagrams.cmd
- GenerateDoc.cmd
- GenerateHTML.cmd
- GenerateTabularReports.cmd
- ImportTabularReport.cmd
- RunAll.cmd
- SetPaths.cmd
- ShowListOfAvailableCommands.cmd
- Validate.cmd

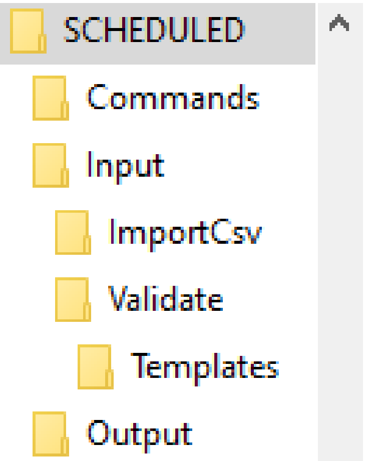
The paths in this command were reset automatically



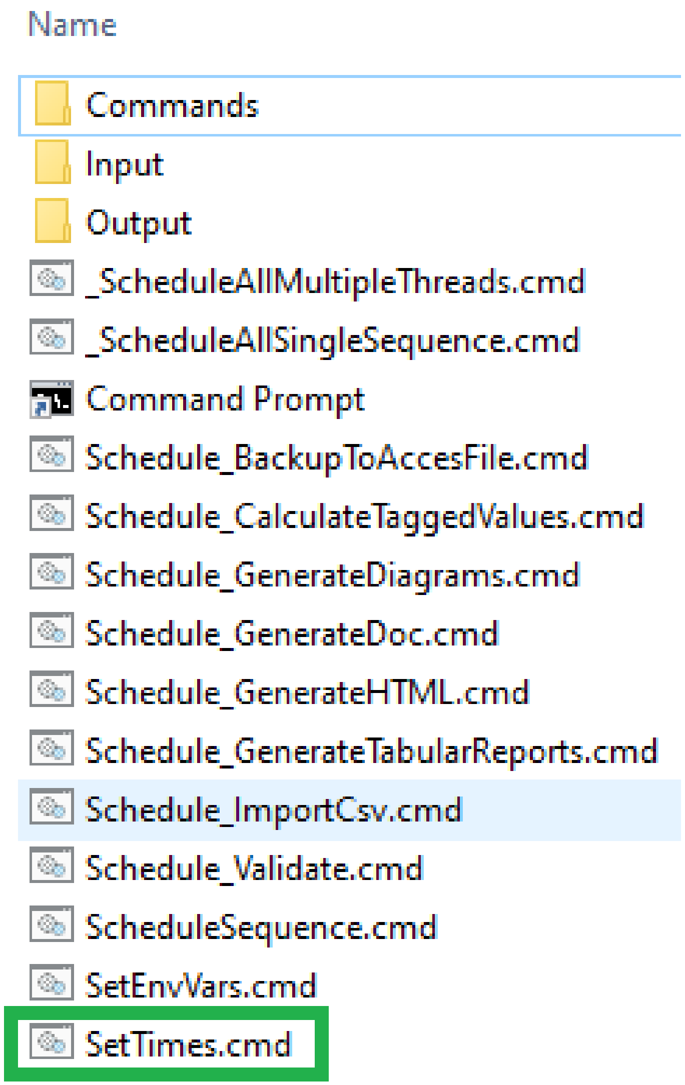
SetPaths.cmd

```
set LABNAF_POWERSHELL=C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe

set SCHEDULED_DIR=C:\Users\User\Desktop\Labnaf_PowerShell\SCHEDULED
set COMMANDS_DIR=%SCHEDULED_DIR%\Commands

set INPUT_DIR=%SCHEDULED_DIR%\Input
set OUTPUT_DIR=C:\Users\User\Desktop\Labnaf_PowerShell\SCHEDULED\Output

set REPOSITORY=%INPUT_DIR%\Repository.eap
```

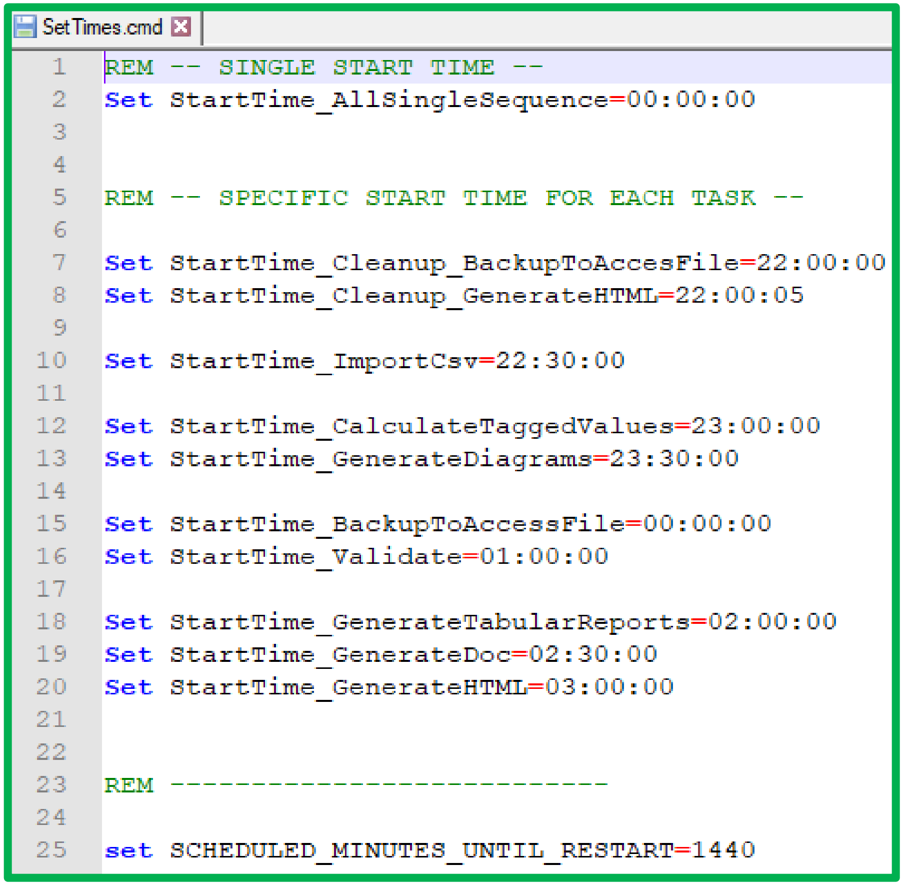
Preconfigured Command Scheduling



SCHEDULED
Commands
Input
ImportCsv
Validate
Templates
Output



Name
Commands
Input
Output
_ScheduleAllMultipleThreads.cmd
_ScheduleAllSingleSequence.cmd
Command Prompt
Schedule_BackupToAccesFile.cmd
Schedule_CalculateTaggedValues.cmd
Schedule_GenerateDiagrams.cmd
Schedule_GenerateDoc.cmd
Schedule_GenerateHTML.cmd
Schedule_GenerateTabularReports.cmd
Schedule_ImportCsv.cmd
Schedule_Validate.cmd
ScheduleSequence.cmd
SetEnvVars.cmd
SetTimes.cmd

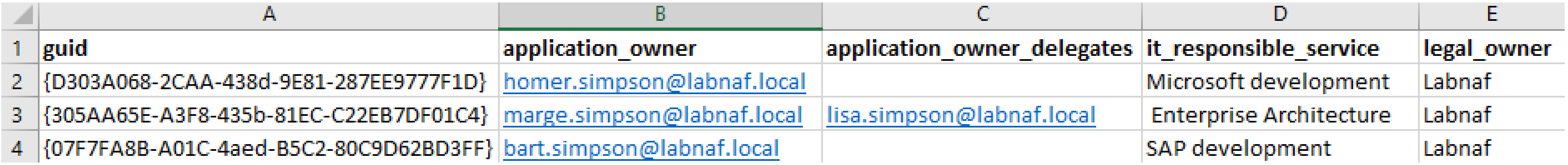


SetTimes.cmd

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


REM -- SPECIFIC START TIME FOR EACH TASK --

Set StartTime_Cleanup_BackupToAccesFile=22:00:00
Set StartTime_Cleanup_GenerateHTML=22:00:05

Set StartTime_ImportCsv=22:30:00

Set StartTime_CalculateTaggedValues=23:00:00
Set StartTime_GenerateDiagrams=23:30:00

Set StartTime_BackupToAccessFile=00:00:00
Set StartTime_Validate=01:00:00

Set StartTime_GenerateTabularReports=02:00:00
Set StartTime_GenerateDoc=02:30:00
Set StartTime_GenerateHTML=03:00:00


REM ---------------------------

set SCHEDULED_MINUTES_UNTIL_RESTART=1440
```

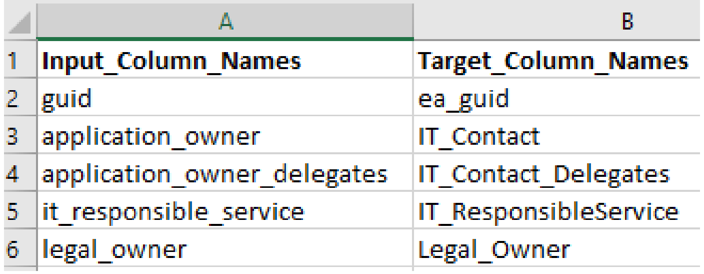
ImportTabularReport

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



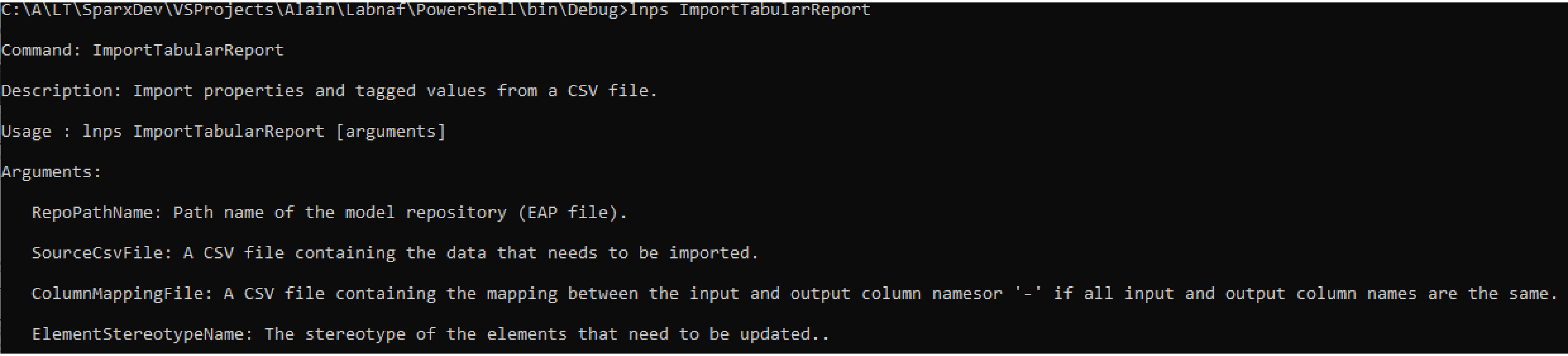
	A	B	C	D	E
1	**guid**	**application_owner**	**application_owner_delegates**	**it_responsible_service**	**legal_owner**
2	{D303A068-2CAA-438d-9E81-287EE9777F1D}	homer.simpson@labnaf.local		Microsoft development	Labnaf
3	{305AA65E-A3F8-435b-81EC-C22EB7DF01C4}	marge.simpson@labnaf.local	lisa.simpson@labnaf.local	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

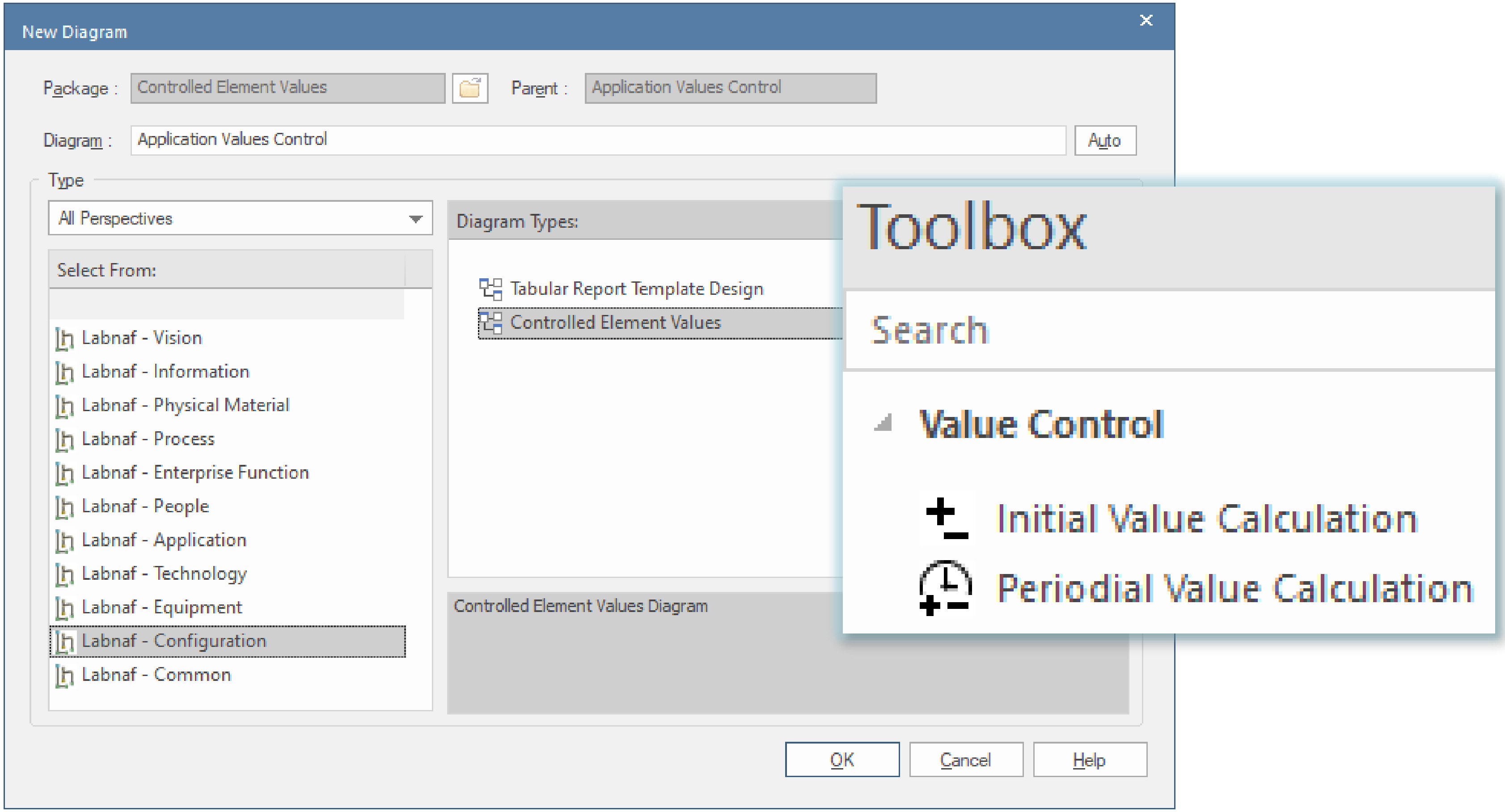
To start the import



```
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..
```

Calculate Tagged Values

Model your value calculations



New Diagram

Package : Controlled Element Values

Parent : Application Values Control

Diagram : Application Values Control

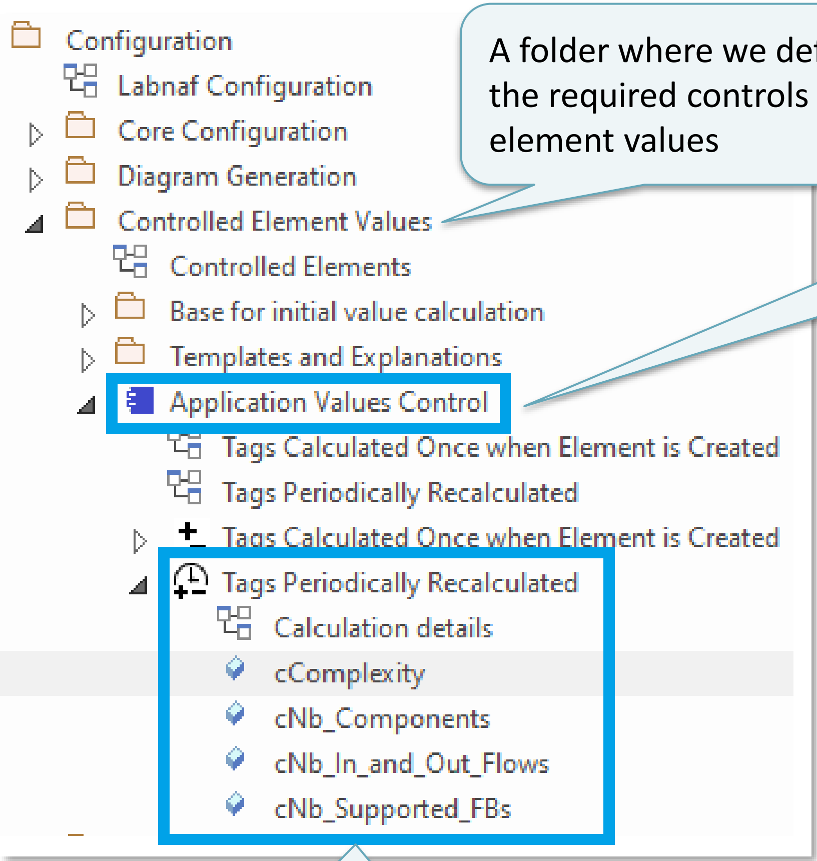
Type: All Perspectives

Diagram Types: Controlled Element Values

Controlled Element Values Diagram

Calculate Tagged Values (cont.)

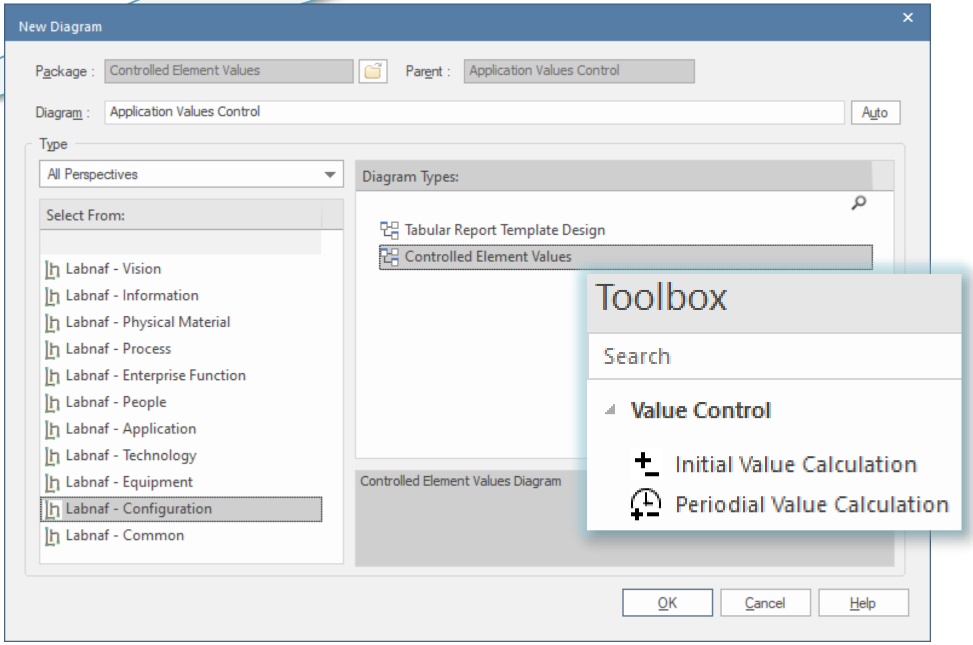
Model your value calculations



- Configuration
  - Labnaf Configuration
  - Core Configuration
  - Diagram Generation
  - Controlled Element Values
    - Controlled Elements
    - Base for initial value calculation
    - Templates and Explanations
    - Application Values Control
      - Tags Calculated Once when Element is Created
      - Tags Periodically Recalculated
      - Tags Calculated Once when Element is Created
      - Tags Periodically Recalculated
        - Calculation details
        - cComplexity
        - cNb_Components
        - cNb_In_and_Out_Flows
        - cNb_Supported_FBs

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.



New Diagram

Package : Controlled Element Values

Parent : Application Values Control

Diagram : Application Values Control

Type: All Perspectives

Select From:

- Labnaf - Vision
- Labnaf - Information
- Labnaf - Physical Material
- Labnaf - Process
- Labnaf - Enterprise Function
- Labnaf - People
- Labnaf - Application
- Labnaf - Technology
- Labnaf - Equipment
- Labnaf - Configuration
- Labnaf - Common

Diagram Types:

- Tabular Report Template Design
- Controlled Element Values

Controlled Element Values Diagram

Toolbox

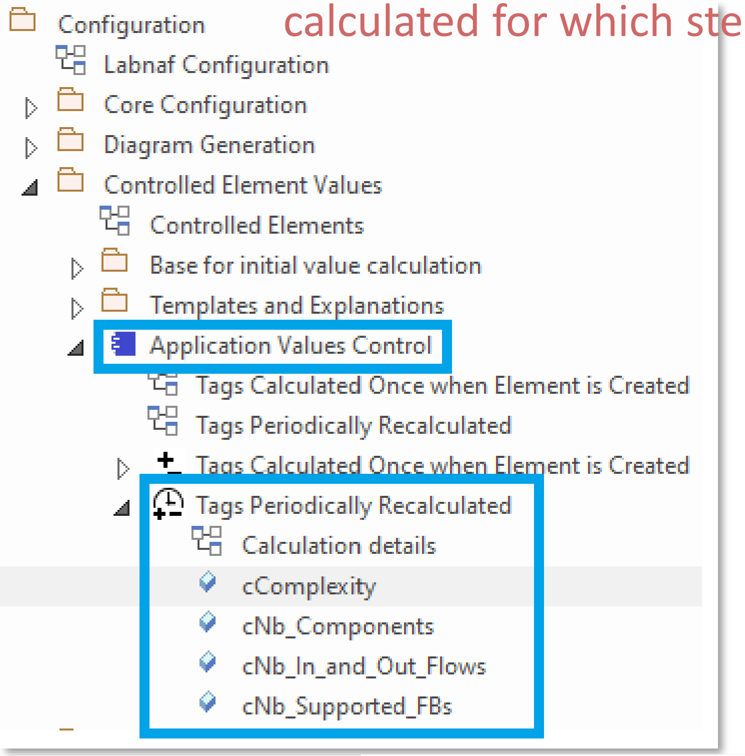
Value Control

- Initial Value Calculation
- Periodial Value Calculation

The tagged values that must be automatically calculated

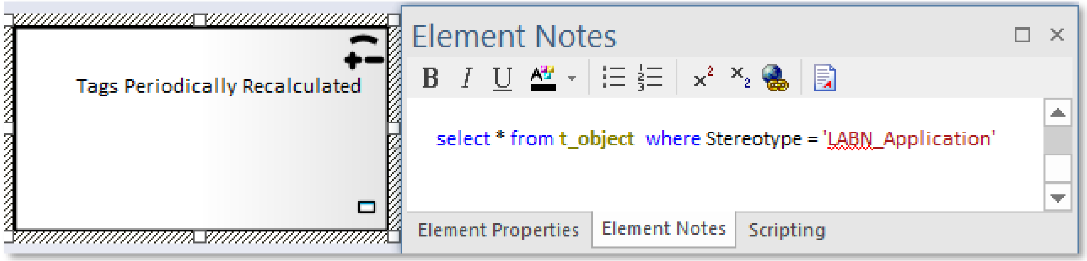
Calculate Tagged Values (cont.)

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



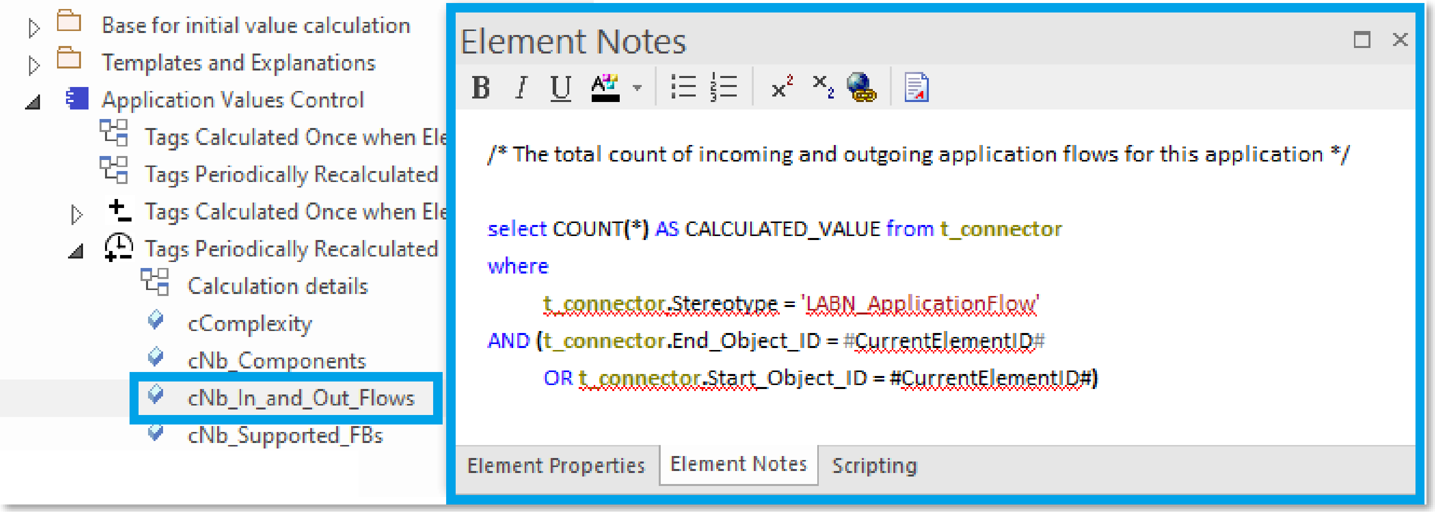
- Configuration
  - Labnaf Configuration
  - Core Configuration
  - Diagram Generation
  - Controlled Element Values
    - Controlled Elements
    - Base for initial value calculation
    - Templates and Explanations
    - Application Values Control
      - Tags Calculated Once when Element is Created
      - Tags Periodically Recalculated
      - Tags Calculated Once when Element is Created
      - Tags Periodically Recalculated
        - Calculation details
        - cComplexity
        - cNb_Components
        - cNb_In_and_Out_Flows
        - cNb_Supported_FBs

2. Scope: Which elements need to be updated?



```
select * from t_object  where Stereotype = 'LABN_Application'
```

3. Calculation: How shall we calculate the value?

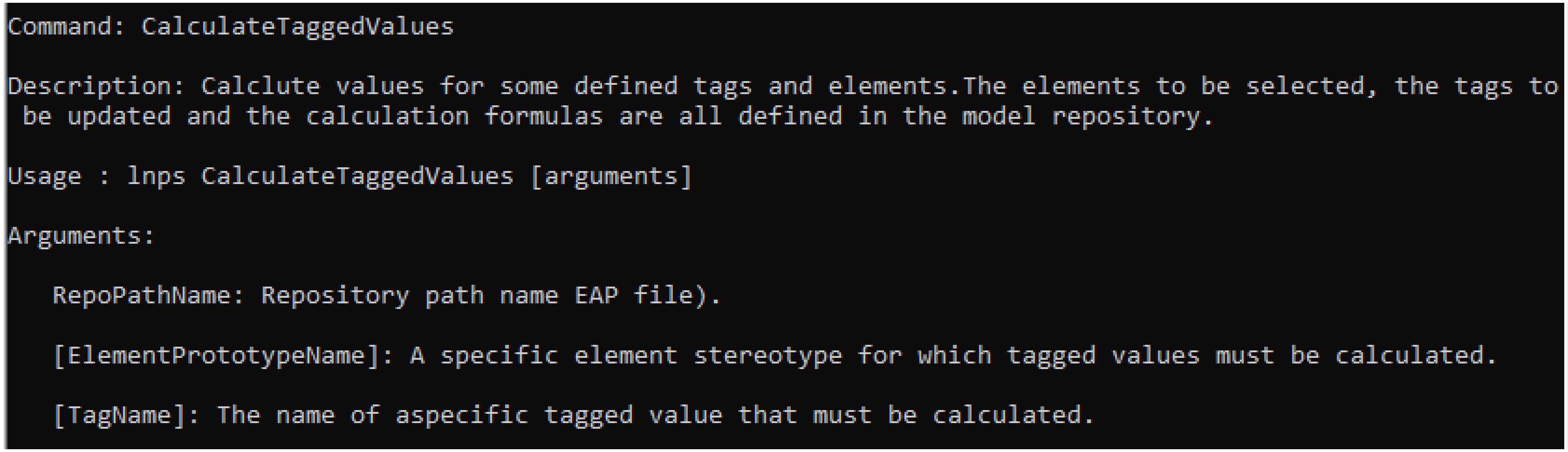


- Base for initial value calculation
- Templates and Explanations
- Application Values Control
  - Tags Calculated Once when Ele
  - Tags Periodically Recalculated
  - Tags Calculated Once when Ele
  - Tags Periodically Recalculated
    - Calculation details
    - cComplexity
    - cNb_Components
    - cNb_In_and_Out_Flows
    - cNb_Supported_FBs

```
/* The total count of incoming and outgoing application flows for this application */

select COUNT(*) AS CALCULATED_VALUE from t_connector
where
        t_connector.Stereotype = 'LABN_ApplicationFlow'
AND (t_connector.End_Object_ID = #CurrentElementID#
        OR t_connector.Start_Object_ID = #CurrentElementID#)
```

Calculate Tagged Values (cont.)

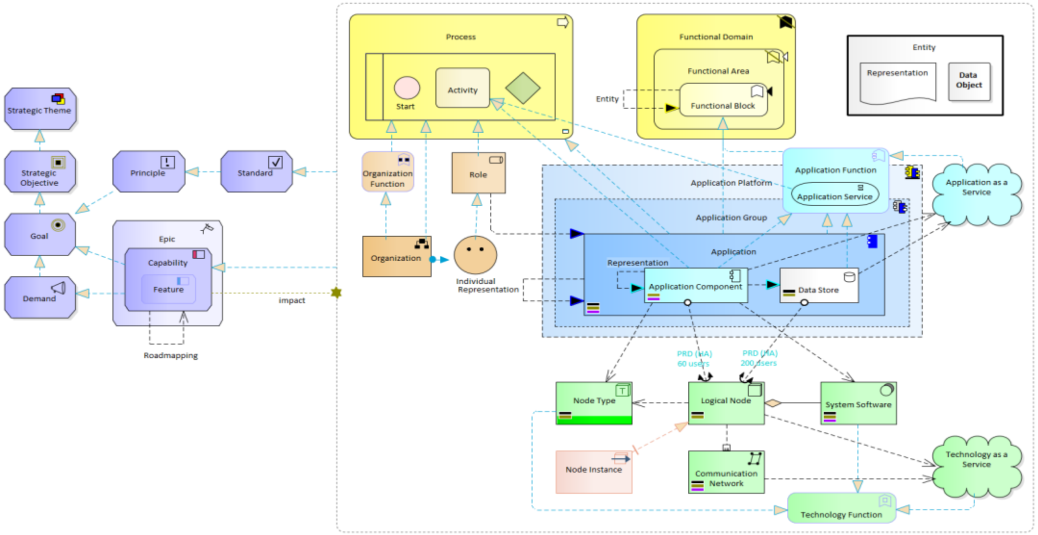
4. To start calculation:

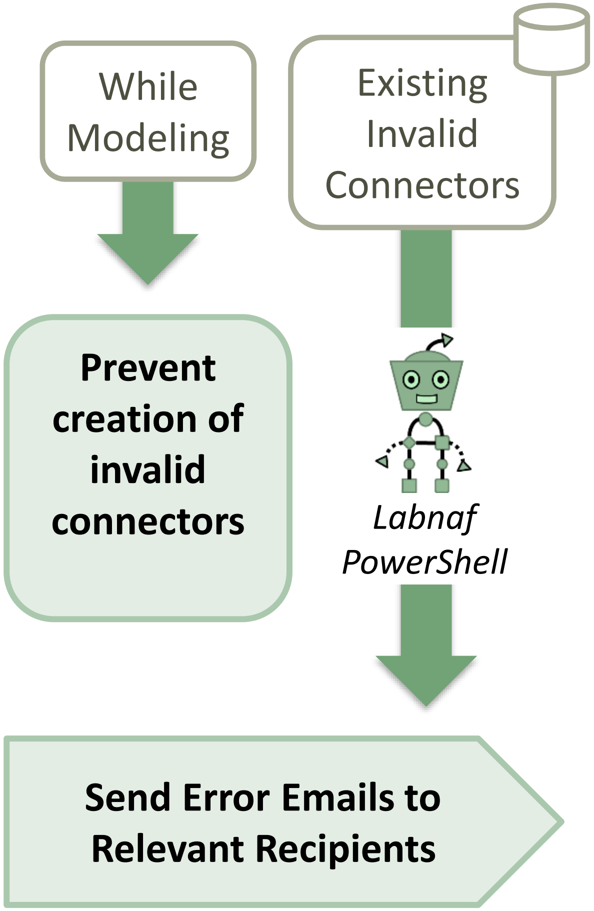


```
Command: CalculateTaggedValues

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.
```

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

Validate





While Modeling

Existing Invalid Connectors

Prevent creation of invalid connectors

Labnaf PowerShell

Send Error Emails to Relevant Recipients

Why do we need periodical validation?

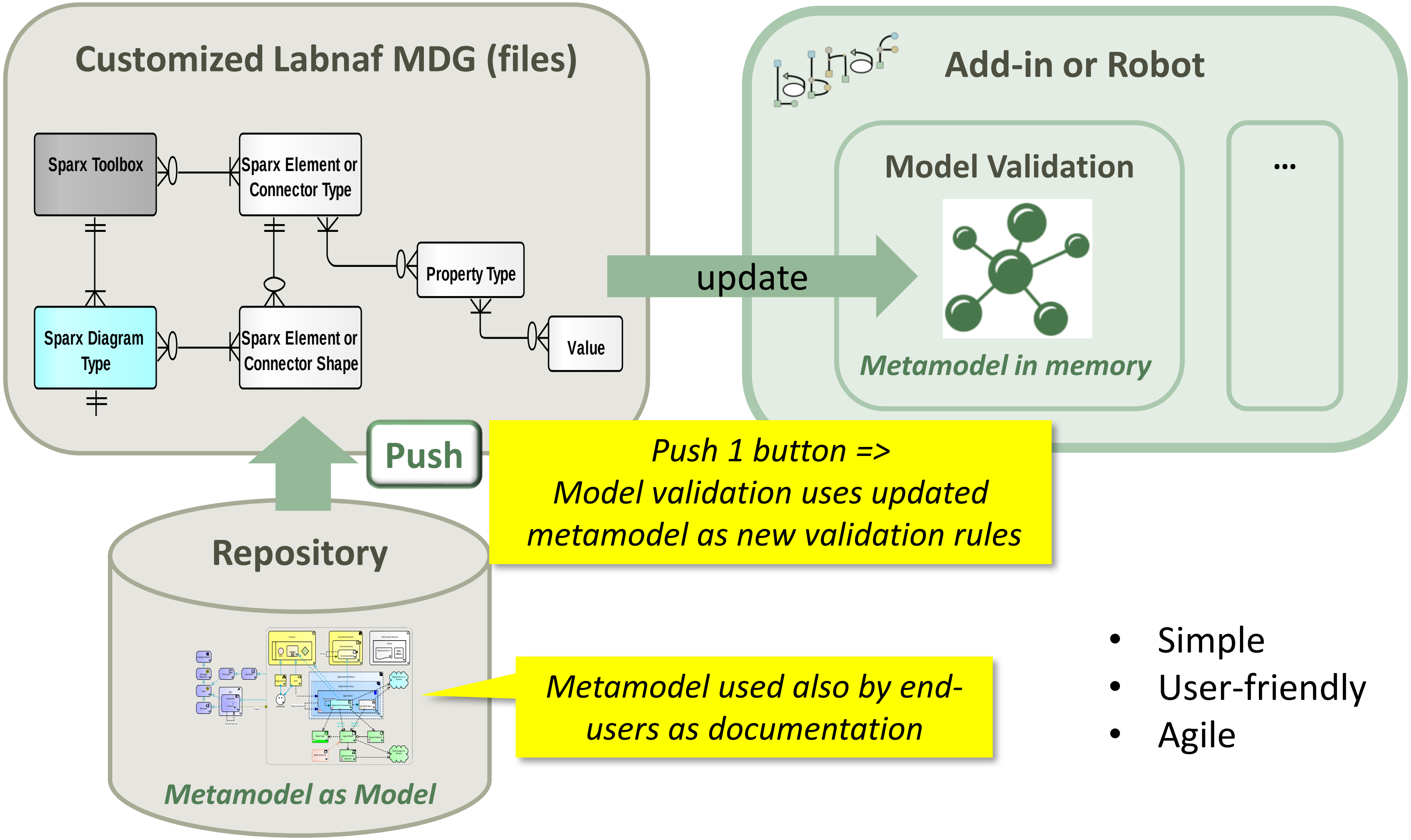
How could we have errors if we prevent users from entering errors?

- 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.

Validate (cont.)

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

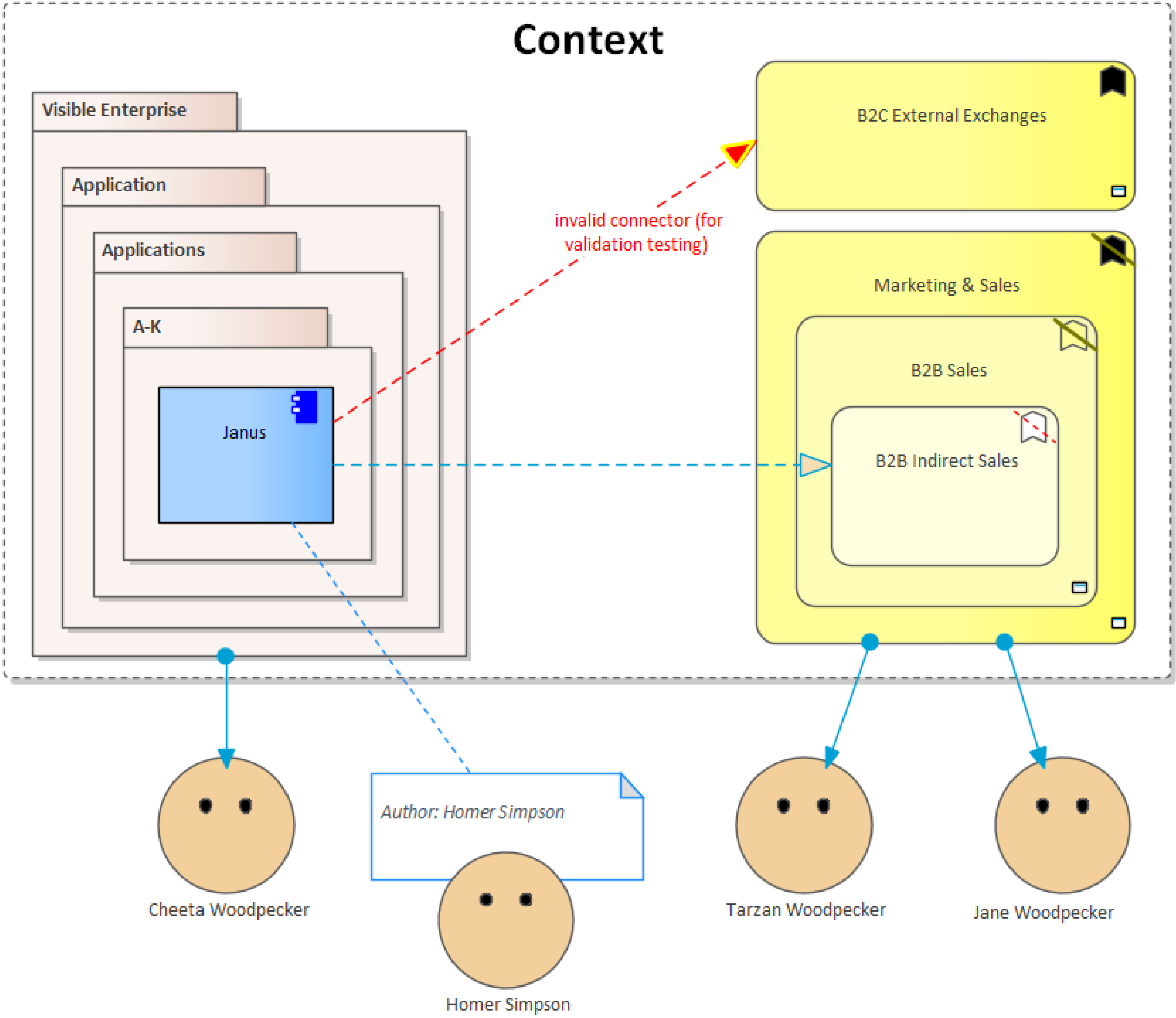


Push 1 button => Model validation uses updated metamodel as new validation rules

Metamodel used also by end-users as documentation

- Simple
- User-friendly
- Agile

Error message routing is based on architecture management assignments

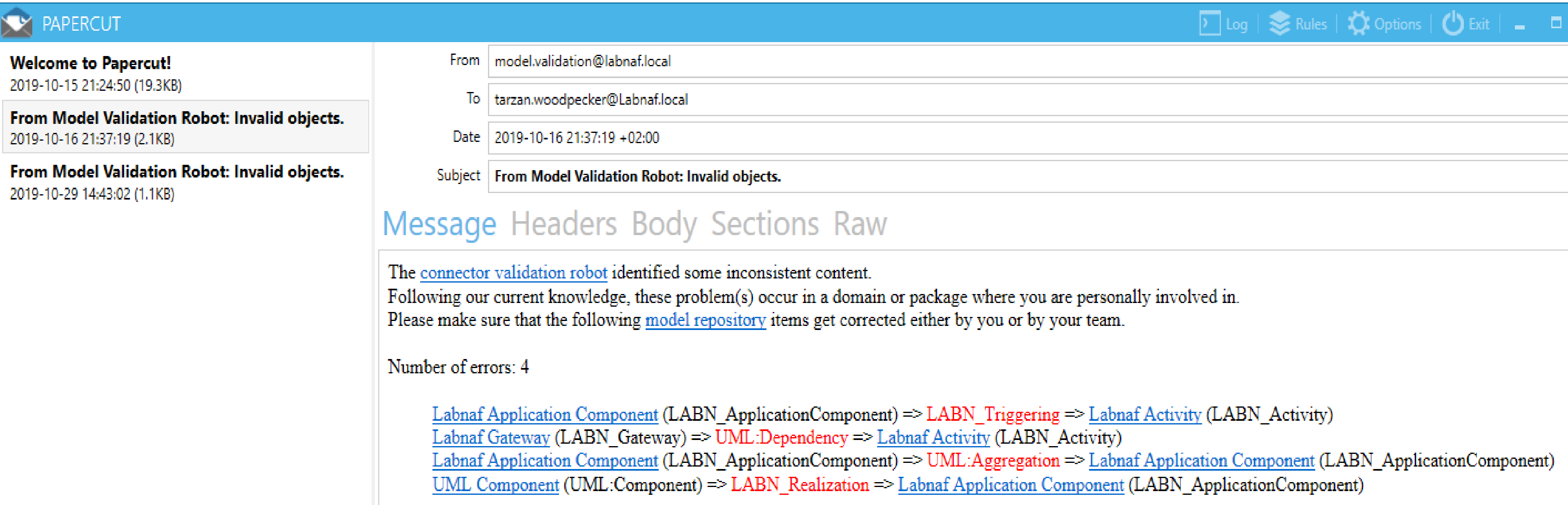


Error message distribution depends on

- 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



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

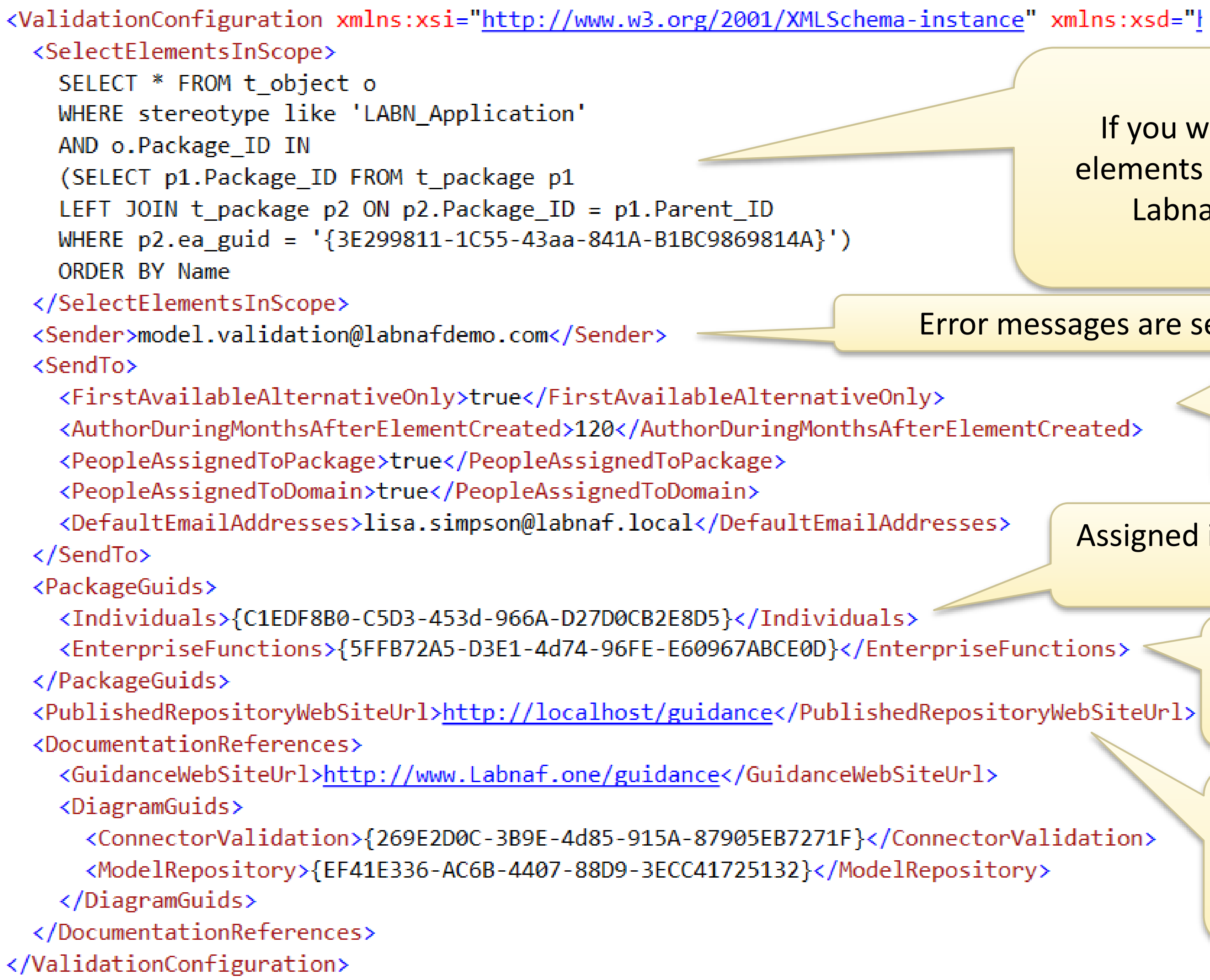
Subject: **From Model Validation Robot: Invalid objects.**

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="
  <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.

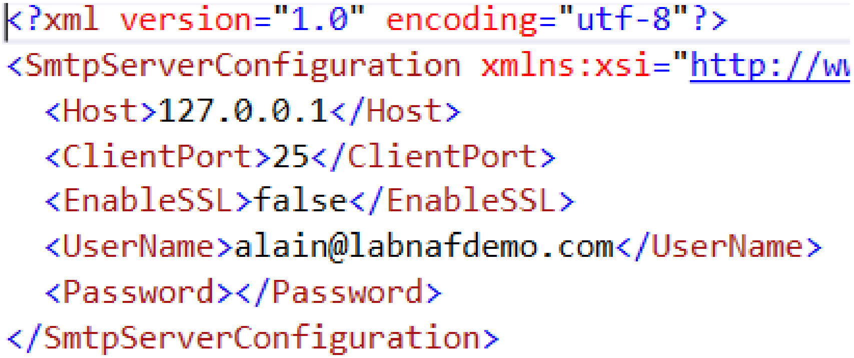
Assigned individuals with their emails are listed in a package

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

Validate (cont.)

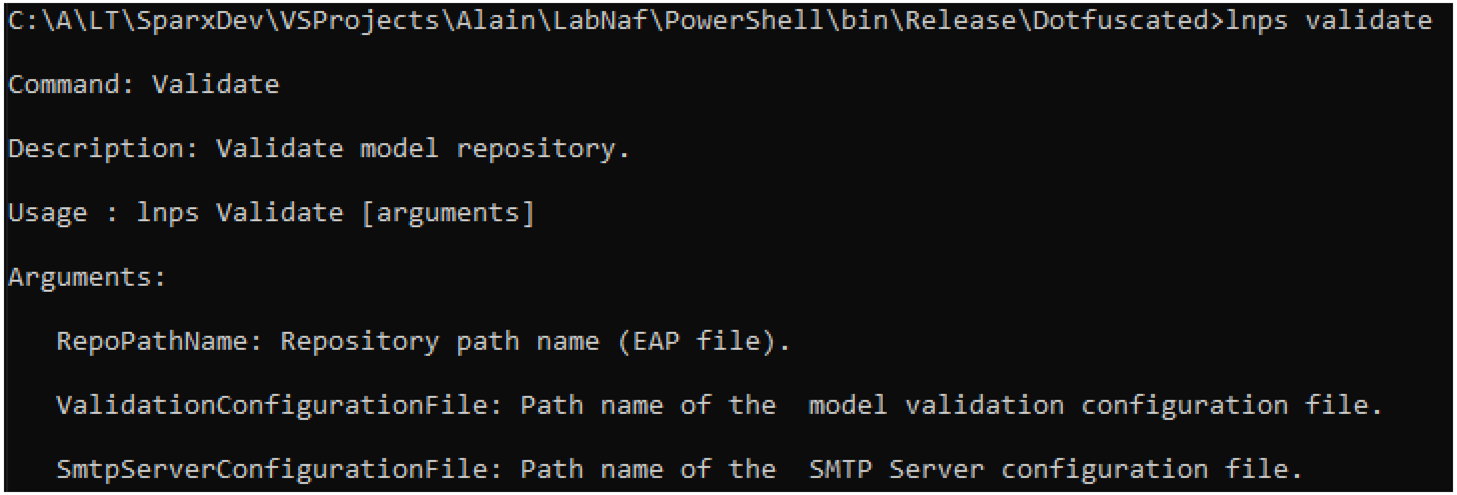
Smtp Server configuration is straightforward



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

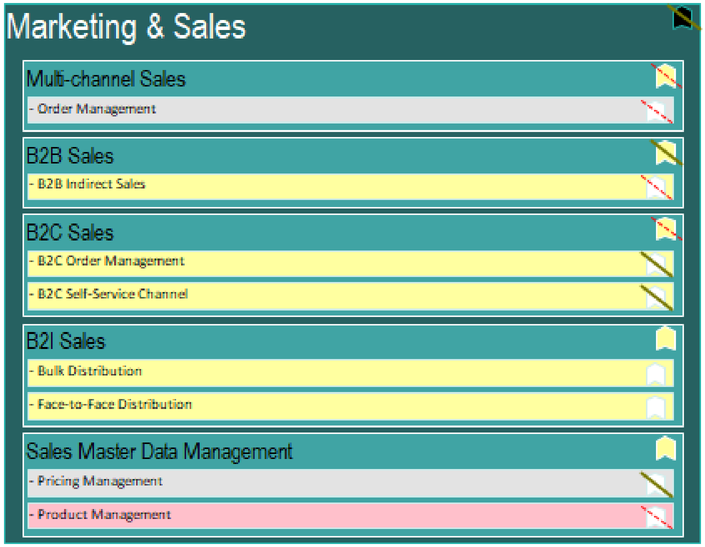


```
C:\A\LT\SparxDev\VSProjects\Alain\LabNaf\PowerShell\bin\Release\Dotfuscated>lnps validate

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.
```

Generate Diagrams

Sample Results

Enterprise Function Taxonomy & Applications Supporting Level 1



Marketing & Sales

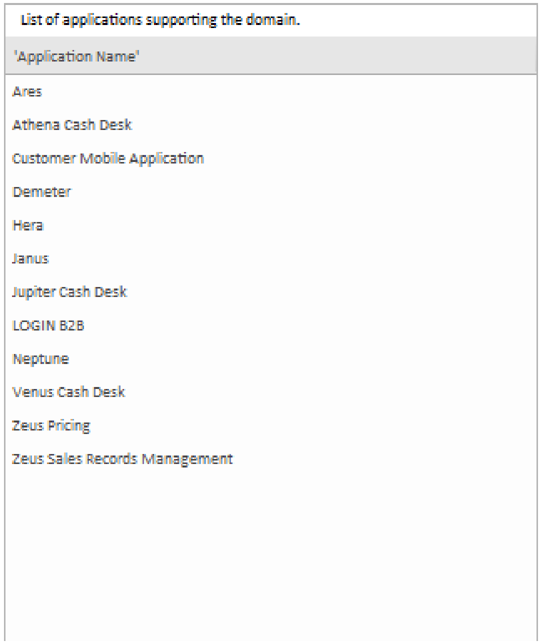
Multi-channel Sales
- Order Management

B2B Sales
- B2B Indirect Sales

B2C Sales
- B2C Order Management
- B2C Self-Service Channel

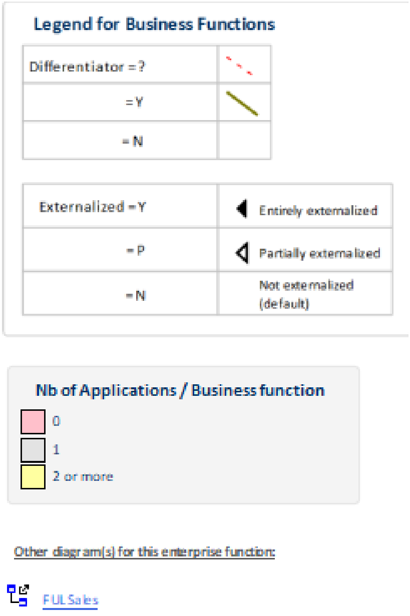
B2I Sales
- Bulk Distribution
- Face-to-Face Distribution

Sales Master Data Management
- Pricing Management
- Product Management



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)

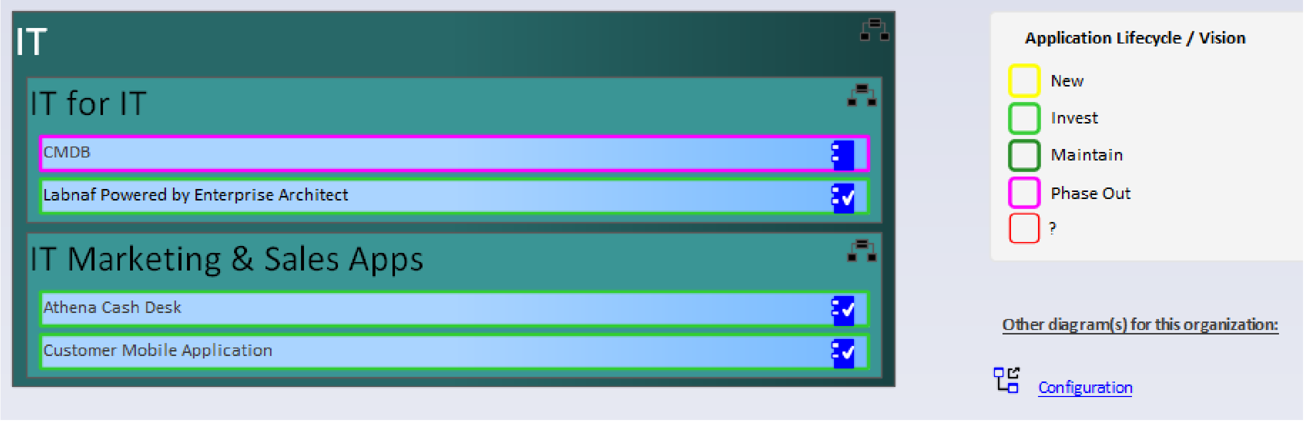
Nb of Applications / Business function

🟥	0
⬜	1
🟨	2 or more

Other diagram(s) for this enterprise function:

FUL Sales

Applications managed by organizations



IT

IT for IT
- CMDB
- Labnaf Powered by Enterprise Architect

IT Marketing & Sales Apps
- Athena Cash Desk
- Customer Mobile Application

Application Lifecycle / Vision

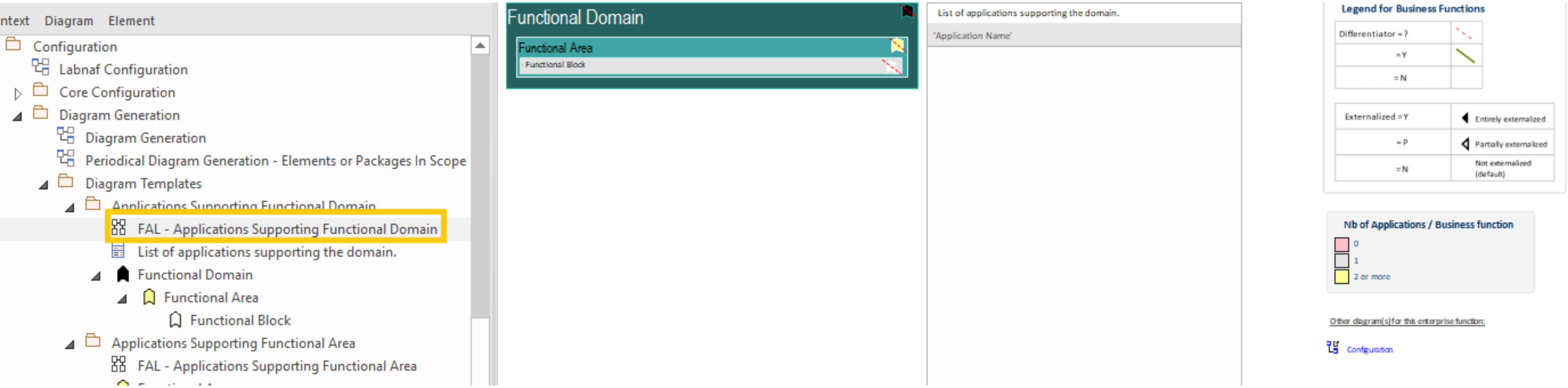
- □ New
- □ Invest
- □ Maintain
- □ Phase Out
- □ ?

Other diagram(s) for this organization:

Configuration

Sample Diagram Templates

Enterprise Function Taxonomy & Applications Supporting Level 1



ntext Diagram Element

- 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

List of applications supporting the domain.

'Application Name'

Legend for Business Functions

Differentiator = ?	
= Y	
= N	

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

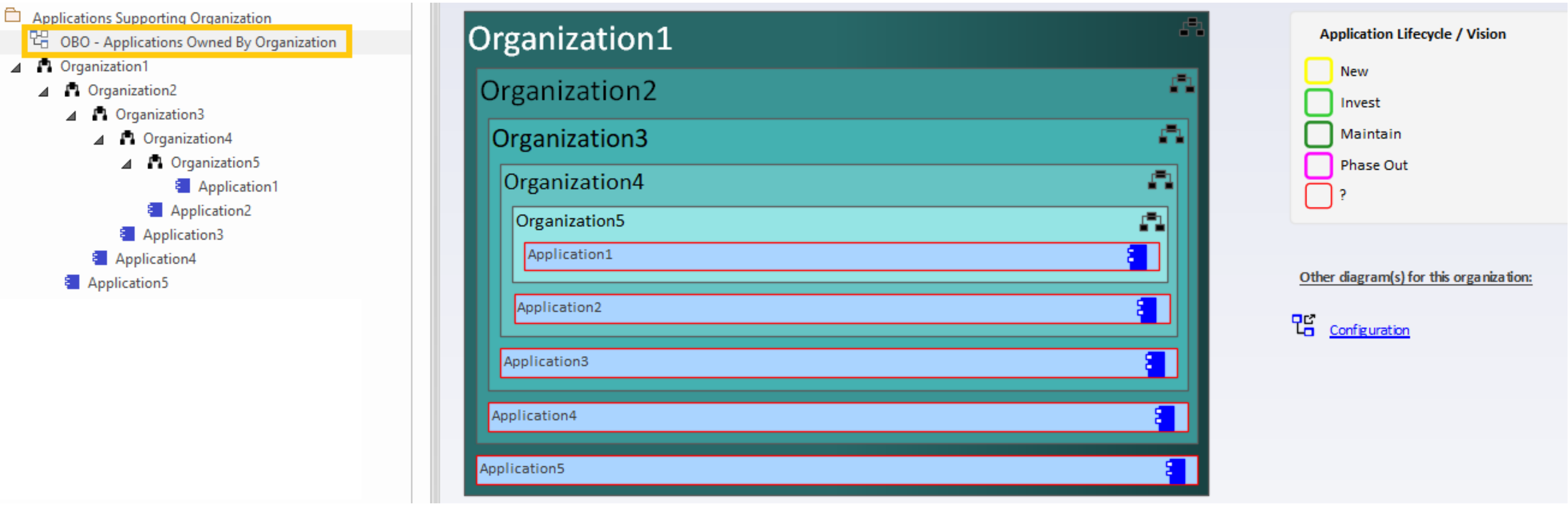
Nb of Applications / Business function

- 🟥 0
- ⬜ 1
- 🟨 2 or more

Other diagram(s) for this enterprise function:

Configuration

Applications managed by organizations



- Applications Supporting Organization
  - OBO - Applications Owned By Organization
  - Organization1
    - Organization2
      - Organization3
        - Organization4
          - Organization5
            - Application1
          - Application2
        - Application3
      - Application4
    - Application5

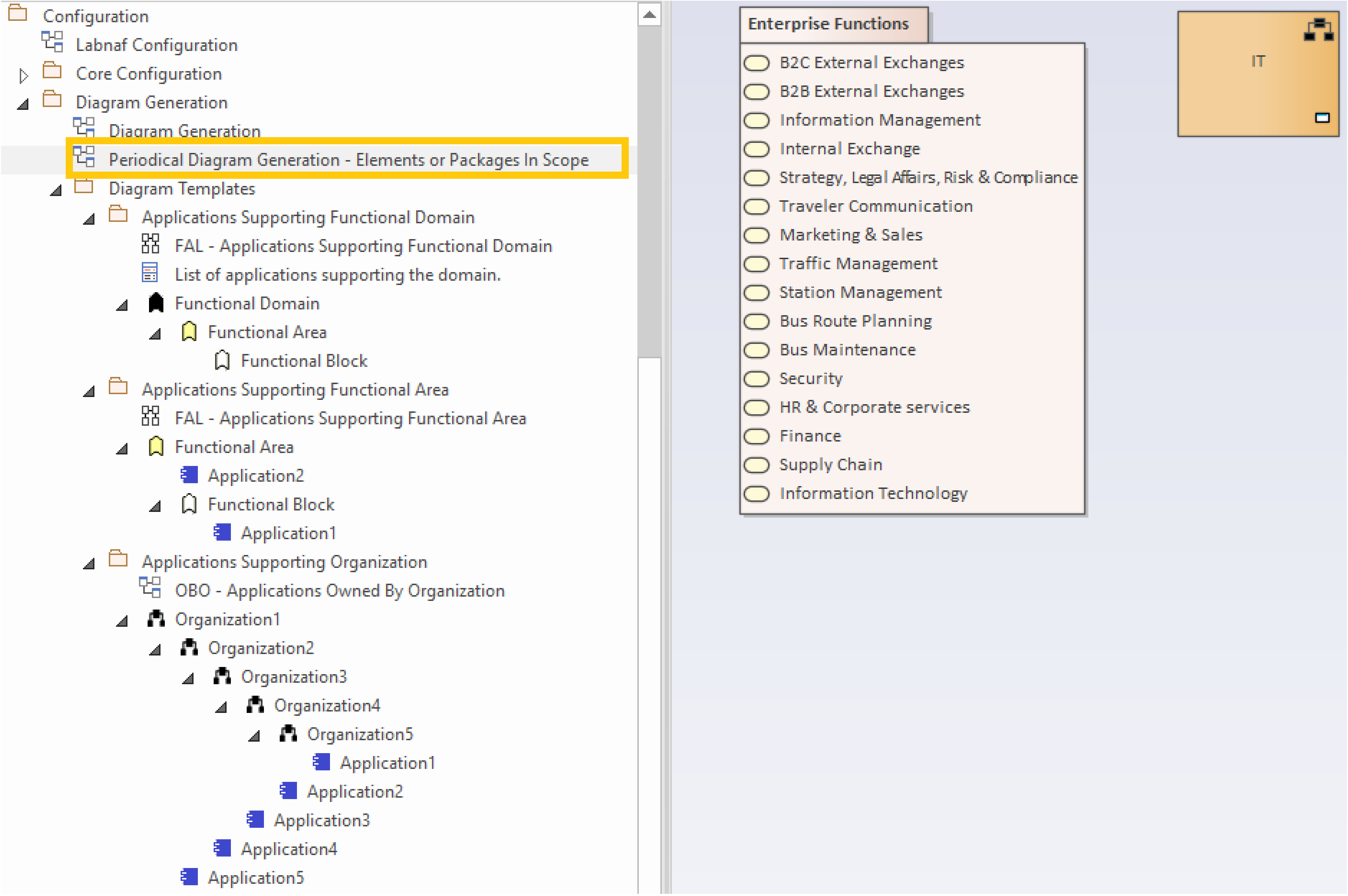
Application Lifecycle / Vision

- 🟨 New
- 🟩 Invest
- 🟩 Maintain
- 🟪 Phase Out
- 🟥 ?

Other diagram(s) for this organization:

Configuration

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

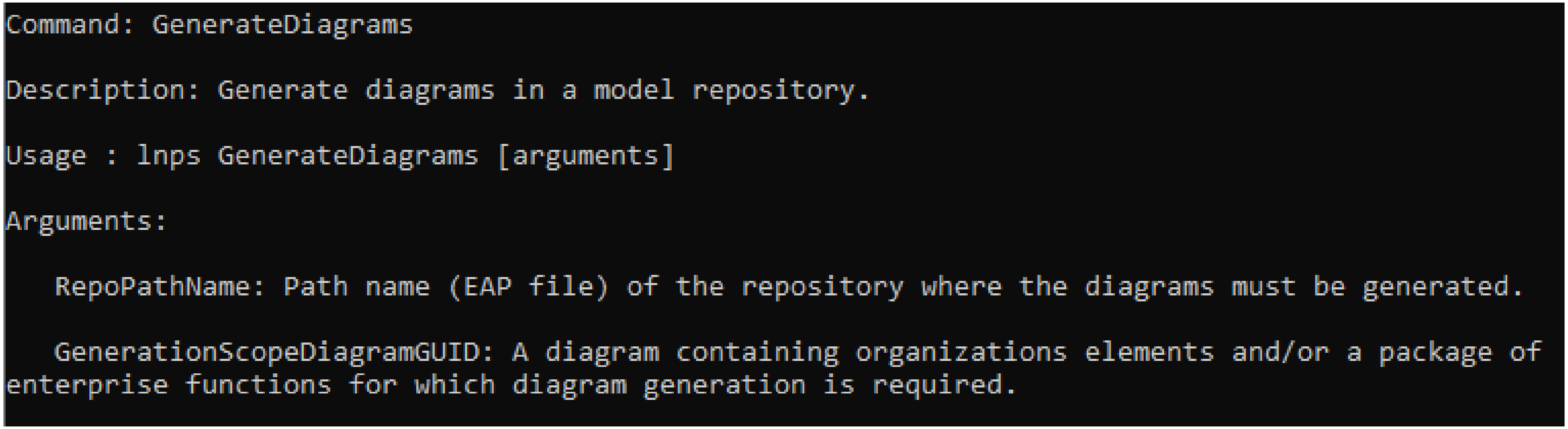
Enterprise Functions

- 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

IT

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.
```

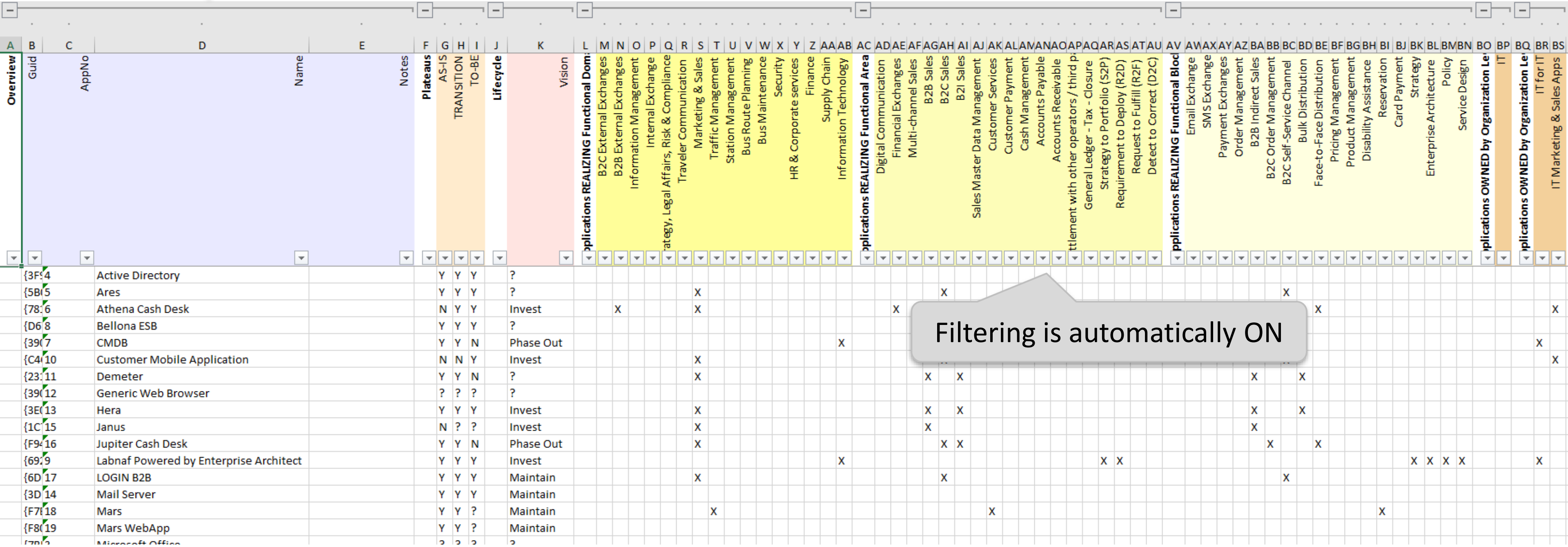
Generate Tabular Reports

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.

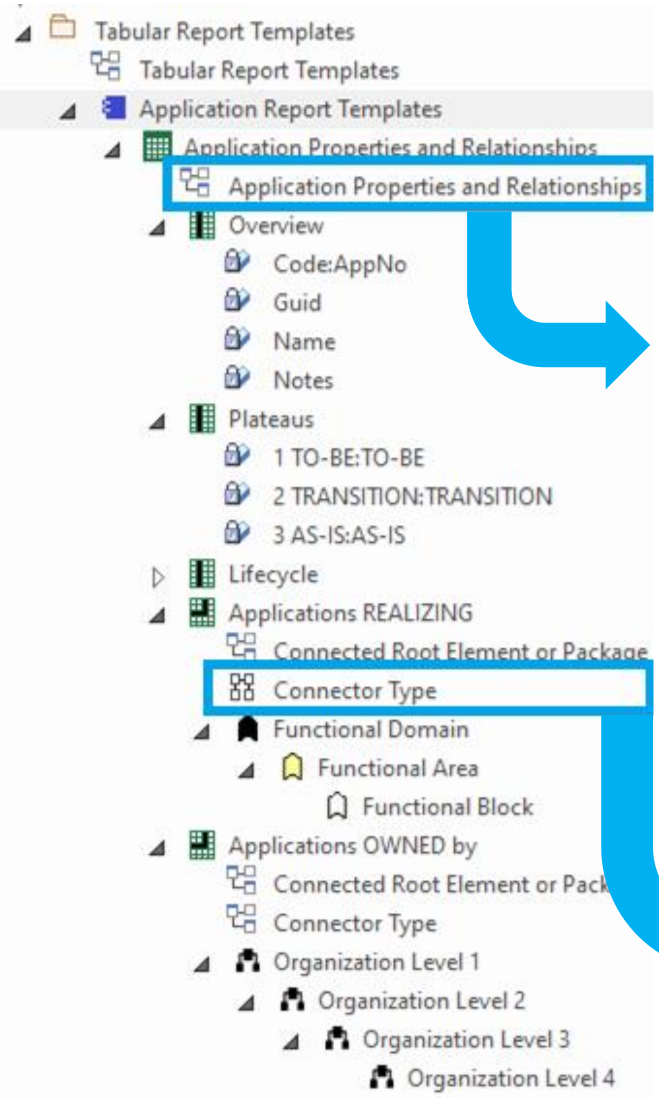


A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS
Overview	Guid	AppNo	Name	Notes	**Plateaus**	AS-IS	TRANSITION	TO-BE	**Lifecycle**	Vision	**pplications REALIZING Functional Dom:**	B2C External Exchanges	B2B External Exchanges	Information Management	Internal Exchange	ategy, 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	**pplications REALIZING Functional Area**	Digital Communication	Financial Exchanges	Multi-channel Sales	B2B Sales	B2C Sales	B2I Sales	Sales Master Data Management	Customer Services	Customer Payment	Cash Management	Accounts Payable	Accounts Receivable	ttlement with other operators / third p	General Ledger - Tax - Closure	Strategy to Portfolio (S2P)	Requirement to Deploy (R2D)	Request to Fulfill (R2F)	Detect to Correct (D2C)	**pplications REALIZING Functional Bloc**	Email Exchange	SMS Exchange	Payment Exchanges	Order Management	B2B Indirect Sales	B2C Order Management	B2C Self-Service Channel	Bulk Distribution	Face-to-Face Distribution	Pricing Management	Product Management	Disability Assistance	Reservation	Card Payment	Strategy	Enterprise Architecture	Policy	Service Design	**pplications OWNED by Organization Le**	IT	**pplications OWNED by Organization Le**	IT for IT	IT Marketing & Sales Apps
	{3F!	4	Active Directory			Y	Y	Y		?																																																												
	{5B(	5	Ares			Y	Y	Y		?								X															X																						X															
	{78:	6	Athena Cash Desk			N	Y	Y		Invest			X					X												X																											X														X
	{D6	8	Bellona ESB			Y	Y	Y		?																																																													
	{39(	7	CMDB			Y	Y	N		Phase Out																	X																																										X	
	{C4(	10	Customer Mobile Application			N	N	Y		Invest								X															X																						X														X
	{23:	11	Demeter			Y	Y	N		?								X														X		X																		X			X															
	{39(	12	Generic Web Browser			?	?	?		?																																																											
	{3E(	13	Hera			Y	Y	Y		Invest								X														X		X																		X			X															
	{1C'	15	Janus			N	?	?		Invest								X														X																				X																		
	{F9/	16	Jupiter Cash Desk			Y	Y	N		Phase Out								X															X	X																			X			X														
	{69:	9	Labnaf Powered by Enterprise Architect			Y	Y	Y		Invest																	X																X	X																			X	X	X	X				X	
	{6D	17	LOGIN B2B			Y	Y	Y		Maintain								X															X																						X															
	{3D	14	Mail Server			Y	Y	Y		Maintain																																																											
	{F7E	18	Mars			Y	Y	?		Maintain									X																	X																									X										
	{F8(	19	Mars WebApp			Y	Y	?		Maintain																																																											
	{7B(	3	Microsoft Office			?	?	?		?																																																											

Filtering is automatically ON

Generate Tabular Report (cont.)

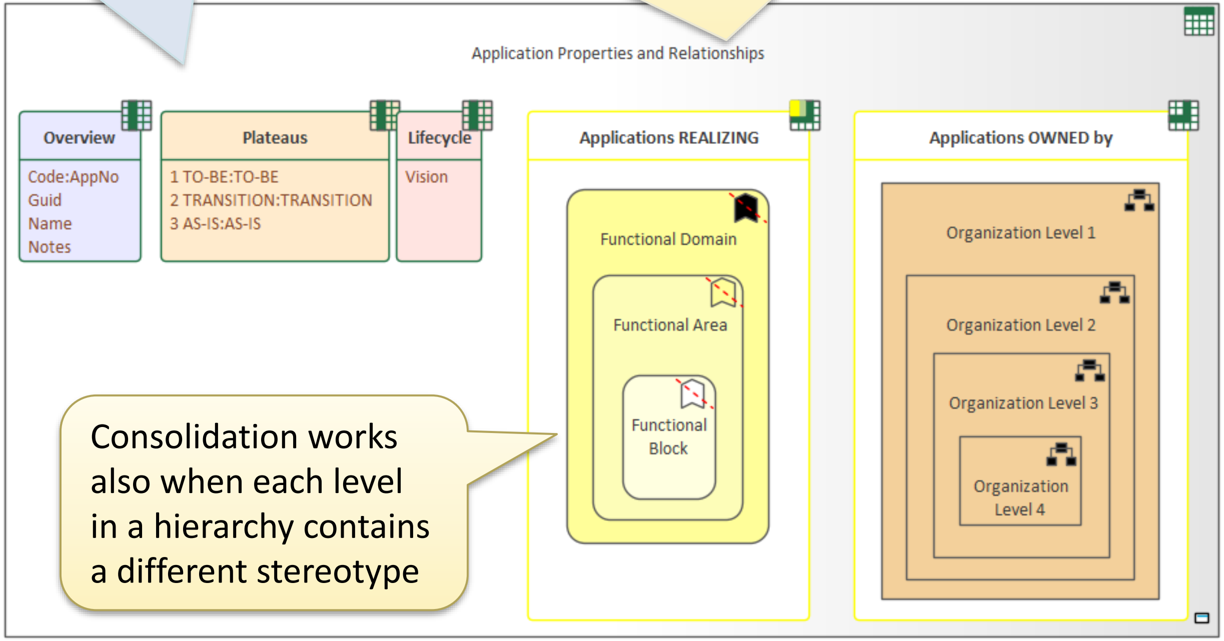
Model your tabular report



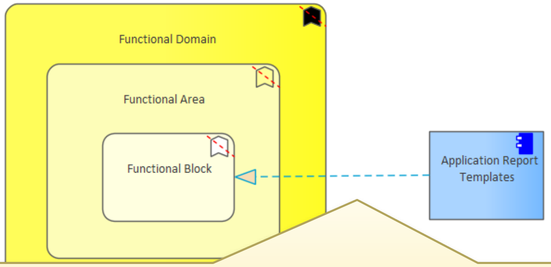
- Tabular Report Templates
  - Tabular Report Templates
  - Application Report Templates
    - Application Properties and Relationships
      - Application Properties and Relationships
      - Overview
        - Code:AppNo
        - Guid
        - Name
        - Notes
      - Plateaus
        - 1 TO-BE:TO-BE
        - 2 TRANSITION:TRANSITION
        - 3 AS-IS:AS-IS
      - Lifecycle
      - Applications REALIZING
        - Connected Root Element or Package
        - Connector Type
        - Functional Domain
          - Functional Area
            - Functional Block
      - Applications OWNED by
        - Connected Root Element or Pack
        - Connector Type
        - Organization Level 1
          - Organization Level 2
            - Organization Level 3
              - Organization Level 4

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

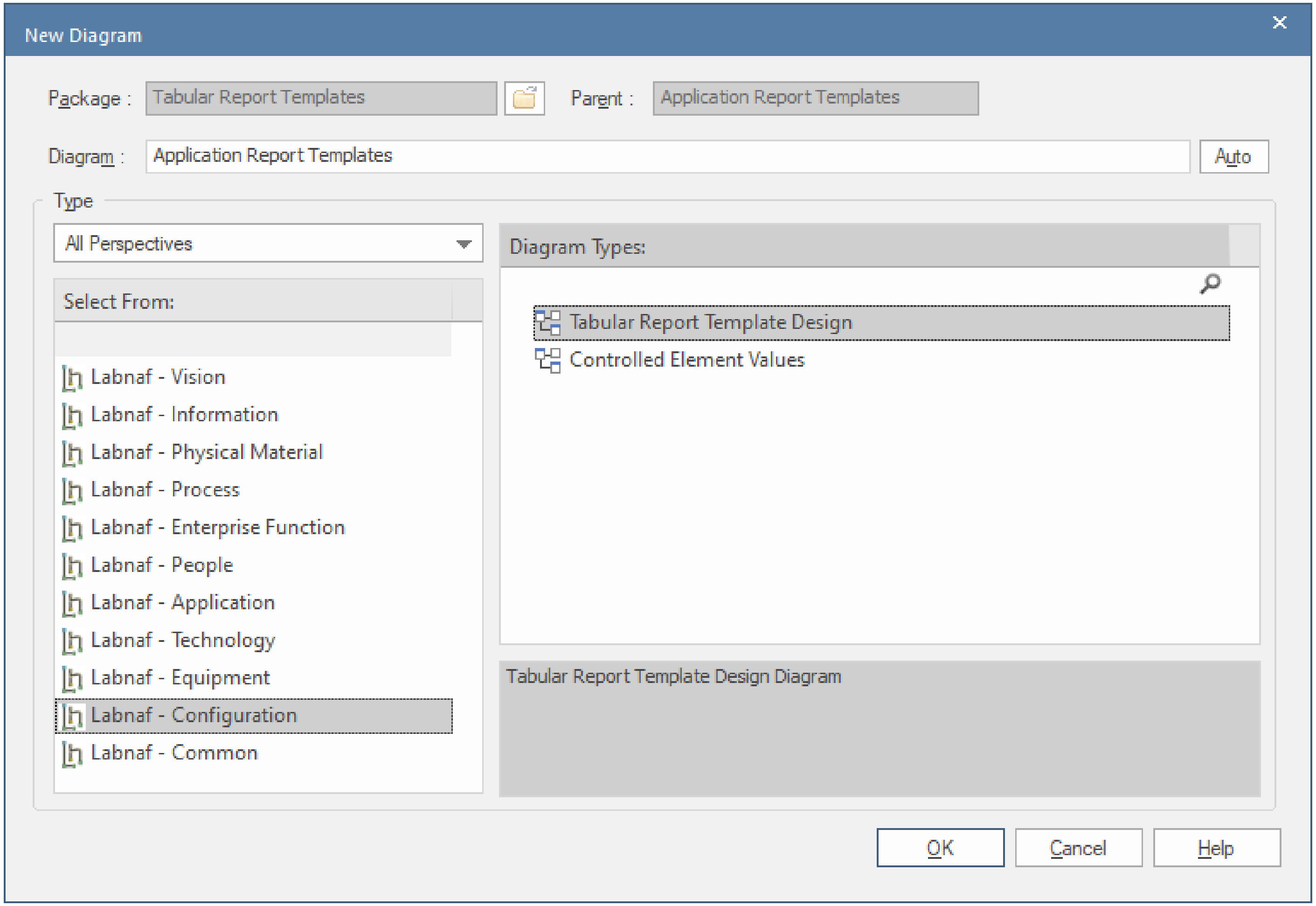


Model leaf element **connector** to be reported and consolidated

Generate Tabular Report (cont.)

Model your tabular report

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



New Diagram

Package : Tabular Report Templates

Parent : Application Report Templates

Diagram : Application Report Templates

Type: All Perspectives

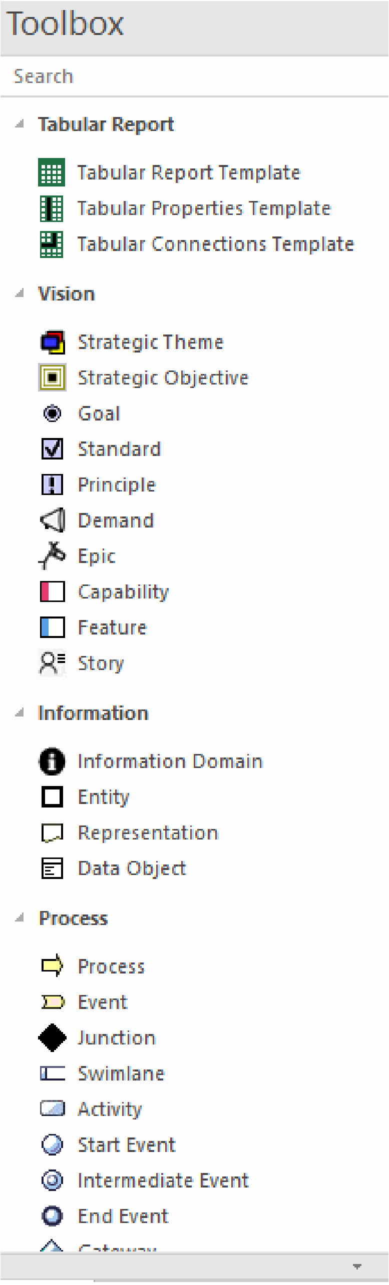
Select From:

- Labnaf - Vision
- Labnaf - Information
- Labnaf - Physical Material
- Labnaf - Process
- Labnaf - Enterprise Function
- Labnaf - People
- Labnaf - Application
- Labnaf - Technology
- Labnaf - Equipment
- Labnaf - Configuration
- Labnaf - Common

Diagram Types:

- Tabular Report Template Design
- Controlled Element Values

Tabular Report Template Design Diagram



Toolbox

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

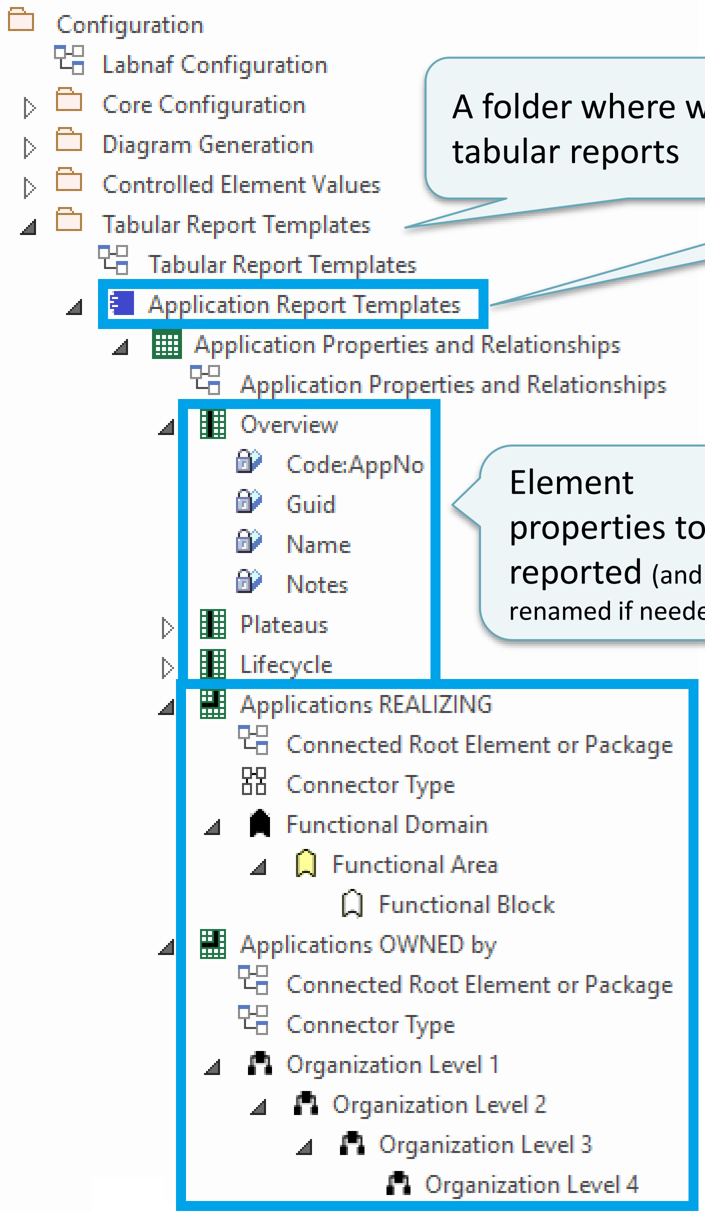
Vision
- Strategic Theme
- Strategic Objective
- Goal
- Standard
- Principle
- Demand
- Epic
- Capability
- Feature
- Story

Information
- Information Domain
- Entity
- Representation
- Data Object

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

Generate Tabular Report (cont.)

Model your tabular report



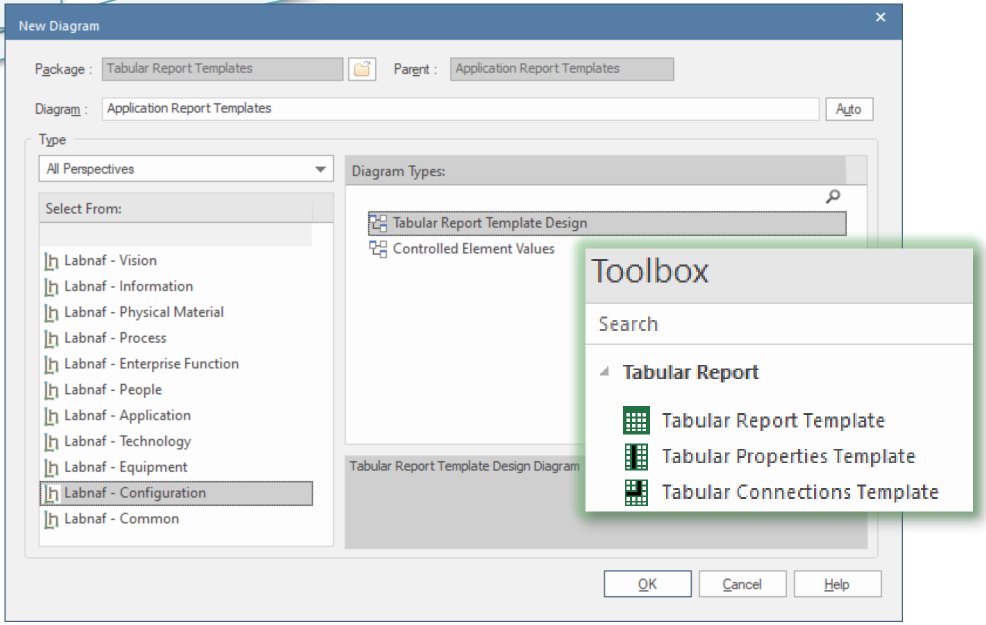
- Configuration
  - Labnaf Configuration
  - Core Configuration
  - Diagram Generation
  - Controlled Element Values
  - Tabular Report Templates
    - Tabular Report Templates
    - Application Report Templates
      - Application Properties and Relationships
        - Application Properties and Relationships
        - Overview
          - Code:AppNo
          - Guid
          - Name
          - Notes
        - Plateaus
        - Lifecycle
        - 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

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.



New Diagram

Package : Tabular Report Templates

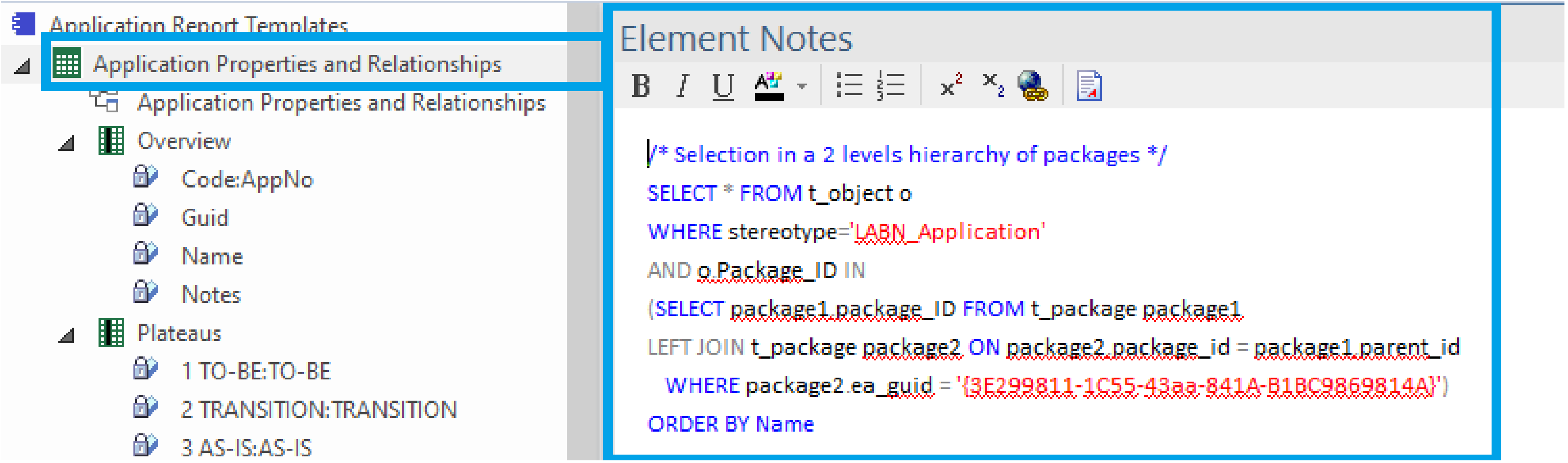
Parent : Application Report Templates

Diagram : Application Report Templates

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



Application Report Templates
Application Properties and Relationships
Application Properties and Relationships
Overview
Code:AppNo
Guid
Name
Notes
Plateaus
1 TO-BE:TO-BE
2 TRANSITION:TRANSITION
3 AS-IS:AS-IS
Element Notes

```
/* 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.

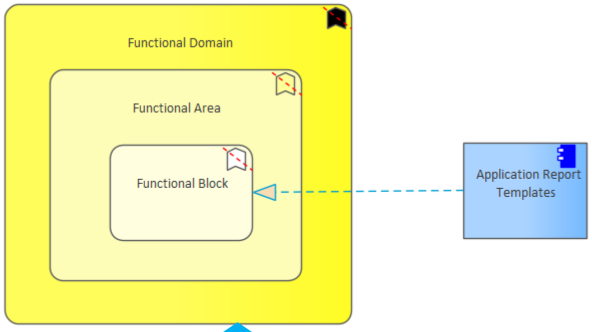
Generate Tabular Report (cont.)

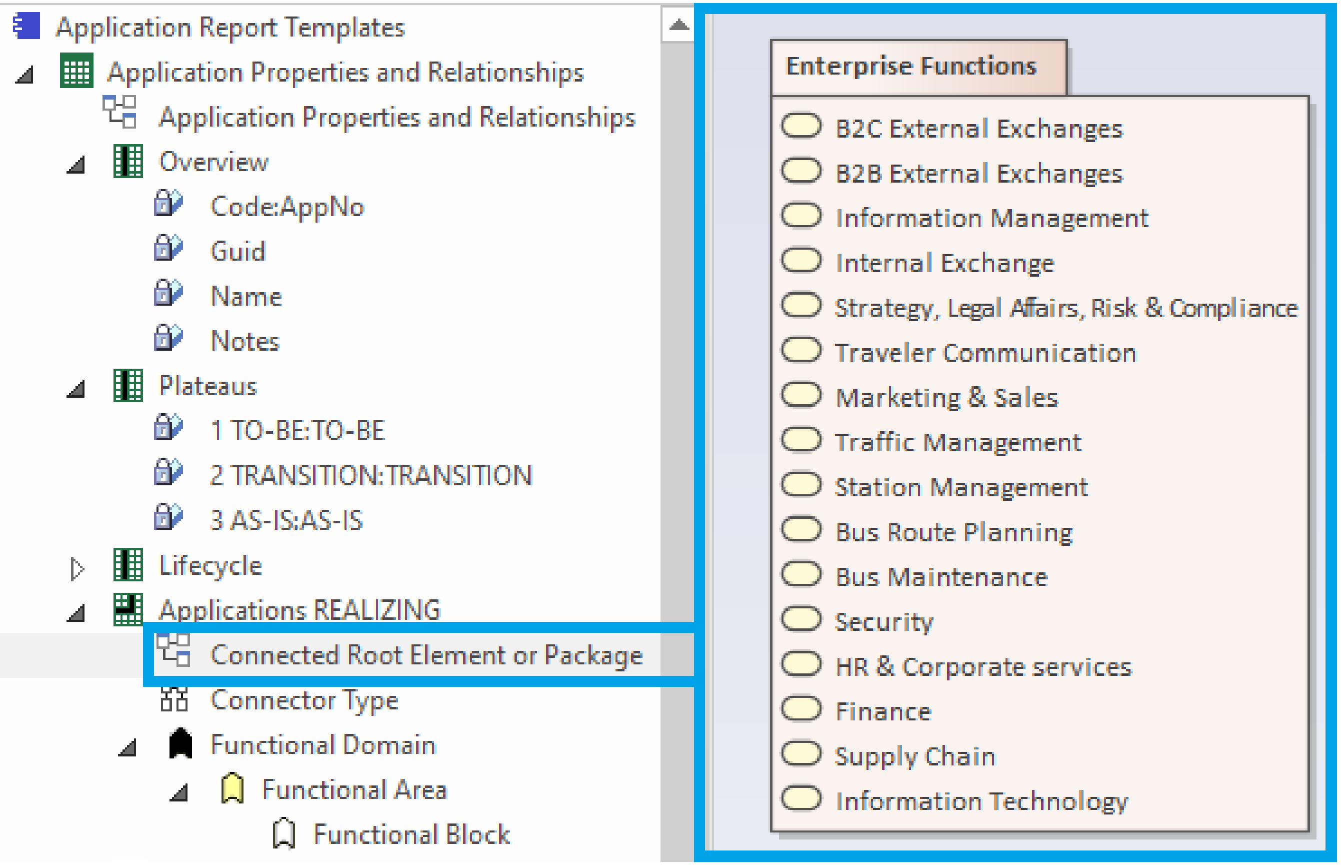
If you want to limit the scope of the reported connections…

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





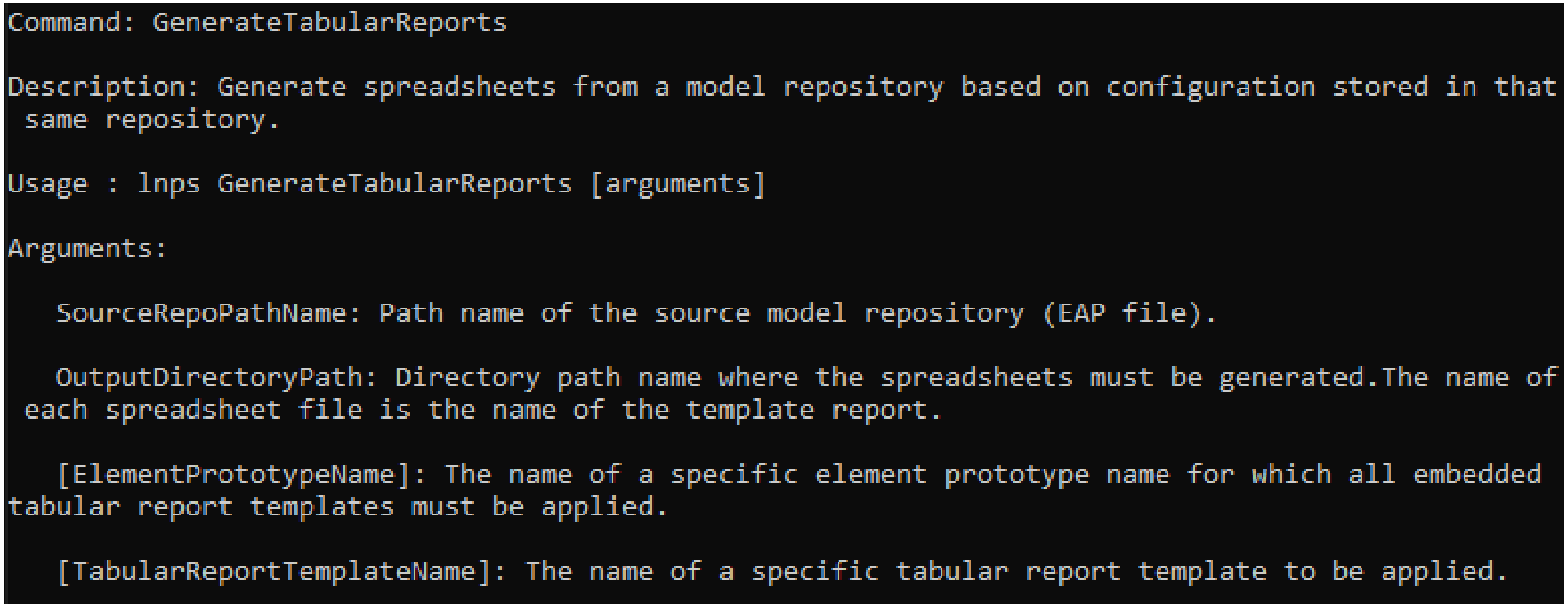
- Application Report Templates
  - Application Properties and Relationships
    - Application Properties and Relationships
    - Overview
      - Code:AppNo
      - Guid
      - Name
      - Notes
    - Plateaus
      - 1 TO-BE:TO-BE
      - 2 TRANSITION:TRANSITION
      - 3 AS-IS:AS-IS
    - Lifecycle
    - Applications REALIZING
      - Connected Root Element or Package
      - Connector Type
      - Functional Domain
        - Functional Area
          - Functional Block

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

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.
```

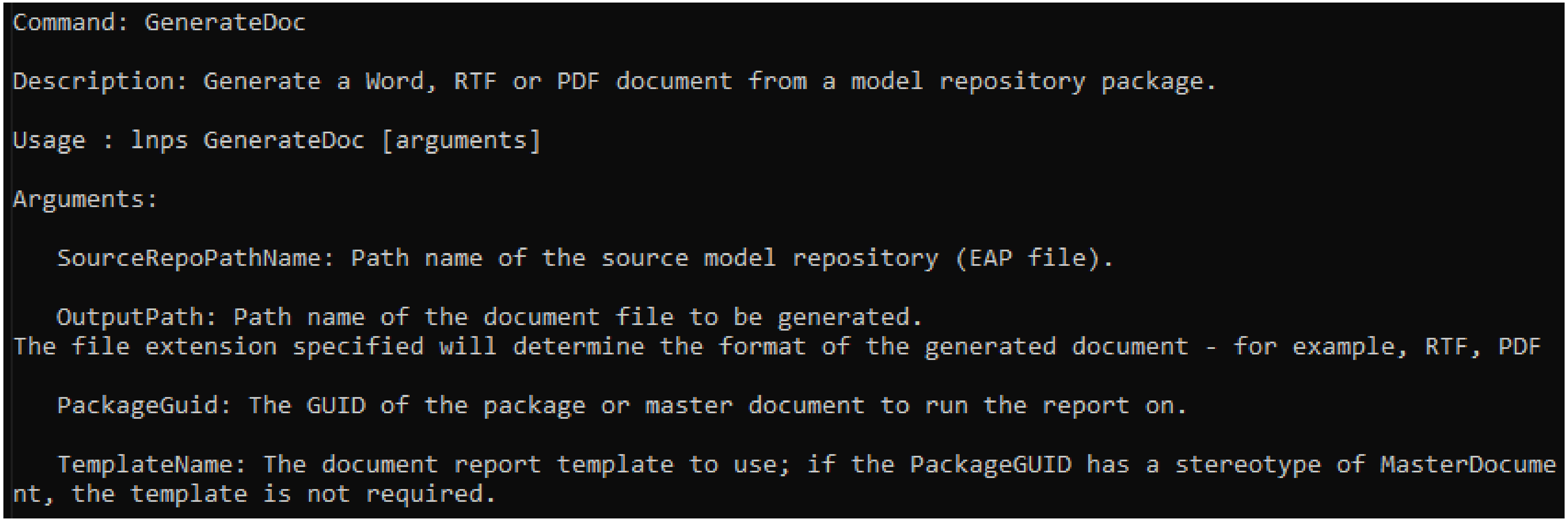
By default, all report templates will be applied.

But you can also be specific.

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

Generate Doc
(Word, RTF, PDF)

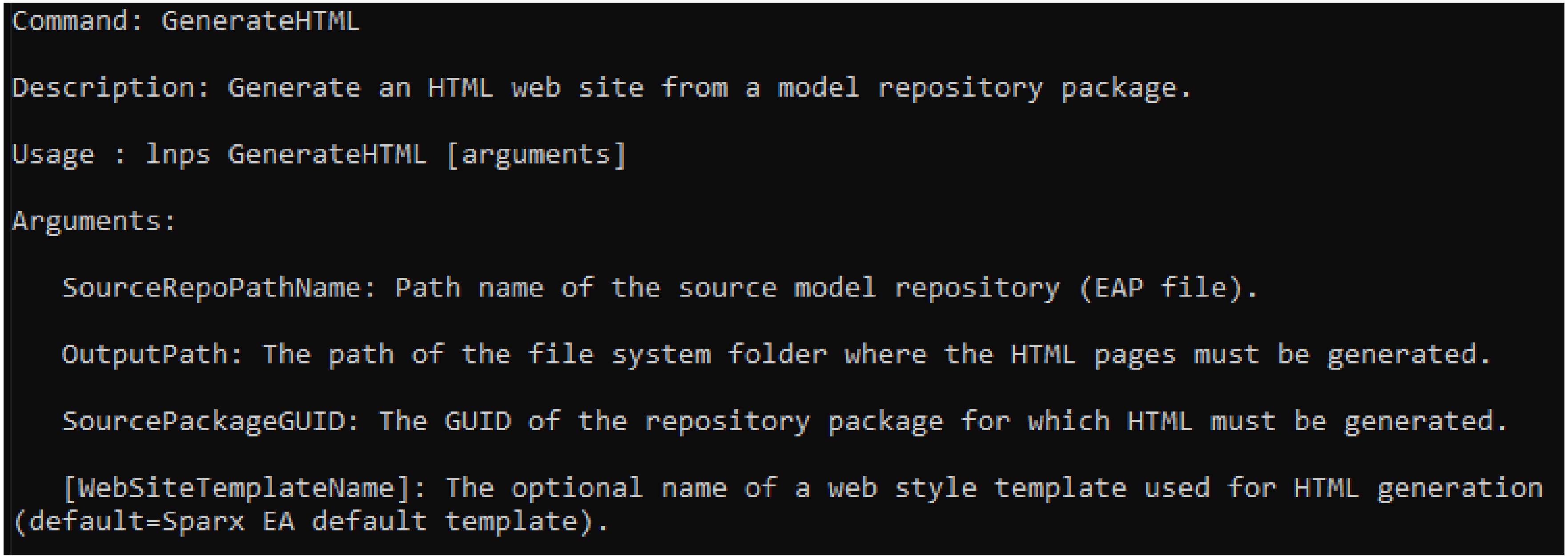
To start document generation:



```
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.
```

Generate Html

To start HTML generation:



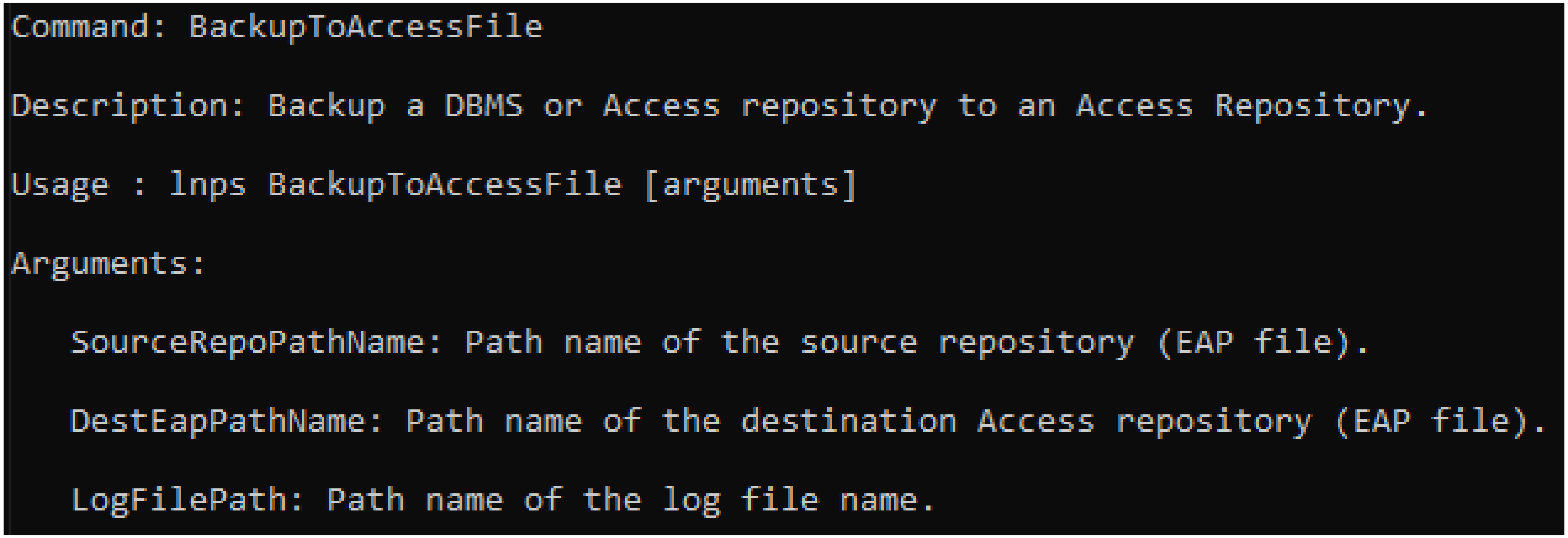
```
Command: GenerateHTML

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.

✉

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

Schedule Command

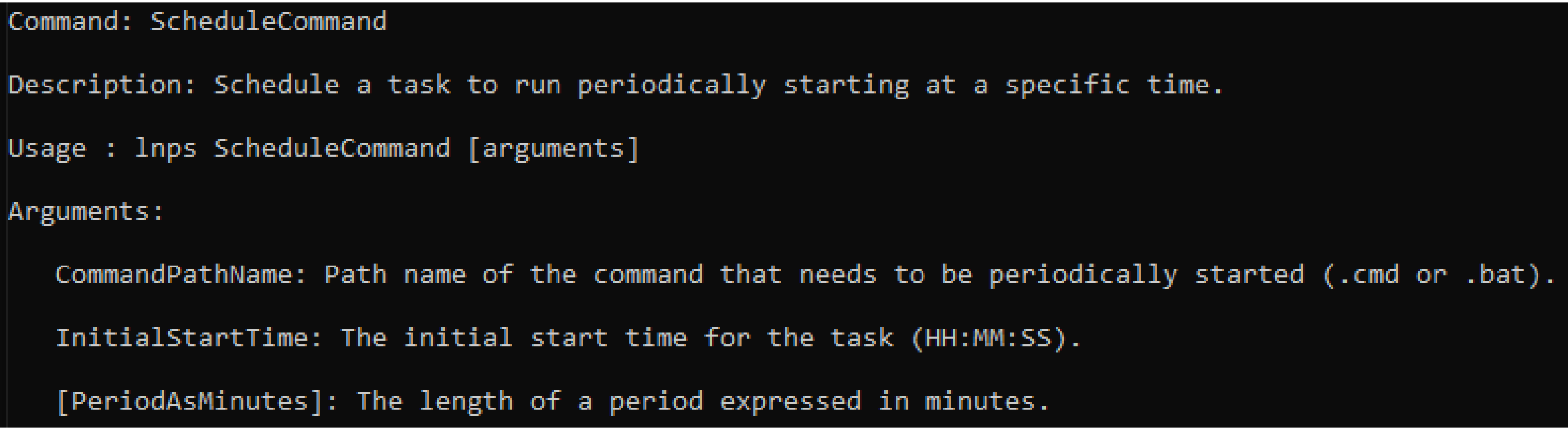
To schedule a **nightly** command **starting at midnight**:

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

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

There are 1440 minutes in a day

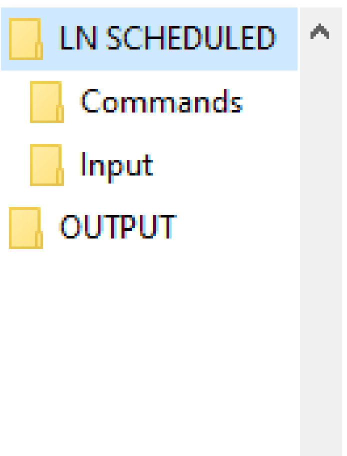
To start the schedular:



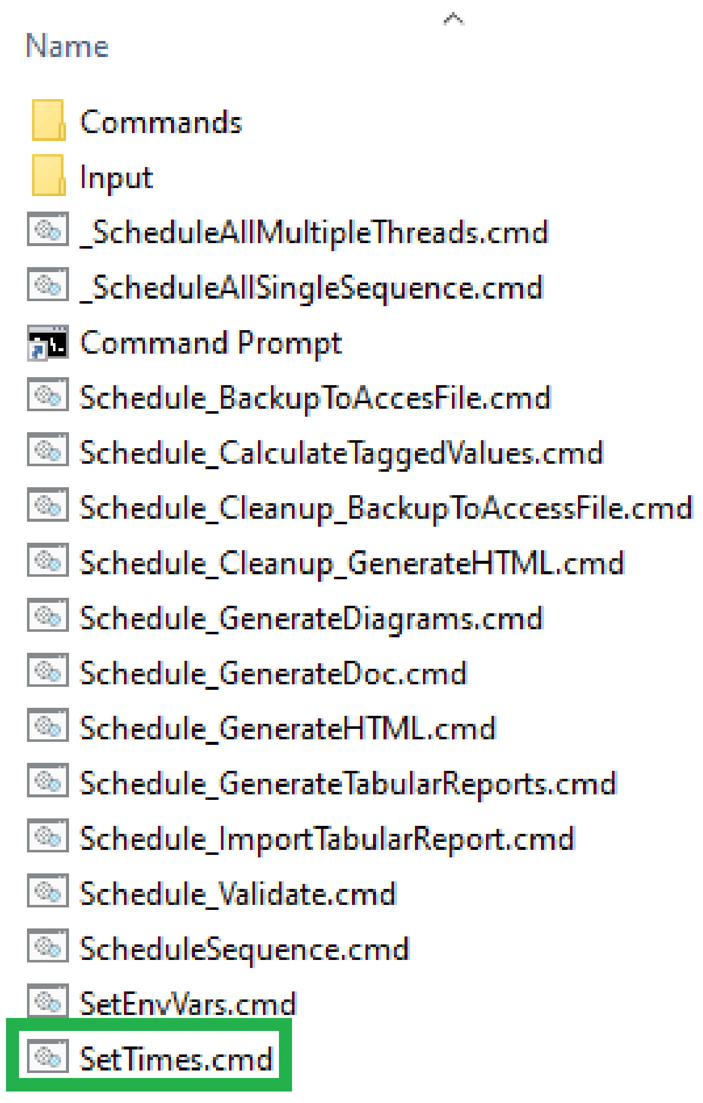
```
Command: ScheduleCommand

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.
```

Final Recommendation

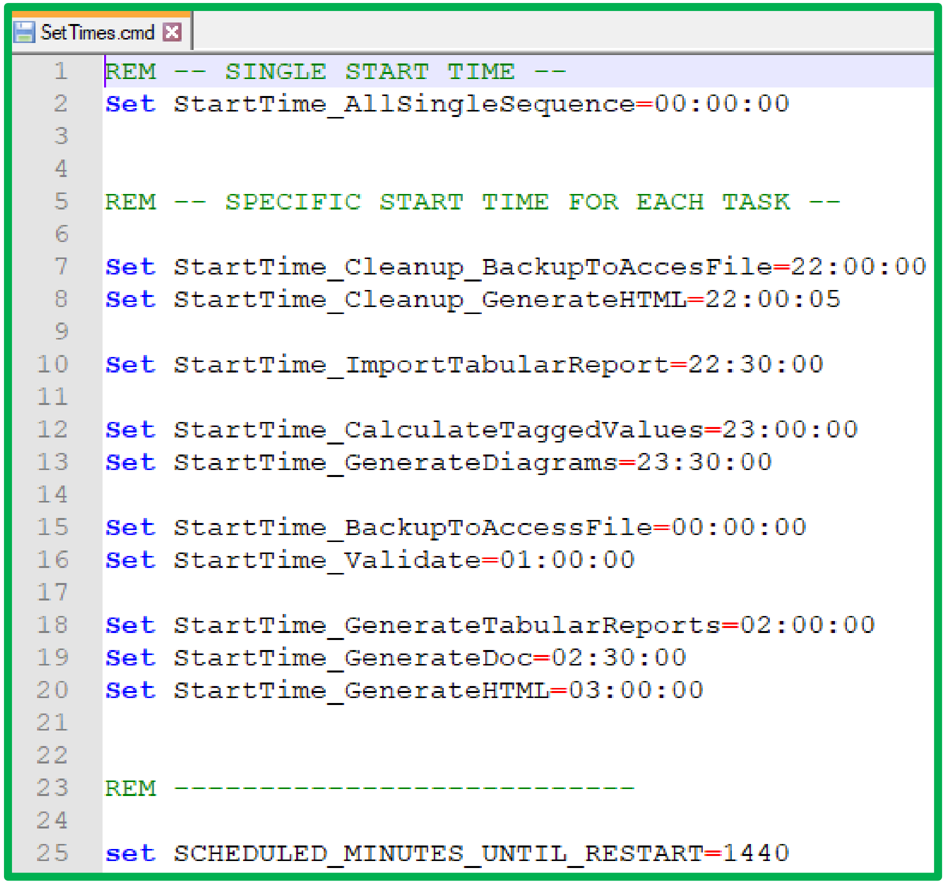
Use the preconfigured batches and settings



LN SCHEDULED
Commands
Input
OUTPUT



Name
Commands
Input
_ScheduleAllMultipleThreads.cmd
_ScheduleAllSingleSequence.cmd
Command Prompt
Schedule_BackupToAccesFile.cmd
Schedule_CalculateTaggedValues.cmd
Schedule_Cleanup_BackupToAccessFile.cmd
Schedule_Cleanup_GenerateHTML.cmd
Schedule_GenerateDiagrams.cmd
Schedule_GenerateDoc.cmd
Schedule_GenerateHTML.cmd
Schedule_GenerateTabularReports.cmd
Schedule_ImportTabularReport.cmd
Schedule_Validate.cmd
ScheduleSequence.cmd
SetEnvVars.cmd
SetTimes.cmd



SetTimes.cmd

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


REM -- SPECIFIC START TIME FOR EACH TASK --

Set StartTime_Cleanup_BackupToAccesFile=22:00:00
Set StartTime_Cleanup_GenerateHTML=22:00:05

Set StartTime_ImportTabularReport=22:30:00

Set StartTime_CalculateTaggedValues=23:00:00
Set StartTime_GenerateDiagrams=23:30:00

Set StartTime_BackupToAccessFile=00:00:00
Set StartTime_Validate=01:00:00

Set StartTime_GenerateTabularReports=02:00:00
Set StartTime_GenerateDoc=02:30:00
Set StartTime_GenerateHTML=03:00:00


REM ---------------------------

set SCHEDULED_MINUTES_UNTIL_RESTART=1440
```

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