# Unified Framework for Driving Transformations

# Labnaf PowerShell
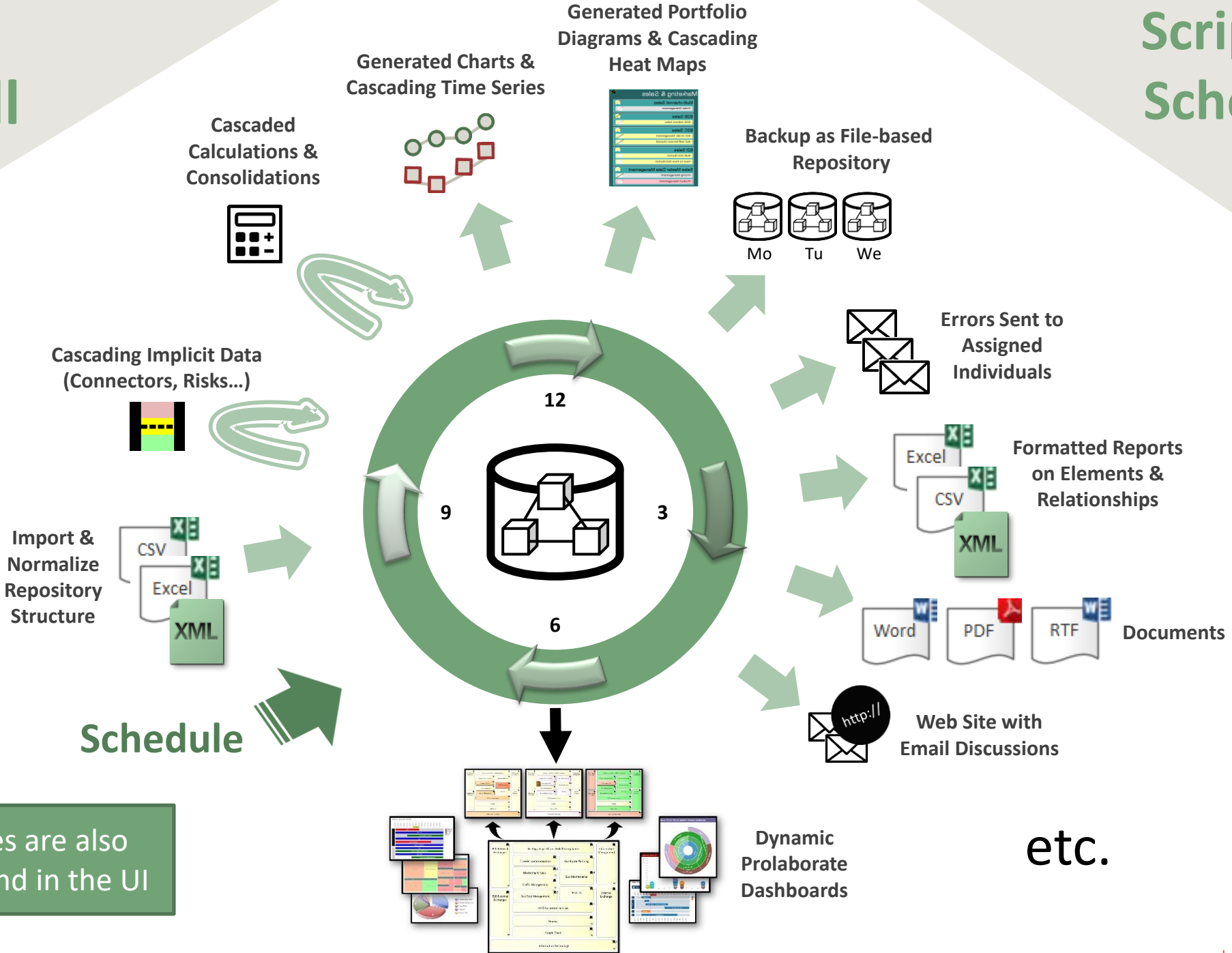
# **User Guide**

# Labnaf PowerShell Commands

Related resources (latest versions)

- **Labnaf PowerShell Reference Guide**: https://www.labnaf.one/EndUserMaterial/Labnaf_PowerShell/Labnaf%20PowerShell%20-%20Reference%20Guide.pdf
- **Labnaf On-line Guidance:** https://www.labnaf.one/guidance/index.html?guid=569FF62A-5210-4359-923F-4EB00EB03D61
- **Sample data:** Provided with the Labnaf PowerShell software

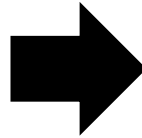www.labnaf.one

# Labnaf PowerShell

# Scripting & Scheduling

Generated Portfolio Diagrams & Cascading Heat Maps

Generated Charts & Cascading Time Series

Cascaded Calculations & Consolidations

Backup as File-based Repository

Mo  Tu  We

Cascading Implicit Data (Connectors, Risks...)

Errors Sent to Assigned Individuals

12

9    3

6

Import & Normalize Repository Structure

CSV

Excel

XML

Formatted Reports on Elements & Relationships

Excel

CSV

XML

Word  PDF  RTF   Documents

Schedule

Web Site with Email Discussions

http://

Dynamic Prolaborate Dashboards

The same features are also available on demand in the UI

etc.

# Running commands on the command line

**lnps**  ➡

```
Usage : "C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe" [-]Command [arguments]
Available Commands: (Prefix the command name with '-' to run in non verbose mode)
    AutoConnectorsDelete
    AutoConnectorsGenerate
    BackupToAccessFile
    CalculateTaggedValues
    ClonePackage
    CreatePackage
    DeleteGeneratedCharts
    DeleteGeneratedDiagrams
    ExportToXmi
    GenerateCharts
    GenerateDiagrams
    GenerateDoc
    GenerateHTML
    GenerateImplicitData
    GenerateTabularReports
    ImportConnections
    ImportFromXmi
    ImportTabularReport
    MoveElementsToCalculatedParent
    MoveElementsToPackage
    MovePackagesToPackage
    RenameItem
    ScheduleCommand
    SetDiagramProperty
    SqlExportToCsv
    Validate

    ? => Show details for all commands
```

**Labnaf PowerShell Program**

- "**LNPS**" is the name of the Labnaf PowerShell program.

- Full path is "**C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe**"

4

# Running commands on the command line

**lnps ?**

⬇
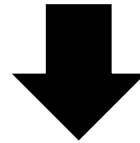
**Shows a detailed description of all commands and their usage**

```
C:\Program Files (x86)\Labnaf\PowerShell>lnps ?

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).


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).


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.


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
```

# Running commands on the command line

**lnps [-]{command name}**

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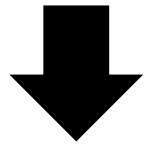
**Example: if you type « lnps GenerateTabularReports » you get the following info.**

```
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.
```

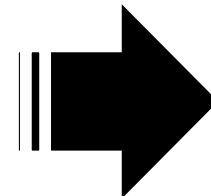Prefix the command name with '-' to run in non verbose mode

www.labnaf.one
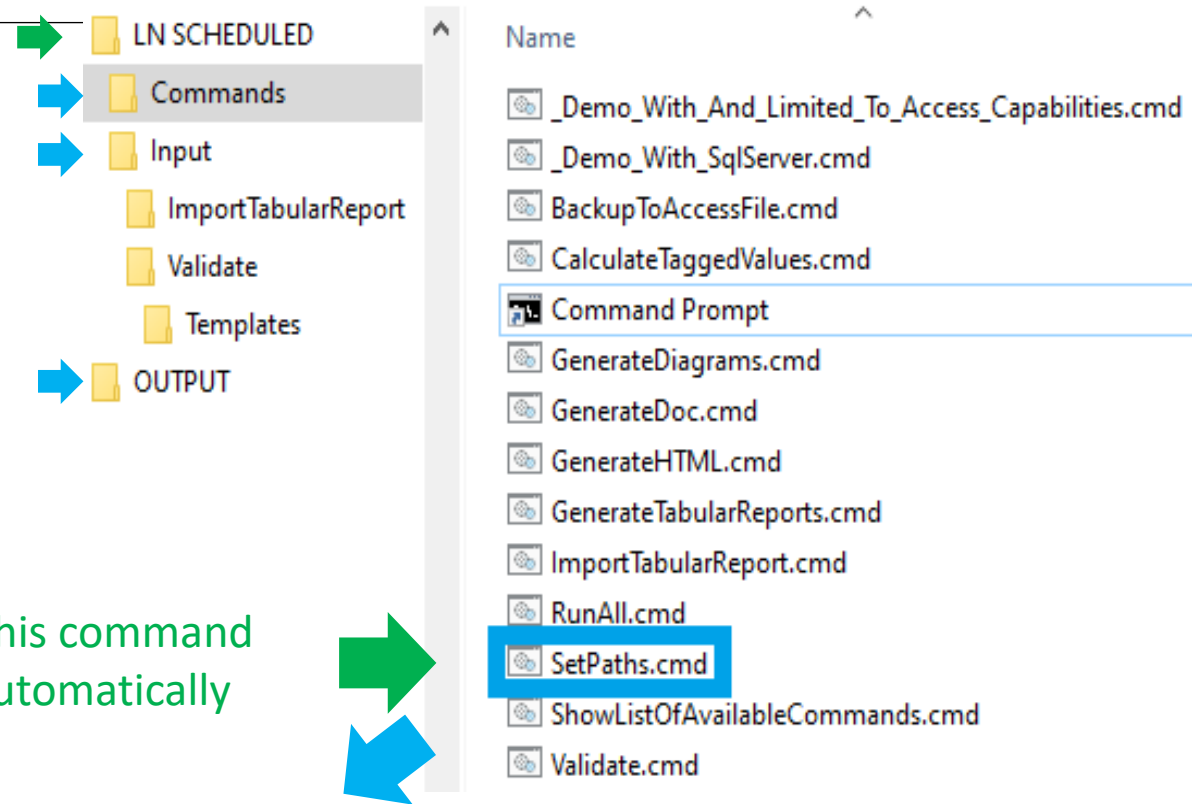
# Automatic configuration

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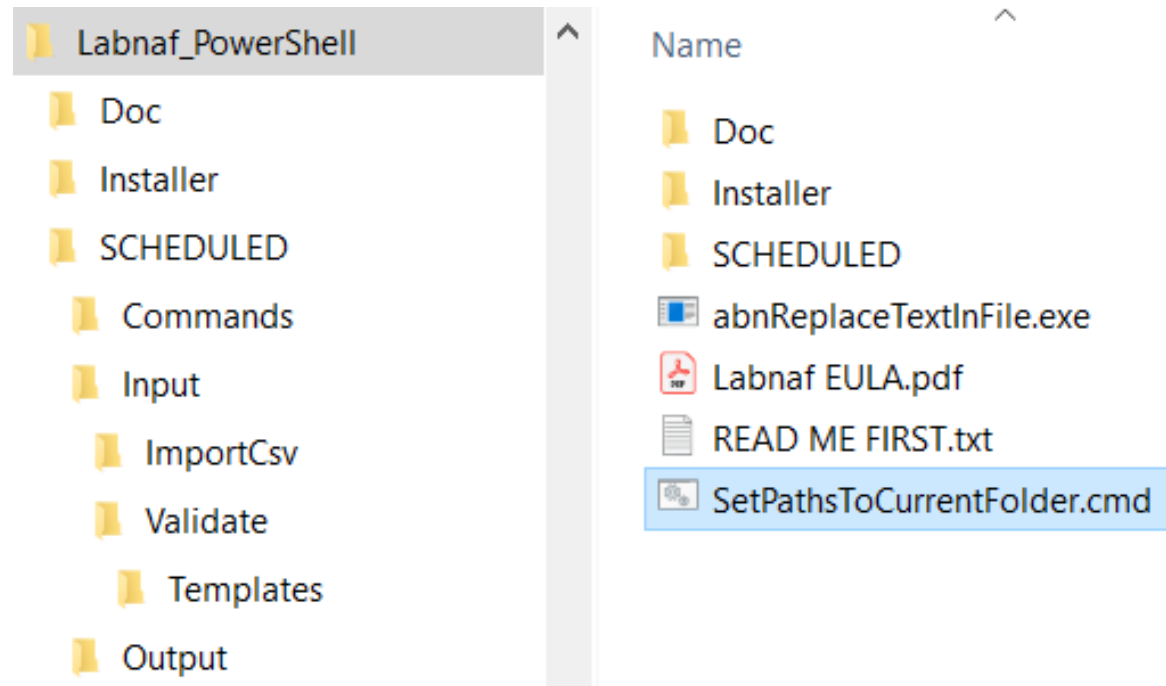


*This updates the Labnaf PowerShell configuration files following the "Labnaf_PowerShell" folder location.*

# Preconfigured batches calling commands with predefined settings

**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

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



```
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
```

**Folder tree:**
- 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

# Labnaf PowerShell Commands

# Labnaf PowerShell commands for **Strategy and architecture operations**

- Import Tabular Report (Excel, CSV)

- Auto Connectors Generate / Delete

- Generate Implicit Data

- Calculate Values

- Validate and send emails to assigned individuals

- Generate Charts / Delete Generated Charts

- Generate Diagrams

- Generate Tabular Report (Excel, CSV), Doc (Word, RTF, PDF), Html

- Backup To Access File

- Schedule Command (not only Labnaf PowerShell commands)

# ImportTabularReport

Sample input data for updating tagged values of existing elements

**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 |

www.labnaf.one

# Import Tabular Report (cont.)

## To start the import:

```
C:\Program Files (x86)\Labnaf\PowerShell>lnps ImportTabularReport

Command: ImportTabularReport

Description: Import elements, properties and tagged values from a CSV or Excel file into a SQL Server database.

        If a field name mapping file (CSV) is provided, the first line must contain the following headers:
            Input_Column_Names, Target_Column_Names
        Works also with Access databases but only for updating existing elements.

        Identify existing elements using
            either the ea_guid,
            or an alternative unique key.
            To define a unique key, you simply add a '#' in front of the property or tag name
            If multiple unique keys are provided, they are searched in this order: Tagged Value, Name, Alias.

        Create the element if the element is not found and if 'EnableCreate' option is present on the command line.
        The package where new elements are added is identified by a package guid provided on the command line
        Initial value calculation applies to any imported new element.


Usage : lnps ImportTabularReport [arguments]

Arguments:

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

   SourceFile: A CSV or Excel file containing the data that needs to be imported.

   ColumnMappingFile: An optional 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.

   [EnableCreate]: Enable creation of new elements if they are not found.

   [TargetPackageGuidForNewElements]: The package where the new elements must be stored.
```

**Beware of the different definitions of the "Notes" / "Note" property name in Sparx EA.**
In the Sparx EA API, the property name is called "Note**s**".
In the database, the corresponding column name is called "Note".

**For Imports**
- In the import file use the word « **Note<u>s</u>** » (and not « Note »)

**For SELECT statements**

- In SQL Server, use the word « Note » in queries.
  Example: `select Name, Note from t_Object`

- In the Sparx EA user interface, use « Note as {Something} ». Indeed « Note » is a reserved word.
  Example: `select Name, Note as Element_Notes from t_Object`

# CalculateTaggedValues

**You first need to model your value calculations**

See the **Value Calculation** User Guide document and the examples on the Guidance Web Site

www.labnaf.one
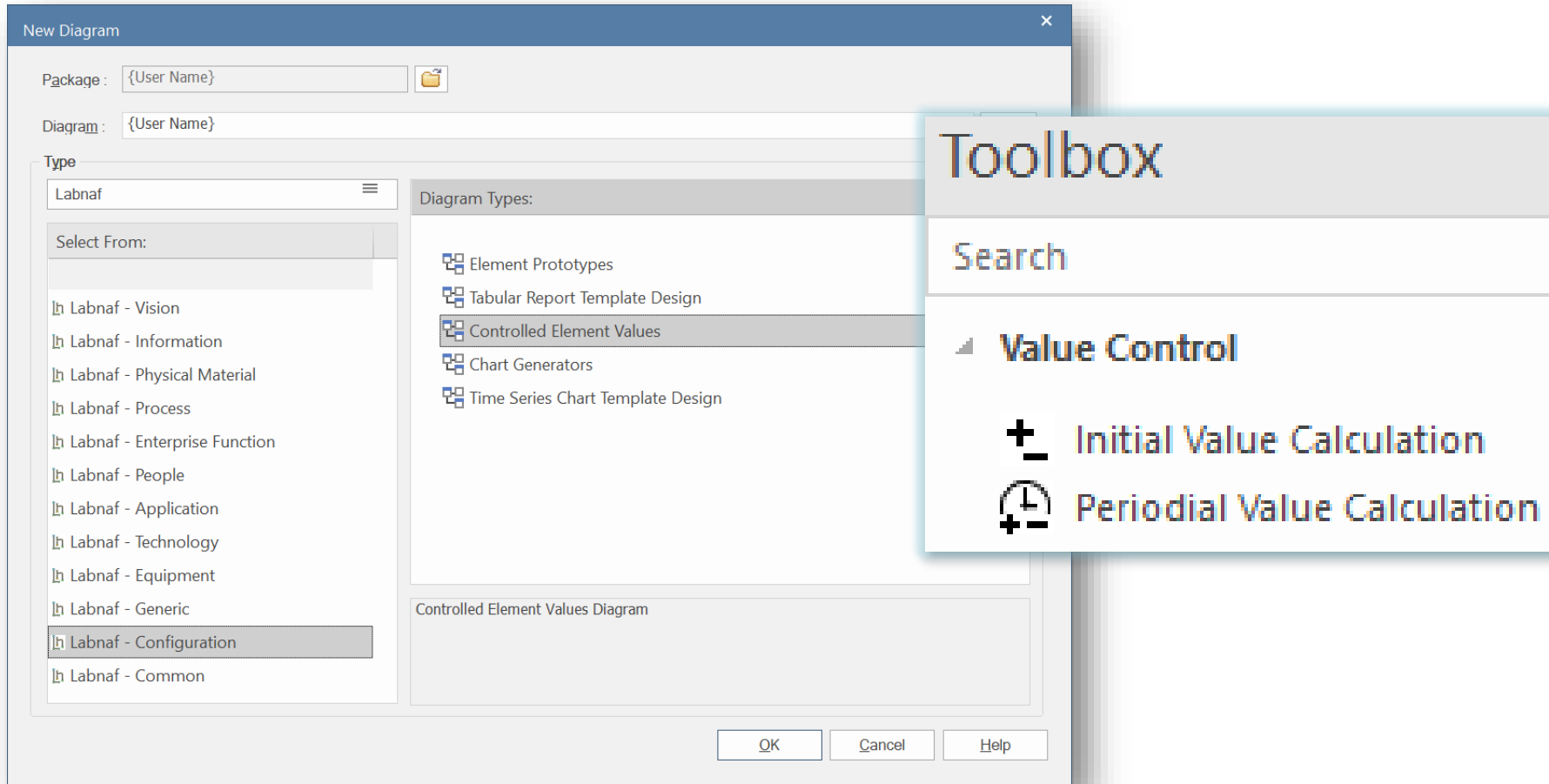
# Calculate Tagged Values (cont.)

**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.
```

# AutoConnectorsGenerate

**To start implicit connector generation:**

```
Command: AutoConnectorsGenerate

Description: Generate implicit connectors following numerous patterns and options, including:
- for child elements following defined element stereotype hierarchies (this option is enabled by default)
- for information elements (entities) used, owned, aggregated or exchanged directly or indirectly by any other element.
See 'Implicit Connectors' on the Labnaf guidance web site to learn more about enabling and disabling options.

Usage : lnps AutoConnectorsGenerate [arguments]

Arguments:

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

The generated implicit connectors are aggregations.

*See the different implicit connector generation options on the guidance web site.*

**Benefits of implicit connector generation**

- Know which information is used by whom, by which role, organization, function, process, activity, data flow, application, component, data store, server, equipment, network, etc
- Know which information is stored where (for example in which country)
- Dramatically simplifies  information security and GDPR compliance analysis
- Simplifies traceability as embedded elements get aggregation connectors
- Normalizes the way elements are related in a repository i.e. based on connectors
- Enables powerful reporting capabilities including Prolaborate charts

www.labnaf.one

# AutoConnectorsDelete

**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.

www.labnaf.one

# GenerateImplicitData

**Implicit Data Generation is configured using Labnaf AddIn user interface**

See **Implicit Data Generation** on the Guidance Web Site
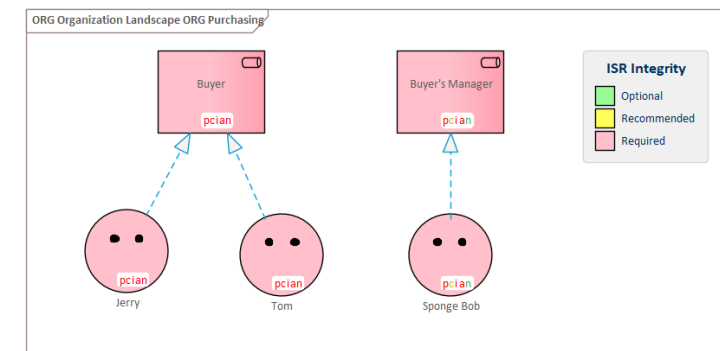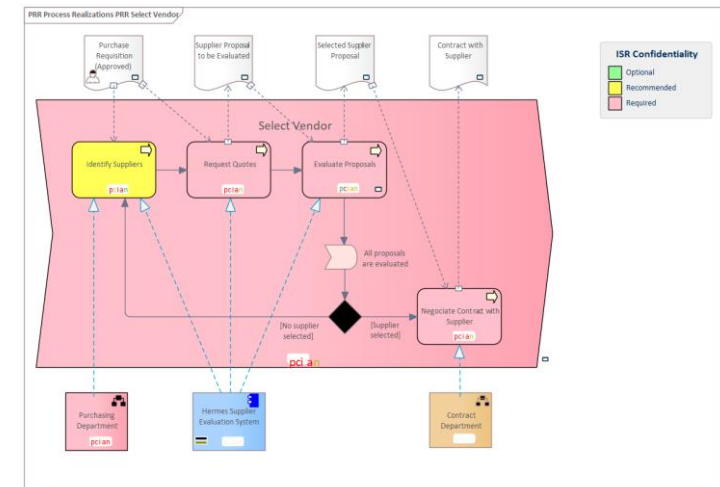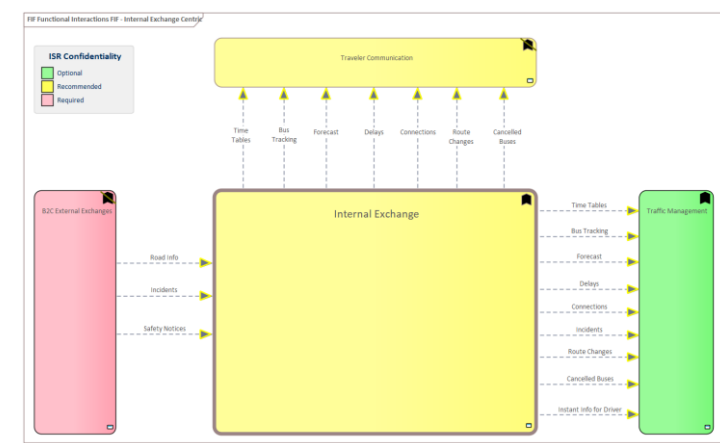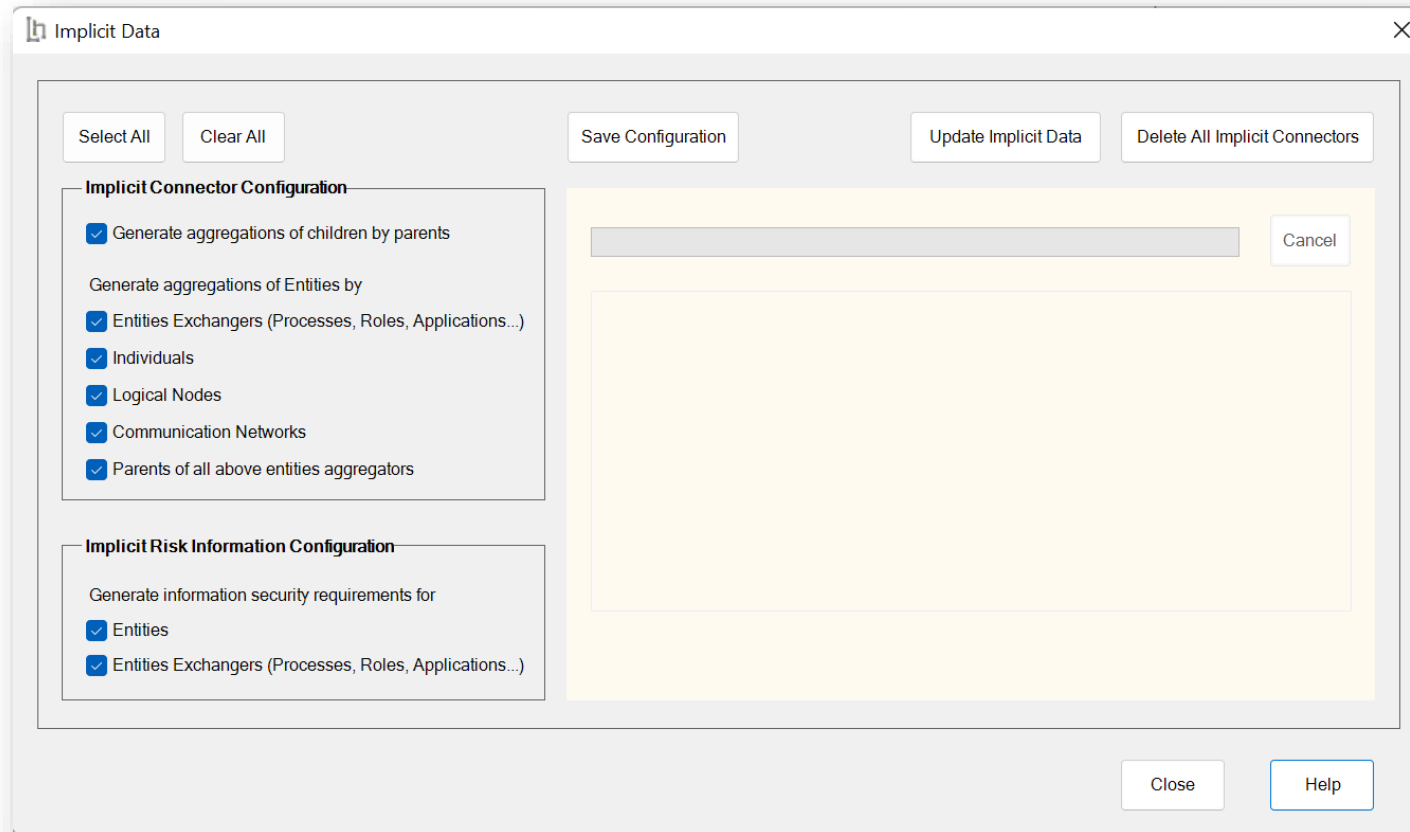
www.labnaf.one

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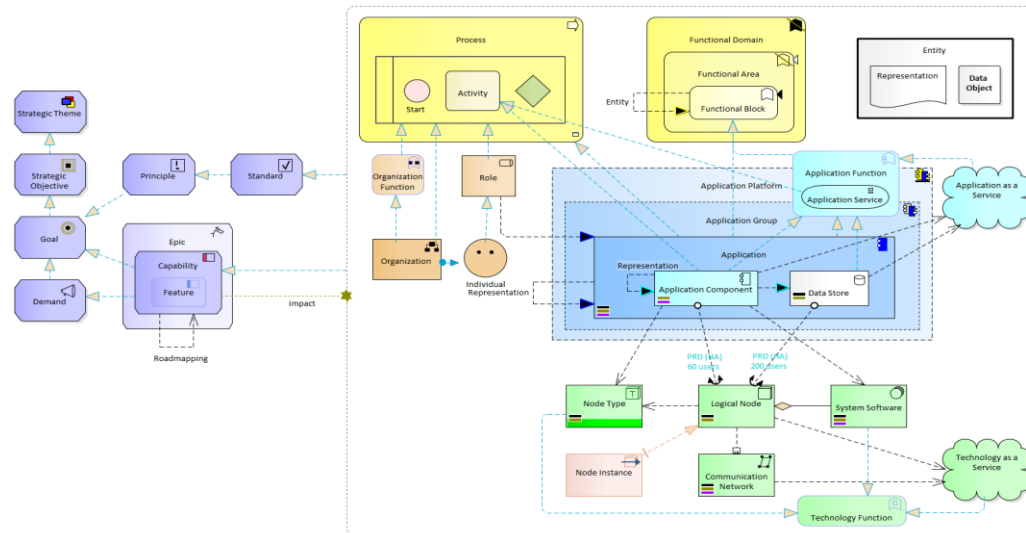
While Modeling

Existing Invalid Connectors

**Prevent creation of invalid connectors**

*Labnaf PowerShell*
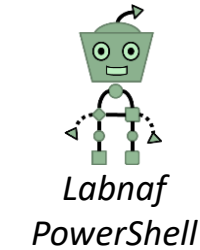
**Send Error Emails to Relevant Recipients**

www.labnaf.one

# 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.

www.labnaf.one

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



**Customized Labnaf MDG (files)**

Sparx Toolbox

Sparx Element or Connector Type

Property Type

Sparx Diagram Type

Sparx Element or Connector Shape

Value

**Add-in or Robot**

**Model Validation**

*Metamodel in memory*

...

update

**Push**

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

**Repository**

*Metamodel used also by end-users as documentation*

*Metamodel as Model*

- Simple
- User-friendly
- Agile

22

# Error message routing is based on architecture management assignments
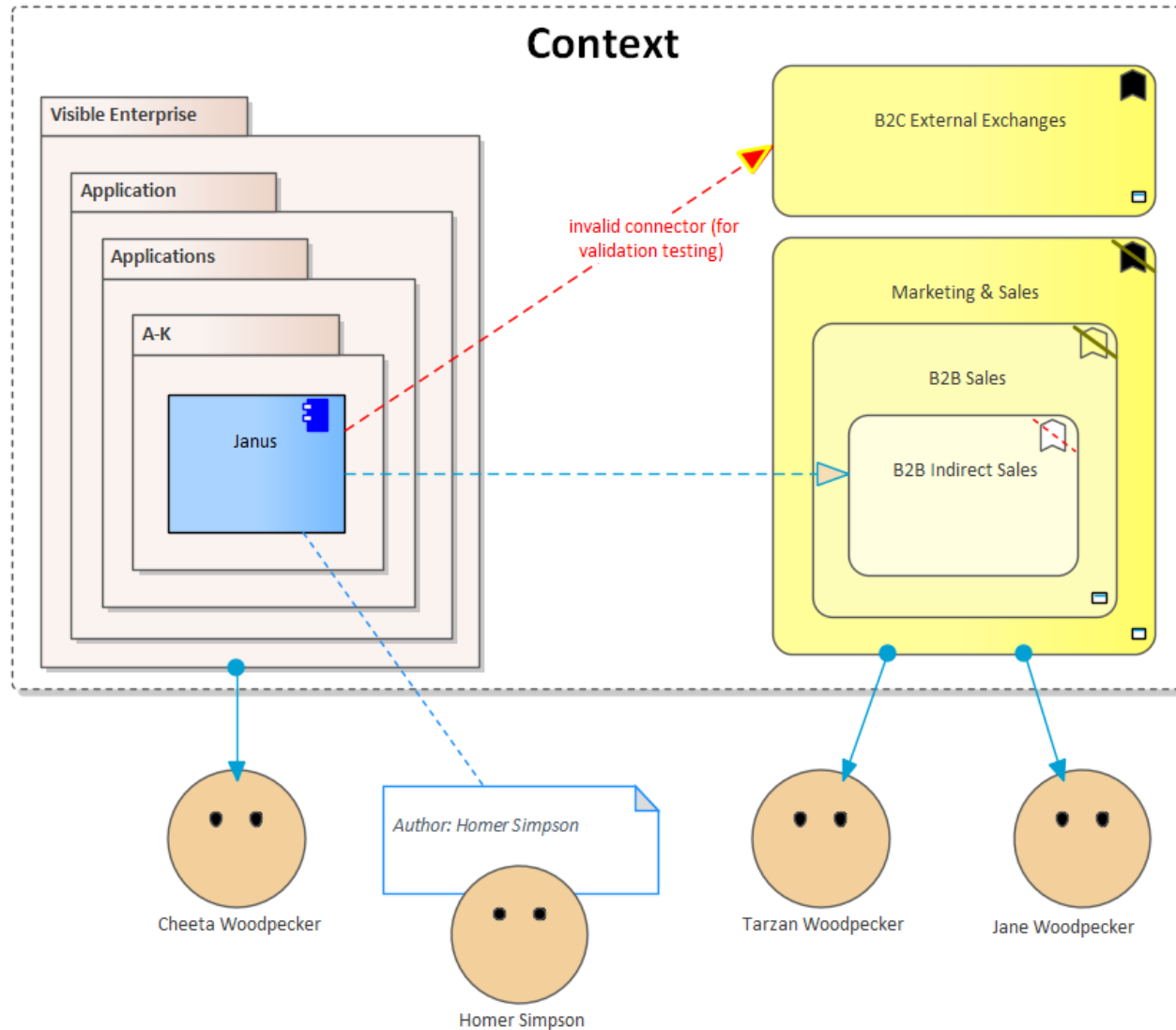


Error message distribution depends on
- recent authors
- individuals assigned to packages
- individuals assigned to domains
- default recipients
- and rules that you can use to combine these different combinations

The catalog of individuals **includes email addresses**

23

# Sample error message sent to an assigned individual

www.labnaf.one

# Validation rules can be further customized

```xml
<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

# Smtp Server configuration is straightforward

```xml
<?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.
```

# GenerateCharts

**Labnaf comes with predefined chart templates,
but you might want model your own chart templates**

See **Chart Generation** on the Guidance Web Site
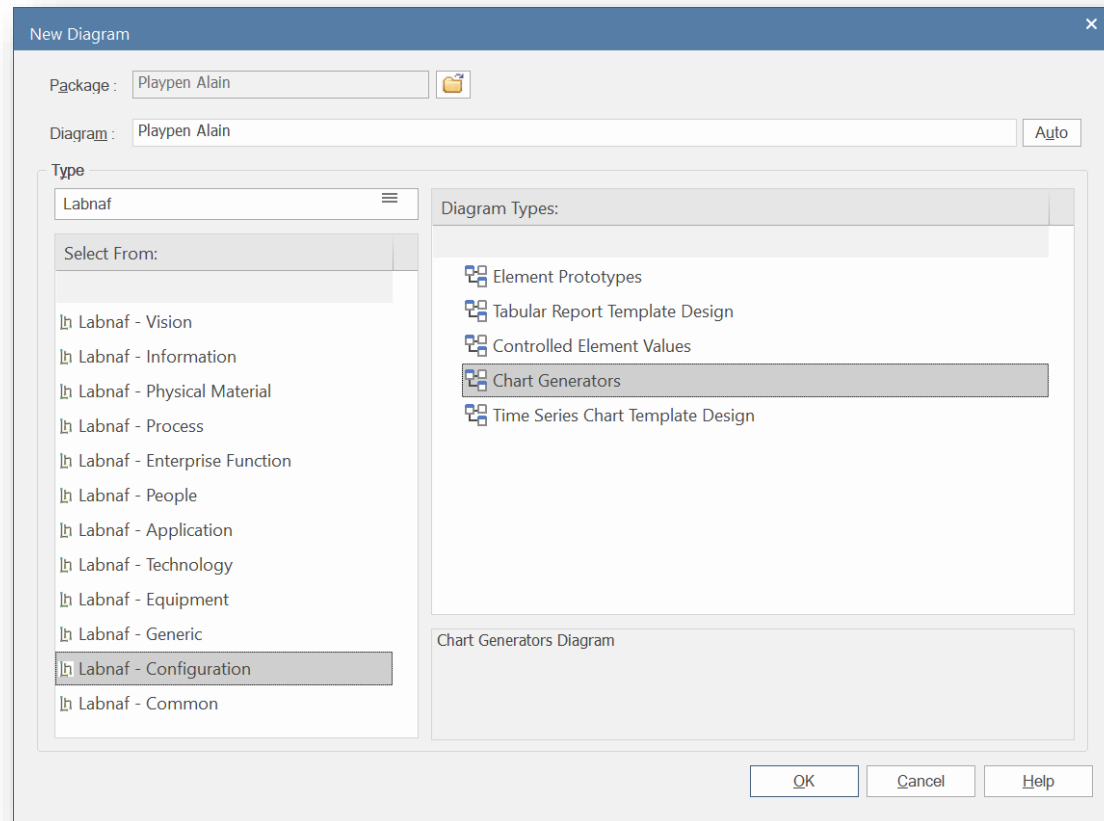


**Sample Generated Time Series Chart**
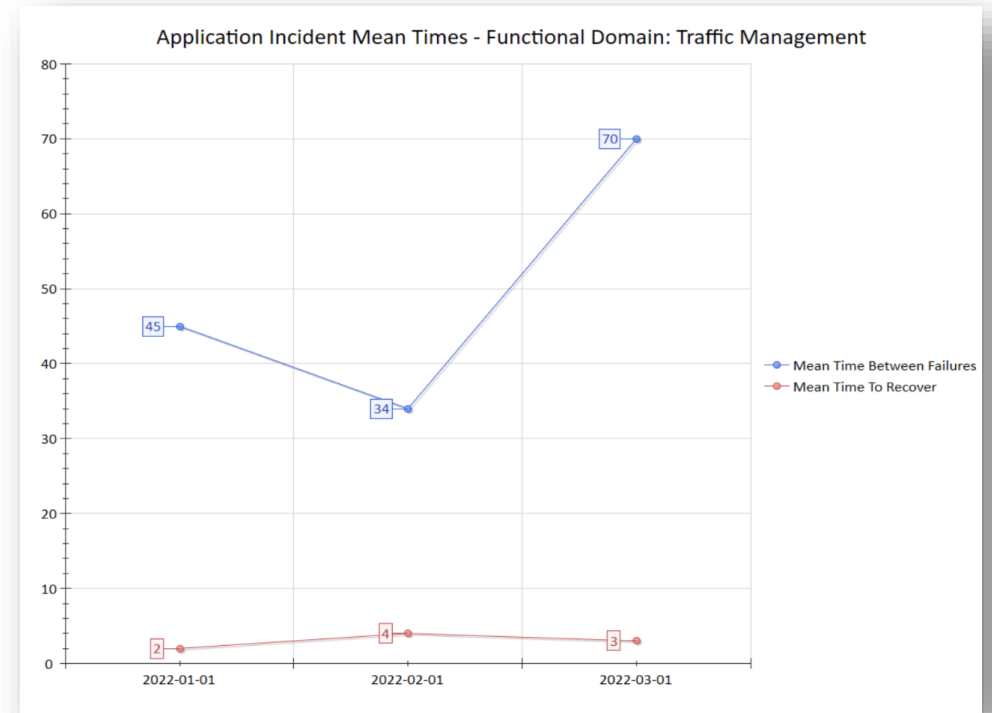
# Generate Charts (cont.)

**To start chart generation:**

```
Command: GenerateCharts

Description: Generate charts for some configured types of element.
        The elements to be selected and the charts to be generated are all defined in the model repository.
        Chart Generator elements that have a name starting with "--" will be ignored.
Usage : "C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe" GenerateCharts [arguments]

Arguments:
    RepoPathName: Repository path name (EAP file).
    [ElementPrototypeName]: A specific element stereotype for which charts must be generated.
    [TemplateChart]: The template chart name defining the type of chart diagram and chart element to be generated.
```

**To delete all generated charts:**

```
Command: DeleteGeneratedCharts

Description: Delete generated charts.
Usage : "C:\Program Files (x86)\Labnaf\PowerShell\lnps.exe" DeleteGeneratedCharts [arguments]

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

# GenerateDiagrams

## Sample Results

### Enterprise Function Taxonomy & Applications Supporting Level 1



### Applications managed by organizations

www.labnaf.one

# Sample Diagram Templates

## Enterprise Function Taxonomy & Applications Supporting Level 1



## Applications managed by organizations

www.labnaf.one

# Items in Scope for Generation



**This is optional**

By default, all elements are selected from the related catalog

# 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.
```

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

Element properties and/or tagged values with optional renaming and coloring

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

**Properties**

**Relationships**

www.labnaf.one

# Generate Tabular Report (cont.)

**Labnaf comes with many tabular report templates, but you might want model your own templates**

See the **Tabular Report Generation** <u>User Guide document</u> and the <u>examples on the Guidance Web Site</u>

**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 scheduled 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 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.

# GenerateDoc
## (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.
```

# GenerateHtml

**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.

# BackupToAccessFile

**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

38

# ScheduleCommand

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 one 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.
```
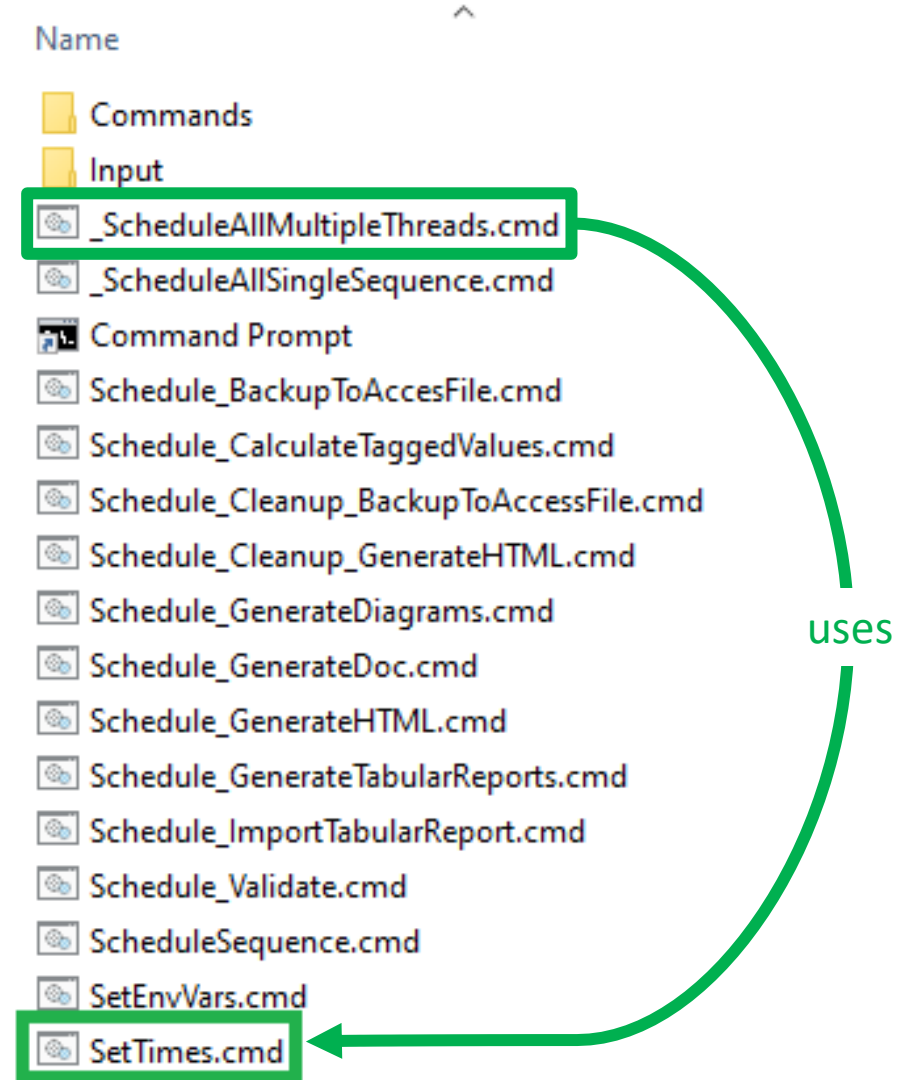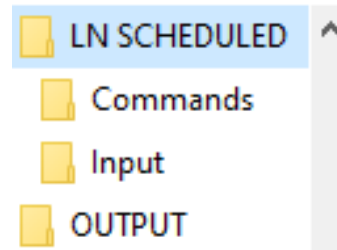
www.labnaf.one

# Final Recommendation in case you use "ScheduleCommand"

**-** 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
```

**Folder tree:**
- 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

uses

**-** Take computer reboots into account

# Labnaf PowerShell Commands

www.labnaf.one

# Labnaf PowerShell commands for
## Systems integrations and content refactoring

- ClonePackage

- CreatePackage

- ExportToXmi

- ImportConnections

- ImportFromXmi

- ImportTabularReport

- MoveElementsToCalculatedParent

- MoveElementsToPackage

- MovePackagesToPackage

- RenameItem

- ScheduleCommand

- SetDiagramProperty

- SqlExportToCsv

Detailed information in the
**Labnaf PowerShell Reference Guide**
Latest version:
https://www.labnaf.one/EndUserMaterial/Labnaf_PowerShell/Labnaf%20PowerShell%20-%20Reference%20Guide.pdf

www.labnaf.one

# Labnaf PowerShell Commands

1. Overview

2. Strategy and Architecture Operations

3. Systems Integrations and Content Refactoring

4. Command Compatibility Matrix

# Labnaf PowerShell
# Command Compatibility Matrix

| Power Shell Commands | Sql Server | Pro Cloud Server | Access |
|---|:---:|:---:|:---:|
| AutoConnectorsDelete | X | X | -> UI |
| AutoConnectorsGenerate | X | X | -> UI |
| BackupToFile / BackupToAccessFile | X | X | |
| CalculateTaggedValues | X | X | |
| ClonePackage | X | X | X |
| CreatePackage | X | X | X |
| ExportToXmi | X | X | X |
| GenerateDiagrams | X | X | X |
| GenerateCharts | X | X | |
| GenerateDoc | X | | |
| GenerateHTML | X | | |
| GenerateImplicitData | X | X | |
| GenerateTabularReports | X | X | X |
| ImportConnections | X | X | |
| ImportFromXmi | X | X | X |
| ImportTabularReport | X | X | -> UI |
| MoveElementsToCalculatedParent | X | X | X |
| MoveElementsToPackage | X | X | X |
| MovePackagesToPackage | X | X | X |
| RenameItem | X | X | X |
| ScheduleCommand | X | X | X |
| SetDiagramProperty | X | X | X |
| SetProperty | X | X | X |
| SqlExportToCsv | X | X | X |
| Validate | X | X | X |

**Legend**
**-> UI** : Use the Labnaf AddIn
 i.e. the user interface

www.labnaf.one