

# LABNAF 6.0 PowerShell Reference Guide

Labnaf PowerShell commands ordered by name

## Contents

AutoConnectorsDelete .....	4
AutoConnectorsGenerate .....	4
BackupToFile / BackupToAccessFile .....	4
CalculateTaggedValues.....	5
ClonePackage .....	5
CreatePackage .....	5
DeleteGeneratedCharts.....	6
DeleteGeneratedDiagrams.....	6
ExportToXmi .....	6
GenerateCharts .....	6
GenerateDiagrams.....	7
GenerateDoc .....	7
GenerateHTML .....	7
GenerateImplicitData .....	7
GenerateTabularReports.....	8
ImportConnections.....	9
ImportFromXmi .....	10
ImportTabularReport.....	10
MoveElementsToCalculatedParent.....	11
MoveElementsToPackage .....	11
MovePackagesToPackage.....	12
Renameltem .....	12
ScheduleCommand.....	12
SetDiagramProperty .....	13
SqlExportToCsv .....	13
SetProperty.....	13
Validate.....	14



### Labnaf PowerShell Program

- “**LNPS**” is the name of the Labnaf PowerShell program. Full path is “**C:\Program Files (x86)\Labnaf\PowerShell\Inps.exe**”
- **Log files:** All commands yield a specific log file (in addition to standard output). Each log file name includes a time stamp, the name of the command, and additional context when applicable. The log directory path is specified by the “**OUTPUT\_DIR**” environment variable. By default, the provided Labnaf PowerShell scripts set the following value to this variable: “...Labnaf\_PowerShell\SCHEDULED\Output”. If the environment variable “**OUTPUT\_DIR**” is not set then log files are created in “...\Desktop\Labnaf\_Cmd\_Output\Log”.

### Related resources (latest versions)

- **Labnaf PowerShell User Guide:** [https://www.labnaf.one/EndUserMaterial/Labnaf\\_PowerShell/Labnaf%20PowerShell%20-%20User%20Guide.pdf](https://www.labnaf.one/EndUserMaterial/Labnaf_PowerShell/Labnaf%20PowerShell%20-%20User%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

## 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).

## 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).

## BackupToFile / BackupToAccessFile

**Description:** Backup a DBMS repository to a file-based Repository (.qea, .qeax, .feap, eadb, .eap, .eapx,).

**Usage:** LNPS BackupToFile [arguments]

**Arguments:**

- **SourceRepoPathName:** Path name of the source database repository (EAP file containing a connection string).
- DestPathName:** Path name of the destination file-based repository (.qea, .qeax, .feap, eadb, .eap, .eapx,).
- **DetailedLogFilePath:** Alternative path for the detailed log file (supersedes the default detailed log file path).

## CalculateTaggedValues

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

Periodical Value Calculation elements that have a name starting with "--" will be ignored.

**Usage:** LNPS CalculateTaggedValues [arguments]

### Arguments:

- **RepoPathName:** Repository path name (EAP file).
- **[ElementPrototypeName]:** A specific element stereotype for which tagged values must be calculated.
- **[TagName]:** The name of a specific tagged value that must be calculated.

## ClonePackage

**Description:** Create a clone of a source package and make it a child of a target parent package.

**Usage:** LNPS ClonePackage [arguments]

### Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **SourcePackageGUID:** The GUID of the package to be cloned.
- **TargetParentPackageGUID:** The GUID of the parent package where the source package must be cloned as a child.
- **[NewPackageName]:** An optional name for the new package. By default it is the same name as the source package to be cloned.

## CreatePackage

**Description:** Create a package, with an optional stereotype, and make it a child of a target parent package then send the GUID of the new package to the output.

**Usage:** LNPS CreatePackage [arguments]

### Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **TargetParentPackageGUID:** The GUID of the parent package where the source package must be cloned as a child.
- **NewPackageName:** The name of the new package to be created.
- **[Stereotype]:** An optional package stereotype to be set. If no stereotype must be set then use '-' instead.
- **[OutputStringWithOptionalPackageGUIDVariable]:** An optional string containing the variable {PackageGUID} which will be replaced by the GUID of the new package.
  - The resulting string will then be sent to the output.
  - This can be used to create a command that sets an environment variable to the new package GUID.

# DeleteGeneratedCharts

**Description:** Delete generated charts.

**Usage:** LNPS DeleteGeneratedCharts [arguments]

**Arguments:**

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

# DeleteGeneratedDiagrams

**Description:** Delete generated diagrams.

**Usage:** LNPS DeleteGeneratedDiagrams [arguments]

**Arguments:**

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

# ExportToXmi

**Description:** Export a package to an XMI file.

**Usage:** LNPS ExportToXmi [arguments]

**Arguments:**

- **SourceRepoPathName:** The path name of the source model repository.
- **SourcePackageGUID:** The GUID of the package to be cloned.
- **DestXMIFilePath:** The full path of the XMI file to be created.

# 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:** LNPS 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.

# 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.

# GenerateDoc

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

**Usage:** LNPS GenerateDoc [arguments]

## Arguments:

- **SourceRepoPathName:** Path name of the source model repository (EAP file).
- **OutputPath:** Path name of the document file to be generated. The file extension specified will determine the format of the generated document - for example, RTF, PDF
- **PackageGuid:** The GUID of the package or master document to run the report on.
- **TemplateName:** The document report template to use; if the PackageGUID has a stereotype of MasterDocument, the template is not required.

# 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).

# GenerateImplicitData

**Description:** Generate implicit data 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.

- risk information for entities and for elements that are, directly or indirectly, exchanging entities.  
See 'Implicit Data' on the Labnaf guidance web site to learn more about enabling and disabling options.

**Usage:** LNPS GenerateImplicitData [arguments]

**Arguments:**

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

## GenerateTabularReports

**Description:** Generate spreadsheets from a model repository based on configuration stored in that same repository.  
Tabular report templates that have a name starting with “—” will be ignored.

**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.
- **[ElementPrototype]:** A specific element prototype name, a stereotype or a friendly stereotype name (e.g. 'Application') for which all or specific tabular report templates must be applied.
- **[TabularReportTemplateName]:** The name of a specific tabular report template to be applied.

By default, the resulting file is in Excel format.

To generate a CSV, add a '.CSV' extension to the tabular report template name.

The name of the CSV file can define the CSV character encoding and column delimiter.

The syntax is: '...[CP={code page number or name};DELIM={COMMA|SEMICOLON|TAB|SPACE|a character code number}]...CSV

If no format is specified in the file name, then the default values are '[CP=utf-8;DELIM=COMMA]'

For further information, see ['Inbound/Outbound Content Formatting'](#) on the Guidance Web Site

# ImportConnections

**Description:** Import connections from a CSV or Excel file.

Identify the 'From' and 'To' elements to be connected using

- EITHER an ea\_guid,
- OR an optional element stereotype + a unique key that can be a property or tagged value. 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 connection if unique elements are found for the two sides of the connection.

In the input file, the first line can contain the following headers:

```
ConnectorStereotype,  
From#Stereotype,  
From#GUID,  
From#Name,  
From#MyUniqueKey,  
From#Alias,  
To#Stereotype,  
To#GUID,  
To#Name,  
To#MyUniqueKey,  
To#Alias
```

If a field name mapping file (CSV) is provided, the first line must contain the following headers:

```
Input_Column_Names, Target_Column_Names
```

**Usage:** LNPS ImportConnections [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 names

# ImportFromXmi

**Description:** Export a package to an XMI file.

**Usage:** LNPS ImportFromXmi [arguments]

## Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **SourceXMIFilePath:** The full path of the XMI file to be imported.
- **TargetPackageGUID:** The GUID of the package where the content must be imported.
- **Copy | Merge:** Indicates how the imported content that already exists in the repository will be imported
  - Copy = Create new GUIDs => If the imported content already exists in the repository then a new copy is created with new GUIDs.
  - Merge = Keep existing GUIDs => If the imported content already exists then it is merged based on the matching GUIDs found in the repository.

# 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 QEA and 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.

When the import is completed, a textual log file is created in a 'Log' folder.

The Log folder is created in the same folder that the one where the imported file is located.

The log file name is derived from the imported file name: { Import file name }\_Log.txt

**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.

The name of the CSV file can define the CSV character encoding and column delimiter.

The syntax is: '...[CP={code page number or name};DELIM={COMMA|SEMICOLON|TAB|SPACE|a character code number}]...CSV  
If no format is specified in the file name, then the default values are '[CP=utf-8;DELIM=COMMA]'

For further information, see ['Inbound/Outbound Content Formatting'](#) on the Guidance Web Site

- **ColumnMappingFile:** An optional CSV file containing the mapping between the input and output column names or '-' 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.

## MoveElementsToCalculatedParent

**Description:** Select any elements and make them child of their calculated parent.

Parent calculation is based on element stereotype, connector stereotype and connector direction.

**Usage:** LNPS MoveElementsToCalculatedParent [arguments]

### Arguments:

- **SourceRepoPathName:** Path name of the source model repository where the connectors must be changed.
- **ChildElementStereotype:** The stereotype of the child elements to be selected and to be moved to their selected calculated parent.
- **ParentElementStereotype:** The stereotype of the parent elements that need to contain the selected child elements OR '\*' if the parent stereotype is not a selection criteria.
- **ConnectorStereotype:** The stereotype of the connector between the parent and the child which is used to calculate the desired target parent child hierarchy in the model.
- **ChildToParent|ParentToChild:** The connector direction used to define which is the parent and which is the child
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL statement to select specific child elements to be moved.
- **[Verbose]:** Display the name of each element being moved along with the name of its new parent.

## MoveElementsToPackage

**Description:** Select any elements and move them to a package.

**Usage:** LNPS MoveElementsToPackage [arguments]

### Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **TargetPackageGUID:** The GUID of the package where the selected elements must be moved.
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL statement to select specific elements to be moved.

# MovePackagesToPackage

**Description:** Select any packages and move them to a package.

**Usage:** LNPS MovePackagesToPackage [arguments]

## Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **TargetPackageGUID:** The GUID of the package where the selected elements must be moved.
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL statement to select specific packages to be moved.

The result set must contain a column called 'Package\_ID'.

Example 1: SELECT \* FROM t\_package

Example 2: SELECT Package\_ID FROM t\_package

Example 3: SELECT PDATA1 AS Package\_ID FROM t\_object

WHERE object\_type='Package' AND stereotype='LNCAT\_Process'

- **[NewMovedPackageStereotype]:** The optional new stereotype of the packages after they have been moved.  
By default the stereotype is unchanged  
Provide an empty string ("") to remove the stereotype if any.

# RenameItem

**Description:** Rename a package, an element or a diagram based on a GUID

**Usage:** LNPS RenameItem [arguments]

## Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **GUID:** The GUID of the package, element or diagram that needs to be renamed.
- **NewItemName:** The new name of the package, element or diagram.

# 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.

# SetDiagramProperty

**Description:** Select diagrams and set one of their property value.

**Usage:** LNPS SetDiagramProperty [arguments]

## Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL statement which selects diagrams. The result set must contain a column called 'Diagram\_ID'.  
Example 1: `SELECT * FROM t_Diagram`  
Example 2: `SELECT Diagram_ID FROM t_Diagram`
- **DiagramPropertyName:** The name of a diagram property (FreezeVisible | Locked | Name | Stereotype) for which the value must be updated.
- **DiagramPropertyValue:** The new value of the diagram property. Boolean values can be 'True' or 'False'.

# SqlExportToCsv

**Description:** Run SQL statement in a model repository and save the result into a CSV.

**Usage:** LNPS SqlExportToCsv [arguments]

## Arguments:

- **RepoPathName:** Repository path name.
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL Select statement.
- **OutputCsvFileName:** The path name of the CSV file that will contain the result set.  
The name of the CSV file can define the CSV character encoding and column delimiter.  
The syntax is: '...[CP={code page number or name};DELIM={COMMA|SEMICOLON|TAB|SPACE|a character code number}]...CSV'  
If no format is specified in the file name, then the default values are '[CP=utf-8;DELIM=COMMA]'  
For further information, see 'Inbound/Outbound Content Formatting' on the Guidance Web Site
- **[NoHeader]:** Do not generate headers in first row.

# SetProperty

**Description:** Select any elements or packages and set some property value.

**Usage:** LNPS SetProperty [arguments]

## Arguments:

- **SourceRepoPathName:** The path name of the source model repository.
- **PropertyName:** Name of the property that will get a new value. The property/tag will be created if it does not exist.
- **PropertyValue:** New value for the property.
- **SqlSelectStatementFileName:** The path name of a file that contains some SQL statement to select specific elements or packages that needs a property value to be set.

# 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.