

Living Systems® Process Suite

Quickstart

Living Systems Process Suite Documentation

3.1
Tue Jan 12 2021

Whitestein Technologies AG | Hinterbergstrasse 20 | CH-6330 Cham
Tel +41 44-256-5000 | Fax +41 44-256-5001 | <http://www.whitestein.com>

Copyright © 2007-2021 Whitestein Technologies AG
All rights reserved.

Copyright © 2007-2021 Whitestein Technologies AG.

This document is part of the Living Systems® Process Suite product, and its use is governed by the corresponding license agreement. All rights reserved.

Whitestein Technologies, Living Systems, and the corresponding logos are registered trademarks of Whitestein Technologies AG. Java and all Java-based trademarks are trademarks of Oracle and/or its affiliates. Other company, product, or service names may be trademarks or service marks of their respective holders.

Contents

1	Quickstart	1
1.1	What is LSPS	1
1.2	Launching Process Design Suite	2
1.3	Creating the Hello World Model	2
1.4	Running Hello World	6
1.5	What Just Happened	7
1.6	What to Do Next	8

Chapter 1

Quickstart

This guide is the place to start if you are new to the product and need to get familiar with its architecture, environment, and related documentation. It introduces you to Living Systems Process Suite (LSPS) and leads you through the hello-world model scenario.

1.1 What is LSPS

LSPS is a suite of tools that allow you to **develop a flexible process-driven company application** that will distribute work to your employees, communicate with external systems and users based on BPMN or GO-BPMN processes and that **in a very short time frame**. You will be able to manage the processes and any related data, monitor key performance indicators of processes and activities, integrate with social media and much more.

When you install **Process Design Suite** (PDS) as instructed in the [Installation Guide](#), you installed the IDE for development your models.

PDS comes with the locally running application server called **PDS Embedded Server**. It is generated when you first request to run it. It is stored in the `.LSPSEmbedded` directory of your workspace.

PDS Embedded Server has the LSPS Application deployed: this is a JEE application that allows you to deploy, run, and manage GO-BPMN models. The heart of the application is the **LSPS Server**, which stores and runs your models, manages model instance execution, their update, etc.

PDS provides also the thick client for management of the LSPS Server in the form of the Management perspective.

Summary:

- **Process Design Suite** is the IDE where you will design your models here, connect to LSPS Application, and manage all resources.
- **LSPS Application** is a JEE application that works with your models: it manages the uploaded models, runs the models and manages the related resources, such as, user rights and roles. It has the following web applications:
 - **Application User Interface** intended for end-users to interact with the LSPS Server.
 - **Management Console** intended for administrator to manage the LSPS Server.
- **PDS Embedded Server** is an out-of-the application server with the LSPS Application deployed.

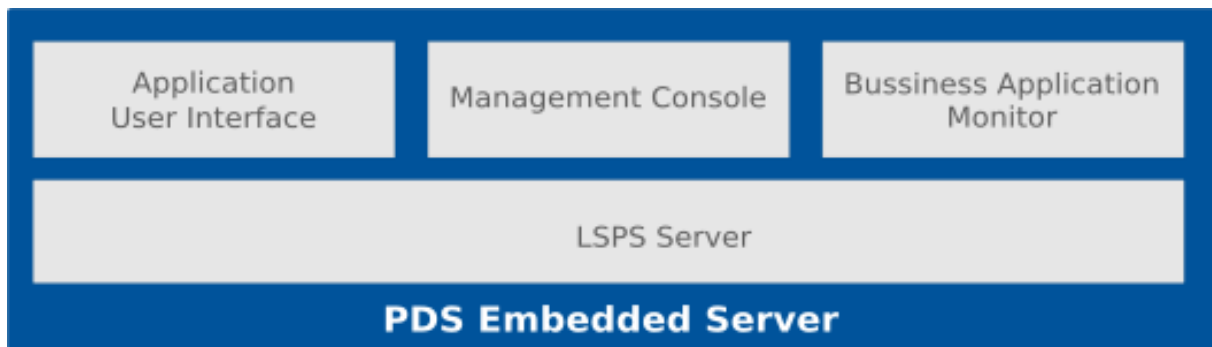


Figure 1.1 PDS Embedded Server that you start and control from PDS

Note: Apart from the *PDS Embedded Server*, LSPS comes with yet another embedded Server called the *SDK Embedded Server*. It is provided by the SDK extension of PDS as part of the Enterprise Edition: its LSPS Application EAR along with the Application User Interface sources are part of your workspace (you generate the server from PDS). While the LSPS Server part of the application cannot be modified, you extend it with custom items and EJBs to implement custom business logic and adapt the thin client *Application User Interface*. Further instructions and details are available in the [development documentation](#).

1.2 Launching Process Design Suite

To launch the *Process Design Suite*, do the following:

1. Go to `$LSPS_HOME`.
2. Run the `lsps-design` binary for your platform.
3. In the Workspace Launcher dialog box, choose a workspace folder and click OK.

If the chosen workspace folder does not exist, it will be created: it is simply a folder, where the resources of the session are located.

The Welcome page appears. Click the Modeling Perspective icon or close the Welcome page.

1.3 Creating the Hello World Model

To be able to design valid processes, you need to create your processes as well as other resources in a set structure: First you create *projects*: they exist only in your workspace and are intended for organization purposes. A GO-BPMN project contains *modules*, which allow you to organize the resources still further, but also provide the visibility and executability features, similarly to Java packages: a module contains definitions of processes as well as other resources, such as, variables, organization hierarchy of involved people, localization units, etc.

Modules are what you upload to the server. If a module is executable, it can be used to create a model instance. The server then executes the model instance, based on the definitions in the module and its module imports, which are simply imported modules.

To create the project structure with the module and process, do the following:

1. In PDS, create a project:

- (a) On the main menu, go to **File > New > GO-BPMN Project**.
- (b) In the *New GO-BPMN Project* dialog box, enter the *HelloWorld* project name.

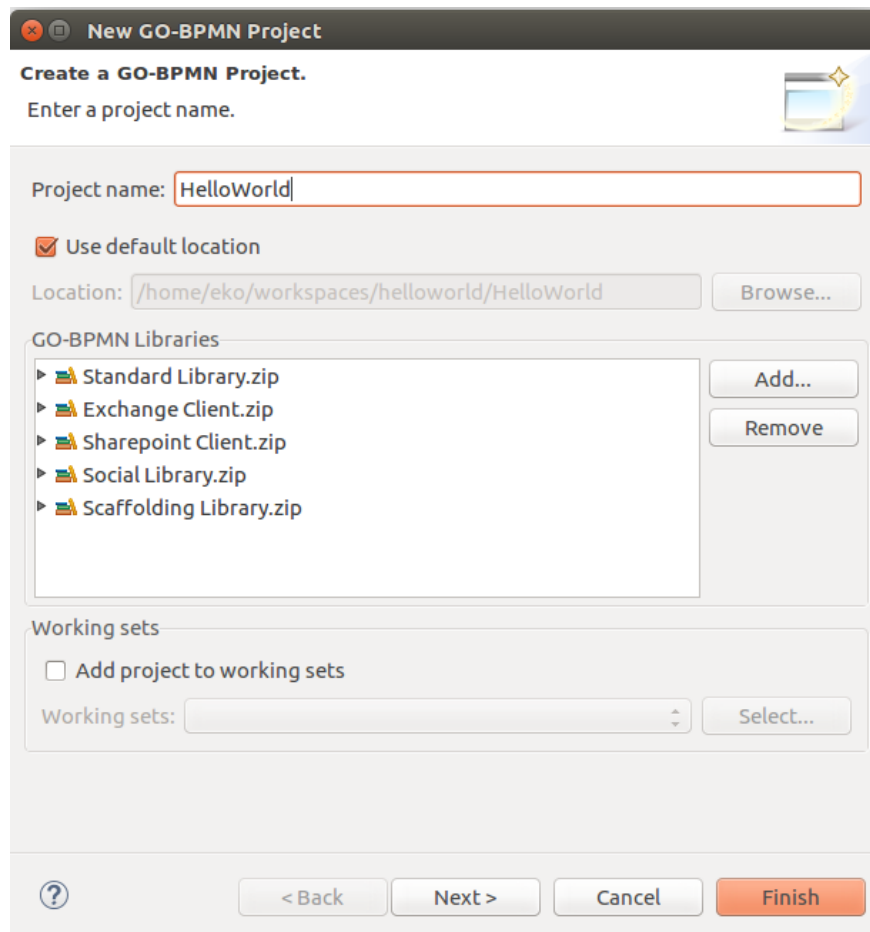


Figure 1.2 New GO-BPMN Project dialog

- (c) Click Finish.

2. In the project, create the HelloWorld module:

- (a) In the GO-BPMN Explorer view, right-click the HelloWorld project and go to **New > GO-BPMN Module**.
- (b) In the New GO-BPMN Module dialog box, enter the HelloWorld module name.

Note: Leave the *executable module* check box selected: an executable module can become a *Model* instance (the flag somewhat resembles the logic of the *main* method in Java).

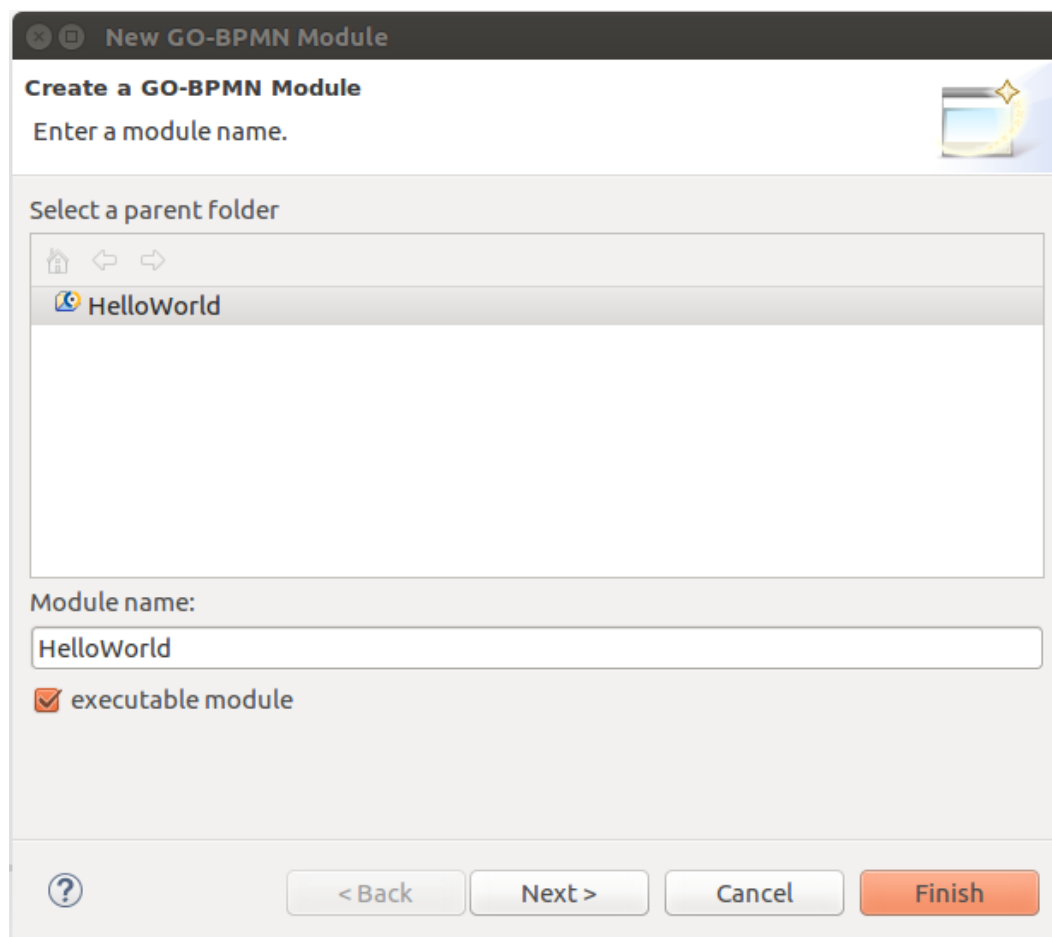


Figure 1.3 New GO-BPMN Module dialog

You have created a HelloWorld project with a HelloWorld module. Note that the module contains the Standard Library imports.

(c) Create a process definition in the HelloWorld module:

- i. In the GO-BPMN Explorer, right-click the HelloWorld module and go to **New > Process Definition**.
- ii. In the *New Process Definition* dialog box, enter the `helloWorldProcess` as the process name and click **Finish**. Make sure the Type is set to *BPMN-based process*.

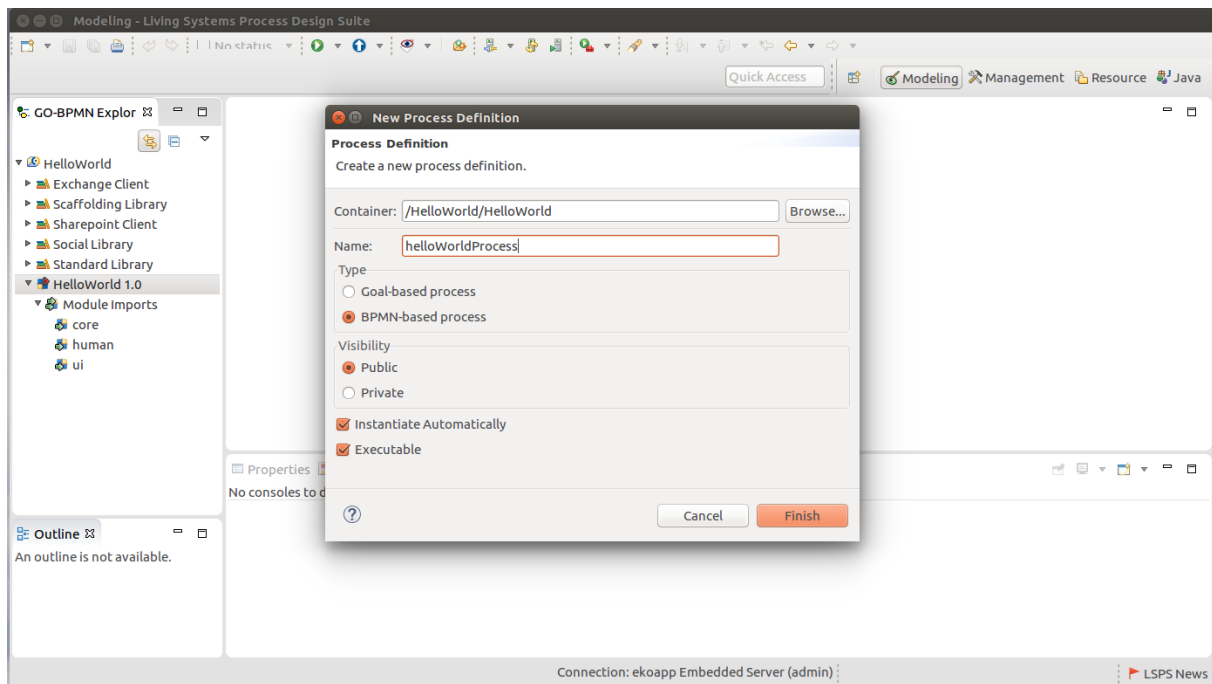



Figure 1.4 Creating a process definition file

3. Now you can design the process content:

- (a) In GO-BPMN Explorer, double-click the process definition.
- (b) In the Modeler editor on the right, create a process with a None Start Event and a Simple End Event connected with the Flow:
 - i. In the palette on the right, click the None Start Event.
 - ii. On the canvas, click where you want to position the None Start Event.
 - iii. Select the None Start Event and click-and-drag the quick linker icon  next to it to the position where you want to place the Simple End Event. In the context menu, select Simple End Event.

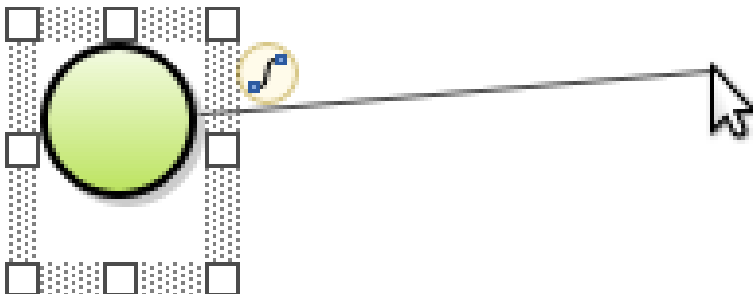


Figure 1.5 Using quick linker

- (c) Select the Normal Flow element between the None Start Event and Simple End Event: in its Properties view, click the Assignment tab.

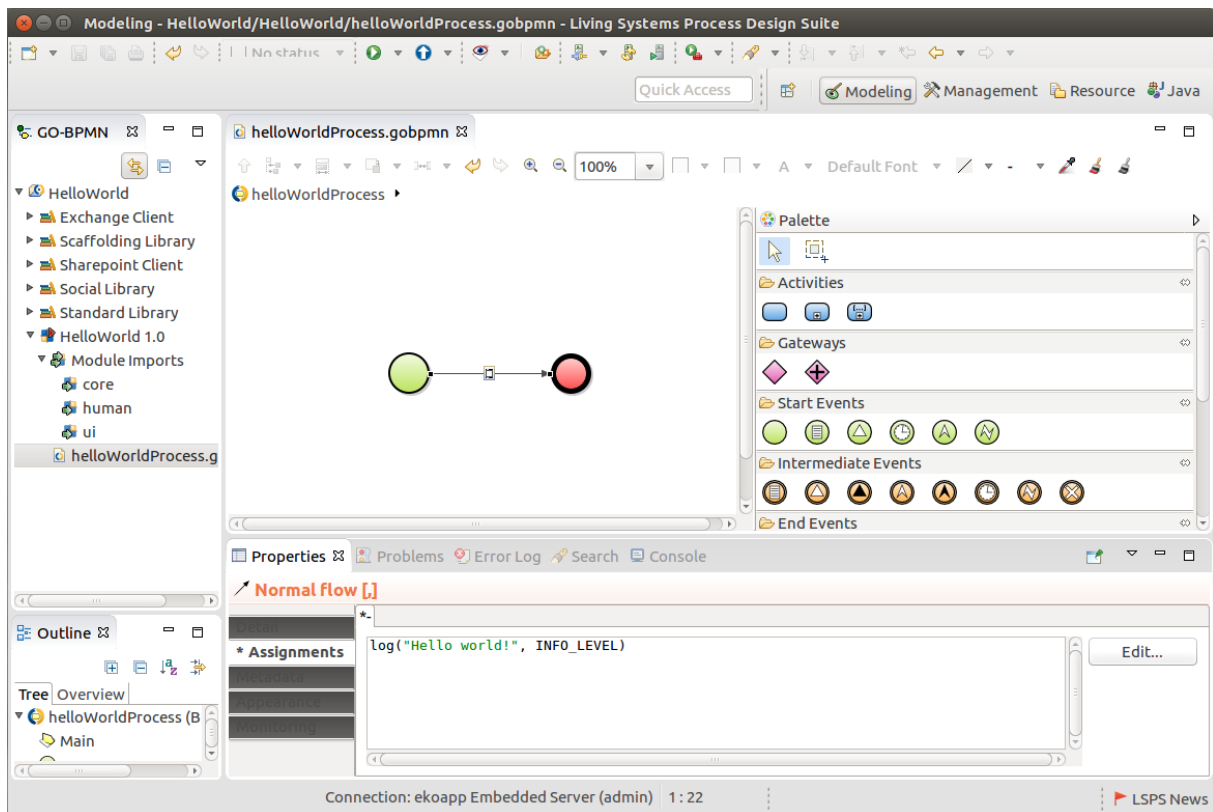


Figure 1.6 Hello World with the assignment expression

(d) On the Assignment tab, enter the following expression:

```
log("Hello World!", INFO_LEVEL)
```

`log` is a call to the `log()` function of the Standard Library that logs the specified message with its level to the database log of the application: the Logs can be viewed in the Management perspective.

If you have followed the steps as described above, the module should be valid: in case there is something wrong, check the Problems view. If the view contains a problem entry, you will need to solve it. Check the respective instructions above again and remedy any mistakes.

If there are no problems in the view, you can upload the model to the PDS Embedded Server and run it.

1.4 Running Hello World

To run an instance of the HelloWorld model on the PDS Embedded Server, do the following:

1. Click  to run the PDS Embedded Server.

PDS creates the `.LSPSEmbedded` folder in your workspace and generates the resources for the server with the LSPS Server, and runs the server with the LSPS Server deployed.

2. Upload and instantiate the model: right-click the HelloWorld module and go to **Run As > Model**.

This uploads the module and its module imports to the LSPS Server and creates its Model instance.

3. You can check that the model instance was created in the Management perspective:

- (a) Switch to the Management perspective (on the main menu, go to Window > Perspective > Open Perspective > Management).
- (b) In the Module Management view, click the Refresh (🔄) button: The view will display the HelloWorld module and the Standard Library modules.
- (c) In the Model Instances view, click the Refresh (🔄) button: The view will contain the instance of the HelloWorld model.
- (d) In the Logs view, click the Refresh (🔄) button: The view will contain the log message of the HelloWorld model.

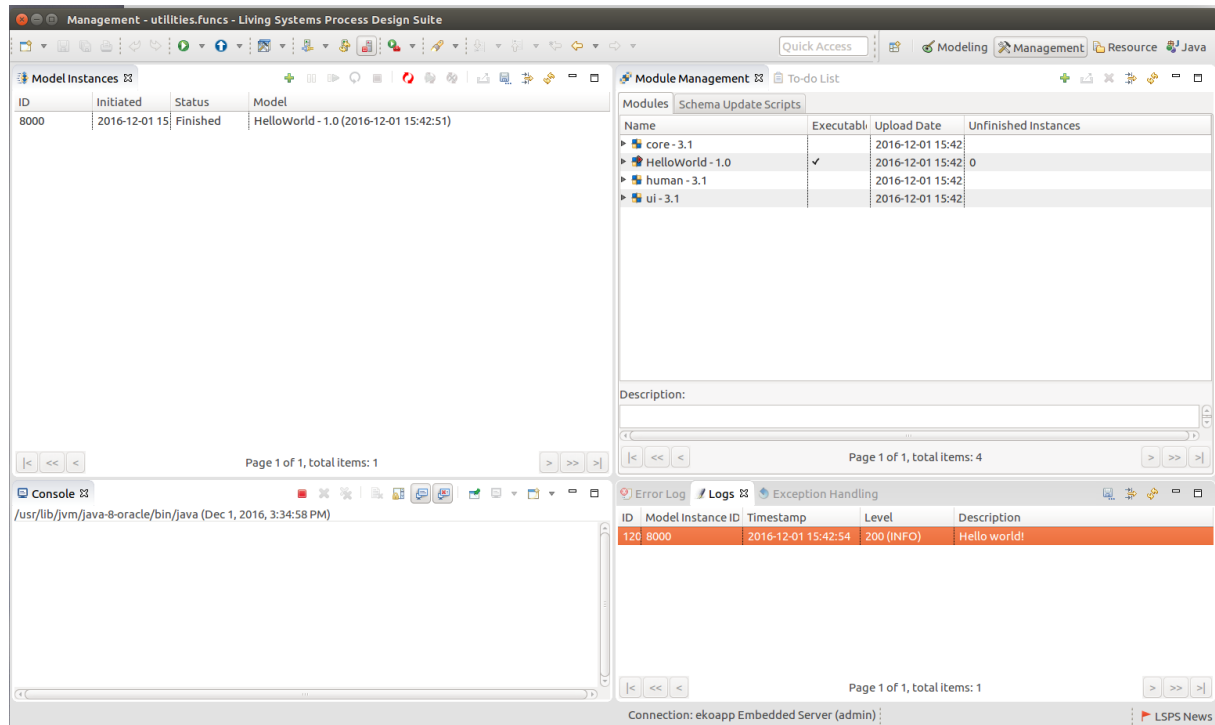


Figure 1.7 Hello World model instance entry in the Management perspective

4. Optionally, check the system data in the underlying H2 database. Set the database URL to `jdbc:h2:tcp://localhost/h2/h2;MVCC=TRUE;LOCK_TIMEOUT=60000` and connect to it with the user `lsp`s and password `lsp`s.

1.5 What Just Happened

When clicked **Run As > Model** on your executable module, the following happened:

1. The module with all its module imports, in this case only modules of the Standard Library, were uploaded to the Module repository of the LSPS Server.
2. The LSPS Server created a model instance based on your model.
3. The model instance checked for Start Events in processes it could trigger: it found the None Start Event in your `helloWorldProcess`. Hence it created an instance of the process.
4. The None Start Event produced a token, which indicates where the execution is currently at.
5. The LSPS Server moved the token to the End Event, which consumed the token.
6. Since no more tokens were present in the process instance, the process instance finished.
7. Since no other processes were running in the model instance, the model instance finished.

1.6 What to Do Next

Now that you have a rough overview of how LSPS works, you can refer to one of the following:

- [Academy guide](#) to get familiar with LSPS in a training-like style
 - [Modeling chapters in the PDS guide](#) to gain deeper knowledge on PDS features
 - [Modeling language](#) to learn about process elements
-