WHITESTEIN
Technologies

Living Systems® Process Suite

Quickstart

Living Systems Process Suite Documentation

3.3
Mon Nov 1 2021

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

Contents

Chapter 1

Quickstart

This guide is the place to start if you are new to Living Systems Process Suite (LSPS). It will familiarize you with the product's architecture, its environments, and related documentation, and create your first process model.

1.1 What is LSPS

LSPS is a suite of tools that allows you to develop a **flexible company application** that will distribute work to your employees, communicate with external systems and users, and do so in a very short time.

You will be able to manage processes and any related data, monitor key performance indicators of processes and activities, integrate with social media platforms, etc.

1.2 Tools

First, you need to `install Process Design Suite (PDS)`: PDS is the environment for the development of your application and process models.

You will set up and use a local application server, called **PDS Embedded Server**, with *LSPS Application* deployed: LSPS Application is a JEE application that allows you to deploy, run, and manage GO-BPMN models. The heart of the application is **LSPS Server**; it stores and runs your models, manages their execution and updates, as well as, system users, authentication, access rights, etc.

From PDS, you can manage LSPS Server from of a dedicated perspective, called the Management perspective.

This guide deals only with process models, and the extending and modification of *LSPS Application* is out of its scope. Refer to the `developer documentation` for the information.

Summary:

- **Process Design Suite** is the environment (IDE) where you will design your business models and perform all management tasks.
- **PDS Embedded Server** is an application server with LSPS Application already deployed. It is intended for testing and demonstration purposes.
- **LSPS Application** is an application that allow you to interact with and manage your business models.

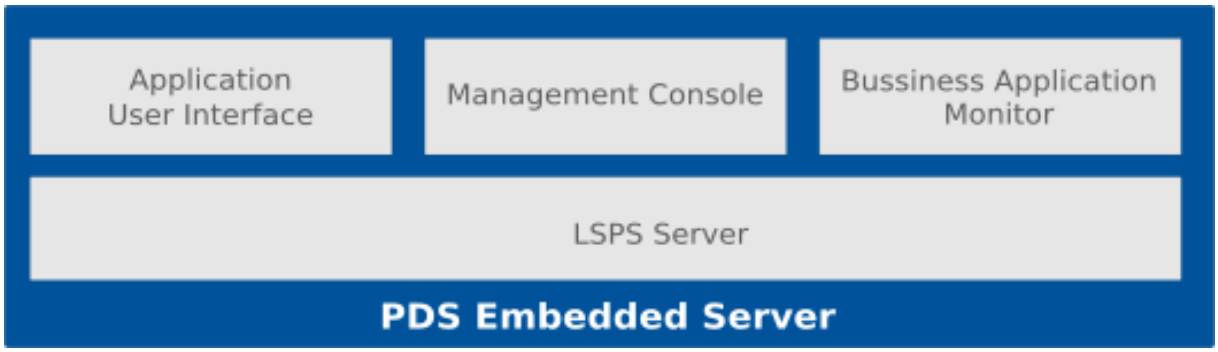

Figure 1.1 PDS Embedded Server that you start and control from PDS

1.3 Hello World Model

1. Go to the folder where you installed LSPS, `$LSPS_HOME`.
2. Run `lsps-design_64`.
3. In the *Workspace Launcher* dialog box, enter a path to the folder where you want to store everything you will create.
4. Click OK.
5. On the *Welcome* page, click the *Modeling Perspective* icon.

1.4 Creating the Hello World Model

First you will create *GO-BPMN projects*: projects 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 creates the model instance and creates in its context instances of the executable module and any imported modules.

To create the project structure with a module and a 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.

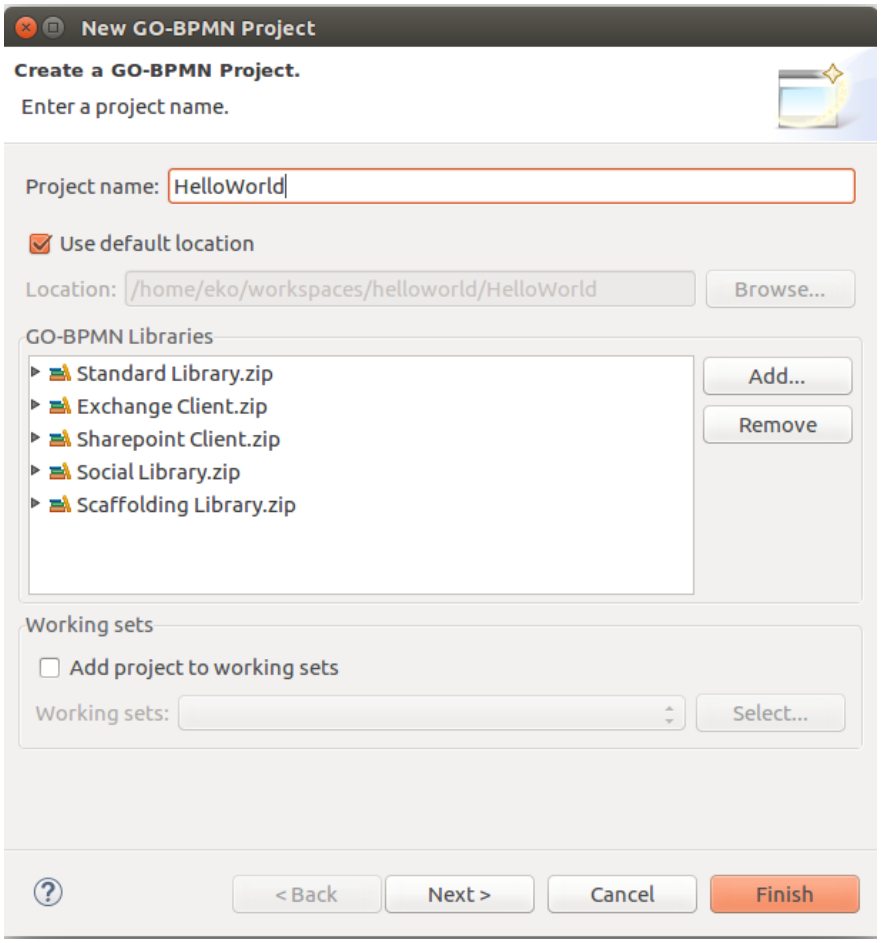

Figure 1.2 New GO-BPMN Project dialog

(c) Click Finish.

2. In the project, create a *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 is considered a model and can become a `model instance` (the flag somewhat resembles the logic of the *main* method in Java).

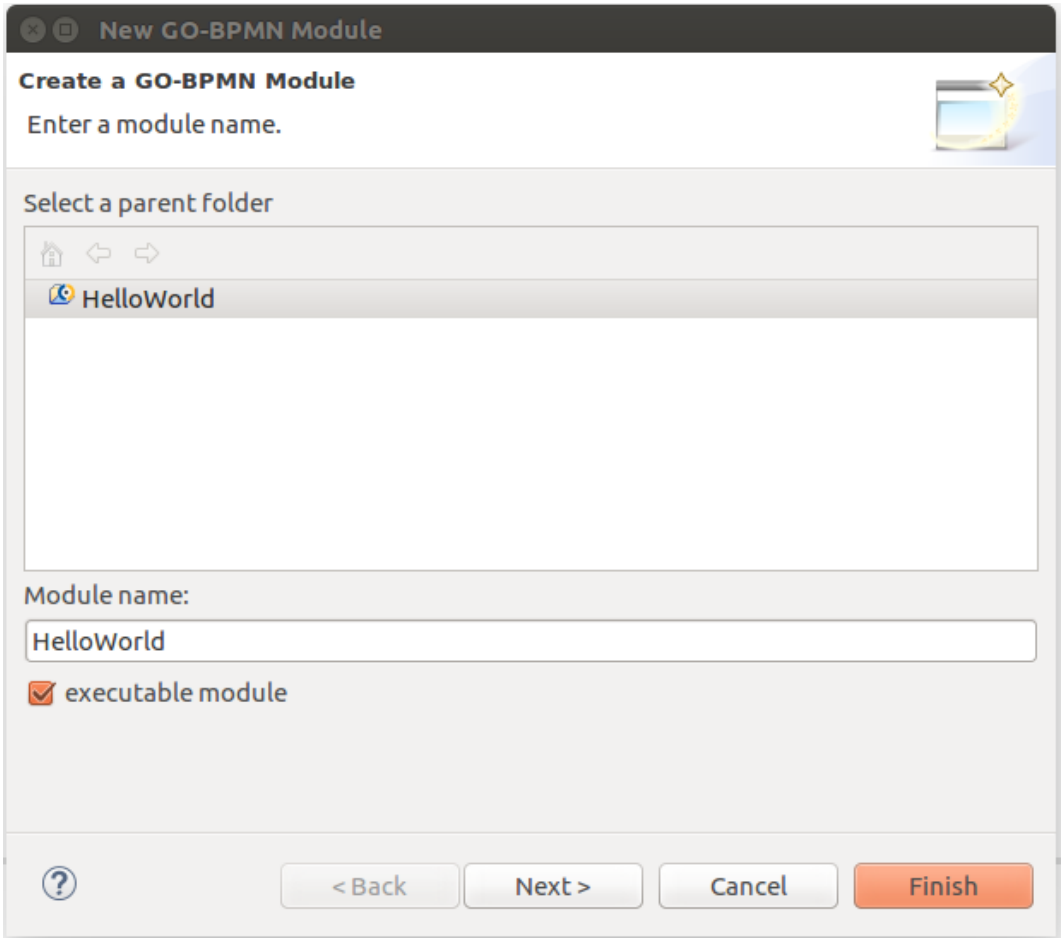

Figure 1.3 New GO-BPMN Module dialog

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*.

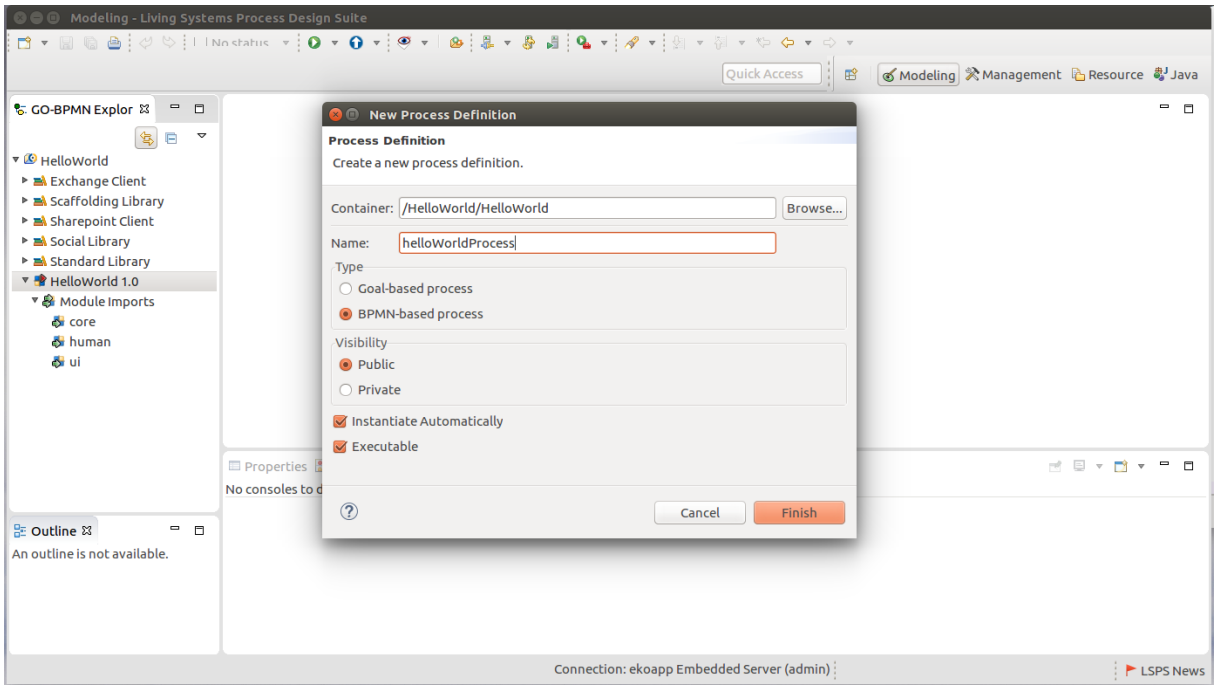

Figure 1.4 Creating a process definition file

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.

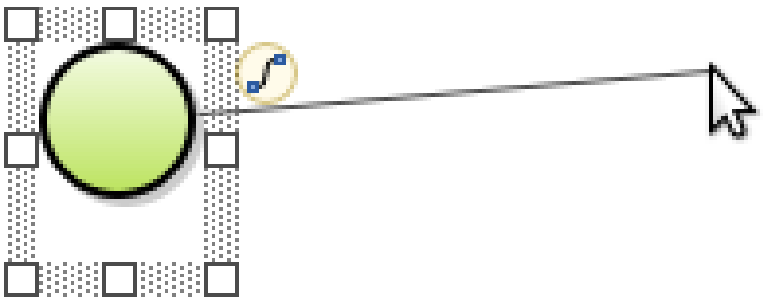

Figure 1.5 Using quick linker

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

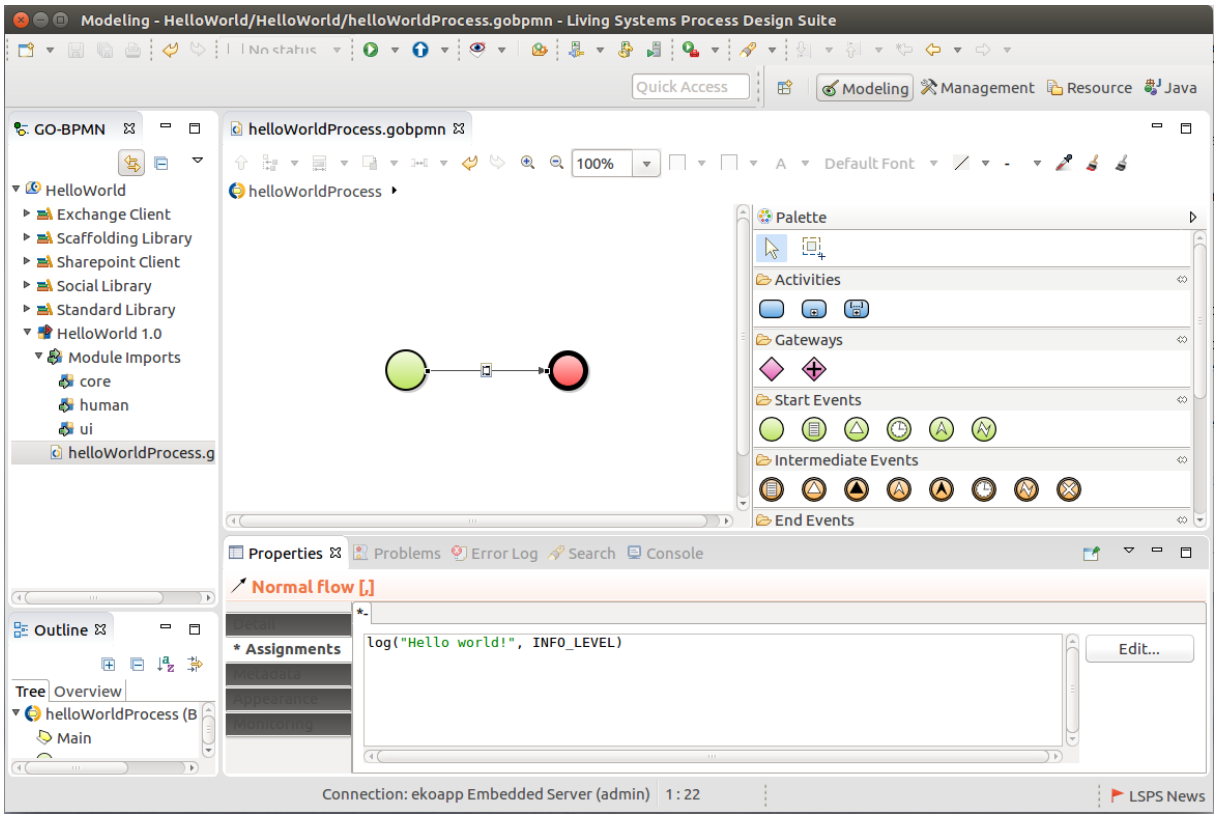

Figure 1.6 Hello World with the assignment expression

(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.

The module should be valid: Open the *Problems* view. If the view contains a problem entry, you need to resolve 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 PDS Embedded Server and run it.

1.4.1 Running Hello World

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

1. Click  to run 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.

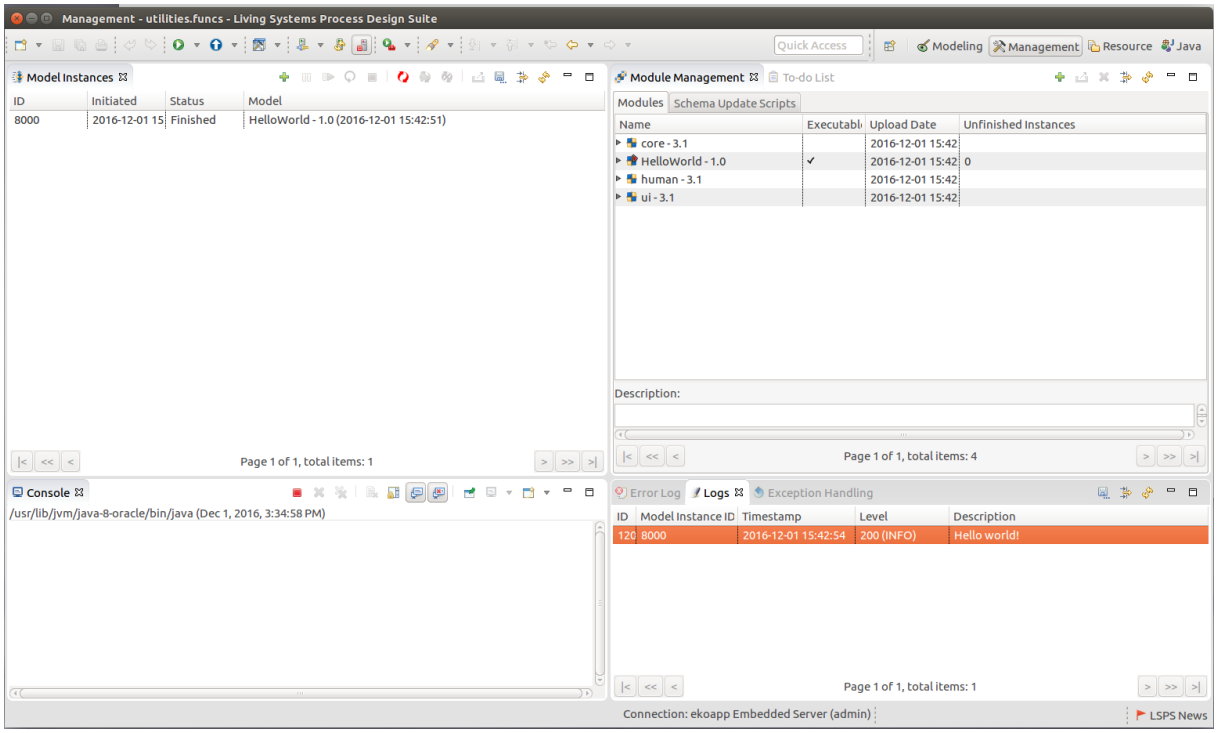

Figure 1.7 Hello World model instance entry in the Management perspective

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 `lsps` and password `lsps`.

1.4.2 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, was 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 and therefore created an instance of the process.
4. The None Start Event of the process was triggered: it produced a token, which indicates what is currently executed.
5. The token left the None Start Event via the flow to the End Event.
6. The End Event consumed the token.
7. Since no more tokens were present, the process instance finished.
8. Since no other processes were running, the model instance finished.

Chapter 2

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
- `Developers' Guide` to learn how to customize the LSPS application
- `Server Deployment Guide` to learn how to set up the server environment and deploy the LSPS Application
- `Modeling language` to learn about process elements