medexter
clinical decision support

Medexter Healthcare

HOW TO COMPILE, TEST, AND DEPLOY MLMS WITH THE ARDENSUITE

SIRS Notification as an Example

Datum: 11.05.2017

Version: 1.0

Medexter Healthcare GmbH, Borschkegasse 7/5, A-1090 Vienna, Austria, Telephone +43-1-968 03 24, Facsimile +43-1-968 09 22 www.medexter.com, Handelsgericht Wien: FN 225564m, UID: ATU54901101, IBAN: AT602011128014988900, BIC: GIBAATWW

SUMMARY 2

DOCUMENT INFORMATION 2

TARGET AUDIENCE 2

IMPRESSUM 2

INTRODUCTION 3

REQUIREMENTS 3

THE ARDENSUITE IDE AND ARDENSUITE SERVER 3

REST CLIENT 3

FILES 4

COMPILING AN MLM IN THE ARDENSUITE IDE 4

TESTING AN MLM IN THE ARDENSUITE IDE 6

DEPLOYING COMPILED MLMS ON THE ARDENSUITE SERVER 8

Summary

The aim of this *how-to* instruction manual is to show how to compile and test Arden Syntax Medical Logic Modules (MLMs) in the ARDENSUITE IDE and—subsequently—how to deploy them on an ARDENSUITE Server.

Document Information

Target Audience

This instruction manual was created for ARDENSUITE users and developers interested in developing MLMs and deploying them on an ARDENSUITE Server.

Impressum

Media owners, editors, publishers:

Medexter Healthcare GmbH, Borschkegasse 7/5, A-1090 Vienna, Austria

Telephone: +43-1-968 03 24, Fax: +43-1-968 09 22, Internet: www.medexter.com

Email: office@medexter.com

CEO: Klaus-Peter Adlassnig, PhD, MSc

Editorial, project management, coordination:

Klaus-Peter Adlassnig, PhD, MSc

Figures: © Medexter Healthcare GmbH

Use: This document contains the intellectual property of Medexter Healthcare GmbH. The use for educational purposes without license and usage fees is permitted. Other kinds of use and reproduction are subject to the approval of the media owner.

Vienna, May 2017

Version: 1.0

Download at www.medexter.com

Introduction

In this *how-to* instruction manual, we will guide you step-by-step in the loading, compilation, testing and deployment of Arden Syntax Medical Logic Modules (MLMs). Throughout this *how-to*, we provide MLM use case examples containing notification rules for the systemic inflammatory response syndrome (SIRS). Clinically, the rules for SIRS notifications are as follows:

SIRS Notification

ALERT if ≥ 2 Criteria

Temperature > 38°C (100.4°F) or < 36°C (96.8°F)
and/or

Heart rate > 90 beats per minute
and/or

Respiratory rate > 20 breaths per minute or arterial carbon dioxide tension (PaCO2) < 32 mm Hg
and/or

White blood cell count (>12,000/µL or < 4,000/µL or >10% immature [band] forms)

Following these provisions, MLMs were constructed that implement these notification rules and generate alerts when patient data match these criteria.

Requirements

For optimal use of this *how-to*, please make sure the following software is installed on your computer or accessible from your location:

- The ARDENSUITE IDE and ARDENSUITE Server (for deployment and remote invocation)
- A REST client (for testing deployed MLMs)

The ARDENSUITE IDE and ARDENSUITE Server

In case you do not have access to the ARDENSUITE IDE or the ARDENSUITE Server yet, please contact us at support@medexter.com. A 30-day trial version of the ARDENSUITE can also be downloaded here.

REST Client

For instructional and testing purposes, REST calls to MLMs and their return values are illustrated using a web browser. In this document, we recommend using Google Chrome, together with the Postman

browser plugin for the visualization of REST communication.

Files

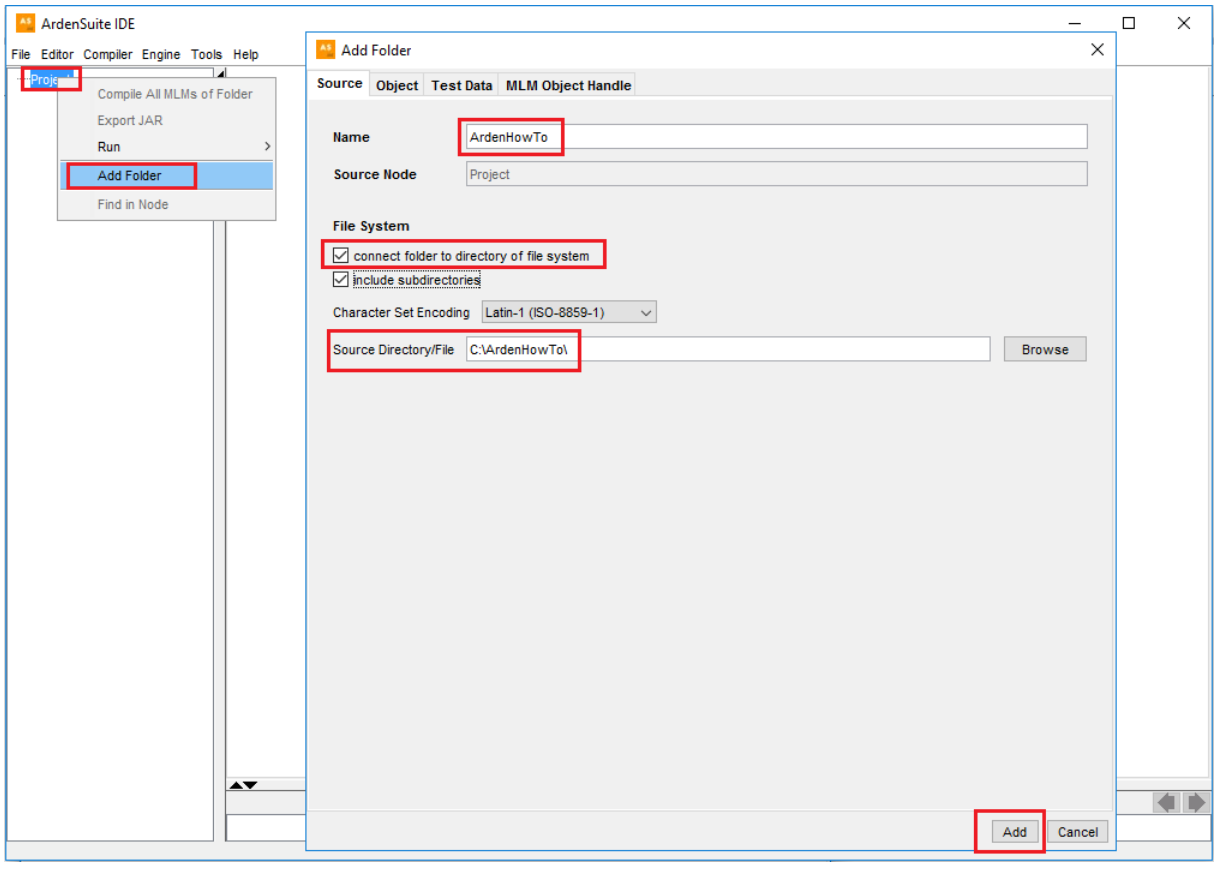
This *how-to* is accompanied by two MLM files (extension `.mlm`):

- `SIRS-Notification1`: This MLM evaluates the four SIRS notification criteria and returns an alert if two or more criteria are met. To illustrate error handling by the ARDENSUITE IDE, we inserted several errors in the Arden Syntax code.
- `SIRS-Notification2`: This MLM is a corrected version of `SIRS-Notification1`. This MLM contains no errors and can be tested and deployed.

NOTE: *The MLM files can be opened using any standard text editor or viewer, but in order to compile and upload the MLMs, the ARDENSUITE IDE and ARDENSUITE Server are required.*

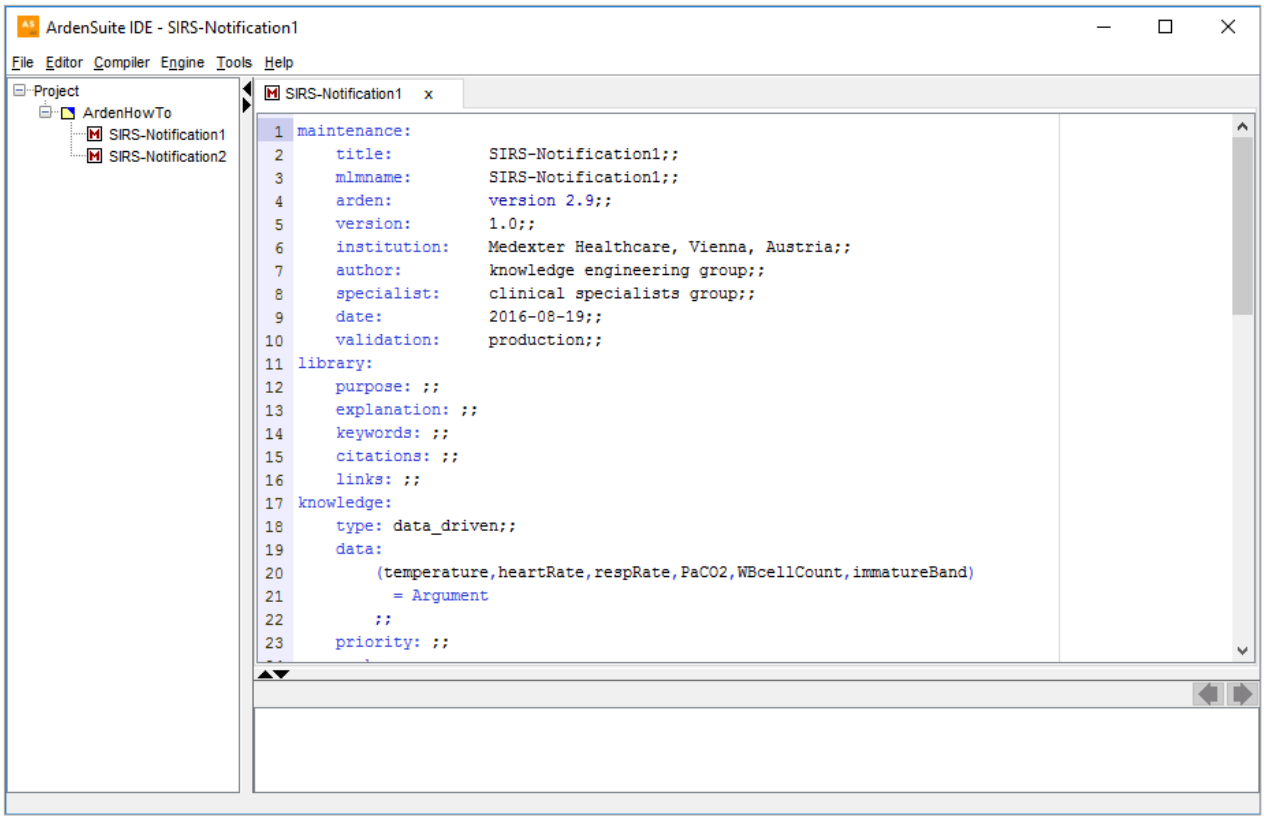
Compiling an MLM in the ARDENSUITE IDE

Before you can create or load MLMs, you have to create a project folder first. To create a project folder, right click on the `Project` tree element on the left panel of the screen, and select `Add Folder` (see figure below).



In the `Add Folder` dialog, enter a name for the project (e.g., `ArdenHowTo`). If a folder with MLMs already exists on your computer—as is the case for this *how-to*—, select `connect folder to directory of the file system` under `File System` (e.g., `C:\ArdenHowTo`). Optionally, you can also select `include subdirectories`, if your MLMs are spread over several subdirectories. Under `Source Directory/File`, enter the directory path; alternatively, you can select the path by clicking on the `Browse` button. Finally, click on the `Add` button to create the project folder.

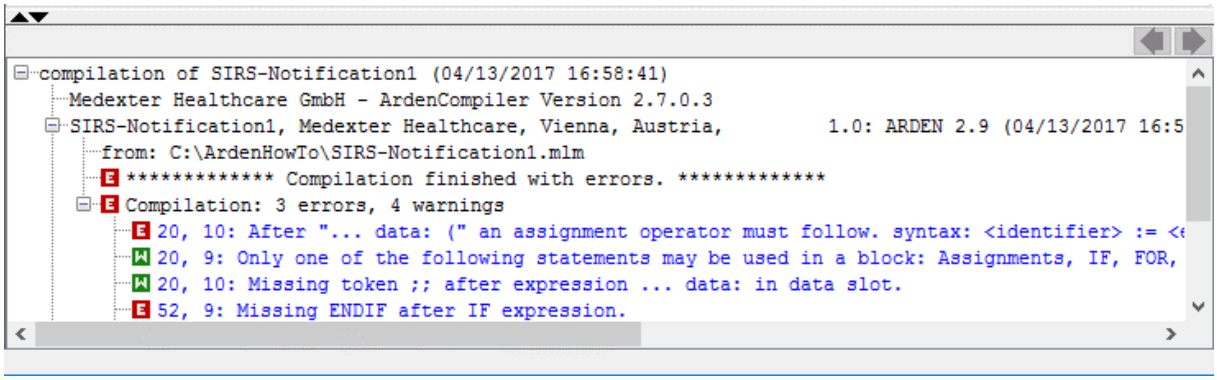
After creating a new project folder, it appears under the `Project` tree element on the left panel of the screen. If you connected the folder to a directory of the file system, all MLMs in that folder will be displayed as well. If you double-click on an MLM, the contents of this MLM will be displayed in the main screen on the right (see figure below).



There are several ways to compile an MLM:

- Right-click on the MLM in the tree structure on the left pane, and click on the menu item `Compile All MLMs of File`
- Right-click on the code in the main screen on the right, and click on the menu item `Compile current MLM`
- Press `Ctrl+R` on the keyboard

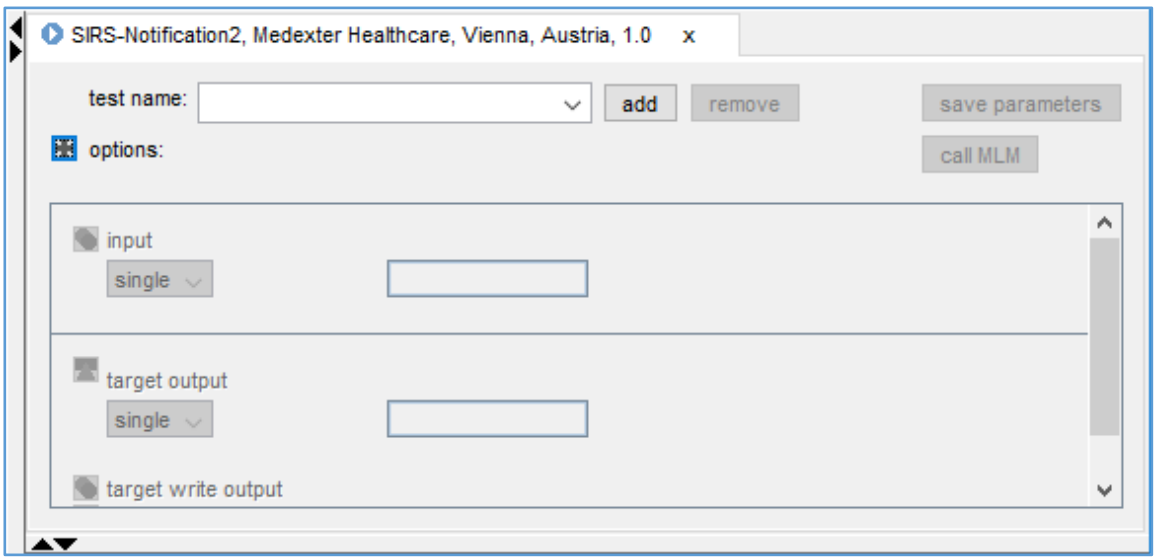
When compiling an MLM, compilation feedback will appear in the bottom pane of the screen. When you compile SIRS-Notification1, you will notice that several error messages appear (see figure below).



Error messages are tagged with a red icon with the letter 'E', whereas warnings are tagged with green icons with the letter 'W'. The MLM SIRS-Notification1 contained two program errors, for which three error messages were generated. Upon compiling a correct MLM, i.e., SIRS-Notification2, compilation feedback will show no errors. Furthermore, two extra entries appear in the tree structure on the left pane: a metadata file with a red round icon and a white 'M', and the compiled MLM with a blue round icon and white triangle.

Testing an MLM in the ARDENSUITE IDE

To test a compiled MLM, you need to double-click on it in the tree structure on the left pane; the compiled MLM is tagged with a blue round icon and white triangle. Upon double-clicking, a test dialog appears (see figure below).



Before testing can commence, first you need to enter a name for the test scenario in the `test name` field, and press the `add` button. After that, inputs and outputs can be entered in their respective dialog parts. Both fields can have three different types:

- `single`: Input or output consists of a single value, e.g., a string or a number
- `flat list`: Input or output consists of a list of values. Upon entering values in a field, a new field appears behind that, until you entered all the values
- `compound list`: Input or output consists of a list of lists, whereby each list is of the type `flat list`

The MLM `SIRS-Notification2` expects a flat list with 6 values as input, as shown in the following code snippet:

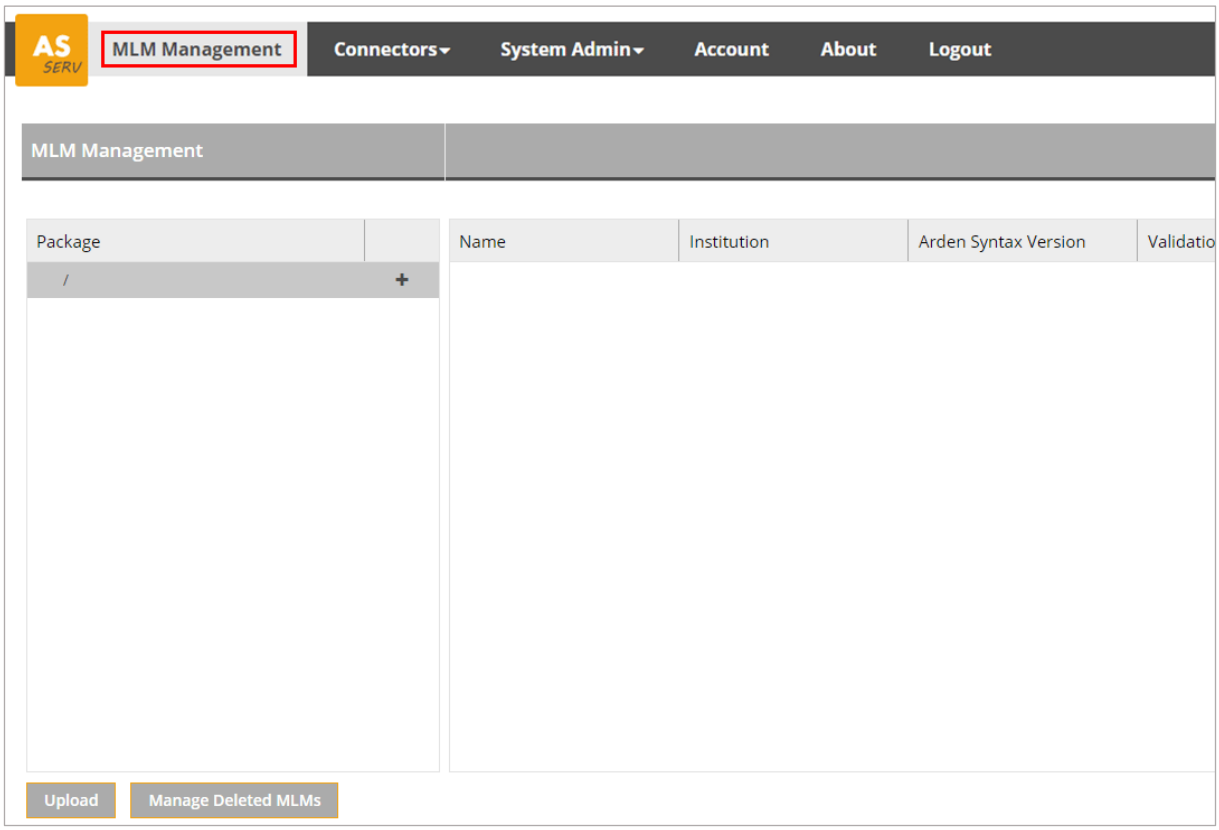
```
(temperature,heartRate,respRate,PaCO2,WBcellCount,immatureBand) = Argument
```

Please select the type `flat list` for input, and insert values for six list elements. As an example, consider the `SIRSTest` scenario in the figure below.

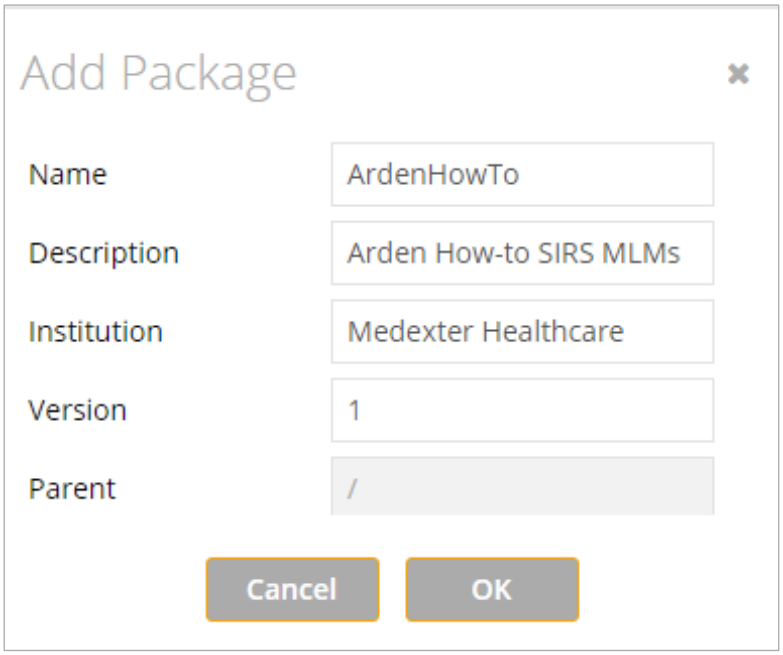
Furthermore, based on the inputs provided, the expected output would be a string containing an alert for SIRS, thus as output the `single` type should be selected, with the value `"Alert for SIRS"`. Next, press `save parameters` to save the test configuration, and then press `call MLM` to start the test. If you entered the data from the `SIRSTest` scenario, you will notice a test feedback stating `"current result equals target result"` (see figure below).

Deploying Compiled MLMs on the ARDENSUITE Server

In order to call MLMs remotely, they need to be deployed on the ARDENSUITE Server. To deploy MLMs, login to the ARDENSUITE Server and select `MLM Management` in the top menu (see figure below).



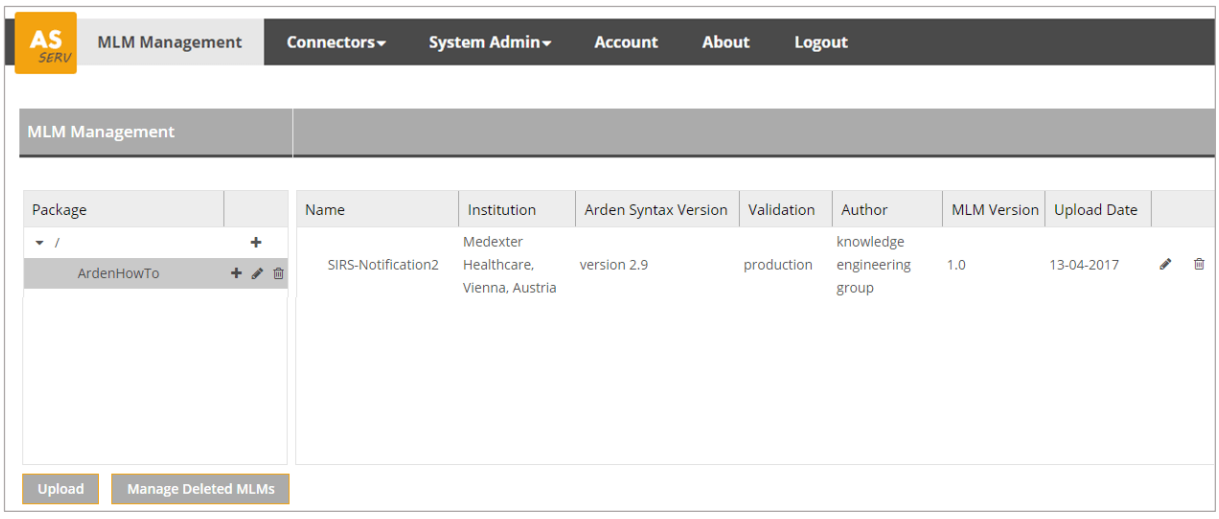
To upload a compiled MLM to the root directory, simply select `/` in the left pane under `Package` and press `Upload`. Alternatively, you can create a separate package for your MLMs by pressing `+`. Please provide a name, description, institution, and version for the package (see figure below).



To add the package to the ARDENSUITE Server, press the `Add package` button. A package will appear on the left pane of the screen. To upload MLMs to this package instead of to the root directory, select the package (e.g., `ArdenHowTo`) and press the `Upload` button. You are then asked to select the compiled MLM you want to upload (file extension `.gz`); here, only one MLM can be uploaded at a time.

Note: *It is possible to upload several MLMs at once. First, pack all the compiled MLMs you want to upload in a .zip container. Then upload the .zip file to the ARDENSUITE Server. The server will then automatically unpack the container and deploy the MLMs.*

After deployment of the compiled MLM `SIRS-Notification2` in the package `ArdenHowTo`, the screen looks as follows:



To call this MLM, start the Postman browser plugin and construct the REST call. For more information on how to call MLMs using REST, we refer to the corresponding *how-to* document, which can be found here. After starting Postman, use the following URL for the MLM REST call:

```
http://localhost:8080/REST/CALLMLM?mlmName=SIRS-
Notification2&mlmInstitution=Medexter Healthcare, Vienna, Austria
```

In case of our test scenario, the JSON data that needs to be supplied in the `Body` segment of the REST call is:

```
{
  "type": "list",
  "primaryTime": null,
  "applicability": 1,
  "values": [
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
      "value": 39
    },
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
      "value": 89
    },
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
      "value": 23
    },
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
```

```
      "value": 33
    },
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
      "value": 3965
    },
    {
      "type": "number",
      "primaryTime": null,
      "applicability": 1,
      "value": 16
    }
  ]
}
```


If everything worked out as it should and no errors occurred, the server returns an alert for SIRS. The alert looks like this:


```
{
  "type": "string",
  "primaryTime": null,
  "applicability": 1,
  "value": "Alert for SIRS"
}
```