

# **ARDENSUITE Integrated Development and Test Environment (IDE)**

**IDE-Version:** 2.6  
**Datum:** August 2015

# Table of Contents

1	Introduction .....	3
2	Setup .....	3
2.1	System Requirements .....	3
2.2	Installation .....	4
2.3	License Activation .....	7
2.3.1	Online activation .....	8
2.3.2	Offline activation .....	8
3	Understanding the ARDENSUITE IDE .....	9
3.1	Arden Syntax – Basic Concepts .....	9
3.1.1	Medical Logic Modules (MLMs).....	9
3.1.2	Calling MLMs.....	9
3.1.3	Data In- and Output .....	10
3.1.4	ARDENSUITE IDE – Components.....	10
3.1.5	Authoring Component .....	11
3.1.6	Compiler Component .....	12
3.1.7	Engine Component .....	12
4	Using the ARDENSUITE IDE – Five Minutes Tutorial .....	14
4.1	Start.....	14
4.2	Adding MLMs.....	14
4.3	Compiling MLMs .....	16

# 1 Introduction

We consider it the overall goal of our ARDENSUITE software systems to offer tools for the representation and processing of medical knowledge based on a standardized syntax. The ARDENSUITE software components developed by [Medexter Healthcare](#) allow to write, compile, test, manage, and execute Arden Syntax medical logic modules (MLMs). These MLMs can be written and processed according to the rules of the medical knowledge representation language [Arden Syntax](#) versions 2.1, 2.5, 2.6, 2.7, 2.8, 2.9 and 2.10. As of version 2.9, the Arden Syntax is augmented by fuzzy concepts, whereas version 2.10's new feature ArdenML—an XML representation of these MLMs—allows the Arden Syntax to be compatible with all other HL7 standards based on XML (HL7 version 3, VmR, and others). All Arden Syntax versions are [Health Level Seven International](#) (HL7) standards and are continuously revised and extended by HL7's Arden Syntax Work Group. Arden Syntax is also approved by the [American National Standards Institute](#) (ANSI).

Our ARDENSUITE software products include the ARDENSUITE compiler for compiling MLMs, the ARDENSUITE engine for executing MLMs and the ARDENSUITE server, which is built around the ARDENSUITE engine and enables service-oriented access for arbitrary client applications. Additionally, the ARDENSUITE server allows easy management of MLMs and compiled knowledge files. Our database connector serves as an add-on for the ARDENSUITE server and is responsible for accessing external databases from within a medical logic module. The ARDENSUITE integrated development and test environment (IDE) is a simple and intuitive tool for writing and compiling MLMs in Arden Syntax. The compiled modules can then be tested in the integrated test environment. The ARDENSUITE components were developed so that they can directly be integrated into healthcare host applications such as hospital, medical, clinical, and laboratory information systems, intensive care patient data management systems, medical practice software systems, telemedical applications, web-based electronic health records, web-based consumer and provider health portals, mHealth applications, and others.

The manual at hand concentrates on describing the ARDENSUITE Integrated Development and Test Environment (IDE). It includes a guide through the setup procedure, an introduction to Arden Syntax, a description of the ARDENSUITE IDE components as well as a Five Minutes Tutorial. Detailed information on how to use the ARDENSUITE IDE is also available in the integrated help menu.

## 2 Setup

This section guides through the installation of the ARDENSUITE IDE using the provided install package as well as its activation.

### 2.1 System Requirements

- Windows XP/Vista/Windows 7/Windows 8 (for use with other operating systems, please contact [office@medexter.com](mailto:office@medexter.com))
- A current Java 6 runtime environment (download: <http://java.sun.com>)

## 2.2 Installation

The ARDENSUITE IDE package contains—apart from this document—the installer “installArdenIDE.jar”. Double-click to start the installation and open the installer screen:

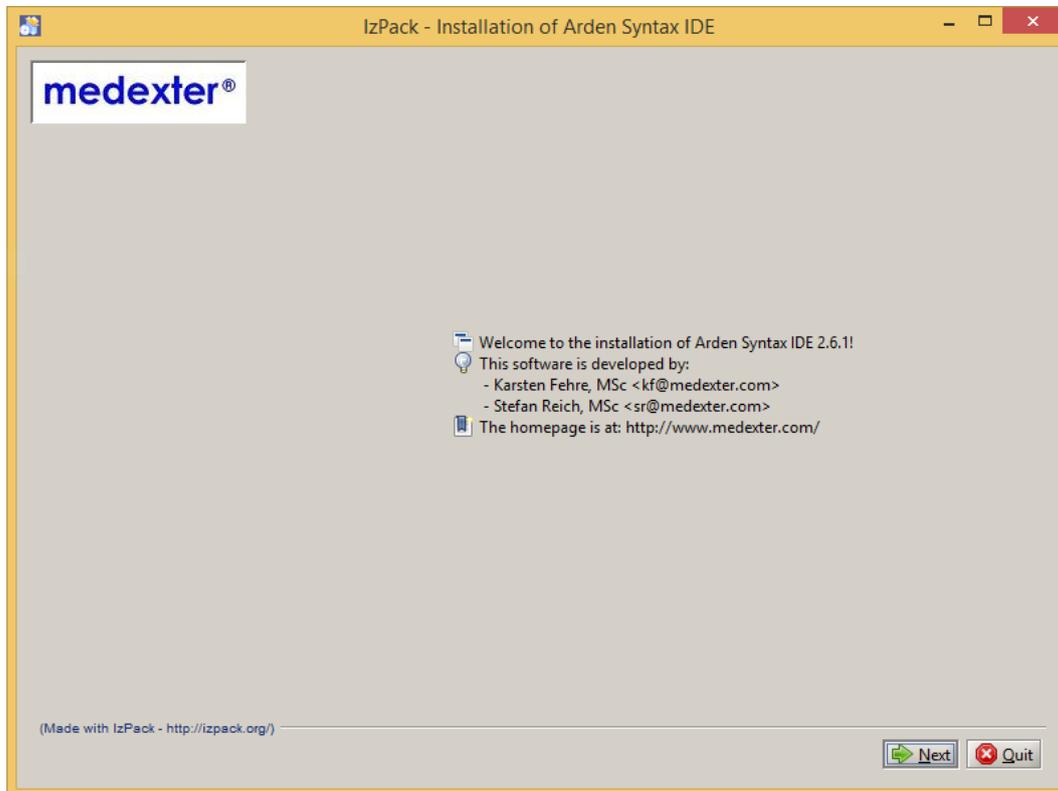


Figure 1: First installation window

Click **Next** to proceed.

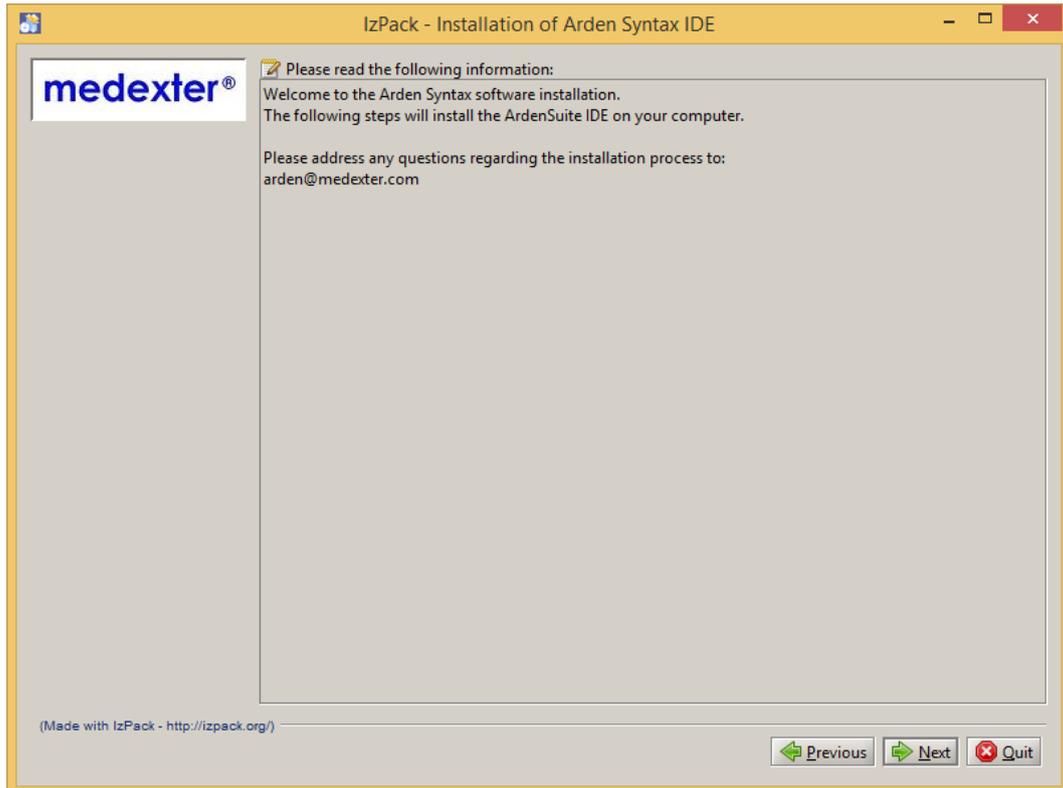


Figure 2: Installation information

Click **Next** to specify the installation destination.

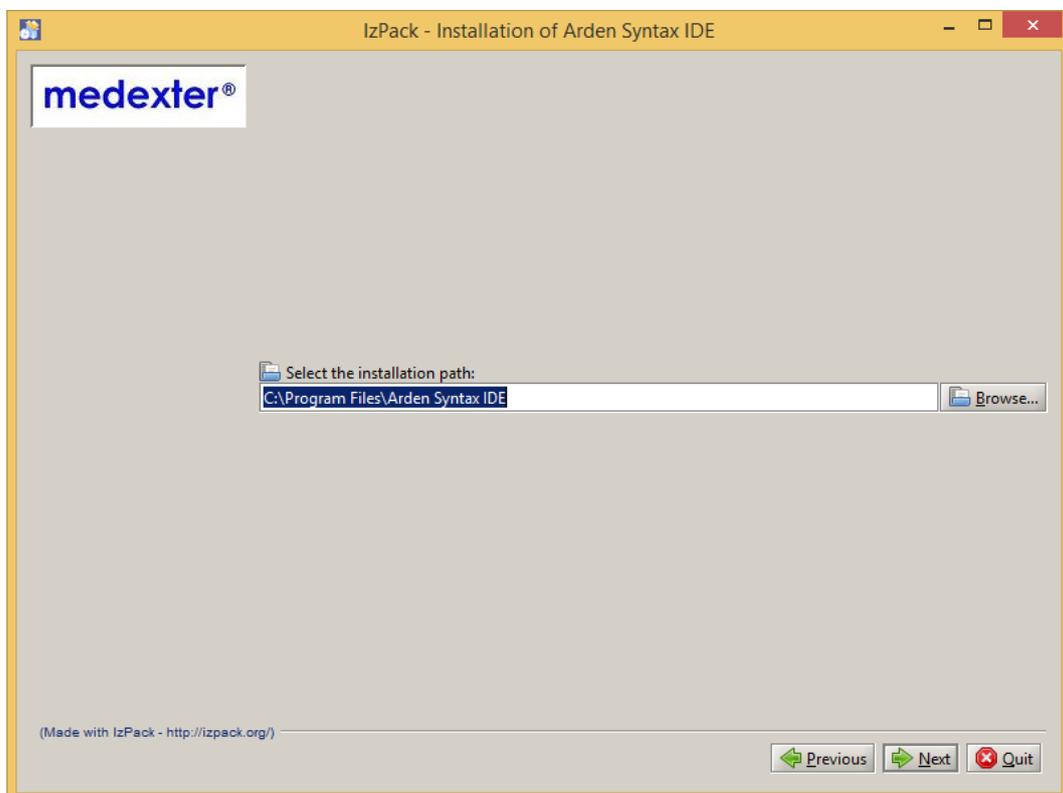


Figure 3: Installation path

By clicking **Next**, the installer creates the directory structure and installs the IDE.

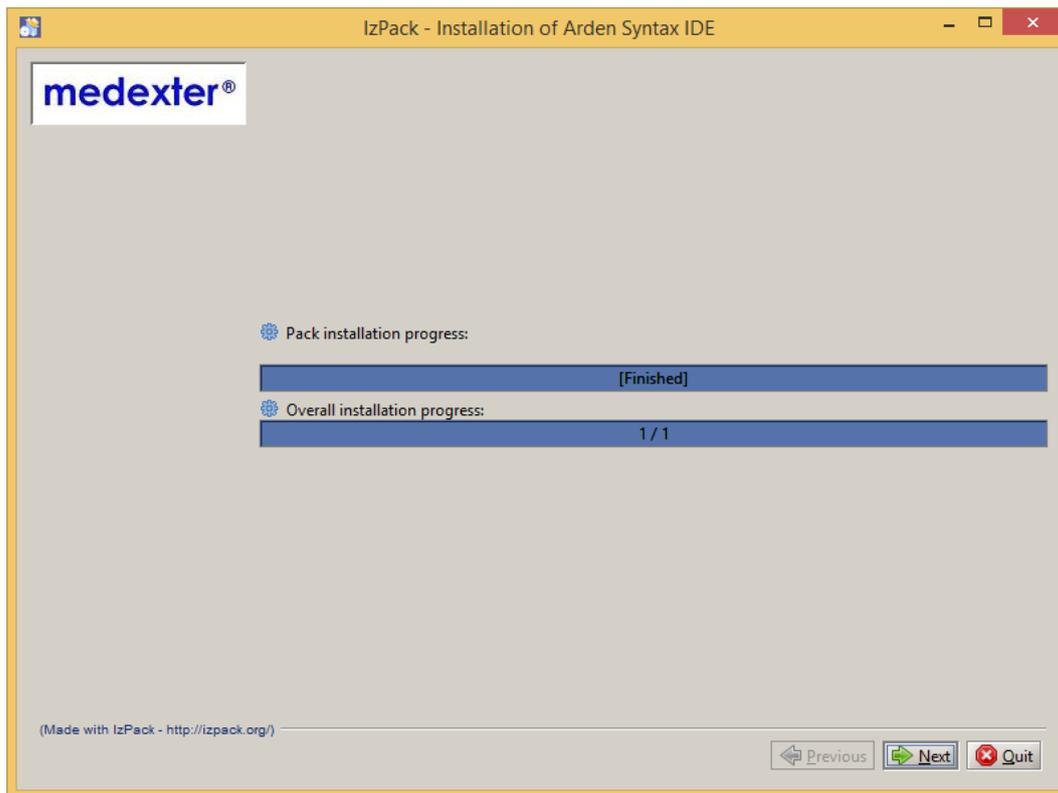


Figure 4: Successful installation

After a click on **Next**, the installer sets the necessary permissions to the IDE files.

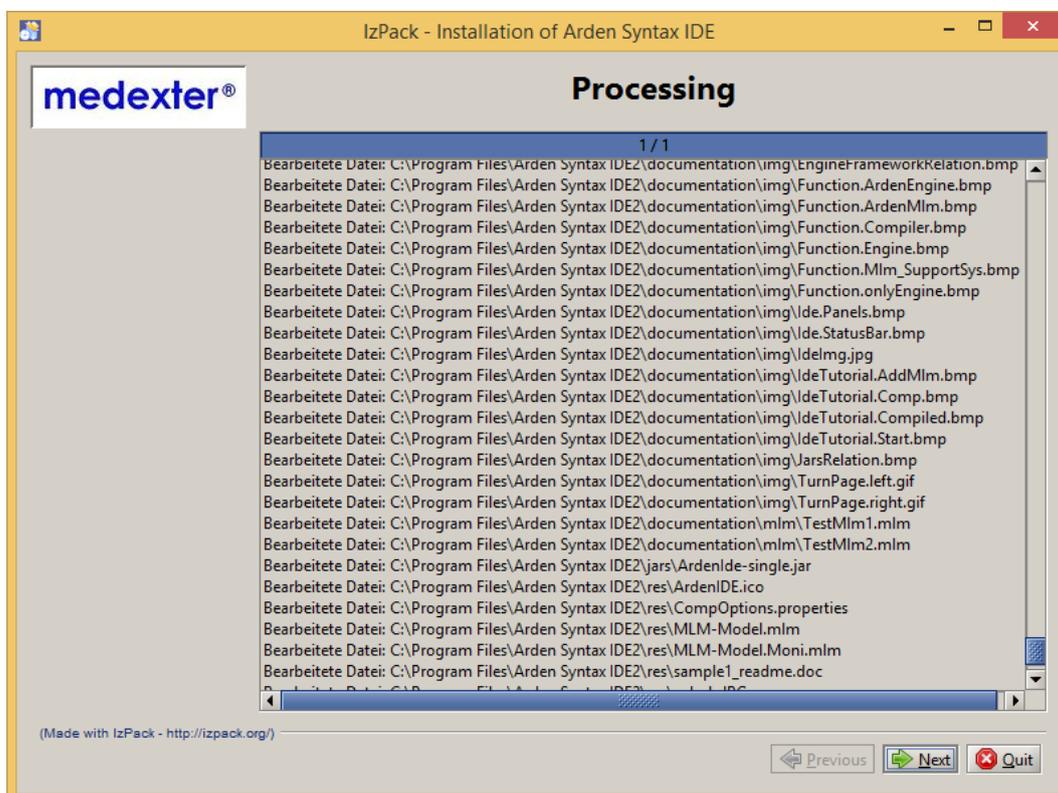


Figure 5: Setting file permissions

Please proceed to the next screen by clicking **Next** to create the desktop and start menu entries. Click **Done** on the last screen to close the installer. Your ARDENSUITE IDE is now installed.

## 2.3 License Activation

After the successful installation of your ARDENSUITE IDE, you have to activate it. By **starting** your IDE via the **start menu**, you will see the license information window. If you already had an active license of our IDE on your computer, you can access this window by clicking **Help->Activate Arden IDE** in your IDE.

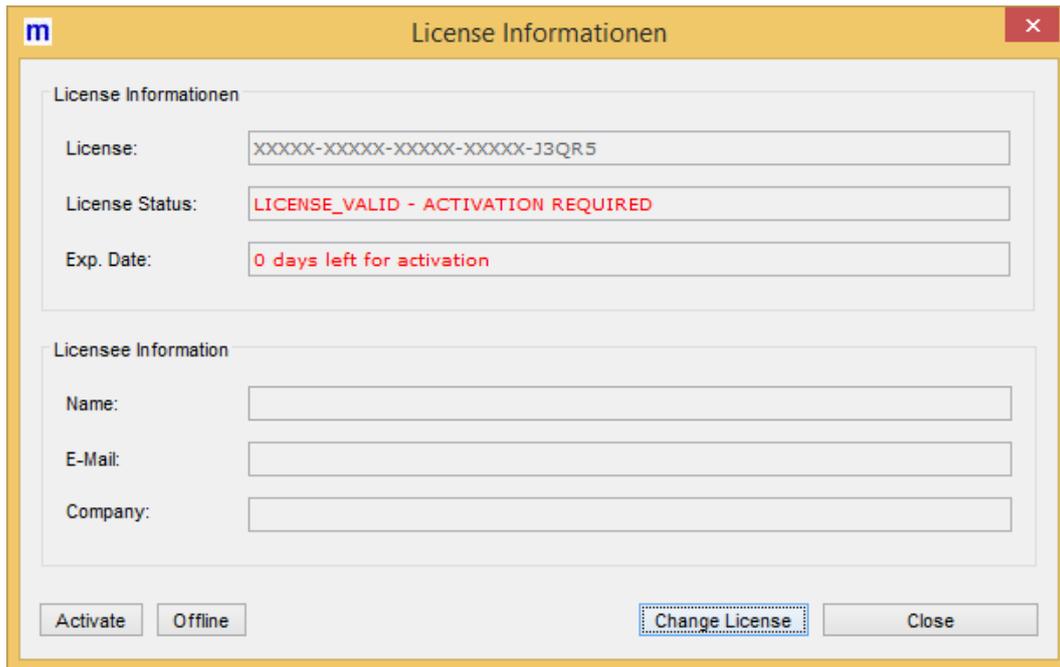


Figure 6: License information window

By clicking on **Change License** you will be asked for a license key or if you want to use a trial license:

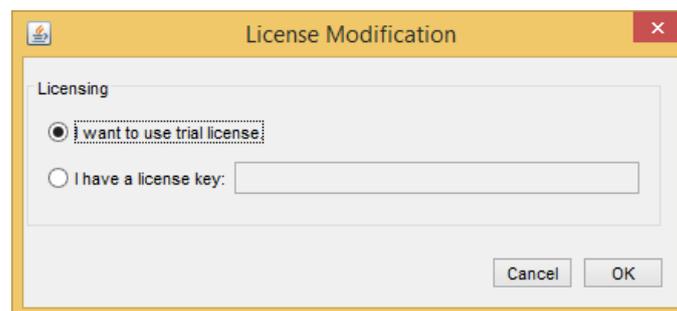


Figure 7: License modification window

Medexter Healthcare offers you a free trial license which lasts 30 days from the activation date. If you want to use the trial, select "I want to use a trial license" and click **OK**.

If you already have purchased a full license, please fill in the license key Medexter Healthcare sent you and press **OK**.

After that, you have to activate the license by using one of the two offered methods.

### 2.3.1 Online activation

Please ensure that you are working on a computer with internet connection and press **Activate**. The IDE will contact Medexters license server and after a short time, the license and hence the IDE will be activated.

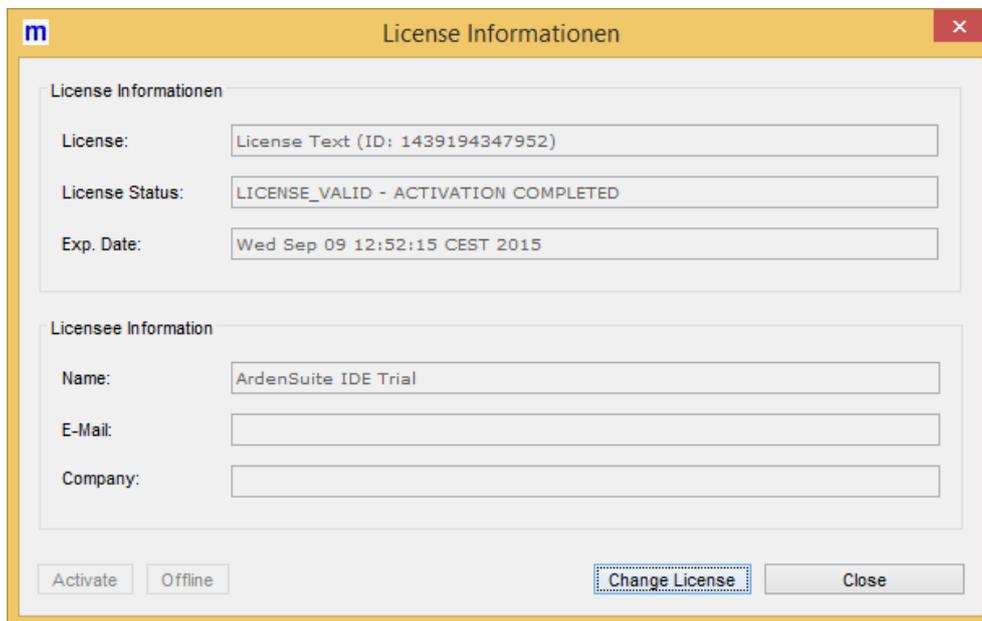


Figure 8: Activated license

### 2.3.2 Offline activation

If you do not have an internet connection on your working computer, you can perform an offline activation. For that purpose click on **Offline**. The following window will appear:

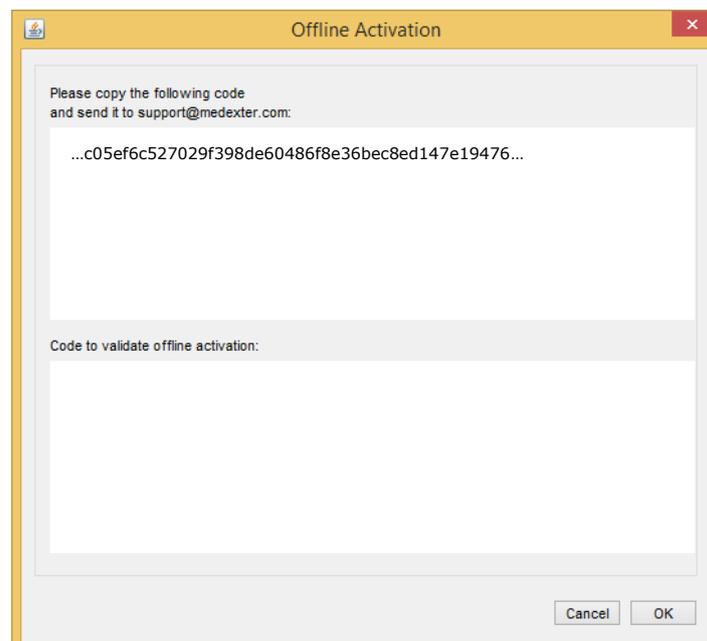


Figure 9: Offline activation

The code in the box represents your computer and has to be sent to [support@medexter.com](mailto:support@medexter.com). Medexter Healthcare will send you the corresponding validation code to activate your IDE. After filling in the code in the corresponding field ("Code to validate offline activation") and clicking **OK**, your IDE is activated.

### 3 Understanding the ARDENSUITE IDE

By describing its key features, this chapter aims at providing you with an enhanced understanding of the ARDENSUITE integrated development and test environment and its functionality. In the first section, basic concepts of the Arden Syntax will be illustrated. The second section will address the ARDENSUITE IDE's three core components.

#### 3.1 Arden Syntax – Basic Concepts

Several important features of the Arden Syntax will be summarized below. For a more detailed description of the Arden Syntax, see HL7's [Arden Syntax specification](#).

##### 3.1.1 Medical Logic Modules (MLMs)

Medical knowledge in the Arden Syntax is arranged within Medical Logic Modules (MLMs). One MLM represents sufficient knowledge to make a single decision. One or more MLMs are stored within a file that has the extension ".mlm".

##### 3.1.2 Calling MLMs

MLMs are managed by a runtime environment, namely the ARDENSUITE engine. The engine manages a list of compiled MLMs and handles communication with requesting entities. With the engine assuming these tasks, there is no concern for the host system in terms of implementation or physical presence of the MLMs. When the host system calls an MLM, the engine reacts by executing the respective MLM and returning the results (= output) to the host system:

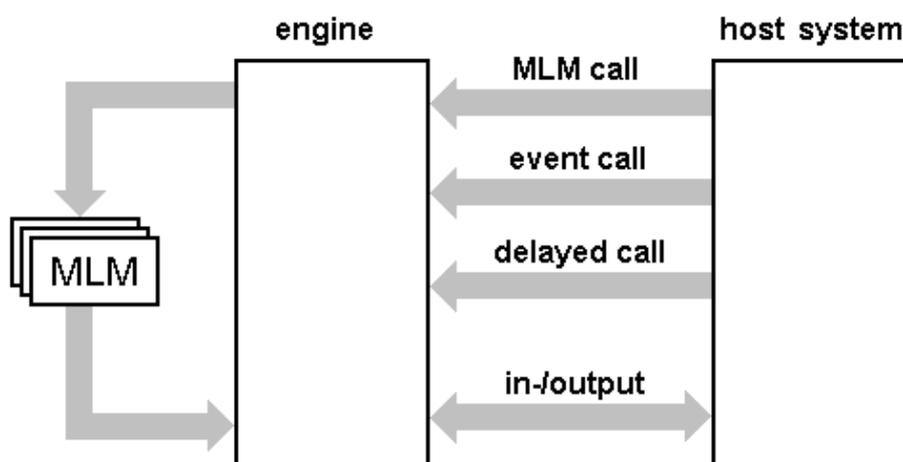


Figure 10: Communication between Arden Syntax Engine and Host System

There are three ways to call an MLM:

- **MLM Call:** A single MLM is executed by its name (and optionally by its institution).
- **Event Call:** All MLMs that are defined to be sensitive to the stated events are executed.
- **Delayed Call:** MLMs can be defined to be called a specified amount of time after the occurrence of certain events (or at an absolute point of time).

### 3.1.3 Data In- and Output

The main feature of an MLM is to apply the knowledge that is stored in it to input data, subsequently sending the results back to the requesting entity. The interaction between an MLM and the requesting entity (i.e., the host system) is illustrated below:

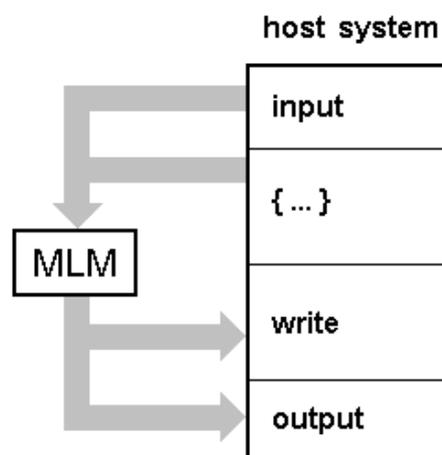


Figure 11: MLM Data Input and Output

- **Input:** Data is sent to the MLM during an MLM call or event call.
- **Curly Brace Expressions {...}:** These expressions provide a number of dynamic interaction possibilities between an MLM and any external system. They can for example be used to query databases or to call external functions.
- **Write Statements:** Texts can be written to destinations that are maintained by the Arden Syntax engine.
- **Output:** After the successful execution of an MLM, the results are sent back to the requesting host system.

### 3.1.4 ARDENSUITE IDE – Components

The ARDENSUITE IDE consists of three core components: Authoring, compiler and engine. While the authoring component is needed during the entire MLM developing process, both the compiler and engine component are working in the background. The compiler component translates MLMs into an executable format and the engine component is used for testing the developed MLMs.

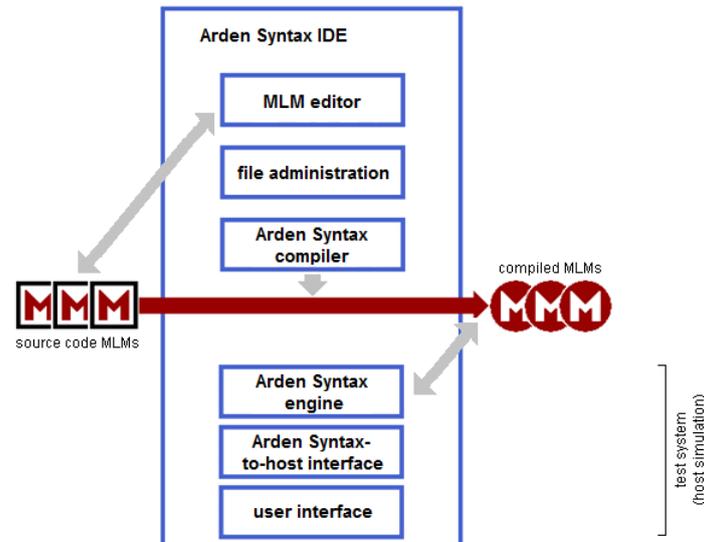


Figure 12: ARDENSUITE IDE Components

### 3.1.5 Authoring Component

This component allows writing, testing and organizing MLMs. It includes the following sub-components:

- **MLM Editor:** Word processing tool with supporting components that allow easy MLM editing (e.g., syntax highlighting, searching, auto-complete syntax constructs).
- **Test Environment:** Enables calling MLMs and events by using the Arden Syntax engine internally. MLMs can be tested and verified by providing sample input and sample curly brace expression evaluation results.
- **Log Component:** Allows continuous logging while executing MLMs.
- **File and Project Administration:** Administration system for MLMs. Possibility to organize MLMs into individual projects.

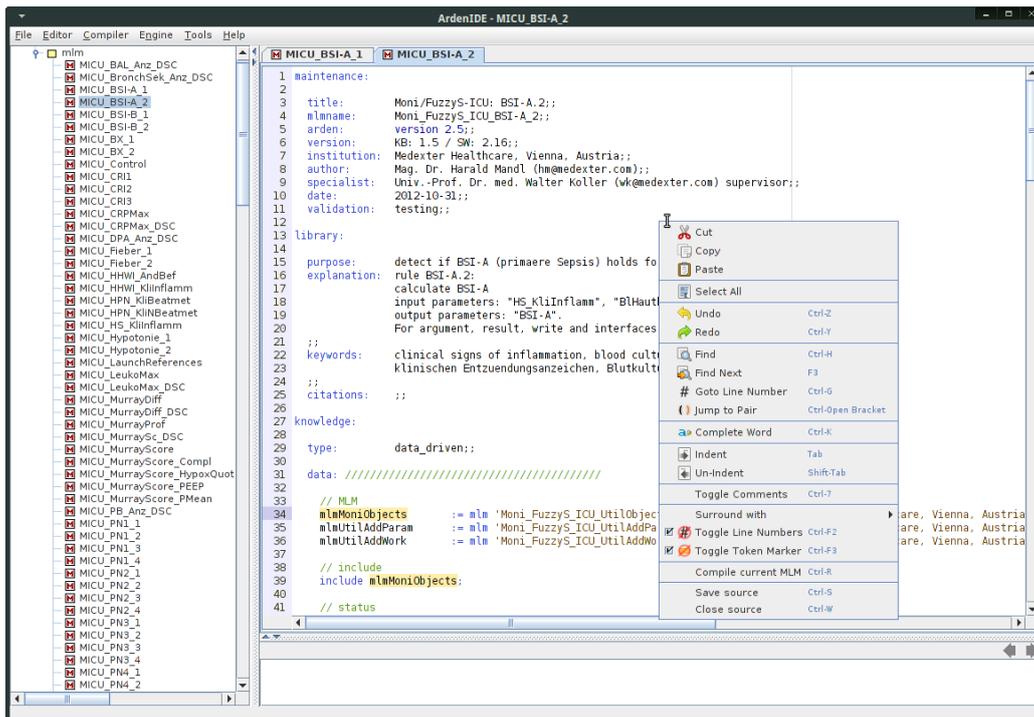


Figure 13: MLM Editor and File and Project Administration

### 3.1.6 Compiler Component

When given MLM source code files, the ARDENSUITE compiler creates JAVA binary objects. These compiled MLMs are managed and executed by the ARDENSUITE engine. In addition to its use as part of the ARDENSUITE IDE, the ARDENSUITE compiler is also available as command line application.

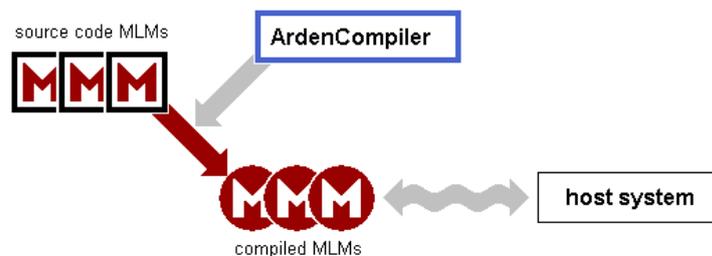


Figure 14: MLM Compilation

### 3.1.7 Engine Component

The engine component is responsible for executing compiled MLMs and for communicating with a host system. That is, it manages incoming MLM requests from the host system, executes the respective compiled MLMs, manages data in- and output (see 3.1.3), and returns the results to the host system. Communication between engine and host system requires an implementation of the so called Arden-Syntax-to-host interface. The ARDENSUITE IDE comes with a built-in implementation of this interface that is used to enable the communication between the ARDENSUITE engine and the test environment (that simulates the host system).

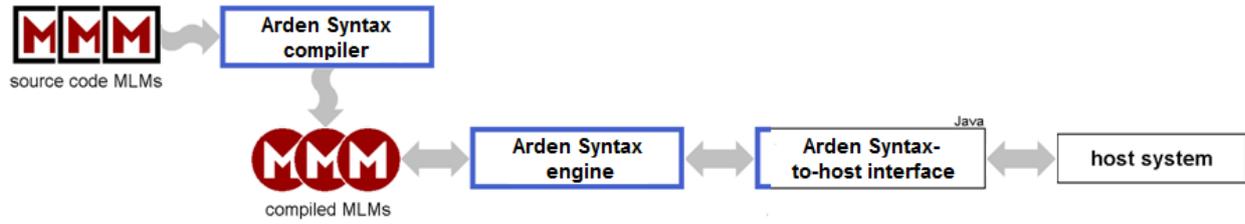


Figure 15: ARDENSUITE Engine Interaction with Compiler and Host Interface

## 4 Using the ARDENSUITE IDE – Five Minutes Tutorial

On the following pages we will guide you through starting the ARDENSUITE IDE for the first time and creating, compiling, and testing an MLM. A more detailed user guide can be found in the ARDENSUITE IDE help menu (**Help** → **Open Documentation**).

### 4.1 Start

Since the ARDENSUITE IDE is a JAVA application, a JAVA virtual machine must be installed in order to run it. JAVA can be downloaded at the Sun Developer Network [site](#) (JAVA 6.0 or later is needed). After installing JAVA, call the JAR that includes the IDE. This is possible for example by using the following command:

```
# javaw -jar jars\ArdenIDE-single.jar -L en
```

Please note that the location the application is started from will be the main directory of the application. For example, when the Arden Syntax IDE is installed into directory `c:\programs\ArdenIDE`, then `c:\programs\ArdenIDE\` becomes the main directory of the application and will contain several additional directories like `c:\programs\ArdenIDE\jars`, or `c:\programs\ArdenIDE\documentation\`, etc. In this case, call the jar from `c:\programs\ArdenIDE\`. Please make sure that the working directory is also the main directory of the application (`c:\programs\ArdenIDE\`).

In case the extension `.jar` is already connected to `javaw`, calling `'jars\ArdenIDE-single.jar -L en'` is sufficient.

For an English version use `-L en`, for a German version use `-L de`. For detailed localization use the `-C` flag (e.g., `-C AT` or `-C US`).

On systems running a Microsoft operating system, start menu entries that can be used to start the ARDENSUITE IDE are created during the installation process.

### 4.2 Adding MLMs

After starting the application, a predefined project ('Projekt') appears as top element of the library (file overview on the left side).

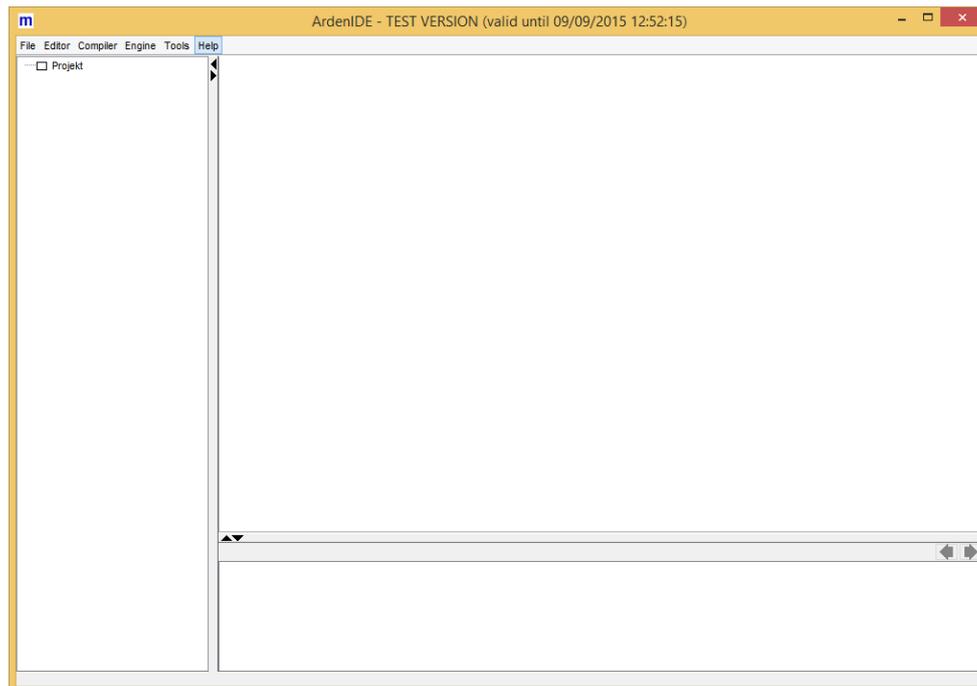


Figure 16: ARDENSUITE IDE First Start

To add an MLM (see Figure 17), click the right mouse button on the folder where you want to add the MLM (initially, there is only 'Projekt') and choose **Add File**. The dialog **Add File** opens. Under **File System** → **Source Directory** click the button **Browse**, then choose the MLM you want to add (for example: ...ArdenIDE\documentation\mlm \TestMlm1.mlm) and click the button **Add**.

After an MLM file is added, it can be opened by double-clicking the respective library node. In order to add a folder, click **Add Folder** and proceed in the same way as adding a file.

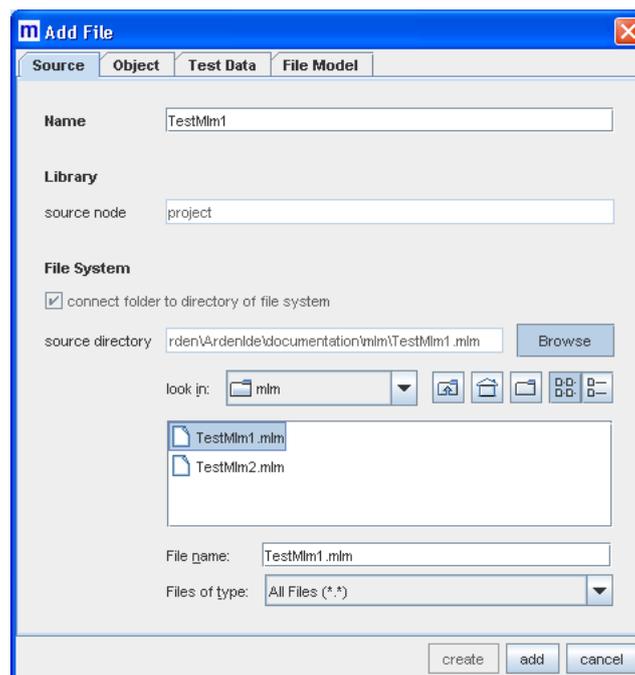


Figure 17: Add File Dialog

### 4.3 Compiling MLMs

To compile an MLM, please click the right mouse button within the MLM editor and choose **Compile Current MLM**:

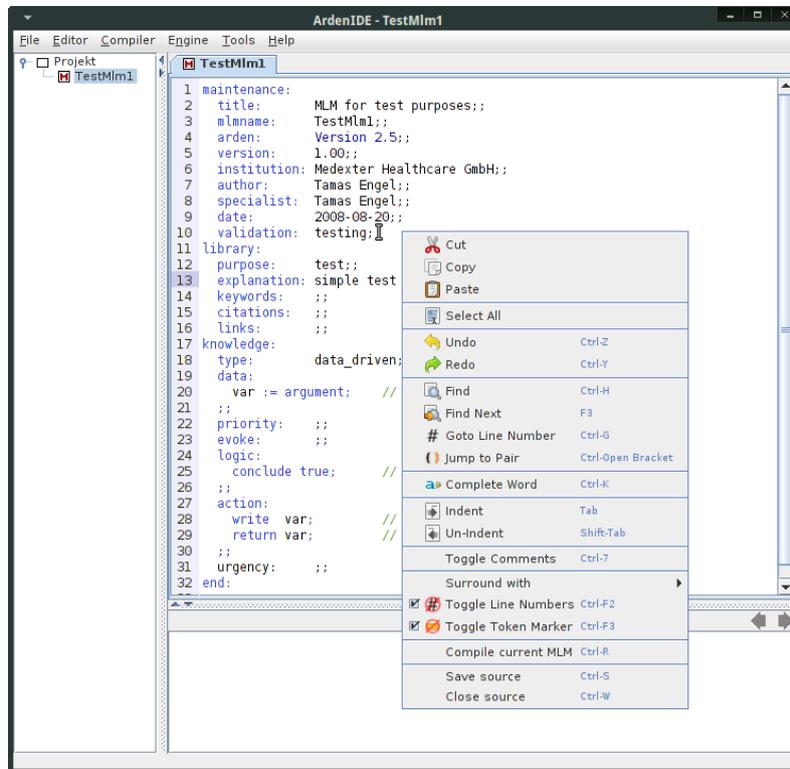


Figure 18: Compile Current MLM

When the compilation has finished, messages relevant for this compilation are shown at the bottom of the screen in the message panel. During compilation, a file with the compiled MLM object  and a file with test data  are created for the compiled MLM. They are located in the file overview on the left side of the screen (see Figure 19):

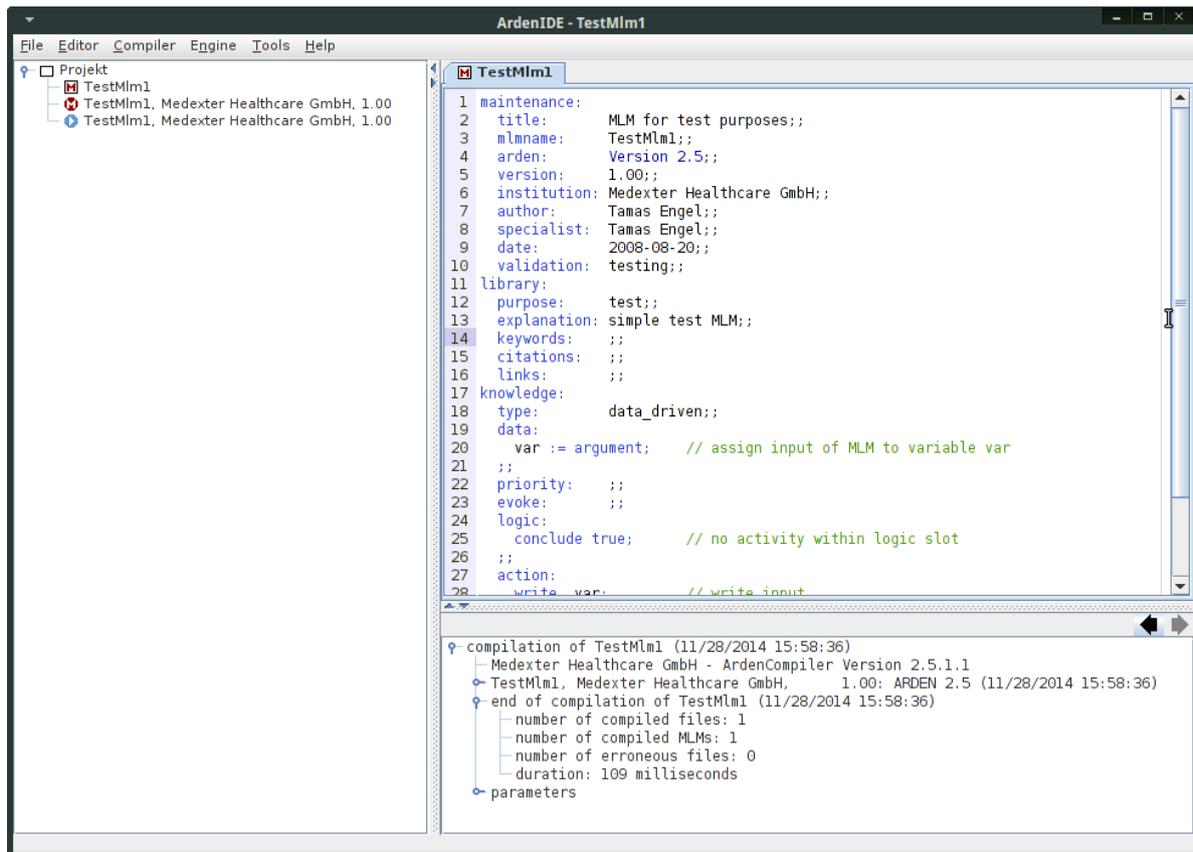


Figure 19: Compiled MLM

To open the test screen for the compiled MLM, double-click on the created test data . On the test screen (see Figure 20) you can add new test cases by clicking the **Add** button. Depending on the compiled MLM, various input fields will be activated (e.g., if an MLM contains a curly brace expression, an input field for a curly brace expression will be shown). Please provide test data in the designated fields. Click **Save Parameters** to save the edited test case. In order to run the MLM with the entered test data, please click the **Call MLM** button. The generated output/results (e.g., write and return statements) will be shown in the bottom part of the test screen.

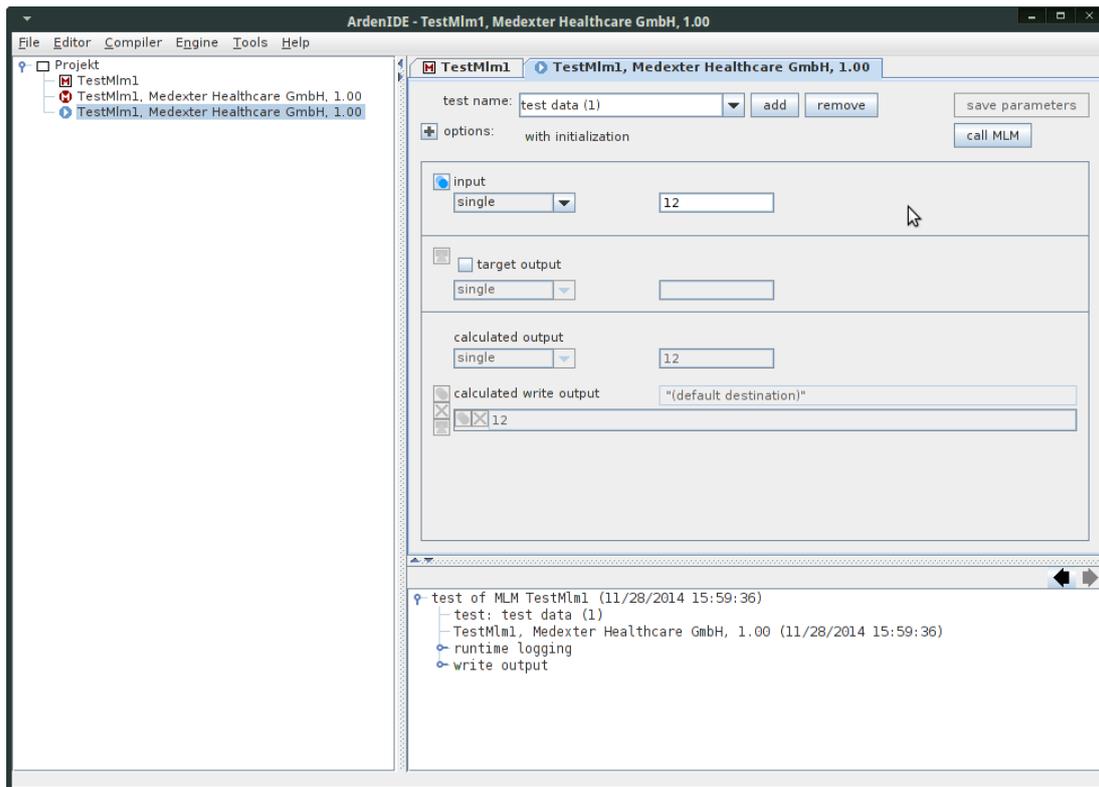


Figure 20: Test Screen

This Five Minutes Tutorial is designed to help you get started using your ARDENSUITE IDE. A considerable amount of additional information on functions and details of the ARDENSUITE IDE can be found in the ARDENSUITE IDE's help menu. If you encounter any difficulties while using the software, please consult the help menu that can be accessed via the ARDENSUITE IDE menu under **Help** → **Open Documentation**. If you need any further help or information, please visit our website [www.medexter.com](http://www.medexter.com) or write an email to [office@medexter.com](mailto:office@medexter.com).