

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

SIRS Notification as an Example

Date: 12/17/2019

Version: 2.2

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
MLM INPUT	4
INPUT DATA VIA REST	4
DATABASE CONNECTOR VIA CURLY BRACES	4
FHIR CONNECTOR VIA CURLY BRACES	4
COMPILING AN MLM IN THE ARDENSUITE IDE	5
TESTING AN MLM IN THE ARDENSUITE IDE	9
DEPLOYING COMPILED MLMS ON THE ARDENSUITE SERVER	11
FURTHER USER SUPPORT AND DOCUMENTATION	15



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: https://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, December 2019

Version: 2.2

Download at https://www.medexter.com/



Introduction

In this *how-to* instruction manual, we will guide you step-by-step in 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 Postman for all REST communication.



Files

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

- CTD_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.
- CTD_SIRS-Notification2: This MLM is a corrected version of CTD_SIRS-Notification1. This MLM contains no errors and can be tested and deployed.

and two text files with the input data for CTD_SIRS-Notification2 in JSON format:

(rsc_REST1.txt and rsc_REST2.txt)

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.

MLM Input

For the MLMs used in this *how-to*, data is provided with each REST call. However, this is not the only way to provide external data for processing inside MLMs. Using the ARDENSUITE Server, there are altogether three different ways for an MLM to receive data from the outside:

Input Data via REST

As described in the *how-to* document at hand. (For more details, see also our *how-to* document "How to Call Arden Syntax MLMs on an ARDENSUITE Server Using REST and SOAP" available in our online Learning Center)

Database Connector via Curly Braces

Described in the following *how-to* document available in our online Learning Center: "Achieving Database Connectivity in Arden Syntax Using the ARDENSUITE Database Connector—SIRS Notification as an Example"

FHIR Connector via Curly Braces

Described in the following *how-to* document available in our online Learning Center: "Achieving FHIR Connectivity in Arden Syntax Using the ARDENSUITE FHIR Connector"



Compiling an MLM in the ARDENSUITE IDE

Start your ARDENSUITE IDE and choose your preferred workspace location. A workspace is a folder on your system, where projects and custom IDE settings are stored. For example, if you want to show line numbers: Window -> Preferences -> General -> Editors -> Text Editors -> Check 'Show line numbers', this setting will be stored in and for your workspace.

Before you can create or load MLMs, you have to create an Arden Syntax Project. Do this by clicking File -> New -> Project in the Menu (see figure below).

(w	workspace - test2/MLMs/testmlms/Template.mlm - ArdenSuite IDE									
File	Edit Navigate Search Project Run ArdenSuite Window	w Help								
	New Alt+Shift+N >	📑 Project								
	Open File	🗳 Folder 🦿								
۵,	Open Projects from File System	File								
	Close Ctrl+W	Untitled Text File								

Select Arden Syntax Project from the category Arden Syntax. Choose a project name. Click Finish to create the project. The wizard will automatically generate a template .mlm file. This template MLM is simply able to accept two input parameters and return a string as result.

Wizards:	Τ
type filter text	
> 🗁 General	
🗸 🗁 Arden Syntax	
🛕 Arden Syntax Project	
> 😕 Java	

At default settings, the newly created Arden Syntax Project will appear in the Project Explorer on the left side of the IDE. It includes two folders named MLMs and templates. You may delete these folders and create new ones with custom names at any time.



workspace - ArdenSuite IDE File Edit Navigate Search Project Run ArdenSuite Window Help	- 0
File Edit Navigate Search Project Run ArdenSuite Window Help	
The Edit Navigate Search Project Kun Ardensuite Window Help	
📸 🕶 🗐 🐚 🔼 💶 🖓 🕶 🦛 🖷 🧗 🖛 🏷 🗢 🗢 🕶	Quick Access
	· · · · ·
V 🕒 How locompile restrictions	
v > templates	
Mul Template.mlm	
i outline is not available.	
	r≓ E • r * •
No conceles to display at this time	

Generate a new MLM file by clicking File -> New -> Other to open another wizard. Select New MLM File from the category Arden Syntax. This wizard allows you to choose a file name (e.g., CTD_SIRS-Notification1.mlm) and the location where this file should be generated (Browse button). Choose a folder from your newly created project.

🖨 v File	vorkspace - test2/MLMs/testmlms/Template.mlm - ArdenSuite IDE Edit Navigate Search Project Run ArdenSuite Window	Не	lp.			Wizards:
	New Alt+Shift+N >		Project			type filter text
۵,	Open File Open Projects from File System	≅ ∎	Folder File	3		> 🔁 General
	Close Ctrl+W	ľ	Untitled Text File			A Arden Syntax Project
	Close All Ctrl+Shift+W		Other	Ctrl+N .9	;	MUM New MLM File

Copy the content of our provided CTD_SIRS-Notification1.mlm to your newly created file and do the same with CTD_SIRS-Notification2.mlm. You can also drag-and-drop those files onto one of the project's folders (choose copy files) or just copy them to the project location on your hard disk (right-click on your project and click Refresh for newly added files to show up).

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 MLM

🖨 workspace - test1	/MLMs/template	s/Template	.mlm - Arder	nSuite IDE	
File Edit Navigat	e Search Proj	ject Run	ArdenSuite	Window	Help
-	0 9 - 4	/ 🚽	- 祠 - や	⇔ - ⇒	• 🔬
陷 Project Explorer	22			E 4	\$ ▽ □
∽ 📂 test1					
V 🗁 MLMs					
V 🖂 templa	tes ObiHandles.csv				
👷 Min	Template, Mede	xter Health	are, 0.1.mlm	obj.gz	
MUM T	New			×	
> 🗁 test2	INCOV				
	Open			F3	
	Open With			>	
	Сору		C	Ctrl+C	
Ē.	Paste		(Ctrl+V	
×	Delete		[Delete	
	Move				
	Rename			F2	
<u>1</u>	Import				
2	Export				
8	Refresh			F5	
	Team			>	
	Compare With			>	
	Replace With	-		>	
🗄 Outline 🛛 🔼	Compile MLM)		<	<mark>la</mark> la ⊓
	Properties		Alt+	Enter	



• Right-click on the code in the main screen on the right, and click on the menu item

Compile MLM

T002MLM.mlm			
⊖ maintenance:			
title: MLM1;;			
mlmname: MLMName;;			
arden: version 2.9;;			
version: 0.1;;			
institution: TestI;;			
author: TestI;;	2	Undo	Ctrl+Z
<pre>specialist: TestI;;</pre>		Revert File	
date: 2018-05-29;;			
validation: testing;;		Save	Ctrl+S
⊟ library:		0.01.0	
purpose: template mlm to show ba	15	Open Declaration	F3
explanation: description how the	LS	Open Generated File	
keywords: ;;		Quick Outline	Ctrl+ O
citations: ;;		Quick Outline	Cui+O
links: ;;		Open With	>
Knowledge:		Show In	Alt+Shift+W >
<pre>data:</pre>			
allValues:= Argument:	of	Cut	Ctrl+X
value1 :=allValuec[1]:		Carry	Chillion Chillion
value2 :=allValues[2];		Сору	Ctri+C
valuezalivalaes[z],		Copy Qualified Name	
	1	Paste	Ctrl+V
priority: ::			curre t
evoke: ;;		Rename Element	Alt+Shift+R
logic:		M-Edute	
		validate	
var := 1990-03-02T00:00:00 2	-	Quick Fix	Ctrl+1
		Source	>
conclude true;		Jource	
11		Find References	Ctrl+Shift+G
action:			
return plext;		Run As	>
jj	*	Debug As	>
end:		Team	>
		a unu	
		Compare With	>
		Replace With	>
		Source	>
		Find References	
(A	Compile MLM	Ctrl+Shift+C

Use the main menu: ArdenSuite -> Compile MLM (this will compile the MLM currently displayed in the editor)



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





Additionally, the editor's syntax highlighting may give a hint where to search for an error. The MLM CTD_SIRS-Notification1 contains two program errors. The assignment in line 20 is missing a : character and the if statement in line 45 is missing an endif; statement. Upon compiling a correct MLM, i.e., CTD_SIRS-Notification2, compilation feedback will show no errors. Furthermore, two extra entries appear in the tree structure on the left pane: a metadata file (MlmObjHandles.csv) and a compiled MLM (.mlmobj.gz) which may be uploaded to the ARDENSUITE Server or used for testing/debugging purposes.

Testing an MLM in the ARDENSUITE IDE

To test a compiled MLM, you need to right-click on a compiled MLM (.mlmobj.gz) in the Project Explorer to open a context menu and choose Debug MLM. This will open an input window, where input parameters can be specified in JSON format, the same format you will use when calling an MLM on the ARDENSUITE Server via REST.



Tasks E Console X		
[iconTeDetaus]] converted icon stains to icon obiestern medeuter andre ide andreabieste And		
[]soniouatavalue] converted]son string to]ava objectcom.medexter.arden.ide.ardenobjects.Ard	enL1st@/c552041	
start BLOCK		~
start ASSTGNMENT	Uebug – L	~
start ASIGNENT		
and ADGINENT is (20, 90, 22, 22, 2065, 16) medewtee ander lang on us statement OnAngument	Please enter a JSON string as you would when calling an MILM that was	
end Accoment := (39,09,25,55,5965,10) medexter.arden.tang.op.vc.statement.opArgument	uploaded to the ArdenSuite Server	
end Assidurent Vold		
and BLOCK := Void medexter.arden.lang.op.vc.statement.opbiock		_
START BLUCK	1 1 10	
STAPT ASSIGNMENT	type: list,	
end ASSIGNMENT Counter := 0	primary Lime : null,	
Start IF_IHEN	applicability: 1,	
start OR_FUZZY	"values": [
start GREATER		
start IDF_VAR	"type": "number",	
end LUF_VAR temperature := 39	"primary lime": null,	
end GKEALEK := true medexter.arden.lang.op.vc.comparison.OpGreater	applicability": 1,	
end GREATER := true medexter.arden.lang.op.vc.comparison.OpGreater	"value": 39	
start LESS	h.	
start IDF_VAR		
end IDF_VAR temperature := 39	"type": "number",	
end LESS := false medexter.arden.lang.op.vc.comparison.OpLess	"primary l ime": null,	
end LESS := false medexter.arden.lang.op.vc.comparison.OpLess	"applicability": 1,	
end OR_FUZZY := true medexter.arden.lang.op.vf.OpOrFuzzy	"value": 89	
start ELSEIF_CASES	<u>}</u> ,	
start OR_FUZZY	{	
start GREATER	"type": "number",	
start IDF_VAR	"primaryTime": null,	
end IDF_VAR temperature := 39	"applicability": 1,	
end GREATER := true medexter.arden.lang.op.vc.comparison.OpGreater	"value": 23	
end GREATER := true medexter.arden.lang.op.vc.comparison.OpGreater	3.	
start LESS		
start IDF_VAR	"type": "number",	
end IDF_VAR temperature := 39	"primary Lime": null,	
end LESS := Talse medexter.arden.lang.op.vc.comparison.OpLess	"applicability": 1,	
end LESS := Taise medexter.arden.iang.op.vc.comparison.opLess	value : 33	
end OK_PUZZY := true medexter.arden.lang.op.vt.Opurruzzy		
start block	1 New arts the second and	
	type : number , "nrimen Time", null	
	"applicability" 1	
	"uplus": 2065	
and TDE VAR counter := 0	Value . 3903	~
end PUIS == 1 medester anden lang on ve arithmetic OnCale (al Cal		
end ASSIGNMENT counter	Close Debug	
end BLOCK := void medexter.arden.lang.op.vc.statement.OpBlock	close Debug	
end ELSETF CASES := void medexter.arden.lang.op.vc.statement.0pIfCase		
end IF THEN := void medexter.arden.lang.op.vc.statement.OpIf		
start IF THEN		
start GREATER		
start IDF VAR		
end IDF VAR heartRate := 89		
end GREATER := false medexter.arden.lang.op.vc.comparison.OpGreater		
end GREATER := false medexter.arden.lang.op.vc.comparison.OpGreater		
start ELSEIF_CASES		

The MLM CTD_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 enter the following list of parameters to test/debug the CTD_SIRS-Notification2 MLM:

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

}

Note: This JSON input can also be found in the rsc_REST1.txt in the .zip-folder accompanying this how-to document.

Based on these inputs, the expected output would be a string containing an alert for SIRS. In the console, you can see that all internal operations while executing the MLM are printed to the console in the same order as they are processed by the Arden Syntax Engine. This output might help to track down logic flaws or runtime errors.

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).



AS SERV MLM Management	Connectors -	System Admin	- Account	About	Logout
MLM Management					
Package	Name	Institution	Arden Syntax Version	Validation	Author
/ +					
Upload					

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).

Add Package	:	×
Name	ArdenHowTo	
Description	Arden How-to SIRS MLMs	
Institution	Medexter Healthcare	
Version	1	
Parent	1	
Cance	ЮК	

To add the package to the ARDENSUITE Server, press the OK button. A package will appear on the left pane of the screen. To upload MLMs to this package instead of uploading them 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 .mlmobj.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 CTD_SIRS-Notification2 in the package ArdenHowTo, the screen looks as follows:

AS SERV	MLM Management	Con	nectors 👻 System A	dmin - Accour	nt About Log	out						
MLM	MLM Management											
Packa	Package Name Institution Arden Syntax Version Validation Author MLM Version Upload Date CDS Hook Edit Delete									Delete		
▼ /	ArdenHowTo +	+	CTD_SIRS-Notification1	Medexter Healthcare, Vienna, Austria	version 2.9	production	knowledge engineering group	1.0	05-11-2019	ē		Û
Uplo	ad									1	🕅 Recy	cle Bin

To call this MLM, start Postman 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 (POST):

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

Do not forget to add a basic authorization header to your REST call (Postman – Authorization tab). In case of our test scenario, the JSON data that needs to be supplied in the Body segment of the REST call is the same as before, used to test/debug the MLM with the IDE:

Note: Again, this JSON input can also be found in the rsc_REST1.txt file in the .zip-folder accompanying this how-to document.

```
{
    "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"
}
```

}

The following input will not trigger an alert for SIRS. The result is null, since the MLM does not return anything in that case.

Note: This JSON input can also be found in the rsc_REST2.txt file in the .zip-folder accompanying



this how-to document.

```
{
   "type": "list",
   "primaryTime": null,
   "applicability": 1,
   "values": [
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 37
       },
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 89
       },
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 19
       },
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 30
       },
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 5000
       },
       {
           "type": "number",
           "primaryTime": null,
           "applicability": 1,
           "value": 9
       }
   ]
}
```

Further User Support and Documentation

This *how-to* instruction manual is part of our online Learning Center. Further available *how-to* documents are:

• Achieving **Database** Connectivity in Arden Syntax Using the ARDENSUITE Database Connector— SIRS Notification as an Example



- How to Call Arden Syntax MLMs on an ARDENSUITE Server Using **REST and SOAP** *SIRS Notification as an Example*
- Achieving FHIR Connectivity in Arden Syntax Using the ARDENSUITE FHIR Connector
- Achieving System Connectivity between **Activiti** BPMN Platform and the ARDENSUITE— Hepatitis B in Pregnancy as an Example

Visit the **ARDENSUITE Support Pages** online for a detailed documentation:

https://www.medexter.com/ardensuite_support

Or contact us with any questions at support@medexter.com.