**Object Oriented Matlab Multi-Path Algorithmic Medical Device Information Model**

**User Guide**

**Version 1.3.1**

**ISSUE DATE: TBD**

**DOCBOX Inc.**

**Contents**

[**1.0 Product Description**](#h.4d34og8)

[1.1 Acronyms](#h.2s8eyo1)

[1.2 References](#h.3rdcrjn)

[1.3 Contact Information](#h.1ksv4uv)

[**2.0 Document History**](#h.2jxsxqh)

[**3.0 Software Requirements**](#h.z337ya)

[**4.0 Getting Started with Matlab**](#h.3j2qqm3)

[4.1 Setting Matlab Enviornment](#h.1y810tw)

[4.2 Dynamic Algorithms](#h.2xcytpi)

# 

# 

# 

# 

# 

# 

# 

# 

# **1.0 Product Description**

Top level design and system description consists of Matlab’s language (ML) , and Matlab’s DDS PSP (MDDSPSP)

Using MDDSPSP, the program is able to subscribe to data that is specified by a domain ID number, a Topic Type, and a Topic Name. In order for the publisher and subscriber relationship to be functional, all three of the items mentioned above must match exactly. In addition, any naming conventions, structures, and data types must match exactly within the Topic Type.

Algorithms are implemented in Matlab to process the incoming data and then designed to interpret the meaning of this data. Using MDDSPSP, the program is able to collect data from external medical devices and run analysis in real-time.

## 1.1 Acronyms

PCA: Patient Controlled Analgesia

QOS: Quality of Service

DP: Domain Participant

DDS: Data Distribution Service

PSP: Pilot Support Package

MDDSPSP: Matlab/Data Distribution Service Pilot Support Package

## 1.2 References

MathWorks

Corporate Headquarters

3 Apple Hill Drive

Natick, MA 01760-2098

Phone: 408-647-7000

Fax: 408-647-7001

## 

## 1.3 Contact Information

Tracy Rausch, DOCBOX CCE CTO, [tracy@docboxinc.com](mailto:tracy@docboxinc.com)

Andrew Shruhan, DOCBOX Software Engineer, [ashruhan@docboxinc.com](mailto:ashruhan@docboxinc.com)

# **2.0 Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Authors | Description |
| 8/12/13 | 1.0.0 | APS | Initial Version |
| 11/06/13 | 1.1.0 | APS | Bringing up to date as far as software requirments |
| 12/20/13 | 1.2.0 | APS | MDPnP Initial Design |
| 01/02/14 | 1.3.0 | APS | Full MDPnP Design and description |
| 01/06/14 | 1.3.1 | APS | TItle Change |

# **3.0 Software Requirements**

Complete List of Required Software: (in order of installation)

-MATLAB

-RTI Connext

-MATLAB/DDS PSP (Download Link below)

<ftp://ftp.mathworks.com/pub/customer_pickup/DDSBlocksetPSP>

-Visual Studio (Recommended not required to run ML or MDDSPSP)

-Windows .NET 4.0 (Needed to Run Visual Studio)

-Windows SDK 7.1 (Needed to Run Visual Studio)

Additional Installation Instructions:

1) Refer to the MATLAB/DDS PSP User Guide provided with download for its integration into the Matlab environment. The User Guide is located in the following path:

<ftp://ftp.mathworks.com/pub/customer_pickup/DDSBlocksetPSP/users_guide.pdf>

2) Ensure the following environment variables are specified:   
 System Variables: (Not “User Variables for \*account name\*”)

**Create new environment variables:**

1) Variable Name: NDDSHOME

Variable Value: <\*RTI Installation Folder\*>\RTI\ndds.5.0.0

(Eg: C:\Program Files (x86)\RTI\ndds.5.0.0)

2) Variable Name: RTI\_LISCENCE\_FILE

Variable Value: <\*RTI Installation Folder\*>\RTI\rti\_license.dat

(Eg: C:\Program Files (x86)\RTI\rti\_license.dat)

Add Variable Value to Variable Name “Path”:

-Edit “Path” variable and add a semicolon to the end of the existing value.

-Paste the following after the semicolon :

<\*RTI Installation Folder\*>\RTI\ndds.5.0.0\lib\x64Win64VS2010

(Eg: C:\Program Files (x86)\RTI\ndds.5.0.0\lib\x64Win64VS2010)

# **4.0 Getting Started with Matlab**

Using GitHub follow the link below to download the Matlab source code.

<https://github.com/ashruhan>

In this GitHub account two notable accounts exist,

1. ML\_demo\_apps\_template

Before cloning repository make sure branch is TemplateStablev. Template is created to subscribe to the MDPnP\_demo\_apps devices and stream in static and waveform data. The Application is intended to provide a basic platform in how algorithms are supported using MDPnP\_demo\_app as a data source. Data is sorted by domain, topic, types, data structure, and key. Readers are initialized to read from a topic and type using specific data structure. If data is found on that topic then the data is sorted using the corresponding metric\_ID.

1. ML\_demo\_apps

Before cloning repository make sure branch is AlgorithmStablev. Program is created to subscribe to the MDPnP\_demo\_apps devices and stream in static and waveform data. The Application is intended to provide a basic algorithm platform in which simple algorithms are executed on the sourced MDPnP device data. Data is sorted by domain, topic, types, data structure, and key. Readers are initialized to read from a topic and type using specific data structure. If data is found on that topic then the data is sorted using the corresponding metric\_ID.

In other words the ML\_demo\_apps\_template is a blank algorithm platform that researchers can build upon to test prebuilt algorithms. Were as the ML\_demo\_apps is a functional program that executes simple algorithms using object oriented matlab.

## 4.1 Setting Matlab Enviornment

In Matlab, specify the “Current Folder” as the location corresponding to that which the clone / zip folder was extracted to. Highlight the folders visible, right-click, and choose “Selected Folders” in the “Add to Path” option of the menu. The final step before running any of the programs is to be sure the file “USER\_QOS\_PROFILES.xml” exists in the “Current Folder” window, outside of any folder that was just added to the path.

To execute the program, multiple options are available. The user can use any of the following methods:

-click the desired file and drag into the Matlab “Command Window”

-double-click the desired file to open the script and then press the green “Run” button at the top of the window

-type the filename (without “.m”) in the “Command Window” and press enter to execute

## 4.2 Dynamic Algorithms

Matlab language is used in the development of algorithms for this system, allowing for full customization. The MDDSPSP is integrated into the Matlab environment for the acquisition and distribution of data. Data being published from a variety of medical devices can be subscribed to directly in Matlab and evaluated accordingly inside of fully customizable algorithms

Throughout the course of execution, events and/or alarms can be triggered if a vital parameter exceeds either threshold or the specified percent change in the allotted time period (duration). For example if a patient’s heart rate slows below 60 beats per minute for longer than 10 seconds, an event can be triggered sending a stop command to a PCA infusion pump.

The capability to simply plug an algorithm into the system is certainly available, but would require minor code alterations.

## 