

BYK.RoboticDataService

Data converting and storing service software for

bykLINK

Documentation

Last update: Jun 3, 2016

Table of Contents

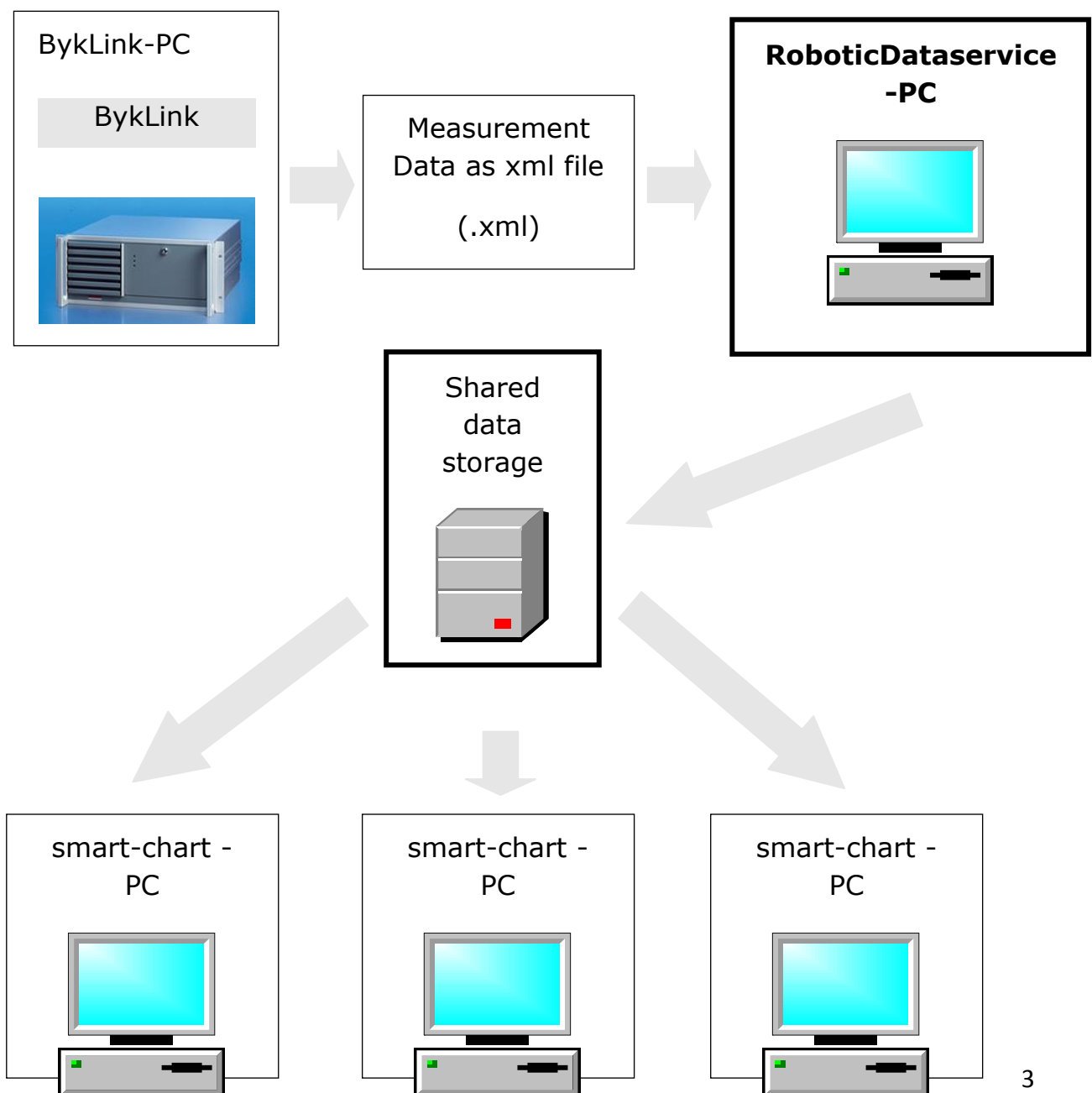
1. General.....	3
2. Installation and Start-up	4
2.1 Technical Conditions.....	4
Minimum hardware requirements:	4
Software	4
Installation	4
3. The Graphical User Interface	5
4. Configuration BYK.RoboticDataService.....	5
4.1 Database Settings	6
4.2 Source Folders and Files	7
4.3 Symbol mapping	8
4.4 Organizer	11
4.5 Color Standards	12
4.6 Paintlines	13
4.7 Checkzones	13
5. Run procedure	14

1. General

1.1 System Description

BYK.RoboticDataService is a data converting and storing service software for bykLINK. Because bykLINK stores all measurement data in text files, it is needed to read and convert them in order to let smart-chart users see the data in smart-chart -> analysis module. The conversion runs automatically in the background. The users can do their analyzing continuously during BYK.RoboticDataService runs.

1.2 Overview of components



2. Installation and Start-up

2.1 Technical Conditions

Minimum hardware requirements:

Processor:	2GHz or faster
RAM:	4 GB
Available disk space:	500 MB

If the data is to be stored on the PC the Minimum size of the used hard disk depends on the number of measuring data, which are to be kept long-term on the Computer.

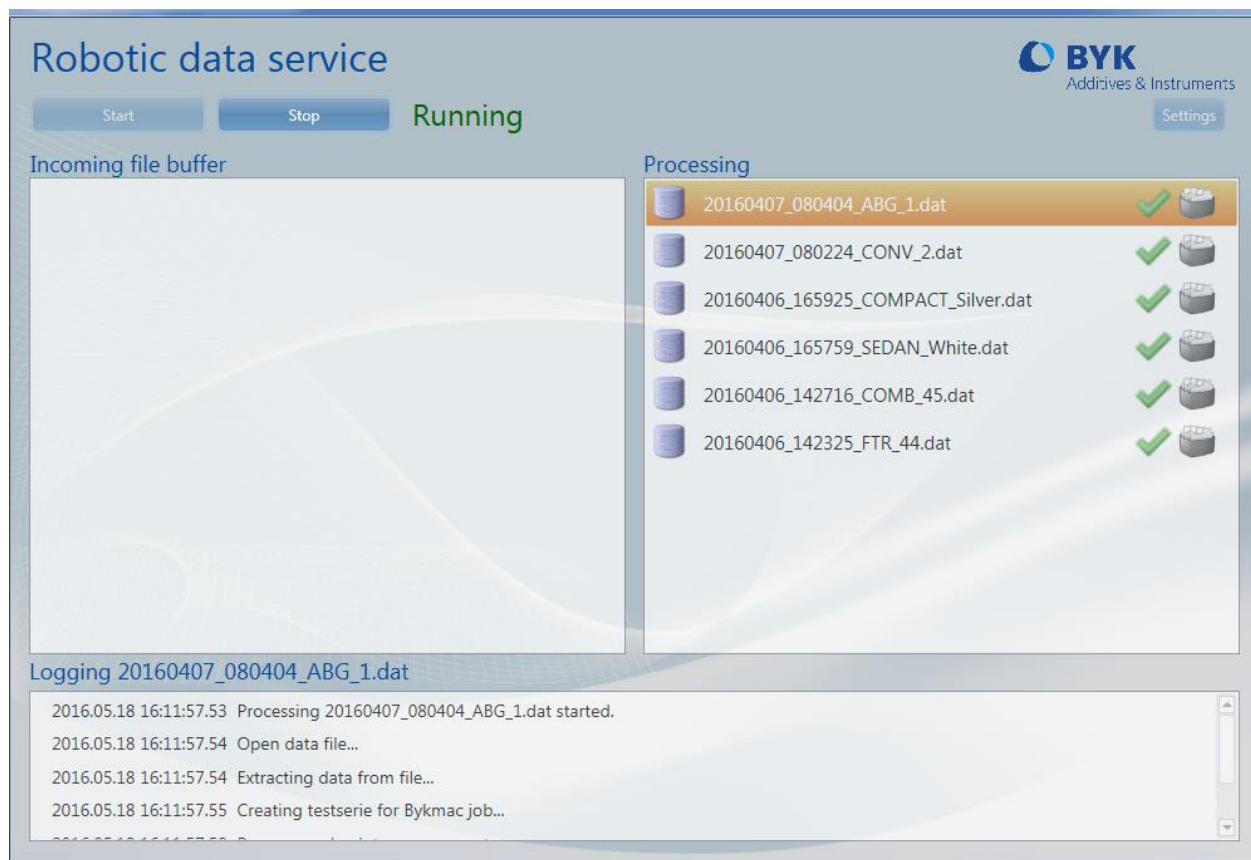
Software

Operating system:	Windows Vista, Windows 7, 8, 8.1, 10
Framework:	Net 4
BYK-Gardner:	smart-chart 3.x.x. or higher

Installation

Install smart-chart. You can read about the smart-chart installation details in the documentation. BYK.RoboticDataService is part of the installation.

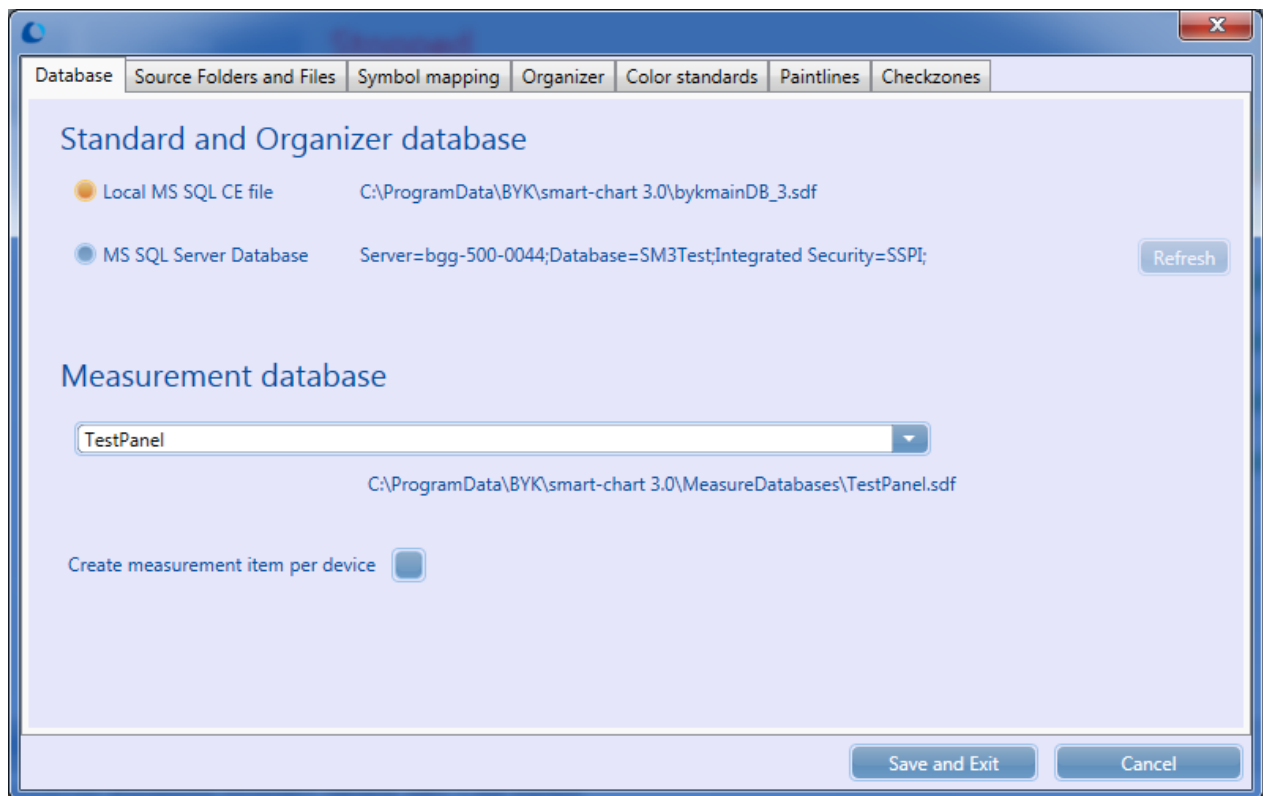
3. The Graphical User Interface



It is quite simple. There are only three buttons “Start”, “Stop”, “Settings” and three windows to show the incoming files, the already processed files and the log content.

4. Configuration BYK.RoboticDataService

The configuration is performed with the config file `RoboticServiceConfig.xml`. The config file defines the main database path, the connection string, the resource folder path and the so called mapping areas. Editing the file is not necessary; the software provides a user friendly window to set the parameters. The **Settings** button opens the window.



4.1 Database Settings

Standard and Organizer database

The software needs information about the used color standards, organizers and check zones. They are stored in the so called main database of smartchart. The main db can be defined as a local database file (.sdf) or a MS SQL server database. (or both, but only one is active) The user can choose the place of the main db with the radio buttons. If the server connection is changed after the last start of BYK.Dataservice the “Refresh” button will be activated. Pressing it updates the connection string.

Measurement database

All the data coming from the robotic system will be stored in the database selected in the combo box. The box contains all the databases linked to smartchart.

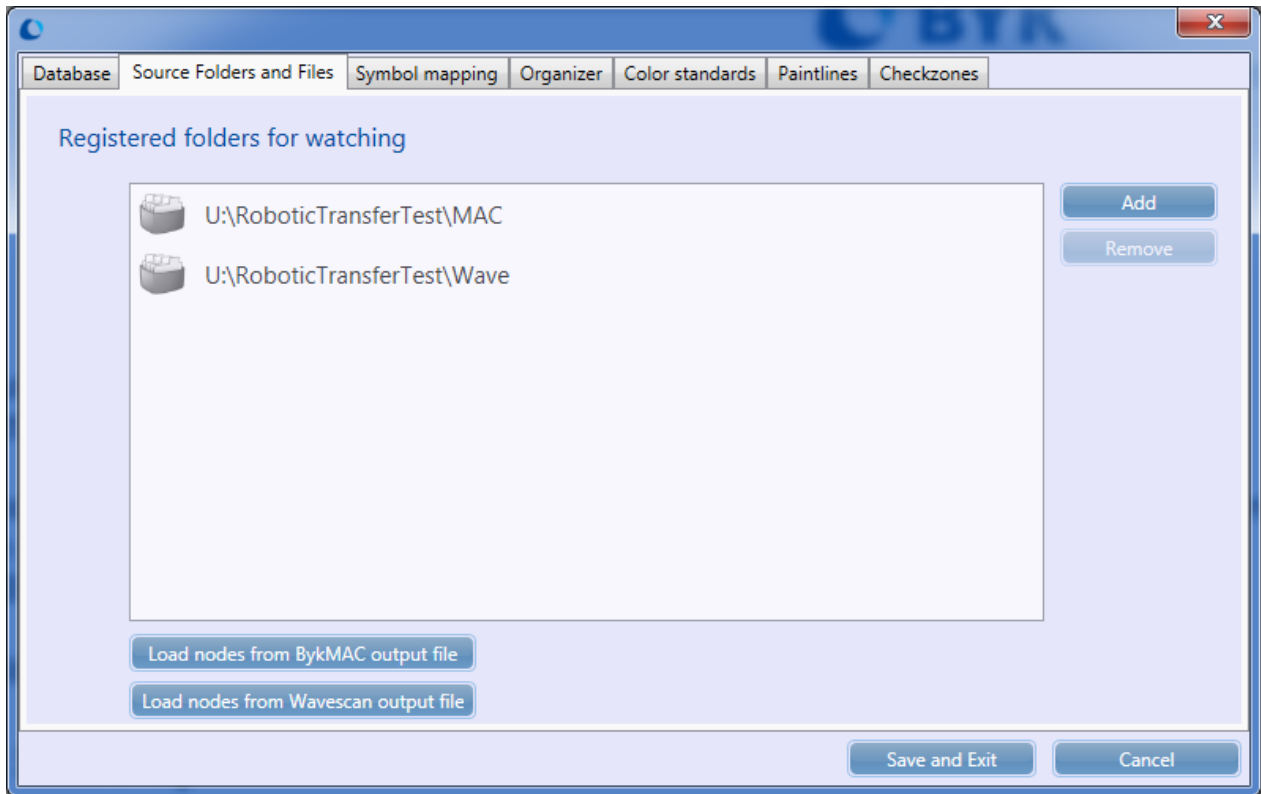
Create measurement item per device ☐

With bykLink it is possible to control up to 9 Byk-mac and 9 wave-scan devices. (In arbitrary variations) If the check box is:

- unchecked, all the data in one input file will be stored as one measurement,

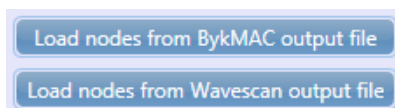
- checked, the program stores for all devices found in the input file an own measurement item into the database.
-

4.2 Source Folders and Files



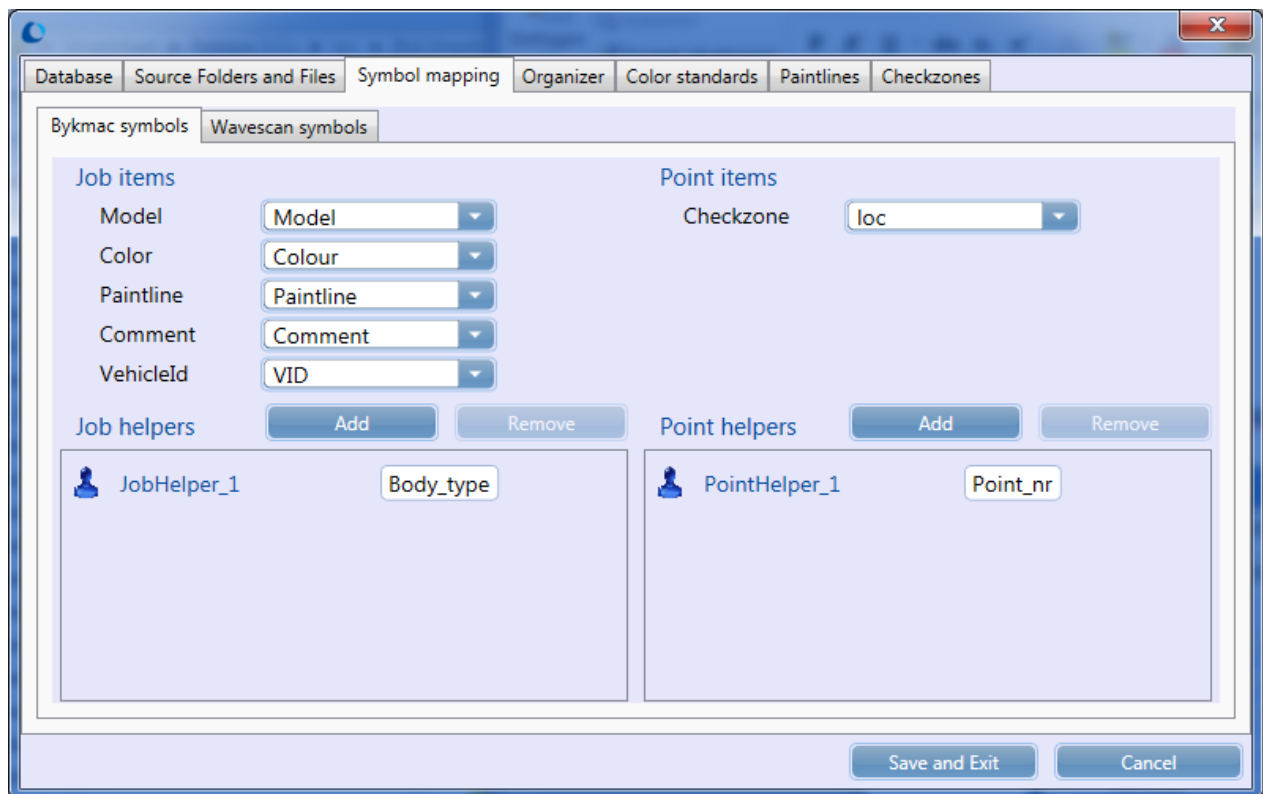
Registered folders for watching

In this window the user can define the folder path of bykLINK output files. Because it is possible to define more output folders in bykLINK (e.g. Wavescan, Bykmac) it is also possible to create more entry here. The software checks all the defined paths and reads the new data files if any exist.



BykLink offers xml files as output for Byk-mac and / or wave-scan data. If the user already has a data file, with clicking the button (depends on the instrument type) and selecting a file, it will be analyzed and the used symbols temporary stored for helping the user to define the right symbols in the "Symbol mapping" tab. (See below)

4.3 Symbol mapping



Both device families have their own symbol mapping, therefore also an own sub tab. Here only the Byk-mac tab is explained because the wave-scan ~~one~~ has absolutely the same construction and rules.

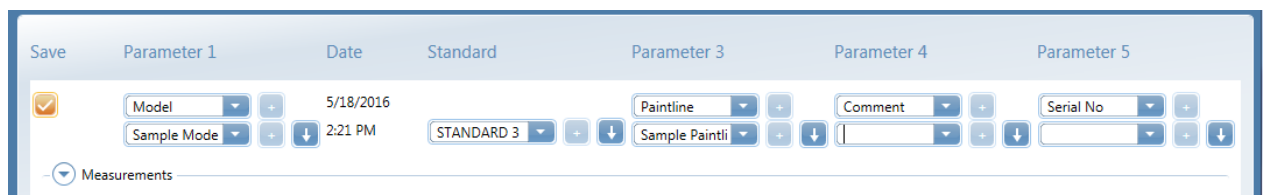
There are 2 groups of parameters:

Job items: they are valid for the whole dataset.

Point items: they values vary with the check zones.

Job items

There are five predefined parameters. They are identical with smart-chart's parameter 1 to 5 in the data transfer module. (See picture below)



The dataservice defines Parameter 1 as Model. This field is used to identify the car body.

Parameter 2 is always the color standard. Usually, parameter 3 is the paintline, Parameter 4 is a comment and Parameter 5 is the vehicle identification number (VID), but the user can define a different purpose for the last 3 parameters or ignore them completely.

. For example smart-chart needs the parameter value of “VehicleId”. In a bykLINK created output file it would be defined as the content of <VID> node.

```
<?xml version="1.0"?>
<byklink_service_bykmac>
  <jobdata>
    <timestamp>2016-04-06T16:11:57</timestamp>
    <VID>Vhcl48714</VID>
  </jobdata>
</byklink_service_bykmac>
```

So, the configuration needs to be like this:

A configuration window with a label 'VehicleId' and a dropdown menu. The dropdown menu is open, showing 'VID' as the selected option.

Sometimes the mapping is more complicated, because a robotic control system is not as flexible as PC software. To provide more flexibility at symbol mapping it is possible to define helper parameters to store values provisionally and use it for example as a “Rule”. (Job helpers, Point helpers) See below: Organizer.

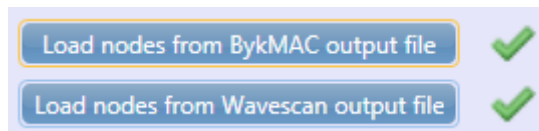
Point items

Point items work in a similar manner as Job items. They are all optional; however “Checkzone” is needed if the user works with smart-chart defined “Organizers”. There is only one predefined point parameter, the check zone. This will be used to store the information about the physical position of a single measurement. Smartchart shows them in the Data Analysis Module. (See below)

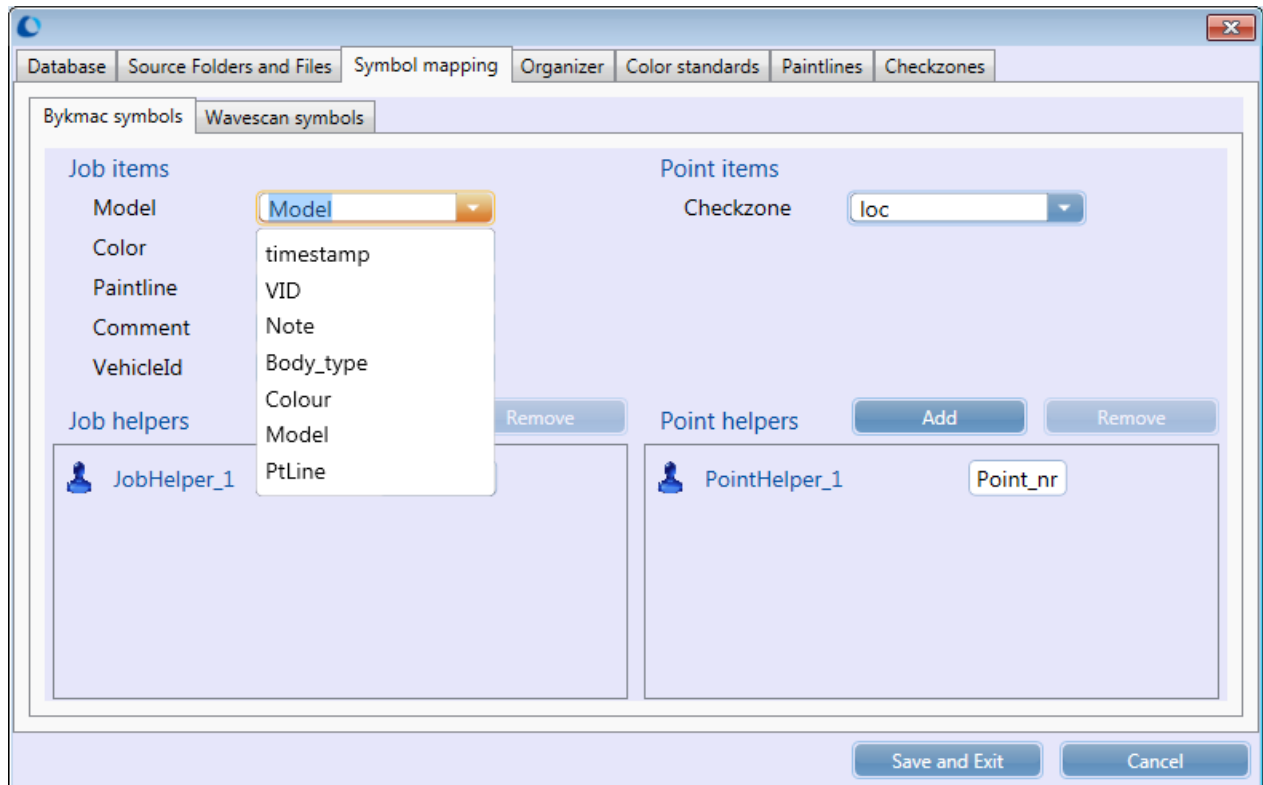
The screenshot shows the Data Analysis Module interface. The main window displays a table with columns: Model, Color Standard, Paintline, Comment, and Serial Nr. (VID). The table is currently empty. On the right side, there is a sidebar with a 'Data' tab and a 'Schem' tab. Under the 'Data' tab, there is a 'CheckZone' section with a dropdown menu labeled 'Name'. Below this, there is a 'Match to Std' section with a table of check zones. The table has four rows: Left_Back_Door, Left_Front_Door, Right_Back_Door, and Right_Front_Door. Each row has a color-coded circle and a vertical bar. Arrows point from the 'Check zones' label to the table.

Model	Color Standard	Paintline	Comment	Serial Nr. (VID)

CheckZone	Match to Std
Left_Back_Door	Red circle, Red bar
Left_Front_Door	Yellow circle, Yellow bar
Right_Back_Door	Green circle, Green bar
Right_Front_Door	Red circle, Red bar



If the “Load nodes from output file” buttons where used, the symbol mapping combo boxes are filled with the node names defined in bykLink output files. In this case, setting the right node name to a parameter is an easy job. See picture.



In the output data:

```
<?xml version="1.0"?>
<byklink_service_bykmac>
  <jobdata>
    <timestamp>2016-04-06T16:11:57</timestamp>
    <VID>Vhcl48714</VID>
    <Note>This is a comment</Note>
    <Body_type>5DR</Body_type>
    <Colour>SBlue</Colour>
    <Model>MDA123</Model>
    <PtLine>1</PtLine>
  </jobdata>
</byklink_service_bykmac>
```

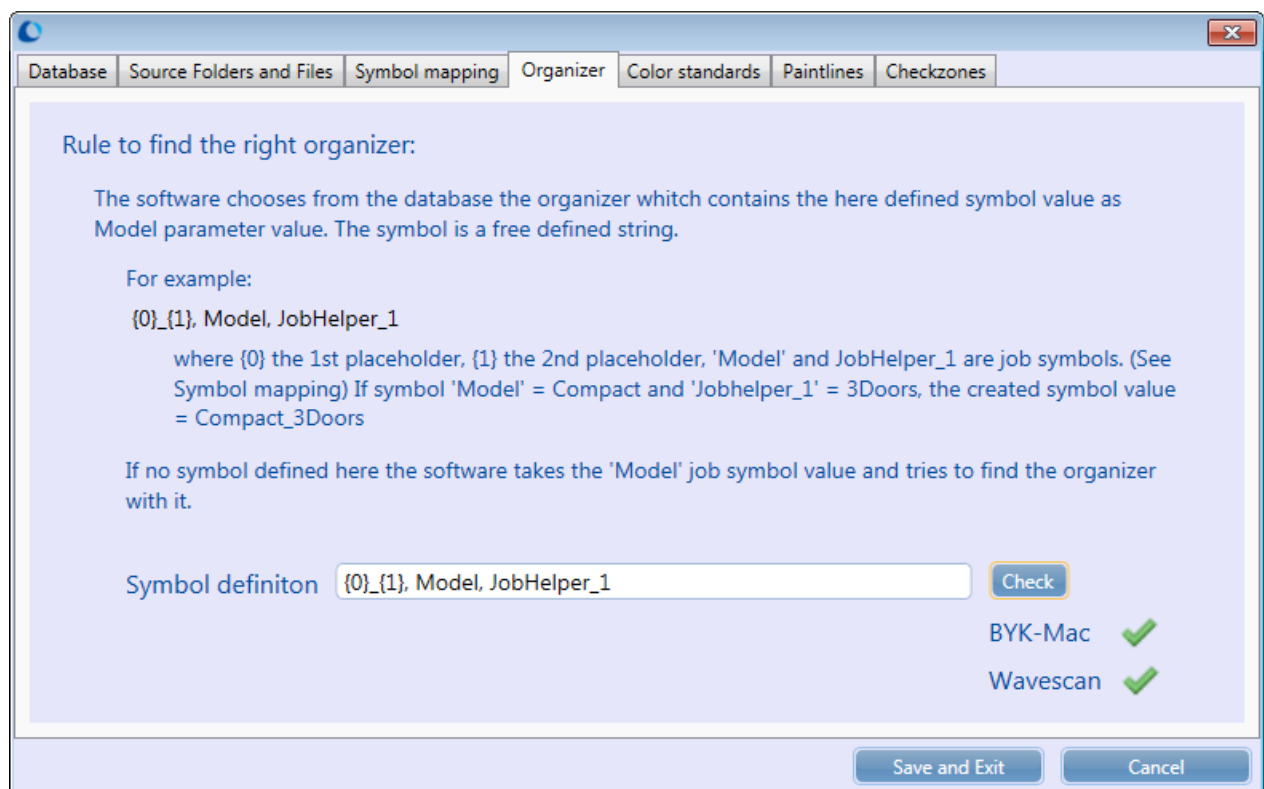
4.4 Organizer

A smart-chart organizer can have a lot of models. The RoboticDataService finds the right organizer over the model name. Smart-chart users sometimes create compound names from different parameters.

For example, a car model with name MDA123. MDA123 has three chassis variants:

4DR, 5DR, Convertible, so three model names will be defined in smart-chart:

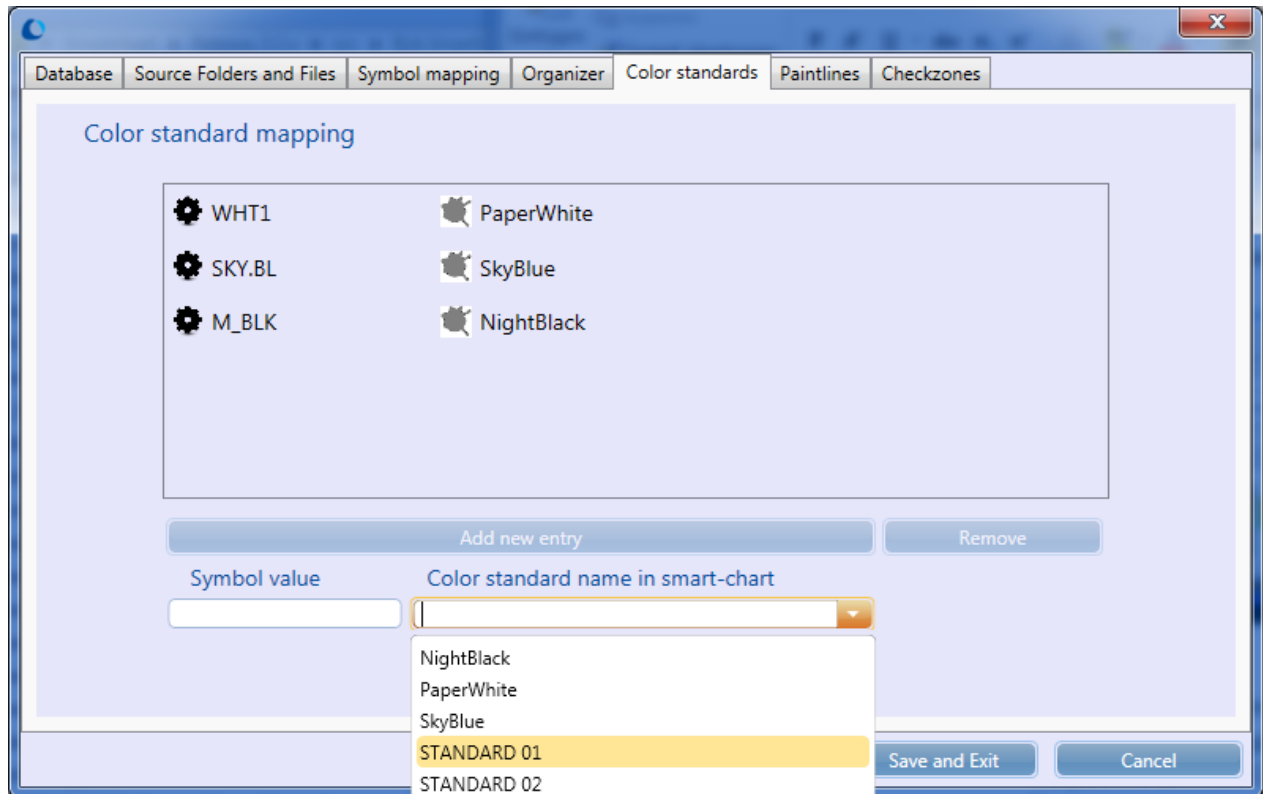
MDA123_5DR, MDA123_4DR and MDA123_Convertible. RoboticDataService.exe has to be able to create the right name to find the organizer. Because the model name and the body type are in different nodes in the output file the user has to define an “OrganizerModelRule” like this:



The entry says: get the Model symbol value, add a “_” character to it and append the value of the JobHelper_1. The result is the Model parameter value in smart-chart. It can be used for searching for the right organizer. The user can check the Rule definition with the “Check” button.

4.5 Color Standards

Smart-chart uses color standards. If the user wants to compare the data with a color standard, it has to be placed in the main database and the software has to be able to find it.



If the robotic system is not able to provide the exact color standard name, it is possible on this tab create mapping, what navigates between smart-chart standard names and the output file defined Color node values.

Example:

The service searches in the output file the node “Colour” (symbol mapping)

The value of the <colour> node is “M_BLK” in the input file.

```
<VID>Vehicle-1245</VID>  
<Colour>M_BLK</Colour>  
</jobdata>
```

The right color standard is called “NightBlack” in the main database, so the needed mapping should be this:

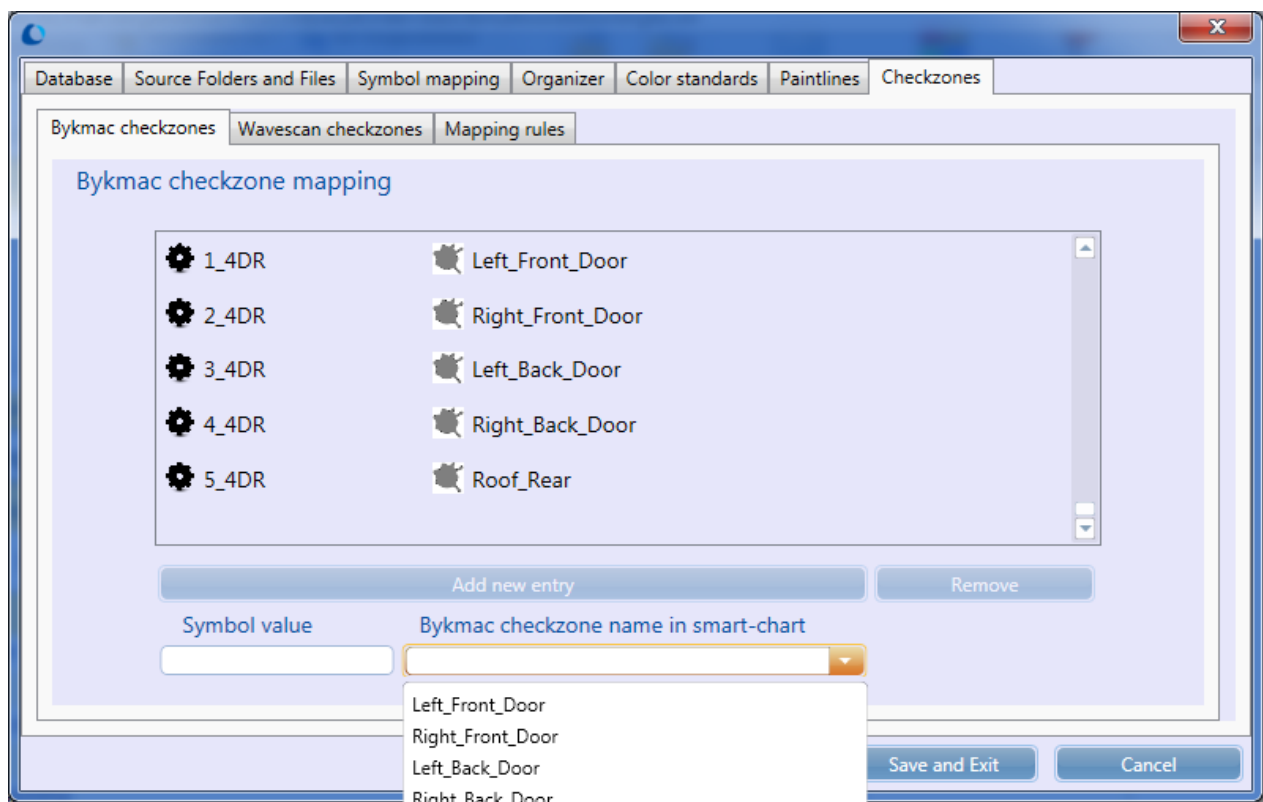


4.6 Paintlines

Paintline mapping works like the Colorstandard mapping, but it is based on the “Paintline” input parameter.

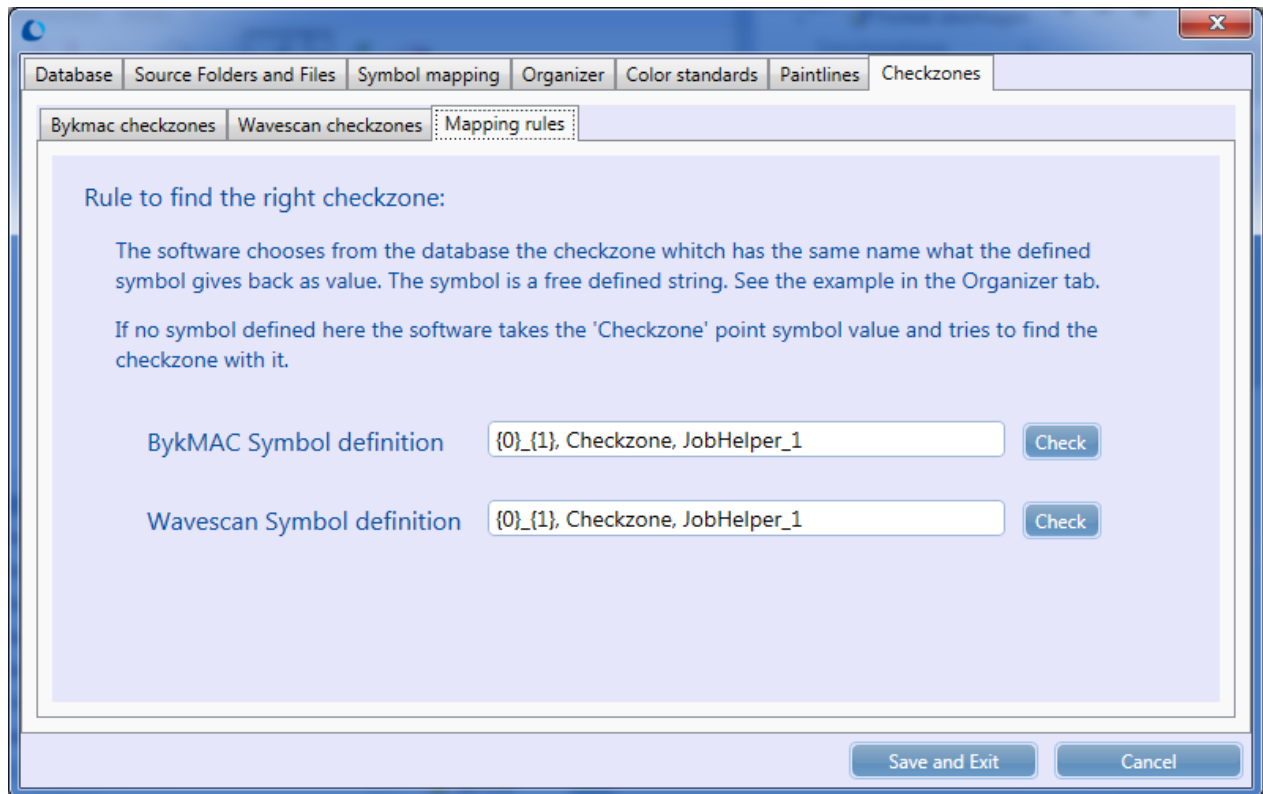
4.7 Checkzones

Users like to give meaningful names to check zones e.g. Right_Front_Door_Middle, but the SPS (PLC) has sometimes only limited character amount for data transfer. To solve the problem the user can define mapping for check zone names here.



The Checkzone mapping has the point item “Checkzone” as an input parameter. Bykmac and Wavescan devices have their own, independent check zone mapping, similar to smartchart Organizer module.

It is also possible to build a compound symbol. The method is the same as shown in OrganizerMapping.



5. Run procedure

After clicking on the “Start” button the software reads RoboticServiceConfig.xml to get the settings, localizes the input file directories (bykLINK output folders) , the database, where the data will be saved to and load all the mappings and rules.

When the initialization is finished, it opens the first text file in the incoming file buffer.

Program steps:

1. Extracting the color standard name, organizer id, the parameters like paintline name, Vehicle id, comment.
2. Check if a color standard exists in the main database with the given name.
3. When the standard is found, check if an organizer exists with the given model in it. Otherwise the data will be stored as an “absolute measurement”.
4. When the organizer is found, the service reads out the check zone names. If no organizer is identified the data will be stored as a “difference measurement”.
5. Start reading the point data from the file. From the header nodes it tries to get check zone information. After reading and mapping over existing rule it checks if the specified check zone exists in the organizer. If the result is positive it stores the data with the found check zone information, otherwise it creates a pseudo check zone with name SAMPLE X. (X will automatically incremented)
6. When the storing process is finished, the file will be closed and moved into a subdirectory called “Processed”.