

## HowTo 3 - Using the DataPool in GSE

A guide to enhance your GSE project with a DataPool

Product:	Guiliani Streaming Editor (GSE)
Release version:	2.2
Release date:	August 31, 2018

## **Disclaimer**

The information in this document is subject to change without notice and does not represent a commitment on the part of TES Electronic Solutions GmbH. The software described by this document is subject to a License Agreement and may not be copied to other media. No part of this publication may be copied, reproduced or otherwise transmitted or recorded, for any purpose, without prior written permission by TES Electronic Solutions GmbH.

“Guiliani.de”, “Guiliani”, “Guiliani Streaming Editor”, “GSE” and associated logos are (registered) trademarks of TES Electronic Solutions GmbH.

Windows, Visual Studio and Visual C++ are registered trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are the property of their respective owners and use of them does not imply any affiliation with or endorsement by them.

## **Guiliani, guiliani.de and GSE are products of**

TES Electronic Solutions GmbH  
Hanauer Landstrasse 328-330  
60314 Frankfurt am Main  
Germany

Email: [support@guiliani.de](mailto:support@guiliani.de)  
Website: <http://www.guiliani.de>  
Forum: <https://www.guiliani.de/forum/>  
HelpDesk: <https://guiliani.on.spiceworks.com/portal>

## **Table of contents**

1.	Introduction .....	5
1.1.	Assumed knowledge .....	5
1.2.	Prerequisites .....	5
1.3.	Documentation conventions .....	5
1.4.	Shortcuts .....	6
1.5.	This document's goal .....	7
2.	Adding a DataPool without programming .....	8
2.1.	Step 1: Load project "step by step" .....	8
2.2.	Step 2: Outlay Preparation .....	8
2.2.1.	Add a number to the cooking pot .....	8
2.2.2.	Add numbers to the burners .....	10
2.3.	Step 3: Connecting the wheel's value with a burner using a DataPool.....	11
2.3.1.	Using an Image Stack.....	12
2.3.2.	Using a DataPool.....	15
2.3.3.	Show the wheel's value in the burner using the DataPool.....	17
2.4.	Step 4: How to continue? .....	18
2.4.1.	Sample solution .....	18
2.4.2.	Continuing HowTos .....	18
3.	Index.....	19

## **Table of figures**

Fig. 1 This “How To” result.....	7
Fig. 2 The pot's StandardText attributes .....	9
Fig. 3 Run project - with pot number .....	10
Fig. 4 Run project - with burner numbers .....	11
Fig. 5 Added images to the stack .....	13
Fig. 6 Run project - with burner ring .....	14
Fig. 7 Select Wheel as an observer .....	15
Fig. 8 Finishing DataPool 1 .....	16

## 1. Introduction

This document explains step by step how to use a DataPool in a GSE project.

### 1.1. Assumed knowledge

- Basic handling of GSE
- It is recommended to read “How to 1 - build a project step by step”, first.
- For a better understanding on how the project works, we suggest reading “How to 2 - using Commands in GSE”, too.

### 1.2. Prerequisites


- Unpacked Guiliani SDK including GSE
- Your project “step by step” created in “HowTo 2” (or the “step by step” project inside the folder called “*How to 2 - sample solution*”)


### 1.3. Documentation conventions

Whenever you can use keys from your computer’s keyboard, these will be displayed in square brackets (e.g.,” To run your project press [Ctrl] + [r].”).

Menu commands or file path used in this document will be shown in *italic*.

Text that appears in the software on controls will be printed in **bold and blue**.

 Whenever the reader of this document has to do something in his project, the text will start with this triangle.

 Results will be shown using this arrow.

In this document, we use icons whenever we will warn the user or will give him additional or important information.



The speech bubble icon will show additional helpful information.



Whenever a text begins with an exclamation mark icon, it contains important information that is essential for the current chapter.



A warning sign icon signals serious issues and potential risks that require your full attention.

## 1.4. Shortcuts

In the documents, we often select a command from the window. These can be selected by the following short cuts, too:

<i>File</i> → <i>New Project...</i>	[Ctrl] + [Shift] + [n]
<i>File</i> → <i>New Dialog...</i>	[Ctrl] + [n]
<i>File</i> → <i>Save Project...</i>	[Ctrl] + [s]
<i>File</i> → <i>Run Simulation...</i>	[Ctrl] + [r]

## 1.5. This document's goal

At the end of this HowTo you will have learned

- how to use a DataPool
- send information between different controls
- and more

You will be able to save the heating state of a burner.  
The main dialog will look like this:

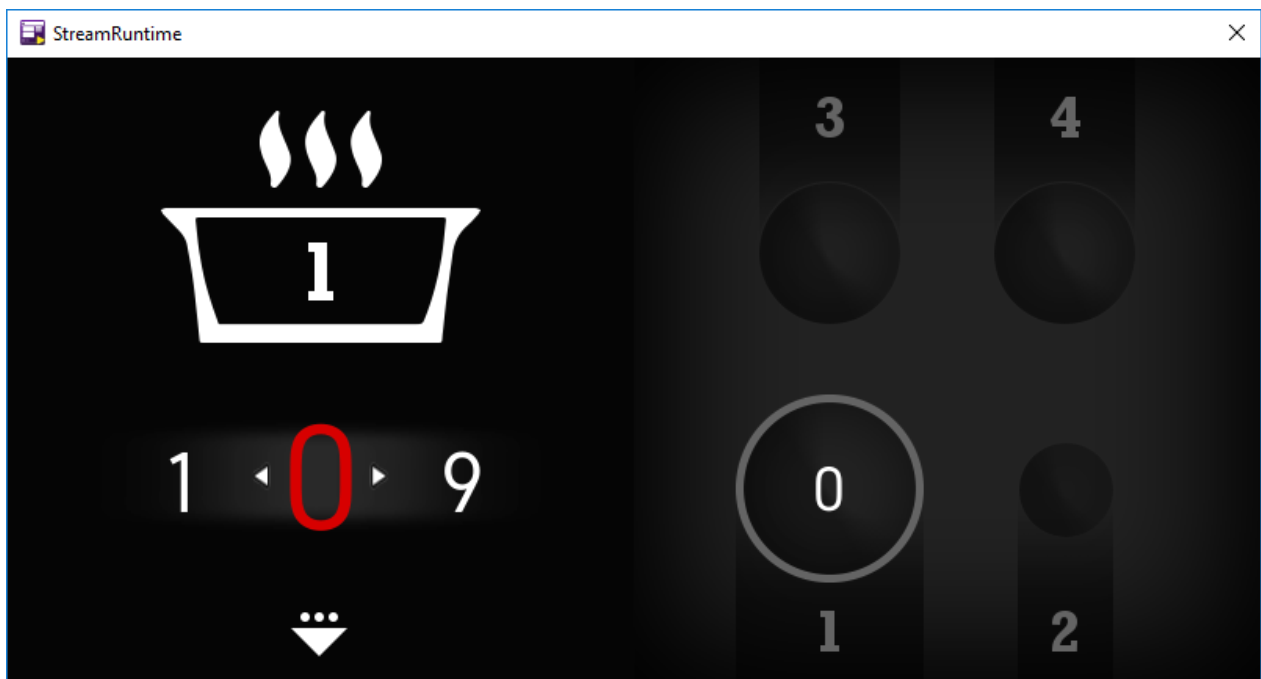


Fig. 1 This "How To" result

## 2. Adding a DataPool without programming

### 2.1. Step 1: Load project “step by step”

- ▶ Start the GSE and load the created Step by Step project from HowTo 2 (step\_by\_step.gpr).

### 2.2. Step 2: Outlay Preparation

Before we include the DataPool, we will have to add some images to our project. We need to add a number to the pot and we have to prettify the chosen burner. For this HowTo we will set burner 1, only. So we have to add a “1” into the pot and prettify burner 1.

#### 2.2.1. Add a number to the cooking pot

- ▶ Add a text field from the “**Controls**” window to the **main** dialog.
- ▶ Set the following text field’s attributes:

<b>XPos</b>	175
<b>YPos</b>	107
<b>Width</b>	50
<b>Height</b>	65

Now we have to make some changes to the text.

- ▶ Look for the **TextField**’s attribute **StandardText**.
- ➔ There you will find the corresponding attributes list.
- ▶ Here, make sure that **TextWidth** and **TextHeight** have the same values as **Width** and **Height**.
- ▶ The color for the text should be white, so you have to set all attributes beginning with “**TextColor**” to white (0xFFFFFFFF).
- ▶ Replace “**text field**” with “1” (burner number) in the text field behind the attribute “**Text**”.

The text is too small, so we have to change the text font.

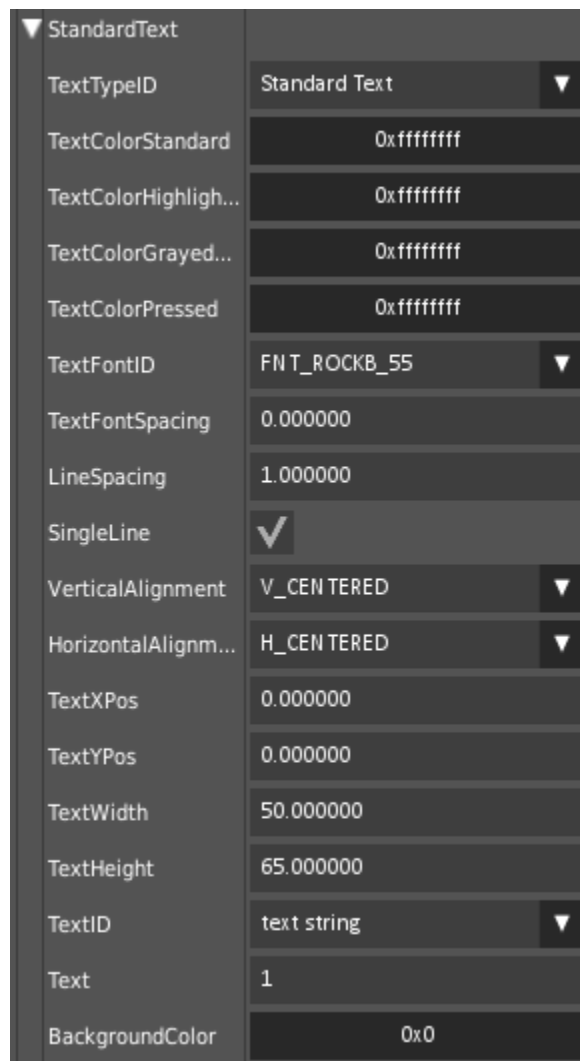
- ▶ Inside the folder “*How to 3 - resources*” of your SDK’s documentation folder, you will find the font “ROCKB.ttf”. Use this font with a size of 55 pixels.



If you have problems at this point, please take a look into the document “How to 1 - build a project step by step” and read through chapter 2.12. Setting a control’s font.



➔ The attributes should be set like this:



**Fig. 2** The pot's StandardText attributes

▶ Save your project and run it.

➔ The project should look like this:

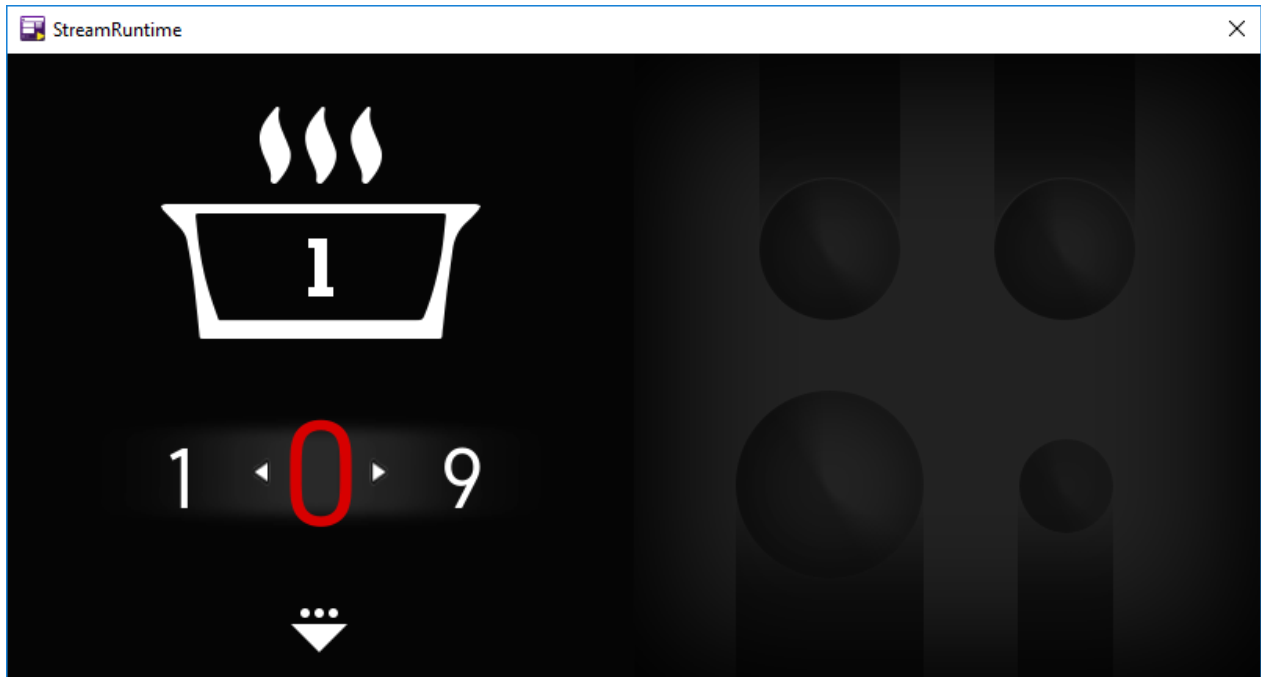


Fig. 3 Run project - with pot number

## 2.2.2. Add numbers to the burners

Before the value we select using the wheel can be displayed inside the appropriate burner, we have to number them.

- ▶ Add four text fields to the main dialog and set their attributes as stated below.

Attribute	Burner 1	Burner 2	Burner 3	Burner 4
<b>XPos</b>	505	655	505	655
<b>YPos</b>	348	348	20	20
<b>ObjectID</b>	<b>AID_ TEXTFIELD_2</b>	<b>AID_ TEXTFIELD_3</b>	<b>AID_ TEXTFIELD_4</b>	<b>AID_ TEXTFIELD_5</b>

! For all four text fields make sure, that the **TextWidth** and **TextHeight** attributes are similar to their **Width** and **Height** attributes (all 40).

- ▶ Additionally all attributes beginning with “**TextColor**” should be set to 0XFF646464 and the **TextFontID** to FNT\_ROCKB\_38.

! The chosen font (called ROCKB.ttf) can be found inside the folder “How to 3 - resources” of your SDK’s documentation folder. Use this font with a size of 38 pixels.

- ▶ The text field’s attribute **Text** should contain the burner’s number.

▶ Save your project and run it.

➔ The project should look like this:

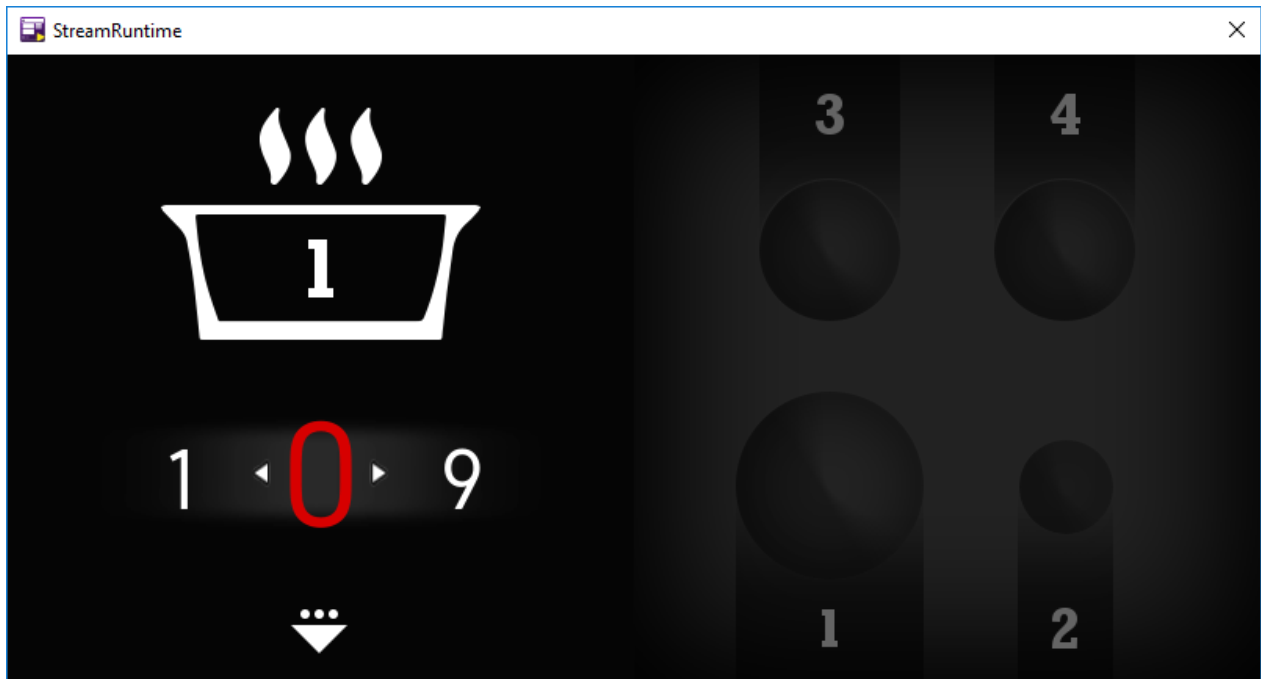


Fig. 4 Run project - with burner numbers

### 2.3. Step 3: Connecting the wheel's value with a burner using a DataPool

What should happen if we select a value with the wheel?

The appropriate burner (shown in the pot above the wheel) should show the value we have selected using the wheel. The burner's ring should be "glow" according to the heating intensity (shown as the number inside the wheel).

So according to the chosen value a different image should be shown at the appropriate burner. That means, we should have a stack of images and should show the appropriate burner image with the "right" red glowing ring and the chosen stage from the wheel.

And that is the solution: every time the wheel's value changes, the value should be stored. The burner should be looking for a change of the stored value and should then show the appropriate image.

We can realize the needed stack of images with the control called "[image stack](#)".

## 2.3.1. Using an Image Stack

- ▶ Add an “**image stack**” control to the main window.
- ▶ Set the image stack’s attributes **MaxScale**, **MinScale** and **StaticScale** to 1 and **XPos** to 465, **YPos** to 215 and **Width** and **Height** to 120.
- ▶ Scroll down the attribute list to **Image**. Here you see the corresponding attributes for the handling of the images to be shown.
- ▶ Click onto the button behind “**NumberOfImages**”. The text will change from “**0 (click to add more)**” to “**1 (click to add more)**” and another attribute appears, called “**ImageID**”.
- ▶ Click onto the button behind “**ImageID**”. Now you can select an image. Add the image “1\_big\_0.png” from inside the folder “*How to 3 – resources/turner1*” of your SDK’s documentation folder.
- ▶ Repeat this nine times, adding the images “1\_big\_1.png” to “1\_big\_9.png”.

➔ The attributes should be set like this:



Fig. 5 Added images to the stack

▶ Save your project and run it.

➔ The project should look like this:

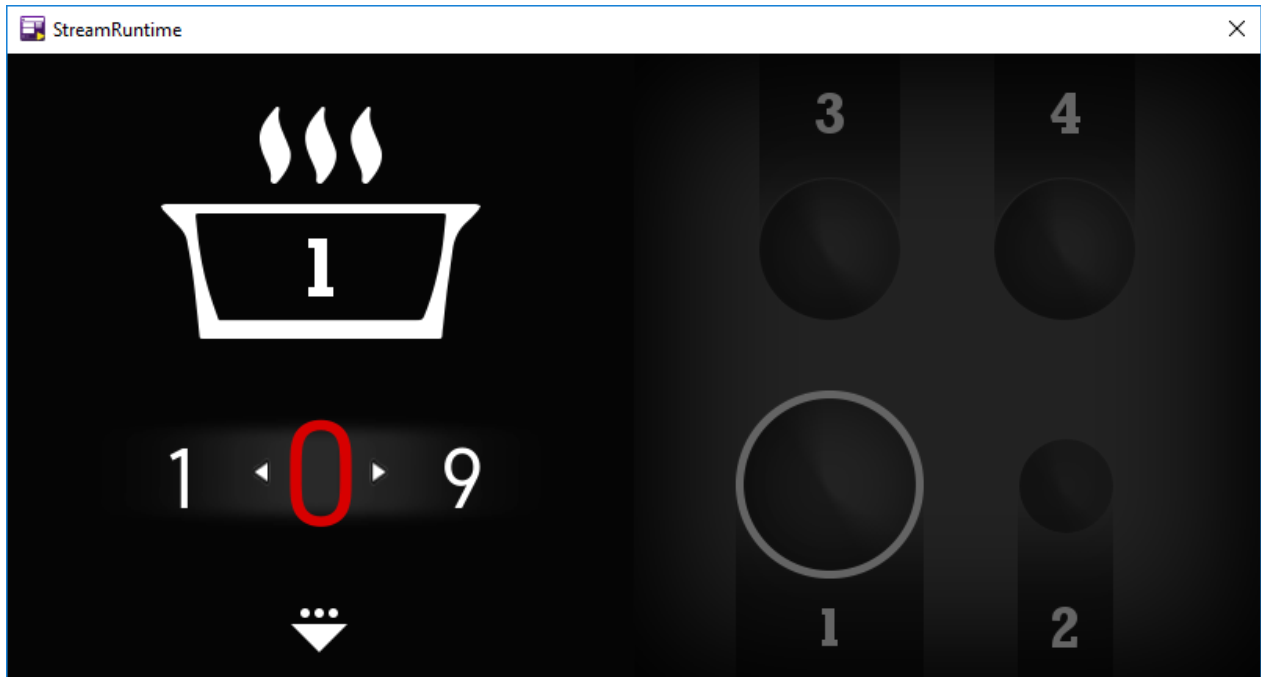


Fig. 6 Run project - with burner ring

## 2.3.2. Using a DataPool

To provide the number selected in the carousel, we need a DataPool. The creation and use of a DataPool is very simple. Just follow the next steps.

- ▶ Select *Manage DataPool* from the menu *Resources*.
- ▶ Press the button “**Add new Entry**”.
- ▶ Select “**AID\_WHEEL\_1**” from the pull down menu.

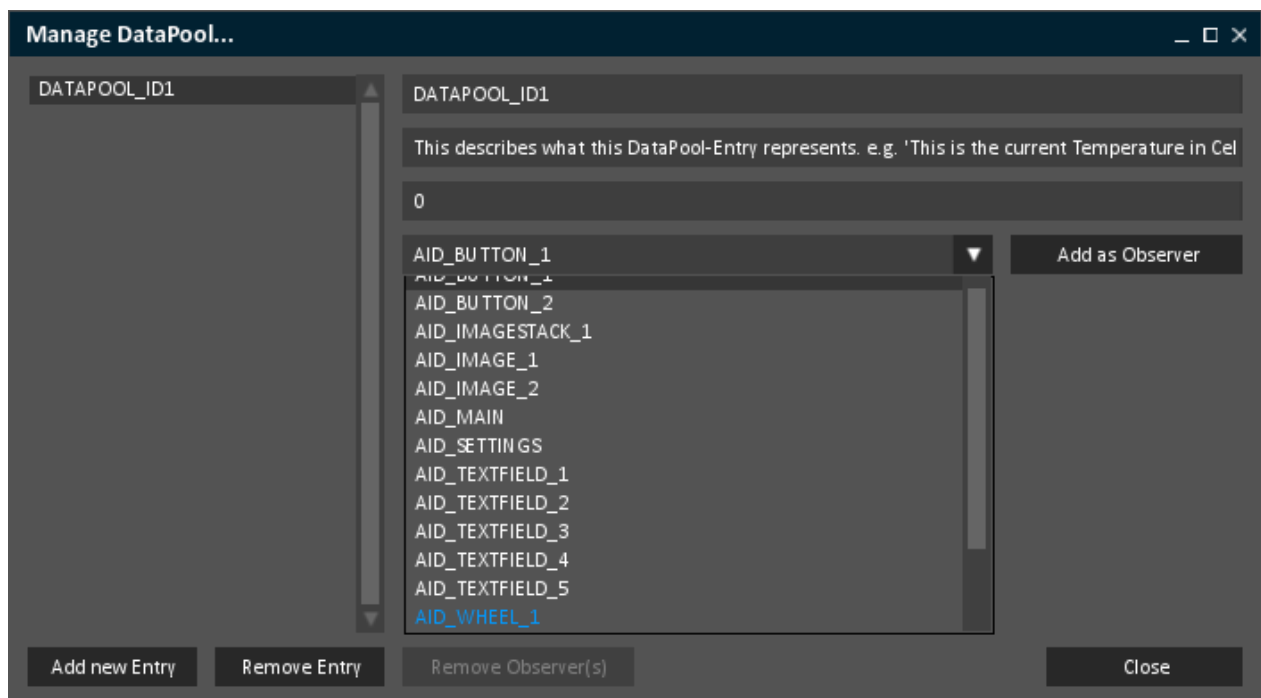
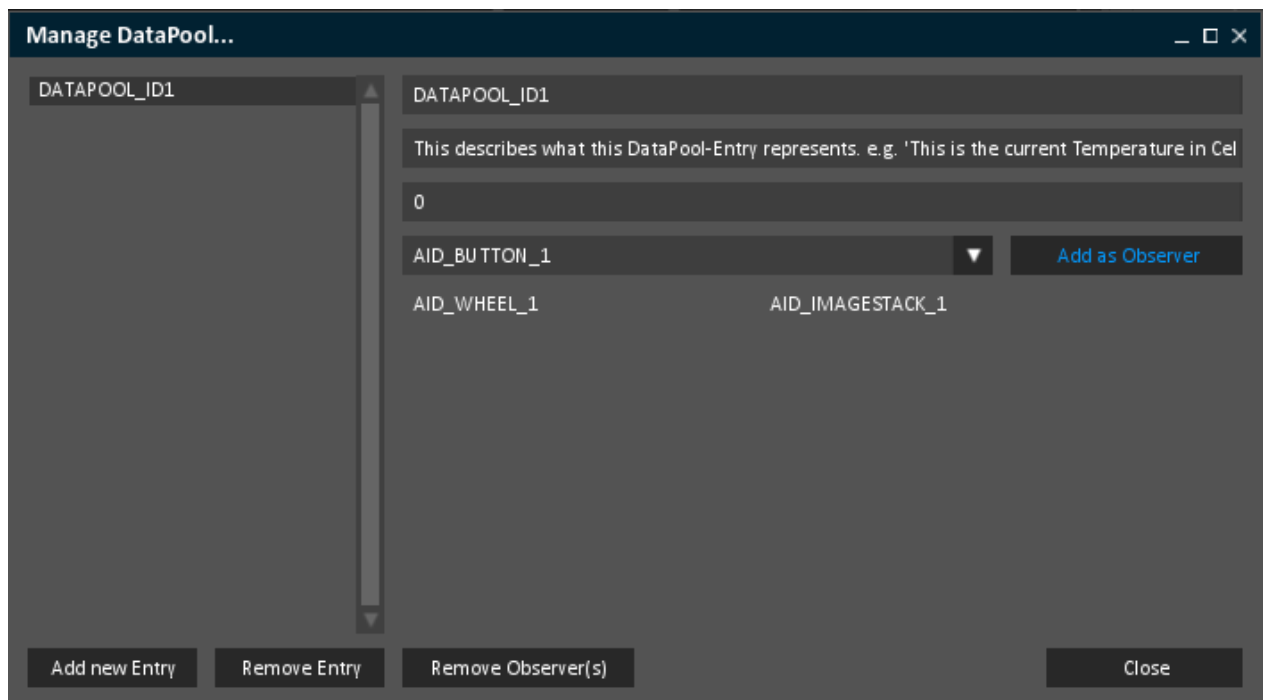


Fig. 7 Select Wheel as an observer

- ▶ Press the button “**Add as Observer**”.
- ▶ Select “**AID\_IMAGESTACK\_1**” from the pull down menu.
- ▶ Again press the button “**Add as Observer**”.

Now we have two objects that use the **DATAPOOL\_ID1** for exchanging information.



**Fig. 8 Finishing DataPool 1**

- ▶ Save your project and run it.
- ▶ Change the value of the wheel and look what happens with burner 1.
- ➔ Now the burner ring will be shown in a dark or light red, according to the value we choose in the wheel (or gray, if the value is 0).

We have to create a text field and connect it with the DataPool, too. So it will be updated automatically.

This will be done in the next chapter.



## 2.3.3. Show the wheel's value in the burner using the DataPool

First, we have to add a text field into the burner 1.

- ▶ Add a **Textfield** to the main dialog.
- ▶ Set **XPos** to 499, **YPos** to 252, **Width** to 50 and **Height** to 50.



Make sure, that the **TextWidth** and **TextHeight** attributes are similar to the **Width** and **Height** attributes (all 50).

- ▶ The color for the text should be white, so you have to set all attributes beginning with “**TextColor**” to white (0xFFFFFFFF).
- ▶ Replace “**text field**” with “0” in the text field behind the attribute “**Text**”.
- ▶ The text is too small, so we have to change the text font to “Gidole-Regular” 40.



If you have problems at this point, please take a look into the document “How to 1 - build a project step by step” and read through chapter 2.12. Setting a control's font.

- ▶ Select *Manage DataPool* from the menu *Resources*.
- ▶ Select “**AID\_TEXTFIELD\_6**” from the pull down menu.
- ▶ Press the button “**Add as Observer**”.
- ▶ Save your project and run it.
- ▶ Change the value of the wheel and look what happens with burner 1.

The chosen value from the wheel will be displayed in the burner and the burner's ring will be displayed according to the value.  
And all this without programming!

## 2.4. Step 4: How to continue?

### 2.4.1. Sample solution

Hopefully you have been successful in adding a DataPool to your project, which has been developed during the HowTos 1 and 2.

If you encountered problems or wish to have the solution without creating the project on your own, we have added the sample solution into the folder called “*How to 3 - sample solution*” inside the documentation folder. Here you will find the GSE Project (step\_by\_step.gpr).

All resources you need for this HowTo will be found in the folder “*How to 3 - resources*”.



For Windows user inside the folder “temp” there is an executable StreamRuntime.exe.

### 2.4.2. Continuing HowTos

You will find an overview of continuing HowTos in the document “HowTo 0 - an overview of building GSE projects”.

Don't forget to visit our homepage [www.guiliani.de](http://www.guiliani.de) to get more information, demos, help, videos and the latest news about guiliani and GSE.

## 3. Index

---

### A

Adding a DataPool · 8  
Assumed knowledge · 5

---

### C

Connecting a DataPool · 11  
Continuing How Tos · 18

---

### D

DataPool  
  Adding · 8  
  Connecting · 11  
  Using · 15  
Documentation conventions · 5

---

### H

How to continue? · 18

---

### I

Image Stack  
  Using · 12

---

### L

Load project · 8

---

### P

Preparation · 8  
Prerequisites · 5  
Project  
  Load · 8

---

### S

Sample solution · 18  
Short cuts · 6

---

### U

Using a DataPool · 15  
Using an Image Stack · 12