MSControl Suite

Ronny Friedrich

May/25/2017

# MSControl

## Qtegra Software

### Remote Control

In order to connect MSControl to the Thermos Qtegra Software, the RemoteControlServer script is being used.

Currently Qtegra does NOT load this script automatically, meaning that after each restart of Qtegra the script has to be loaded and run.

Go to “Status Panel” in Qtegra and load the script “LDEO\_RemoteControlServer.cs” from “C:\ProgramData\Thermo\Qtegar\\_Application Data\PluginData\HelixMC\Scripts”.

Find the newest version of the RemoteControlServer under <https://github.com/NMGRL/qtegra>. Be careful since the script does not support the HelixMC but only the Argus used in James Ross’ lab. One has to adjust the script to control the HelixMC and to the specific detector configuration

### Problems with Qtegra

When changing the position of the magnet using “SetMagnet” in Qtegra, the RemoteControlServer does not return any detector readings anymore.

The “Intesity Display” in Qtegra MUST show the same masses as set in the Cupconfiguration. E.g. if the Cupconfiguration for Helium shows that the AX mass is set to 4, the mass of AX MUST be set to 4.

Sometimes MSControl is not able to get any detector readings even though everything is set up correctly. In that case restart Qtegra and/or the whole PC.

## Before you start MSControl

* Make sure all the instruments are connected and turned on
* Makre sure Qtegra is running and the RemoteControlServer Script is running

## Before you do an Analysis

* Makre sure Qtegra is running and the RemoteControlServer Script is running
* In Qtegra: Make sure the correct CupConfiguration is selected
* In Qtegra: Make sure that all the all detectors that should be used are activated (Ribbon bar: button “CDD Enabled” is green, Scan settings: check mark next to the detectors)
* In Qtegra: Make sure the mass (as visible in the intensity display) of the detectors are set to the default value as given in the CupConfiguration

## Running a Script

### Individual Script

Select the menu item “RunScripts” that will show all the available scripts in the script folder.

Click on a script in oder to run it. The script does not start automatically.

### Multiple Scripts/SampleLists

In order to run a set of scripts (set of analysis) automatically. Choose the menu item “SampleList” and select a set of scripts that will run automatically.

## Script Commands

Basic Syntax: command(param1;param2,...) ->plot\_name

commands without parameters can end with () but don't have to, e.g. close\_window = close\_window()

### Script related

|  |  |
| --- | --- |
| // | Lines starting with // are comments and will be ignored |
| wait(number) | Wait *number* of seconds |
| message(text) | Displays *text* in the message window (right hand corner of the main window) |
| message(clear) | Clears the message window |
| alert(text) | Displays a dialog with *text* and waits for confirmation |
| open\_script(rel.path\scriptname) | Opens a new script with name *scriptname.* The script must be in thebase ScriptFolder “RunScripts”.  If the script is in a subfolder inside of “RunScripts” add the relative path to the scriptname.  **Example:**  Open\_script(test.txt) opens the script “test.txt” that is located in “RunScripts\test.txt”  Open\_script(test\test.txt) opens the script “test.txt” that is located in “RunScripts\test\test.txt” |
| stop\_and\_wait(name) | Creates a new flag or re-uses a flag named *name* and waits until the command “proceed(name)” is processed. If the command “proceed” was processed already, “stop\_and\_wait” proceeds immediately |
| proceed(name) | Tells a “stop\_and\_wait” command to proceed. *Name* is the name of the flag. |
| close\_window | Closes the current script window |

### Switches

|  |  |
| --- | --- |
| open(switch\_name) | Opens switch named *switch\_name* |
| close(switch\_name) | Closes switch named *switch\_name* |

### Temperature controllers

|  |  |
| --- | --- |
| set\_temp(name;temp;range) | Sets the temperature of the controller *name* to *temp* and waits until the actual temperature is within *range* |

### Plots and peak centering

Displaying data and processing displayed data

|  |  |
| --- | --- |
| plot\_open(plot\_name) | Opens a new plot window named *plot\_name* |
| plot\_xlabel(plot\_name; label) | Sets the xlabel of the plot *plot\_name* to *label* |
| plot\_ylabel(plot\_name;label) | Sets the ylabel of the plot *plot\_name* to *label* |
| plot\_close(plot\_name) | Closesa new plot window named *plot\_name* |
| plot\_find\_single\_peak(plot\_name; MFT\_name; isotope\_name(MFT); threshold) | Finds the center of a single peak by using the data displayed in the plot *plot\_name*. If a peak was found the result is written into the magfield table (MFT) named *MFT\_name* using the isotope *isotope\_name*. That isotope must already exist in given MFT. If the signal is below *Threshold* no peak centering will be performed |
| plot\_find\_double\_peak(plot\_name; MFT\_name; isotope\_name(MFT); threshold; offset) | Finds the center of a double peak by using the data displayed in the plot *plot\_name*. If a peak was found the result is written into the magfield table (MFT) named *MFT\_name* using the isotope *isotope\_name*. That isotope must already exist in given MFT. If the signal is below *Threshold* no peak centering will be performed.  If only one peak is found offset is being used to calculate the peak position of the desired isotope, e.g. a value of -0.01 is equal to a peak that is 0.01 units below (lower mass) the found peak. Respectively, a positive number represents a peak that is above (higher mass) the found peak. |
| plot\_fit\_data(plot\_name; fittype[linear, 2nd, 3rd]) | Fits the data displayed in the plot *plot\_name*. using fittype. Fitype can have the values: linear, 2nd or 3rd.  Linear = linear fit  2nd = polynomial fit 2nd order  3rd = polynomial fit 3rd order |

### Analysis

|  |  |
| --- | --- |
| analysis\_session\_new(AnalysisID; AnalysisName; AnalysisType) | Creates a new AnalysisSession. AnalysisID can be used to identify the analysis (e.g. sample ID). AnalysisName can be used to store the name of the sample of standard. AnalysisType is used to identify the type of analysis (sample, air, blank,…).  If those parameters are not given, a dialog appears and asks for the parameters.  If a SampleList is used to automatically run samples those information are provided by the SampleList. The SampleList will overwrite any explicitly given parameter. |
| analysis\_session\_add\_switch\_counter(switch\_name) | The activation of all switches is monitored and counted. Using this command saves the counter of the given switch. |
| analysis\_session\_add\_standard\_data(standard\_name) | Adds the standard information of the standard *standard\_name* to the AnalysisSession that is being processed by the script (or subscript). |
| analysis\_session\_add\_pipette\_data(pipette\_name; number\_of\_shots) | Adds the pipette information of the pipette *pipette\_name* to the AnalysisSession that is being processed by the script (or subscript). *Number\_of\_shots* is the number of used pipettes for this analysis |
| analysis\_session\_dump | Writes all Analysis data into the textfile or database (depending on the settings). At this point only text files are supported. |
| analysis\_session\_close | Closes the AnalysisSession and removes the AnalysisObject from Memory. Should be performed after data acquicition is finished to free up memory. |

### Data Aquicition from Instruments

|  |  |
| --- | --- |
| set\_time\_zero(instrument) | Sets the value “time zero” of the *instrument*. Time zero is typically used in spectrometers to mark the inlet of the gas. The command is ignored if the given instrument doesn’t support this function. |
| set\_magnet(instrument; value) <-- for Qtegra value is DAC | Sets the *magnet* of the *instrument* to value. The units of *value* are the units that the *instrument* used to control the magnet, e.g. Thermos Qtegra uses magnet voltage. |
| signal\_vs\_magnet(instrument; magnet\_value; detector; integrtime; settling\_time) -> plotname | Sets the magnet of *instrument* to *magnet\_value*, waits *settling\_time* before starting the integration of the signal for *integrtime*. After that the measured signal of the *detector* is returned and plotted versus the *magent\_value* in plot *plot\_name*.  Qtegra supports only certain values of *integrtime*. |
| signal\_vs\_time(instrument;magnet\_value;detector;integrtime;settling\_time) -> plotname | Sets the magnet of *instrument* to *magnet\_value*, waits *settling\_time* before starting the integration of the signal for *integrtime*. After that the measured signal of the *detector* is returned and plotted versus time in plot *plot\_name*.  Qtegra supports only certain values of *integrtime*. |
| protect\_detector(instrument;detector;[0,1]) | Protects (*1*) or unprotects (*0*) the *detector* of the *instrument*  **Example:**  protect\_detector(qtegra,H1\_CDD,1) |
| set\_cupconfiguration(instrument;cupconfig\_name;subcupconfig\_name) | Changes the cup and subcup configuration of the *instrument*. Make sure the cup and subcup configurations actually exsist! |
| set\_detector\_deflection(instrument;detector;voltage) | Sets the voltage of the deflection of the detector |
| set\_detector\_supply(instrument;detector;voltage) | Sets the supply voltage of the detector |
| systemdevice\_vs\_time(SystemDeviceName) -> plotname | Returns the time and the value of *SystemDevice* (pressure sensor, temp sensor etc etc) and plots the data in a plot with the name *plotname*  The data are stored during an analysis session and saved with the whole session.  A list of available SystemDevices is given in MSControl SystemDevice.ini |

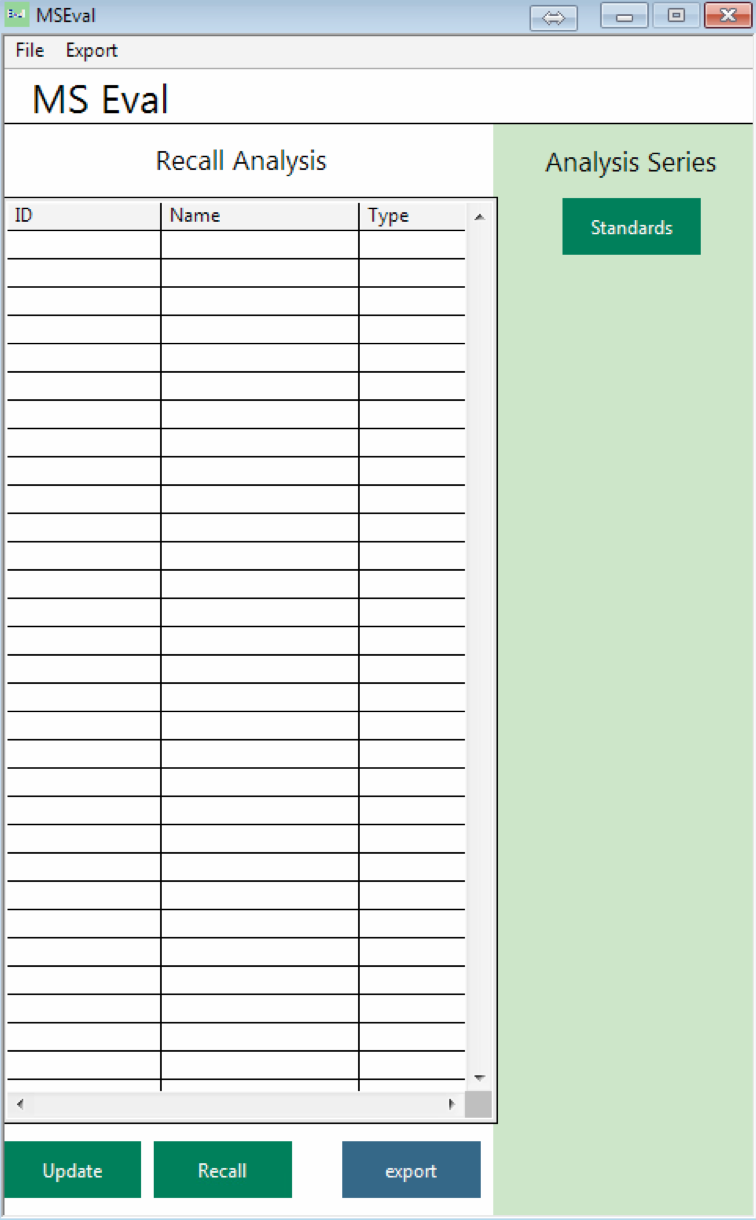
### Sample List

|  |  |
| --- | --- |
| SampleList\_proceed | If a SampleList is used to run multiple samples automatically, the sample list starts running the next sample when this command is processed.  The command is being ignored if samples are run without a SampleList. |

## General script structure



# MSEval



## Setup Dialog

Data Source: selects the type of data source. At this point only a “folder” with text data is supported

Source Folder: Folder where the analytical data are stored as text files

Export Folder: Folder to store exported data

## Display List of Analysis

Click on “Update” to display all analytical data that are currently present in the data source folder.

## Recalling Analysis Data

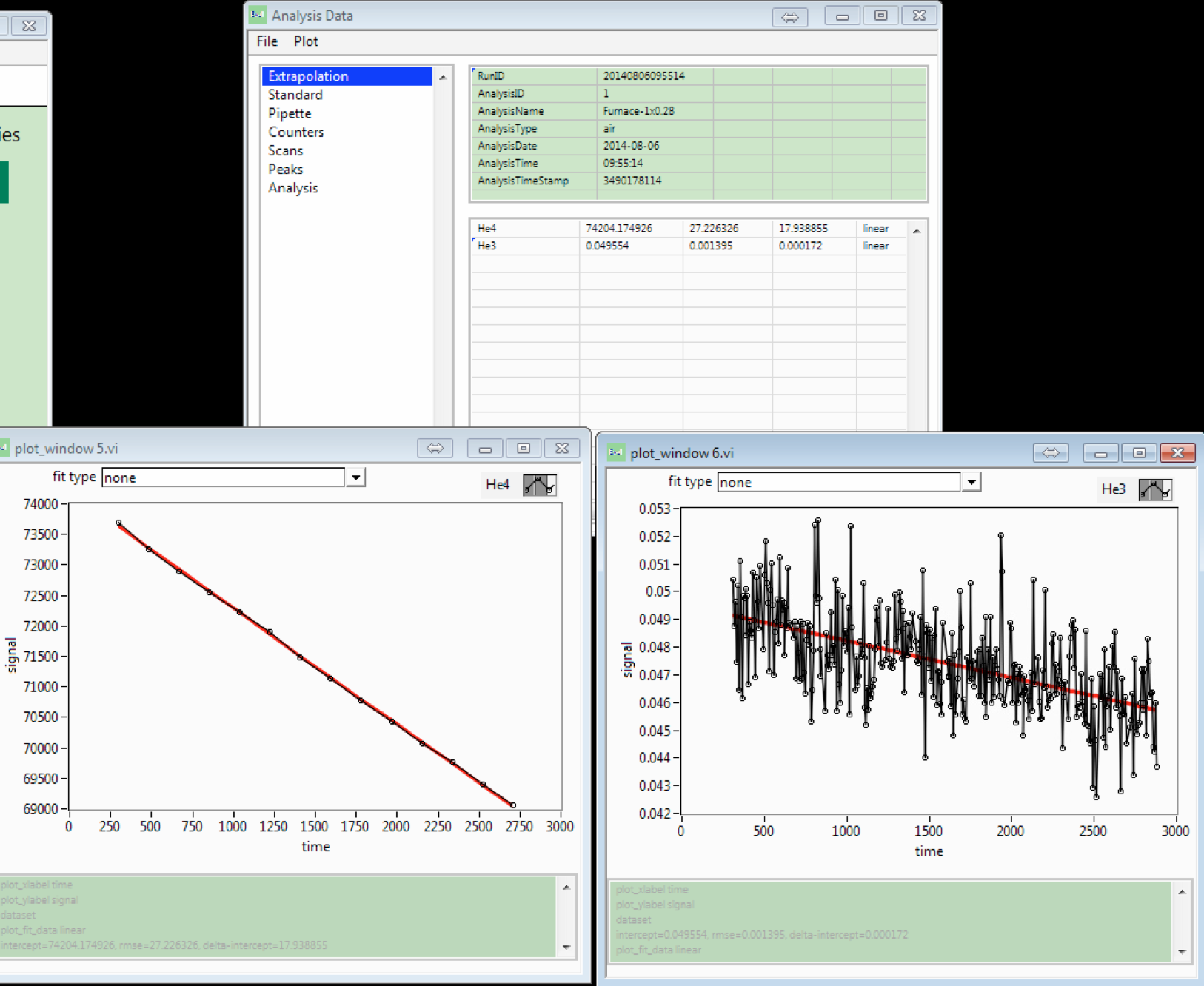
Double click on an analysis in the list or click on “recall”. A window displaying the analytical results will open.

## Exporting Data

At this point the export function is meant to export the final results of all selected analysis into one file in order to process it in excel or other software.

Select the analysis in the list and click “export” to open the export dialog.

## Displaying and editing analytical results



Recalling an analysis opens a window that shows all data of that analysis in detail.

Selecting “Plot->isotope name” plots the selected data of the isotope. The same type of fit is applied to the data as was applied during the analysis.

By clicking on a data point the point is removed from the data and the fit is updated to reflect the new data.

If the type of fit was changed or data points where removed, select File->Save in order to save the new extrapolated results for each isotope.

# Qtegra

## Saving new tune parameters

When the tuning has changed save the settings in a tune settings file by going to Setting -> button: new or button: save (which will over right the settings of the selected tune file). The new tune settings will also have to be saved to the CupConfiguration by pressing the button “Pass Source Settings” in the ribbon bar.

## Saving collector settings

If any collector settings where changed, those new settings have to be saved to the CupConfiguration by pressing the button “Pass Collector Settings” in the ribbon bar