

## Gamry's Sequence Wizard – An easy way to create complex test sequences

### Purpose of This Note

This application note discusses Gamry's Sequence Wizard. It describes its setup and shows by means of practical examples how to work with the Sequence Wizard and how to easily build measurement sequences beginning from simple structures right up to complex measurement procedures.

### Introduction

For more flexibility, Gamry Instruments enables the user several ways to perform electrochemical measurements.

The user can choose from a variety of pre-built techniques which are subordinated in specific groups in the Framework software. In addition, Gamry's concept of "Open Source Scripting" allows the user to modify the experiment's script source and to customize experiments.

If you need more information on the Explain script language, see Gamry's Application Note section at [www.gamry.com](http://www.gamry.com).

For combining single experiments, the Sequence Wizard can be used which is the subject of this application note. The Sequence Wizard has the advantage that complex measurement sequences can be easily created by using a drag and drop approach. In this way customized experiments can be executed consecutively without interfering disruptions.

### Gamry's Sequence Wizard

#### Setup

Nearly all experiments of Gamry's Framework can be used in the Sequence Wizard. Additionally, different functions are included with which the settings of the potentiostat, the electrode, and further devices such as the Auxiliary Electrometer and the Electrochemical Multiplexer ECM8 can be controlled.

Generally, the following categories are preinstalled on the Sequence Wizard:

- Hardware
- Sequence Utilities
- Electrode
- General

In addition, optional software packages for different experiments can be installed:

- DC105
- EN120
- EFM140
- PHE200
- PV220
- EIS300
- PWR800
- Aux Electrometer

The Sequence Wizard can be opened in the Experiment menu of Gamry's Framework. Its setup is shown in Figure 1.

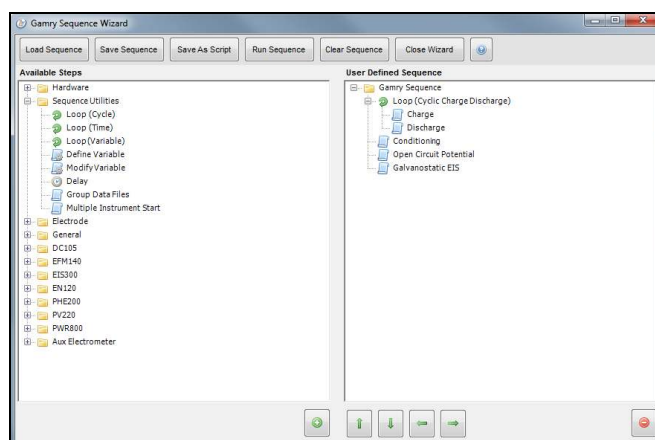


Figure 1 – Setup of Gamry's Sequence Wizard.

The setup consists of two columns. On the left side are all usable experiments and tools, divided in subfolders. The right column is used for the own sequence.

To place single functions in a sequence, they can be easily dragged and dropped from the left to the right column and placed at the desired position.


Another way is using the buttons on the bottom of the window. The plus (+) and minus (-) sign add and remove single steps. Arrow symbols are used to move single steps upwards (↑) and downwards (↓) in a sequence. The indent can be changed by left (←) and right (→) arrows. In this way, single steps can be assigned to loops which will be discussed later.

By double-clicking on a single step, a setup window opens where all important measurement parameters can be customized.

A sequence can be saved with **Save Sequence**. The generated file has the ending \*.CSequence and can be modified again in the Sequence Wizard. With **Save As Script** a sequence can be saved as an Explain script. An experiment can be started without opening the Sequence Wizard again.

The sequence is started with **Run Sequence**. From top to bottom, each step in the sequence is passed through sequentially. If not otherwise specified, measurements are automatically performed with the first activated potentiostat.

The following sections will go more into detail with functions that are indispensable for building complex sequences. These tools can be found in the category "Sequence Utilities".

For additional information about Gamry's Sequence Wizard and its functions use the help menu  at the top of the setup window.

## Sequence Utilities

### Delay


A sequence can be paused using the function **Delay** represented by a clock  symbol. It is placed between those experiments where a pause is required. During this period the cell is normally turned off. The sequence continues with the next step when the delay is finished.

Figure 2 shows the setup of the *Delay* function. The delay time and the corresponding unit can be set. In this example a sequence would pause for 60 minutes.

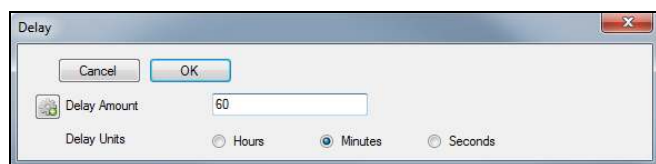


Figure 2 – Setup for the function *Delay*.

### Creating loops


One of the most important tools for building sequences is the loop. In the Sequence Wizard, single measurements are executed consecutively. To re-run a single experiment or multi-steps several times, loops are indispensable. They can simplify the complexity of a sequence drastically. Loops in the Sequence Wizard have a green circular arrow  symbol.

Figure 3 shows two examples to perform a sequence of eight consecutive electrochemical impedance spectroscopy (EIS) measurements in potentiostatic mode. The right sequence is expressed with a loop and the left without one.

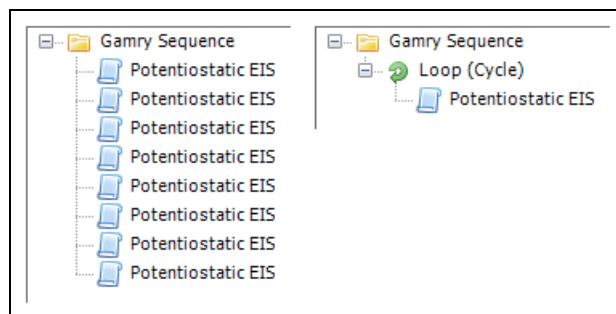


Figure 3 – Examples for a setup of the same sequence without (left) and with a loop (right).

Note that all parts that belong to a loop are indented. Indents can be changed using the left and right arrows located at the bottom of the Sequence Wizard setup.

The example of Figure 3 shows that instead of eight steps only two are needed with a loop to execute this sequence. Using loops pays off more with increasing complexity of a sequence.

In addition, parameters and the file name have to be customized for each single measurement if no loop is used. With a loop, parameters have to be set only once.

Both sequences shown in Figure 3 generate eight files for each potentiostatic EIS experiment. The loop number is automatically added to each file name if loops are used.

The Sequence Wizard has five different types of loops:

- **Loop (Cycle):** Steps are repeated for a user defined number of cycles.
- **Loop (Time):** Steps are repeated for a user defined time.
- **Loop (Variable):** Loops continue until a defined variable meets a final criterion.
- **Loop (Mux Channel):** Loops are defined by the first and the last channel of a Multiplexer.
- **Loop (Cyclic Charge Discharge):** Cycle number and end criteria define the period of cyclic charge discharge (CCD) tests.

Figure 4 shows an example how to use loops within other loops.

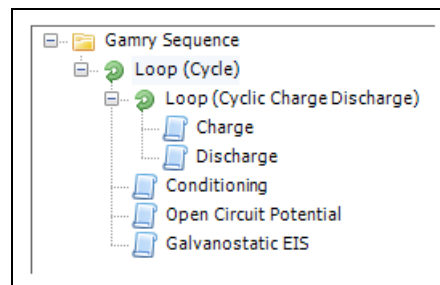



Figure 4 – Example of a sequence with a loop in a loop.

In this sequence, the outer loop consists of four different experiments. The first step of this outer loop is again a loop consisting of charge and discharge steps repeated for a given cycle number or if an end criterion is reached, e.g. capacity falls below a set value.

When the inner loop is finished, a conditioning phase starts followed by an open circuit potential (OCP) measurement and a galvanostatic EIS test. All steps of the outer loop are then repeated for a user specified number of cycles.

### Define and modify variables

By including and modifying variables, parameters can be transferred from one step to other steps. In combination with loops, this is an appropriate way to perform complex sequences in an easy way.

Figure 5 shows an example of a sequence with several cyclic voltammetry (CV) tests. The scan rate is increased after each loop until a final value is reached. Functions for adjusting variables have the symbol .

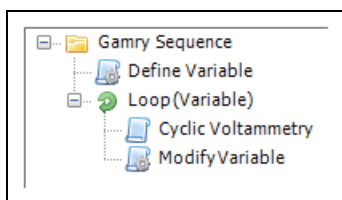


Figure 5 – Example of a sequence with modified variables.

The sequence starts with the function **Define Variable**. In its setup the variable name, type (potential, real, integer), and value of the variable are determined. Figure 6 shows the setup of this function.

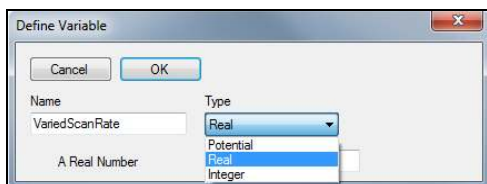


Figure 6 – Setup for the function *Define Variable*.

Note that variable names must be unique and have to start with a letter from A to Z. Three parameters are already predefined (VLAST, VLASTvsREF, and ILAST) and their names are reserved.

In the next step of the sequence a loop is started which contains a CV experiment and the function for modifying a variable, **Modify Variable**.

At the end of each loop cycle the value of the defined variable is updated. The setup for the function *Modify Variable* is shown in Figure 7.

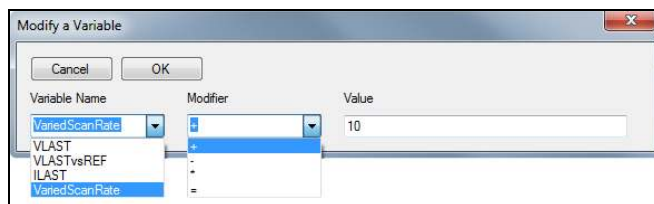


Figure 7 – Setup for the function *Modify Variable*.

A variable can be selected from a drop-down list. It is changed by a modifier (addition, subtraction, multiplication, parity) and a constant.

The loop in this sequence shall end when the selected variable reaches a defined end criterion. Hence the function **Loop (Variable)** is used. Figure 8 shows the setup.

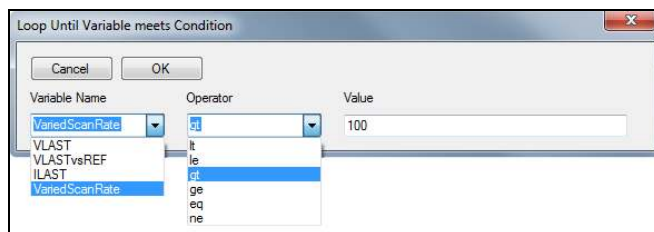


Figure 8 – Setup for the function *Loop (Variable)*.

The prior defined variable can be selected from a drop-down list. The loop end criterion is defined by comparative operators and a final value.

Finally, the defined variable has to be linked with the executed experiment which can be done in its setup. In this case, the variable is the scan rate of the CV test. Figure 9 shows three sections of a CV setup.

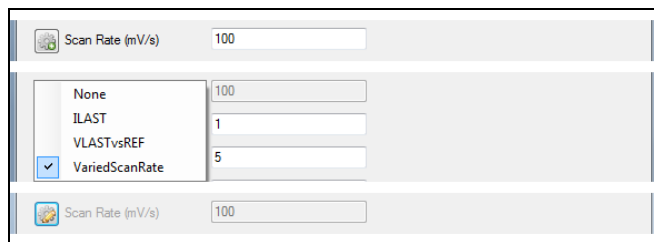



Figure 9 – Three sections of the setup for a cyclic voltammetry test. For details, see text.

Note that the value of the modified parameter does not have a unit in the setup of function *Define Variable*. The corresponding unit is shown in the setup of the experiment where the parameter is customized.


By clicking on the symbol left to the parameter, showing a gear with a plus sign , a drop-down list appears where the defined variable can be selected. As a result, the whole parameter line is grayed out and it is linked to the defined variable. By default no variable is selected.

After selecting a variable the grayed out line still shows a value for the parameter. Note this value does not have a function anymore. The parameter uses only the value that was set in the function *Define Variable* and updated in *Modify Variable*.

In this example the variable is named “*VariedScanRate*” and it is initially set to 10 mV/s. It is increased by 10 mV/s after each CV experiment. The total loop ends when a final value of 100 mV/s is exceeded. Ten CV files are automatically generated, each containing the actual loop number in its file name.

### File grouping

A complex sequence with several measurements like the example before can create a lot of files. The Sequence Wizard offers a function to store these files in one place – **Group Data Files**.

This function has the symbol . This symbol is also used for most experiments and functions which control the hardware and electrode settings.

The function *Group Data Files* groups all files that are generated in a sequence in one directory or adds a precursor to each file name. Figure 10 shows the setup.

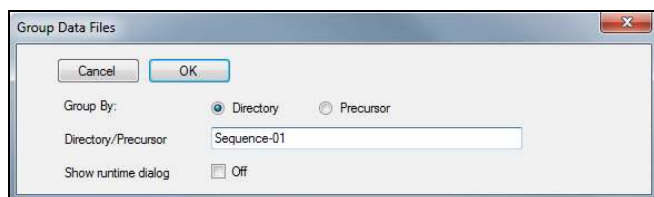


Figure 10 – Setup for the function *Group Data Files*.

The user can define the name for a group or a precursor respectively. If runtime dialog is enabled, a window pops up prior to the sequence start where each group/precursor name can be entered.

Figure 11 shows an example for a sequence that uses the function *Group Data Files*. It is placed before those steps which are supposed to be grouped. The function can be used multiple times in a sequence to create several groups.

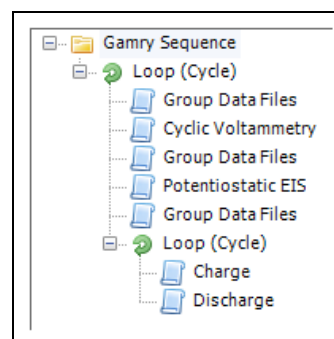


Figure 11 – Example of a sequence using the function *Group Data Files*.

In the example of Figure 11, three different directories are created. The first one includes all CV experiments, the second contains all EIS files, and the third all charge and discharge files. The actual loop cycle number is automatically added to each file name.

Using the grouping function, measurement files can be stored at one place and better organized. Further, repetitive sequences can be run by changing only a minimum of parameters. That makes it especially interesting for multi-channel applications as only the group name is changed for each channel or device.

### Conclusion

This application note described Gamry’s Sequence Wizard. By means of examples, the setup and the use of several sequence utilities were discussed.

Complex sequences of different experiments can be created with only few modifications, saving a lot of time in preparation and execution. Additionally, multiple experiments can run without interruptions and external disturbances.

Gamry’s Sequence Wizard – An easy way to create complex test sequences. Rev 1.0 6/8/2012 © Copyright 1990 - 2012 Gamry Instruments, Inc.

