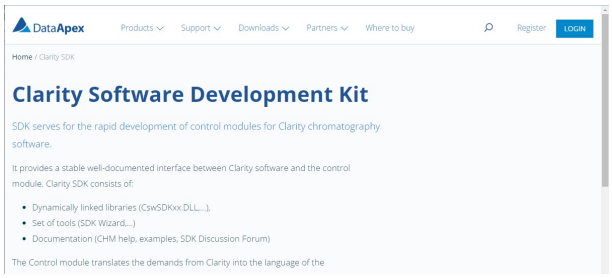
UNI Ruby

Set of tools for easy and quick development of control modules

Control modules are pieces of software creating a bridge between Clarity CDS and chromatographic hardware as detectors, thermostats, pumps, etc. Sometimes you might want to control a device that is rare or even custom made. And you feel that it should not take much effort to do it, because what the device does is relatively simple. If you know a bit about programming, such as how to write macros in Excel, and you are willing to learn Ruby language and also something about how Clarity interfaces control modules and chromatography hardware, then you can have it done. UNI Ruby provides code examples, mature projects, documentation and script editor all integrated with Clarity CDS to help you develop your control module. Control modules can be used for both personal and public use (see Script distribution). UNI Ruby's official website can be found on http://www.dataapex.com/uniruby.

For what is UNI Ruby **NOT** intended



If you are thinking about:

- development of complex control module (such as PDA detector, MS detector)
- control module with rich user interface

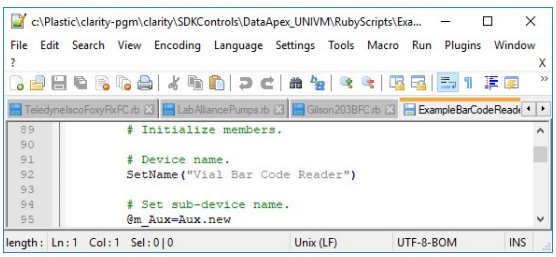
Then you should use **Clarity SDK**, see more at https://www.dataapex.com/product/sdk

Documentation

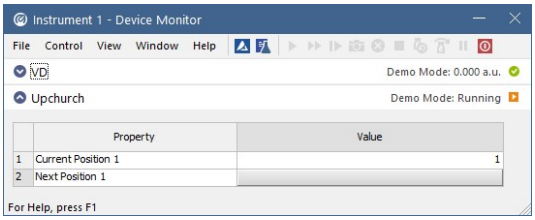
What does it take to write a Ruby control module? It is described on the first page when you double-click C:\Clarity\Bin\CswUNIVM.chm file (assuming default installation location). You will find there how to specify items you want to show in the user interface, how to establish communication with your device, etc. You will find a lot of links to examples where a particular feature is implemented. Or you can select an example that is close to your problem from C:\Clarity\Bin\Utils\Uni_Drivers\Examples folder, open it in Editor (see below) and open context documentation for its content. You can learn from the production Ruby projects as well.

Editor

If you check Utils - Notepad++ in custom Clarity installation, you will get Notepad++ installed on your computer along with the plug-in integrating the editor with documentation. Then, if you open your Ruby file in the Notepad++ and place the cursor on some word of interest, you can get help and documentation for it by pressing Ctrl+Shift+F1. Syntax check can be performed by pressing Ctrl+Shit+F7. All other features can be found in menu Plugins->UNI Ruby Plugin along with shortcuts and plugin configuration.



Clarity

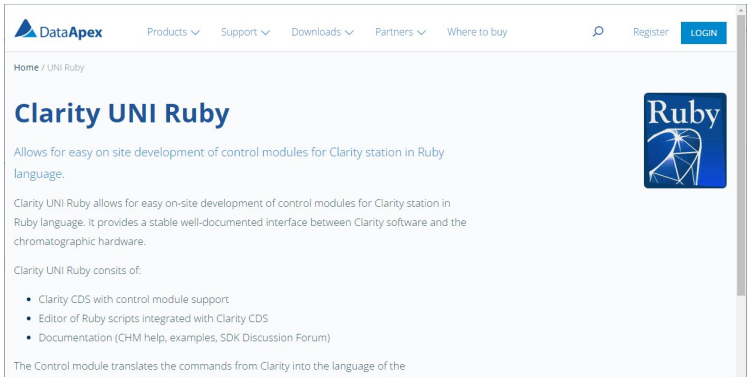


You already want to see your script in action. Run Clarity CDS, go to Configuration and Add UNI Ruby control module (in the Auxiliary category of the Available Control Modules dialog). In the displayed configuration dialog you can open your Ruby script file and bring it to life. When you leave your editor running and some runtime error occurs in your script, the culprit line will be selected in your editor.

❗ To use the script in Clarity, it is necessary to have control module license (e.g. A24, A26, ...) according to the device type (device type is determined by UNI ruby).

Script distribution

UNI Ruby can be used to write personal (private) scripts and official (public) scripts.



1. Personal scripts can be used without any approval by DataApex and you already have all needed tools and documents. Only in such use there will always be a warning in the Instrument window, that you are using an unauthorized driver.
2. If you want to distribute the script to multiple users or use it in a regulated environment, the official script is recommended. The official script does not have the above limitation as it has to pass internal testing and be approved by DataApex. To get more information contact support@dataapex.com so you can get full access to specific approval documentation:
    - D081 External control module development checklist
    - D115 Control Module Approval procedure
    - D126 DataApex CTRL Module Specification

Support

UNI Ruby documentation is provided without any support. DataApex support services are not required for personal scripts. Public scripts require DataApex testing and inclusion into Clarity family products. For consulting/testing following services are offered:

- p/n S001 SDK/UNI Ruby Support - annual fee, includes consultations and testing & incorporation of drivers/scripts
- p/n S028 UNI-Ruby script testing - one-off testing of one UNI-Ruby script (if the script does not comply with requirements outlined in D081 it will not be approved for public distribution).

Prices are charged according to a valid pricelist.

Step-By-Step

1) Install Clarity CDS and in custom installation select Utils – Notepad++ with UNI Ruby plugin.
2) Double click on C:\Clarity\Bin\CswUNIVM.chm and read the first (rather long) page.
3) Select an example or production Ruby script (*.rb) which is similar to your project.
4) Copy the selected script under the new name and open it in Notepad++.
5) Get an overview of what is done in the script by reading function names and comments included with them.
6) Get context help for the selected function by pressing Ctrl-Shift-F1.
7) Perform syntax check by pressing Ctrl-Shift-F7.
8) Run Clarity.
9) Go to System Configuration.
10) Add UNI Ruby driver in the Auxiliary section.
11) Select your script (by pressing the "…" button).
12) You can see user interface items in the UNI Ruby driver configuration dialog.
13) Close UNI Ruby driver configuration dialog.
14) Drag and drop just added driver onto Clarity Instrument and close Configuration.
15) Open Instrument.
16) Open Monitor Window to see your user interface items (if there are any).
17) Open the Method dialog to see your user interface items (if there are any).
18) Modify your script to implement communication parameters and communication protocol.
19) Modify your script to implement driver logic and error handling.

Add script into Clarity Installation:

20) Make sure the driver works in the DEMO mode too
21) See the approval documents listed above to make sure that your script meets the approval requirements.
22) Contact DataApex to arrange the approval of a new script.