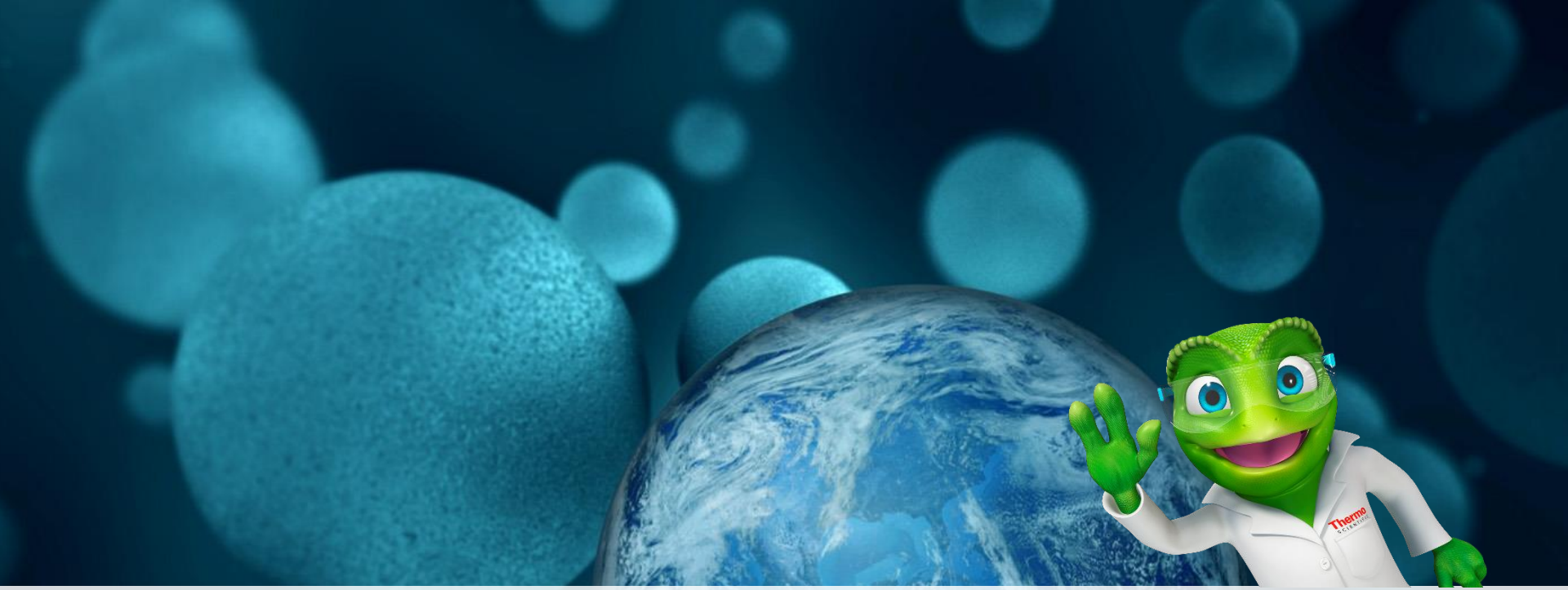


ThermoFisher SCIENTIFIC

Chromeleon DDK Development Training – Day5

Anton Kyosev, CMD Senior Staff Software Engineer
Yubo Dong, CMD Software Engineering Manager

Day 5 Agenda

- ❑ Advanced & Miscellaneous Topics
- ❑ Q&A

Inter-communication Between Modules

With `IDDK.RequestPropertyValue` method the DDK driver may trigger a request call to the instrument controller to get informed about the current value of a property of another driver instance running on this server. It is possible to monitor properties of a device and sub properties of a structure of a device. The access to the properties is read only.

```
m_DDK.RequestPropertyValue(string instrumentName,
                           string deviceName,
                           string propertyName,
                           RequestPropertyValueCallback cb)
```

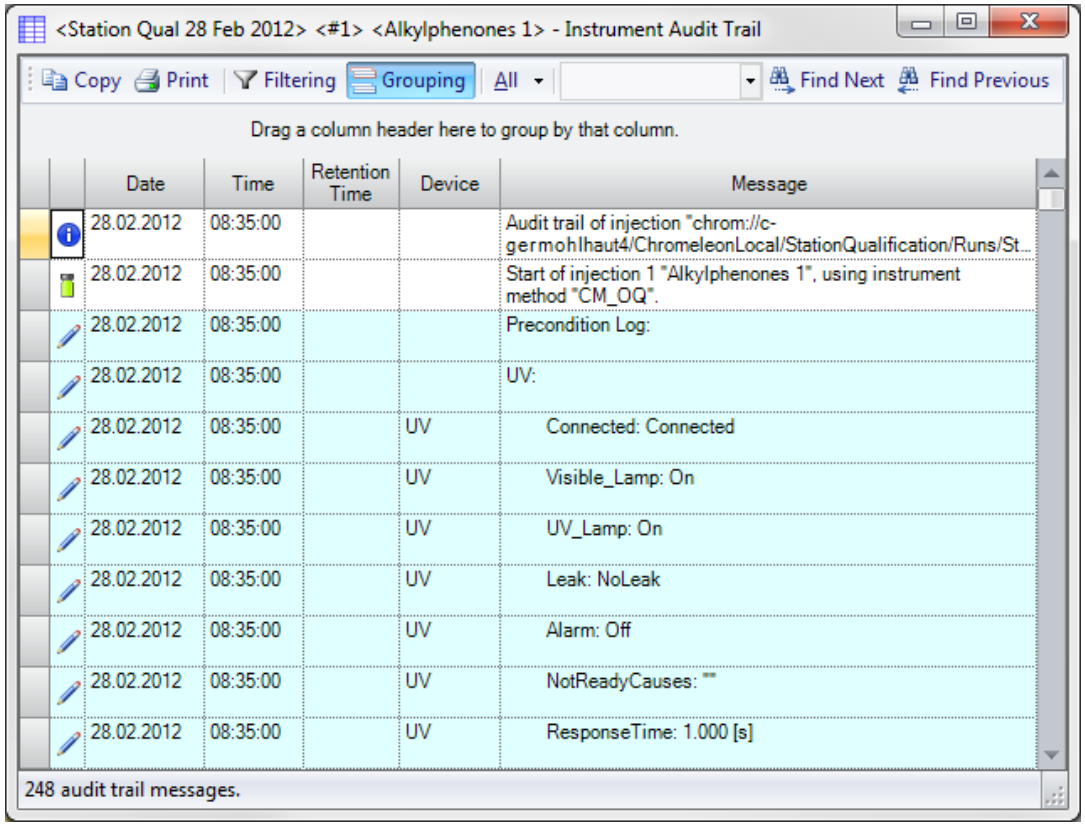
Inter-communication Between Modules

```
m_DDK.RequestPropertyValue("Demo_2", "Pump", "DeviceType",
                                                OnReceivedRequestedPropertyValue);


private void OnReceivedRequestedPropertyValue(RequestPropertyValueArgs args)
{
  if (args.Found)
  {
    Trace.WriteLine(args.Instrument + "." +
                    args.DeviceName + "." +
                    args.PropertyName + " = " + args.Value + " " + args.Unit);
  }
}
```

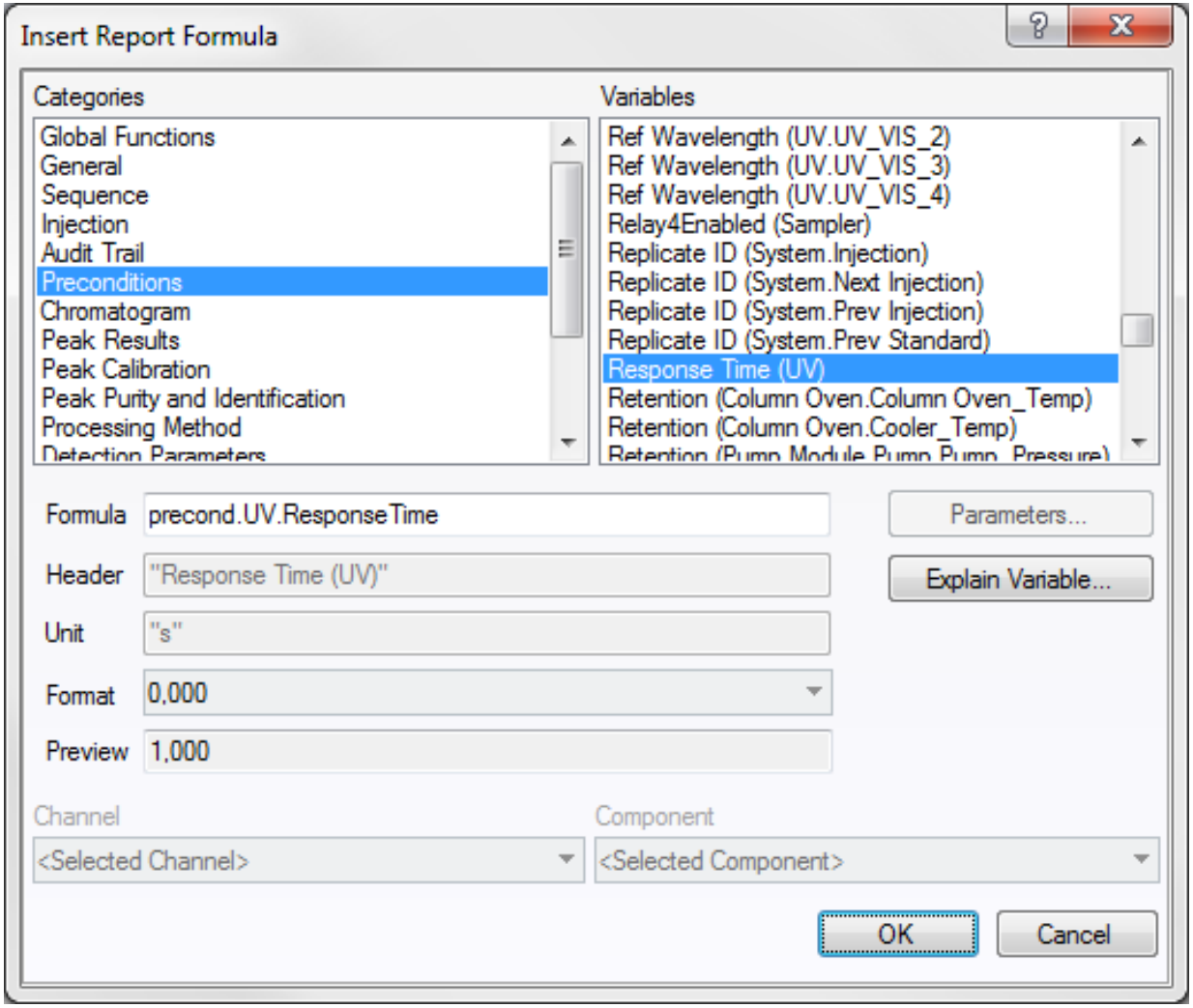
Precondition Log entries

At the start of each Injection, the Instrument Controller automatically generates a Precondition Log (PL). The PL is stored as part of the Injection Audit Trail and provides a dump of all properties that the Instrument offers, along with their current values:



Reporting Support - Precondition Log entries

Users can access individual items of the Precondition Log via Report Variables when designing reports:



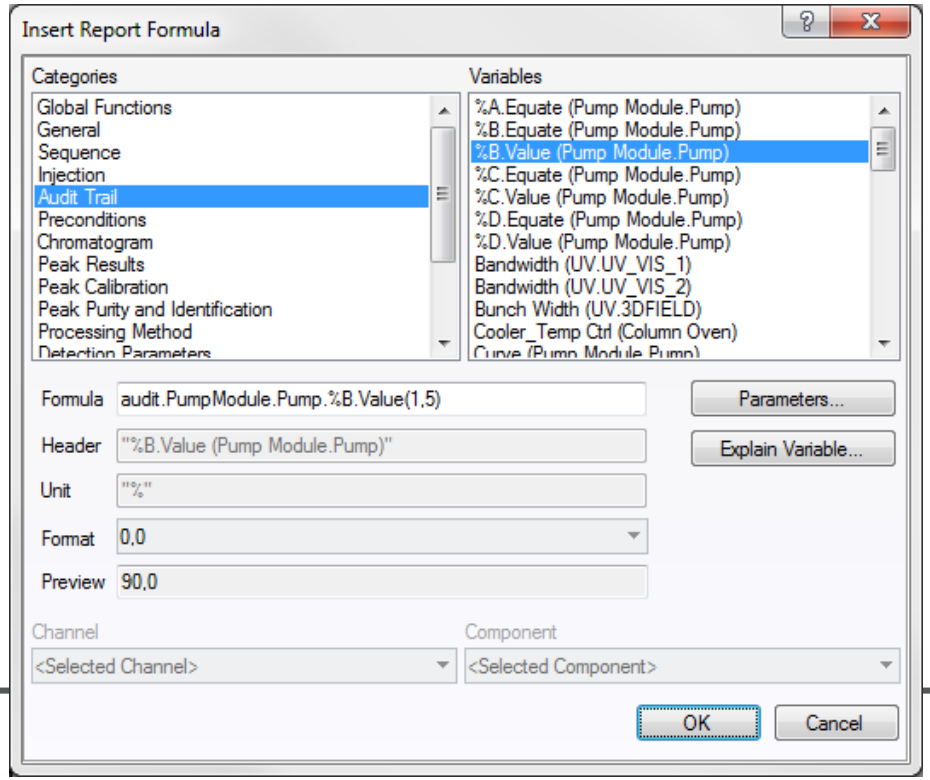
Reporting Support - Precondition Log entries

It is assumed that all drivers keep the values of all the properties that the drivers offer in sync with the state of the hardware. Only then can the Precondition Log provide a reliable snapshot of the hardware's state at the beginning of an Injection. To ensure that, do not change any injection related method parameters before the IDevice.OnPreflightToRun call, except for the Volume and Position setting updates of the IInjectHandler interface instance.

Audit Trail Report Variables

During an injection, the script execution engine writes all property assignments and commands to the Injection Audit Trail (IAT) before it dispatches the request to the corresponding driver. In other words, the IAT documents what the drivers **should** do (and not, what they did).

If an Injection has completed successfully (Status has changed to Finished), users can access individual items of the IAT via Report Variables when designing reports:



AutoSampler Injection Overlapping

Increases throughput by preparing the next injection in advance.

Some autosamplers support overlapped sampler preparation (e.g. Headspace samplers). Even if the autosampler firmware does not support such a feature explicitly it might be possible to start parts of the sampling process for the next pending injection while the previous analysis is still running (e.g. pre-injection washes, drawing the solvent from the vial).

To support such features, the driver must cover the following features

- **Injection Lock**: Before the preparation of an injection is started the driver must lock the corresponding entry in the Injection List to ensure that the user can't change the relevant parameters (Position, Volume and Instrument Method name) of this injection anymore.
- **Look-ahead**: The driver must be able to retrieve the related information (Position, Volume, and Instrument Method), for the injection to be prepared, which is not the currently running injection.

AutoSampler Sample Overlapping

This is possible by using the `ISequencePreflight` object, which is provided by the IDevice.OnSequenceStart, IDevice.OnSequenceChanged and IDevice.OnSequenceEnd events.

With the `ISequencePreflight` object, one can:

- Lock one or several injections by setting LastPreparingIndex
- Retrieve information about upcoming injections by using the Samples list

To get informed about changes of the running sequence while it is performed, one has to set `IDevice.UpdatesWanted` to `true`.

Instrument Method Wait

The **Wait** command interrupts instrument method execution until the specified condition is fulfilled.

Time	Command	Value
0.000	Inject Preparation	
	UV.Autozero	
	Wait	UV.Ready
	Wait	UV.Ready And Sampler.Ready
0.000	Inject	

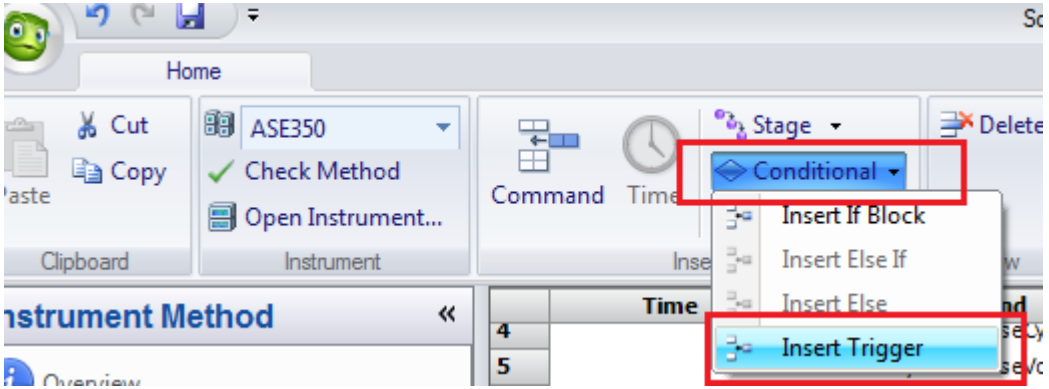
Instrument Method Trigger

It consists of a trigger step and a block of method steps. These commands are executed if the trigger condition Flow > 100,

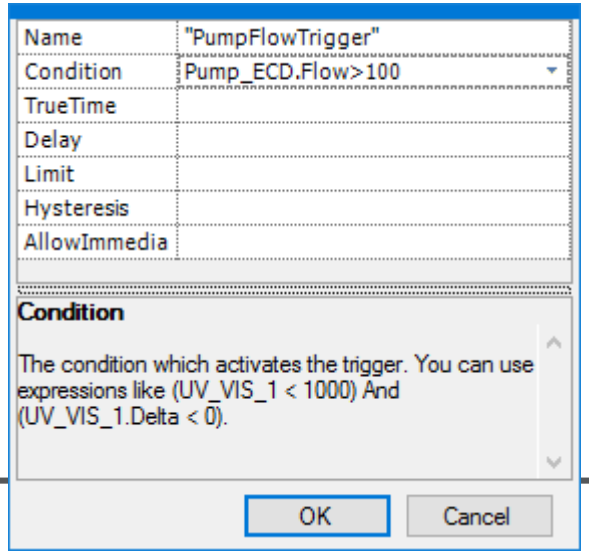
becomes true, for example if a signal exceeds a specified threshold. The commands in the trigger block are executed every time the specified condition changes (edge triggering), i.e., upon each transition from false to true.

The condition can be complex:

(UV_VIS_1>100+1) AND (UV_VIS_2>100)



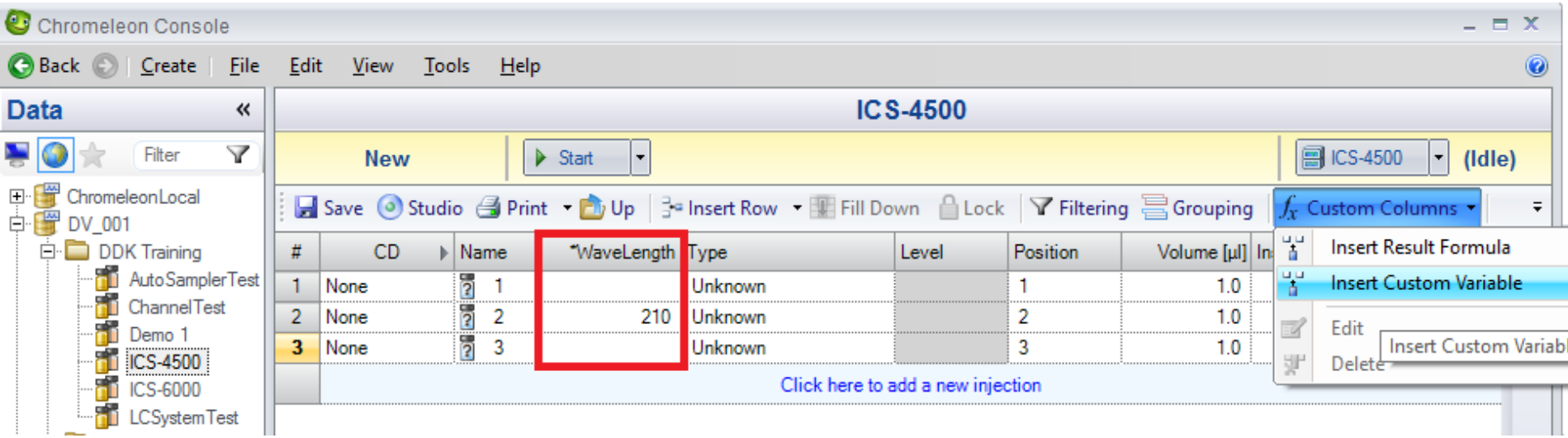
	Time	Command	Value
2		EDet.Mode	IntAmp
3	Trigger		"PumpFlowTrigger", Pump_ECD.Flow>100
4		Log	Pump_ECD.Flow.Value
5		Log	Pump_ECD.Pressure
6	End Trigger		
		System.AcqOn	



ThermoFisher SCIENTIFIC

Custom Variable

Injection and sequence specific custom variables can be used in the Instrument Method. You can use them when assigning values to control commands in the Script Editor, either directly or in expressions.

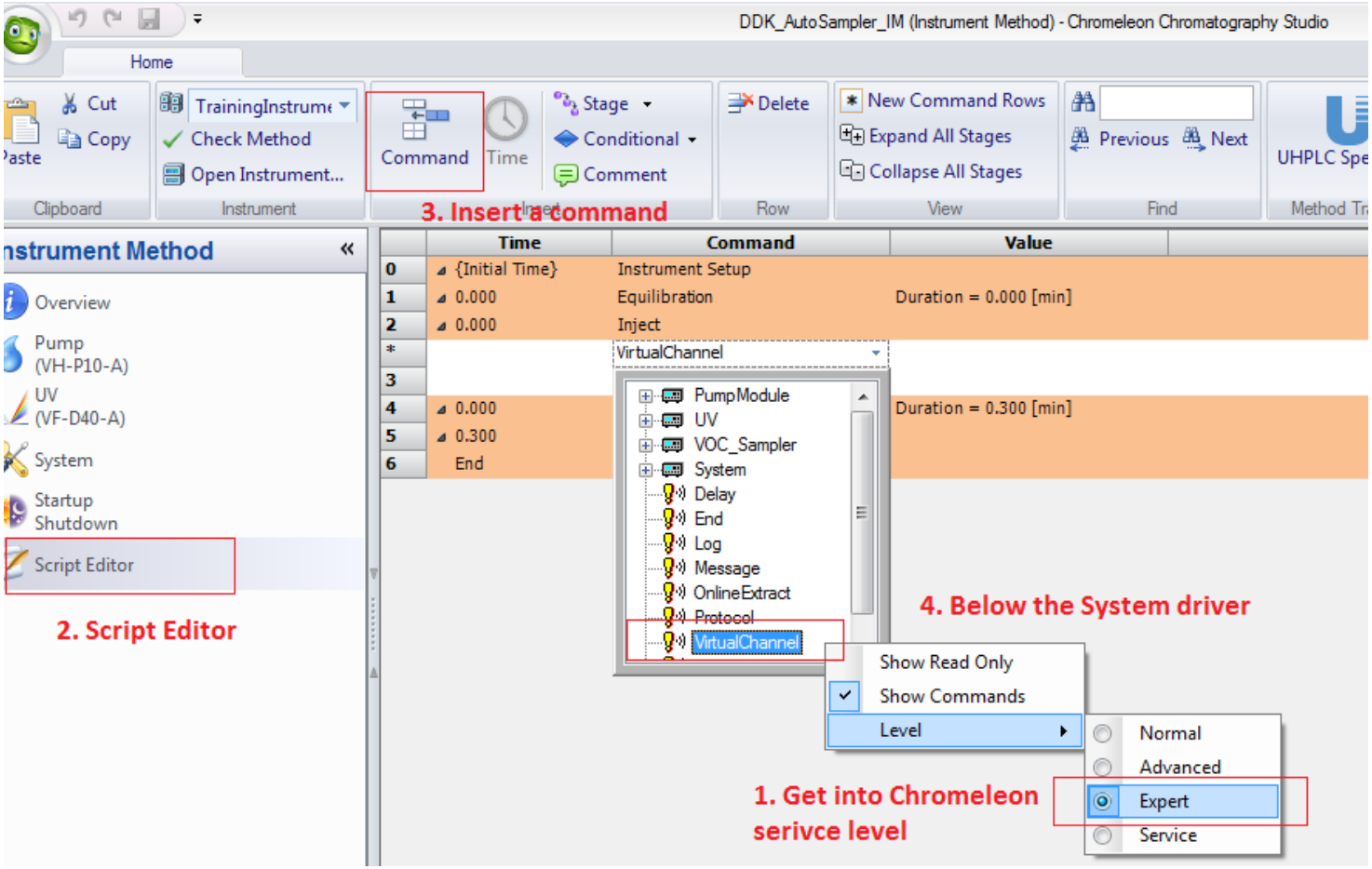


Use in IM: Sysem.Injection.CusotmVariable.WaveLength

	Time	Command	Value
39		Log	System.Injection.CustomVariables.WaveLength

Virtual Channel

Virtual Channels can be used to record a device property or calculate an arbitrary numeric formula during data acquisition, and then save the result as an additional signal. Any combination of numeric expressions can be used in the formula that is used for calculating the virtual signal.

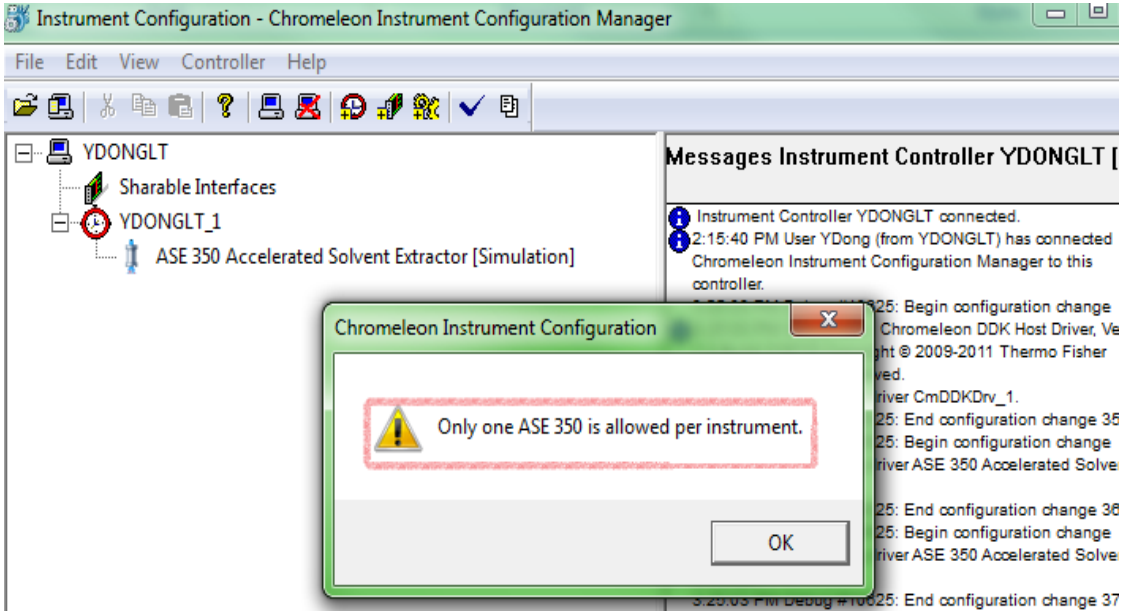


Limit One Module in a Instrument

If it is required to restrict only one module per instrument, a Thermo developer has to modify some files in the legacy folder:

- Legacy\CRBIOS\CmInst\CmInst**RSource.h**
- Legacy\CRBIOS\CmInst\CmInst**GSInstal.rc**
- Legacy\CRBIOS\CmInst\CmInst**GSInsVW.cpp**

Everybody else – Contact DDK support



ThermoFisher SCIENTIFIC

Localization

Chromeleon 7 officially supports English, Chinese, Japanese

- ICM report : All strings for the ICM report should be in resource file, and can be localized
- UI components (dialogs, strings) should be put into resource file
- Show/Parse numbers in user current culture
- Store numbers invariant

```
value.ToString(CultureInfo.InvariantCulture)
```

- Symbol name should NOT be localized
- Allow enough UI space for different languages

DDK Driver Certification Requirements

Specified in

Chromeleon DDK Driver Certification Requirements.pdf