



ThermoFisher
SCIENTIFIC

Chromeleon DDK Development Training – Day1

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

Day 1 Morning Agenda

- ❑ Welcome
- ❑ Coffee & Facilities
- ❑ Breaks
 - 10:00 – 10:15
 - 12:00 – 1:00 pm Lunch
 - 3:00 – 3:15 pm
- ❑ Introduction
- ❑ Chromeleon Overview

Chromeleon Installation

Chromeleon default installation folder

C:\Program Files (x86)\Thermo\Chromeleon\Bin\

Developer should not install in C:\Program Files (x86)\

Change the installation folder to

C:\Thermo\Chromeleon\Bin\



Install Thermo.Chromeleon.DDK.msi

Follow the steps in
Release Notes - Chromeleon 7 DDK.pdf on
page 5 1.6 Further Preparations:

Add the Chromeleon DDK V2 components to
Visual Studio Toolbox

Chromeleon Installation – License

License file: **Chromeleon7.cmlic**

Chromeleon Administration Console (USSVL-GWJJJC2)

The screenshot shows the Chromeleon Administration Console window. On the left, a tree view navigation pane lists several modules: License Manager (localhost) (selected), Scheduler, Global Policies, User Database, eWorkflow Tags, Audit Trail, Domain Resources, and Local Machine. The main right pane displays four icons with associated text:

- Client Licenses**: Assign client licenses to user accounts or instrument controllers.
- Instrument Controller Licenses**: Assign licenses to instrument controllers.
- Update License File**: Upload a new license file to the license server. This item is highlighted with a red rectangular border.
- Information**: Shows details about the license server: License Server at computer USSVL-GWJJJC2 (localhost), Serial No.: 111002, License Server Version: 7.3.0.0, and Operating System: Microsoft Windows 10 Pro Vers. 10.0 Build 14393.

At the bottom left, the text "License Manager (localhost)" is displayed. At the bottom right, the text "anton.kyosev" is shown next to a small profile icon.

Chromeleon Installation – DDK Developer License

Set the license DDK Developer bit

The screenshot shows the Chromeleon Administration Console interface. On the left, there's a navigation tree with items like 'License Manager (localhost)', 'Client Licenses', 'Instrument Controller Licenses' (which is selected and highlighted with a red box), 'License Overview', 'Scheduler', 'Global Policies', 'User Database', 'eWorkflow Tags', 'Audit Trail', 'Domain Resources', and 'Local Machine'. The main window title is 'Instrument Controller Licenses' with the subtitle 'Assign licenses to instrument controllers.' Below the title is a note 'Drag a column header here to group by that column.' A table lists instrument controllers. For 'USSVL-GWJJJC2', the 'DDK Developer' checkbox is checked and highlighted with a red box. At the bottom of the table, there's a checkbox for 'Installation Mode (every request from instrument controller is saved immediately)'. There are 'Save Changes' and 'Revert' buttons at the bottom right. The status bar at the bottom shows 'SQ / QqQ GC-MS Data Acquisition, Total: 200, Granted: 0, Available: 200'.

Chromeleon DDK Development Support

DDK development support email

DDKSupport@ThermoFisher.com

Chromeleon Terminology

Term	Meaning
Instrument	a set of Modules that share a common clock
Module	hardware (pump, detector, etc.) represented as a single unit
Device	functional part of a Module
Driver	software that is needed to control a Module
Script	a series of property assignments, commands and other instructions controlling the execution order
Injection	a single run represented by a single line in the Injection List of a Sequence
Sequence	a collection of Injections
Queue	a collection of Sequences queued for execution. Each item within the queue might have been added by a different user.
Audit Trail	messages that were generated during instrument controller operation. They have different level of importance: Info, Warning, Error
Data Vault	a storage location for Chromeleon data

Chromeleon Terminology

Term	Meaning
Channel	a device that produces data during an Injection. It has Acquisition On and Acquisition Off commands.
Signal	data produced by a Channel. It can be 2D such as Pump.Pressure, Oven.Temperature, UV.VIS_1 or 3D such as UV.3DFIELD or FLD.3DFIELD.
Instrument Controller ICT	Chromeleon service responsible for forwarding commands to the Drivers. It also controls the Queue runs and manages the incoming Data.
Real Time Kernel RTK	a service, which hosts and manages the individual Driver instances

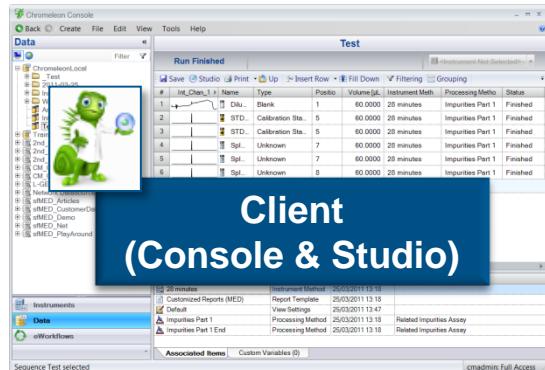
Chromeleon Terminology

Old	New
Timebase	Instrument
Batch	Queue
Sample	Injection
Tray	Rack
Manual Acquisition	Monitor Baseline
Program (PGM)	Instrument Method (IM)
Server	Instrument Controller

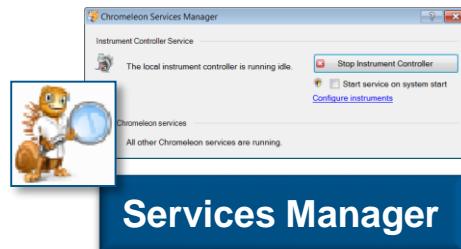
Chromeleon Architecture

Application
(with UI)

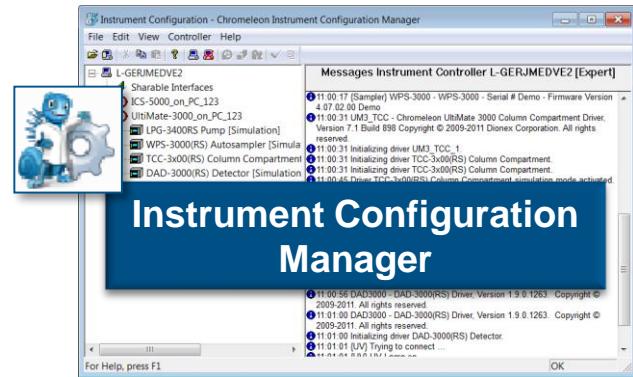
Service
(No UI)



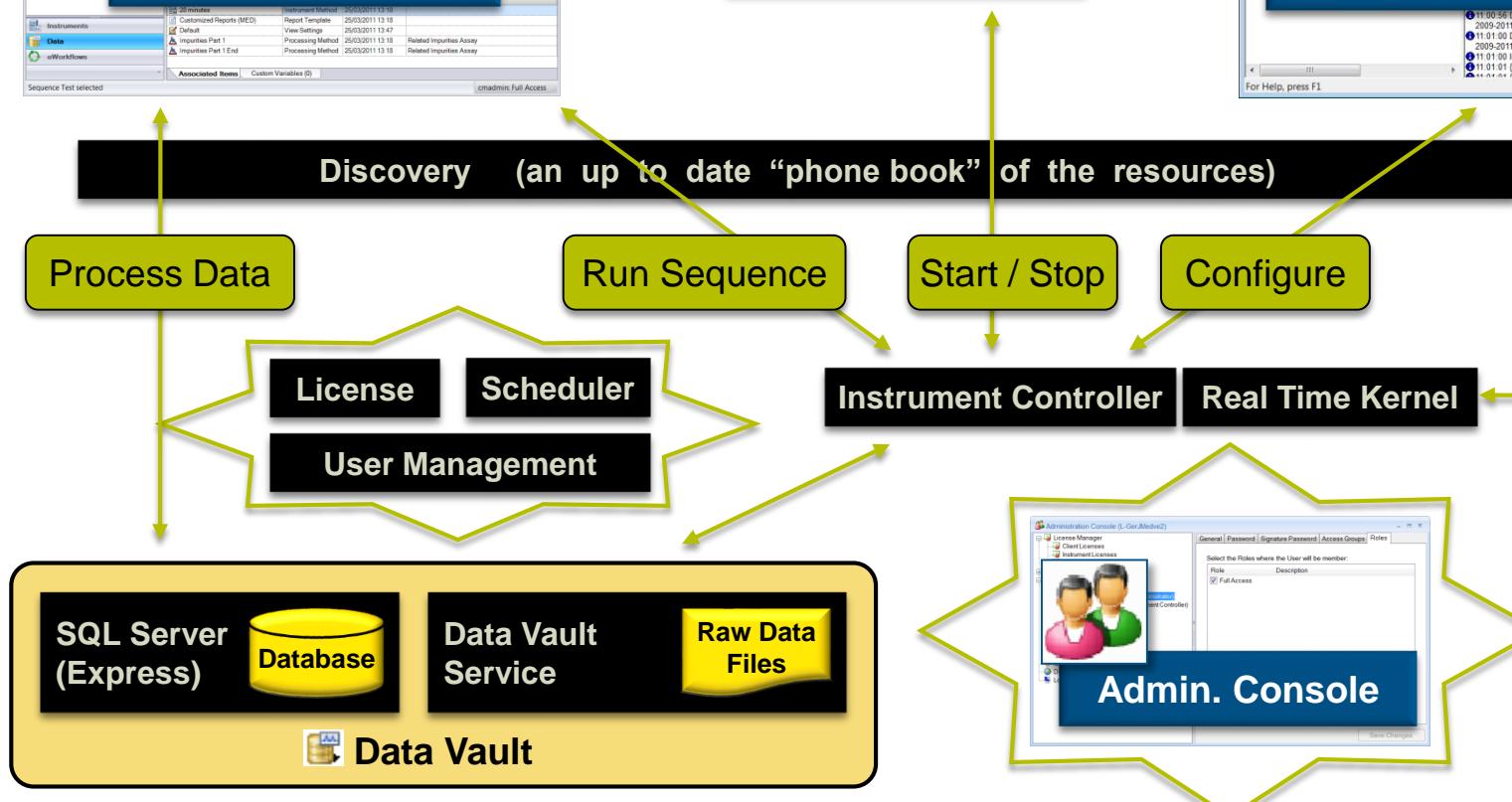
**Client
(Console & Studio)**



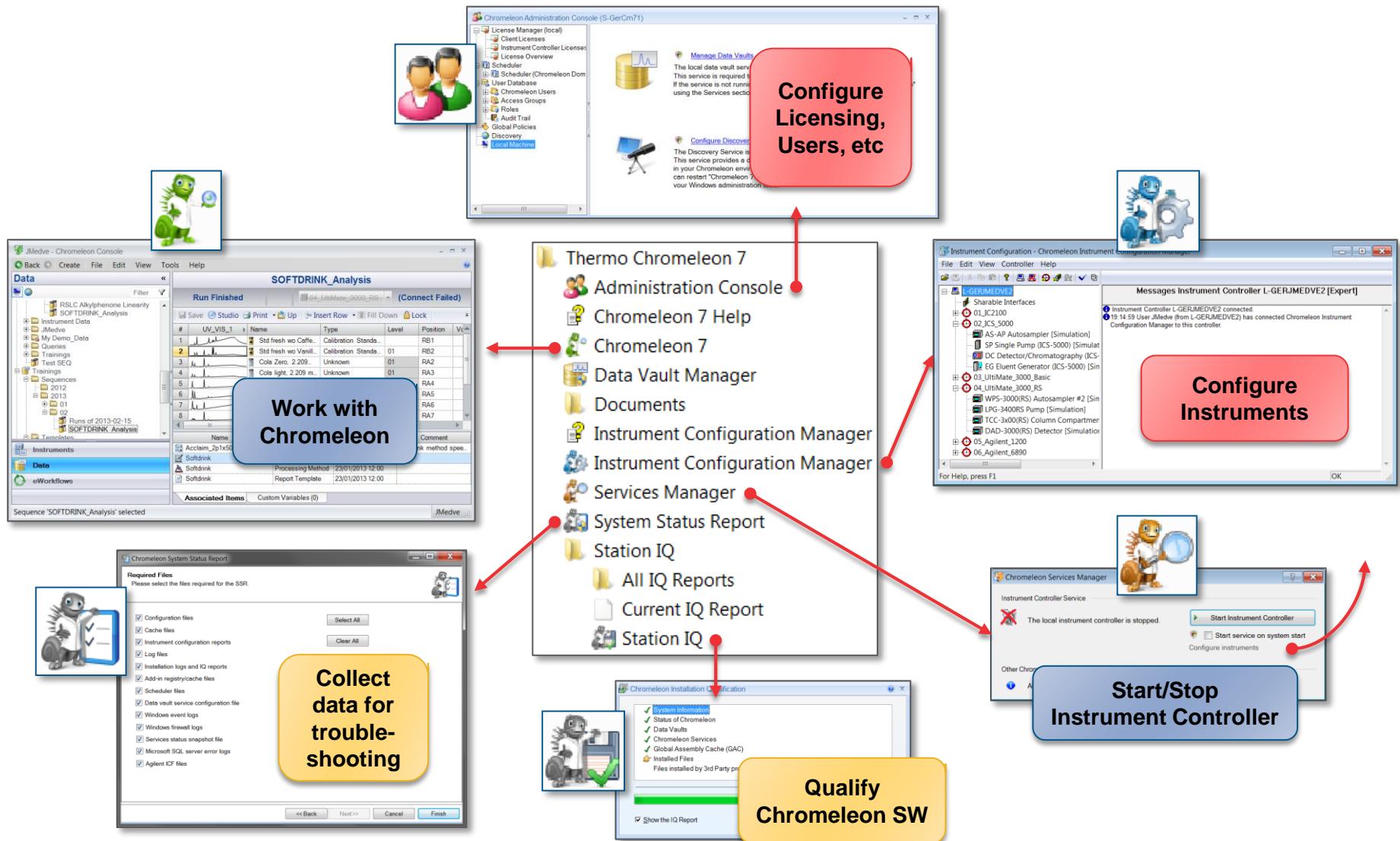
Services Manager



Instrument Configuration Manager



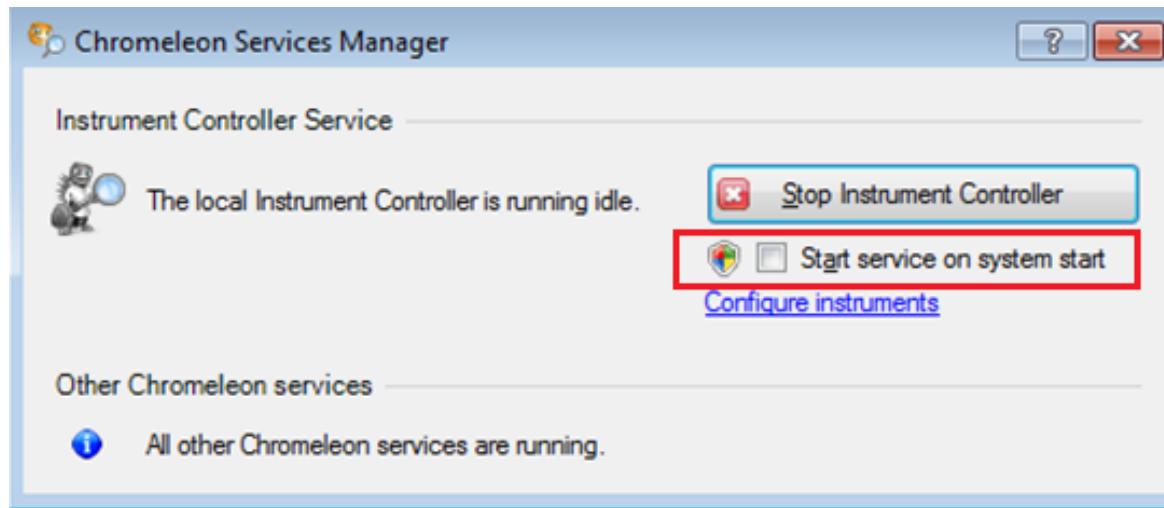
Chromeleon Applications in the Start Menu





Services Manager – Use and Start

- Use the Services Manager to:
 - Monitor the status of the **Instrument Controller** and other services
 - Start/Stop the **Instrument Controller** service (Default is not started)
 - Launch the **Instrument Configuration Manager** tool

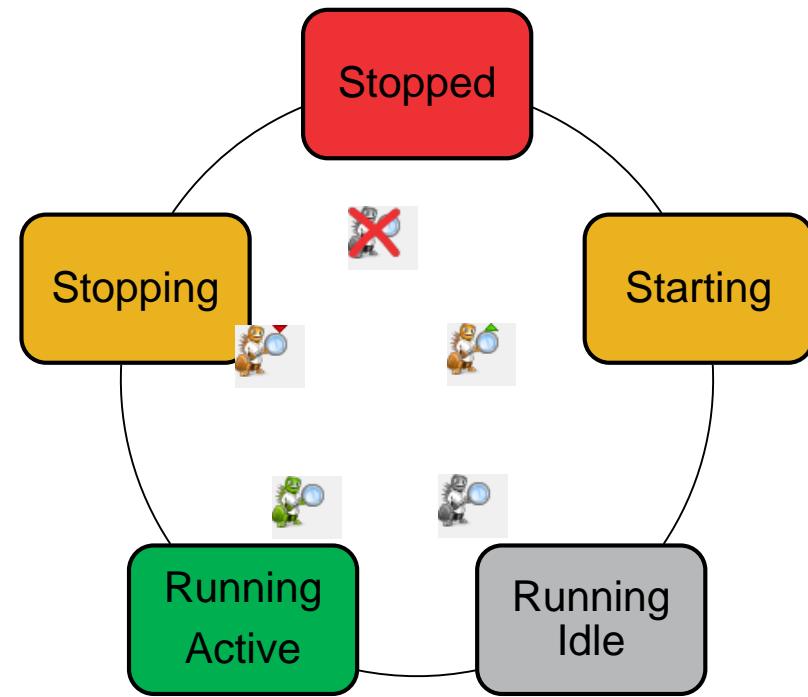
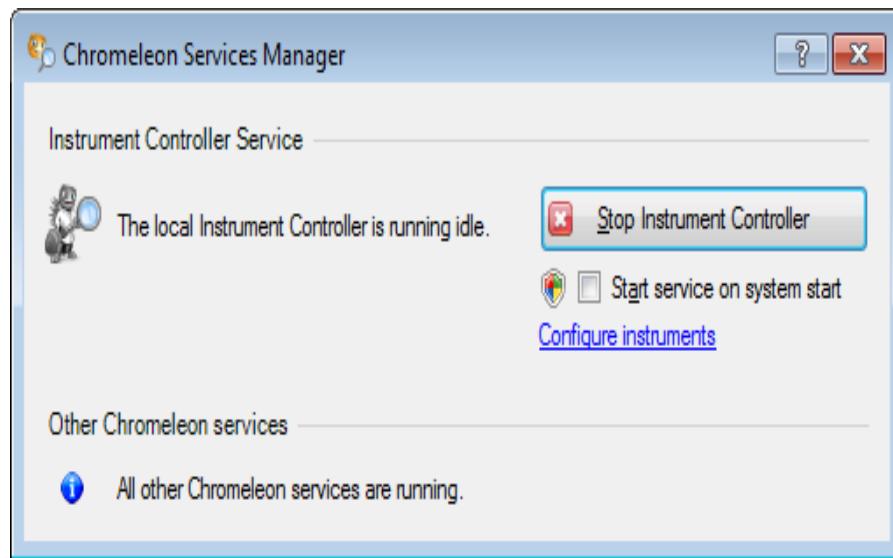


- Starting the Services Manager
 - It is normally **minimized** to the Notification Area of the Taskbar.
 - If not, you can (re)start it from the Start Menu:
Start > All Programs > Chromeleon 7 > Services Manager



Services Manager – Instrument Controller Status

The Services Manager displays the status of the Instrument Controller Service.



**You can log off from Windows while the Instrument Controller is Running.
But you cannot change the Instrument Configuration.**

Client = Console + Studio



Console – entry point to Chromeleon

- Controls and monitors **Instruments**
- Organizes the **Data**
- Works with **eWorkflows**

The screenshot shows the Chromeleon Console interface. On the left, there's a navigation tree with categories like ChromeleonLocal, Test, 2011-03-25, Instrument Data, Workshops, etc. The main area displays a table titled "RSLC Linearity Customizing_02Sep11 18.57" with columns for #, UV_VIS_1, Name, Position, Type, Level, Volume [µl], Status, and Inject Time. Below the table is a chromatogram showing five peaks labeled 1 through 5. At the bottom, there's a message "Click here to add a new injection". A status bar at the bottom says "Sequence RSLC Linearity Customizing_02Sep11 18.57 selected" and "cmadmin: Full Access".

Studio – The place to work with your chromatography data

- View/Edit the **Injection List**
- View/Edit the **Instrument Method**
- View and process data (via the **Processing Method**)
- Create **Reports & (signed) Electronic Reports**
- Work with **Spectra Libraries**

The screenshot shows the Chromeleon Chromatography Studio interface. On the left, there's a sidebar with "Data Processing" expanded, showing sections for Injections, Channels, Processing Methods, and View Settings. The main area features a chromatogram with ten labeled peaks and their corresponding chemical names and areas. Below the chromatogram is a "Summary Table" table with columns for Sample No., Sample Name, Ret. Time min, Area mAU/min, Height mAU, and Amount n.a. The table has rows for samples 1 through 5, an average row, and a Rel.Std.Dev. row. The status bar at the bottom says "Integration \ Calibration \ Summary /" and "cmadmin: Full Access".

Typical Tasks of a Chromatographer – 1

- The Core Task: Analyzing Samples

- Analyze a list (**Sequence**) of samples (**Injections**) using a given instrument and report the results.



- Typical stages involved

- Create supporting files (e.g. Instrument Method)

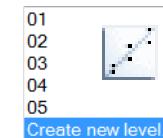
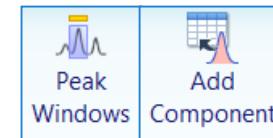


- Run the Sequence (submit to the **Queue**)



- Process the data

- ✓ Detect peaks
 - ✓ Identify peaks
 - ✓ Calibrate



- Report the results



Chromleon 7 takes you from samples to results with the fewest possible steps thanks to its design principle: Operational Simplicity™.

Typical Tasks of a Chromatographer – 2

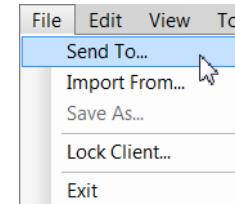
- Other Typical Daily Task: Manual Instrument Control

- Manual instrument control for startup, equilibration and shutdown
- Monitoring the baseline



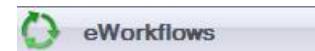
- Occasional Task: Data transfer

- Transfer data between labs
- Collect data for troubleshooting



- Automation Possibilities

- Smart Startup, Smart Standby, Smart Shutdown
- eWorkflows – an electronic procedure for automating the laboratory processes related to a chromatographic analysis.



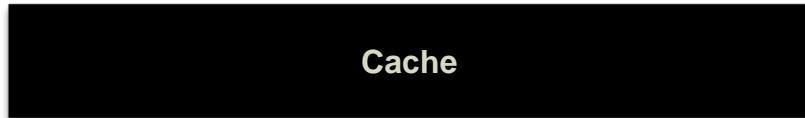
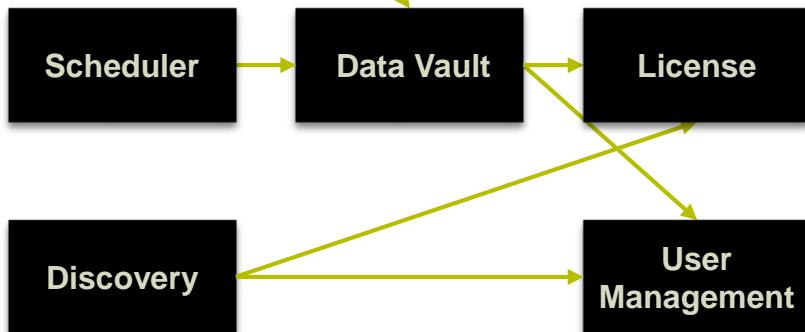
Chromeleon Objects in Console > Data Category

-  **Sequence:** The basic unit of data in Chromeleon. It contains all the data and metadata that are necessary to (re)create the results
 -  **Instrument Method:** Set of timed property settings, commands for an instrument
 -  **Processing Method:** Collection of parameters used for evaluating a chromatogram including peak detection, identification and calibration
 -  **Report Template:** Spreadsheet-like file that defines how data is printed or exported. When applied on a Sequence (to view, print or export the results) the output is referred to as a Report.
 -  **Electronic Report:** Electronic snapshot of the results of a Sequence
 -  **View Settings:** Defines how data is presented on the screen, including settings for the Interactive Result Tables, chromatogram and calibration curve.
 -  **Spectral Library:** Collection of spectra used for peak identification.
- * **Attachments:** Any file can be added to a Sequence as attachment.

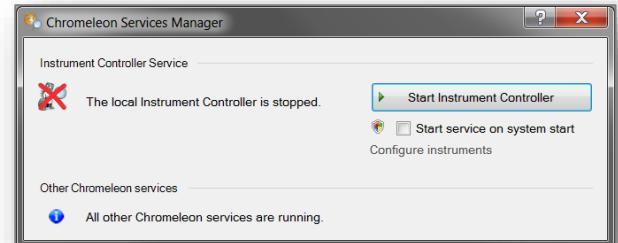
Other objects:

-  **Folder:** Container for objects helping you to organize data
-  **Query:** Search function to find data that match criteria you provide

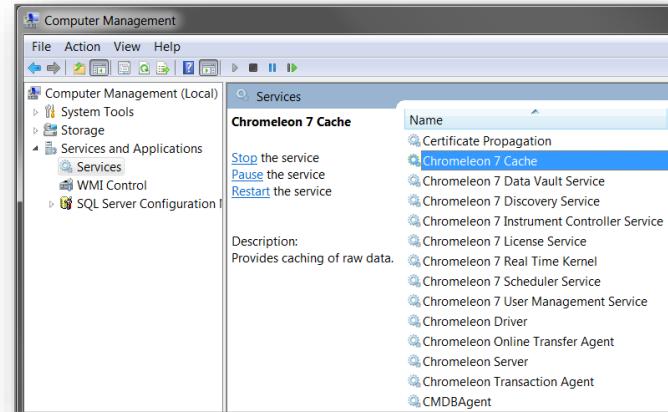
Troubleshooting: Start/Stop Services (Dependencies)



Use Services Manager to start/stop Instrument Controller



Use Computer Management to start/stop other services



Service on the Left
depends on
Service to the Right

Comprehensive Instrument Control - drivers

The Driver Development Kit (DDK) is the Chromeleon's solution for third party instrument control.

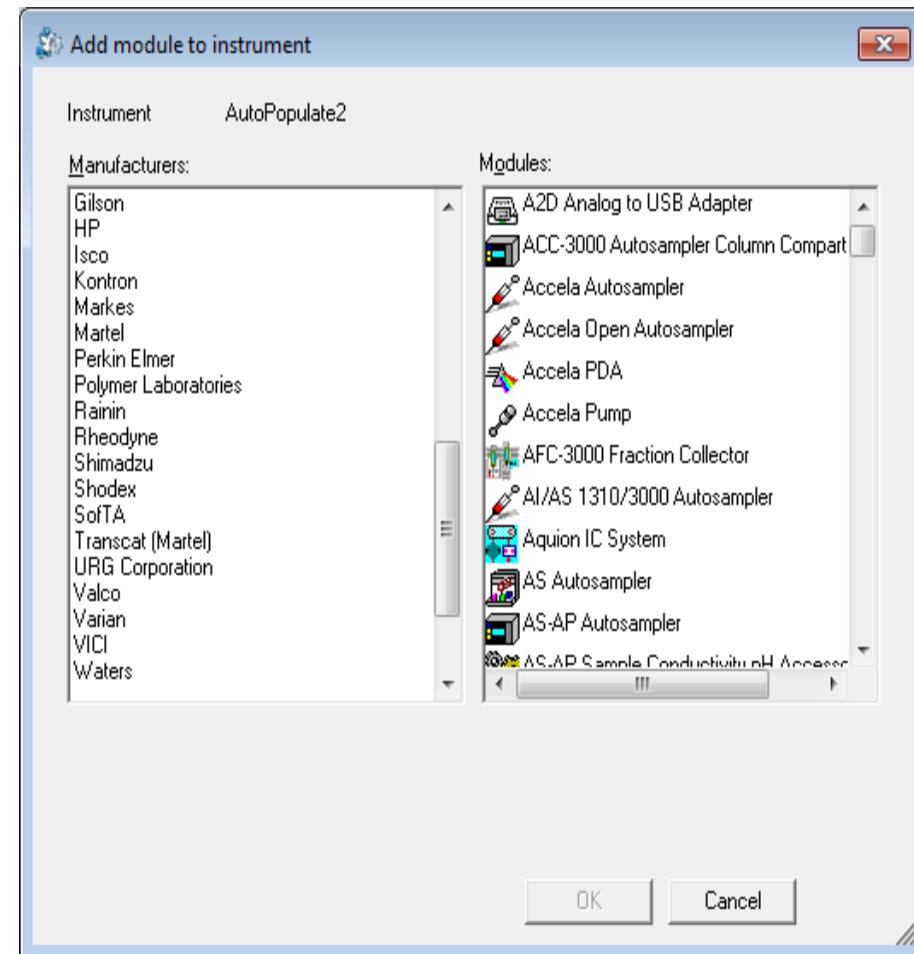
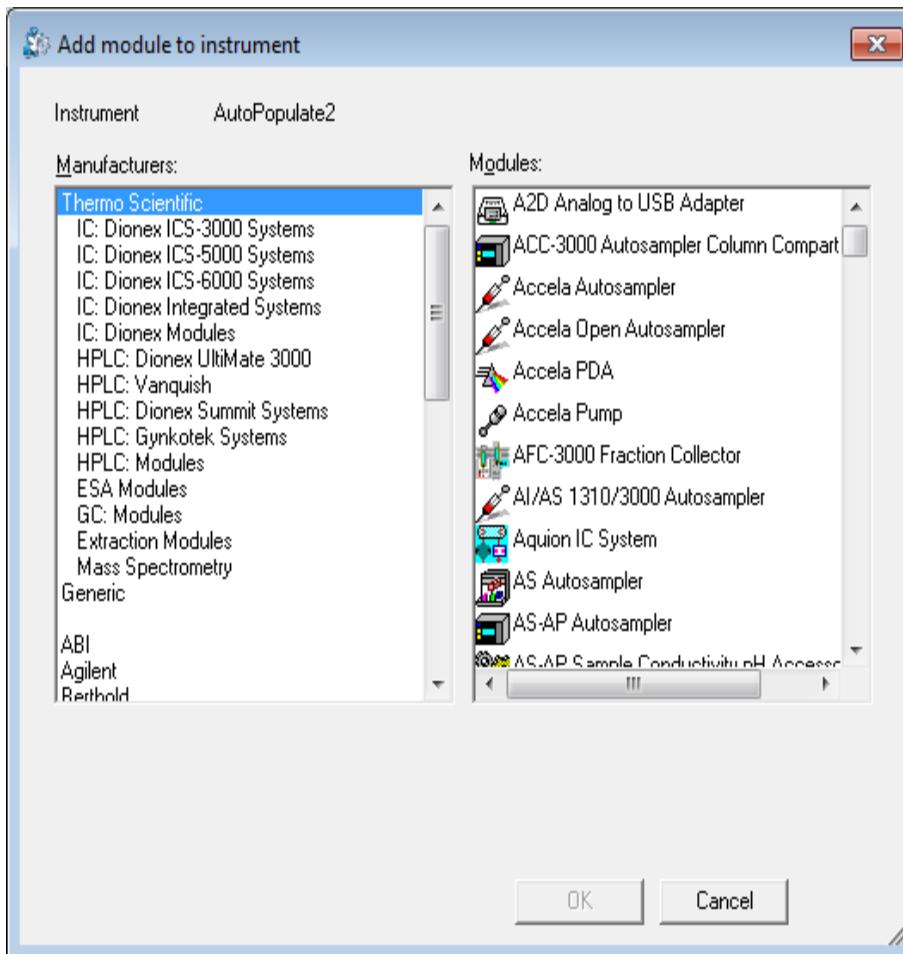
- Used by Thermo Fisher Scientific for driver development
- Available to other suppliers
- Supports plugins to add features
- Integration of wrapped driver plugin solutions from leading instrument manufacturers like:

Agilent Instrument Control Framework

Waters Instrument Control Software

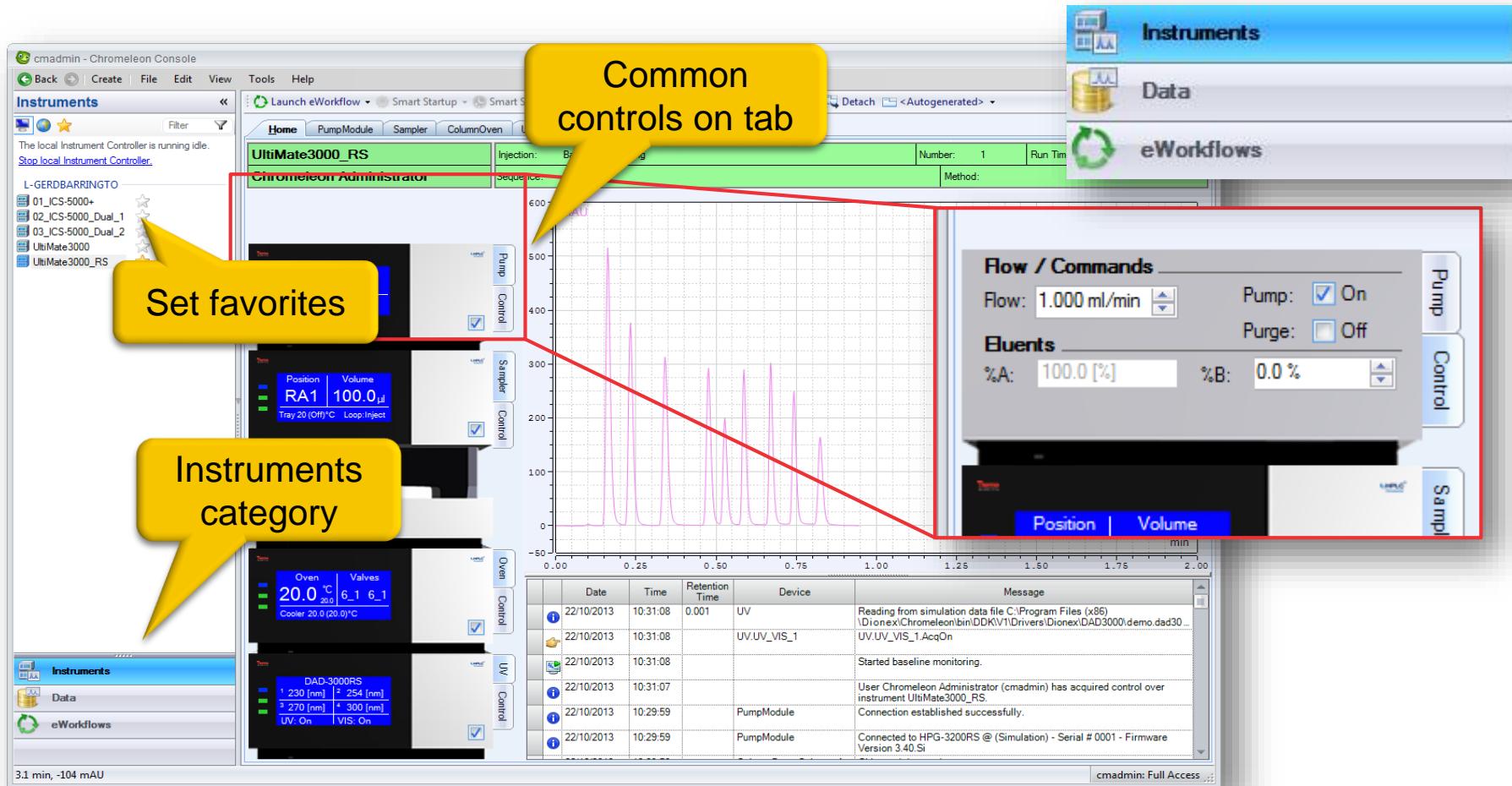
Support Multiple Instrument Vendors

Chromeleon CDS supports over 400 different modules from 18 different manufacturers



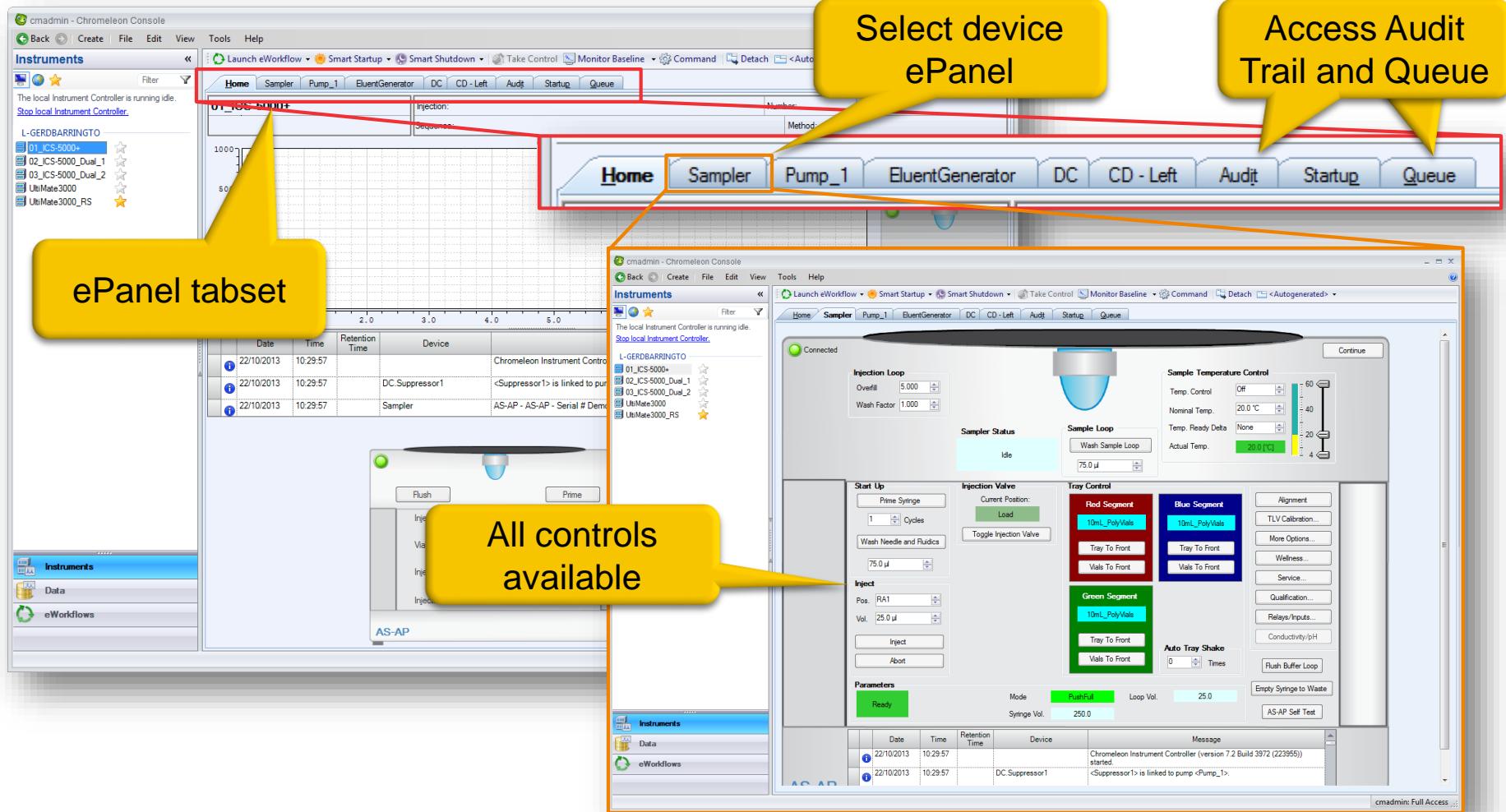
Comprehensive Instrument Control - ePanels

The Chromeleon Console Instruments category provides access to all instruments, and Chromeleon ePanels provide a customizable graphical interface with comprehensive access to instrument commands



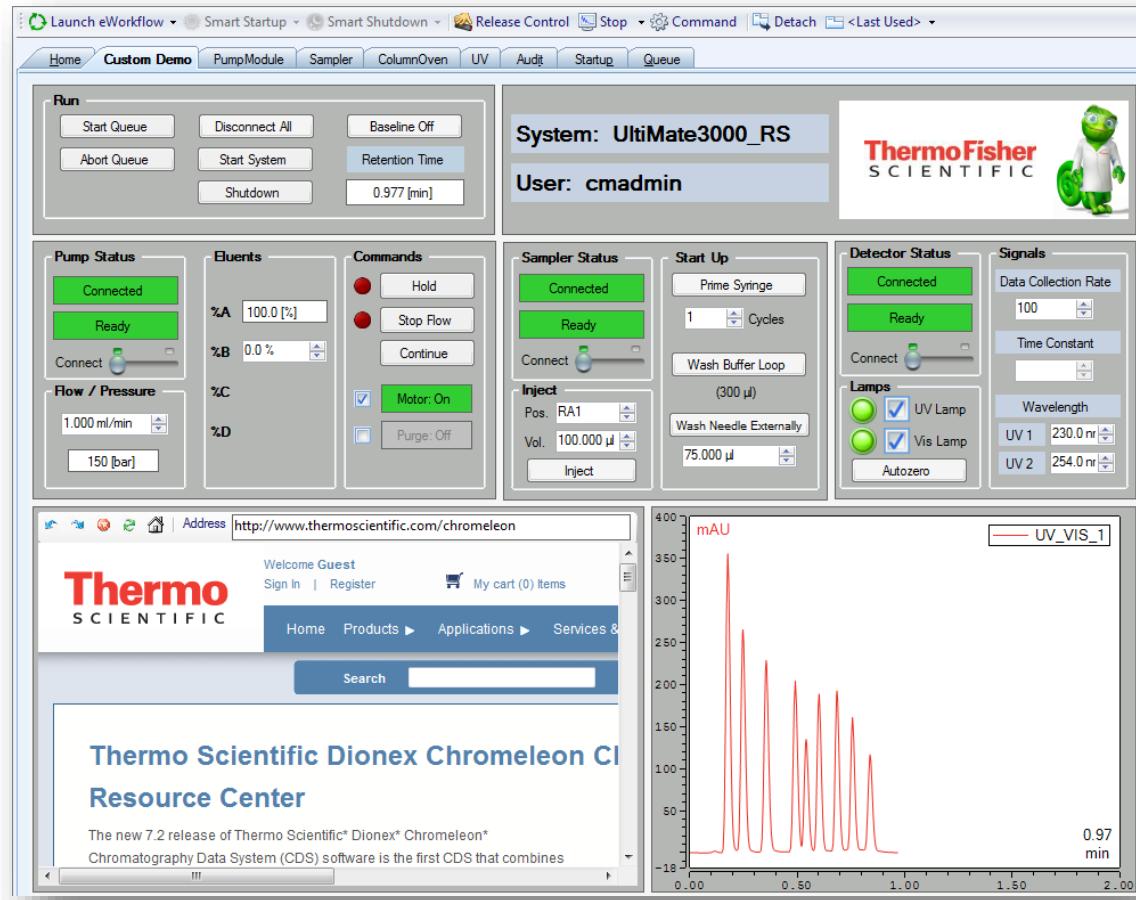
Comprehensive Instrument Control - ePanels

All controls available on device ePanels



Comprehensive Instrument Control - ePanels

- Uniquely customizable – create your own universal ePanels!
- Create a common ePanel for all your instruments.
- Add:
 - Images
 - Text
 - Advanced controls
 - Plots
 - PDF viewer
 - or even a Web browser!



Ensuring “Right First Time” Analyses

- **Controlled Sequence Creation**

Easily start your analysis in accordance with your SOPs with just a few clicks using eWorkflows.

- **Sequence Ready Check**

Run-time checking of your sequences ensures correctness.

- **Ensuring Instrument Readiness**

Configure your instruments to run with exactly the right settings for your analysis with Smart Startup

- **Run-time decision making**

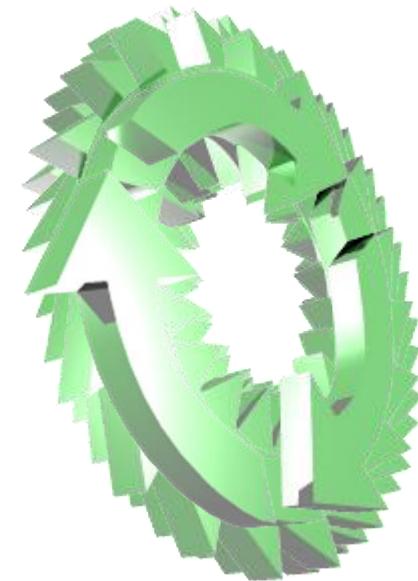
Built-in System Suitability Tests with Intelligent Run Control take automated, in-sequence actions based on the actual chromatographic results.



Right First Time Analyses – Sequence Creation

To ensure that a sequence is created and structured correctly and has the right methods and reports - use eWorkflows. They contain:

- Instruments on which the analysis can be run
- All associated files:
 - Instrument methods
 - Processing methods
 - Reports
 - Spectral libraries
 - External documents (e.g. SOP)
- Template for sequence name and storage location
- Custom variables
- Rules for sequence layout



Controlled Sequence Creation - eWorkflows

eWorkflow is the fastest way from sample to results with minimal effort and training. It is an electronic procedure for automating the laboratory processes related to a chromatographic analysis. It assists the user in creating an appropriate sequence with predefined associated items and a well-defined structure. An eWorkflow is typically created and configured by an administrator or lab manager in the eWorkflow Editor.

The screenshot shows the Chromleon Console interface with the following details:

- Left Sidebar:** Shows a tree view of Chromleon Local data, including Chromleon_Demonstration, Chromleon_ExtensionPack, and Demonstration Data.
- Main Window:** Displays the "Demonstration Sequence 13-03-2013 15-01" table. The table has columns: #, Chromatog, Name, Type, Level, Spike Group, Position, Volume [µl], Instrument Method, Processing Method. Rows 1-20 show entries like "None SST Check Standard 01 RA3 1.000 PROCESSING METH". Row 21 is a button: "Click here to add a new injection".
- Bottom Tab:** Associated Items table with rows: AcqMethod, Agilent_1260_Method, DEFAULT, DEFAULT, Nexera_Method.
- Right Panel:** A circular chromatogram showing various peaks labeled with letters (A1-A4, B1-B4, C1-C4, D1-D4) and numbers (1-10).

Right First Time Analyses – Sequence Readiness

To ensure that there will be no issues during a run, Chromeleon Ready Check pre-flights the instrument method, before running a sequence and check for problems.

Demonstration Sequence 14-06-2013 12-40

#	Chromatog	Name	Type	Level	Spike Group	Position	Volume (ml)	Instrument Method	P
1	None	SST	SST					NSTRUMENT MET..	R
2	None	SST	SST					NSTRUMENT MET..	R
3	None	SST	SST					NSTRUMENT MET..	R
4	None	SST	SST					NSTRUMENT MET..	R
5	None	SST	SST					NSTRUMENT MET..	R
6	None	Standard 1	Standard 1					NSTRUMENT MET..	R
7	None	Standard 1	Standard 1					NSTRUMENT MET..	R
8	None	Sample	Sample					NSTRUMENT MET..	R
9	None	Sample	Sample					NSTRUMENT MET..	R
10	None	Sample	Unknown			R45	1.000	NSTRUMENT MET..	R

Please Wait...

Waiting for Ready Check to complete. This may take a few minutes.
Pressing 'Close' now will not prevent the queue from starting.

Close

Injection Rack View <<

Sequence Readiness – Ready Check

Ready Check is executed when the sequence is started and before each injection

- Runs throughout sequence in case extra injections or sequences are queued
- Can run manually at any time – also in the Instrument Method Editor
- Calculates the amount of solvents required
- Ensures only consistent, correct sequences can be started

Ready Check Results

Ready check result: Failed.

Source	Device	Message
INSTRUMENT METHOD (Instrument Method)		The instrument method belongs to a different instrument (UltiMate3000_RS).
INSTRUMENT METHOD (Instrument Method)		Unknown symbol: UV.3DFIELD.MinWavelength

Cancel

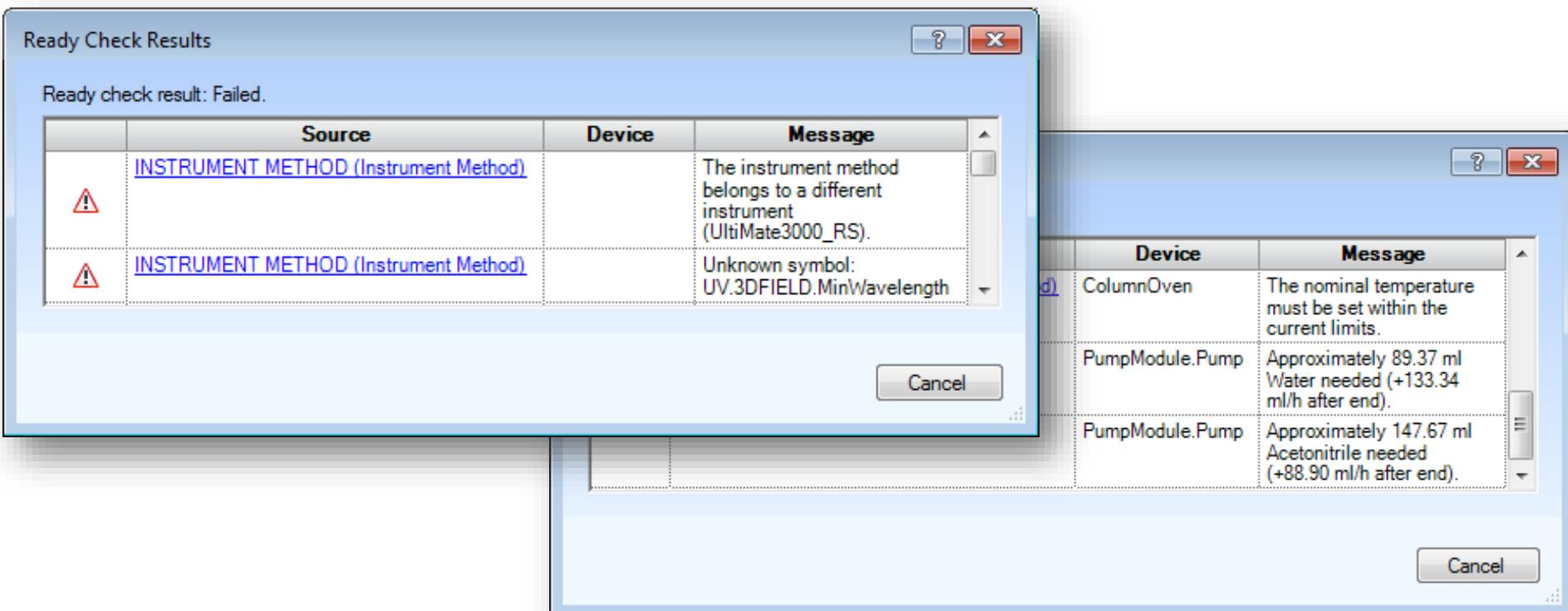
Device Message

d) ColumnOven The nominal temperature must be set within the current limits.

PumpModule.Pump Approximately 89.37 ml Water needed (+133.34 ml/h after end).

PumpModule.Pump Approximately 147.67 ml Acetonitrile needed (+88.90 ml/h after end).

Cancel



Right First Time Analyses – Instrument Readiness

Chromeleon has Smart Startup, that initializes and equilibrates the instrument with the correct chromatographic conditions before the first injection.

Demonstration Sequence 14-06-2013 12-40

New

Save Studio Print

Start

Start with Smart Startup

Smart Startup

Add to Queue with the Smart Startup option and Start Queue

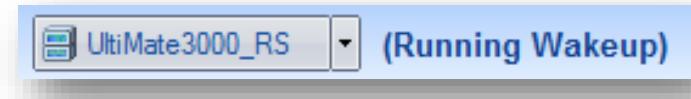
Submit

Start your sequence with "Start with Smart Startup"

#	Chromatog	Name	Spike Group	Position	Volume (μl)	Inj
1	None	SST	R41	1.000		
2	None	SST	R41	1.000		
3	None	SST	R41	1.000		
4	None	SST	R41	1.000		
5	None	SST	R41	1.000		

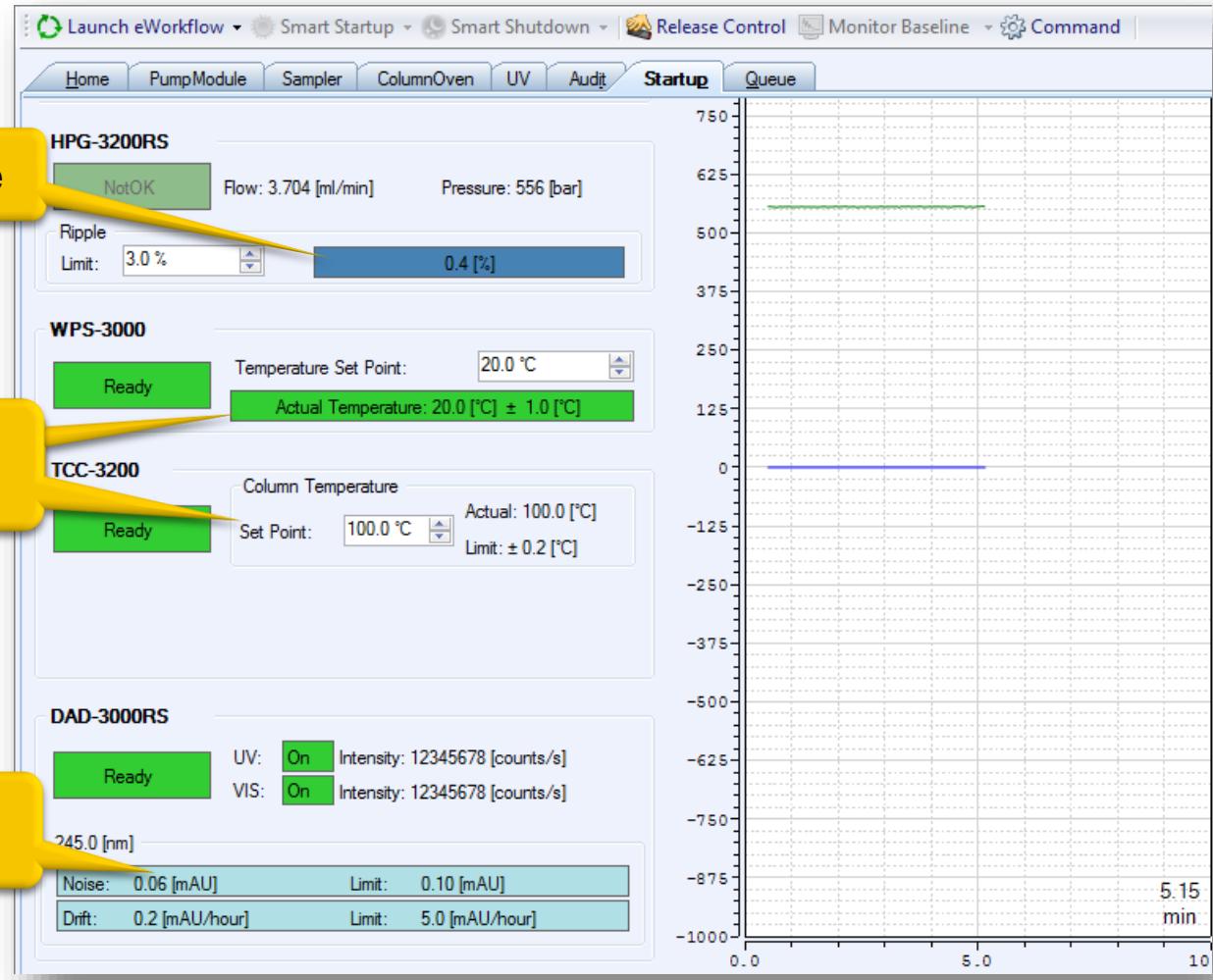
Instrument Readiness – Smart Startup

- Initial conditions are automatically extracted from Instrument Method
 - Stored inside Instrument Method – can be edited
 - Also Smart Standby and Smart Shutdown settings
- First runs instrument wake up routine
 - Ensures modules are active and connected
- Next prepares instrument
 - Purges pump, runs gradient wash, flushes system
 - Washes auto sampler
 - Switches column oven valves
 - Heats or cools temperature-controlled modules
 - Switches lamps on
 - Sets initial conditions for all modules and equilibrates



Instrument Readiness – Smart Startup

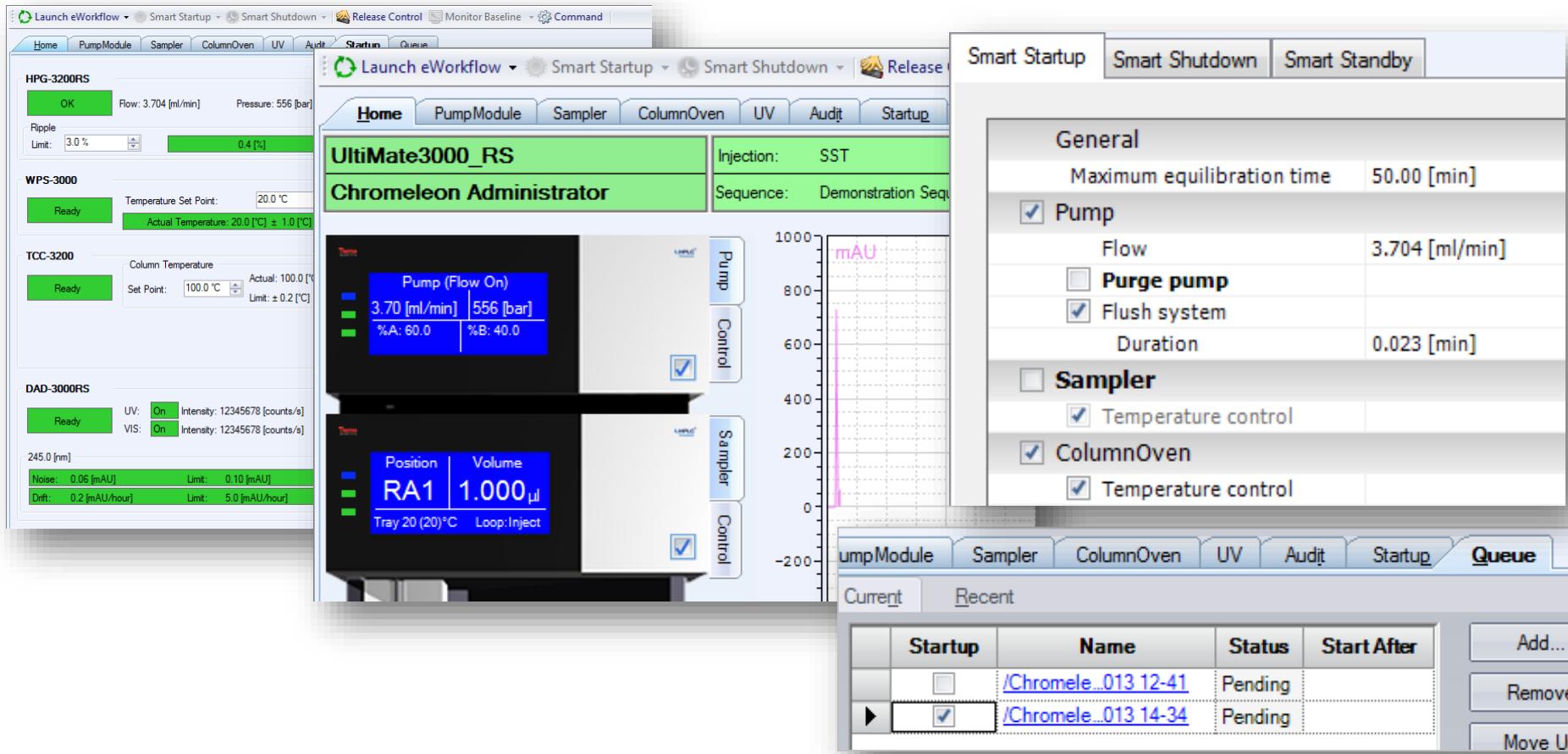
After 10 mins warming up, key parameters are monitored:



Instrument Readiness – Smart Startup

Smart Startup

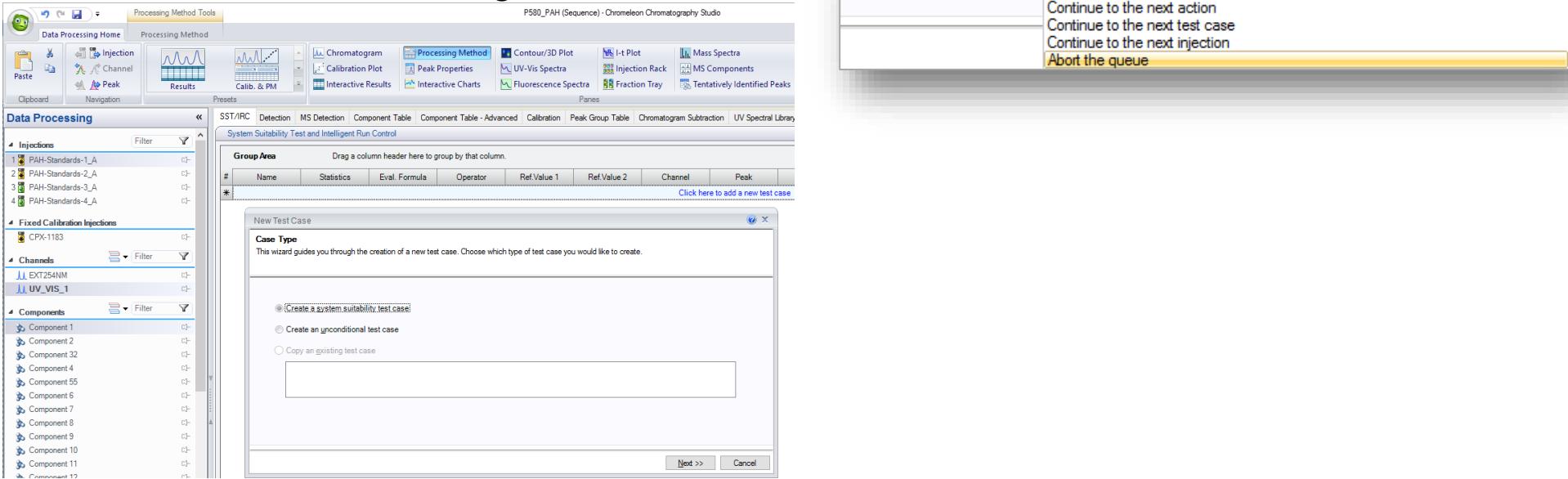
- Ensures first injection is good
- Removes subjectivity from system readiness and frees up analyst time
- Eliminates wasted solvent, sample and run time



Run-time control

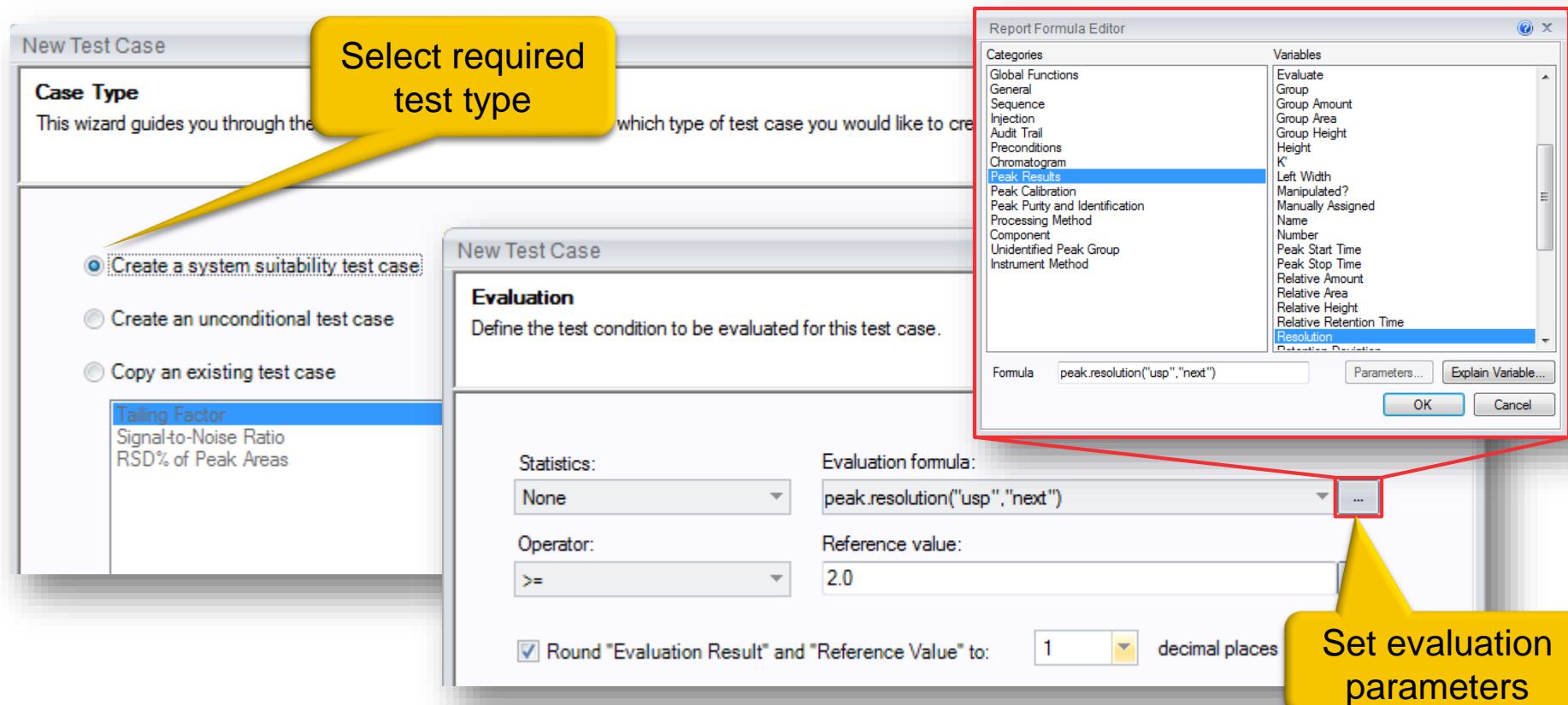
Conditional System Suitability Tests (SST) with Intelligent Run Control (IRC) in Studio Processing Method

- Allows decisions to be made during run:
 - Insert Injections
 - Re-inject
 - Perform arithmetic processes
 - Extract Channels or optimum integration from a PDA 3D field
- Can be performed Conditionally or Unconditionally
- Includes error handling



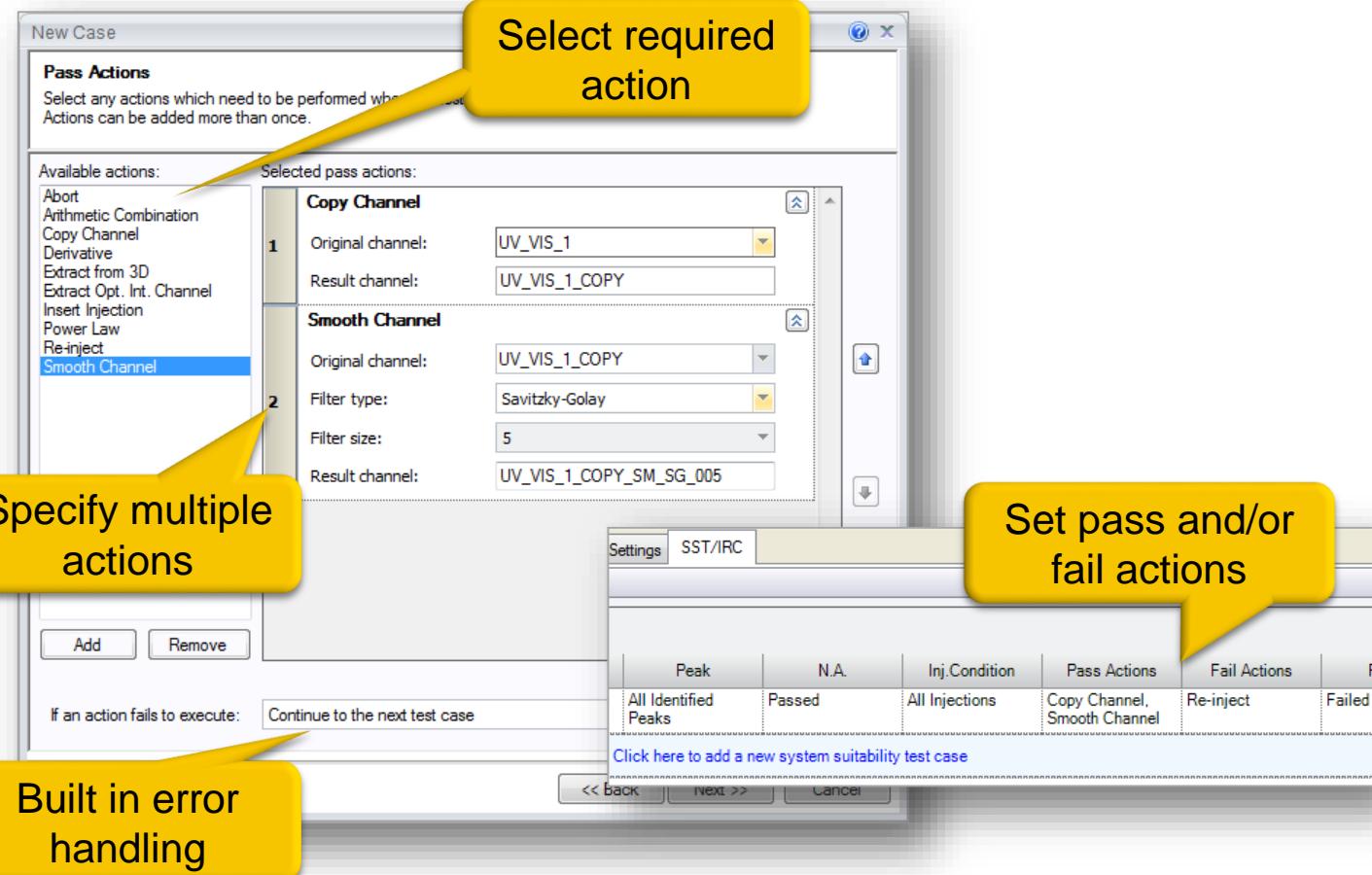
Run-time Sequence Control – IRC (Intelligent Run Control)

Any measurable parameter can be monitored and reacted to using built-in SSTs (System Suitability Tests). It can also check results against specifications.



Run-time Sequence Control – IRC

Configure pass and/or fail actions – multiple actions are possible



Run-time Sequence Control – IRC

Example: Re-inject check standards if **Relative Standard Deviation of Peak Areas** fails

The screenshot shows the Agilent Chem3D Pro software interface with the following components:

- Top Navigation:** Injection Condition, Evaluation, Peak / Channel, Pass Actions, Fail Actions.
- Evaluation Configuration:** Statistics: Relative Standard Deviation, Evaluation formula: peak.area, Operator: <, Reference value: 2.0, Round "Evaluation Result" and "Reference Value" to: 1 decimal places.
- Fail Actions Configuration:** Available actions: Abort, Arithmetic Combination, Copy Channel, Derivative, Extract From 3D Channel, Extract Opt. Int. Channel, Insert Injection, Power Law, Re-inject, Smooth Channel. Selected fail actions: Re-inject (radio button selected), Current injection, Completed injections, From most recent, Check Standard (dropdown set to block), Max. no. of re-injections: 1.
- Sequence Table:** **IRC Workshop Sequence 1** (Running). The table lists seven rows of data:

#	UV_VIS_1	Name	Type	Level	Re-injections	Position	Volume [µl]	Instrument Me
1		SST	Check Standard	01	0	R41	1.000	INSTRUMENT
2		SST	Check Standard	01	0	R42	1.000	INSTRUMENT
3		SST	Check Standard	01	0	R43	1.000	INSTRUMENT
4		SST	Check Standard	01	1	R41	1.000	INSTRUMENT
5	None	SST	Check Standard	01	1	R42	1.000	INSTRUMENT
6	None	SST	Check Standard	01	1	R43	1.000	INSTRUMENT
7	None	Standard 1	Calibration Stan...	01	0	R44	1.000	INSTRUMENT

System Suitability Tests (SST) and Intelligent Run Control (IRC) ensure you get more right first time analyses

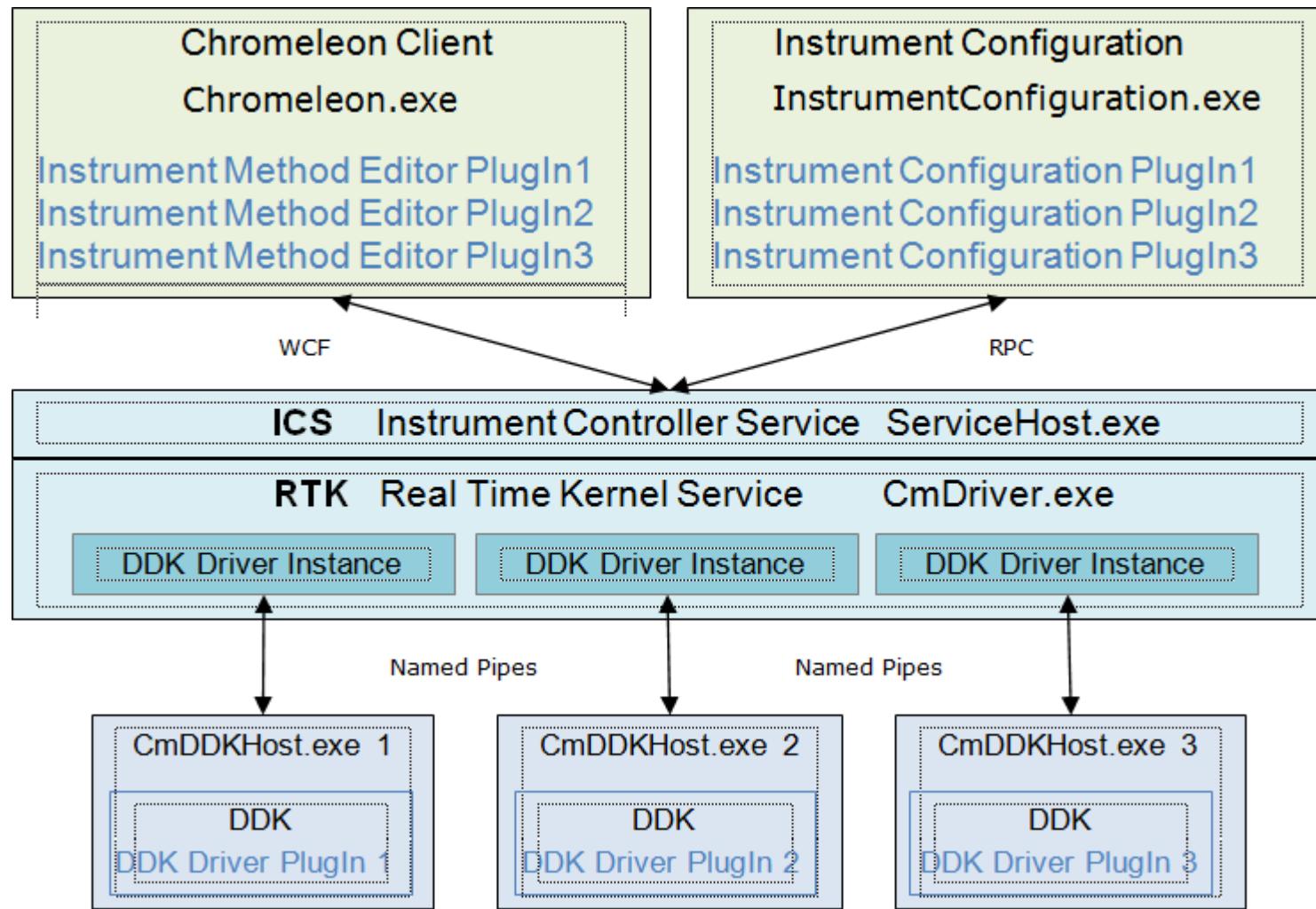
Summary – Getting It Right First Time

- eWorkflows
 - Ensure fast, easy sequence creation with correct structure, methods and reports with minimal training required.
 - Ensure compliance with your SOP's.
- Ready Check
 - Checks that every sequence will run correctly and discovers any issues before and during the run.
- Smart Startup
 - Ensures your instrument is prepared and equilibrated correctly and in a timely manner.
 - Aids method switching increasing instrument utilization.
- Intelligent Run Control
 - Automatically checks in-run chromatographic results and reacts accordingly giving confidence in results and reducing re-runs.

Day 1 Afternoon Agenda

- ❑ Driver Development Kit (DDK)
- ❑ Symbol Table
- ❑ Driver Configuration

Instrument Controller Service Details



DDK – Driver Development Kit

DDK provides a standard way for instrument implementation.

The DDK driver consists of four components:

1. Instrument Driver Configuration – software configures an instrument

2. Driver – the software controls a module. Runs as a CmDDKHost.exe process

3. Instrument Method Editor – software easily manages methods

SmartX: Start up, shutdown, standby – not available for 3rd party contributors

4. ePanels – graphically access instrument

Integrated with Thermo MS software Xcalibur (SII for Xcalibur)

DDK – Driver Development Kit

DDK V1

.NET 3.5

- **Instrument Configuration**
- **Driver** – runs as a CmDDKHost.exe process

Task Manager				
File Options View				
Processes	Performance	App history	Startup	Users
Name	PID	Status	Command line	
CmDriver.exe	13744	Running	C:\Thermo\Chromleon\Bin\CmDriver.exe	Real Time Kernel
CmDDKHost.exe	10348	Running	C:\Thermo\Chromleon\Bin\DDK\CmDDKHost.exe "MyCompany.Demo Demo" /developmode	MyCompany.Demo driver
ServiceHost.exe	13980	Running	C:\Thermo\Chromleon\Bin\ServiceHost.exe /s:data	
ServiceHost.exe	12460	Running	C:\Thermo\Chromleon\Bin\ServiceHost.exe /s:Discovery	
ServiceHost.exe	14184	Running	C:\Thermo\Chromleon\Bin\ServiceHost.exe /s:Instruments	Instrument Controller Service
ServiceHost.exe	15028	Running	C:\Thermo\Chromleon\Bin\ServiceHost.exe /s:licensing,Users	
ServiceHost.exe	12816	Running	C:\Thermo\Chromleon\Bin\ServiceHost.exe /s:Scheduler	
ServiceHost64.exe	1664	Running	C:\Thermo\Chromleon\Bin\ServiceHost64.exe /s:dataprocessing	
SharedCache.Win...	11208	Running	C:\Thermo\Chromleon\Bin\SharedCache.WinService.exe	

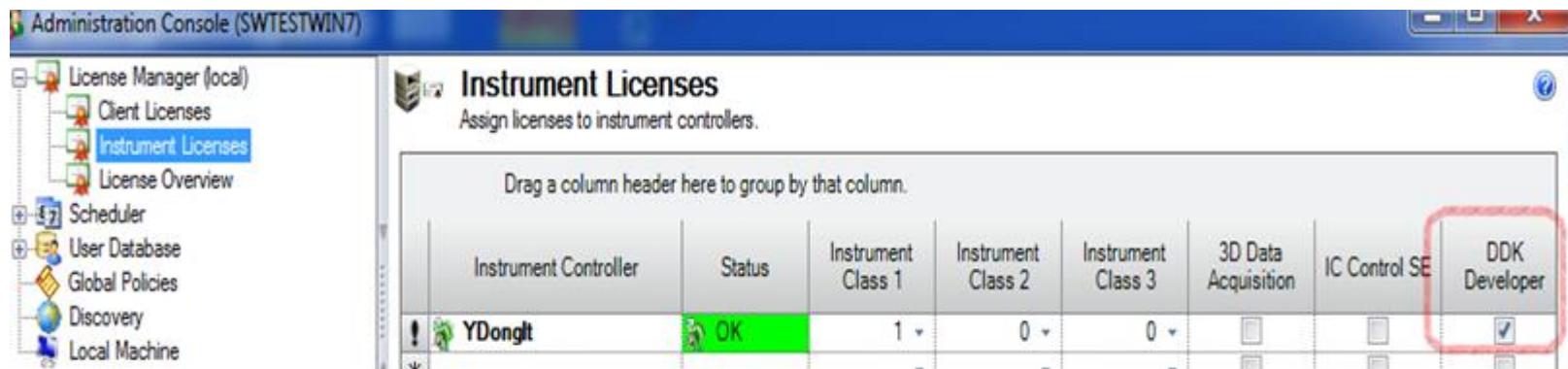
DDK V2 – Editor plug-ins

.NET 4.6.1

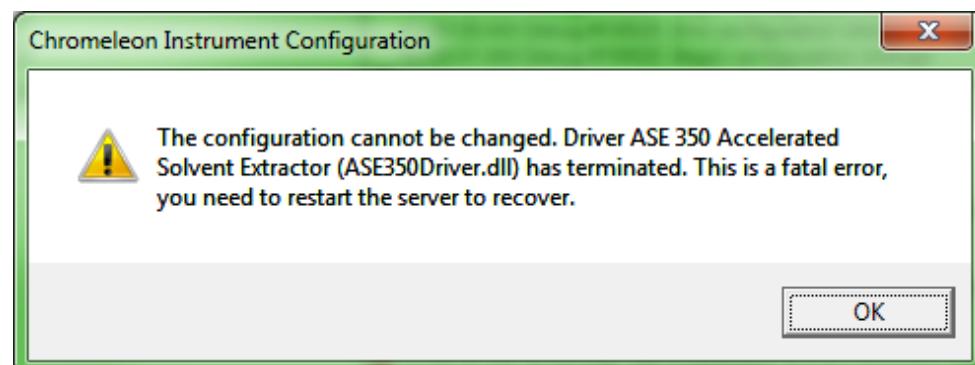
- **Instrument Method Editor**
- **ePanels**

DDK Development License

- Run AdminConsole.exe, DDK Developer license is enabled to develop a DDK driver (Developers, QA Testers, Evaluation Users)



- If the DDK development license is not enabled or the license file is missing, the DDK framework will fail to load the driver, and user cannot configure it. The terminating messages below are not obvious to understand.



DDK Development License – Driver Certification

- End-user installations do not have the **DDK Developer** license. On such systems, only certified drivers can be installed. **The preliminary certification won't work there!**
- To have your driver certified, contact Thermo Fisher Scientific
- TimebaseClass (Timebase = Instrument)
 - Class 1 – internal: Drivers developed by one of the Thermo Fisher Scientific CM software teams
 - Class 2 – Drivers developed by external teams on behalf of Thermo Fisher Scientific
 - Class 3 – Drivers developed by external software teams. Deployment and support is provided by the driver manufacturer.

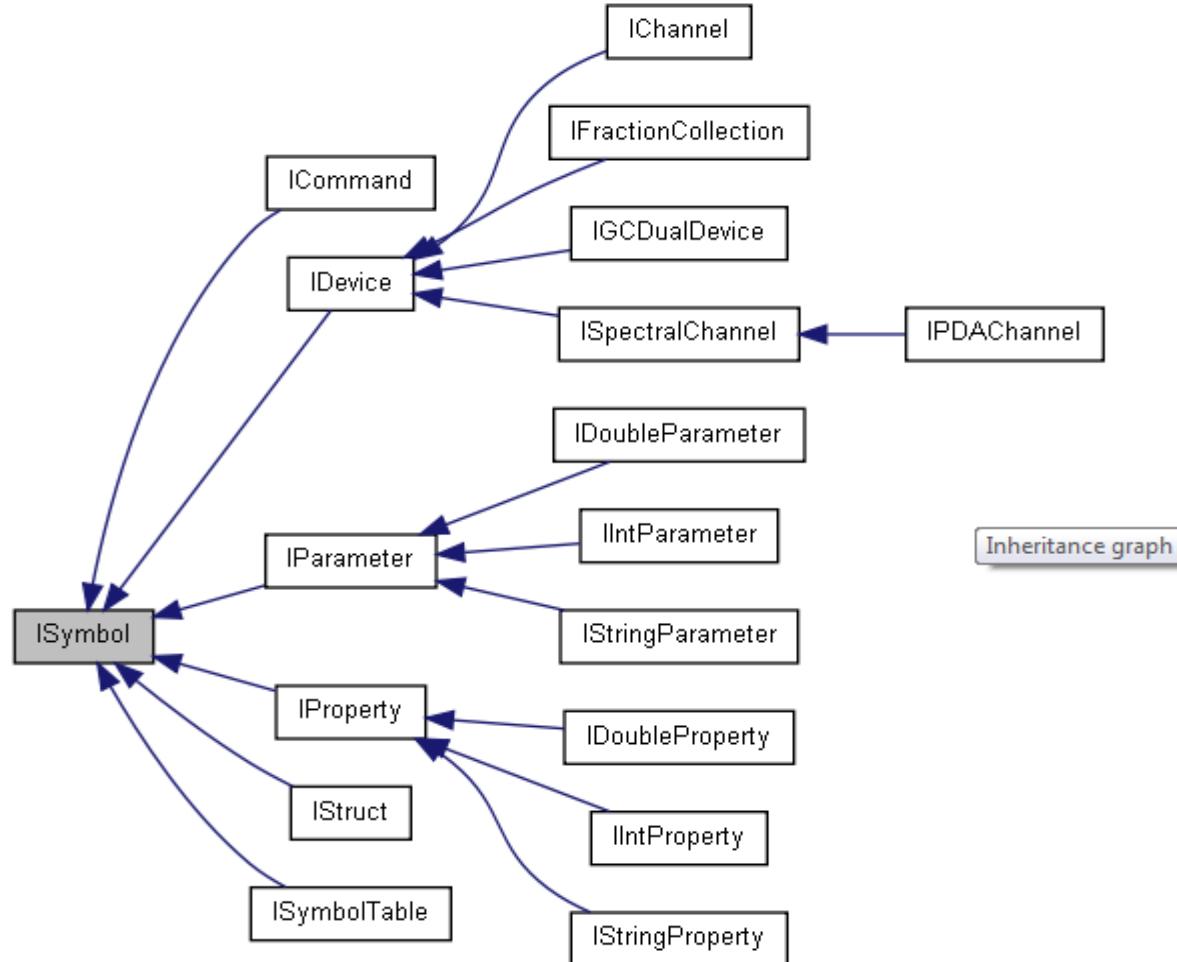
```
"C:\Thermo\Chromleon\Bin\DDK\DriverSignatureDDKV1.exe" -timebaseclass=1 -verbose=true -driverfolder="C:\Thermo\Chromleon\Bin\DDK\V1\Drivers\AAA\Demo\"
```

As part of the certification, Thermo Fisher Scientific will determine the license requirements of the driver. These will depend on the capabilities of the instrument.

Symbol Table

All clients use an instrument's Symbol Table (ST) to learn about what an instrument offers in terms of instrument status and control

Inheritance diagram for ISymbol:



Symbol Table

- The symbol table is a description of an instrument in form of a tree.
- All symbols are derived from `ISymbol`
- The root symbol is represented by `ISymbolTable`
- Devices are represented by `IDevice`
- Device Properties are represented by `IProperty`
- Device Commands are represented by `ICommand`

Symbol Table Navigation

- `ISymbol` allows navigation within the symbol tree
- To get the **child symbol** simply use `ISymbol.Child("Name")`
Can also use component notation:
`ISymbol.Child("Name.Child.GrandChild")`
- To get the **parent symbol** use `ISymbol.Parent`
- To get all **children** of a **specific type** use
`ISymbol.ChildrenOfType(SymbolType.NumericProperty)`

`ISymbol` can be cast to

`IStruct`, `IDevice`, `IStringProperty`, `INumericProperty`,
`IStringParameter`, `INumericParameter`, `ICommand`

Overview – Instrument Configuration



The Instrument Configuration Manager allows you to:

- Add, configure, modify, or delete instruments and modules.
- Save, import or export the Instrument Configuration.

C:\ProgramData\Dionex\Chromleon\InstrumentControllerConfiguration.cmic

- Check the configuration and create a report.
- Define the “Access Control” for instruments.

Preparation: Start the Instrument Controller

To work with the Configuration Manager you need to start the Instrument Controller first.



- Right-click the **Chameleon tray icon** in the Windows taskbar.
- Click “**Start Chromeleon Instrument Controller**”.

OR

- From the Chromeleon **Services Manager**.
- Click “**Start Instrument Controller**”.

OR

- In the Chromeleon **Console**, click the **Instrument** category.
- Select the **Locals** icon and click “**Start Local Instrument Controller**.”



Instrument Configuration Manager – UI

The User Interface of the Configuration Manager shows:

- The actual **Instrument Controller** – the computer that is connected, the actually configured **Sharable Interfaces** and **Instruments**.
- A tool bar, that makes it easy to configure an Instrument.

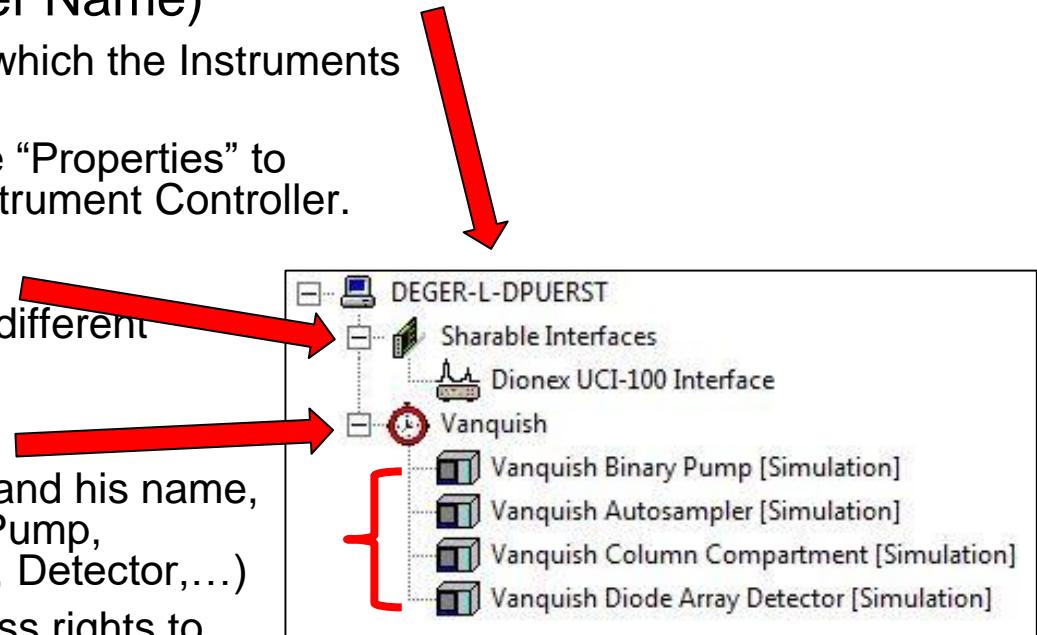


- The Instrument Controller **Audit Trail** messages.

A screenshot of the Chromelone Instrument Configuration Manager application. The window title is 'Instrument Configuration - Chromelone Instrument Configuration Manager'. The menu bar includes File, Edit, View, Controller, and Help. The toolbar has the same set of icons as the previous screenshot. The left pane shows a hierarchical tree view of the instrument setup, with 'DEGER-L-DPUERST' selected. Under it are 'Sharable Interfaces' (with 'Dionex UCI-100 Interface') and 'Vanquish' (with four simulation components: 'Vanquish Binary Pump [Simulation]', 'Vanquish Autosampler [Simulation]', 'Vanquish Column Compartment [Simulation]', and 'Vanquish Diode Array Detector [Simulation]'). A green arrow points from the toolbar to the right pane. The right pane is titled 'Messages Instrument Controller DEGER-L-DPUERST [Expert]' and displays the audit trail message: 'Instrument Controller DEGER-L-DPUERST connected. 16:02:44 User DPuersti (from DEGER-L-DPUERST) has connected Chromelone Instrument Configuration Manager to this controller.' There is an 'OK' button at the bottom right of the pane.

Instrument Configuration Manager - Objects

- PC Name (Instrument Controller Name)
 - Shows the name of the computer to which the Instruments are connected to.
 - Right click on the symbol and choose “Properties” to change the Access Control to the Instrument Controller.
- Sharable Interfaces
 - Interfaces that are used (shared) for different Instruments (A/D Interfaces)
- Instruments
 - Represents the system (Instrument) and his name, contains the different modules (e.g. Pump, Auto sampler, Column Compartment, Detector,...)
 - Right click the symbol to assign access rights to users and groups or change the Queue settings.
- Modules
 - The properties allow to specify the communication type, hardware configuration and additional options (e.g. Relays).



Driver Configuration

Main Form

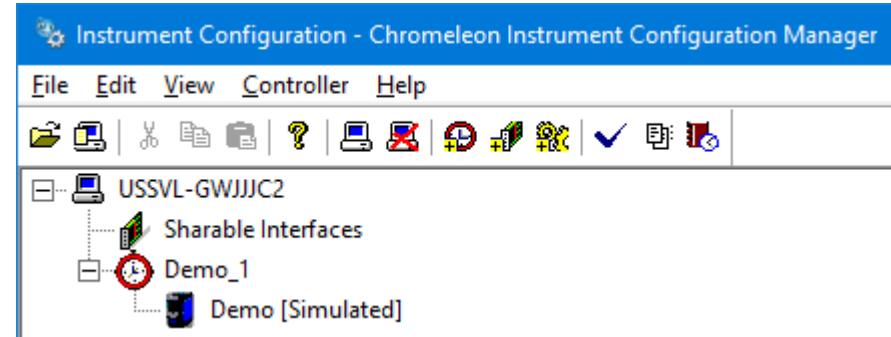
```
using Dionex.Chromeleon.DDK;

[DriverID("MyCompany.Demo")]
[CustomIcon("MyCompany.Demo.bmp")]

public partial class MainForm : Form, IConfigurationPlugin,
                               IConfigurationDescriptors
```

Xml Configuration

```
<Configuration>
  <Driver>
    </Driver>
  </Configuration>
```



The configuration and the driver assemblies must be certified:

```
"C:\Thermo\Chromeleon\Bin\DDK\DriverSignatureDDKV1.exe" -timebaseclass=1 -  
verbose=true -driverfolder="C:\Thermo\Chromeleon\Bin\DDK\V1\Drivers\AAA\Demo\"
```

Note: TimeBase = Instrument

Driver Configuration Interfaces

```
public interface IConfigurationPlugin
{
    string Configuration { get; set; }

    string ConfigurationReport { get; }

    bool ShowConfigurationDialog(IConfigDriverExchange cde);
}

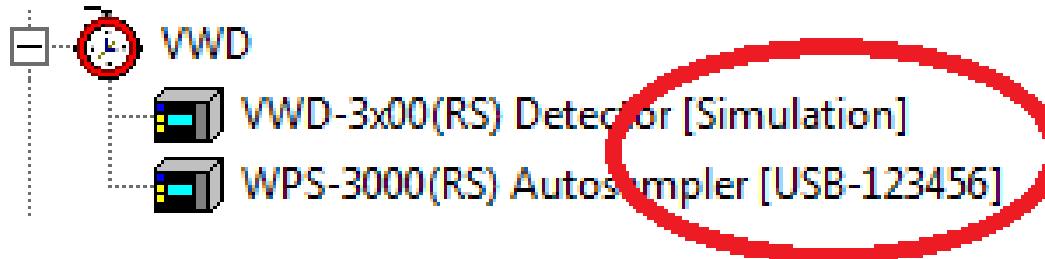
public interface IConfigurationDescriptors
{
    // This is added to text where the module name is displayed
    string ConnectInfo { get; }

    // Required for drivers that are shared among a few instruments
    string InstrumentInfo(InstrumentID Instrument);
}
```

Configuration

Adds information such as the driver USB address or whether it runs in Simulation mode

```
string IConfigurationDescriptors.ConnectInfo
{
    get
    {
        // This is added to text where the driver/module name is displayed
        return m_Driver.IsSimulated ? "Simulation" : "USB-123456";
    }
}
```

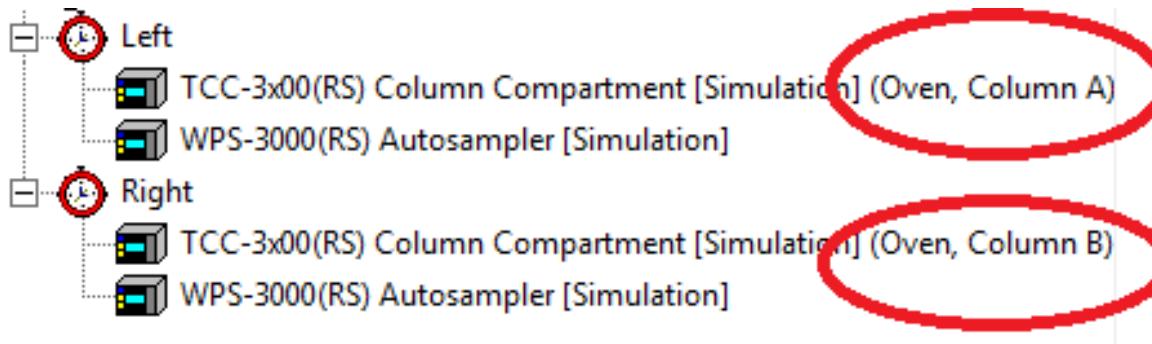


Configuration

Required for drivers that are shared among a few instruments.

Indicate which components of the module are available on which instrument

```
string IConfigurationDescriptors.InstrumentInfo(InstrumentID instrumentId)
{
    return m_Driver.UsesColumnA ? "Oven, Column A" : "Oven, Column B";
}
```



Instruments Ids are bitmapped and are limited to 16:

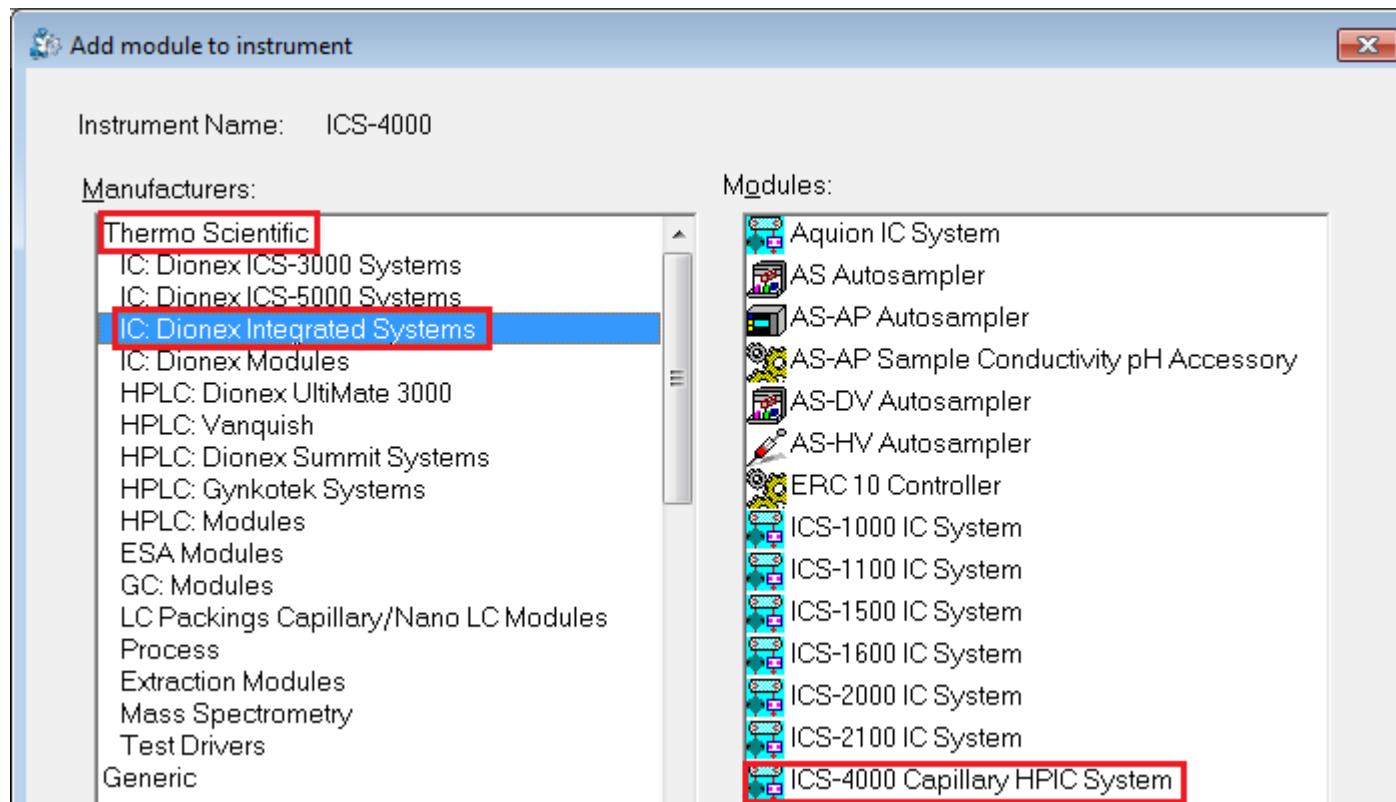
0001
0010
0100

Driver Configuration

DriverInfo.resx

The **FriendlyName** must be no more than **37 characters**

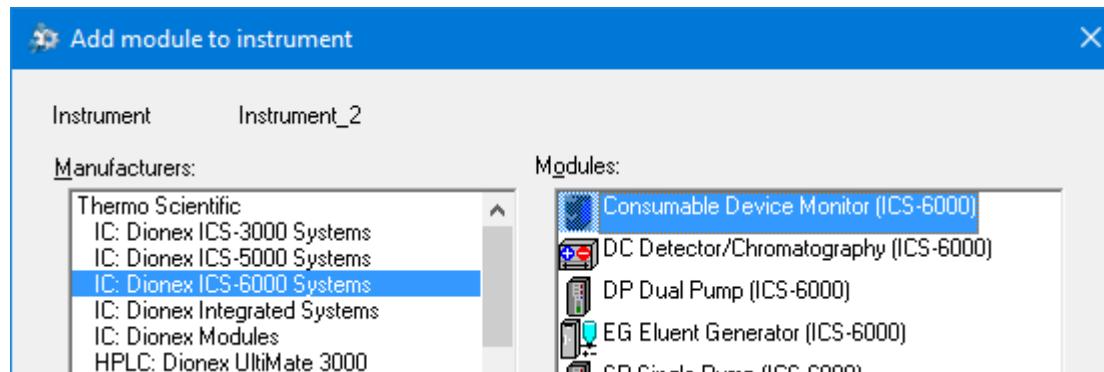
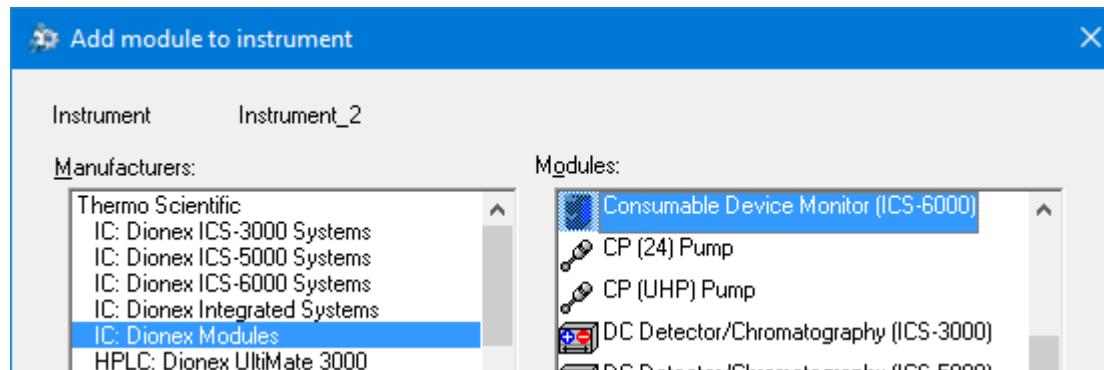
Name	Value	Comment
DeviceManufacturer	Thermo Scientific	Required string resource: Manufacturer of the hardware device being controlled by the driver.
FriendlyName	ICS-4000 Capillary HPIC System	Required string resource: Name of the driver (also used to list the driver in Instrument Configuration Manager).
ManufacturerSubgroup	IC: Dionex Integrated Systems	Optional string resource: A manufacturer-specific subgroup for the driver. If the subgroup doesn't yet exist it is created.



Driver Configuration – 2 Sub groups

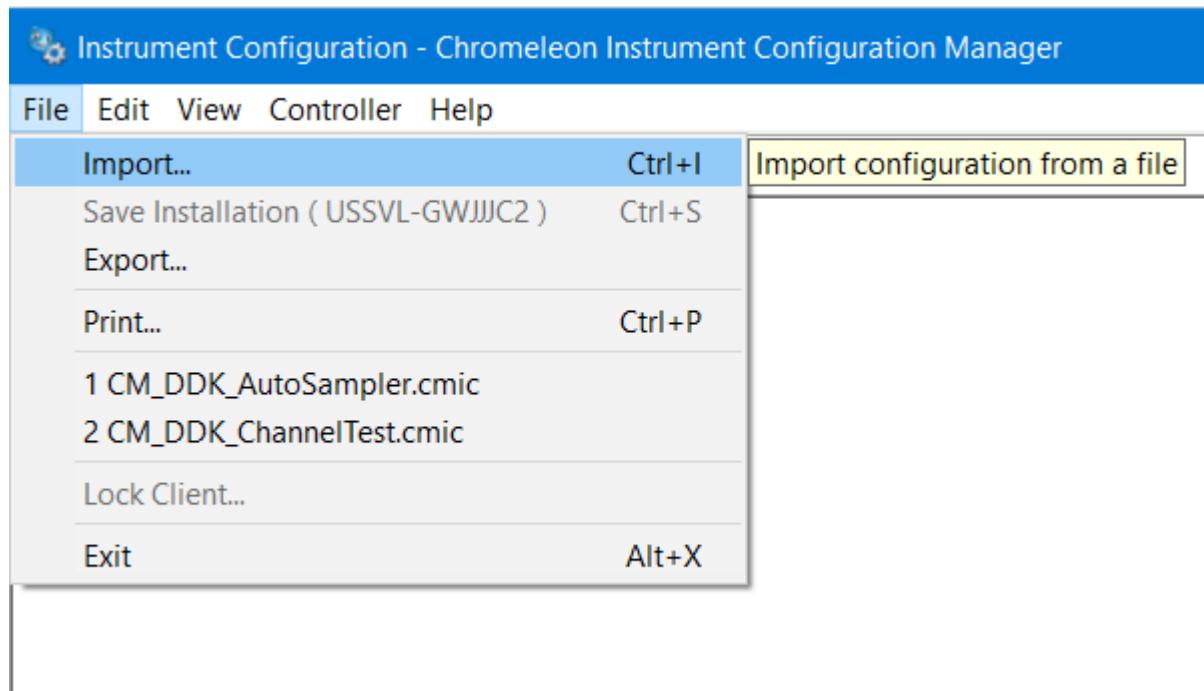
DriverInfo.resx

Name	Value
DeviceManufacturer	Thermo Scientific
FriendlyName	Consumable Device Monitor (ICS-6000)
ManufacturerSubgroup1	IC: Dionex Modules
ManufacturerSubgroup2	IC: Dionex ICS-6000 Systems



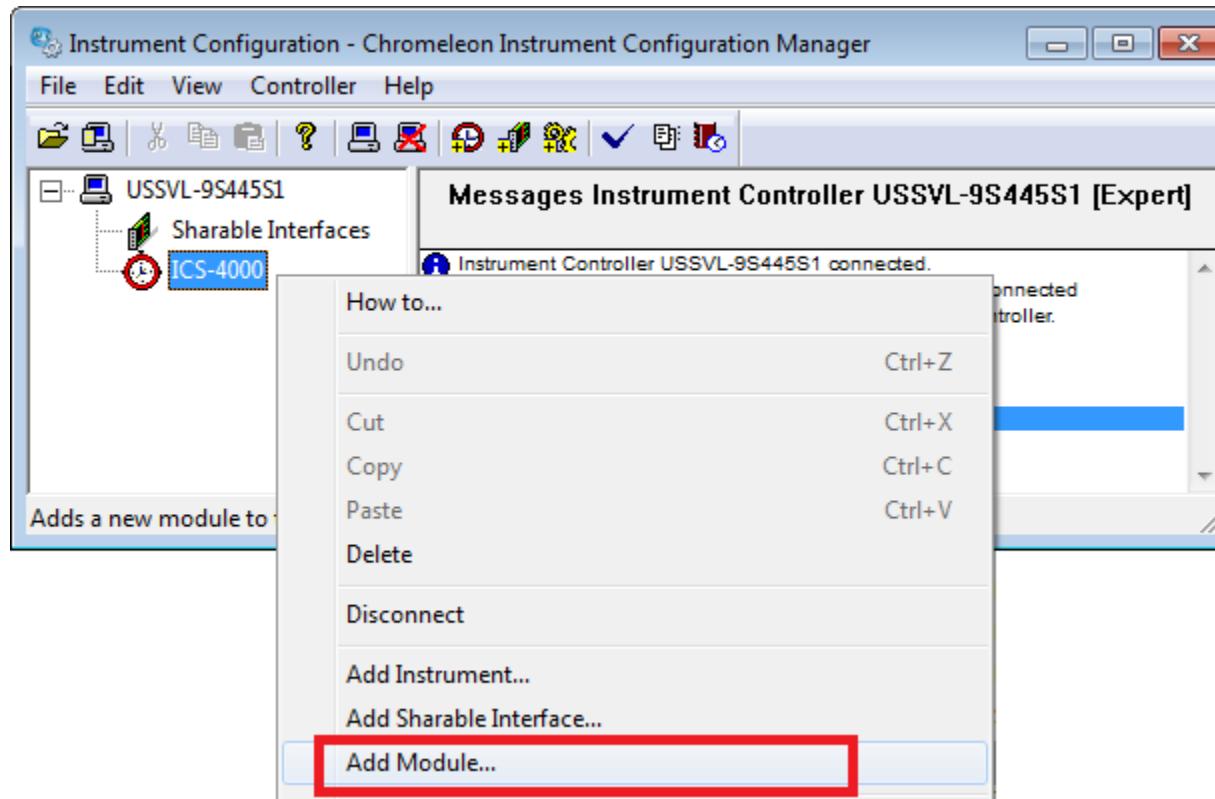
Driver Configuration – Import

- All configurations are saved in:
C:\ProgramData\Dionex\Chromleon\InstrumentControllerConfiguration.cmic
- A configuration can be imported. Warning: It will replace the current configuration!

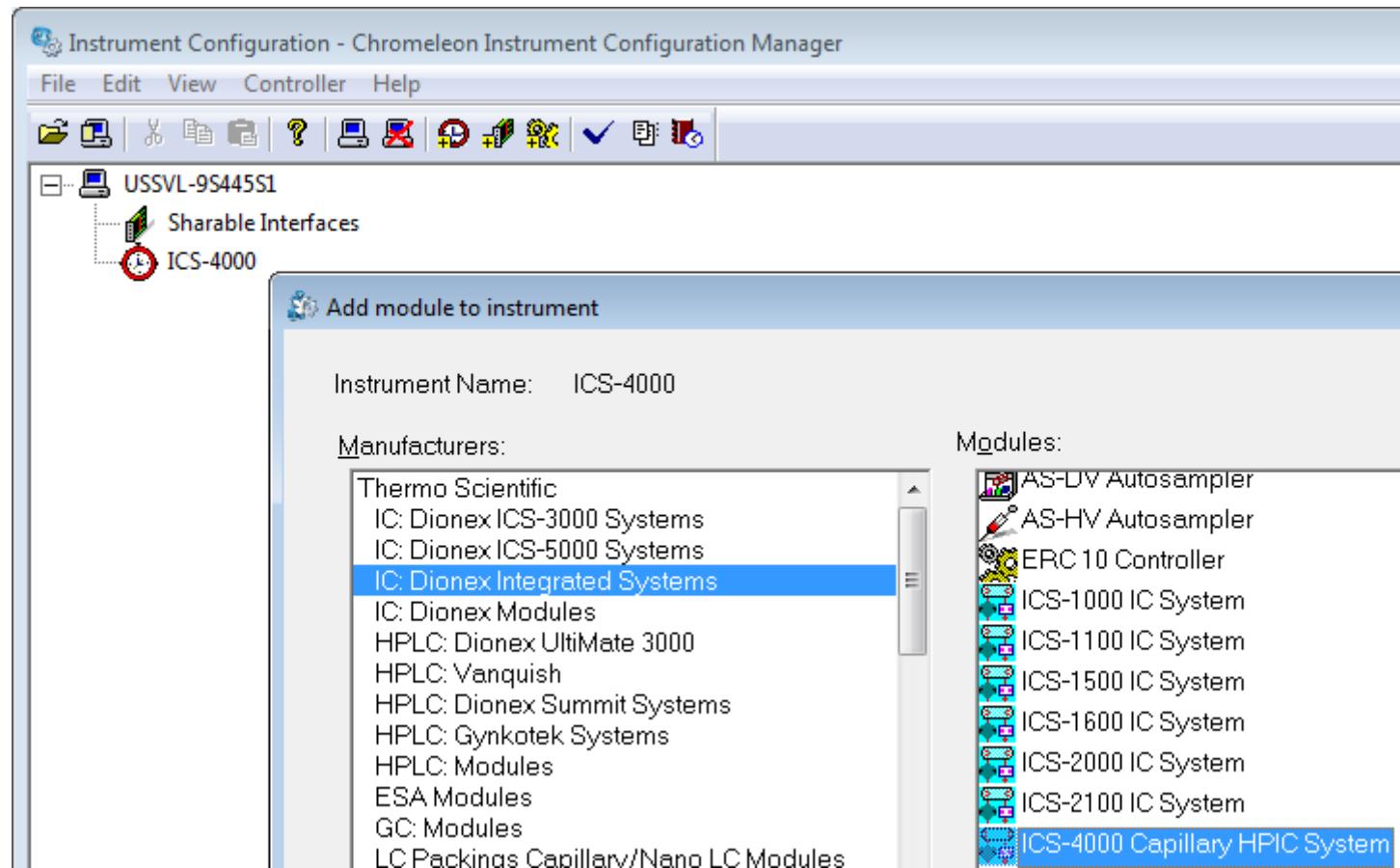


Driver Startup

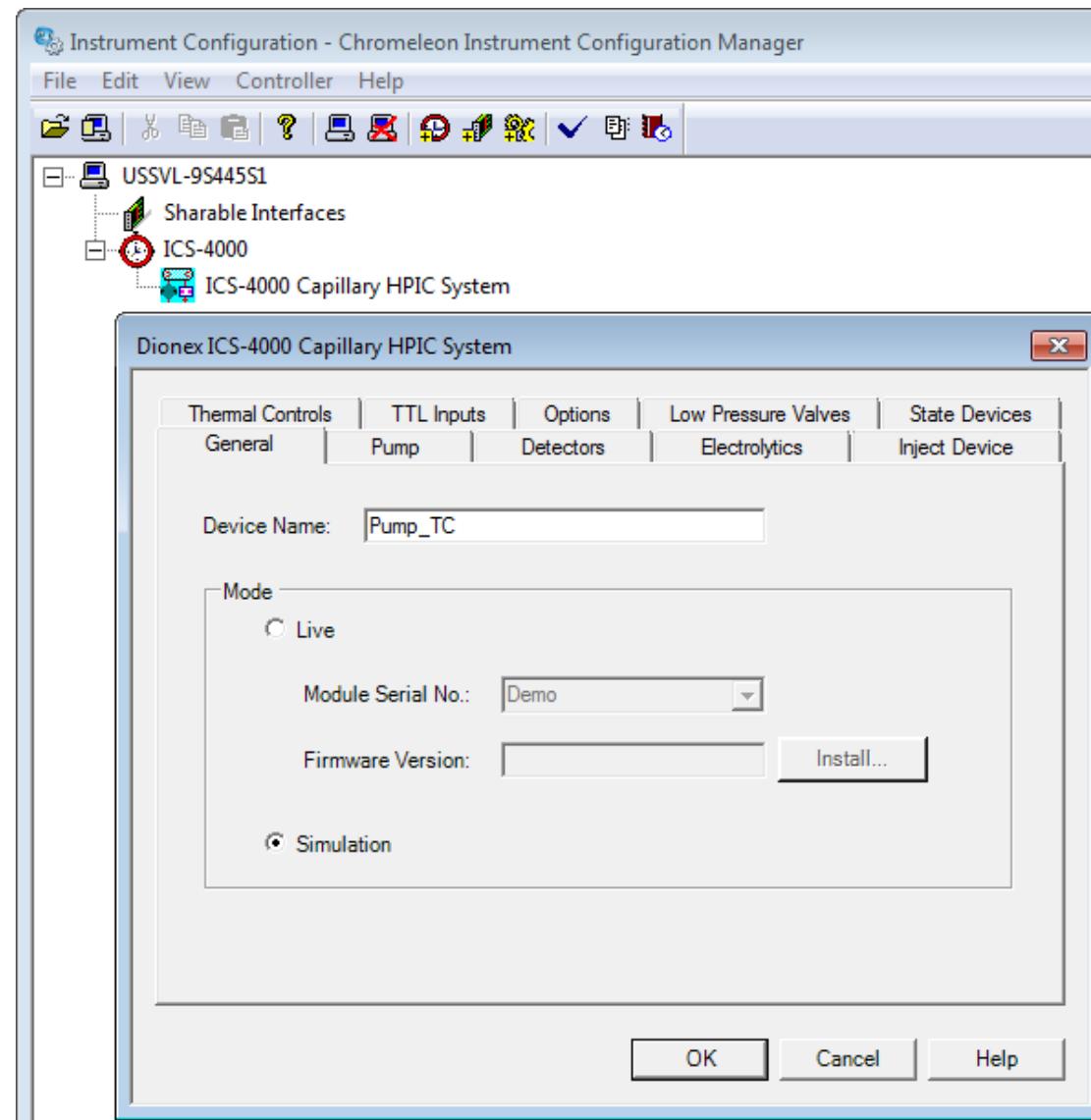
From the **ICM** (Instrument Configuration Manager), the user creates a new instrument and adds a module (driver).



Driver Startup

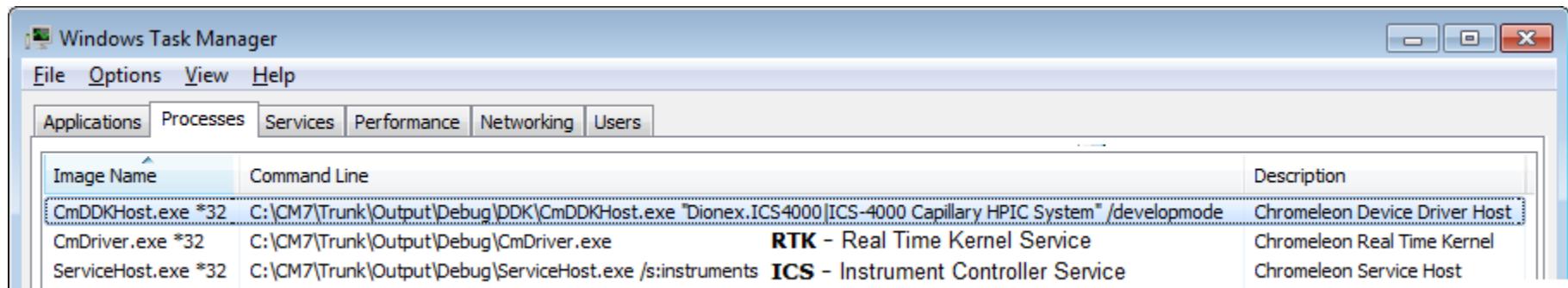


Driver Startup



Driver Startup

- **RTK** (Real Time Kernel) a service, which hosts and manages driver instances) runs a new **CmDDKHost.exe** and tells it to load the corresponding driver assembly.



The screenshot shows the Windows Task Manager window with the 'Processes' tab selected. It displays three running processes:

Image Name	Command Line	Description
CmDDKHost.exe *32	C:\CM7\Trunk\Output\Debug\DDK\CmDDKHost.exe "Dionex.ICS4000 ICS-4000 Capillary HPIC System" /developmode	Chromelone Device Driver Host
CmDriver.exe *32	C:\CM7\Trunk\Output\Debug\CmDriver.exe	RTK - Real Time Kernel Service
ServiceHost.exe *32	C:\CM7\Trunk\Output\Debug\ServiceHost.exe /s:instruments	ICS - Instrument Controller Service

- **CmDDKHost.exe** creates an instance of the type implementing **IDriver**.
- Calls **IDriver.Configuration** to get the XML string of the configuration:

```
<Configuration>
  <Driver>
  </Driver>
</Configuration>
```

Driver Startup

- This XML string is passed from the **ICT** (Instrument Controller) to the **ICM** (Instrument Configuration Manager).
- **ICM** loads the corresponding configuration plug-in assembly.
- **ICM** creates an instance of the type implementing **IConfigurationPlugin** in this assembly.

```
[DriverID("MyCompany.Demo")]
[CustomIcon("MyCompany.Demo.bmp")]
public partial class MainForm : Form,
    IConfigurationPlugin,
    IConfigurationDescriptors
```

- **ICM** calls **IConfigurationPlugin.Configuration** setter to set the XML string.
- **ICM** calls **IConfigurationPlugin.ShowConfigurationDialog**
- The driver configuration **MainForm** is now displayed.
- The **ICM** may call **IDriverSendReceive.OnSendReceive** to request information from the driver (**IDriver**). Keep in mind, that if this is a new configuration the **IDriver.Init** has not been called yet and therefore no devices are created.

Driver Startup

Communication from the **configuration** to the **driver** is implemented with
interface IConfigSendReceive

```
public partial class MainForm : Form, IConfigurationPlugin
{
    private IConfigSendReceive m_ConfigSendReceive;

    bool IConfigurationPlugin.ShowConfigurationDialog(
        IConfigDriverExchange configDriverExchange)
    {
        m_ConfigSendReceive = configDriverExchange as IConfigSendReceive;

        string response;
        m_ConfigSendReceive.SendReceive("GetIsIdle", out response);
    }
}
```

Recommendation: use XML for the command/request (to be able to add parameters) and the response (to return the requested information and an error)

Driver Startup

Communication from the **driver** to the **configuration** is implemented with interface `IDriverSendReceive`:

```
public class Driver : IDriver, IDriverSendReceive
{
    void IDriverSendReceive.OnSendReceive(IDDK ddk,
                                         string arguments,
                                         out string response)
    {
        string[] arguments = argumentsText.Split(
                                         CommonConstants.DelimiterChar);
        string commandName = arguments[0];
        if (commandName == "GetIsIdle")
        {
            response = IsIdle.ToString();
        }
    }
}
```

When the user clicks the **OK** button on the configuration form (and it closes with `DialogResult = DialogResult.OK`):

- **ICM** calls `IConfigurationPlugin.Configuration` which returns the XML string of the configured driver.
- **ICM** passes this XML string to the **ICT**.
- **RTK** unloads the `IDriver` assembly and stops **CmDDKHost.exe**
- **RTK** runs a new **CmDDKHost.exe** and loads the required driver assembly.
- **CmDDKHost.exe** creates an instance of the type implementing `IDriver`
- `IDriver.Configuration` setter is called with the **ICM** configured XML string.
- `IDriver.Init` is called. The implementation has to configure the driver with the configuration XML string passed in the previous step and create all **Devices** and **Sub-Devices**.

Instrument Configuration Report

Instrument Configuration - Chromeleon Instrument Configuration Manager

File Edit View Controller Help

USSVL-J419

Sharable
DDK_1
DDK_Si
Loc

Connect to Chromeleon Instrument Controller...
Disconnect...
Check Configuration...
Create Report...
Shows Instrument Configuration Audit Trail...

Messages Instrument Controller USSVL-J419NX1 [Experimental]
M 0.000 {Extractor} Extractor.SolventBType = "Solvent B"
M 0.000 {Extractor} Extractor.SolventAType = "Solvent A"

Report

Configuration Report

General:
Username: YDong
Date: Friday, August 04, 2017
Time: 16:03:23

Instrument Controller:
Computer: USSVL-J419NX1
Access Control: <none>

Configuration:
Sharable Interfaces:

Instruments:
DDK_Training
Access Control: <none>
Automatic upload of remote sequences: All finished sequences

SimpleDriver DDK Example
Device Name: SimpleDevice
Localization DDK Example
Configured with the generic configuration module

ASE350
Access Control: <none>
Automatic upload of remote sequences: All finished sequences

ASE 350 Accelerated Solvent Extractor
ASE 350 module

Print Save Close

Implement instrument driver and configuration