Customized Real-Time Control of Benchtop Orbitrap MS

Andreas Kühn[1], Florian Grosse-Coosmann[1], Thomas Rietpietsch[1], Jan-Peter Hauschild[1], Tim Stratton[2], Katja Tham[1], Derek Bailey[1], Robert Malek[1], Markus Kellmann[1], Christoph Henrich[1], Oliver Lange[1], Andreas Wieghaus[1], Stevan Horning[1] and Alexander Makarov[1]
[1]Thermo Fisher Scientific GmbH, Bremen, Germany; [2]Thermo Fisher Scientific, San Jose, CA

Thermo SCIENTIFIC

Overview

We developed a novel implementation of an Application Programming Interface (API), which is a software component that defines the communication between a Thermo Scientific™ Orbitrap™ benchtop mass spectrometer and an application in order to access and control the instrument at runtime. In this poster we give an introduction to the capabilities of the API and present some example applications.

Introduction

The API can be easily integrated into different scripting and programming languages, like C#, C++, VB, PowerShell® etc. as it is based on the Microsoft .NET® Framework. It comprises the commands an application can use to access and control the mass spectrometer (Figure 1).

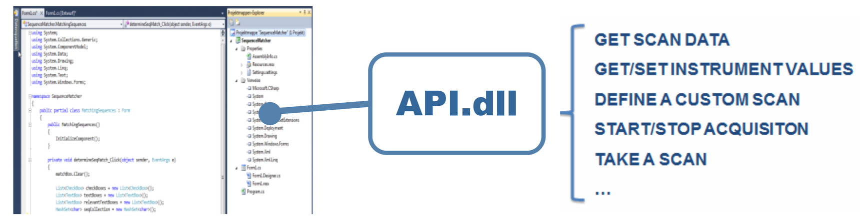


FIGURE 1. API integration into common development environments.

The novel implementation of an API allows high performance access to a benchtop Orbitrap mass spectrometers. The user application can 'listen' to scan data streams or monitor and refine instrument parameters or place user defined scans at runtime (Figure 2).

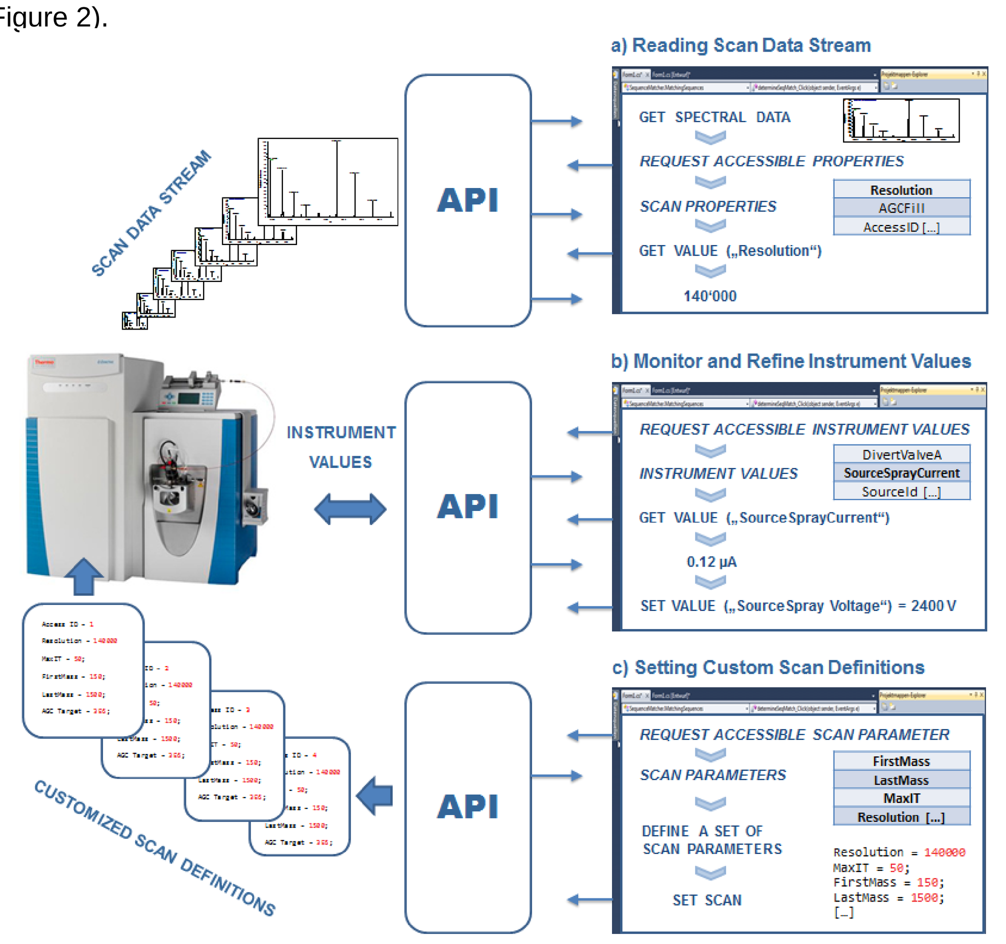


FIGURE 2. API modes of operation.

The different modes of access and control do not introduce any additional delays. A Top10 experiment (full MS followed by 10 data-dependent MS² scans) realized either by an API application or the built-in method runs at about 12Hz (Figure 3).

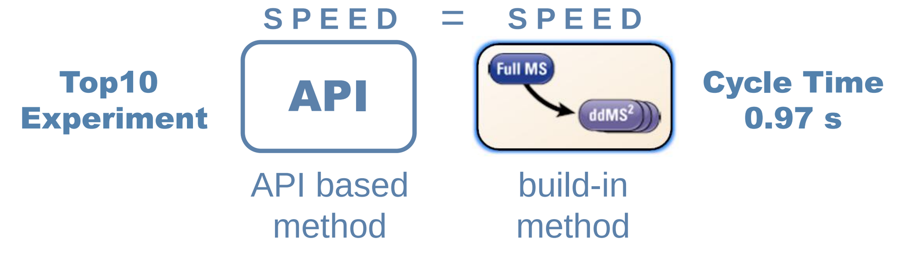


FIGURE 3. API speed comparison.

Event Driven Programming

The event-driven architecture is capable of synchronizing application workflows very simply and seamlessly. Applications can subscribe and unsubscribe to a set of API provided events at runtime. The user program will be notified whenever specific instrument values change or new scan data arrive and can react instantly, if needed (Figure 4).

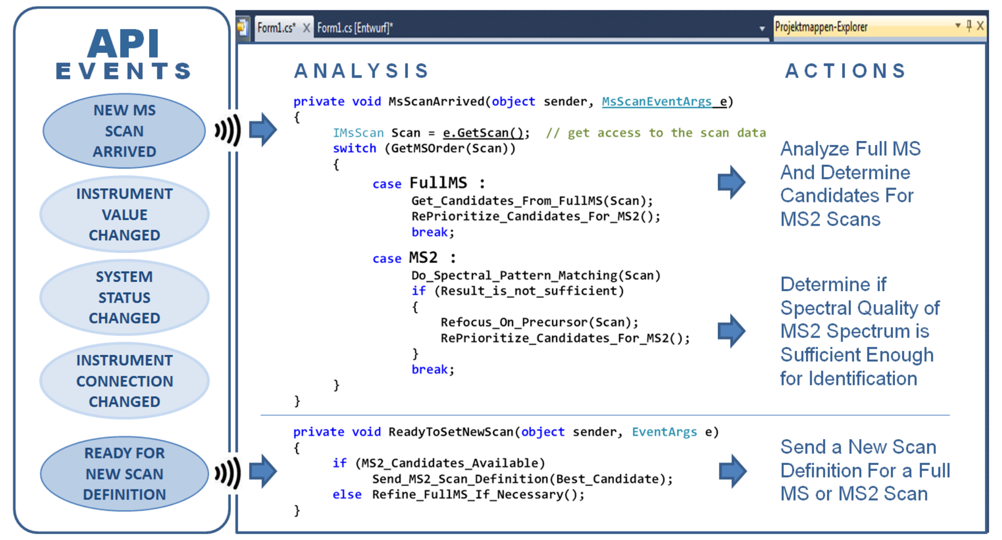


FIGURE 4. Use of API events in user applications.

Asynchronous Handling of Scan Data

Access and analysis of the incoming scan data does not affect the scan speed and can be done in parallel to controlling the instrument and setting customized scan definitions. This asynchronous approach is supported efficiently by the use of threads, which register to different events or use thread pools for data analysis (Figure 5).

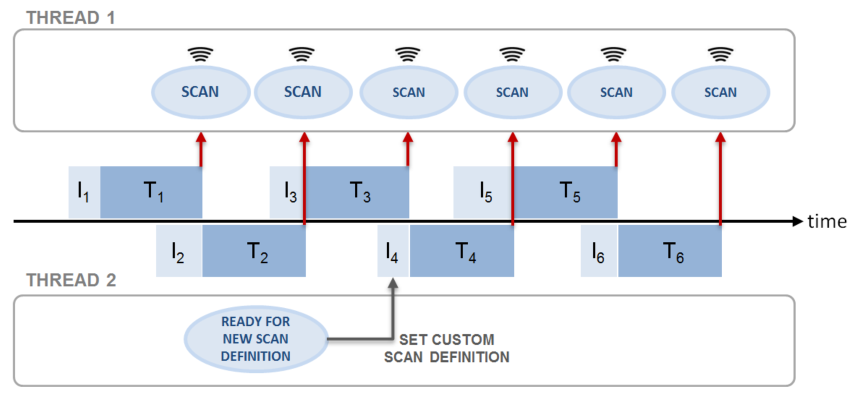


FIGURE 5. Flowchart of parallel operation. (I: Ion injection, T: Transients acquisition)

Different Level of Scanning

The provided scan hierarchy enables real-time prioritization. An application could either take over the scan control completely by defining every scan or just deploy a set of scans, when detecting special data scenarios (e.g. during a method execution, Fig. 6).

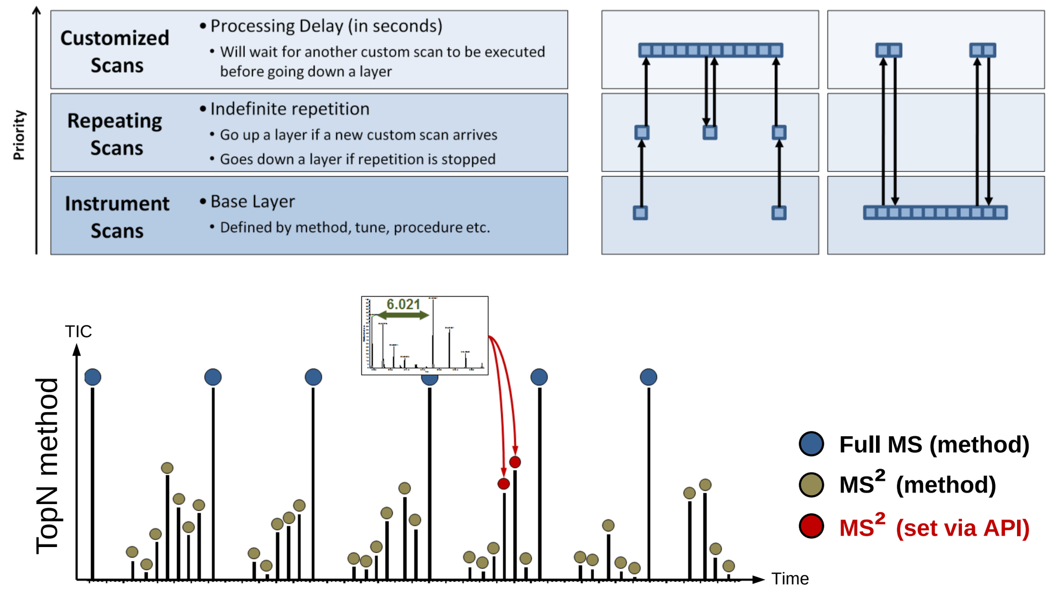


FIGURE 6. Insertion of customized scans.

Example Application: Generating quality-controlled data for mzCloud

We have developed a C# application to acquire sets of high resolution fragment scans at varying collision energies in order to populate a MS2 spectral database. High quality of data is guaranteed by monitoring the MS1 stream of an liquid chromatography-mass spectrometry (LC/MS) experiment for specific quality criteria like signal/noise, ion flux and spectral purity of a target isolation window to prevent "co-fragmentation".

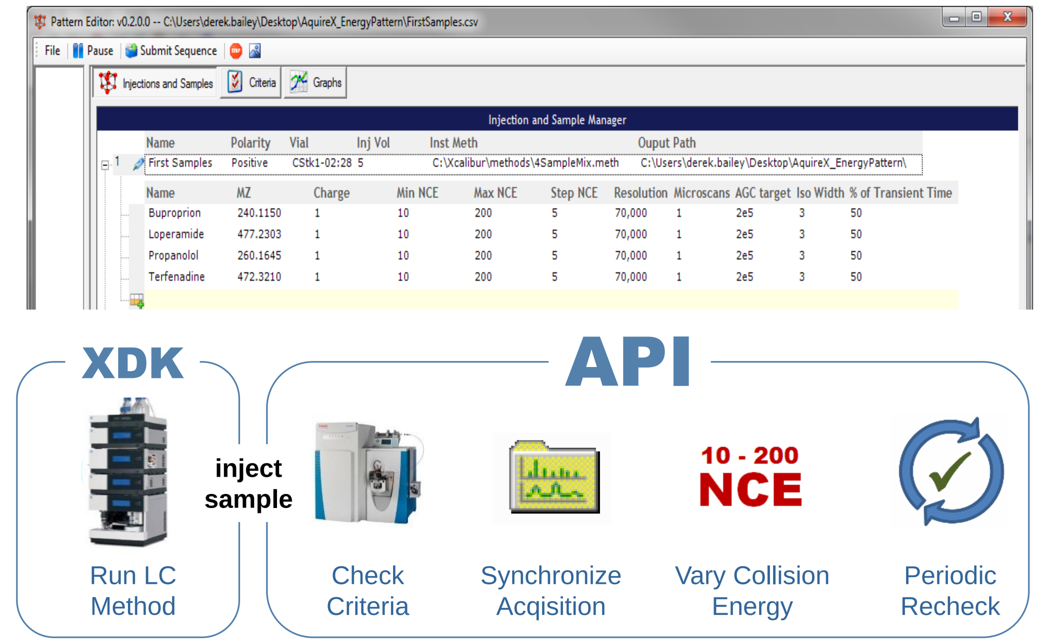
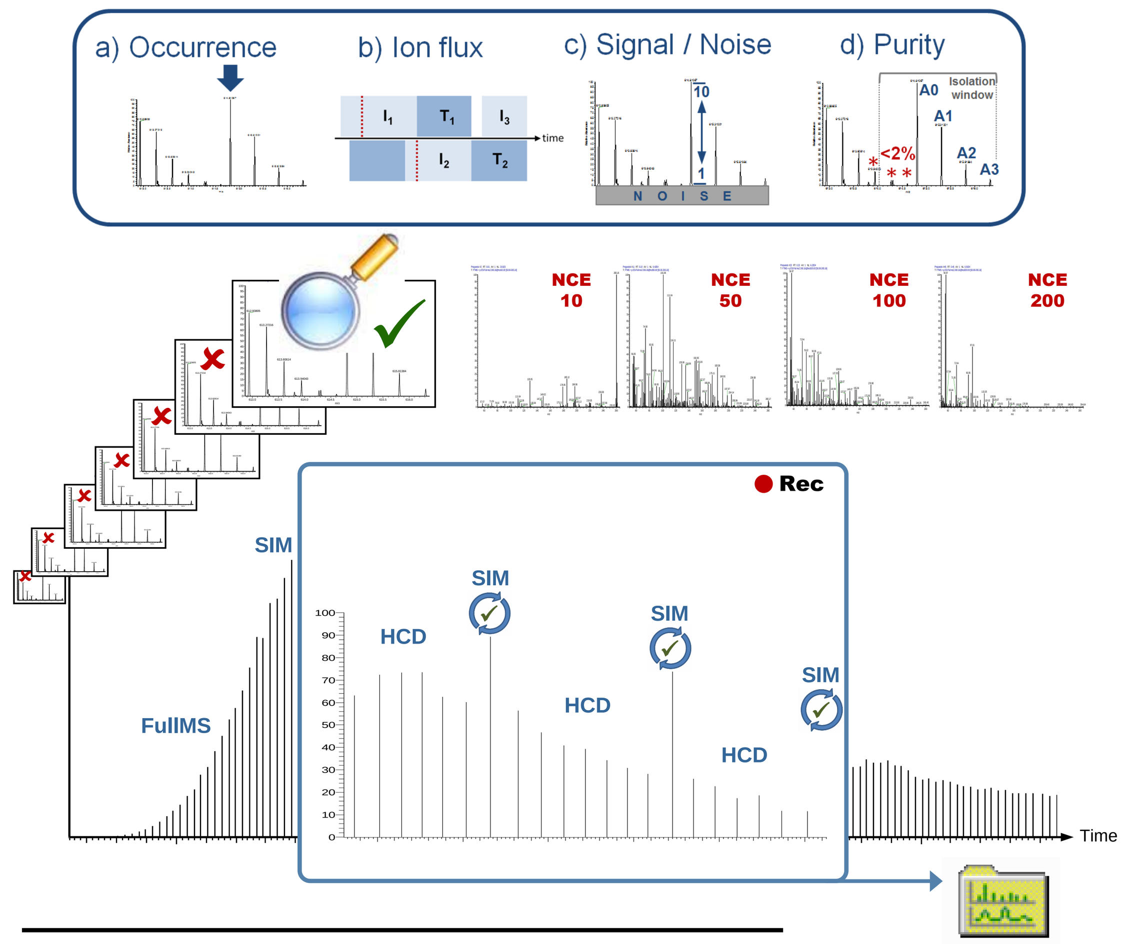


FIGURE 7. GUI and workflow of our quality controlling data collection software.

A graphic user interface (GUI) allows to set all necessary parameters for each target and per injection. The LC system is controlled by the acquisition libraries provided by the Thermo Scientific™ Xcalibur™ Development Kit (XDK). The pre-defined quality criteria are checked before and rechecked during the acquisition of the fragment spectra. The resulting data may be used to build up the cloud-based mass spectral database, mzCloud (Figures 7 and 8).



ID	Name	NCE	STATUS
1	Buproprion	10 – 200	COMPLETE
2	Loperamid	10 – 200	COMPLETE
3	Propanolol	10 – 200	COMPLETE
4	Terfenadin	10 – 110	RESUBMIT (120-200)

FIGURE 8. Real-time control via API.

Example Application: Dynamic Mass Range Optimizer

In the case of a single, high-abundant ion signal, a high target value, i.e. 3E6 can be reached within a few milliseconds inject time, contributing to the majority of the target value and thus limiting the number of ions available from low abundance species. We developed an application to boost the dynamic range in these scenarios. It constantly tracks the ion flux distribution of the Full MS and sets up multiplexed Full MS scans with different mass ranges and individual inject times, when necessary (Figure 9).

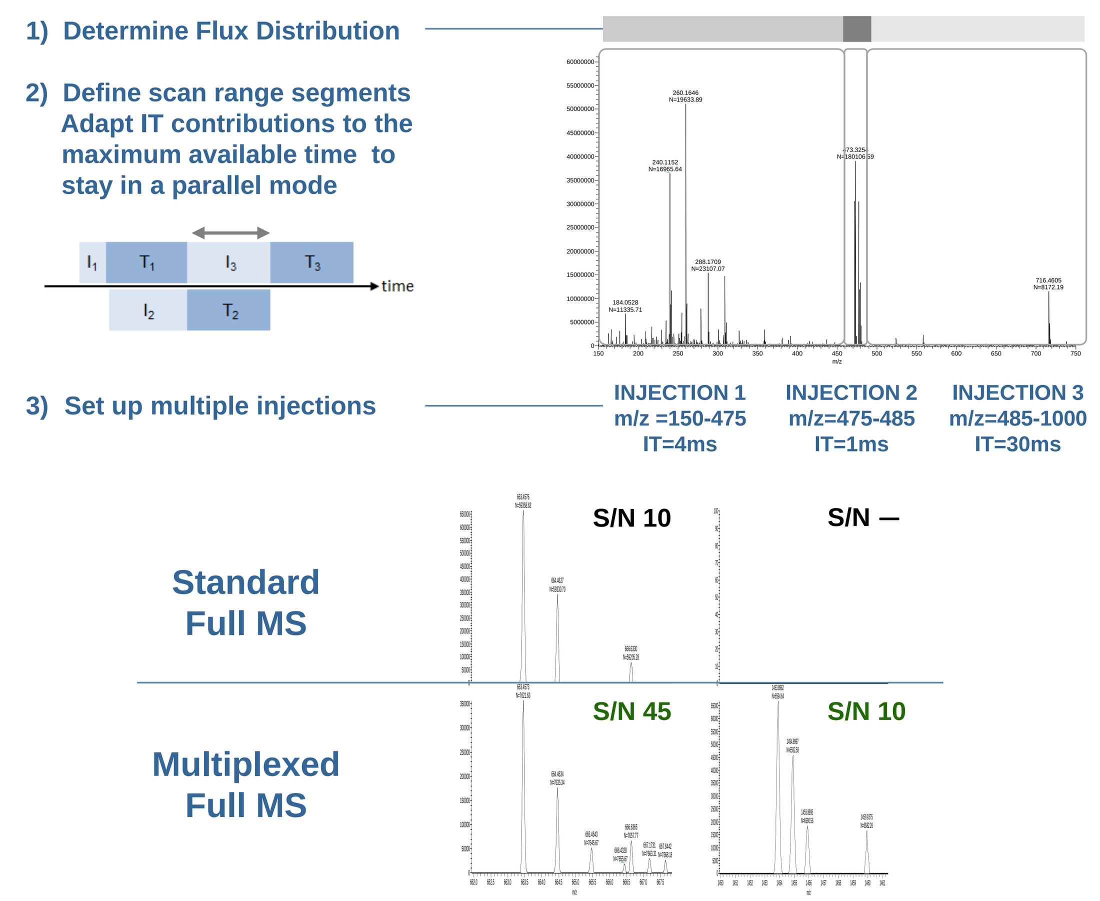


FIGURE 9. Real-time optimization of scan parameters.

Integration of API applications into service routine queues

It is possible to integrate a customized, real-time control into daily service queues by listening to the "AcquisitionStreamOpened" event. Such an event will be initiated whenever a new raw file acquisition has been started. A dynamic scan workflow application may only take over the instrument control when samples with specific IDs (filename) or vial positions, for example, are active. The rest of the submitted queue will be handled as defined, using common instrument methods (e.g. TopN experiments) (Figure 10).

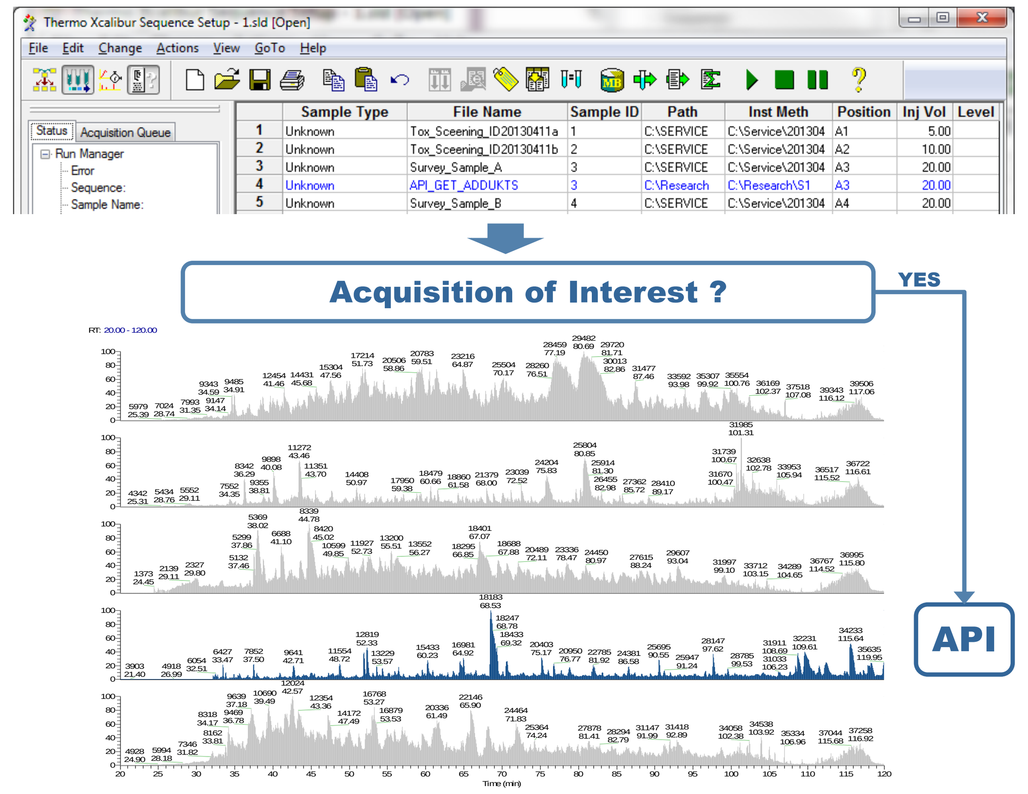


FIGURE 10. API integration into acquisition sequences.

Example Application: Tracking System Status

Another example shows the usage of Windows PowerShell® to remain updated on the instrument status and able to react instantly when problems occur. Based on the incoming API event, a customized program can interact with the operation of the instrument, or it can send a message to a recipient, e.g. the smartphone of the operator (Figure 11).

Since the PowerShell® is readily available on current Microsoft Windows® computers; the API provides a simple approach to optimize lab workflows.

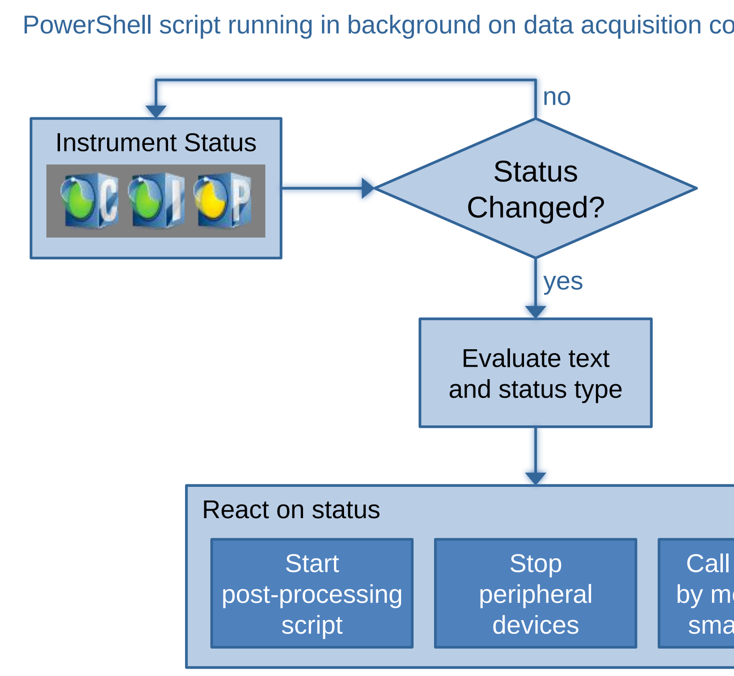


FIGURE 11. Workflow of a PowerShell® script using API to track the instrument status.

Conclusion

The novel implemented API for benchtop Orbitrap MS allows flexible control over built-in instrument methods without compromising scan speed. We have demonstrated this for various practical example applications.

Characteristics

- Easy development
- Integration using standard software development tools
- Simple connection to the instrument
- Straightforward code
- Asynchronous control due to event driven architecture

NOTE

Access to the API is not part of the standard software delivered with the instrument and requires a scpeical license agreement.

Acknowledgements

Derek Bailey contributed to the example application to generate quality-controlled data e.g. for a spectral database like mzCloud (www.mzcloud.org). He is now working at the University of Wisconsin-Madison.

[Presented at ASMS 2013]