nanalysis

NMReady JSON API Description (Version 6a)

Overview

The JSON API is one of two ways developers can programmatically control NMReady spectrometers. The second way (Microsoft .NET) is built on top of this JSON API.

This document assumes a specific knowledge level of NMReady spectrometers (e.g. auto-shimming, how to interpret a JCAMP/DX file).

Conventions

JSON is used to represent objects.

JSON discourages the use of lists since they can be dangerous to interpret in JavaScript, but the .net deserialization classes have a hard time using dictionaries. As a result, lists are used when appropriate:

```
["value0", "value1", ...]
```

Networking

All JSON messages to/from spectrometers are HTTP messages. The default port number is 5000. You will need to know the IP address of the spectrometer.
Example: http://192.168.5.122:5000/interfaces/iStatus/OperationalMessages

Enabling the API on a Spectrometer

To enable PUTs on /interfaces/iFlow and /interfaces/iStatus, check config/pygui.py to make sure that RPC_API_ENABLED = True (add this line if it does not exist).

Then, restart the GUI (if you had to change pygui.py) then go to:
Setup Button > System > Remote and click "Enable".

Testing

This API has been tested using the Google Chrome browser and Boomerang, a REST client implemented as a Chrome extension. It has also been tested via a C# Microsoft .NET application that implements the JSON API within a translation layer. On top of the translation layer is a test client.

Methods

URL	GET	PUT[1]
/interfaces/iStatus/OperationalMessages	x	
/interfaces/iStatus/SpectrometerStatus	x	
/interfaces/iStatus/PingSpectrometer	x	
/interfaces/iStatus/StandbyMode	x	x
/interfaces/iStatus/RpcEnabled	x	
/interfaces/iStatus/StartupTestStatus	x	
/interfaces/iStatus/Solvents	x	
/interfaces/iStatus/Solvents/<int:group_id>	x	
/interfaces/iFlow/ExperimentSettings	x	x
/interfaces/iFlow/RunExperiment	x	x
/interfaces/iFlow/CancelExperiment		x
/interfaces/iFlow/CalibrateSolvent	x	x
/interfaces/iFlow/ExperimentStatus	x	
/interfaces/iFlow/PeakParameters	x	x
/interfaces/iFlow/ManualIntegrals	x	x
/interfaces/iFlow/Shim	x	x
/interfaces/iFlow/Settings/1D	x	x

1. PUTs to /interfaces/iStatus and /interfaces/iFlow are protected, and will return the message below if not the API is not enabled

   403 Forbidden:
   Core Connected: True
   RPC Enabled: False

   Enable the API/RPC by pressing the **Enable** button on the **Setup Button > System > Remote** tab.

Method Descriptions

/interfaces/iStatus/OperationalMessages

Use this method to query any system recommendations, warnings or errors.

GET

Request

```
{ }
```

Response

```
{
 "messages": [
  {
   "Message": "Run an autoshim (full)",
   "Type": "Autoshim"
  }
 ]
}
```

Note: This method is heavily language dependent and will change in the near future to become language independent.

/interfaces/iStatus/SpectrometerStatus

Use this method to check the current status of the spectrometer. The values below are mostly represented on the status screen in the touch screen application.

GET

Request

```
{ }
```

Response


```
{
 "Drift": 0.0008557449111069529,
 "FirmwareVersion": "9.9.8",
 "Resolution": {
  "LineWidths": [
   {
    "Threshold": 5.0,
    "Width": 6.0272216796875
   },
   {
    "Threshold": 10.0,
    "Width": 4.1961669921875
   },
   {
    "Threshold": 50.0,
    "Width": 1.373291015625
   }
  ],
  "TimeStamp": "Tue Apr  7 10:09:07 2015"
 },
 "Sensors": {
  "ControlBoardTemperature": 36.0,
  "EnclosureTemperature": 28.100000381469727,
  "MagnetTemperature": 29.100000381469727
 },
 "SerialNumber": "mark12-04",
 "SoftwareVersion": "1.1.5 - 2851M",
 "SpectrometerFrequency": 60000133.12634938,
 "StandbyMode": false,
 "TimeStamp": "Tue Apr  7 10:14:43 2015"
}
```


Note: Linewidths are only in the dictionary if there has been an auto-shim run

/interfaces/iStatus/PingSpectrometer

Use this method to see if you can communicate with the spectrometer.

GET

Request

```
{ }
```

Response

```
{
 "connected": true
}
```

/interfaces/iStatus/StandbyMode

Use this method to turn standby mode on and off. Standby mode should be turned on whenever the spectrometer is idle for long periods of time. Within standby mode, auto-shimming is performed on a scheduled basis.

GET

Request

```
{ }
```

Response

```
{
  "StandbyMode": true
}
```

PUT

Request

```
{
   "StandbyMode" : true
}
```

Response

```
{
   "ResultCode" : 0
}
```

Result Codes

Value	Description
0	Succeeded
1	Failed

Note
If standby mode is turned on via the API, it will be turned off when the spectrometer is manually taken out of RPC/API mode.

/interfaces/iStatus/RpcEnabled

This method can be called when the spectrometer does not have remote calls enabled. In other words, you can check to see if the API (aka RPC) is enabled at any time. RPC stands for “Remote Procedure Calls”.

GET

Request

```
{ }
```

Response

```
{
  "RpcEnabled": false
}
```

/interfaces/iStatus/StartupTestStatus

Startup tests run when the spectrometer powers up. These tests can be lengthy, especially if the magnet is cold and needs to warm up.

GET

Request

```
{ }
```

Response

```
{
  "ResultCode": 1,
  "PercentComplete": 50,
  "Message": "Waiting for magnet temperature"
}
```

Note: This method will change in the near future to become language independent.

Result Codes

Value	Description
0	Startup tests completed
1	Startup tests are still running

/interfaces/iStatus/Solvents

Lists the solvent groups and solvents that have been configured on the instrument.

GET

Request

```
{ }
```

Response

```
{
"SolventGroups": [
      {
         "name": "(1H) Hydrogen",
         "solvents": [
            "D2O",
            "DMSO-d6",
            "Chloroform-d",
            "Methanol-d4",
            "Acetone-d6",
            "Acetonitrile-d3",
            "Benzene-d6",
            "TFA-d",
            "Ethanol-d6",
            "THF-d8"
         ]
      },
      {
         "name": "(13C) Carbon",
         "solvents": [
            "D2O",
            "Acetone-d6",
            "Chloroform-d",
            "DMSO-d6"
         ]
      }
   ]

}
```

/interfaces/iStatus/Solvents/<int:group_id>

Gets the specific solvents in a solvent group.
E.g. /interfaces/iStatus/Solvents/0

GET

Request

```
{ }
```

Response

```
{
 "name": "(1H) Hydrogen",
 "solvents": [
  "D2O",
  "DMSO-d6",
  "Chloroform-d",
  "Methanol-d4",
  "Acetone-d6",
  "Acetonitrile-d3",
  "Benzene-d6",
  "TFA-d",
  "Ethanol-d6",
  "THF-d8"
 ]
}
```

Requesting a solvent group that does not exist will return:

```
{
 "name": "() ",
 "solvents": []
}
```

/interfaces/iFlow/ExperimentSettings

These setting are general and apply to all experiments.

GET

Request

```
{ }
```

Response


```
{
 "ActiveTimeScanInSeconds": 2.5559999644756317,
 "Apodization": 0.20000000298023224,
 "DigitalResolutionInHz": 0.0762939453125,
 "Experiment": 1,
 "NumberOfPoints": 2048,
 "NumberOfScans": 1,
 "PeakIntegrationMethod": 0,
 "PulseWidthInMicroseconds": 16.628877639770508,
 "ReceiverGain": 14,
 "ScanDelayInSeconds": 0.0,
 "Solvent": 8,
 "SolventGroup": 0,
 "SpectralCentreInPpm": 5.0,
 "SpectralWidthInPpm": 22.0,
 "TimePerScanInSeconds": 2.5559999644756317,
 "TotalDurationInSeconds": 2.5559999644756317,
 "ZeroFillingFactor": 7.0
}
```


A description of all general experiment settings are found in the appendix.

/interfaces/iFlow/ExperimentSettings (continued)

PUT

Request


```
{
 "ActiveTimeScanInSeconds": 2.5,
 "Apodization": 0.2,
 "DigitalResolutionInHz": 0.07,
 "Experiment": 1,
 "NumberOfPoints": 2048,
 "NumberOfScans": 1,
 "PeakIntegrationMethod": 0,
 "PulseWidthInMicroseconds": 16.6,
 "ReceiverGain": 14,
 "ScanDelayInSeconds": 0.0,
 "Solvent": 8,
 "SolventGroup": 0,
 "SpectralCentreInPpm": 5.0,
 "SpectralWidthInPpm": 22.0,
 "TimePerScanInSeconds": 2.5,
 "TotalDurationInSeconds": 2.5,
 "ZeroFillingFactor": 7.0
}
```


See appendix A for details on general experiment settings.

Response


```
{
 "ResultCode": 0
}
```


Result Codes

Value	Description
0	Succeeded, values updated
1	Failed

/interfaces/iFlow/RunExperiment

This method starts an experiment using the current settings. The active parameters are returned in the response.

PUT

Request

```
{}
```

Response


```
{
 "ExperimentNumber": 1,
 "ResultCode": 1,
 "Settings": {
  "ActiveTimeScanInSeconds": 2.5559999644756317,
  "Apodization": 0.20000000298023224,
  "DigitalResolutionInHz": 0.0762939453125,
  "Experiment": 1,
  "NumberOfPoints": 2048,
  "NumberOfScans": 1,
  "PeakIntegrationMethod": 1,
  "PulseWidthInMicroseconds": 16.628877639770508,
  "ReceiverGain": 14,
  "ScanDelayInSeconds": 0.0,
  "Solvent": 8,
  "SolventGroup": 0,
  "SpectralCentreInPpm": 5.0,
  "SpectralWidthInPpm": 22.0,
  "TimePerScanInSeconds": 2.5559999644756317,
  "TotalDurationInSeconds": 2.5559999644756317,
  "ZeroFillingFactor": 7.0
 },
 "TimeStamp": "Fri Mar 27 11:09:26 2015"
}
```


Result Codes

Value	Description
0	Succeeded, experiment started
1	Failed due to active auto-shimming session
2	An experiment is already running
3	No response
4	Bad parameters
5	No such experiment

/interfaces/iFlow/CancelExperiment

This method cancels an active experiment.

PUT

Request

```
{}
```

Response

```
{
  "ResultCode": 0
}
```

Result Codes

Value	Description
0	Succeeded, experiment has been cancelled
1	Failed, experiment still running

/interfaces/iFlow/CalibrateSolvent

Some applications may require the calibration of the solvent.

PUT

Use this method to start a calibration.

Request

```
{ }
```

Response

```
{
  "ResultCode": 0
}
```

Result Codes

Value	Description
0	Succeeded, calibration started
1	Failed, calibration did not start

GET

Use this method to poll for the status of the calibration.

Request

```
{ }
```

Response

```
{
  “Message”: “Searching for Signal...”,
  “PercentComplete”: 20,
  “ResultCode”:1
}
```

Result Codes

Value	Description
0	Solvent calibration completed
1	Solvent calibration running

Note: This method will change in the near future to become language independent.

/interfaces/iFlow/ExperimentStatus

Use this method to poll for the status of an active/running experiment.

GET

Request

```
{ }
```

Response


```
{
   "JDX_FileContents_FD": "",
   "JDX_FileContents_TD": " (too large to display here) ",
   "JDX_Filename": "NMR_API_1H_20180105_008.jdx",
   "NumberOfScansRun": 4,
   "OriginalReceipt": {
      "ExperimentNumber": 1,
      "ResultCode": 1,
      "Settings": {
         "ActiveTimeScanInSeconds": 3.4232799857436262,
         "Apodization": 1.100000023841858,
         "DigitalResolutionInHz": 0.05261651426553726,
         "Experiment": 1,
         "NumberOfPoints": 2048,
         "NumberOfScans": 4,
         "PeakIntegrationMethod": 1,
         "PulseWidthInMicroseconds": 16.33333396911621,
         "ReceiverGain": 12,
         "ScanDelayInSeconds": 1,
         "Solvent": 0,
         "SolventGroup": 0,
         "SpectralCentreInPpm": 6,
         "SpectralWidthInPpm": 14,
         "TimePerScanInSeconds": 4.423279985743626,
         "TotalDurationInSeconds": 4.423279985743626,
         "ZeroFillingFactor": 7.0
      },
      "TimeStamp": "2017-12-20 10:26:02"
   },
   "PeakList": [
      4.986481653462923
   ],
   "PeakThresholdValue": 4.725261211395264,
   "IntegralReport": {
      "Integrals": [
         {
            "Integration": 2691.8537039676908,
            "PeakIntensity": 109.48105495762438,
            "PeakLocation": 0.10728013580023443,
            "RegionEnd": 0.6353613232149158,
            "RegionStart": -0.33872792796580065
         }
      ],
      "NumIntegrals": 1,
      "ReferenceEnergy": 7572.05
   },
   "ResultCode": 2
}
```

The entire contents of the experiment results JCAMP/DX file is included in **JDX_FileContents_TD**. The name of the saved results file on the spectrometer is found in **JDX_Filename**.

The parameter **JDX_FileContents_FD** has been deprecated.

IntegralReport is populated when integration regions have been defined (see how to setup integrals).

PeakList is populated when spectral peaks are detected above the specified threshold value (**PeakThresholdValue**). The threshold is modified with **iFlow/PeakParameters**

Result Codes

Value	Description
0	Command succeeded
1	Command failed, due to active auto-shimming
2	An experiment is already running
3	No response
4	Bad parameters
5	No such experiment

/interfaces/iFlow/PeakParameters

Use this method to adjust the threshold at which peaks are detected. When peaks are detected, they are reported in the Peaks array within ExperimentResults.

The relationship between the threshold multiplier and the value is:
PeakThresholdValue = PeakThresholdMultiplier * (detected noise level)

PUT

Request

```
{
  "PeakThresholdMultiplier": 15.0
}
```

Response

```
{
  "ResultCode": 0
}
```

Result Codes

Value	Description
0	Succeeded
1	Failed

GET

Request

```
{ }
```

Response

```
{
  "PeakThresholdMultiplier": 15.0
}
```

/interfaces/iFlow/ManualIntegrals

Use this method to setup integration regions. These regions can also be setup manually using the spectrometer itself. When integration regions are defined, experiment results include them in post-processing.

GET

Request

```
{ }
```

Response

```
{
  "Integrals": [
    {
      "RegionEnd": 2.0,
      "RegionStart": 1.0
    }
  ],
  "ReferenceEnergy": 70386.53250336811
}
```

PUT

Request

```
{
  "Integrals": [
    {
      "RegionEnd": 2.0,
      "RegionStart": 1.0
    }
  ],
  "ReferenceEnergy": 70386.5
}
```

Response

```
{
  "ResultCode": 0
}
```

Result Codes

Value	Description
0	Set integrals successful
1	Failed: Error using integral info

/interfaces/iFlow/Shim

Use this method to start an auto-shimming session (quick, medium or full) or cancel and active session.

PUT

Request

```
{
  "ShimmingMethod": 0,
  "SolventShimming": false
}
```

ShimmingMethod can be one of the following:

Value	Description
0	Set to zero to cancel an active shimming session. When reading, zero means the shim is complete or not running
1	Quick auto-shim
2	Medium auto-shim
3	Full auto-shim

SolventShimming is currently not supported and must be set to false.

Response

```
{
   "ResultCode": 0
}
```

Result Codes

Value	Description
0	Succeeded
1	Failed

GET

Poll for the status of the auto-shimming session.

Request

```
{ }
```

Response

```
{
  "PercentComplete": 100,
  "ShimmingMessage": "Done",
  "ShimmingMethod": 0,
  "SolventShimming": false
}
```

PercentComplete and **ShimmingMessage** are read-only.

/interfaces/iFlow/Settings/1D

Use this method to setup a 1D NMR experiment.

GET

Request

```
{ }
```

Response

```
{
  "AutoBaseline": false,
  "AutoGain": true,
  "AutoPhase": false,
  "CurrentGain": 12.0,
  "PulseAngle": 85.92698762441455,
  "PulseWidth": 15.0,
  "ReceiverGain": 12.0
}
```

PUT

Request

```
{
  "AutoBaseline": false,
  "AutoGain": true,
  "AutoPhase": false,
  "CurrentGain": 12.0,
  "PulseAngle": 85.92698762441455,
  "PulseWidth": 15.0,
  "ReceiverGain": 12.0
}
```

For more details on **1D Settings**, see appendix A.

To set **PulseAngle** instead of the **PulseWidth**, set **PulseAngle** to the desired value and set **PulseWidth** to -1. Conversely, to set **PulseWidth** instead of the **PulseAngle**, set **PulseWidth** to the desired value and set **PulseAngle** to -1.

Response

```
{
  "ResultCode": 0
}
```

Result Codes

Value	Description
0	Succeeded
1	Failed

Appendix A

Experiment Settings (General)

Parameter	Type	Description
ActiveTimeScanInSeconds	Double	READ-ONLY: The time during which the spectrometer is busy acquiring and processing data. This depends on the Spectral Width, Number of points, and zero filling factor.
Apodization	Double	Apodization is a mathematical function applied to the raw FID data.
DigitalResolutionInHz	Double	READ-ONLY: The digital resolution is the sweep width (in Hz) divided by the total number of digital points, the sum of number of acquisition and zero filling points. Finer digital resolution enables greater definition of fine features.
Experiment	Integer	0 = unknown, 1 = 1D, 2 = nutation, 3 = T1, 4 = T2, 5 = COSY, 6 = JRES, 7 = DEPT, 8 = HSQC, 9 = HMBC, 10 = TOCSY, 11 = Kinetics
NumberOfPoints	Integer	The number of data points acquired by the spectrometer (a binary multiple of 1024)
NumberOfScans	Integer	The number of scans to run
PeakIntegrationMethod	Integer	0 = manual, 1 = automatic
PulseWidthInMicroseconds	Double	Length of the 90 degree pulse in microseconds
ReceiverGain	Double	The receiver gain controls the attenuation of the receiver circuit.
ScanDelayInSeconds	Double	A delay to allow for sample NMR relaxation between scans
Solvent	Integer	The selected solvent. Query all the solvents and solvent groups via /interfaces/iStatus/Solvents
SolventGroup	Integer	The group of solvents (linked to the selected NMR nucleus of interest)
SpectralCentreInPpm	Double	The position, in the continuous spectrum, of the center of the desired spectral width (in ppm).
SpectralWidthInPpm	Double	The range of frequencies over which NMR signals are to be detected. It is the reciprocal of the sampling interval.
TimePerScanInSeconds	Double	READ-ONLY: The amount of time to complete a single scan, including acquisition time, delay time, and processing overhead time. One scan is the single acquisition of an NMR spectrum.
TotalDurationInSeconds	Double	READ-ONLY: The estimated time the entire experiment will take to run (all scans)
ZeroFillingFactor	Double	Increasing the zero filling enhances the digital resolution of the spectrum.

Experiment Settings (1D)

Parameter	Type	Description
AutoBaseline	Boolean	Set to true to perform an automatic baseline correction
AutoGain	Boolean	Set to true to perform an automatic gain setting
AutoPhase	Boolean	Set to true to perform an automatic phase correction
CurrentGain	Double	Read-only
PulseAngle	Double	In degrees
PulseWidth	Double	In microseconds
ReceiverGain	Double	The receiver gain controls the attenuation of the receiver circuit.