# Bruker XRFConnectProtocol Module

###### Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Editor | Description | Software Versions | Date |
| 01 | KDW | Initial release |  | 6/11/2018 |
|  |  | BrukerS1 | 2.54.54.305 Beta |  |
|  |  | Bruker Instrument Tools (BIT) | 1.7.0.113 |  |
|  |  | XRFConnectProtocol Module | 1.0.\* |  |

**Overview**

**The Bruker XRFConnectProtocol Module is a loadable library compatible with the Bruker Connectivity Server. Its primary function is to monitor a local subnet for instrument advertisement broadcasts and react to them according to a defined configuration.**

**Theory of operation:**

**The primary function of the XRFConnectProtocol module is to manage connections to instruments visible on a local network in order to facilitate automatic transfer and storage of spectral and elemental concentration data. The module accomplishes this task by maintaining a list of “known” instruments, connecting to them and processing receipt of post assay transmissions by storing the data in defined locations on disk. Configuration settings define where the received data will be stored, automatic vs. manual connection to individual instruments, how attempts are made to connect to an instrument, actions taken when various events happen and log file management.**

**Configuration Files:**

1. **XRFConnectProtocol.config:**

**This is the primary configuration file for the module, it contains entries that are common across the module itself and all instruments visible to it. The entries in this file define the server core <-> module inter-process communication mechanism, number of connection attempts, base, working instrument and data folders, common and specific action file names and log file definitions.**

1. **CommonActions.XML:**

**This file defines actions that are common to all instruments and triggered by various events.**

1. **Actions.XML:**

**This file contains actions that are associated with a specific instrument. Entries in this file add to or optionally replace or completely remove actions defined in the common actions file.**

**XRFConnectProtocol.config tag descriptions:**

**ipcmechanism : Defines the server core to module communication transport mechanism**

**internalipcendpoint : Defines the ipc end point**

**udplistenport : The UDP port listen on for instrument broadcasts**

**xrfconnecttcpport : The instrument TCP port that supports XRFConnect protocol communications**

**filetransfertcpport : TCP port used for file transfers (currently only used for post assay data transfers)**

**maxconnectionattempts : Number of attempts made by the module to connect to an instrument**

**baseworkingfolder : The root folder for the folder structure defined by the other folder tags**

**workingdatafolder : Folder where spectral and elemental concentration results file(s) are written**

**workinginstrumentfolder : Folder where instrument specific IDF and Action files will be found**

**spectraprefix : a string that will be added to spectra file names**

**resultsprefix : a string that will be added to results file names**

**commonactionfile : Name of a file containing action definitions that are common to all instruments**

**actionfile : Name of a file containing action definitions that are specific to an instrument, must be located in an instrument’s defined working folder.**

**\*\* The remaining tags in the appSettings section are not used in the beta version of the server \*\***

**\*\* DO NOT edit these settings as they may be used in future versions or removed completely \*\***

**The “log4net” section of this file contains definitions that control creation, location and layout of log files. The user should refrain from editing tags in this section with the exception of:**

**file : Defines the full path and name of the log file**

**maximumFileSize : Max size a log file is allowed to grow to before it is rolled to a backup.**

**maxSizeRollBackups : Maximum number of backups to retain.**

**\*\* The remaining tags in this section should not be changed \*\***

**Folder specifications:**

**Illegal Symbols:**

**The following symbols are not allowed in folder path specifications or file names.**

**["\*?<>:|]**

**If any of these symbols exist in a file name or folder path specification they will be replaced with an underscore “\_”.**

**Folder definition variables:**

**The definitions of the base, working data and instrument folders support use of several variables to enable flexible specification of instrument specific folders were data and configuration files will be read/written.**

**The following “variables” are currently supported:**

**%SERIALNUMBER% - Replaced with the serial number of the instrument**

**%DATE% - Replaced with the current date when the folder or file in question is created in yyyy-MM-dd format**

**%TIME% - Replaced by the time the folder or file is created in the HH-mm-ss format**

**%TIMESTAMP% - Replaced with a timestamp representing the date/time to the second when the folder or file is created. The format of this timestamp is as follows:** yyyyMMdd-HHmmss

Date and time format designators are defined as follows:

yyyy – year including the century, ex: 2018

MM – numeric calendar month, ex: 05 -> May

dd – day of the month, ex: 05 -> day 5 of whatever month

HH - hours in 24 hr format, ex 10:00 PM is represented by 22

mm – minutes

ss – seconds

\*\* All dates and times are relative to the server, not the instrument \*\*

**\*\* If the specified folder does not exist it will be created when new files are added by the server. \*\***

The following example folder definitions demonstrate how these entries work:

In these examples the instrument serial number is “900N3005”

Date is: June 11th, 2018

Time is: 2:47:45 PM.

Example 1:

baseworkingfolder = c:\bruker\

workingdatafolder = %SERIALNUMBER%\%DATE%\

workinginstrumentfolder = %SERIALNUMBER%\

**spectraprefix = “”**

**resultprefix = “”**

**Data files (PDZs, CSVs and TSVs) will be written to c:\bruker\900N3005\2018-06-11\**

**Instrument related files such as the IDF will be searched for in c:\bruker\900N3005\**

**Example 2:**

baseworkingfolder = c:\bruker\

workingdatafolder = %SERIALNUMBER%\%DATE%\

workinginstrumentfolder = %SERIALNUMBER%\

**spectraprefix = “%TIMESTAMP%\_”**

**resultprefix = “”**

**Data files (PDZs, CSVs and TSVs) will be written to c:\bruker\900N3005\2018-06-11\**

**PDZ files will have a timestamp appended to the beginning of their file name.**

**Instrument related files such as the IDF will be searched for in c:\bruker\900N3005\**

**Example 3:**

baseworkingfolder = c:\bruker\

workingdatafolder = %DATE%\

workinginstrumentfolder = %SERIALNUMBER%\

**spectraprefix = “%SERIALNUMBER%\_%TIMESTAMP%\_”**

**resultprefix = “%SERIALNUMBER%\_”**

**Data files (PDZs, CSVs and TSVs) will be written to c:\bruker\2018-06-11\**

**PDZ files will have the serial number and a timestamp appended to the beginning of their file names, results files will have the serial number appended to the beginning of the file name.**

**Instrument related files such as the IDF will be searched for in c:\bruker\900N3005\**

**The results.csv will be written to c:\bruker\2018-06-11\900N3005\_results.csv**

**The Alloys PDZ file for assay #500 will be written to**

**c:\bruker\2018-06-11\900N3005\_20180611-144745\_00500-Alloys.pdz**

**The workingdatafolder and workinginstrumentfolder values are by default defined relative to baseworkingfolder but this is not required. If the user wishes they can define one or both of these folders with a full path designation. These definitions MUST begin with a drive letter followed by a colon (:).**

**Ex: workingdatafolder = c:\project\data\%SERIALNUMBER%\**

**could be used to set the working data folder to a project folder tree while leaving the working instrument folder in its default location.**

**Action file entry descriptions:**

**The common and specific action files use the same definition syntax, the descriptions that follow apply to both types of files.**

**Action tags found in these files will be one of 3 types:**

**Add:**

**These tags add an action rule to the list of rules that will be processed when a system event is detected.**

**An Add tag must have the following attributes:**

**type : sets the type of tag, always “action” in the current system**

**action : sets the “action” to be performed when the associated trigger event is detected**

**trigger : sets the system event that prompts the module to perform the defined “action”**

**parameter : optional parameters that further define the action performed**

**\*\* this attribute must be present but can consist of an empty string**

**Remove:**

**These tags should only be found in the instrument specific actions file, their purpose is to remove common rule entries from the list of rules that are associated with a specific instrument. The tag layout is identical to the Add tag and the attribute values should exactly match those for the common Add rule that is to be removed from a specific instrument’s list.**

**Ex: A common Add rule: *<add type=”action” action=”synctime” trigger=”connect” parameter=automatic” />***

**A *<remove type=”action” action=”synctime” trigger=”connect” parameter=automatic” />* entry in the specific file removes the action from the specific instrument’s list.**

**Replace:**

**These tags should only be found in the instrument specific actions file, their purpose is to replace common rule entries with rules that are associated with a specific instrument. The tag layout is identical to the Add tag and the attribute values should exactly match those for the common Add rule with the exception of the parameter attribute.**

**Ex: A common Add rule: *<add type=”action” action=”connect” trigger=”broadcast” parameter=manual” />***

**A *<replace type=”action” action=”connect” trigger=”broadcast” parameter=automatic” />* entry in the specific file replaces the “manual” parameter value with “automatic”. The effect of this replacement is to change the reaction of the server module to receipt of a broadcast message from the specific instrument. If the replace rule did not exist then the module would wait for a control program to issue a “connect” command before attempting a connection to the instrument in question. When the replace entry is present the server module will automatically attempt to connect to the instrument upon receipt of a broadcast message from it.**

**Supported “Actions”:**

**This version of the server and associated instrument software support a limited set of actions. The example common and specific action files (appendix A and B) contain entries that demonstrate how to define automatic and manual connection, auto synchronization of instrument date time with the server and configuring enable/disable of results and spectra transmission at assay completion. All other entries in the example file(s) show potential future additions to the capabilities of the system.**

**Action : Connect**

**This action rule defines how the server reacts to a broadcast message received from an instrument**

**\*\* There should only be one active rule of this type in a rule list \*\***

**Trigger – Must contain “broadcast”**

**Parameter – “automatic” or “manual“**

**automatic – sets the server to attempt to connect as soon as a broadcast message is received**

**manual – sets the server to receive broadcasts but wait for a control program to prompt a connection attempt.**

**Action : Configure**

**This action configures various aspects of the server module and/or an instrument when the trigger event is detected.**

**Trigger – “connect” is currently the only supported trigger event for this action type**

**Parameter – the following entries are supported**

**“transfer:disable:spectra” – disables spectra (PDZ) file transmission by the instrument**

**“transfer:disable:reults” – disables results (CSV and TSV) file transmission by the instrument**

**“transfer:enable:spectra” – enables spectra (PDZ) file transmission by the instrument**

**“transfer:enable:reults” – enables results (CSV and TSV) file transmission by the instrument**

**Action : SyncTime**

**This action synchronizes the instrument date/time with the server.**

**Trigger – “connect” is currently the only supported trigger event for this action type**

**Parameter – the following entries are supported**

**“transfer:disable:spectra” – disables spectra (PDZ) file transmission by the instrument**

**“transfer:disable:reults” – disables results (CSV and TSV) file transmission by the instrument**

**“transfer:enable:spectra” – enables spectra (PDZ) file transmission by the instrument**

**“transfer:enable:reults” – enables results (CSV and TSV) file transmission by the instrument**

**Appendix A: Example CommonActions.xml file**

<?xml version="1.0" encoding="UTF-8"?>

<actionfile>

<fileversion>0.01</fileversion>

<actionlist>

<!-- \*\*\*\*\* SUPPORTED ACTIONS \*\*\*\*\* -->

<!-- set default "connect" action on receipt of instrument broadcast packets to manual -->

**<add type="action" action="connect" trigger="broadcast" parameter="manual" />**

<!-- synchronize the instrument's clock with the server's upon connection -->

**<add type="action" action="synctime" trigger="connect" parameter="automatic" />**

<!-- configure the instrument to not transmit spectra and results after each assay -->

<!-- the parameter can be formatted as follows:

parameter="transfer:disable:spectra" - Turns off transmission of PDZ files

parameter="transfer:spectra" - Turns off transmission of PDZ files

parameter="transfer:enable:spectra" - Turns on transmission of PDZ files

-->

**<add type="action" action="configure" trigger="connect" parameter="transfer:disable:spectra" />**

**<add type="action" action="configure" trigger="connect" parameter="transfer:disable:results" />**

<!-- \*\*\*\*\* SUPPORTED ACTIONS \*\*\*\*\* -->

<!-- \*\*\*\*\* POTENTIAL FUTURE CAPABILITIES \*\*\*\*\* -->

<!-- transfer specified file(s) from the instrument upon connection -->

<!--

filepull action transfers one or more files from the instrument to the server

filepush action transfers one or more files to the instrument

The filepull examples below will transfer the instrumentdef.idf file, all .uap file(s), all the contents of the titancals and system folders from the instrument to the server. The files are placed in the instrument specific working folder

The filepush example tranfers brukerS1.ini found in the system subfolder of the instrument specific workingfolder to the instrument.

It is not considered an error by the server if none of the specified files exist.

-->

**<add type="action" action="filepull" trigger="connect" parameter="\bruker\instrumentdef.idf" />**

**<add type="action" action="filepull" trigger="connect" parameter="\bruker\\*.uap" />**

**<add type="action" action="filepull" trigger="connect" parameter="\bruker\titancals\\*" />**

**<add type="action" action="filepull" trigger="connect" parameter="\bruker\system\\*" />**

**<add type="action" action="filepush" trigger="connect" parameter="%workinginstrumentfolder%\system\brukers1.ini" />**

<!-- \*\*\*\*\* POTENTIAL FUTURE CAPABILITIES \*\*\*\*\* -->

</actionlist>

</actionfile>

**Appendix B: Example Actions.xml file**

<?xml version="1.0" encoding="UTF-8"?>

<actionfile>

<fileversion>0.01</fileversion>

<instrumentserialnumber>900N3005</instrumentserialnumber>

<actionlist>

<!-- \*\*\*\*\* SUPPORTED ACTIONS \*\*\*\*\* -->

<!-- set automatic connect when broadcast packets are received -->

<!-- this overrides the default from commonactions.xml -->

**<replace type="action" action="connect" trigger="broadcast" parameter="automatic" />**

<!-- configure the instrument to transmit spectra and results after each assay -->

<!-- These entries "disable", more literally “remove” from processing, the default common entries that configure

the instrument to disable transmission of results and spectra file(s) at assay completion. -->

**<remove type="action" action="configure" trigger="connect" parameter="transfer:disable:spectra" />**

**<remove type="action" action="configure" trigger="connect" parameter="transfer:disable:results" />**

<!-- add new entries that configure the instrument to transmit results and spectra -->

**<add type="action" action="configure" trigger="connect" parameter="transfer:enable:spectra" />**

**<add type="action" action="configure" trigger="connect" parameter="transfer:enable:results" />**

<!-- \*\*\*\*\* SUPPORTED ACTIONS \*\*\*\*\* -->

<!-- \*\*\*\*\* POTENTIAL FUTURE CAPABILITIES \*\*\*\*\* -->

<!-- send new calibrations and the associated AEN file(s) to the instrument upon connection -->

**<add type="action" action="filepush" trigger="connect" parameter="%workinginstrumentfolder%\%serialnumber%\\*.aen" />**

**<add type="action" action="filepush" trigger="connect" parameter="%workinginstrumentfolder%\%serialnumber%\cals\\*" />**

<!-- The common action filepush of brukers1.ini is disabled (removed). -->

**<remove type="action" action="filepush" trigger="connect" parameter="%workinginstrumentfolder%\system\brukers1.ini" />**

<!-- \*\*\*\*\* POTENTIAL FUTURE CAPABILITIES \*\*\*\*\* -->

</actionlist>

</actionfile>