# BIT Connectivity Server Support

###### Revision History

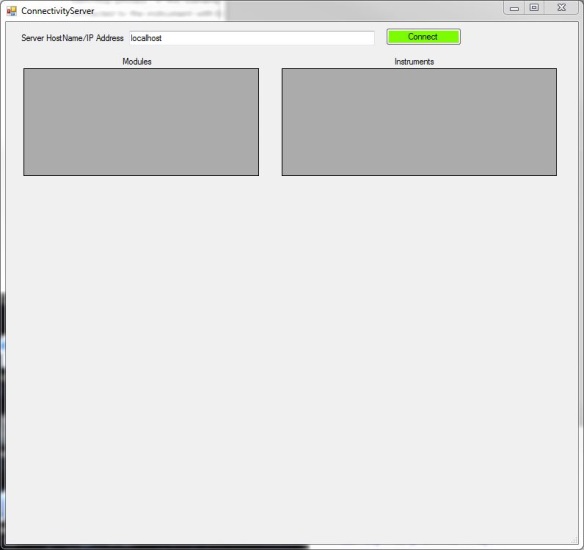
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Revision | Editor | Description | Software Versions | Date |
| 01 | KDW | Initial release |  | 6/12/2018 |
|  |  | BrukerS1 | 2.54.54.305 Beta |  |
|  |  | Bruker Instrument Tools (BIT) | 1.7.0.113 |  |
|  |  | XRFConnectProtocol Module | 1.0.\* |  |

**[*Insert after section 6. Instrument Definitions*]**

**Connectivity Server**

**The Tools -> Connectivity Server in BIT**

**provides the ability to connect to one of the Bruker Connectivity Servers. Through BIT the user can see status and configure the server. The current implementation of this form in BIT is limited to interaction with the XRFConnectProtocol server module (Handheld XRF instruements).**

**[*need figure of the Connectivity wondow*]** 

**Connect to a Connetivity Server:**

**To connect to a Connectivity Server you must either provide the hostname or IP address. If BIT and the Conenctivity Server are running on he same computer “localhost” can be used, which is the default.**

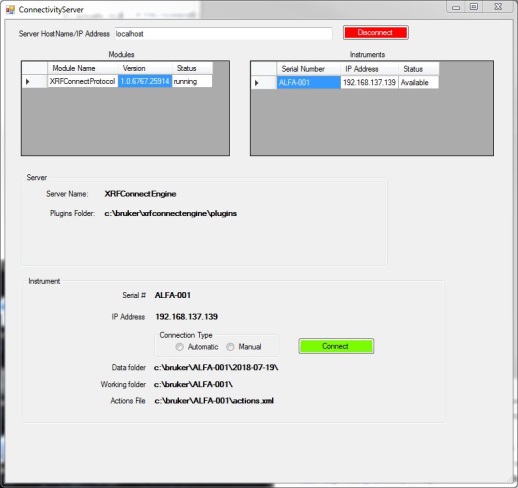
**Ex: version 4 IP address – 192.168.1.10**

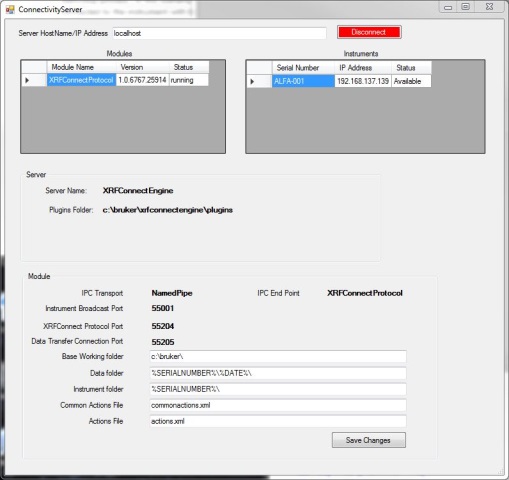
**version 6 IP address -** ::ffff:192.168.1.10

host name – KNN-HMW9C42 (these names are generally assigned by an IT department)

\*\* If a host name is entered it must be resolvable using DNS or a Windows Domain server \*\*

\*\* If this means nothing to you then use an IP address \*\*

[*show new figure with tables showing*]



After a successful connection the form will be updated to display 2 tables. The left hand table displays all the modules the Connectivity Server is managing, along with its version and status, the right hand yable displays data specific to the module highlighted in the left table. Below the tables information is displayed related to the selected server or selected instrument, depending on which table has active input control.

Each table will display several columns.

The modules grid contains the following columns:

**Module Name –** Displays the module name as it reported by the module to the server

**Version –** Shows the module’s version number, as reported by the module

**Status –** Displays the module status, running or stopped

The XRFConnectProtocol module associated grid contains the following columns:

**Serial Number –** Shows the instrument serial number

**IP Address –** Shows the instrument’s network address, if the instrument is not connected to a network 0.0.0.0 will be shown

**Status –** Shows the connected/availability status of the instrument.

**Connected**: indicates the module has established a connection to the instrument

**Available**: indicates the module has received a broadcast message but has not connected

**Not Available**: The instrument’s IDF file was processed, but no broadcast packets have been received from it ??????? How do we get the IDF if the instrument hasn’t broadcast its present?

**Module status and configuration:**

\*\* NOTE \*\*

This version of BIT does not support interactions with modules other than XRFConnectProtocol.

Due to this limitation if a module other than XRFConnectProtocol is selected, the right hand grid and the lowest section of the form will be empty.

Selecting one of the displayed modules from the list is as simple as clicking on its row in the left hand grid.

When an active module is selected the lowest section of the form will be updated to show data associated with it and the right hand grid will be updated to display the instruments it is managing.

Following is a description of the fields in the lowest section of the form associated with the XRFConnectProtocol module:

\*\* These fields are not editable using BIT

**IPC Transport –** Active inter-process communication mechanism the module uses to communicate with the server core

**IPC End Point –** Inter-process communication end point definition

**Instrument Broadcast Port –** UDP port the module listens to for broadcasts from instruments

**XRFConnect Protocol Port –** TCP port used to exchange XRFConnect protocol messages with an instrument

**Data Transfer Connection Port –** TCP port used for file transfers from the instrument

\*\* NOTE \*\*

The preceding fields can be changed using a test editor. They are found in the XRFConnectProtocol.config file described in the documentation specific to that module.

\*\* These fields are editable

**Base Working Folder –** The base working folder definition

**Working Data Folder –** The working data folder definition

**Working Instrument Folder –** The working instrument folder definition

**Common Actions File –** Specifies the actions file with entries common to all instruments

**Actions File –** Specifies the name of the action file that will read from an instrument’s working folder

**After editing the contents of these fields the new value can be posted to the server module by clicking the “Save Changes” button. The contents of the fields will be sent to the module where they will be validated for content and written to the config file. If the update is successful the module will immediately make the changes the new active settings.**

**Illegal Symbols:**

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

**["\*?<>:|]./\=,^**

**If any of these symbols exist in a file name or folder path specification they will be replaced with an underscore “\_”. I would not quetly do this, put up a warning when they type it. ISn**

**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.**

**Instrument status and configuration:**

Selecting one of the displayed instruments from the list is as simple as clicking on its row in the right hand grid.

When an active instrument is selected the lowest section of the form will be updated to show data associated with it.

Following is a description of the fields in the lowest section of the form associated with an instrument:

\*\* These fields are not editable \*\*

**Serial # - Instrument serial number**

**IP Address – Instrument’s network IP address**

**Data Folder – The instrument’s working data folder at the current time**

**Working Folder – The instrument’s working folder at the current time**

**Actions File – The name of the instrument specific actions file**

**The “Connection Type” sets the action response to a broadcast message from an instrument.**

**“Automatic” indicates the server module automatically attempt to connect to the instrument when a broadcast message is received from the instrument.**

**“Manual” indicates the server will wait for a command from a control program instructing it to connect to the instrument.**

**The “Connect/Disconnect” toggles the state of the connection from the module to the instrument.**

**If this button has a green background and contains “Connect”, clicking it sends a command to the server module instructing it to attempt to establish a connection to the selected instrument.**

**If the button has a red background and contains “Disconnect”, clicking it sends a command instructing the server module to disconnect from the instrument.**

**Performing either of these actions results in a change in status of the connection to an instrument. This status change will be reflected in the instrument grid status column and the state of the connect/disconnect button.**

**Manually initiating a connection will trigger all “connect” actions associated with the instrument.**

**Changing the “Connection Type” by selecting the non-active radio control immediately updates the associated actions list at the server and writes the change to the associated instrument specific action file. If an instrument specific action file doesn’t exist it will be created.**

**\*\*\* NOTE \*\*\***

**The instrument side of the USB cable network connection does not tolerate removal of the cable while network connections are active. Due to this limitation of the device it is STRONGLY recommended that network connections to the instrument be disconnected before the cable is removed. Disconnection from the server is accomplished using the Connect/Disconnect button in the Instrument section of this form.**

**Removal of the USB cable while network connections are active should not result in corrupt data or instrument malfunctions, but the instrument will most likely require a power cycle before connection to the server can be re-established.**