

PippinPrep v.4.5 Remote Control

Capabilities and API

Rev.1.5
April 26, 2011

Table of Contents

Purpose.....	3
PippinPrep Network Settings.....	4
Using VNC Client.....	5
Using FTP Client.....	6
Pippin API: General Description.....	7
Pippin API: Configuration of UDP Server.....	8
Pippin API: Command Format.....	9
Pippin API: Command Description.....	10
1 ListCassettes – List defined cassettes	11
2 ListProtocols – List defined protocols	12
3 LoadProtocol – Load a protocol.....	13
4 GetLoadedProtocol – Get the name of the loaded protocol.....	14
5 GetProtocolProperties – Get properties of any protocol.....	15
6 ChangeProtocolState – Change state of a loaded protocol	16
7 GetProtocolStatus – Get status of a loaded protocol.....	17
8 ListLogs – List logs contained in a YYYY-MM folder	18
9 GetLogContents – Get log contents as a Base64 string.....	19
10 GetSystemInfo – Get firmware and software versions, build date	20
Pippin API: Error Codes	21

Purpose

PippinPrep supports several methods of remote control and monitoring. They are:

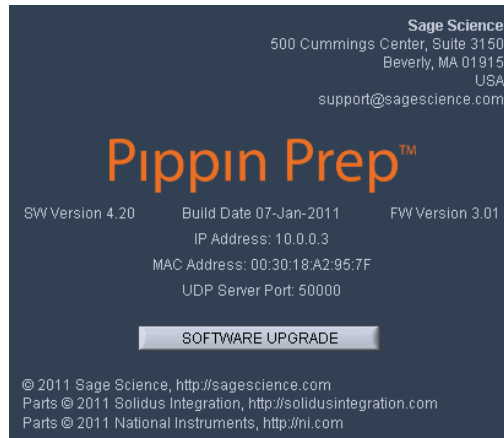
- Remote access using VNC client software;
- Remote retrieval of log and image files using FTP client software;
- Programmatic remote control using PippinPrep API.

All of these methods are implemented over Ethernet.

PippinPrep Network Settings

By default, PippinPrep is configured for IPv4 networks to use DHCP for obtaining its IP address. Once PippinPrep starts, the user can review instrument's network configuration by clicking on the INFO button on the user interface. Below is the screen shot of the pop-up Info window showing the following relevant information:

- Instrument IP address;
- Ethernet adapter MAC address;
- UDP server port address (used for programmatic remote control).



Additionally, the following information may be useful:

- VNC server port: 5900;
- FTP server port: 21.

If DHCP is not acceptable for the customer's network, network settings can be modified using the following procedure:

1. Press Ctrl-Alt-RightCursorArrow to move to an empty desktop;
2. Right-click anywhere on the desktop and select "Open In Terminal";
3. At the command prompt, type "gnome-panel" (without quote marks);
4. Move the cursor to the bottom of the screen, click on Computer and select "Control Center";
5. In Control Center, select "Network Connections", go to "Wired" tab, select "Eth1" and "Edit";
6. Go to "IPv4 Settings" and configure desired network settings;
7. Close all open windows except the terminal window: click on "Close", "Accept" and so on;
8. In the terminal window, enter Ctrl-C, this will close the Gnome panel;
9. Close the terminal window;
10. Press Ctrl-Alt-LeftCursorKey to return to the PippinPrep user interface.

Using VNC Client

PippinPrep runs a VNC server, which enables users to access it from any platform (Windows, Linux, iPad, etc.) using any VNC client.

1. Install a VNC client on your computer;
2. Connect your computer to the network that hosts PippinPrep;
3. Start VNC client software, enter the IP address of PippinPrep as shown in its INFO pop-up window, if necessary enter the port value of 5900 (default for VNC);
4. When prompted for a password, enter “pipr” (without quote marks).

Using FTP Client

PippinPrep runs an FTP server, which enables users to access it from any platform (Windows, Linux, iPad, etc.) using any FTP client.

1. Install an FTP client on your computer;
2. Connect your computer to the network that hosts PippinPrep;
3. Start FTP client software, enter the IP address of PippinPrep as shown in its INFO pop-up window, if necessary enter the port value of 21 (default for FTP);
4. Log in as an anonymous user – no password required.
5. You will be given access to the Logs folder and all its subfolders.

Pippin API: General Description

PippinPrep software runs a UDP server. The instrument listens for incoming UDP datagrams, parses them, takes action based on their contents, and responds by sending a UDP datagram to the client's address. In principle, it is possible to control the instrument from the user interface *and* more than one remote IP address at the same time, so care should be taken to avoid conflicting commands (although it is acceptable to use the user interface or one IP address to control the instrument and multiple other addresses to monitor its status).

Pippin API: Configuration of UDP Server

By default, this server listens on port 50000. If necessary, this value can be modified using the following procedure:

1. Press Ctrl-Alt-RightCursorArrow to move to an empty desktop;
2. Right-click anywhere on the desktop and select “Open In Terminal”, leave the terminal window open;
3. At the command prompt, type “nautilus” (without quote marks);
4. When the file manager opens, browse to the folder /home/pippin/PippinPrep/Bin;
5. Double-click on the file PippinPrep.conf and select “Display”;
6. Find the line `UDPport=50000`, change the value to any other value in the range 50000-51000 (inclusive), save changes and close the file;
7. Close the file manager and the terminal window;
8. Press Ctrl-Alt-LeftCursorKey to return to the PippinPrep user interface;
9. Shut down and restart the instrument for the changes to take effect.

Be careful not to make any other changes to the PippinPrep.conf file or to the contents of the Bin folder!

Pippin API: Command Format

All client requests and PippinPrep responses are formatted as XML strings. All requests and responses have the format as shown below for the “ListLogs” request.

Sample request:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<PippinRequest>                                <!--tag always present-->
  <ListLogs>                                    <!--request tag-->
    <Year>2011</Year>                            <!--request-specific tag, value-->
    <Month>03</Month>                            <!--request-specific tag, value-->
  </ListLogs>
</PippinRequest>
```

Sample response:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<PippinResponse>                                <!--tag always present-->
  <ErrorCode>0</ErrorCode>                      <!--tag always present-->
  <ErrorMessage>No Error</ErrorMessage>        <!--tag always present-->
  <ListLogs>                                     <!--request tag always present-->
    <TotalLogs>3</TotalLogs>                   <!--request-specific-->
    <LogName1>PippinLogA</LogName1>           <!--request-specific-->
    <LogName2>PippinLogB</LogName2>           <!--request-specific-->
    <LogName3>PippinLogC</LogName3>           <!--request-specific-->
  </ListLogs>
</PippinResponse>
```

Notes:

- Tags describing individual items in a list use one-based indexing. For example, in response to the “ListLogs” request, the tag for the first returned log is “Log1”, not “Log0”.
- All XML commands and parameters are case-sensitive.

Pippin API: Command Description

This page intentionally left blank. Command description starts on the next page.

1 *ListCassettes – List defined cassettes*

PippinRequest:

Request tag: **ListCassettes**

Request-specific tags / examples:

--	--

PippinResponse:

Request-specific tags / examples:

TotalCassettes	2
CassetteName1	1.5% Marker A
CassetteName2	2.0% Marker B

2 *ListProtocols – List defined protocols*

PippinRequest:

Request tag: **ListProtocols**

Request-specific tags / examples:

--	--

PippinResponse:

Request-specific tags / examples:

TotalProtocols	2
ProtocolName1	Kyoto Protocol
ProtocolName2	Protocol of the Sage Science

3 *LoadProtocol – Load a protocol*

PippinRequest:

Request tag: `LoadProtocol`

Request-specific tags / examples:

ProtocolName	Kyoto Protocol
--------------	----------------

PippinResponse:

Request-specific tags / examples:

--	--

Note: “Manual Mode” (without quote marks) is a valid protocol name.

4 ***GetLoadedProtocol – Get the name of the loaded protocol***

PippinRequest:

Request tag: `GetLoadedProtocol`

Request-specific tags / examples:

--	--

PippinResponse:

Request-specific tags / examples:

ProtocolName	Kyoto Protocol
--------------	----------------

5 *GetProtocolProperties* – Get properties of any protocol

PippinRequest:

Request tag: **GetProtocolProperties**

Request-specific tags / examples:

ProtocolName	Kyoto Protocol
--------------	----------------

PippinResponse:

Request-specific tags / examples:

CassetteName	1.5% Marker A
RunTimeSec	7200
TotalLanes	5
Lane1	
LaneMode	Tight (Range,Time,Peak,Ref,Off)
BPTarget	300
BPStart	280
BPEnd	320
BPPause	0
TStartSec	0
TEndSec	0
TPauseSec	0
BPThresh	0
Comment	Tight Distribution Test
SigMonitor	true
Lane2	
LaneMode	Ref
BPTarget	0
BPStart	0
BPEnd	0
BPPause	0
TStartSec	0
TEndSec	0
TPauseSec	0
BPThresh	0
Comment	Reference Lane
SigMonitor	true
...	

Note: the protocol does not have to be loaded.

6 *ChangeProtocolState – Change state of a loaded protocol*

PippinRequest:

Request tag: `ChangeProtocolState`

Request-specific tags / examples:

ProtocolAction	Start (Pause, Resume, Abort)
----------------	------------------------------

PippinResponse:

Request-specific tags / examples:

--	--

7 *GetProtocolStatus* – Get status of a loaded protocol

PippinRequest:

Request tag: **GetProtocolStatus**

Request-specific tags / examples:

--	--

PippinResponse:

Request-specific tags / examples:

ProtocolName	Kyoto Protocol
ProtocolState	Running (Paused, Idle)
ElapsedTimeSec	3650
BPAAtLED	450
BPAAtBranch	390
Lane1	
LaneState	Idle (Separate, Elute)
LaneCurrentMa	2.34
PhotoCurrentMa	1.23
ElutionTimerSec	30
Lane2	
LaneState	Idle (Separate, Elute)
LaneCurrentMa	2.34
PhotoCurrentMa	1.23
ElutionTimerSec	30
...	

8 *ListLogs – List logs contained in a YYYY-MM folder*

PippinRequest:

Request tag: **ListLogs**

Request-specific tags / examples:

LogYear	2011
LogMonth	1

PippinResponse:

Request-specific tags / examples:

TotalLogs	2
LogName1	2011-01-17_14-50-20_PLog
LogName2	2011-01-19_09-30-15_PLog

9 ***GetLogContents – Get log contents as a Base64 string***

PippinRequest:

Request tag: **GetLog**

Request-specific tags / examples:

Year	2011
Month	1
LogName	2011-01-17_14-50-20_PLog

PippinResponse:

Request-specific tags / examples (note that the body of the returned log is base64-encoded):

LogContents	(base64 listing)
-------------	------------------

10 *GetSystemInfo* – Get firmware and software versions, build date

PippinRequest:

Request tag: **GetSystemInfo**

Request-specific tags / examples:

--	--

PippinResponse:

Request-specific tags / examples:

InstrumentName	MyFirstPippin
InstrumentID	X0356
FirmwareVersion	3.11
SoftwareVersion	4.3
BuildDate	22-Mar-2011
FieldReversalEnabled	TRUE
FieldReversalSeconds	5

Pippin API: Error Codes

Error Code	Error Message
0	No error
5001	Unexpected root element
5002	Undefined request tag
5003	Unexpected parameter tag
5004	Undefined parameter value