



PRACTICAL INNOVATION



DataPaq™ Single Remote Control

User Guide





PRACTICAL INNOVATION



DataPaq™ Single Remote Control User Guide

All Rights Reserved. **Ziath™** and **DataPaq™** are registered trademarks of Ziath Ltd. No part of this publication, in either its printed or electronic format, may be copied, reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose involving resale for profit or gain, through any form of paid membership or subscription service, without the express permission of Ziath Ltd.

DataPaq™ User Guide © Copyright 2008, 2009, 2010, 2011, 2012, 2013, 2014 Ziath Ltd.

Revised 07/02/2014



DataPaq Single Remote Control

Introduction

The headless enhancement to **DataPaq Single** allows for the interaction of a separate computer program to control the scanner and retrieve the results. This is enabled by a separate program entitled *DataPaqSingleServer.exe* which is provided. This will interact with **DataPaq** and control the scanner without a Graphical User Interface showing on the screen.

There are two modes in which the server can be started up: (i) command line mode, and (ii) server mode.

Command Line Mode

Command line mode allows the user to simply execute a command line at the command prompt: the software will start up and wait a certain period for scan – the resulting scan will be returned on stdout; if no tube is scanned in that time NO_CODE will be returned. To start the server in command line mode you simply need to specify the following options:

Short Form	Long Form	Params	Default	Description
t	timeout	Milliseconds with no unit eg 1000	5000	The time the scanner will wait for a read before returning 'NO_CODE'
v	verbose	none	None	If specified DataPaq Single will post comments on progress to the screen.

Therefore, due to the defaults, the simplest command line operation of **DataPaq Single** is as follows:

```
DataPaqSingleServer
```

This will run the scanner to scan in command line mode, returning the scanned barcode in text format to the command line.

Server Mode

The server mode starts **DataPaq** and listens on a specified port for incoming commands. Multiple clients can attach to **DataPaq Single**; however, only one operation at a time can be performed. If starting in server mode, you need to specify the command line option *s* (server for longform).

DataPaq will start and listen to a specified port for commands (by default the server will listen on port 9999) – however, this can be changed with the following command:

Short Form	Long Form	Params	Default	Description
p	port	port number	9999	The port that the server will listen on.



Once the server is running in socket mode, the process will continue to listen to the socket until it is either closed or receives the command SHUTDOWN. An example of a command line to start the server is below:

```
Server -s -p 8899
```

This will start the server running in socket mode which will listen for incoming connections on port 8899. The server will send each barcode as it is scanned to the client followed by a carriage return and line feed. This will happen spontaneously and does not require the user to issue a command. In addition the server responds to a set of commands: all commands must be terminated by a carriage return and a line feed; all responses will be terminated with a carriage return and a line feed.

Commands

- **Version**

Command: VERSION\r\n

Response: <version code>\r\nOK\r\n

When the server receives the command, it returns the version number of **DataPaq Single** that it is connected to.

Example

Received by Server: VERSION\r\n

Response from Server: 1.1\r\nOK\r\n

- **Scanner Status**

Command: SCANNER_STATUS\r\n

Response: <scanner_status>\r\nOK\r\n

When the server receives the command, it returns the status of **DataPaq Single**.

These can be as follows:

- SEEKING_CAMERA– **DataPaq Single** is looking for the camera to connect to. This is displayed either while the camera is being initialized or if the camera is not plugged in.
- RUNNING – **DataPaq Single** is connected to the camera and looking for a code



Example

Received by Server: `SCANNER_STATUS\r\n`

Response from Server: `RUNNING\r\nOK\r\n`

- **Status**

Command: `STATUS\r\n`

Response: `<status>\r\nOK\r\n`

When the server receives the command, it returns the status of **DataPaq Single**.

These can be as follows:

- IDLE – **DataPaq Single** is ready to accept a command
- BUSY – **DataPaq Single** is currently executing a command
- ERROR – **DataPaq Single** encountered an error; it will keep this status until a command is successfully executed

Example

Received by Server: `STATUS\r\n`

Response from Server: `IDLE\r\nOK\r\n`

- **Close**

Command: `CLOSE\r\n`

Response: `OK\r\n`

Upon receiving this command, the scanner will close the client which requested the close.

To cleanly disconnect your client, it is recommended to call this method. Note that this will not affect any other clients attached to **DataPaq Single** at the time.

Example

Received by Server: `CLOSE\r\n`

Response from Server: `OK\r\n` [The server disconnects the client]



- **Shutdown**

Command: SHUTDOWN\r\n

Response: OK\r\n

Upon receiving this command, the scanner will shutdown and the server process will exit, but only if the scanner is idle. Should the scanner fail to shut down, ERR\r\n will be written, followed by a description of the error message; however should this happen the Server will *still* shutdown (except in the case of the server being busy).

Example

Received by Server: SHUTDOWN\r\n

Response from Server: OK\r\n [The scanner shuts down] OK\r\n

- **Force Shutdown**

Command: FORCE_SHUTDOWN\r\n

Response: OK\r\n

Upon receiving this command, the scanner will shutdown and the server process will exit, even if the scanner is running – use this command with caution. Should the scanner fail to shut down, ERR\r\n will be written, followed by a description of the error message; however should this happen the Server will *still* shutdown.

Example

Received by Server: SHUTDOWN\r\n

Response from Server: OK\r\n [The scanner shuts down] OK\r\n



Error Codes

Should the server encounter an error, it will return the error plus a description in the following format:

```
<ERROR_CODE>\r\n
```

```
<ERROR_DESCRIPTION>r\n
```

In some cases there will not be a specific error message but the description will always be present and two lines will always be returned. An example of an error return is below:

```
ERR3\r\n
```

```
Unknown Command\r\n
```

In the case of an error with a description an example is below:

Error Code	Error Description
ERR1	Failed to shutdown
ERR2	Camera Busy
ERR3	Unknown Command