KC Wirefree

kcSerial Reference Guide

5 June 2006

TABLE OF CONTENTS

1 Preface

The document describes an embedded application that provides a kcSerial cable replacement service using the Bluetooth Serial Port Profile. This software was originally developed by Zeevo, Inc. under the name Zerial.

1.1 Purpose

This document provides a detailed description of each command supported by the kcSerial interface. Each description explains parameters and the expected behaviors of each command and response.

Errors responses are also detailed in this document.

1.2 Definitions and Acronyms

The following acronyms are used in this document.

Table 1. Definitions and Acronyms

Term	Description/Meaning
ASCII	American Standard Code for Information Interchange, a standard describing encoding of characters; the use in this document is strictly US 7-bit
BD	Bluetooth Device
DCD	Modem signal "data carrier detect"; indication from a modem that a connection has been made through, for example, a dialup connection
DTE	Data terminal entity, e.g., a computer
DTR	Modem signal "data terminal ready"; indication to a modem that the data terminal is ready for a connection
DUN	Dialup Networking (Profile)
GPIO	General Purpose Input-Output
LAN	Local Area Network
PIN	Personal Identification Number
SPP	Serial Port Profile
UART	Universal Asynchronous Receiver-Transmitter

1.3 Feedback

We are constantly improving our product and would very much like to get your feedback. Please send your feedback in an email to support@kcwirefree.com.

For the latest updates and additional information please visit the KC Wirefree website at: www.kcwirefree.com

2   Overview

This chapter gives a basic overview of the kcSerial interface. For further information, please refer to the *kcSerial User Guide*.

2.1   kcSerial Interface Overview

kcSerial is a cable replacement application that provides point-to-point communication between two Bluetooth devices. A serial port is used to communicate with a host device through an AT command interface as shown below.
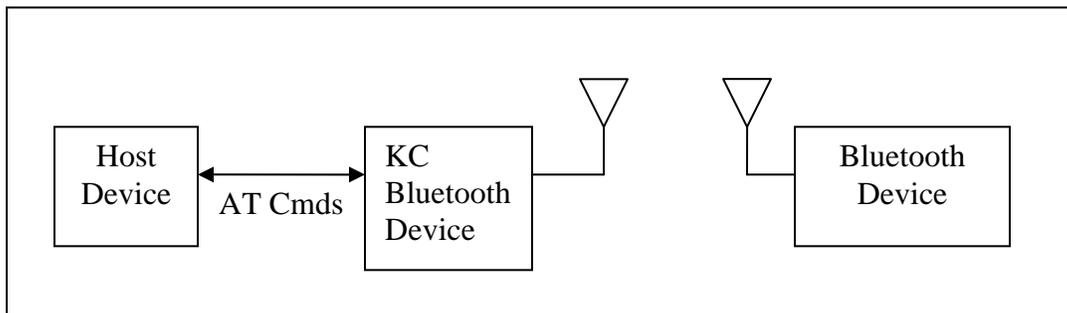
kcSerial provides the following basic features:

- Point-to-point connection – kcSerial only supports a connection with one device at a time.
- Serial Port Profile – SPP is supported with kcSerial for both Client and Server application.
- Dial Up Networking – DUN profile support for Client applications (DUN Server is not currently supported).
- Command and Bypass modes – it is possible to switch between Command and Bypass (data transmit/receive) modes during an active connection
- Security – Bonding and data encryption provides a secure link between two devices.
- Multiple Device Bonding – special security keys can be exchanged with multiple devices to allow different devices to securely connect with kcSerial (although not simultaneously)
- Power conservation – use of Park, Sniff, and Hold features to minimize power consumption
- Variable Baud Rates – the serial port can be configured for the following baud rates: 9600, 19.2K, 38.4K, 57.6K, 115.2K (default), 230.4K, 460.8K, 921.6K

3   Commands

This chapter details the each of the kcSerial AT commands including brief descriptions of behavior, syntax of the command, context of the command, and types of responses.

This *kcSerial Reference Guide* covers the following commands:

Table Key:

✓ – command is supported in this release

U – command has been updated for this release, see release notes

X – command not supported in this release

Table 3: kcSerial Command Summary

Command	kcSerial v1.0	kcSerial v2.0	kcSerial v2.1/kcSerial v2.2
Bond	✓	✓	✓
Bypass	✓	✓	✓
ChangeBaud	✓	✓	✓
ChangeDefaultBaud	✓	✓	✓
DefaultLocalName	X	✓	✓
DeleteSmartCable	✓	✓	✓
DisableBond	✓	✓	✓
Discovery	✓	✓	✓
DunConnect	✓	✓	✓
DunDisconnect	✓	✓	✓
EnableBond	✓	✓	✓
EraseBondTable	✓	✓	✓
ExitPart	✓	✓	✓
ExitSniff	✓	✓	✓
GPIOConfig	✓	✓	✓
GPIORead	✓	✓	✓
GPIOWrite	✓	✓	✓
Hold	✓	✓	✓
HostEvent	✓	✓	✓
LocalName	✓	✓	✓

Command	kcSerial v1.0	kcSerial v2.0	kcSerial v2.1/kcSerial v2.2
Park	✔	✔	✔
RemoteCommand	**X**	**X**	✔
RemoteCmdDisconnect	✔	✔	✔
Reset	✔	✔	✔
Security	✔	✔	✔
SmartCableSetup	✔	✔	✔
Sniff	✔	✔	✔
SPPConnect	✔	✔	✔
SPPDisconnect	✔	✔	✔
StreamingSerial	**X**	**X**	✔
UpdateInquiryScan	✔	✔	✔
UpdatePageScan	✔	✔	✔
Version	✔	**U**	**U**

The following subsections describe each of these commands in detail, including a description of behavior, syntax (including possible parameter values), and types of responses.

Some responses will not be "immediate". Where applicable, these will be noted and will include an approximate delay before response.

For commands with optional parameters, all possible forms will be listed under the syntax subsection.

Error responses are described in Section 4 Error Responses.

3.1 Bond

The `Bond` command is used to initiate bonding with a specified device. A personal identification number (PIN) is also required with this command.

3.1.1   Syntax

```
AT+ZV Bond [BD addr] [PIN]
```

Where `[BD addr]` is the BD Address of the remote device with which to bond and `[PIN]` is the PIN code to use (up to 16 characters).

3.1.2   Responses

If the request is successfully submitted, the response is:

```
AT-ZV BondPending [BD addr]
```

If the operation is successful, the response is:

```
AT-ZV BondOk
```

If the operation fails, the response is:

```
AT-ZV BondFail
```

3.2 Bypass

The `Bypass` command is used to return the kcSerial interface to the bypass mode, if a connection is still available. The possible use for this is to change a setting after a connection has been made (such as the UART baud rate). If the kcSerial interface does not have a connection, it will respond as if the connection is down.

3.2.1   Syntax

```
AT+ZV Bypass
```

3.2.2   Responses

If a connection is still available, the response is:

```
AT-ZV -BypassMode-
```

If there is currently no connection, the response is:

```
AT-ZV ConnectionDown
```

3.3   ChangeBaud

The host sends the `ChangeBaud` command in order to change the local UART speed to a new speed identified by the host. This setting will only remain in effect during the current session - until reset.

3.3.1   Syntax

```
AT+ZV ChangeBaud [rate]
```

where `[rate]` is the new baud rate:

- 9600
- 19,200
- 38,400
- 57,600
- 115,200
- 230,400
- 460,800
- 921,600

3.3.2   Responses

If the change is accepted, the response is:

```
AT-ZV Baudrate Changed
```

The actual change will not occur until the response has been completely transmitted.

3.4 ChangeDefaultBaud

The host sends the `ChangeDefaultBaud` command in order to change the default UART speed to a new speed identified by the host. This command is used to override the default baud rate from the Dynamic Configuration script so that the device does not require reprogramming to update this setting. The new baud rate is updated permanently until the device is either re-programmed or another `ChangeDefaultBaud` command is issued. The baud rate specified in the command will not take effect until the device is reset. To change the baud rate of the current session, use the `ChangeBaud` command.

3.4.1   Syntax

```
AT+ZV ChangeDefaultBaud [rate]
```

where `[rate]` is the new baud rate:

- 9,600
- 19,200
- 38,400
- 57,600
- 115,200
- 230,400
- 460,800
- 921,600

3.4.2   Responses

If the change is accepted, the response is:

```
AT-ZV Baudrate Changed
```

3.5 DefaultLocalName

The `DefaultLocalName` command is used to set the name of the device to the name that is reported during device discoveries. By default, the kcSerial interface uses "KCWirefreeDevice". Changing the name using this command will permanently change the local name, unlike the `LocalName` command.

3.5.1   Syntax

```
AT+ZV DefaultLocalName [name]
```

Where `[name]` is a string for the new local name (up to 50 characters). The space character is allowed; the name is assumed to be all text up to the end of the command.

3.5.2   Responses

If the operation is successful, the response is:

```
AT-ZV LocalNameOk
```

3.6   DeleteSmartCable

The **DeleteSmartCable** command removes the current Smart Cable settings that were entered using the SmartCableSetup command, but not the setting from the dynamic configuration. The Smart Cable will then be deactivated for the remainder of this session. Upon reset, if a dynamic configuration for a Smart Cable exists, it will be activated. If there is no dynamic configuration Smart Cable setup, then this feature will remain deactivated.

3.6.1   Syntax

```
AT+ZV DeleteSmartCable
```

3.6.2   Responses

If the operation is successful, the response is:

```
AT-ZV DeleteSmartCableDone
```

3.7   DisableBond

The **DisableBond** command is used to disallow new bonding with a device.

This command cannot be used while a connection is active.

3.7.1   Syntax

```
AT+ZV DisableBond
```

3.7.2   Responses

If the operation is successful, the response is:

```
AT-ZV BondDisabled
```

3.8   Discovery

The **Discovery** command is used to initiate a device discovery. The command will return the number of responses of nearby devices and then the individual responses with BD address, name of device, and service names - optional. The number of devices returned is limited to 10; and the number of services per device is limited to 8.

Inquiry is performed with an interval of 10.24 seconds.

The devices are reported in the same order as the original inquiry results.

3.8.1   Syntax

```
AT+ZV Discovery

AT+ZV Discovery [CoD]

AT+ZV Discovery [CoD] [profile] [include service
enable/disable]
```

where devices are filtered on the `[CoD]` class:

- `All` (default)
- `Misc`
- `Computer`
- `Phone`
- `LAN`
- `Peripheral`
- `Imaging`
- `Unclass`

where the service name is requested for the profile named `[profile]`:

- `All` (default)
- `SPP`
- `DUN`
- `LAP`
- `FAX`

where `[include service enable/disable]: true` returns both the remote services and names, and `false` skips the remote service discovery and only returns the remote names – this completes the discovery process faster.

3.8.2   Responses

When the discovery command has been accepted, the response is:

```
AT-ZV InqPending
```

Once the initial inquiry is complete and discovery has been started, the response is:

```
AT-ZV DiscoveryPending [num]
```

where `[num]` is the number of devices found, in decimal (up to 10 will be reported).

For each name or service name request that is successful, the response uses the returned names in the following format.

```
AT-ZV Device [BD addr] [name] [service name] [...]
```

where `[BD addr]` is in hexadecimal with the most significant byte first. `[name]` is a string in double quotes " ". `[service name]` is a string without quotes and is reported for each service reported.

For each unsuccessful name request, the corresponding name is replaced by `"Unknown"`. The name request may not be successful if unable to make a connection for the request.

```
AT-ZV Device [BD addr] "Unknown" [service name] [...]
```

For each service request that does not return services, one name is returned as "`NoSvcs`". A request may not return services if unable to connect to the device for the request or if the device does not contain the requested service.

```
AT-ZV Device [BD addr] [name] NoSvcs
```

For each unsuccessful service request, one name is returned as "`SvFail`". This will only occur due to an internal error.

```
AT-ZV Device [BD addr] [name] SvFail
```

If the memory allocated for the service discovery buffers is insufficient, then then result may return an "`Inval`" response. If this occurs, please contact support@kcwirefree.com for assistance.

```
AT-ZV Device [BD addr] [name] [service name 1] [Inval]
[Inval]
```

3.9 DUNConnect

The **DUNConnect** command is used to initiate a connection with the specified device. The remote BD address must be specified. The remote Service is optional. If not specified, the first registered DUN service will be used by default.

3.9.1 Syntax

```
AT+ZV DUNConnect [BD Addr] [Service]
```

Where `[BD Addr]` is the remote devices BD Address to page. `[Service]` is the specific service on the remote device; optional.

3.9.2 Responses

If the connection is successful, the response is:

```
AT-ZV ConnectionUp
AT-ZV -BypassMode-
```

If the connection cannot be completed, the response is:

```
AT-ZV DUNConnectionClosed
```

3.10 DUNDisconnect

The `DUNDisconnect` command is used to terminate a connection with the remote device.

3.10.1   Syntax

```
AT+ZV DUNDisconnect
```

3.10.2   Responses

If the connection is successful, the response is:

```
AT-ZV DUNConnectionClosed
```

3.11 EnableBond

The `EnableBond` command is used to enable bonding with another device. The BD Address, PIN and timeout parameters are optional.

When no BD Address is specified, requests from all BD Addresses are allowed.

If a BD Address is specified, bonding requests from devices with BD Addresses other than the one specified will fail and the existing link key will be deleted for that device.

Optionally, a PIN code may be entered with this command. If no PIN code is specified, the default PIN code will be used. The default PIN code is either the last 4 digits of the device's BD address or the dynamically configured PIN code, depending on the default PIN selection in the dynamic configuration file.

Also, a timeout value, in seconds, may be entered after the PIN code. Bonding will be disabled automatically after the requested timeout. If no timeout is specified, bonding is enabled until reset or until the `DisableBond` command is used.

If this command is issued multiple times, only the last PIN and BD address are saved. Also, if this command is issued before the first timeout occurs, the subsequent command will extend the timeout. The timeout is always set to the specified time beyond the last received `EnableBond`.

3.11.1   Syntax

```
AT+ZV EnableBond
AT+ZV EnableBond [BD addr]
AT+ZV EnableBond [BD addr] [PIN]
AT+ZV EnableBond [BD addr] [PIN] [timeout]
```

Where `[BD addr]` is the BD Address of the remote device with which to bond, `[PIN]` is the PIN code to use (up to 16 characters), and `[timeout]` is the duration of the timeout in seconds (1 to 14,400, in decimal).

3.11.2   Responses

If the operation is successful, the response is:

```
AT-ZV BondEnabled
```

If bonding has been initiated by a remote device, the notification is:

```
AT-ZV BondPending [BD addr]
```

where `[BD addr]` is the BD address of the remote device that initiated the bonding.

If bonding has occurred, the notification is:

```
AT-ZV BondOk [BD addr]
```

where `[BD addr]` is the BD address of the remote device with successful bonding.

If bonding was initiated by a remote device but failed, the notification is

```
AT-ZV BondFail
```

When the time limit for bonding has expired, the notification is

```
AT-ZV BondDisabled
```

3.12 EraseBondTable

The **EraseBondTable** command is used to erase all of the bonded device entries. Single devices cannot be erased with this command

3.12.1   Syntax

```
AT+ZV EraseBondTable
```

3.12.2   Responses

If the operation is successful, the response is:

```
AT-ZV BondTableErased
```

3.13 ExitPark

The **ExitPark** command is used to switch a device from park mode to active mode.

3.13.1   Syntax

```
AT+ZV ExitPark [BD address]
```

Where `[BD address]` is the BD address of the device to be switched to active mode.

3.13.2   Responses

If the operation is successful, the response is:

```
AT-ZV ActiveMode
```

3.14 ExitSniff

The **ExitSniff** command is used to switch a device from sniff mode to active mode.

3.14.1   Syntax

```
AT+ZV ExitSniff [BD address]
```

Where `[BD address]` is the BD address of the device to be switched to active mode.

3.14.2   Responses

If the operation is successful, the response is:

```
AT-ZV ActiveMode
```

3.15 GPIOConfig

The **GPIOConfig** command is used to configure a GPIO pin to input or output.

3.15.1 Syntax

```
AT+ZV GPIOConfig [GPIO Pin] [Configuration]
```

Where [GPIO Pin] is the Pin number, 0 – 15, of the desired GPIO to configure. [Configuration] is "i" or "I" for input and "o" or "O" for output.

3.15.2 Responses

If the operation is successful, the response is:

```
AT-ZV GPIOConfigDone
```

3.16 GPIORead

The **GPIORead** command is used to read a GPIO pin. A GPIO may be read while configured as either an input or output.

3.16.1 Syntax

```
AT+ZV GPIORead [GPIO Pin]
```

Where [GPIO Pin] is the Pin number, 0 – 15, of the desired GPIO to read.

3.16.2 Responses

If the operation is successful, the response is:

```
AT-ZV GPIOReadDone [result]
```

Where [result] is either a 1 to indicate high, or 0 to indicate low.

3.17 GPIOWrite

The **GPIOWrite** command is used to set a GPIO pin to high or low. A GPIO may only be set when configured as an output.

3.17.1 Syntax

```
AT+ZV GPIOWrite [GPIO Pin] [Setting]
```

Where [GPIO Pin] is the Pin number, 0 – 15, of the desired GPIO to read. [Setting] is a 1 to set a pin to high and a 0 to set a pin to low.

3.17.2 Responses

If the operation is successful, the response is:

```
AT-ZV GPIOWriteDone
```

3.18 Hold

The `Hold` command is used to switch a device from active mode to hold mode.

3.18.1 Syntax

```
AT+ZV Hold [BD address] [Hold Duration]
```

Where `[BD address]` is the BD address of the device to be switched to active mode. `[Hold Duration]` is given in slots from 4-10,000.

3.18.2 Responses

If the operation is successful, the response is:

```
AT-ZV HoldMode
```

3.19 HostEvent

The `HostEvent` command is used to enable/disable the host notification strings. This will override the default setting in the dynamic configuration only for the current session; until reset.

3.19.1 Syntax

```
AT+ZV HostEvent [Enable/Disable]
```

Where `[Enable/Disable]` is an "e" or "E" character to enable this parameter and a "d" or "D" character to disable it.

3.19.2 Responses

If the feature is successfully enabled, the response is:

```
AT-ZV HostEvent Enabled
```

If the feature is successfully disabled there is no response because the events have been disabled.

3.20 LocalName

The `LocalName` command is used to set the name of the device to the name that is reported during device discoveries. By default, the kcSerial interface uses "KCWirefreeDevice". Changing the name using this command does not permanently change the local name.

3.20.1   Syntax

```
AT+ZV LocalName [name]
```

Where [name] is a string for the new local name (up to 50 characters). The space character is allowed; the name is assumed to be all text up to the end of the command.

3.20.2   Responses

If the operation is successful, the response is:

```
AT-ZV LocalNameOk
```

3.21 Park

The **Park** command is used to switch a device from active mode to park mode.

3.21.1   Syntax

```
AT+ZV Park [BD address] [Beacon Period]
```

Where [BD address] is the BD address of the device to be switched to active mode. [Beacon Period] is given in slots from 200-10,000.

3.21.2   Responses

If the operation is successful, the response is:

```
AT-ZV ParkMode
```

3.22 RemoteCommand

The **RemoteCommand** command is used to enable/disable the remote command mode. This setting is stored in persistent memory, and will be retained after each reset. Additionally, the new setting will take effect upon the next device reset.

3.22.1   Syntax

```
AT+ZV RemoteCommand [Enable/Disable]
```

Where [Enable/Disable] is an "e" or "E" character to enable this parameter and a "d" or "D" character to disable it.

3.22.2   Responses

If the feature is successfully applied, the response is:

```
AT-ZV RemoteCommand [Enabled/Disabled]
```

3.23 RemoteCmdDisconnect

The **RemoteCmdDisconnect** command is used to disconnect a remote command connection. This command only applies to the server side of the link; the remote device. The client or local device, if using the kcSerial interface, should use a SPPDisconnect command since it is not in Remote Command mode.

3.23.1   Syntax

```
AT+ZV RemoteCmdDisconnect
```

3.23.2   Responses

If the operation is successful, the response is:

```
AT-ZV RemoteCmdModeClosed
```

3.24 Reset

The **Reset** command is used to reset the kcSerial interface. This is provided in the event that a host application wants to perform a software reset for error recovery. There is a response prior to reset to verify the command was received by the kcSerial interface.

3.24.1   Syntax

```
AT+ZV Reset
```

3.24.2   Responses

If the operation is successful, the response is:

```
AT-ZV ResetPending
```

3.25 Security

The `security` command is used to set the security level of the device in use. By default, security level none is used.

- Variable pin type (the pincode request event will always be received by the application from the stack), and
- 128-bit unit key.

Service level security, level 2, is not currently supported. The security setting is not preserved in non-volatile memory.

3.25.1   Syntax

```
AT+ZV Security [level]
```

where `[level]` is the type of security to use:

- None
- Link (default)

3.25.2   Responses

If the operation is successful, the response is:

```
AT-ZV SecurityOk
```

3.26 SmartCableSetup

The `SmartCableSetup` command is used enable and configure a Smart Cable device. A device's BD Address is specified with which to automatically establish a connection; replacing the need for AT connection commands. This command will override the dynamic configuration of a Smart Cable device until the `DeleteSmartCable` command is issued; it is saved in non-volatile memory.

3.26.1   Syntax

```
AT+ZV SmartCableSetup [BD address] [Attempts] [Interval]
```

Where [BD address] is the BD address of the remote device to page and attempt to connect. [Attempts] 0 – 999 is the number of pages the will be attempted to the specified device until a connection is successful. A value of 0 will not automatically page the remote device, however, GPIO 7 may be asserted to manually send a page. A value of 1000 will perform unlimited pages until connected. be switched to active mode. [Sniff Interval] is given in slots from 66-10,000. [Interval] 1-1000 is the number of 100ms intervals (0.1sec to 100 sec) between page attempts. This interval is in addition to the amount of time required by the page attempt itself.

3.26.2   Responses

If the operation is successful, the response is:

```
AT-ZV SmartCableConfigDone
```

3.27 Sniff

The **sniff** command is used to switch a device from active mode to sniff mode.

3.27.1   Syntax

```
AT+ZV Sniff [BD address] [Sniff Interval]
```

Where [BD address] is the BD address of the device to be switched to active mode. [Sniff Interval] is given in slots from 66-10,000.

3.27.2   Responses

If the operation is successful, the response is:

```
AT-ZV SniffMode
```

3.28 SPPConnect

The **SPPConnect** command is used to initiate a connection with the specified device. The remote BD address must be specified. The remote Service is optional. If not specified, the first registered SPP service will be used by default.

3.28.1   Syntax

```
AT+ZV SPPConnect [BD Addr] [Service]
```

Where [BD Addr] is the remote devices BD Address to page. [Service] is the specific service on the remote device; optional.

3.28.2 Responses

If the connection is successful, the response is:

```
AT-ZV ConnectionUp
AT-ZV -BypassMode-
```

If the connection cannot be completed, the response is:

```
AT-ZV SPPConnectionClosed
```

3.29 SPPDisconnect

The `SPPDisconnect` command is used to terminate a connection with the remote device.

3.29.1 Syntax

```
AT+ZV SPPDisconnect
```

3.29.2 Responses

If the connection is successful, the response is

```
AT-ZV SPPConnectionClosed
```

3.30 StreamingSerial

3.30.1 Syntax

```
AT+ZV StreamingSerial [Enable/Disable]
```

Where `[Enable/Disable]` is an "e" or "E" character to enable this parameter and a "d" or "D" character to disable it.

3.30.2 Query

An alternative syntax may be used to query the current StreamingSerial feature status. This syntax is not supported by other commands.

```
AT+ZV StreamingSerial
```

3.30.3 Responses

If the feature is successfully applied or queried, the response is:

```
AT-ZV StreamingSerial [Enabled/Disabled]
```

3.31 UpdateInquiryScan

The `UpdateInquiryScan` command is used to modify the Inquiry scan parameters: mode, duration, and interval.

3.31.1   Syntax

```
AT+ZV UpdateInquiryScan [mode] [duration] [interval]
```

where[mode] is the discoverable mode:

- 0: non-discoverable
- 1: limited discoverability – NOT SUPPORTED
- 2: discoverable

[duration] is the scan length in slots; 18 to 4096. The default duration is 18 slots.

[interval] is the period between scans in slots; 18 to 4096. The default interval is 2048 slots

3.31.2   Responses

If the command is successful, the response is:

```
AT-ZV InquiryScanUpdateDone
```

3.32 UpdatePageScan

The **UpdateInquiryScan** command is used to modify the Page scan parameters: mode, duration, and interval.

3.32.1   Syntax

```
AT+ZV UpdatePageScan [mode] [duration] [interval]
```

Where [mode] is the connectable mode:

- 0: non-connectable
- 1: connectable

[duration] is the scan length in slots; 18 to 4096. The default duration is 18 slots.

[interval] is the period between scans in slots; 18 to 4096. The default interval is 2048 slots

3.32.2   Responses

If the command is successful, the response is:

```
AT-ZV PageScanUpdateDone
```

3.33 Version

The **version** command is used to return the current version of the kcSerial interface.

3.33.1 Syntax

```
AT+ZV Version
```

3.33.2 Responses

If the operation is successful, the response is:

```
AT-ZV kcSerialVer [x.y]
```

where [x.y] is the current version of the kcSerial Interface.

4   Error Responses

This chapter details the error responses that occur under specific circumstances.

There are seven error responses that can occur beyond error responses specific to a particular command (e.g., `Discovery`). They are:

- ***ErrConnect***
- ***ErrExecution***
- ***ErrFormat***
- ***ErrInProgress***

- ***ErrInvalidParam***
- ***ErrNumParam***
- ***ErrUnknownCmd***

The following subsections detail the different error responses. These error responses replace the original error response

```
AT-ZV ErrorCommand
```

4.1   ErrConnect

The ***ErrConnect*** error response will be sent if kcSerial has a valid connection established and the command cannot be executed while connected (even in the command mode). Examples of commands that produce this error response are given in the following table.

Table 2. Example Commands that Generate <ErrConnect>

Example	Reason
AT+ZV Security None	Changing security level while a connection is up.
AT+ZV Discovery	Performing a device discovery while a connection is up.
AT+ZV SPPConnect 00043e000000	Establishing a connection to a device while a connection is exists with another.

4.2   ErrExecution

The ***ErrExecution*** error response will be sent if the command cannot complete for any reason.

Examples of commands that produce this error response are given in the following table.

Table 3. Example Commands that Generate <ErrExecution>

Example	Reason
AT+ZV Security None	Execution of command with correct syntax failed.

Example	Reason
`AT+ZV Discovery`	Execution of command with correct syntax failed.

4.3   ErrFormat

The *ErrFormat* error response will be sent if kcSerial receives a command (text terminated by a carriage return or line feed) that does not match the expected format of starting with "`AT+ZV `".

Examples of commands that produce this error response are given in the following table.

Table 4. Example Commands that Generate <ErrFormat>

Example	Reason
`AT-ZV Discovery`	A valid command name that does not start with the right prefix.
`SPPConnect 00043e000000`	A command does not start with "`AT+ZV `".
`abcdef`	A command does not start with "`AT+ZV `".

4.4   ErrInvalidParam

The *ErrInvalidParam* error response will be sent if the parameters for the requested command are not correct. The parameter(s) will be echoed back to the user starting from the parameter that was rejected.

Examples of commands that produce this error response are given in the following table.

Table 5. Example Commands that Generate <ErrInvalidParam>

Example	Response	Reason
`AT+ZV SPPConnect 8136`	`AT-ZV ErrInvalidParam 8136`	Numeric parameter not specified with required number of digits (BD address must always be 12 hex characters).
`AT+ZV SPPConnect 00043e008136 GOEP`	`AT-ZV ErrInvalidParam goep`	Unrecognized (or unsupported) symbolic parameter used.
`AT+ZV ChangeBaud 1600`	`AT-ZV ErrInvalidParam 1600`	Numeric parameter is out of range (specified baud rate is not supported by command).
`AT+ZV EnableBond 00043e000000 12345678901234567`	`AT-ZV ErrInvalidParam 12345678901234567`	String parameter (PIN)has too many characters.

4.5   ErrNumParam

The *ErrNumParam* error response will be sent if there are too few parameters for the requested command. A command sent with too many parameters does not generate an error; instead, the extra parameters are ignored.

⚠️ **Note: some commands will accept a variable number of parameters.**

Examples of commands that produce this error response are given in the following table.

Table 6.   Example Commands that Generate <ErrNumParam>

Example	Reason
AT+ZV Security	No parameters were specified.
AT+ZV Sniff	The minimum number of parameters was not specified.

4.6   ErrUnknownCmd

The *ErrUnknownCmd* error response will be sent if the requested command is not recognized. The unrecognized command will be echoed back to the host. Any parameters given will be ignored.

```
AT-ZV ErrUnknownCmd [unrecognized command]
```

An accepted command always starts with the command identifier:

```
AT+ZV
```

4.7   ErrInProgress

ErrInProgress is sent in response to **Discovery** command being issued when the previous one is still in progress.

```
AT+ZV Discovery [unrecognized command]
```

Examples of commands that produce this error response are given in the following table:

Table 7.   Example Commands that Generate <ErrInProgress>

Example	Reason
AT+ZV Discovery	Trying to do discovery when the previous one has not completed.

4.8   Commands and Associated Errors

The table below summarizes which of the commands produce specific error responses. If a command can produce an error response, the column will be marked with an X.

Table 8. Possible ErrConnect Error Responses

Example	ErrInvalidParam	ErrNumParam	ErrExecute	ErrConnect	ErrInProgress
Bond	X	X	X		
Bypass					
ChangeBaud	X	X			
ChangeDefaultBaud	X	X			
DefaultLocalName		X	X		
DeleteSmartCable					
DisableBond					
Discovery	X	X	X		X
DUNConnect	X	X		X	
DUNDisconnect					
EnableBond	X	X	X		
EraseBondTable					
ExitPark	X		X		
ExitSniff	X		X		
GPIOConfig	X	X			
GPIORead	X				
GPIOWrite	X	X			
Hold	X	X	X		
LocalName		X	X		
Park	X	X	X		
RemoteCmdDisconnect					
Reset					
Security	X	X	X		
SmartCableSetup	X	X			
Sniff	X	X	X		
SPPConnect	X	X		X	
SPPDisconnect					
UpdateInquiryScan	X	X	X		
UpdatePageScan	X	X	X		
Version					

5   Other Responses

The following subsections describe the 4 types of responses that occur under specific circumstances, not necessarily as a result of a specific command. They are:

- Reset
- Escape Sequence
- Controlled Disconnect
- Unexpected Disconnect

5.1   Reset

Upon either hardware reset or software reset (such as the `Reset` command), kcSerial will respond as follows after the reset is complete:

```
AT-ZV -CommandMode-
AT-ZV BDAddress [BD addr]
```

Because the BD address of the local device is reported during this response, the response is different than a response to the Escape Sequence.

5.2   Escape Sequence

If the Escape sequence, "`^#^$^%`", is received and no connection is active, kcSerial will immediately respond with:

```
AT-ZV -CommandMode-
```

When the Escape Sequence is received while a connection is still active and there is no data for 2 seconds, kcSerial will respond (after those 2 seconds of no data) with the same string.

kcSerial will now be in command mode.

5.3   Controlled Disconnect

If the local host initiates a disconnect, it must first put the kcSerial interface into command mode (see the section on Escape Sequence directly above). After a successful disconnect command, the following response is made:

```
AT-ZV ConnectionDown
```

5.4   Unexpected Disconnect

Bluetooth connections may be unexpectedly dropped (e.g., in changing RF conditions). Although it is generally assumed that a disconnect will be "negotiated" on the application level, the remote device may initiate a disconnect. When that happens, the disconnect may be unexpected. This section applies to both the general case and the unexpected disconnect.

It is useful for the local host to be notified that a connection has been terminated when it isn't controlling the termination. An unexpected disconnect is essentially defined as a disconnect that occurs while in bypass mode. If this happens, kcSerial will respond with:

```
###NO CARRIER
AT-ZV -CommandMode-
```

It is the responsibility of the host to prevent this string from appearing in the data stream during normal operation.

If a remote disconnect occurs during command mode, this notification string is also sent. It will not be sent, however, if an initial setup cannot be established or if the disconnect is requested by the local device.

Hardware handshaking is not used to indicate a disconnection in this implementation. Modems can use DCD (data carrier detect) to notify the DTE (data terminal entity, e.g., computer) that a connection is either available or unavailable.