Confidential: This document may not be distributed without approval from Bibliotheca.

bibliotheca®

CONFIDENTIAL

RFIDIFServer’s WebSocket Interface

User Manual

Version Number: 1.2.14

Although we make every effort to ensure the information is correct at the time of release, it is possible that specifications and features may vary or change over time. bibliotheca therefore cannot give any guarantee as to the completeness and accuracy of the information contained within this document.

CONFIDENTIAL

1 File Version History

Version	Date	Editor	Changes
1.00	2017-01-24	Henrik Jensen	Tested and up-to-date revisions
1.10	2017-02-06	Lasse Ronnenberg	New Template and proof reading
1.20	2017-02-14	Lasse Ronnenberg	Added "Disabled" and "ReaderStatus"

2 The WebSocket Interface for RFIDIFServer

This RFIDIFServer interface is based on a WebSocket connection.

The default WebSocket **port number** is **6000**.

The communication is mainly event based. i.e. The server sends a message each time a change of tag status occurs, but the client can also send requests for more information or for changing the information in a tag.

All messages are encoded in JSON and may contain more data values than this document describes (ignore those extra values). The order of the values is irrelevant as there are no requirements to the sequence.
For more information about the JSON standard, visit: http://www.json.org/

The communication is asynchronous; this means it is possible to send other commands while waiting for an answer, or for tag reports to suddenly show up while you were waiting for an answer to a different command.

If a command fails, a special error response will be posted.

3 How to configure RFIDIFServer for WebSocket

The WebSocket interface is enabled by adding the following line to the `[config]` section of the file **RFIDIFServer.ini** :

```
[config]

StaffInterface=WS
```

The default port number used by bibliotheca is 6000, but since browsers often blocks this port due to the X Window Protocol, you may need to use the port number 7000 (or higher) for testing purposes etc.

To change the port number to 7000 (if needed for testing) you can add the following line:

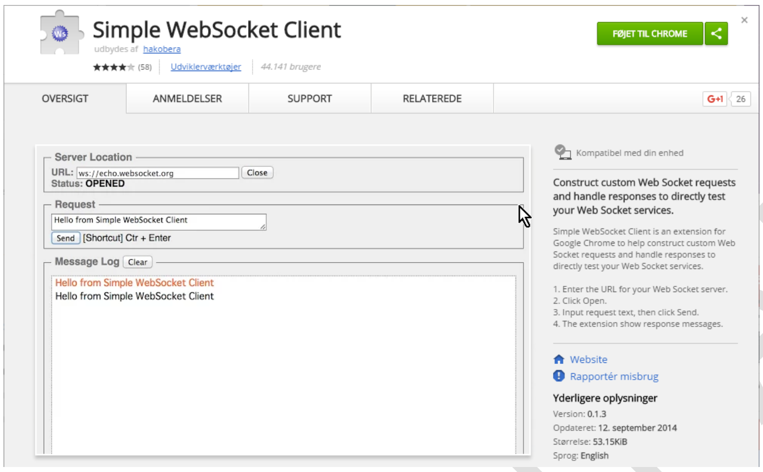
```
[config]

StaffInterface.WS.Port=<new port_number>
```

CONFIDENTIAL

4 Understanding the WebSocket interface

Start by installing a WebSocket plugin for your browser, such as “Simple WebSocket Client” for Google Chrome or Mozilla Firefox.



URL: ws://echo.websocket.org
Status: OPENED

Hello from Simple WebSocket Client
[Shortcut] Ctr + Enter

Hello from Simple WebSocket Client
Hello from Simple WebSocket Client

Construct custom Web Socket requests and handle responses to directly test your Web Socket services.

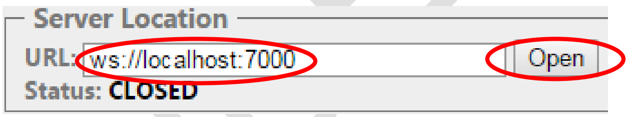
Simple WebSocket Client is an extension for Google Chrome to help construct custom Web Socket requests and handle responses to directly test your Web Socket services.

1. Enter the URL for your Web Socket server.
2. Click Open.
3. Input request text, then click Send.
4. The extension show response messages.

Yderligere oplysninger
Version: 0.1.3
Opdateret: 12. september 2014
Størrelse: 53.15KiB
Sprog: English

Picture 1: Install a WebSocket plugin for your browser.

Enter the URL and port number for RFIDIFServer. It is “localhost” if RFIDIFServer is installed on the same PC as the WebSocket application, and “7000” if you added the line `StaffInterface.WS.Port=7000` to the file **RFIDIFServer.ini**



URL: ws://localhost:7000
Status: CLOSED

Picture 2: Open a connection to RFIDIFServer through the WebSocket plugin.

Once the connection is established, you may enter a command from the list below to verify RFIDIFServer is working as intended.

4.1 Example

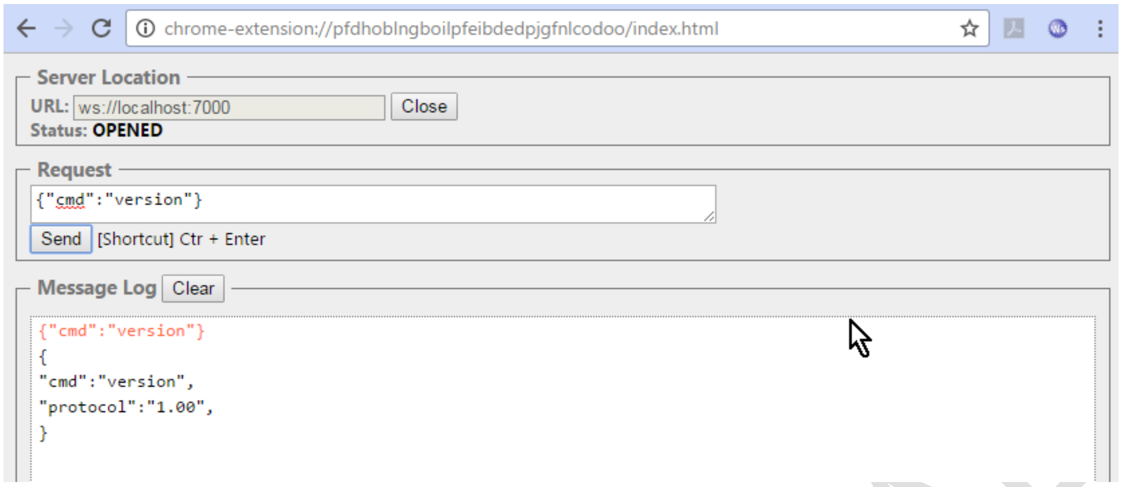
Enter the command:

{"cmd":"version"}

and press the “Send” button. The result should look somewhat like this:

{“cmd”:”version”,”protocol”:”1.00”}

CONFIDENTIAL



chrome-extension://pfdhoblngboilpfeibdedpjgfnlcodoo/index.html

Server Location
URL: ws://localhost:7000 Close
Status: OPENED

Request
{"cmd":"version"}
Send [Shortcut] Ctr + Enter

Message Log Clear

```
{"cmd":"version"}
{
"cmd":"version",
"protocol":"1.00",
}
```

Picture 3: Sending {“cmd”:”version”} will get the reply {“cmd”:”version”,”protocol”:”1.00”,}

4.2 Understanding the command formatting

Example:

```
Tx: {"cmd":"version"}
Rx: {"cmd":"version","protocol":"1.00"}
```

Tx is the (Transmitted) command sent to RFIDIFServer and Rx is the (Received) reply.
Both the command and the reply starts and ends with curly brackets: { … }

In Rx above, you may notice that the reply is two-fold. First RFIDIFServer repeats the command it received:

cmd = “version” ← simplified from "cmd":"version"

A comma (,) separates each part of the reply. The next part says:

Protocol = 1.00 ← simplified from "protocol":"1.00"

The text “cmd” and “protocol” are **fields**. Each field has a **value** just to the right of the “:” colon. All **fields** are surrounded by citation marks “”, as well as all **values** that are not simple numbers.

All commands and replies follow this pattern. A comma separates each field that is parts of the reply.

The next section contains a list of all of the commands that can be sent to RFIDIFServer through the WebSocket interface.

CONFIDENTIAL

5 Commands

5.1 Version: How to get the version number

```
Tx: {"cmd":"version"}

Rx: {"cmd":"version","protocol":"1.00"}
```

Sending this command will get the major and minor version numbers of RFIDIFServer's protocol version.

Field	Data format	Description
Protocol	`major.minor`	All minor protocol upgrades will be backward compatible.

5.2 ReaderStatus: How to verify the RFID reader is working

If you send the following command, RFIDIFServer will give you the current status of the RFID reader.

```
Tx: {"cmd":"readerStatus"}

Rx: {“cmd":"readerStatus","status":"online"}
Or Rx: {"cmd":"readerStatus","status":"fail"}
```

At this time, the only possible answers to this command are “online” or “fail”.
You can use this command to verify that the RFID reader is ready for use.

5.3 Tag: Understanding the format of the Tag reports

There is no command for requesting a Tag report. The report is automatically sent when a new tag is placed on the antenna. A list of events that generate a Tag report is found below.
The field “reason” indicates exactly why the report was sent.

```
Rx: {"cmd":"tag","id":"1234567890","type":"Item",
                         "reason":"Firsttime new complete","reader":"1"}
```

(Please note the first field “id”, which is the tag's barcode).

Field	Data format	Description

CONFIDENTIAL

id	`String`	The barcode of the tag. Used for all reference to this tag. (The barcode may contain letters).
type	`String: 'Undef', 'Item', 'Card'`	What kind of tag was read (see below)
reason	`String: 'None', 'Firsttime new partial', 'Firsttime new complete', 'Firsttime complete', 'Partial', 'Complete', 'Removed', 'Reader empty', 'Moved','Tag found'`	Indicates why the report was sent. Below is a list of the events that can generate a Tag report.
Reader	`Number` (Usually "1")	RFIDIFServer can control multiple readers. This is the number of the reader that generated the report.

Types of Tags:

Value	Description
Undef	Unknown tag - Either an empty tag or a tag with an unknown Data Model.
Item	An item.
Card	A user library card.

Reasons or events that generate a Tag report:

Value	Description
None	Undefined response, this must be ignored.
Firsttime new partial	A new media pack is detected, but it is not complete.
Firsttime new complete	A new item is detected, all parts are present and it can be checked in/out.
Firsttime complete	An item reported as "Firsttime new partial" is now complete.
Partial	One or more parts of a media package was removed.
Complete	The media package is complete again.
Removed	The item was completely removed from the reader.
Reader empty	No more items on the reader.
Tag found	A tag was detected on reader, but data has not been read yet. This is an early warning about a tag. If it is a valid tag a firsttime… report will arrive later.
Moved	Tag was moved from one reader to another reader.
(any other response)	New reasons may be added later; these must be ignored.

Notes:

- ➔ All tags will be reported, empty tags will show up as "undef" and receive a special ID number called a "uid:" which is 16 hexadecimal digits long.
- ➔ When items are place on the reader, the typical reports are: "TagFound", "Firsttime", "new" and "complete".

- ➔ If a tag is found at the limit of the antennas' reading range, a message of "Partial", "Complete" or "Removed" may be reported. This can usually be ignored.
- ➔ When the last item is removed from the antenna, the possible reports are: "Removed" and "Reader empty".

5.4 Value: How to read a specific field from a tag

```
Tx: {"cmd":"value","id":"1234567890","fields":"libraryId"}

Rx: {"cmd":"value","id":"1234567890","index":"","fields":{"libraryId":"bibliotheca"}}
```

Tx:

Field	Data format	Description
id	`String`	The barcode of the tag. Used for all reference to this tag.
Fields	`String`	List of fields to read, use a comma "," between field names.
Index (Optional)	`Integer`	What tag to read in a media package. Tags in a package is numbered from 1 to n. Index 0 or missing index is always valid and will read from any tag.

Rx:

Field	Data format	Description
id	`String`	The barcode of the tag. Used for all reference to this tag.
Index	`Integer`	As above
Fields	`Object`	Each field with value

For a list of valid field names, please see a tag dump of the used data model (This can be made with RFIDIF by entering a special code).

5.5 SetCheckoutState: How to change a tag's Security status

```
Tx: {"cmd":"setCheckoutState","id":"1234567890","security":"Activated"}

Rx: {"cmd":"setCheckoutState","id":"1234567890","security":"Activated"}
```

Tx:

Field	Data format	Description
id	`String`	The barcode of the tag. Used for

CONFIDENTIAL

		all reference to this tag.
Security	`String: 'Activated', 'Deactivated'`	Requested security state

Rx:

Field	Data format	Description
id	`String`	As above
Security	`String: 'Activated', 'Deactivated', 'Undef'`	Confirmation of requested state

If some items are missing in a media package an error response will be returned.

Security options:

Value	Description
Activated	The tag will give alarm in gates
Deactivated	The tag can be taken out of the library without any alarm.
Undef	The current state of the security is undefined.

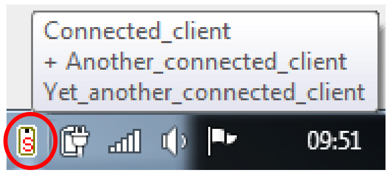
5.6 RemoteName: How to identify different connections to RFIDIFServer

While multiple clients can connect to the RFIDIFServer application at once, only one of them will receive the Tag reports and/or error messages generated. (Enter the command "enable" or "resend" to tell RFIDIFServer that this connection is the 'active' one). To keep track of multiple connections to RFIDIFServer, have each client send the command "cmd":"remoteName" as shown below:

```
Tx: {"cmd":"remoteName", "name":"A_name_for_this_client"}

Rx: {"cmd":"remoteName","server":"RFIDIFServer.exe on KN2-HP"}
```

Once you have named your connection, the tooltip of RFIDIFServer's icon (in the notification area) will change to show the names of all named connections. A plus sign (+) shows you which connection is the 'active' one. (If "Connected_client" submitted an "enable" or "resend" command, then the plus sign would move to that connection instead).


Connected_client
+ Another_connected_client
Yet_another_connected_client

Picture 4: The plus sign "+" marks the connection that will receive all Tag reports and error messages.

CONFIDENTIAL

5.7 Enable: How to tell RFIDIFServer where to send Tag reports

RFIDIFServer sends a Tag report when a new tag is detected, but the tag report only gets sent to the 'active' connection. To make the current connection (the one you are using to send this command to RFIDIFServer with) into the 'active' connection, send the following command. All further messages and error reports from RFIDIFServer will be sent on this connection.

```
Tx: {"cmd":"enable"}

Rx: {"cmd":"enable"}
```

The best time to send an "enable" request is when the user presses a key to "open reader" or whenever the application gets focus.

5.8 Disabled: How RFIDIFServer informs you about lost focus

RFIDIFServer only sends data to the connection that is "enabled". Only one connection is enabled at any given time. If RFIDIFServer sends you a "disabled" message at any given time, it means your connection is no longer the active one. This can happen if another connection sends an "enable" or "resend" command. If you receive a "disabled" message, you can use the "resend" command to make RFIDIFServer "enable" you and send you a Tag report of all the tags currently on the antenna. A "disabled" message will also be sent to the previous active connection.

```
Rx: {"cmd":"disabled"}
```

In short: You will not receive any tag reports when you are disabled. The "enable" and "resend" commands will change that.

5.9 Resend: How to re-send the Tag reports

The following command will (re)send a Tag reports with information about every tag currently found on the antenna. The command also changes the active connection so this connection gets the new Tag report.

```
Tx: {"cmd":"resend"}

Rx: {"cmd":"resend"}
```

In short: Resends all tag data on the reader and performs an "Enable" command. To make sure the report gets sent back to this connection.

A Tag report will be generated for each item on the reader after you request a "resend".

CONFIDENTIAL

5.10 DMName: How to read the name of the tag's Data Model

```
Tx:  {"cmd":"DMName","id":"1234567890"}

Rx:  {"cmd":"DMName","id":"1234567890","DMName":"DK V1"}
```

Tx:

Field	Data format	Description
Id	`String`	The barcode of the tag. Used for all reference to this tag.

Rx:

Field	Data format	Description
Id	`String`	As above
DMName	`String: 'DK V1', 'TagVision'`	Name of data model used on tag

5.11 ValidDataModel: How to validate the Data Model of a tag

The following command is useful for validating a tag after you have manually entered the fields and values. If the command returns "valid":1, then the Data Model is okay.

```
Tx:  {"cmd":"validDataModel","id":"1234567890"}

Rx:  {"cmd":"validDataModel","id":"1234567890","valid":1}
```

Tx:

Field	Data format	Description
Id	String	The barcode of the tag. Used for all reference to this tag.

Rx:

Field	Data format	Description
id	`String`	As above
ValidDataModel	`Integer: 0, 1`	Flag to signal valid data model.

5.12 TagCount: How to count how many items are within the antenna's range

This command counts the number of tags in range of the antenna.

```
Tx:  {"cmd":"tagCount"}
```

CONFIDENTIAL

```
Rx: {"cmd":"TagCount","items":2,"tags":4}
```

Rx:

Field	Data format	Description
items	`Integer`	Number of items in range. Empty tags are ignored and each media package are counted as one.
tags	`Integer`	Number of tags in range. This will include all tags in a media package and empty tags.

5.13 NumberOf: How to find out if an item is part of a collection

If the item belongs to a media package you can get its number in that media package by sending the following command.

```
Tx: {"cmd":"numberOf","id":"1234567890"}

Rx: {"cmd":"numberOf","id":"1234567890","parts":1,"tags":1,"inRange":1}
```

Tx:

Field	Data format	Description
Id	`String`	The barcode of the tag. Used for all reference to this tag.

Rx:

Field	Data format	Description
Id	`String`	As above
parts	`Integer`	Number of parts in this item.
tags	`Integer`	Number of tags in this item
inRange	`Integer`	Number of tags, of this item, in reader range.

5.14 GetItem: How to read a single item from the reader

This command is used when you only want a single item from the reader.

```
Tx: {"cmd":"getItem"}

Rx: {"cmd":"getItem","id":"","tags":1,"pack":1,"items":2}
```

CONFIDENTIAL

Rx:

Field	Data format	Description
id	`String`	The barcode of the tag. Used for all reference to this tag. If more than one item is present, the last reported "new" item is selected.
tags	`Integer`	Number of tags present in item. If this number is equal to "Media pack size" the item is complete.
pack	`Integer`	Number of tags in media pack.
items	`Integer`	Number of items in range. Empty tags are ignored and each media package are counted as one.

Multiple calls to this function will **not** iterate all tags on the reader, for that use "Resend" and collect the "Tag" reports.

5.15 ItemComplete: How to see if all parts of a collection are present on the reader

The following command will scan all of the tags on the antenna and determine whether every part of a collection is present. If no parts are missing, you will get "complete":1

```
Tx: {"cmd":"itemComplete","id":"1234567890"}

Rx: {"cmd":"itemComplete","id":"1234567890","complete":1}
```

Tx:

Field	Data format	Description
id	`String`	The barcode of the tag. Used for all reference to this tag.

Rx:

Field	Data format	Description
id	`String`	As above
complete	`Integer: 0, 1`	Flag to signal if item is complete. This is a test comparing NumberOfTags to NumberOfTagsInRange. If they are equal, then you will get a "complete = 1" (See the Rx example).

5.16 Program: How to program a tag

This command lets you program a specific tag or generate a barcode number for a tag.

```
Tx: {"cmd":"program","fields":{"id":"171994020"}}
   Or
Tx: {"cmd":"program","id":"171994020"}
   Or
Tx: {"cmd":"program","tags":1,"fields":{"id":"171994020","libraryId":"Bibliotheca"}}
```

```
Rx: {"cmd":"program","id":"171994020","tags":1}
```

If the id field is empty a number may be automatically generated, this number is taken from a pre-defined list and check digits might be added (Depends on configuration). The number programmed is returned in the answer.

Tx:

Field	Data format	Description
id	String or empty	The barcode of the tag. Used for all reference to this tag., leave empty for automatic generated number. This value can either be included in the fields or a separate value
tags	Integer	Number of tags to program, this number must match the actual number of tags on the reader. Use 0 or skip the parameter to accept any number of tags.
Fields	List with field names and values	Assign a value to a field, when no fields are specified, the predefined values are used.

See "TagCount" message field "Tag count" for information about number of tags in range.

Rx:

Field	Data format	Description
id	String	The barcode of the tag. Used for all reference to this tag.
tags	Integer	Number of tags programmed

5.17 Discard: How to erase a tag

The following command will erase the data on any tag located on the antenna.

```
Tx: {"cmd":"discard","id":"171994020"}

Rx: {"cmd":"discard","id":"171994020","discarded":1,"pack":1}
```

Tx:

Field	Data format	Description
id	String	The barcode of the tag. Used for all reference to this tag.

Rx:

Field	Data format	Description
id	String	The barcode of the tag. Used for all reference to this tag.
discarded	Integer	Number of tags discarded

CONFIDENTIAL

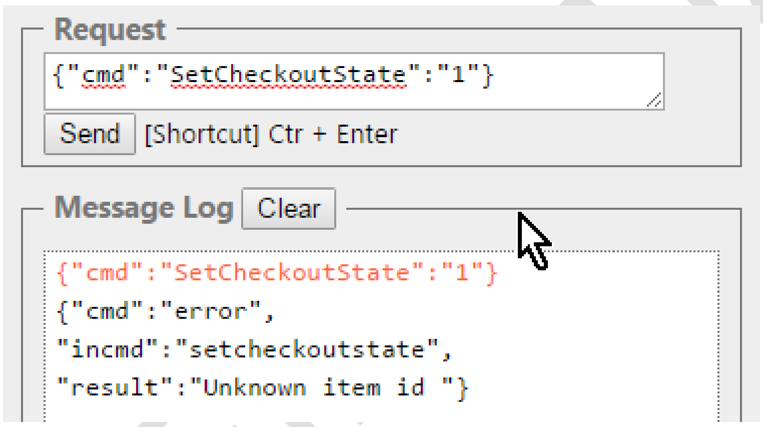
pack	Integer	Number of tags in media pack

5.18 Error: The format of the error messages

These responses are only sent if an error occurs. Check the syntax of your last command or read the error description and find out what went wrong.

> Remember that communication is asynchronous. If you send a command that edits a DataModel followed by a ValidDataModel, they might not execute in the correct order unless you wait in between.

Any failed commands will return an Error response.



```
{"cmd":"SetCheckoutState":"1"}
```

```
{"cmd":"SetCheckoutState":"1"}
{"cmd":"error",
"incmd":"setcheckoutstate",
"result":"Unknown item id "}
```

Picture 5: Example of an error message.

Rx:

Field	Data format	Description
Command id	`String: "error"`	Identification of failed command
Command id	`String`	Identification of command
Message	`String`	An internal error message

CONFIDENTIAL

6 Typical usage

6.1 Handling messages from RFIDIFServer

The client connected to RFIDIFServer must always be ready to receive the correct response for the commands issued, but it must also be able to handle Tag reports from RFIDIFServer at any time when an event like the "Tag found"-event generates them (See section 5.2 *ReaderStatus: How to verify the RFID reader is* working

If you send the following command, RFIDIFServer will give you the current status of the RFID reader.

```
Tx: {"cmd":"readerStatus"}

Rx: {"cmd":"readerStatus","status":"online"}
Or Rx: {"cmd":"readerStatus","status":"fail"}
```

At this time, the only possible answers to this command are "online" or "fail".
You can use this command to verify that the RFID reader is ready for use.
Tag: Understanding the format of the Tag reports). The system must also be ready to handle any Error messages at any time, as well as unexpected "disabled" commands.

6.2 Performing check-ins/check-outs

Simple check-ins/check-outs only need a few of the listed messages (SetCheckoutState).

1. Start by sending an "Enable" or "Resend" command. Both will make RFIDIFServer mark you as the active connection (so the Tag report gets sent to you) and the "resend" command also gets you the information about all tags currently on the antenna.
2. Create a local table for storing incoming Tag reports until you are ready to handle them.
3. Extract and process each tag in your table. (E.g. run "SetCheckoutState" on all items that are being checked out).
4. In case of an incomplete item, notify the user. (The command "NumberOf" can be used to obtain more information about the incomplete item).
5. (Handle any error reports that may be returned by RFIDIFServer).
6. (If, at any time you lose connection to RFIDIFServer, it will send you a "disabled" message. - E.g. If another user activates his connection, you will receive a "disabled" message and must act on it to regain the connection).

6.3 How to read a single item

Use the command "GetItem" and you will get a Tag report for the last item detected by the reader.

CONFIDENTIAL

6.4 How to program a tag

To program an item, the following two methods can be used:

The quick and unsafe way:
Use the command "Program" with "Number of Tags=0".

The safer way:
First use "TagCount". Then, if "Item count" is not zero, ask user for confirmation before overwriting tags.
Then use "Program" with "Number of tags" equal to the number obtained with the "TagCount" message.

6.5 Discard tag

To discard an item, simply use the "Discard" command with the barcode number of that item, (only that item will be discarded, other items on the reader will be ignored).

CONFIDENTIAL

bibliotheca a/s
Banemarksvej 50E, 2 sal.
DK-2605 Brøndby
Danmark

Tel.: +45 7027 1008
E-Mail: info@bibliotheca.com
E-Mail: support-dk@bibliotheca.com
www.bibliotheca.com

CONFIDENTIAL

Although we make every effort to ensure the information is correct at the time of release, it is possible that specifications and features may vary or change over time. bibliotheca therefore cannot give any guarantee as to the completeness and accuracy of the information contained within this document.