



MATRIX **LOGON**

HIGH-QUALITY WEB AUTHENTICATION

USER MANUAL

User Manual

The data and information contained in these documents can be altered without prior notice. Without the express written permission of TDi GmbH, no part of these documents can be reproduced or transmitted for any purpose whatsoever, regardless of how or by which electronic or mechanical means.
The general terms of trade of TDi GmbH apply. Diverging agreements must be made in writing.

Copyright © TDi GmbH TechnoData - Interware. All rights reserved.

Matrix Logon is a patented solution of TDi GmbH.

WINDOWS is a registered trademark of Microsoft Corporation.
The WINDOWS-logo is a registered trademark (TM) of Microsoft Corporation.

Software License

The software and the enclosed documentation are copyright-protected. By installing the software, you agree to the conditions of the licensing agreement.

Licensing Agreement

TDi GmbH (TDi for short) gives the buyer the simple, exclusive and non-transferable licensing right to use the software on one individual computer or networked computer system (LAN). Copying and any other form of reproduction of the software in full or in part as well as mixing and linking it with others is prohibited. The buyer is authorised to make one single copy of the software as a back-up. TDi reserves the right to change or improve the software without notice or to replace it by a new development. TDi is not obliged to inform the buyer of changes, improvements or new developments or to make these available to him. A legally binding promise of certain qualities is not given. TDi is not responsible for damage unless it is the result of deliberate action or negligence on the part of TDi or its aids and assistants. TDi accepts no responsibility of any kind for indirect, accompanying or subsequent damage.

Compliance with the CE/FCC Regulations



The Matrix device was tested for compliance with the limits for Class B digital devices and approved.

Operation is subject to the following requirements:

1. The device must not cause radiated noise
2. The device must be able to handle radiated noise including noise which can lead to undesired operation

The products complies with the limits according to EN55022 Class B, EN50081-1, EN50082-1 and EN55024.
Alterations to the product without the express approval of TDi GmbH can mean that the CE/FCC regulations are no longer complied with. In this case, the user can lose the right to utilise the product.

Preface

Thank you for using TechnoData Interware products. We trust that this manual will prove useful as you integrate the Matrix System into your programming environment . Please feel free to contact us if you have any questions, comments or suggestions regarding this manual.

Our USB-Sticks are called “Dongle” (for the use for software protection). With the use of Matrix Logon and authentication we call them “Token”.

You will find the most up-to-date version, examples and readme files for Mac OS X and Linux plus additional tools for optimizing your work with Matrix on our website www.tdi-matrix.com. We are always interested to hear your comments and suggestions.

You can contact us via E-Mail: support@tdi-matrix.com

Note: Please always include the version and model number when addressing questions to our hotline!

Matrix Secure WEB-Logon

Do you know who is logging in to your web site?

The problem with passwords is that it is too easy to lose control of them. People give their passwords to other people. People write them down, and other people read them. People send them in emails, and that emails are intercepted. People use them to log into remote servers, and their communications are eavesdropped on. Passwords are also easy to guess. And once any of that happens, the password no longer works as an authentication token because you can never be sure who is typing in that password.

Password replacement and logon solution

Matrix Logon is a hardware based Two-Factor-Authentication which solves the password problem.

The Matrix USB dongle is a hardware key which replaces the password. The authentication is performed through data encryption inside of the secure Matrix hardware. The encrypted authentication token changes every time when a logon is requested to ensure that an intercepted token won't be usable twice.

How does it work?

Matrix Logon is based on two components:

1. The Matrix-dongle and the WEB-Client application on the client side
2. The WEB server on the vendor side

When the user needs to log into a server, the server sends a random based request. The user must connect the Matrix USB-dongle and to type in a personal PIN code.

The server request, the users PIN and some kind of other data blocks will be encrypted inside of the secure Matrix-dongle. The encrypted result token is then send back to the server.

Sensitive information such as encryption key is stored safely in the Matrix-dongle and on the server.

Matrix Logon provides developers with a secure solution for client authentication via internet.

Only users who have the valid Matrix-dongle and the corresponding PIN code will be allowed to log into the server application/web site.

Matrix Logon can be used with all web browsers and is NOT based on cookies or Java applets.

On the client side a WEB-Client application (EXE) is used to pass the server requests to the Matrix-dongle.

The server components are available with all PHP sources for easy integration in your server application and database.

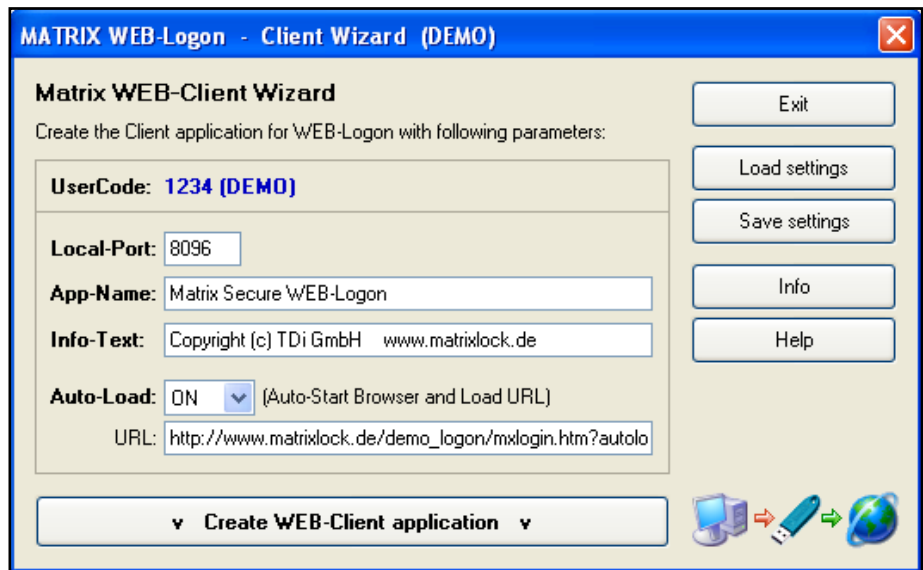
The section below will describe the request/response patterns for client and server side.

WEB-Client application

The WEB-Client application (EXE) which you have to deliver to your customers represents a part of your product/solution. Therefore we do not provide you with the final WEB-Client application. Instead, to offer you highest flexibility, we provide you with the Matrix WEB-Client Wizard.

Matrix WEB-Client Wizard consists of a tool named **MxGenWeb** which allows you to create your own customized WEB-Client application named MxWeb32.exe.

The following basic settings are available in **MxGenWeb**:



Local-Port

Here you can specify which local port should be used by the WEB-Client application for the HTTP requests. The same port has to be used at the server's side (server scripts).

Note: Take care to use only free ports.

App-Name / Info-Text

The name of the WEB-Client application and the message text displayed in its Info-Dialog can be modified here in order to satisfy your particular requirements.

Auto-Load

Here you can specify if the WEB-Client application should automatically start or not the internet browser with your Logon-Site when the user plugs in the Matrix-dongle to the PC. This is a default setting for the generated WEB-Client application which can be modified by the user.

Save/Load settings

To facilitate the management of different WEB-Client application files, you can save your settings as a project. Saved projects can be reloaded for further use.

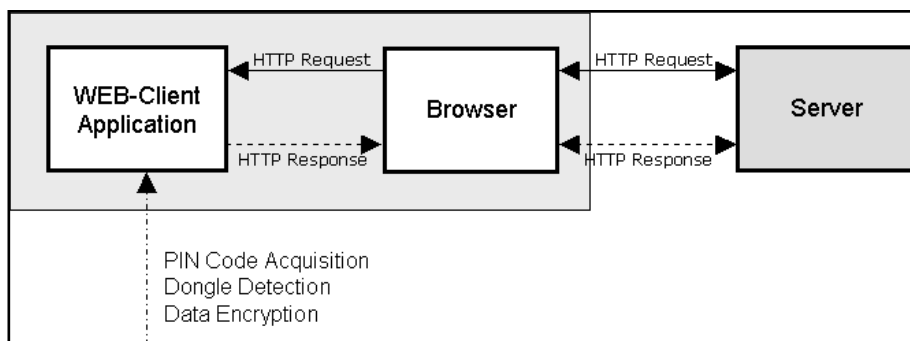
v Create WEB-Client application v

Generate the WEB-Client application (default name MxWeb32.exe), which you send to your customer.

Request / Response Pattern

Requests are generated on the server-side (HTTP-Request), sent to the Web browser and executed in the WEB-Client application.

The WEB-Client application receives the HTTP-Request, performs PIN code entry, dongle detection and data encryption. The encrypted results (HTTP-Response) are send back to the server.



The WEB-Client application receives dedicated HTTP Requests and send out correct HTTP Responses and encrypted data.

The requests will be something like below:

```

GET
/?site=www.test.com/
logon.php&ssid=4f6de45a15a68059a5eb4639fafe1fo6&action=clientOk HTTP/1.1
Host: localhost:8096
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; ja-JP; rv:1.7.8) Gecko/
20050511 Firefox/1.0.4
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9, text/
plain;q=0.8,image/png
.....
  
```

Using the HTTP request, the WEB-Client application must understand what is requested. For this, only the second line of the request is used:

```
/?site=www.test.com/logon.php&ssid=4f6de45a15a68059a5eb4639fafe1fo6&
action=clientOk HTTP/1.1
```

In this example, there are 3 parameters: site, ssid, action.

site	...	this is the site the browser must redirect to
ssid	...	server script's session ID
action	...	what action is expected from the client

The WEB-Client application supports 4 request actions:

“clientOk”, “enterPin”, “readData” and “writeData”.

- For “clientOk” request action, the client response will be “clientOk”.
- For “enterPin” request action, the client response will be “pinEntered”.
- For “readData” request action, the client response will be “dataRead”.
- For “writeData” request action, the client response will be “dataWrite”.

The server request must contain all the information that the WEB-Client application needs to perform the expected operation and to return the result to the server.

Request “clientOk”

This request is the first action to ensure the WEB-Client application is up and running. If the WEB-Client application is not running, the browser gets no response and knows that the WEB-Client application is not running.

In this request the server script will have to pass **4** parameters:

- site** - is the site where the browser must redirect to.
- ssid** - is the server script's session ID “ssid”.
- data1** - must be a 16 byte block of hexadecimal data (e.g. random data) generated by the server.
(16 bytes of hexadecimal data = 32 bytes of text)
- action** - is the action name “clientOk”.

Request example:

```
/?site=www.test.com/logon.php&ssid=4f6de45a15a68059a5eb4639fafe1fo6&data1=223344&action=clientOk
```

Response “clientOk”

For the request “clientOk” the WEB-Client application will respond with the same action name “clientOk”.

In this response the WEB-Client application will send back 4 parameters:

- PHPSESSID** - is the session ID “ssid” received in the request action “clientOk”.
- data1** - the WEB-Client application returns the same unchanged “data1” which has been received from the server in the request action.
- action** - is the same action name “clientOk” as in the request action.
- status** - is always 0 in this response action.

Response example:

```
http://www.test.com/logon.php?PHPSESSID=4f6de45a15a68059a5eb4639f_afe1f06&data1=223344&action=clientOk&status=0
```

Remark: The server script checks whether “data1” contains the same data as it sent to the client. If yes, the communication to the WEB-Client application is ok. Otherwise the server script will trap to the error handler.

Request “enterPin”

This is the second request action from the server which initiates in the WEB-Client application the PIN code acquisition and the generation of encrypted data over the Matrix-dongle.

In this request the server script will have to pass **2** parameters:

- action** - is the action name “enterPin”.
- timeout** - is a timeout value in milliseconds for the PIN code acquisition. When this time elapses before the user has entered the PIN code, the WEB-Client application will auto-close the PIN dialog and not return any response.

Request example:

```
?action=enterPin&timeout=25000
```

In this example the timeout for the PIN acquisition is set to 25 seconds.

Remark: The timeout must be handled also in the server script because the WEB-Client application will not return any response in case of an timeout.

Response “pinEntered”

For the request “enterPin” the WEB-Client application will respond with the action name “pinEntered”.

In this response the WEB-Client application will send back 5 parameters:

- PHPSESSID** - is the session ID “ssid” received in the request action “clientOk”.
- data1** - the WEB-Client application returns the same unchanged “data1” which has been received from the server in the request action “clientOk”.
- data2** - the WEB-Client application returns an encrypted data block. The data block is encrypted over the Matrix-dongle connected to the users PC and will represent the user authentication token.
The structure of this data block is described below.
- action** - the action name in this response is “pinEntered”.
- status** - the status parameter is the return code in this client response can have following values:
 - is the Serial-No. of the users Matrix-dongle
 - PIN code entry was cancelled by user
 - 1 Matrix-dongle not present or hardware error

Response example:

```
http://www.test.com/logon.php?PHPSESSID=4f6de45a15a68059a5eb4639f  
a fe1f06&data1=223344&data2=0011-2233...4455&action=pinEntered&  
status=1234567890
```

Description of the encrypted data block “data2” returned from the WEB-Client application in the response “pinEntered”

The encrypted “**data2**” consists of 6 segments, each 4 byte hexadecimal in the form:



The segments 1, 3, 5, 6 represents the encrypted result of “**data1**” which was initially send from the server.

The 16 bytes hexadecimal block of “**data1**” send form the server in the “**clientOk**” request is splitt in the WEB-Client application in 4 hexadecimal segments (each 4 bytes long) and encrypted over the Matrix-dongle connected to the users PC.

This encrypted 4 segments are returned by the WEB-Client application in the segments 1, 3, 5, 6 of “**data2**”.

The segments 2, 4 represents the encrypted result of “Token Serial-No.” and “User-PIN”.

The server application must decrypt the received “**data2**” block and check its accuracy. Decryption of “**data2**” should take place over each two segment pairs 1-2, 3-4, 5-6.

After decryption of “data2” following content should be found in the 6 decrypted segments:

- segments 1, 3, 5, 6 should be the same as the data received in “**data1**”.
- segment 2 should contain the same Serial-No. as the one received in “status”.
- segment 4 should contain the users PIN code.

In order to decrypt “**data2**”, the server application must have knowledge about the correct key for the decryption (the key which is also present in the Matrix-dongle).

For this purpose, the Serial-No. of the Matrix-dongle is delivered in the “status” parameter. The Serial-No. offers you the possibility of making use of a customer database in which you stored the encryption key for each delivered Matrix-dongle.

With the received Serial-No. you can read out from your database the corresponding Key to process the decryption of “**data2**”.

The authenticity is determined by a comparison of the decrypted results.

The key does not show outside the Matrix-dongle or the server-application. Therefore it can not be intercepted and abused.

Request “readData”

This request is the action to be used to read data from the memory of the Matrix-Token.

In this request the server script will have to pass 4 parameters:

- data1** - the server must send in “data1” the encrypted UserCode of the Matrix-Token. Encryption must be done on the server side with the same Key as the one stored in the Matrix-Token. This buffer is a hexadecimal string with the format:
R1-UC-R2-UC (separator must be “-”)
Where ‘R1’ and ‘R2’ are random numbers and ‘UC’ is the UserCode of the token. The WEB-Client application decrypts the buffer “data1” and checks the accuracy of the UserCode.
- fpos** - is the position (variable number) where to start reading in the token memory.
- fcnt** - is the number of variables to be read from the token memory. The maximum number of variables to be read at once is 79.
- action** - is the action name “readData”.

Request example:

```
/?fpos=1&data1=6B2AB0D7-BC03A6F0-17F07A20-3C1A0D43&fcnt=3&  
action=readData
```

Response “dataRead”

For the request “readData” the WEB-Client application will respond with the action name “dataRead”.

In this response the WEB-Client application will send back 5 parameters:

- PHPSESSID** - is the session ID „ssid“ received in the request action “clientOk”.
- data1** - the WEB-Client application returns in “data1” the variables which were read from the Matrix-token. The variables are returned in an hexadecimal string with the format:
oooooooo-oooooooo-....-oooooooo (separator must be “-”)
- data2** - is the Serial-No. of the Matrix-dongle.
- action** - the action name in this client response is „dataRead“.
- status** - the status parameter is the return code in this client response can have following values:
 - >0 the number of variables which were read from the dongles memory
 - 0 no data read from the dongle
 - 1 Matrix-dongle not present or hardware error
 - 2 Matrix-dongle with different UserCode
 - 4 Matrix-dongle is locked by the Anti-Hacker Lock
 - 5 LPT/USB port cannot be acquired, because is already in use
 - 6 a fault occurred when the LPT/USB port was accessed

Response example:

```
http://www.test.com/logon.php?PHPSESSID=4f6de45a15a68059a5eb46
39fa fe1f06&data1=0A0B0C0D-1A1B1C1D-2A2B2C2D&data2=1234567890&
action=dataRead&status=3
```

Request “writeData”

This request is the action to be used to write data into the memory of the Matrix-token.

In this request the server script will have to pass 5 parameters:

- data1** - the server must send in „data1“ the encrypted UserCode of the Matrix-token. Encryption must be done on the server side with the same Key as the one stored in the Matrix-token. This buffer is an hexadecimal string with the format:
R1-UC-R2-UC (separator must be “-”)
Where ‘R1’ and ‘R2’ are random numbers and ‘UC’ is the UserCode of the token. The WEB-Client application decrypts the buffer “data1” and checks the accuracy of the UserCode.
- data2** - the server sends in “data2” the variables to be stored in the Matrix-token. The variables must be send in an hexadecimal string with the format:
oooooooo-oooooooo-....-oooooooo (separator must be “-”)
- fpos** - is the position (variable number) where to start writing in the token memory.
- fcnt** - is the number of variables to be write into the token memory. The maximum number of variables to be write at once is 79.
- action** - is the action name „writeData“.

Request example:

```
/?data1=6B2AB0D7-BC03A6F0-17F07A20-3C1A0D43&data2=0A0B0C0D-1A1B1C1D-2A2B2C2D&fpos=1&fcnt=3&action=writeData
```

Response “dataWrite”

For the request “writeData” the WEB-Client application will respond with the action name “dataWrite”.

In this response the WEB-Client application will send back 3 parameters:

- PHPSESSID** - is the session ID “ssid” received in the request action “clientOk”.
- action** - the action name in this response is “dataWrite”.
- status** - the status parameter is the return code in this client response can have following values:
 - >0 the number of variables which were stored into the dongles memory
 - 0 no data was stored into the dongle
 - 1 Matrix-dongle not present or hardware error
 - 2 Matrix-dongle with different UserCode
 - 4 Matrix-dongle is locked by the Anti-Hacker Lock
 - 5 LPT/USB port cannot be acquired, because is already in use
 - 6 a fault occurred when the LPT/USB port was accessed

Response example:

```
http://www.test.com/logon.php?PHPSESSID=4f6de45a15a68059a5eb4639fa_fe1f06&action=dataWrite&status=3
```


WWW.TDI-MATRIX.COM

