JGPShell commands

Global commands :

- **/ls** : displays the list of the readers connected to the card.

- **/connect :** connect to the card

  Usage :

  - /connect rdr_number
  - /connect "rdr_name"

- **/select** : select one reader and connect to the card, the authentication (init-update and ext-auth) are also done and the card manager is selected.

  Usage :

  - /select rdr_number
  - /select "rdrName"

- **/reinit** : re-iniailise the current session (useful, for example, after modifying the SQL session variables)

  .

- **/set :**

  Usage :

  - /set var_name "var_value"

- **/echo:**

  Usage :

  - /echo $var_name : displays the value of the variable var_name
  - /echo "Hello world !" : displays "Hello world !" ...

- **/exit** : close all connections and quit

- **/start :**

  Usage :

- /start [parameters] rdr_number : Enable R_thread mode for the reader rdr_number
- /start [parameters] rdr_name

Parameters :

- -p : the R_thread will be permanent, it means that it will not be closed after sending the cmds to the reader. The only way to close it is : /kill rdr_number

- **/end** : Stop the R_thread mode. This does not mean the thread will be closed.

- **/kill** : kill a running thread

Usage :

- /kill rdr_number

- **/threads :** Display running/stopped/finished threads

Usage :

- /threads : Displays running threads.
- /threads -h : Display the history (stopped or finished threads)

- **/run :** Execute shell file

Usage :

- /run “file_name”

Card commands :

- **init-update**
- **ext-auth**
- **authenticate** (equivalent to init-update and ext-auth)
- **exit :** close the connection with the card
- **send**

Usage :

- send “apdu”

- **upload** :

Usage :

- upload “packagePath”

- **install** :

Usage :

- install “packagePath”
- install |packageAID |appletAID
-

- **format :**

  Usage :

  - format

- **ls :** display the content of the card.

- **delete :**

  Usage :

  - delete “packagePath”
  - delete |AID

Session variables :

Variable permissions :

- Read : “1xx”
- Write : “x1x”
- Delete: “xx1”

Shell : (permissions=110)

- Log (0: no log, 1: log int log file, 2: log in stdout)
- Sql_user
- Sql_pass
- Sql_table
- Sql_server
- Keys
- Keys_index

Private :(permissions=100**)**

- SQL_CONNECTED : Is set to “1” if the sql connection success
- RDR_SELECTED : Is set to “1”, if one reader was chosen
- RDR_NAME : the current reader name
- RDR_NUMBER : the index of the current reader

- NB_READERS : the number of readers

public : (permissions=111**)**

Session example :

>/ls

```
[#]>/ls
1: OMNIKEY CardMan 3x21 0  Card inserted  ATR=15: 3BE600FF8131FE454A434F50333106


---- 1 reader(s) found !
```

> /connect 1

```
[#]>/connect 1
---Card into OMNIKEY CardMan 3x21 0 ready !---
OMNIKEY CardMan 3x21 0>
```

OMNIKEY CardMan 3x21 0> init-update
OMNIKEY CardMan 3x21 0> ext-auth
OMNIKEY CardMan 3x21 0> upload "c:/HelloWorld.cap"
OMNIKEY CardMan 3x21 0> install "c:/HelloWorld.cap"
OMNIKEY CardMan 3x21 0> delete "c:/HelloWorld.cap"
OMNIKEY CardMan 3x21 0> close

>/run test1.jgp

>/exit

Script Shell :

Example 1:

The following script connects to the first reader, does the authentication, displays the content of the card, tries to format it and finally displays again the content.

```
/connect 1
authenticate
ls
format
ls
```

Texte 1: test.jgp

Example 2:

The following script install “HelloWorld.cap” in all the connected readers.

The upload and install commands are sent in parallel to the reader. For each reader a thread is launched.

```
for x 1 to $NB_READERS
{
          /start -p $x
          init-update
          ext-auth
          upload « c:/HelloWorld.cap »
          install « c:/HelloWorld.cap »
          /end
}
```

Texte 2: test.jgp