

# How to use ATface 0.1

## 1 Content

### Inhaltsverzeichnis

1	Content.....	1
2	What's ATface for?.....	2
3	Folders and Files in ATface.....	2
3.1	ATface.....	2
3.1.1	Conf.ini.....	3
3.1.1.1	Explanation of Values.....	4
3.1.1.1.1	Prg_Mode.....	4
3.1.1.1.2	UpdateV.....	4
3.1.1.1.3	ViewRequest.....	4
3.1.1.1.4	ViewBatch.....	4
3.1.1.1.5	ViewVT.....	4
3.1.1.1.6	Lang.....	4
3.1.1.1.7	OneUser.....	4
3.1.1.1.8	LogDir.....	5
3.1.1.1.9	ReqDir.....	5
3.1.1.1.10	LangDir.....	5
3.1.1.1.11	BatchDir.....	5
3.1.2	Log files.....	5
3.1.2.1	normal.log.....	5
3.1.2.2	error.log.....	5
3.1.2.3	control.log.....	5
3.2	Requests.....	5
3.2.1	destinations.ini.....	6
3.2.1.1	[Headers].....	6
3.2.1.2	[Areax].....	6
3.2.2	Solution Folder.....	9
3.2.2.1	request.ini.....	9
3.2.2.2	Request definition file.....	11
3.2.2.2.1	[RequestValue].....	13
3.2.2.2.2	[RequestVars].....	13
3.2.2.2.3	[VarsReq].....	13
3.2.2.2.4	[VarsRes].....	13
3.3	Lang.....	13
3.3.1	en.abt.....	14
3.3.2	en.hlp.....	14
3.3.3	en.lang.....	15
3.4	Batch.....	15
4	The user interface.....	15
4.1	Menu File.....	16

## 2 What's ATface for?

ATface allows you to send Request to Wirecard System.

This allows you to perform every type of transaction your WEB-shop sends automatically in a manually. Here you use the same XML-structured file your applications sends normally.

Additional it is possible to send transactions via Virtual Terminal. Thats means you just fill out a form.

It is also possible to send a bunch of transactions at once. To do this you just have to enter them in a spreadsheet (i.e. OpenOffice Calc or Excel) and save them as csv.File, separated by ";"

This means you can send transactions as

- XML-Request directly
- Batch file
- Via Virtual Terminal

You can use it to send it to all Destinations at Wirecard, where only one communication per transactions is necessary.

At the moment there are samples for following services in distribution available

- Credit Card
- EFT
- Risk
- CLICK2PAY
- SCP
- Payout

You can create your own requests or modify existing to fit to your needs by editing the apropiate files

## 3 Folders and Files in ATface

Please note, that samples are from German OS. So some system-folder names will have different names.

### 3.1 ATface

You need a home folder. In this folder the conf.ini file is locate, and in standard settings the logfiles will be written.

So that the program will find it settings this folder as to be named as ATface and be direct subfolder of your home directory.

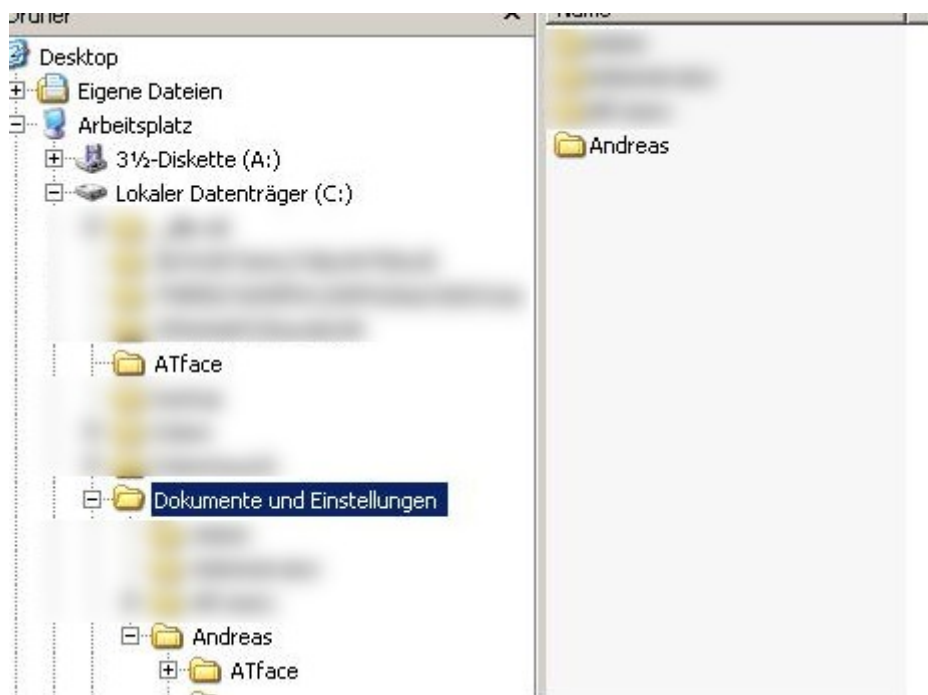
In the following sample the user logged on is andreas.

For **Windows** it is located in

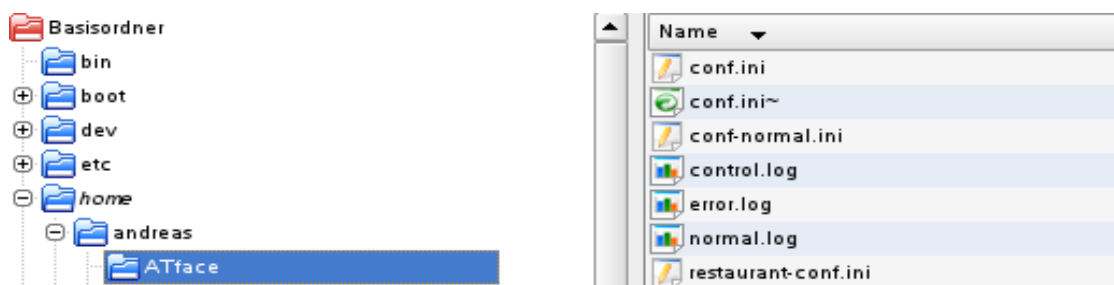
C:\Dokumente und Einstellungen\andreas

in english it should be

c:\documents and settings\andreas



For **Linux** it is located in  
/home/andreas



Please note, that in Linux all names are casesensitiv. That means lang is different to Lang.

### 3.1.1 Conf.ini

In order that your computer finds all necessary data you have to adjust the file conf.ini to your folder structure

Config file from my computer

---

```
#Config file fuer ATface
#Mon Apr 27 19:12:38 CEST 2009
Prg_Mode=1
UpdateV=4
ViewRequest=true
ViewBatch=true
ViewVT=true
Lang=de
OneUser=false
```

LogDir=  
ReqDir=/X\_w1/Requests  
LangDir=/X\_w1/lang  
BatchDir=/X\_w1/batch

### **3.1.1.1 Explanation of Values**

#### **3.1.1.1.1 Prg\_Mode**

0=DebugMode  
1= RunMode

#### **3.1.1.1.2 UpdateV**

Tells which update Version is installed. If Version in conf.ini is lower than the one coded in ATface.jar, program will do some changes in your conf.ini to adapt it new program version.

#### **3.1.1.1.3 ViewRequest**

Defines, wether you can see tab "run Request" in program

#### **3.1.1.1.4 ViewBatch**

Defines, wether you can see tab "run CSV" in program

#### **3.1.1.1.5 ViewVT**

Defines, wether you can see tab "Virtual Terminal" in program

#### **3.1.1.1.6 Lang**

Defines in which language program is run. At the moment de and en are available. Other languages can be created easily by translating the files

en.abt  
en.hlp  
en.lang

and rename them to xx.\*, where xx is the two-letter-code for the language. What to consider is written in main documentation

#### **3.1.1.1.7 OneUser**

Defines wether it is possible to send different access data, or wether access data is taken from configuration file destinations.ini

true: not changeable  
false: changeable

#### **3.1.1.1.8 LogDir**

Defines where logfiles will be stored, empty sets folder to homedir/ATface

#### **3.1.1.1.9 ReqDir**

Defines from where files containing request are read

#### **3.1.1.1.10 LangDir**

Defines from where files, containing language text are read

#### **3.1.1.1.11 BatchDir**

Defines default Batch folder

About pathes

In Windows a path looks like

c:\ATface\batch

Please note, that each normal \ has to be replaced by double \\ for correct function.

In Linux a path looks like

/ATface/batch

### **3.1.2 Log files**

There are three kinds of logfiles

#### **3.1.2.1 normal.log**

Entries are written during program is running. This helps to analyse what happens.

If you have problems forward this to your administrator. This log is overwritten on every new program start

#### **3.1.2.2 error.log**

Entries are written only when an error has occurred.

If you have problems forward this to your administrator. This log is overwritten on every new program start

#### **3.1.2.3 control.log**

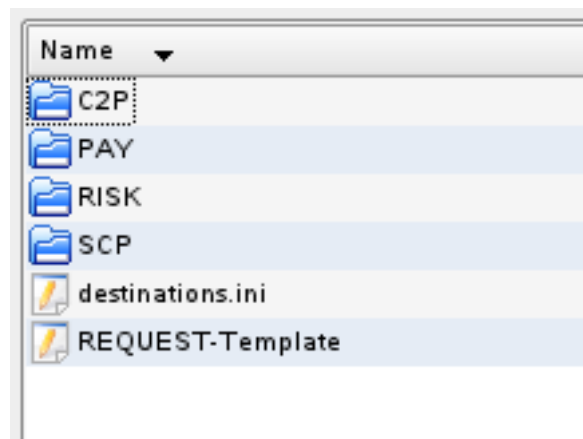
Entries are written everytime program is started or ended.

Here you (and everybody else, having access to this folder) can check when you have used this program. Logfile is never overwritten. So delete it once in a while.

### **3.2 Requests**

This folder contains the definition of the destinations and the requests in a subfolder structure. The foldername can't be changed. Location can be changed in conf.ini.

My Request folder look like this.



In the four folders C2P / PAY / RISK / SCP contains the requests for the four different solutions. They are defined in destinations.ini

Lets have a look at this file

### 3.2.1 destinations.ini

Here you have some sections

#### 3.2.1.1 [Headers]

Contains value numbers. In this case 8, as there are 8 areas following

#### 3.2.1.2 [Areax]

For each destination you have to define parameters

They are

- Name: Name displayed in dropdown box in program
- host: host part of URL where request is to be send
- port: port part of URL where request is to be send (443=https, 80=http)
- path: path part of URL where request is to be send  
the URL resulting of above three parameters in [Area1] of below sample is:  
https\\c3-test.wirecard.com\\secure\\ssl-gateway
- user: Username for authentication on payment system
- pass: Password for authentication on payment system
- BC\_SIG: BusinessCaseSignature, kind of merchant ID
- dir: Foldername, where requests are stored, You can see folder PAY in above screenshot and below sample.

You should have a subfolder for every solution. You can use same folder for test and livesystem, but you need different Area sections in destinations.ini, as the URLs and access data are different.

#### **Sample:**

[Headers]

*numbers=8*

*[Area1]*

*Name=Payment Testsystem*

*host=c3-test.wirecard.com*

*port=443*

*path=/secure/ssl-gateway*

*user=56501*

*pass=TestXAPTER*

*BC\_SIG=56501*

*dir=PAY*

*[Area2]*

*Name=Payment Livesystem*

*host=c3.wirecard.com*

*port=443*

*path=/secure/ssl-gateway*

*user=56500*

*pass=TestXAPTER*

*BC\_SIG=56500*

*dir=PAY*

*[Area3]*

*Name=Risk Testsystem*

*host=c3-test.wirecard.com*

*port=443*

*path=/risk/verify*

*user=000000315B095834*

*pass=WVxeNLqd8*

*BC\_SIG=000000315B09578C*

*dir=RISK*

*[Area4]*

*Name=Risk Livesystem*

*host=c3.wirecard.com*

*port=443*

*path=/risk/verify*  
*user=56500*  
*pass=TestXAPTER*  
*BC\_SIG=56500*  
*dir=RISK*

*[Area5]*  
*Name=C2P Testsystem*  
*host=c3-test.wirecard.com*  
*port=443*  
*path=/click2pay/services/walletPayment*  
*user=*  
*pass=*  
*BC\_SIG=000000315B09578C*  
*dir=C2P*

*[Area6]*  
*Name=C2P Livesystem*  
*host=c3.wirecard.com*  
*port=443*  
*path=/click2pay/services/walletPayment*  
*user=*  
*pass=*  
*BC\_SIG=000000315E8B38DD*  
*dir=C2P*

*[Area7]*  
*Name=SCP Testsystem*  
*host=c3-test.wirecard.com*  
*port=443*  
*path=/issuer/client*  
*user=000000315E8B32AB*  
*pass=QYr0c64-V*  
*BC\_SIG=000000315E8B38DD*  
*dir=SCP*



[Area8]

Name=SCP Livesystem

host=c3.wirecard.com

port=443

path=/issuer/client

user=000000315E8C361F

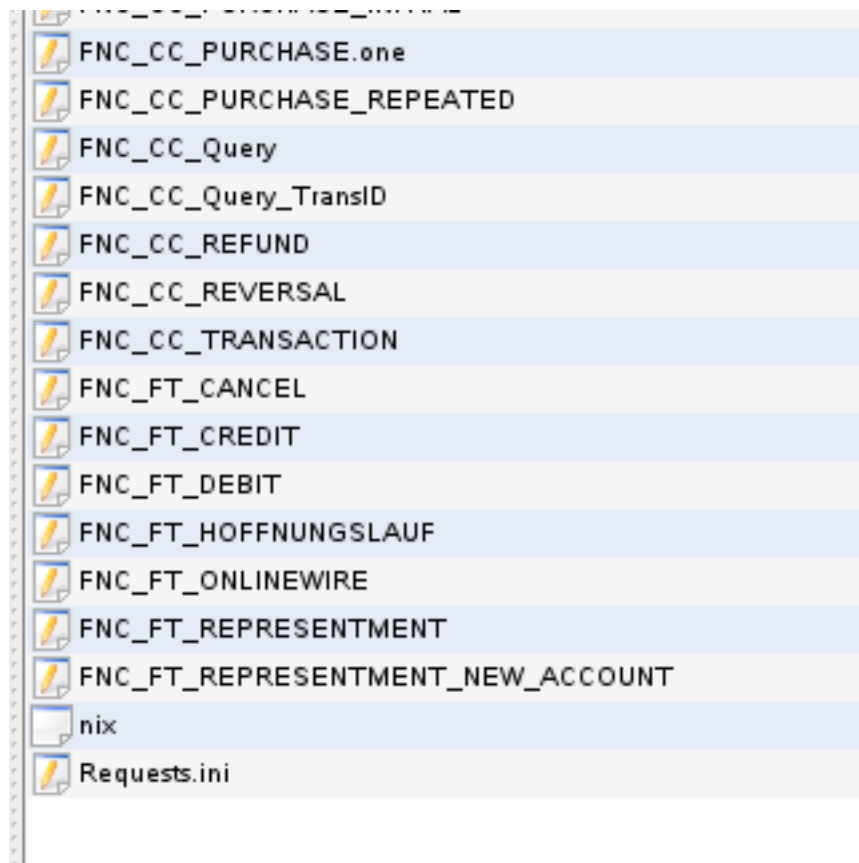
pass=nl0N-6rqM

BC\_SIG=000000315E8B38DD

dir=SCP

### 3.2.2 Solution Folder

Let's have a look of one of these Folder. To continue sample from above, we'll look at folder PAY.



As you can see I've cut of the beginning as other filenames does not explain any more. nix, is an empty file, just to create separators later.

Files FNC\* contains single Requests.

The most important file in this folder is Requests.ini

#### 3.2.2.1 request.ini

Lets have a look at this file.

Sample:

```
numbers=29
001 MHL FNC_FT_HOFFNUNGSLAUF
002 ----- nix
003 EFT-DEBIT FNC_FT_DEBIT
004 EFT-CREDIT FNC_FT_CREDIT
005 EFT-Cancel FNC_FT_CANCEL
006 EFT-REPRESENTMENT FNC_FT_REPRESENTMENT
007 EFT-REPRESENTMENT_New_Account
FNC_FT_REPRESENTMENT_NEW_ACCOUNT
008 ----- nix
009 Authorization FNC_CC_AUTHORIZATION
010 Bookback FNC_CC_BOOKBACK
011 Bookback_on_expired_card FNC_CC_BOOKBACK_EXPIRED
012 BookPreAuth FNC_CC_BOOKPREAUTH
013 Capture FNC_CC_CAPTURE
014 Capture_Notification FNC_CC_CAPTURE_NOTIFICATION
015 Enrollment FNC_CC_ENROLLMENT
016 Notification FNC_CC_NOTIFICATION
017 OCT_Bookback FNC_CC_OCT
018 OCT_Bookback_NonReferenced FNC_CC_OCT_NON_REF
019 Pre_Authorization FNC_CC_PRE_AUTHORIZATION
020 Purchase FNC_CC_PURCHASE
021 Purchase_initial FNC_CC_PURCHASE_INITIAL
022 Purchase_repeated FNC_CC_PURCHASE_REPEATED
023 Query FNC_CC_Query
024 Query_TransID FNC_CC_Query_TransID
025 Refund FNC_CC_REFUND
026 Reversal FNC_CC_REVERSAL
027 Transaction FNC_CC_TRANSACTION
008 ----- nix
029 GIROPAY FNC_FT_ONLINEWIRE
```

In the first line numbers defines how many Requests are listed in file, in this case 29  
The following lines have allways this structure:  
three-digit number-space-displayname-space-filename  
NO ADDITIONAL SPACES ARE ALLOWED  
The numbers have no influence now (maybe later)

Displayname is the name you'll see in dropdown box in program.

Filename is the name of the file containing request details. See also above screenshot.

### 3.2.2.2 Request definition file

As a sample I take FN\_CC\_CAPTURE. The function and correct structure of the different request will be explained by your payment provider. This sample works for Capture at Wirecard. GUWID is here the transaction ID.

#### **Sample:**

[RequestValue]

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="wirecard.xsd">
```

```
<W_REQUEST>
```

```
<W_JOB>
```

```
<JobID>Wirecard Test</JobID>
```

```
<BusinessCaseSignature>56501</BusinessCaseSignature>
```

```
<FNC_CC_CAPTURE_AUTHORIZATION>
```

```
<FunctionID>Wirecard Test</FunctionID>
```

```
<CC_TRANSACTION>
```

```
<TransactionID>Wirecard Test</TransactionID>
```

```
<GuWID>C813752123055634356132</GuWID>
```

```
<AuthorizationCode>481846</AuthorizationCode>
```

```
<Amount>100</Amount>
```

```
<Currency>EUR</Currency>
```

```
<CountryCode>DE</CountryCode>
```

```
</CC_TRANSACTION>
```

```
</FNC_CC_CAPTURE_AUTHORIZATION>
```

```
</W_JOB>
```

```
</W_REQUEST>
```

```
</WIRECARD_BXML>
```

[RequestVars]

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="wirecard.xsd">
```

```
<W_REQUEST>
```

```
<W_JOB>
```

```
<JobID>JOB_ID</JobID>
```

```
<BusinessCaseSignature>BC_SIG</BusinessCaseSignature>
<FNC_CC_CAPTURE_AUTHORIZATION>
  <FunctionID>FUNC_ID</FunctionID>
  <CC_TRANSACTION>
    <TransactionID>TRANS_ID</TransactionID>
    <GuWID>REFGUWID</GuWID>
    <AuthorizationCode>AUTHCODE</AuthorizationCode>
    <Amount>AMOUNT</Amount>
    <Currency>CURRENCY</Currency>
    <CountryCode>COUNTRY_CODE</CountryCode>
  </CC_TRANSACTION>
</FNC_CC_CAPTURE_AUTHORIZATION>
</W_JOB>
</W_REQUEST>
</WIRECARD_BXML>
```

[VarsReq]

httpsuser

httpspw

JOB\_ID

BC\_SIG

FUNC\_ID

TRANS\_ID

REFGUWID

AUTHCODE

AMOUNT

CURRENCY

COUNTRY\_CODE

[VarsRes]

<GuWID>

<AuthorizationCode>

<Info>

<StatusType>

<FunctionResult>

```

<TimeStamp>
<Type>
<Number>
<Message>
<Advice>

```

You have 4 sections here:

#### 3.2.2.2.1 [RequestValue]

Request filled with values. This request is displayed in Tab "run Request" when you select it in drop down. If you send this request it should work properly

#### 3.2.2.2.2 [RequestVars]

This is the Request from above, but Values are replaced by variable names. They can be defined by you. They are displayed in

- a) First line of batch-csv-file
- b) Virtual Terminal Tab

Restrictions are

- no spaces
  - name must not be found in Request anywhere else
- in sample Amount is not allowed, but AMOUNT or Amount\_ is

in above sample i.e.

```
<BusinessCaseSignature>56500</BusinessCaseSignature>
```

switch to

```
<BusinessCaseSignature>BC_SIG</BusinessCaseSignature>
```

#### 3.2.2.2.3 [VarsReq]

List of all Variable names used in above request

These list is used to

- a) read values from batch-csv-file
- b) create input fields for Tab Virtual Terminal

#### 3.2.2.2.4 [VarsRes]

These are values you want to be displayed in

- a) result-cvs-batch-file
- b) result fields in Tab Virtual Terminal.

If result line you are interested in is

```
<GuWID>C812345678901234567<GuWID>
```

and you put in

<GuWID> in this section, C812345678901234567 will be displayed.

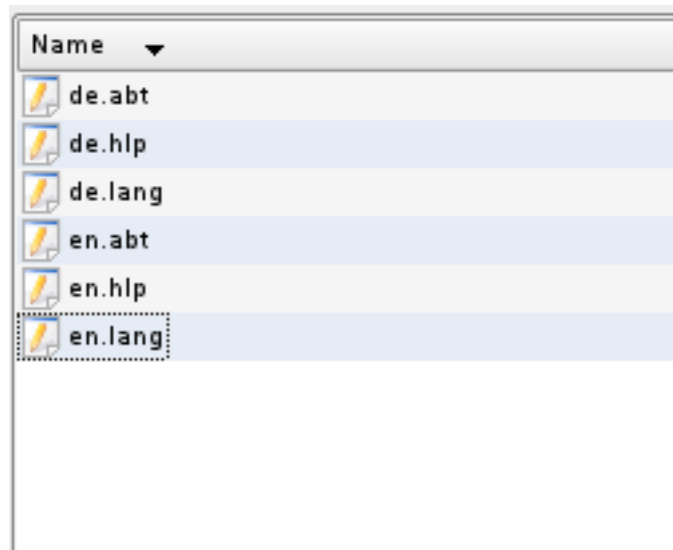
### 3.3 Lang

This folder contains the files with all texts used in the program. You can create your own language Version by copying the files to new name and translate content. The foldername

can't be changed. Location can be changed in conf.ini.

It would be great if you would offer translated text to project, so it can be distributed

This is the content of my language folder for english and german text



We'll have a look at each file for english text

### 3.3.1 en.abt

This text is displayed in about menue

```
ATface %1
*****
```

```
Build with Java 1.6.0
```

```
used libraraies:
```

```
atTools
```

```
JRE System Library [java-1.6.0-sun-1.6.0.u7]
```

If you translate text leave %1 in. Here the version-number is automatically inserted

### 3.3.2 en.hlp

Contains a short version of this manual without pictures.

If you translate the text, nothing to be cared of.

### 3.3.3 en.lang

This file contains the text, which is displayed in the program itself.

```
[LMenue]
Anzahl=6
{Menuepunkte}
001=File
002=Help
003=Settings
{Menue Datei}
004=Exit
{Menue Einstellungen}
005=select language
{Menue Hilfe}
006=About
```

In the above Sample the Text for the menu is defined

If you translate text only change text behind number= nowhere else. Otherwise program won't run properly. i.e. You can change 001=File to 001=Datei for German version.  
%1 and so on are variables, where i.e Filenames, which causes problem are inserted. So leave them in translation.

## 3.4 Batch

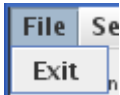
This folder defines the default directory for batch files. It is selected as first when you open file dialog for opening batches. The foldername can be changed. Location can be changed in conf.ini.

## 4 The user interface

First we'll have a look at the interface with all features enabled. As described in [3.1.1 Conf.ini](#) you can disable some features

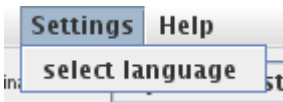
## 4.1 Menue bar

### 4.1.1 Menu File



This Menu contains only submenu Exit. Use it to exit program

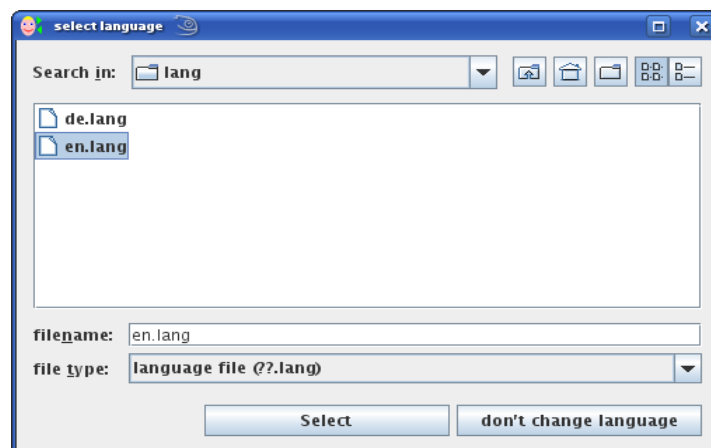
### 4.1.2 Menue Settings



Here you can select your language

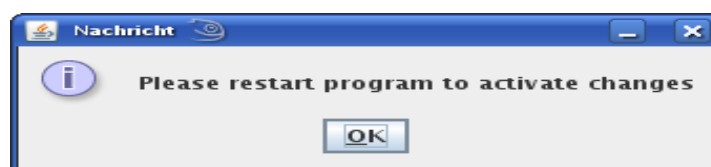
#### 4.1.2.1 How to change language

Click on "select language" And new window opens



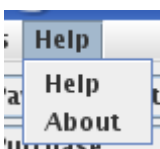
Then select new language and click on "Select" if you want to change language, or on "don't change language" if you decide to keep actual language.

If you have clicked on "Select" you'll see message box



Klick "OK" and restart program.

#### 4.1.2.2 Menu Help



This menue allows you to display help text or about text.

## 4.2 Header Area

This is, what I call the Header Area



Destination: <b>Payment Testsystem</b>	URL: <b>https://c3-test.wirecard.com/secure/ssl-gateway</b>	Https-User: <b>56501</b>	<b>ATface 1.3.1</b> <b>(C) Andreas Tritt 2009</b> <b>no warranty</b>
RequestType: <b>Purchase</b>	Password: <b>TestXAPTER</b>		

Here you have 5 fields, which are prefilled from file [3.2.1 destinations.ini](#)

Destination: To which system do you want to send requests?

URL: URL of selected destination. Field can be overwritten.

Https-User: User for authentication while sending request. Field can be overwritten

Password: Password for https-user. Field can be overwritten

Request Type: Depending on selected destination, appropriate [3.2.2 Solution Folder](#) folder is read and content of [3.2.2.1 request.ini](#) is displayed. Once you choose one request type all three folder described in [4.3 Tab Area](#) will be prefilled ore get additional fields for input and output

## 4.3 Tab Area

### 4.3.1 Tab run Request

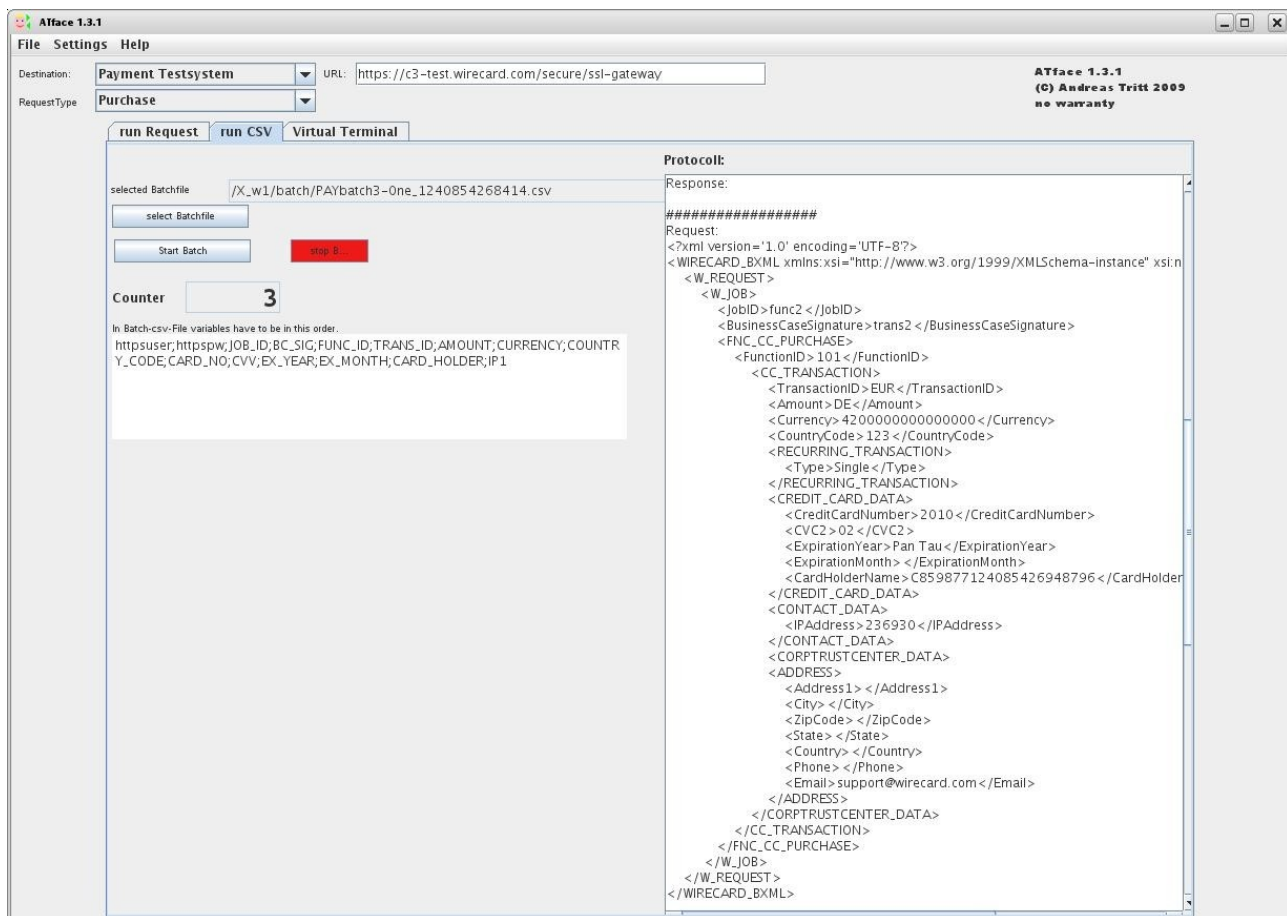
The screenshot shows the ATface 1.3.1 application window. At the top, there are input fields for Destination (Payment Testsystem), URL (https://c3-test.wirecard.com/secure/ssl-gateway), Https-User (56501), Password (TestXAPTER), and RequestType (Purchase). Below these fields are three tabs: 'run Request' (selected), 'run CSV', and 'Virtual Terminal'. The 'run Request' tab is divided into two main sections: 'Request' and 'Response'. The 'Request' section contains a large block of XML code for a purchase request. The 'Response' section shows the received XML response, which includes details like TransactionID, Amount, Currency, and CardHolderName. A green square icon is visible next to the 'Response' label, indicating a successful status. A 'Start' button is located at the top right of the 'Response' section.

For explanation I've choose request type Purchase. Left area named Request: was filled then automatically. You can edit this request directly, to send parameters you want.

Then press Start. After a short while in right area named Response, the response is displayed.

Depending on result values the square right of Response is either green for OK, or red for not OK.

### 4.3.2 Tab run CSV



For the same request type I've created a batch file named PAYbatch3.csv containing 3 payment requests.

Choose the file you want to process and Press "Start Batch". If you recognize problems during batch-run, you can press stop Batch, the red marked button. Please note, during program run only one of these two buttons is visible.

When you press "Start Batch" at first the file is renamed, In this sample from PAYbatch3.csv to PAYbatch3.inuse.

This is to prevent double processing by doubleclick.

Then batch is processed and results are written in PAYbatch3\_number.csv. The number is the system-time in milliseconds. So it should be unique. It is placed in same folder as request csv file.

When batch run is finished batch file is renamed again. In this sample from PAYbatch3.inuse to PAYbatch3.done

This sample request file looks like:

```
httpsuser;httpspw;JOB_ID;BC_SIG;FUNC_ID;TRANS_ID;AMOUNT;CURRENCY;COUNTRY_CODE;CARD_NO;CVV;EX_YEAR;EX_MONTH;CARD HOLDER;IP1
56500;TestXAPTER;job1;56500;func1;trans1;100;EUR;DE;4200000000000000;123;2010;01;Peter Pan;127.0.0.1
```

*56501;TestXAPTER;job2;56501;func2;trans2;101;EUR;DE;4200000000000000;123;2010;02;Pan Tau;*

*56501;TestXAPTER;job2;56501;func2;trans2;100005;EUR;DE;4200000000000000;123;2010;02;Zu Groß;*

This is a csv file in plain text

The result sample looks like:

*httpsuser;httpspw;JOB\_ID;BC\_SIG;FUNC\_ID;TRANS\_ID;AMOUNT;CURRENCY;COUNTRY\_CODE;CARD\_NO;CVV;EX\_YEAR;EX\_MONTH;CARD\_HOLDER;IP1;GuWID;AuthorizationCode;Info;StatusType;FunctionResult;TimeStamp;Type;Number;Message;Advice*

*56500;TestXAPTER;job1;56500;func1;trans1;100;EUR;DE;4200000000000000;123;2010;01;Peter Pan;127.0.0.1;C881273124133477045830;755287;THIS IS A DEMO TRANSACTION USING CREDIT CARD NUMBER 420000\*\*\*\*0000. NO REAL MONEY WILL BE TRANSFERED.;INFO;ACK;2009-05-03 09:12:50*

*56501;TestXAPTER;job2;56501;func2;trans2;101;EUR;DE;4200000000000000;123;2010;02;Pan Tau;;C881406124133477094340;888171;THIS IS A DEMO TRANSACTION USING CREDIT CARD NUMBER 420000\*\*\*\*0000. NO REAL MONEY WILL BE TRANSFERED.;INFO;PENDING;2009-05-03 09:12:51*

*56501;TestXAPTER;job2;56501;func2;trans2;100005;EUR;DE;4200000000000000;123;2010;02;Zu Groß;;C881638124133477178251;034035;THIS IS A DEMO TRANSACTION USING CREDIT CARD NUMBER 420000\*\*\*\*0000. NO REAL MONEY WILL BE TRANSFERED.;INFO;NOK;2009-05-03 09:12:51;REJECTED;05;Authorization Declined.;It is not possible to book the given amount from the credit account, e. g. limit is exceeded.*

In both files first line is heard, all other lines contain data.

### 1.1.1.1 How to create request csv file

On the run csv tab below the counter, there's white are with Headline "In Batch-csv-File variables have to be in this order."

Just copy them in Spreadsheet like OpenOffice-Calc or Excel in the first line. Make sure, by using function Data in columns to have each Item in separate column. Now you can fill in the data.

Once you are finished save file first in program specific format. Second, use save as to save file in csv (comma separated), but make sure

- Separator is ;
- No Text marker are activated " is not allowed in file

As discribed in [3.2.2.2.3 \[VarsReq\]](#) it is possible to define, which variables are needed. If in [3.1.1.1.7 OneUser](#) is set to true, columns httpsuser and httpspassword can be excluded from [3.2.2.2.3 \[VarsReq\]](#) . They will be replaced by values from [3.2.1 destinations.ini](#) .

### 1.1.1.2 How to read response csv file

Right ckclk on response csv-file and select open with. Select here your spreadsheet program. You may need to confirm import format or use function text in columns. You can define in [3.2.2.2.4 \[VarsRes\]](#) which variables should be displayed in response-csv file.

### 4.3.3 Tab Virtual Terminal

ATface 1.3.1

File Settings Help

Destination: **Payment Testsystem** URL: **https://c3-test.wirecard.com/secure/ssl-gateway** Https-User: **56501** Password: **TestXAPTER**

RequestType: **Purchase**

run Request run CSV Virtual Terminal

Fill in Request Data:

JOB\_ID: your job id  
 BC\_SIG: 56501  
 FUNC\_ID: your function id  
 TRANS\_ID: your transaction id  
 AMOUNT: 100  
 CURRENCY: EUR  
 COUNTRY\_CODE: DE  
 CARD\_NO: 4200000000000000  
 CVV: 123  
 EX\_YEAR: 2020  
 EX\_MONTH: 12  
 CARD\_HOLDER: Joe Doe  
 IP1: 22.23.24.25

Start

Response Information

GuWID: C866368124133391882821  
 Authorization Code: 057296  
 Info: THIS IS A DEMO TRANSACTION USING CREDIT CARD NUMBER 420000\*\*\*\*0000  
 StatusType: INFO  
 FunctionResult: PENDING  
 Time Stamp: 2009-05-03 08:58:38  
 Type:  
 Number:  
 Message:  
 Advice:

Request:

```
<?xml version='1.0' encoding='UTF-8'?>
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance" xsi:noNamespaceSchemaLocation="wirecard.xsd">
  <W_REQUEST>
    <W_JOB>
      <JobID>your job id</JobID>
      <BusinessCaseSignature>56501</BusinessCaseSignature>
      <FNC_CC_PURCHASE>
        <FunctionID>your function id</FunctionID>
        <CC_TRANSACTION>
          <TransactionID>your transaction id</TransactionID>
          <Amount>100</Amount>
          <Currency>EUR</Currency>
          <CountryCode>DE</CountryCode>
          <RECURRING_TRANSACTION>
            <Type>Single</Type>
          </RECURRING_TRANSACTION>
          <CREDIT_CARD_DATA>
            <CreditCardNumber>4200000000000000</CreditCardNumber>
            <CVC2>123</CVC2>
            <ExpirationYear>2020</ExpirationYear>
            <ExpirationMonth>12</ExpirationMonth>
            <CardHolderName>Joe Doe</CardHolderName>
          </CREDIT_CARD_DATA>
        </CC_TRANSACTION>
      </FNC_CC_PURCHASE>
    </W_JOB>
  </W_REQUEST>
</WIRECARD_BXML>
```

Here the left area is to be filled with data you want to send, to create the request. The fields again are defined in [3.2.2.2.3 \[VarsReq\]](#) .

Press start and after a short period you will see in right area the response the square will either show green for OK or read for NOT OK.

The fields show the details of the response. Again they are defined in [3.2.2.2.4 \[VarsRes\]](#) . So you are able to create special Request files [3.2.2.2 Request definition file](#) to show only necessary data to user and fill in static data in request. Therefore leave out variable names

As you can see in below sample, in Area [RequestVars] Tag <BusinessCaseSignature> is filled with value and therefore no longer listed in [VarsReq]. You'll find complete sample in file useCaseRestaurant.zip in Download area.

[RequestValue]

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="wirecard.xsd">
```

```
<W_REQUEST>
```

```
<W_JOB>
```

```
<JobID>job 2</JobID>
```

```
<BusinessCaseSignature>56501</BusinessCaseSignature>
<FNC_CC_PURCHASE>
  <FunctionID>WireCard Test</FunctionID>
  <CC_TRANSACTION>
    <TransactionID>2</TransactionID>
    <Amount>100</Amount>
    <Currency>EUR</Currency>
    <CountryCode>DE</CountryCode>
    <GuWID>C874449123917681576192</GuWID>
    <RECURRING_TRANSACTION>
      <Type>Repeated</Type>
    </RECURRING_TRANSACTION>
    <CONTACT_DATA>
      <IPAddress>127.0.0.1</IPAddress>
    </CONTACT_DATA>
    <CORPTRUSTCENTER_DATA>
      <ADDRESS>
        <Address1></Address1>
        <City></City>
        <ZipCode></ZipCode>
        <State></State>
        <Country></Country>
        <Phone></Phone>
        <Email>support@wirecard.com</Email>
      </ADDRESS>
    </CORPTRUSTCENTER_DATA>
  </CC_TRANSACTION>
</FNC_CC_PURCHASE>
</W_JOB>
</W_REQUEST>
</WIRECARD_BXML>
```

[RequestVars]

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<WIRECARD_BXML xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
```

```
xsi:noNamespaceSchemaLocation="wirecard.xsd">
  <W_REQUEST>
    <W_JOB>
      <JobID>JOB_ID</JobID>
      <BusinessCaseSignature>56501</BusinessCaseSignature>
      <FNC_CC_PURCHASE>
        <FunctionID>FUNC_ID</FunctionID>
        <CC_TRANSACTION>
          <TransactionID>TRANS_ID</TransactionID>
          <Amount>AMOUNT</Amount>
          <Currency>EUR</Currency>
          <CountryCode>DE</CountryCode>
          <GuWID>REF_GUWID</GuWID>
          <RECURRING_TRANSACTION>
            <Type>Repeated</Type>
          </RECURRING_TRANSACTION>
          <CORPTRUSTCENTER_DATA>
            <ADDRESS>
              <Address1></Address1>
              <City></City>
              <ZipCode></ZipCode>
              <State></State>
              <Country></Country>
              <Phone></Phone>
              <Email>support@wirecard.com</Email>
            </ADDRESS>
          </CORPTRUSTCENTER_DATA>
        </CC_TRANSACTION>
      </FNC_CC_PURCHASE>
    </W_JOB>
  </W_REQUEST>
</WIRECARD_BXML>
```

[VarsReq]

JOB\_ID

*REF\_GUWID*

*FUNC\_ID*

*TRANS\_ID*

*AMOUNT*

*[VarsRes]*

*<GuWID>*

*<AuthorizationCode>*

*<Info>*

*<StatusType>*

*<FunctionResult>*

*<TimeStamp>*

*<Type>*

*<Number>*

*<Message>*

*<Advice>*

In right bootom area you'll find request and response from communication with Wirecard.com. If there are any problems this information might help a lot