banksys
end-to-end transactions

Technologies & Products

Haachtsesteenweg 1442
1130 Brussels
Belgium

DEP Documentation

DEP RSA Key Loading Program User Manual

Version: 03.04 Classification: Restricted

Version Management Report

Version	Name(s)	Date	Comments
01.00	Paul Stiénon	15/02/2002	First Draft
01.01	Paul Stiénon	25/02/2002	Second draft
02.00	Paul Stiénon	19/09/2002	Verification Certificates/ new RSA Key
03.00	Filip Demaertelaere	27/02/2003	Documentation Platform Independent
03.01	Paul Stiénon	15/09/2003	Add of 3DSecure Pkcs10 Certificate Request
03.02	Sam Yala	18/09/2003	Add Annex D New screen captures Changes in section 4 and in section 6.
03.03	Paul STIENON	24/05/2005	More details in order of commands for self sign certificates.
03.04	Paul STIENON	03/04/2006	New disclaimer

CONFIDENTIALITY

The information in this document is confidential and shall not be disclosed to any third party in whole or in part without the prior written consent of Banksys S.A./N.V.

COPYRIGHT

LEGAL DISCLAIMER

While Banksys S.A./N.V. has made every attempt to ensure that the information contained in this document is correct, Banksys S.A./N.V. does not provide any legal or commercial warranty on the document that is described in this specification. The technology is thus provided "as is" without warranties of any kind, expressed or implied, included those of merchantability and fitness for a particular purpose. Banksys S.A./N.V. does not warrant or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, product or process disclosed.

To the fullest extent permitted under applicable law, neither Banksys S.A./N.V. nor its affiliates, directors, employees and agents shall be liable to any party for any damages that might result from the use of the technology as described in this document (including without limitation direct, indirect, incidental, special, consequential and punitive damages, lost profits).

JURISDICTION AND APPLICABLE LAW

These terms shall be governed by and construed in accordance with the laws of Belgium. You irrevocably consent to the jurisdiction of the courts located in Brussels for any action arising from or related to the use of this document.

1. TABLE OF CONTENTS

2. SCOPE OF THE DOCUMENT

This document describes the *DEP RSA Key Loading Program*. This PC program can be used to generate RSA keys, to import them from files, to export them into files and to put them into the key table of the DEP Crypto Module. The program also allows generating EMV and PKCS#10 public key certificate requests based on the certificate's requester Self-Signed Certificate (SSC). This program can also read and check EMV certificates files.

The document does explain neither the functionalities of the PKI library (most of the DEP functionality used by this program is based on the PKI library) nor the detailed specific data that must be given in order to get the SSC.

3. REFERENCES

This document contains references to other documents about the DEP. This paragraph gives a list of all the documents referred to.

- *DEP Host Interface Protocol*
- *DEP/NT DEP Handler Supervision Program User Manual*
- *DEP/Linux User Manual*

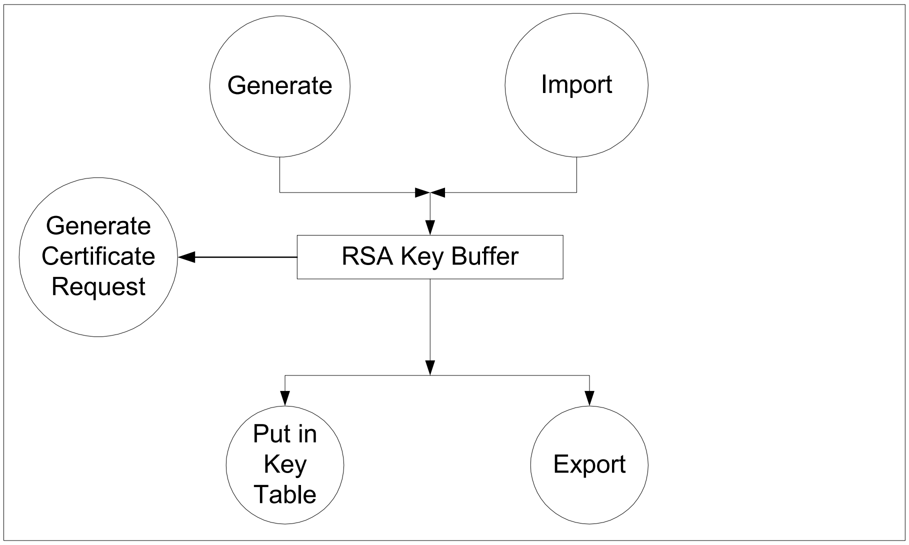
There are no references made to the following documents, but they could be useful to understand this document.

- *PKI Library for DEP - Reference DFS Manual*
- *DEP Introduction to DEP*
- *DEP General Architecture*
- *DEP Glossary*

4. PURPOSE OF DEP RSA KEY LOADING PROGRAM

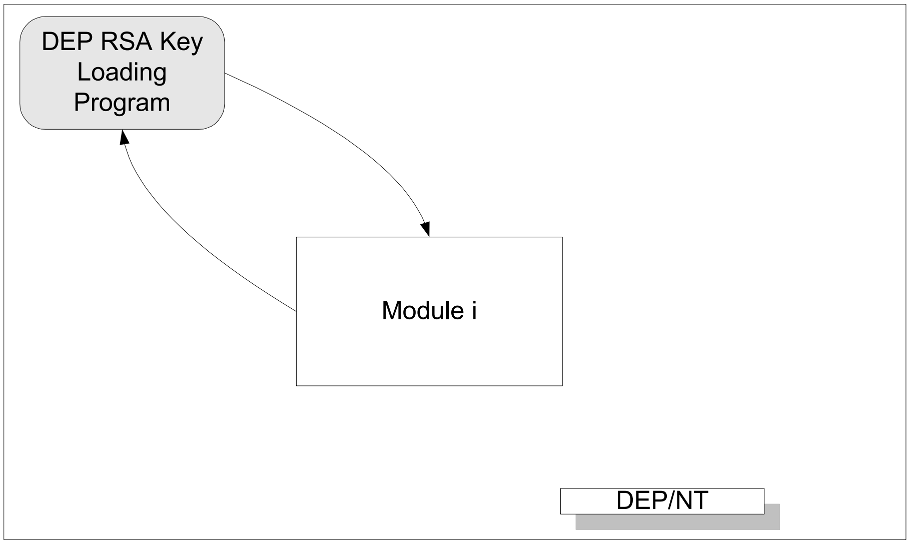
The main purpose of this program is to make the DEP generating an RSA key or importing such a key from an external file, and to use that key for the following purposes:

- Put the RSA key in the key table of the DEP crypto module
- Export the RSA key on files
- Generate EMV and PKCS#10 certificate requests. These certificates are based on self-signed public key certificates. In addition, the program provides some other facilities for managing public key certificates.

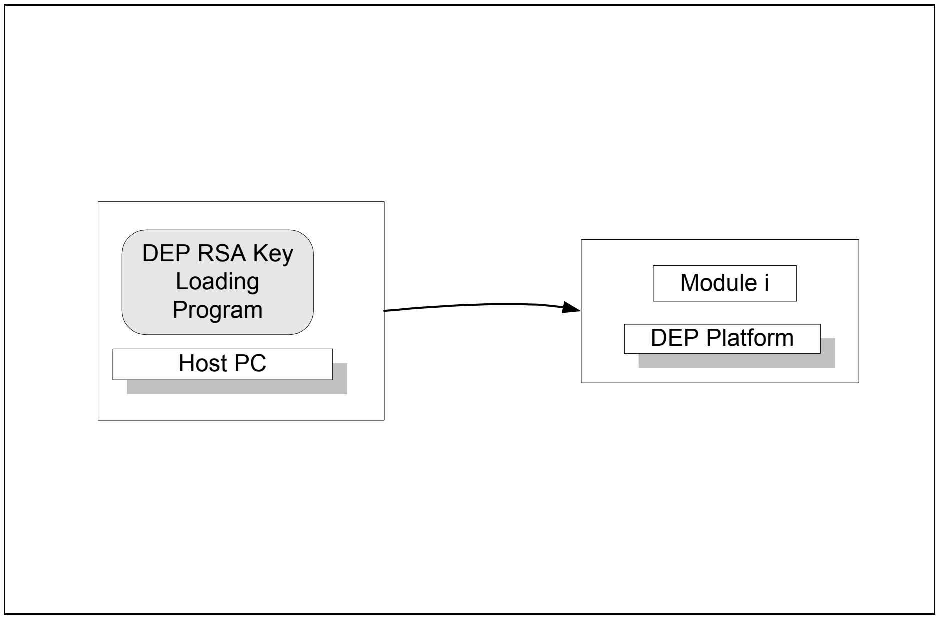


Purpose of the program

The program is intended to be used on a DEP/NT that can generate RSA keys (*configuration 1*) or on a PC that is connected to a DEP Platform that can generate RSA keys (*configuration 2*).



Configuration 1: the program is running on the DEP/NT itself.



Configuration 2: the program is running on a Pc that is connected to the DEP Platform.

5. GENERAL PARAMETERS

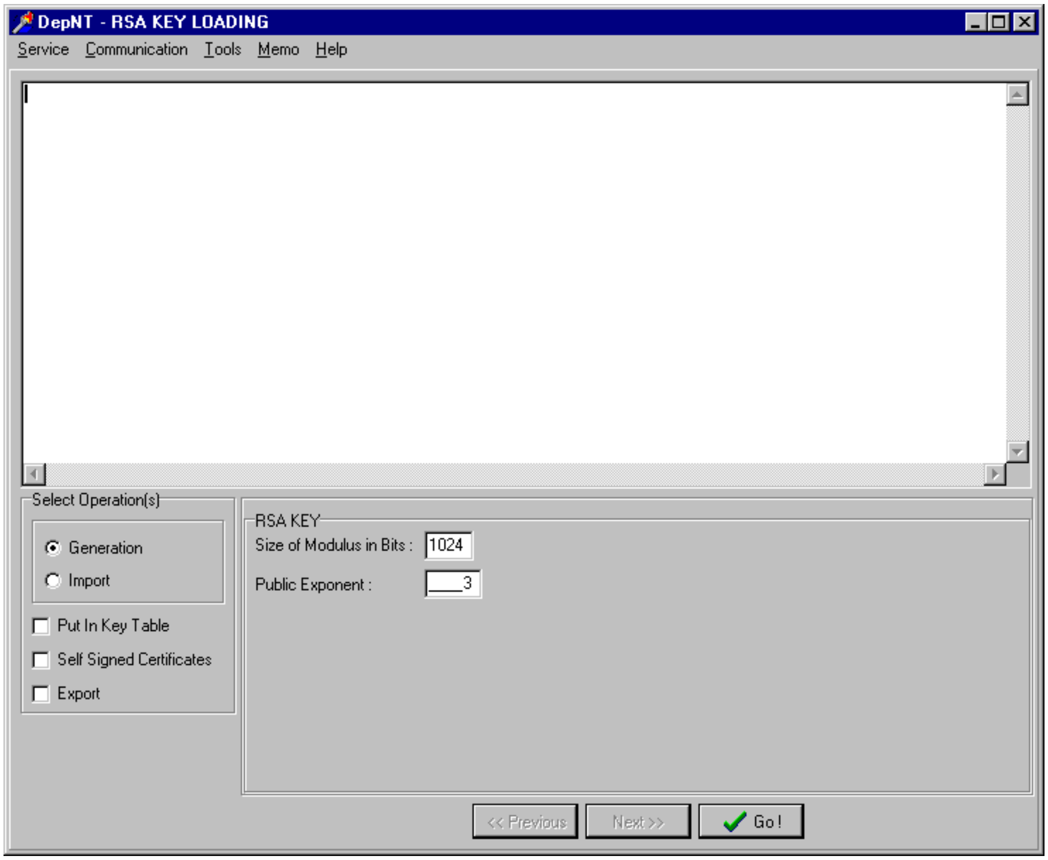
The installation procedure is reported in paragraph 10 on page 36.

5.1. START-UP

The *DEP RSA Key Loading Program* could be launched by executing:

C:\Depnt\Tools\RSA KEY LOADING\RSAKEYLOADING.exe

This is the default path. Possibly another path could be defined during the installation (paragraph 10 on page 36). Once the *DEP RSA Key Loading Program* is started, a dedicated **RSA KEY LOADING** window is opened.



● Generation
○ Import
□ Put In Key Table
□ Self Signed Certificates
□ Export

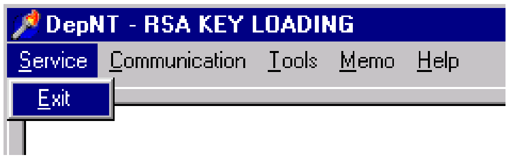
Size of Modulus in Bits : 1024
Public Exponent : 3

In this window, the user can find:

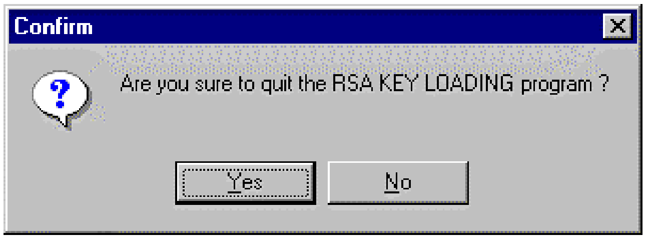
- a **memo** (blank part) which will log the operations and their results
- a menu at the top of the window, that allows to define communication, operate on the memo, have a look at the program version or to exit
- a **Select Operation(s)** panel which defines all the possible operations that can be done: generation or import, put in key table, get SSC, …
- another panel named **Parameters** that allows the definition of the needed parameters for the preceding operations and execute them (through the use of the **Previous**/**Next** and **Go!** Buttons)

5.2. SERVICE

Use the **Exit** in order to exit of the application.



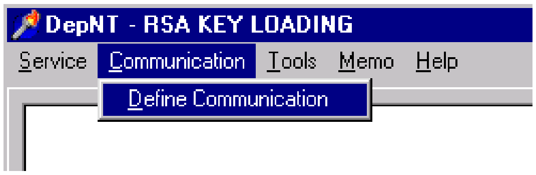
A *Confirmation* window is displayed, click YES to exit.

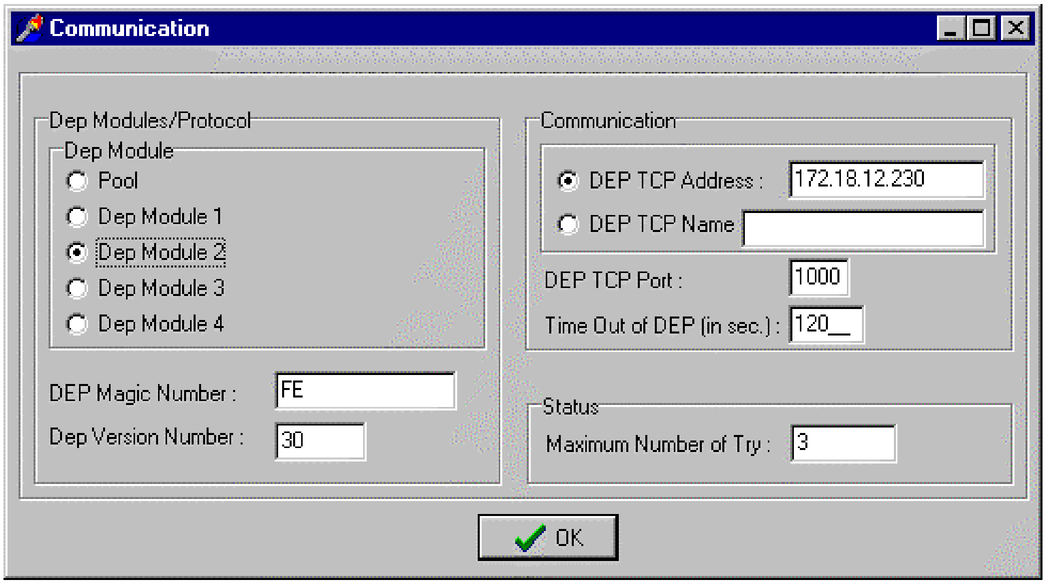


Are you sure to quit the RSA KEY LOADING program ?

5.3. Communication

Click on **Communication** and select **Define Communication** in the menu, to define all the communication properties that are necessary to communicate with the DEP Crypto Module(s) of the DEP Platform that will be connected to this program. A *Communication* window appears.





Dep Modules/Protocol

Dep Module

○ Pool
○ Dep Module 1
● Dep Module 2
○ Dep Module 3
○ Dep Module 4

DEP Magic Number : FE

Dep Version Number : 30

Communication

● DEP TCP Address : 172.18.12.230
○ DEP TCP Name :

DEP TCP Port : 1000

Time Out of DEP (in sec.) : 120

Status

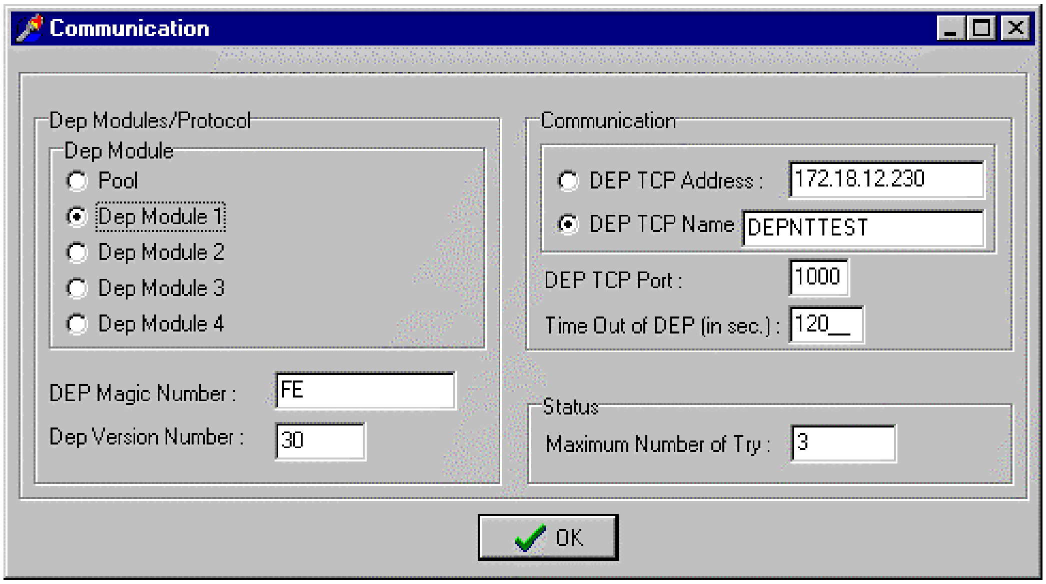
Maximum Number of Try : 3

5.3.1. DEP Modules/Protocol

The used protocol is the Enhanced DEP Protocol (for more information, refer to the *DEP Host Interface Protocol Manual*). Select **Pool** if the DEP Crypto Modules you want to use are working in Pool, otherwise you select a specific DEP Crypto Module. Define also the **DEP Magic Number** and the **DEP Version Number**. The standard values are shown in default.

5.3.2. Communication

The user must select either the **DEP TCP Address** or the **DEP TCP Name** of the DEP Platform that will communicate in TCP/IP with the *DEP RSA Key Loading Program*.



Communication

Dep Modules/Protocol

Dep Module

○ Pool
● Dep Module 1
○ Dep Module 2
○ Dep Module 3
○ Dep Module 4

DEP Magic Number : FE

Dep Version Number : 30

Communication

○ DEP TCP Address : 172.18.12.230
● DEP TCP Name DEPNTTEST

DEP TCP Port : 1000

Time Out of DEP (in sec.) : 120__

Status

Maximum Number of Try : 3

The **DEP TCP Port** and the **Time Out** of this DEP Platform must also be defined. Please note that this time out must be a little bigger than the one that is defined on the DEP Platform (see *DEP/NT DEP Handler Supervision Program User Manual* or *DEP/Linux User Manual* for more details).

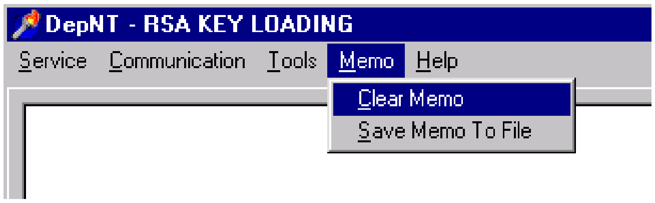
Be aware that the generation of RSA keys could take some time. Therefore it is required that the timeout on the DEP Platform is high enough (three minutes seems a good value).

5.3.3. Status

The value **Maximum Number Of Try** defines the maximum number of time an action with a bad response will be repeated (bad or closed TCP/IP connection, DEP Crypto Module offline, missing capability for a DEP Crypto Module…). When this value is reached, the program stops all the operations.

5.4. MEMO

Selecting **Memo** in the menu allows the user to realize actions on the memo (i.e. the large white band with/without text).

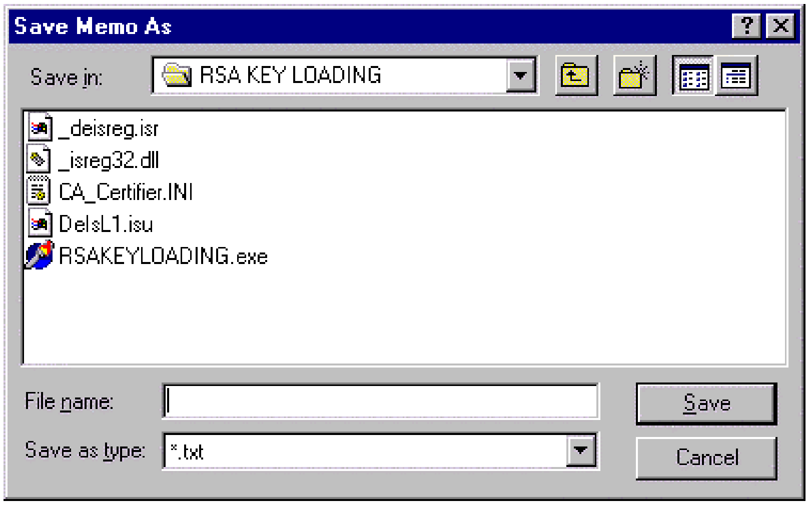


5.4.1. Clear Memo

The effect of this is the clearing of the text of the memo. Remember that this memo will contain information on the evolution of all the selected operations.

5.4.2. Save Memo To File

A *Save Memo As* window appears when the user clicks on the **Save Memo To File** item of the menu.

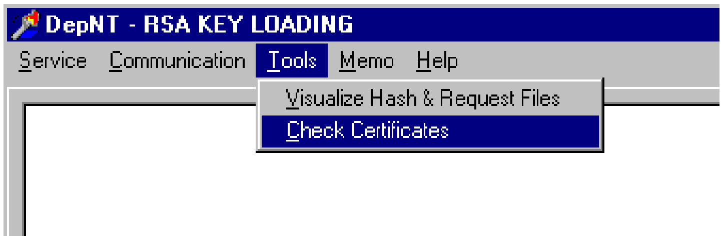


First specify the directory then the file where the text of the memo should be saved, then click on the SAVE button to confirm the selection. An example of file that contains the content of the memo is given in paragraph 9 on page 36.

5.5. TOOLS

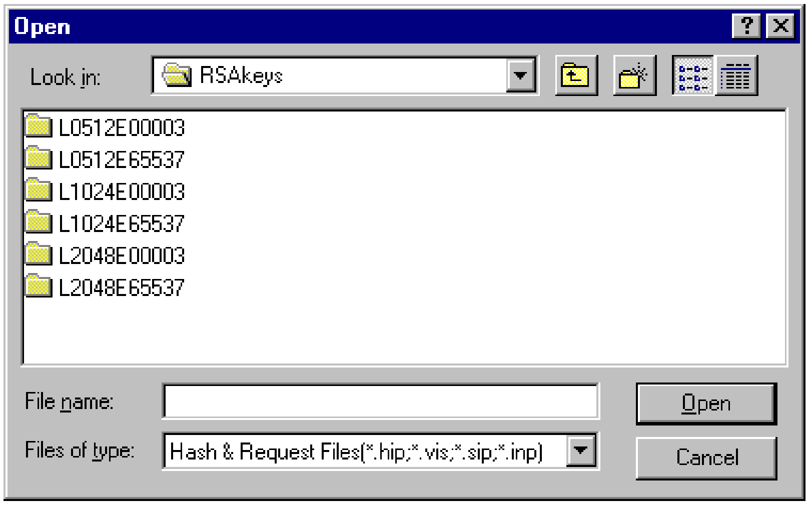
If the user clicks on the **Tools** item of the menu, he/she can select the operation on the certificate files he/she is going to realize:

- either reading an EMV certificate file, certificate hash file or certificate request file (without any check),
- either checking a list of EMV certificate files (Issuer request file, CA public key file, CA certificate file, CA hash file).



5.5.1. Visualize Hash & Request Files

Once the user selects the **Visualize Hash & Request Files** item of the **Tools** menu, an *Open* window appears, whose filter is automatically adapted from the CA_CERTIFIER.INI file in order to look for hash and request files.



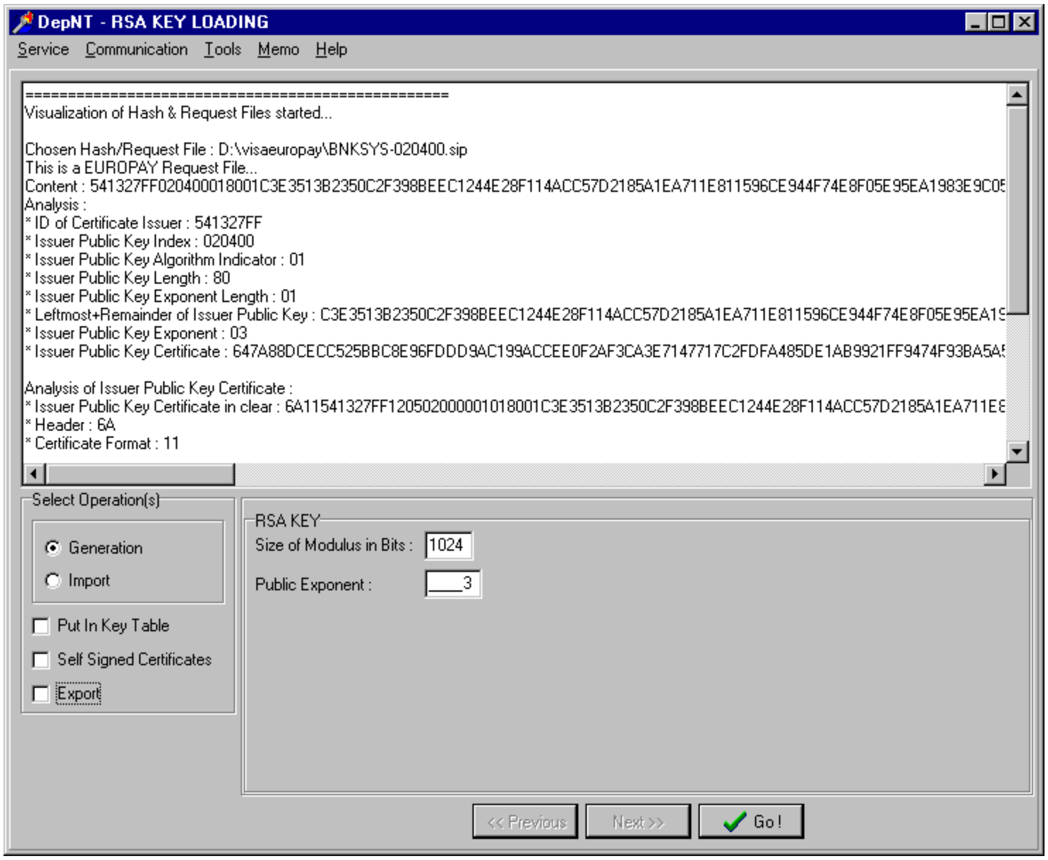
Look in: RSAkeys

L0512E00003
L0512E65537
L1024E00003
L1024E65537
L2048E00003
L2048E65537

File name:

Files of type: Hash & Request Files(*.hip;*.vis;*.sip;*.inp)

As soon as the user has chosen his/her file, the *Open* windows closes and the content of the selected file appears in the memo. In this phase, <u>no check is realized during the reading of the files.</u>



```
===================================================
Visualization of Hash & Request Files started...

Chosen Hash/Request File : D:\visaeuropay\BNKSYS-020400.sip
This is a EUROPAY Request File...
Content : 541327FF020400018001C3E3513B2350C2F398BEEC1244E28F114ACC57D2185A1EA711E811596CE944F74E8F05E95EA1983E9C05
Analysis :
* ID of Certificate Issuer : 541327FF
* Issuer Public Key Index : 020400
* Issuer Public Key Algorithm Indicator : 01
* Issuer Public Key Length : 80
* Issuer Public Key Exponent Length : 01
* Leftmost+Remainder of Issuer Public Key : C3E3513B2350C2F398BEEC1244E28F114ACC57D2185A1EA711E811596CE944F74E8F05E95EA19
* Issuer Public Key Exponent : 03
* Issuer Public Key Certificate : 647A88DCECC525BBC8E96FDDD9AC199ACCEE0F2AF3CA3E7147717C2FDFA485DE1AB9921FF9474F93BA5A5

Analysis of Issuer Public Key Certificate :
* Issuer Public Key Certificate in clear : 6A11541327FF120502000001018001C3E3513B2350C2F398BEEC1244E28F114ACC57D2185A1EA711E8
* Header : 6A
* Certificate Format : 11
```

Select Operation(s)

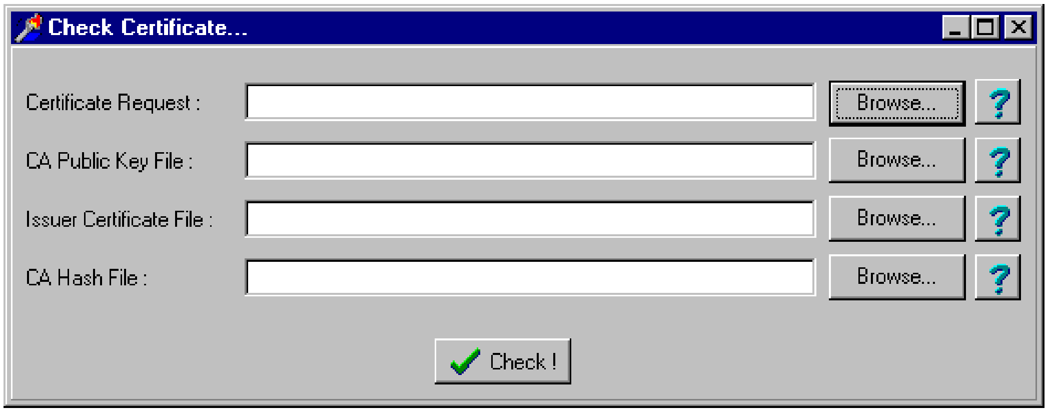
● Generation
○ Import

□ Put In Key Table
□ Self Signed Certificates
□ Export

RSA KEY
Size of Modulus in Bits : 1024
Public Exponent : ___3

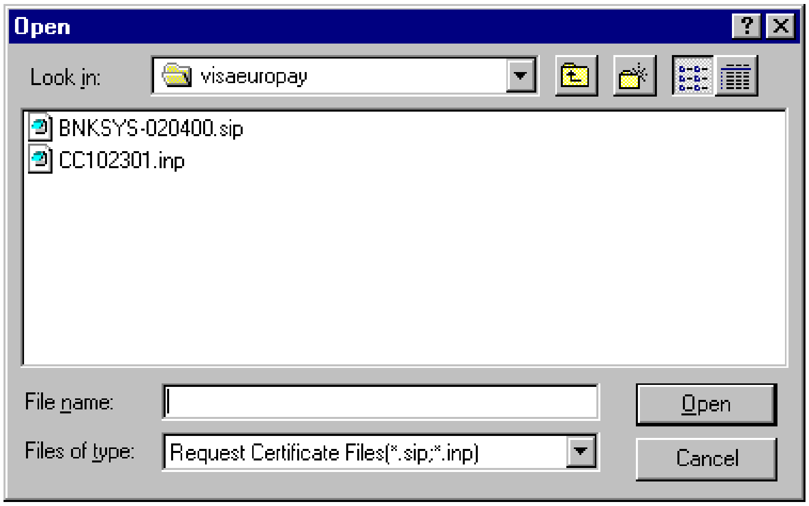
5.5.2. Check Certificates

With this operation, the user will not only read the content of the certificates files but can also realize a crosscheck between these files. The selection of the **Check Certificates** item of the **Tool** menu leads to the appearance of a **Check Certificate** window that allows the selection of several certificate files.



Certificate Request :
CA Public Key File :
Issuer Certificate File :
CA Hash File :

To choose one of the files, the user must click on one of the BROWSE button; this leads to the appearance of an *Open* window with a filter on the possible extensions of the corresponding files (coming from the CA_CERTIFIER.INI file).



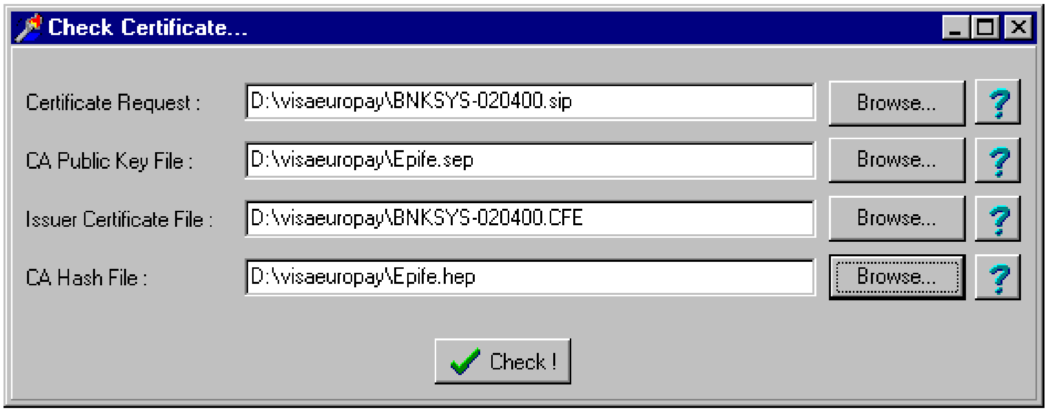
Look in: visaeuropay

BNKSYS-020400.sip
CC102301.inp

File name:

Files of type: Request Certificate Files(*.sip;*.inp)

The user has to select the files for the first three fields, the last one (CA hash file is optional) in order to allow the crosscheck between different files.



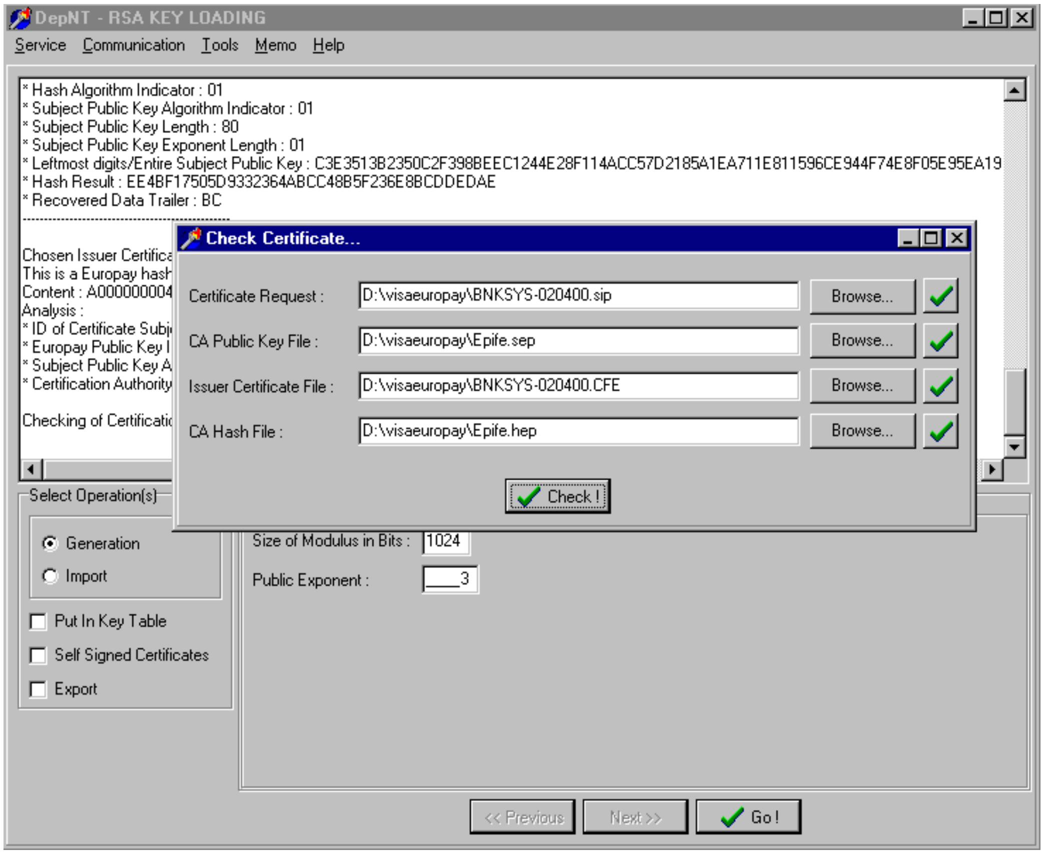
Certificate Request : D:\visaeuropay\BNKSYS-020400.sip

CA Public Key File : D:\visaeuropay\Epife.sep

Issuer Certificate File : D:\visaeuropay\BNKSYS-020400.CFE

CA Hash File : D:\visaeuropay\Epife.hep

The operation is launched while clicking on the CHECK button. The success or not of the check is shown with the buttons *V* or *X* at the right of the BROWSE buttons.



```
* Hash Algorithm Indicator : 01
* Subject Public Key Algorithm Indicator : 01
* Subject Public Key Length : 80
* Subject Public Key Exponent Length : 01
* Leftmost digits/Entire Subject Public Key : C3E3513B2350C2F398BEEC1244E28F114ACC57D2185A1EA711E811596CE944F74E8F05E95EA19
* Hash Result : EE4BF17505D9332364ABCC48B5F236E8BCDDEDAE
* Recovered Data Trailer : BC
--------------------------------------------------

Chosen Issuer Certifica
This is a Europay hash
Content : A000000004
Analysis :
* ID of Certificate Subj
* Europay Public Key I
* Subject Public Key A
* Certification Authority

Checking of Certificatio
```

Certificate Request : D:\visaeuropay\BNKSYS-020400.sip ✓

CA Public Key File : D:\visaeuropay\Epife.sep ✓

Issuer Certificate File : D:\visaeuropay\BNKSYS-020400.CFE ✓

CA Hash File : D:\visaeuropay\Epife.hep ✓

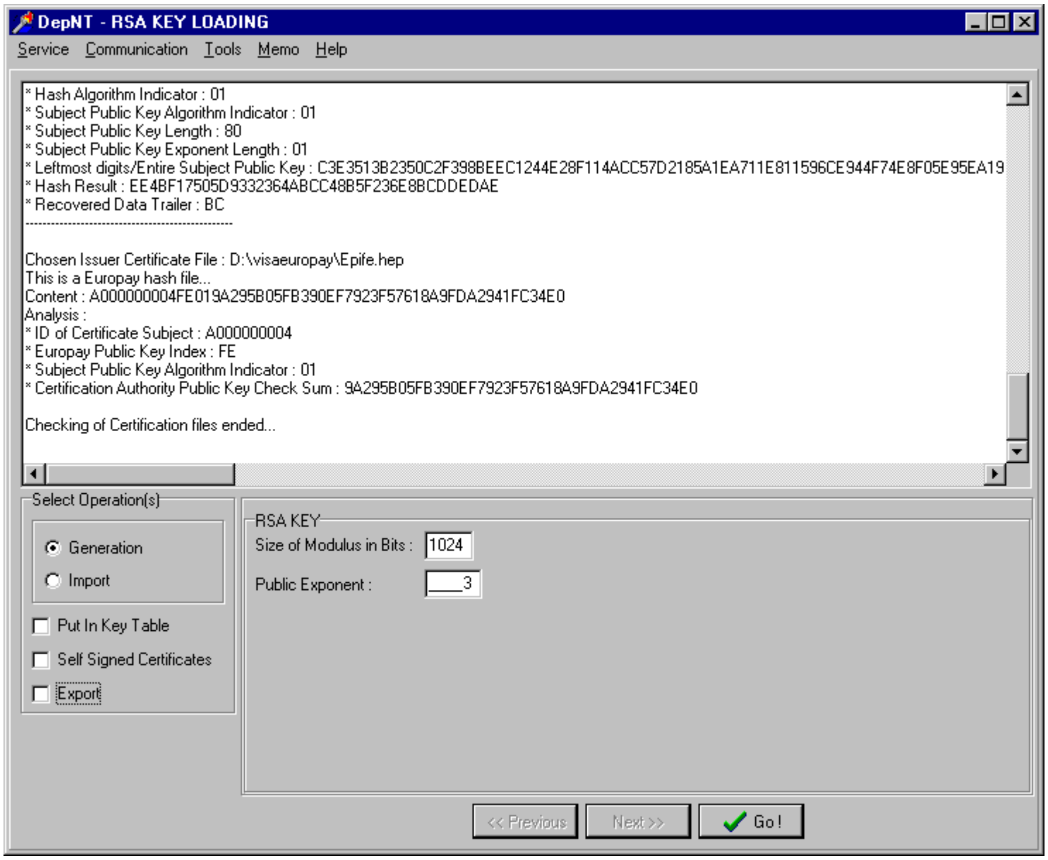
● Generation
○ Import

□ Put In Key Table
□ Self Signed Certificates
□ Export

Size of Modulus in Bits : 1024

Public Exponent : ___3

In order to have a look at the results, the user have to close the *Check Certificate* window.



```
* Hash Algorithm Indicator : 01
* Subject Public Key Algorithm Indicator : 01
* Subject Public Key Length : 80
* Subject Public Key Exponent Length : 01
* Leftmost digits/Entire Subject Public Key : C3E3513B2350C2F398BEEC1244E28F114ACC57D2185A1EA711E811596CE944F74E8F05E95EA19
* Hash Result : EE4BF17505D9332364ABCC48B5F236E8BCDDEDAE
* Recovered Data Trailer : BC
------------------------------------------------

Chosen Issuer Certificate File : D:\visaeuropay\Epife.hep
This is a Europay hash file...
Content : A000000004FE019A295B05FB390EF7923F57618A9FDA2941FC34E0
Analysis :
* ID of Certificate Subject : A000000004
* Europay Public Key Index : FE
* Subject Public Key Algorithm Indicator : 01
* Certification Authority Public Key Check Sum : 9A295B05FB390EF7923F57618A9FDA2941FC34E0

Checking of Certification files ended...
```

Select Operation(s)

● Generation
○ Import

□ Put In Key Table
□ Self Signed Certificates
□ Export

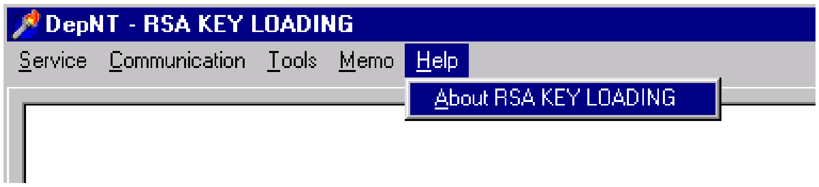
RSA KEY
Size of Modulus in Bits : 1024
Public Exponent : 3

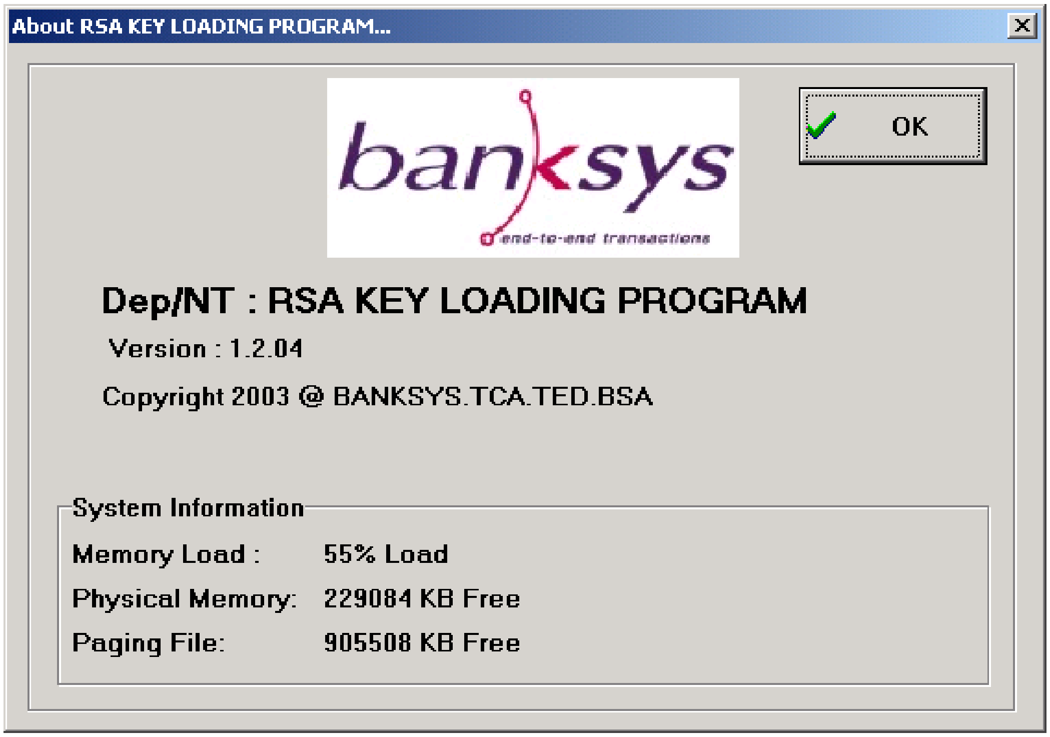
When an error is found, a warning is given and the process of checking certificate files is stopped. The user can save this information of the memo in a file using the **Save to File** item of the **Memo** menu.

Pay attention that this type of crosscheck is not available for all type of certificate requests.

5.6. HELP

When clicking on the **About RSA KEY LOADING** item of the **Help** in the menu, an *About* window appears with information on the *DEP RSA Key Loading Program* (version…) and on the system using it.





Dep/NT : RSA KEY LOADING PROGRAM

Version : 1.2.04

Copyright 2003 @ BANKSYS.TCA.TED.BSA

System Information

Memory Load :	55% Load
Physical Memory:	229084 KB Free
Paging File:	905508 KB Free

6. OPERATIONS

There are two basic operations:

1. *Generate RSA Key* (**Generation** radio button)
2. *Import RSA Key* (**Import** radio button)

Besides these basic operations, three associated operations (also called functions) are defined:

a. *Put RSA Key Into The Key Table* (**Put In Key Table** checkbox)

b. *Generate Certificate Requests* (**Self Signed Certificates** checkbox)

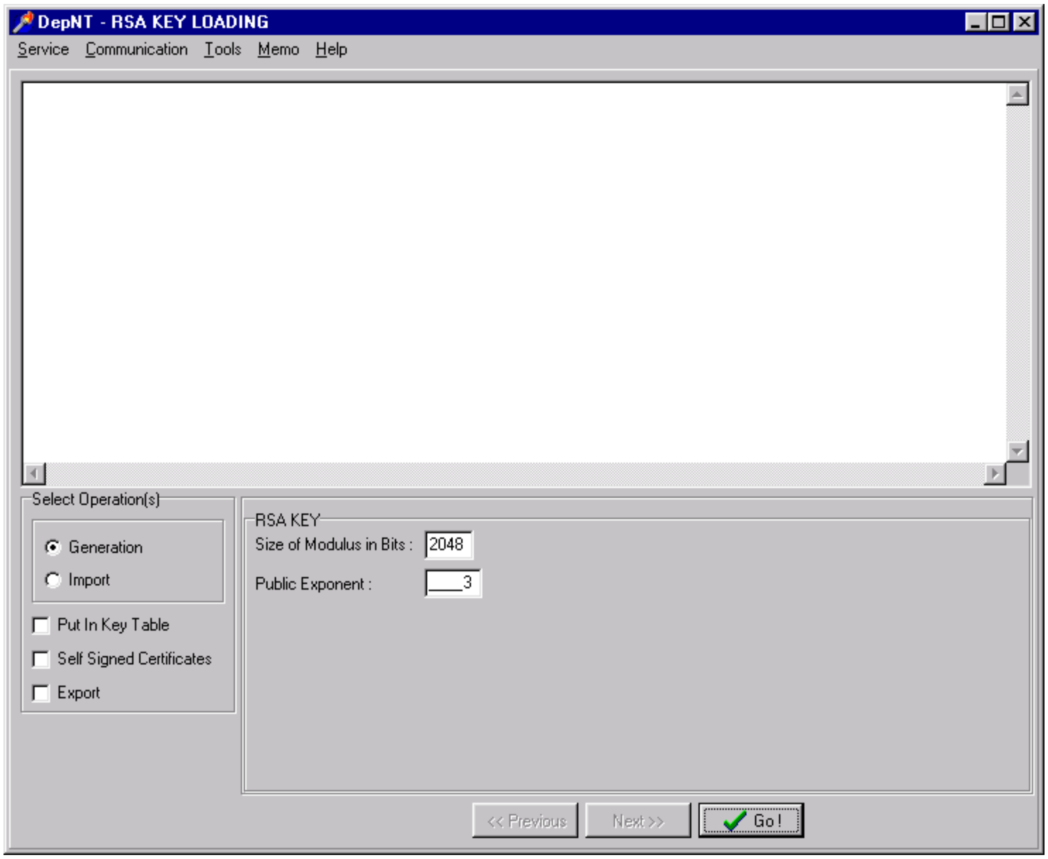
c. *Export RSA Key* (**Export** checkbox)

To obtain a useful functionality, one must combine a basic operation with at least one associated operation.

This is explained in the next sections.

6.1. GENERATE RSA KEY OPERATION

The user must select the radio button **Generation** in order to set the size of the modulus and the public exponent value needed for this operation.



Select Operation(s)
- ● Generation
- ○ Import
- □ Put In Key Table
- □ Self Signed Certificates
- □ Export

RSA KEY
Size of Modulus in Bits : 2048
Public Exponent : 3

6.1.1. Size of Modulus in Bits

Here, the user must give the length in bits of the modulus he/she wants to create (decimal value). For the modulus, the minimum size is 512 bits and the maximum size is 2048 bits.

6.1.2. Public Exponent

Here, the user must give the Public Exponent he/she wants to use for the generation of RSA keys (uneven decimal value between **3** and **65537**).

6.1.3. Running the "Generate RSA key" Operation

Generating an RSA key pair is completely useless unless it is combined with at least one of the following associated operations (functions)

a. "*Put RSA Key Into The Key Table*"

b. "*Generate certificate requests*". In this case, only the option "EMV (Visa or Europay) certificate request " is applicable.

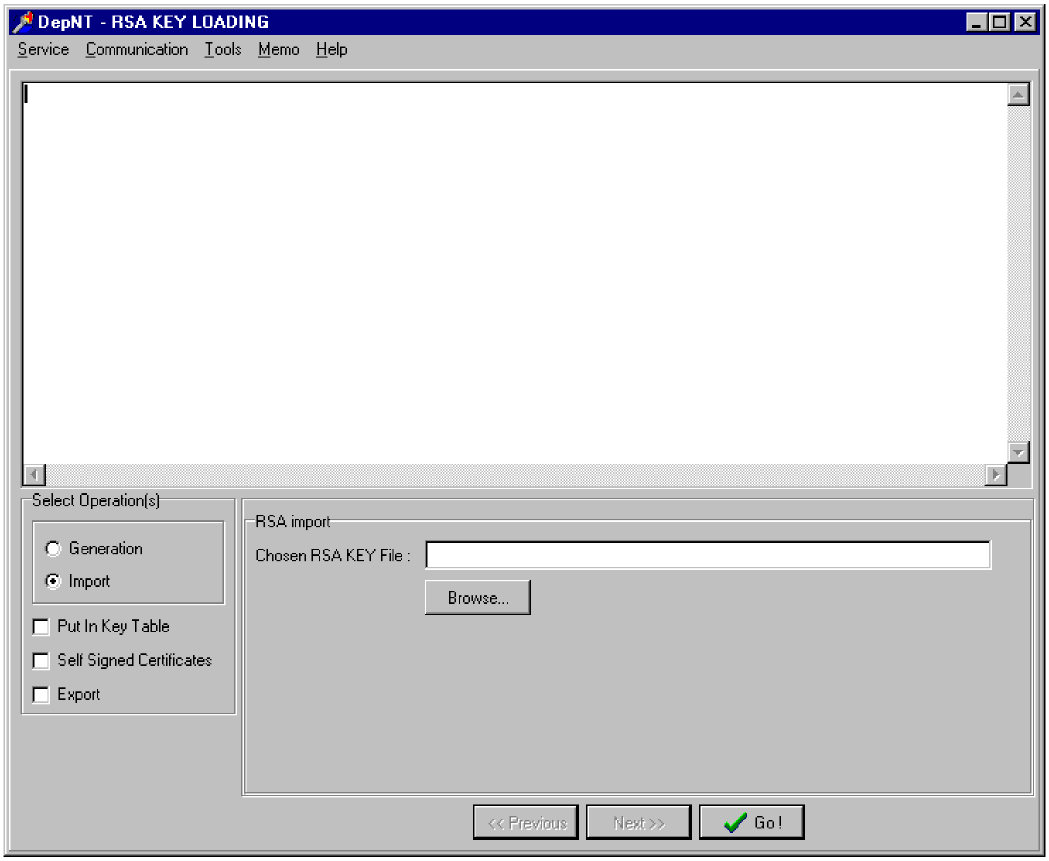
c. "*Export RSA key*".

These associated operations are explained in the next sections.

Once the operation and the associated operation(s) are selected and the corresponding fields are filled, the user can click on the GO! button.

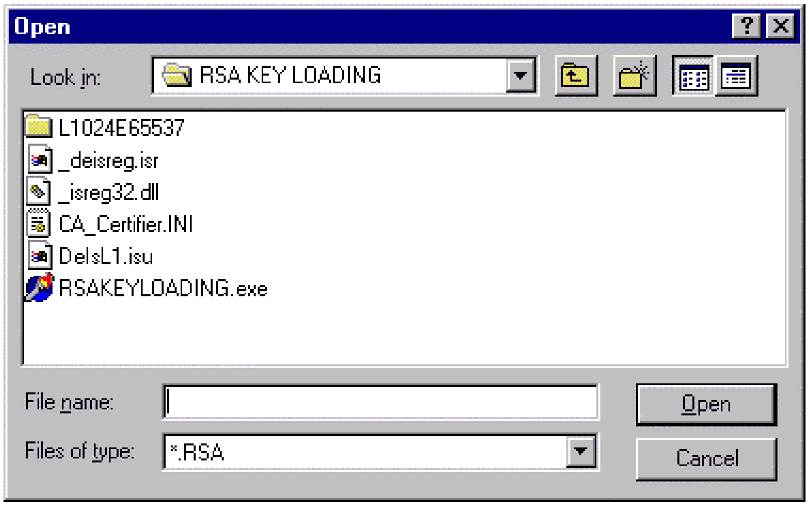
6.2. IMPORT RSA KEY OPERATION

The "*Import RSA Key*" operation allows the DEP getting the RSA key that is written in an .RSA file. The radio button **Import** allows selecting this operation.



6.2.1. Definition of the RSA Key File

The definition of the name of the RSA key file can be done either by writing it into the **Edit** box or by clicking on the button BROWSE. In this case, an *Open* window appears that allows the browsing along the files and directories. The user must confirm his/her selection by clicking on the OPEN button.


Look in: RSA KEY LOADING

File name:

Files of type: *.RSA

6.2.2. Running the RSA Key Import

Importing an RSA key pair is completely useless unless it is combined with at least one of the following associated operations

a. "*Put RSA Key Into The Key Table*"
b. "*Generate Certificate Request*".

These associated operations are explained in the next sections.

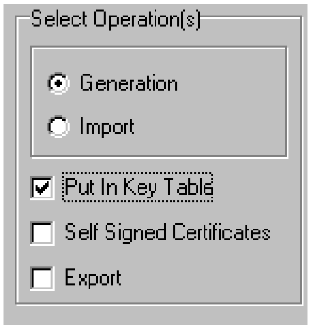
Once the operation and the associated operation(s) are selected and the corresponding fields are correctly filled, the user can click on the GO! button.

Information on the evolution of the import is given in the memo (name of the RSA key file, version and date of this file, clear modulus and public key, encrypted part).

Note that when an import has been done, the **Edit** box with the name of the imported file is erased.

6.3. PUT RSA KEY INTO THE KEY TABLE OPERATION

The activation of the "*Put RSA Key Into The Key Table*" operation is done as follows: the user must select the checkbox **Put In Key Table**. Immediately, the *Parameters* panel adjusts itself in order to present the data for this function.



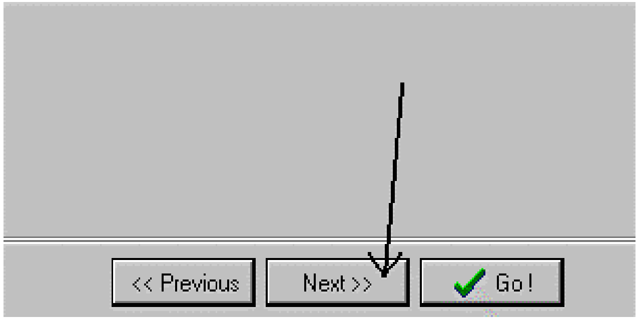
Select Operation(s)

● Generation
○ Import

☑ Put In Key Table
□ Self Signed Certificates
□ Export

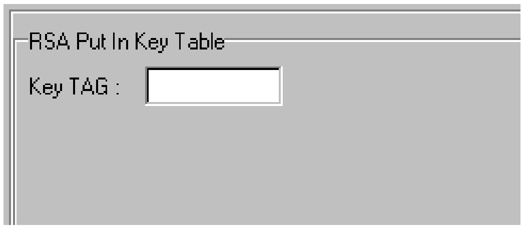
6.3.1. Definition of the Key Destination Tag

In order to properly define the data necessary for the different operations, the user must click on the PREVIOUS and NEXT buttons, which allow the move through all windows defining parameters.



In this case, a panel appears with the capability of defining the identification tag for the RSA key pair. This tag should correspond to an RSA registered key in the key table of the DEP Crypto Module.

Please note that an empty Tag means that this RSA key pair will not be put into the key table of the DEP Crypto Module.

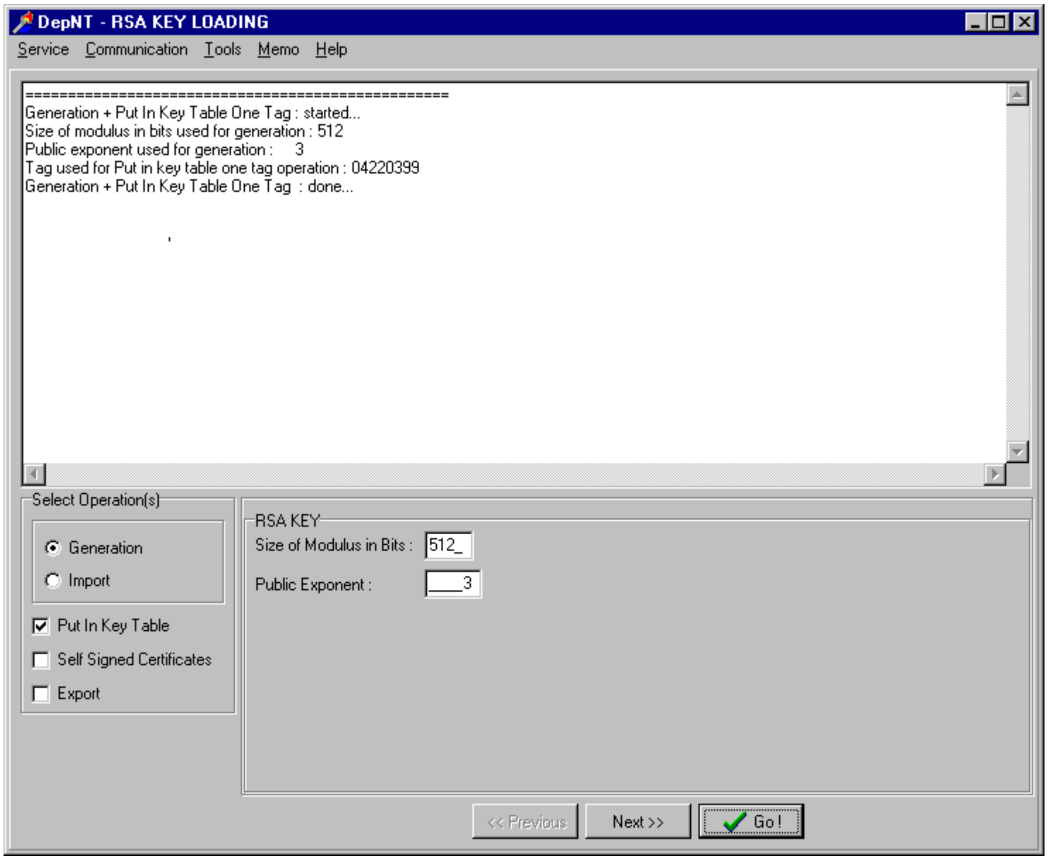


RSA Put In Key Table

Key TAG :

6.3.2. Running the *Put RSA Key Into The Key Table* Operation

As an associated operation, this operation must be combined with the *"Generate RSA Key"* operation (**generation** radio button selected) or with *"Import RSA Key"* operation (**import** radio button selected). By clicking on the PREVIOUS/NEXT buttons, the user can define/redefine the parameters necessary for the associated operation and the basic operation. Then the user can click on the GO! button in order to execute the selected operations (the basic operation and the associated operation). Again some information on the whole process is given in the memo.



```
==================================================
Generation + Put In Key Table One Tag : started...
Size of modulus in bits used for generation : 512
Public exponent used for generation :     3
Tag used for Put in key table one tag operation : 04220399
Generation + Put In Key Table One Tag  : done...
```

Select Operation(s)

● Generation
○ Import
☑ Put In Key Table
☐ Self Signed Certificates
☐ Export

RSA KEY
Size of Modulus in Bits : 512_
Public Exponent : ____3

The user can see that the PREVIOUS/NEXT buttons are disabled when he/she is defining the first/last operation.

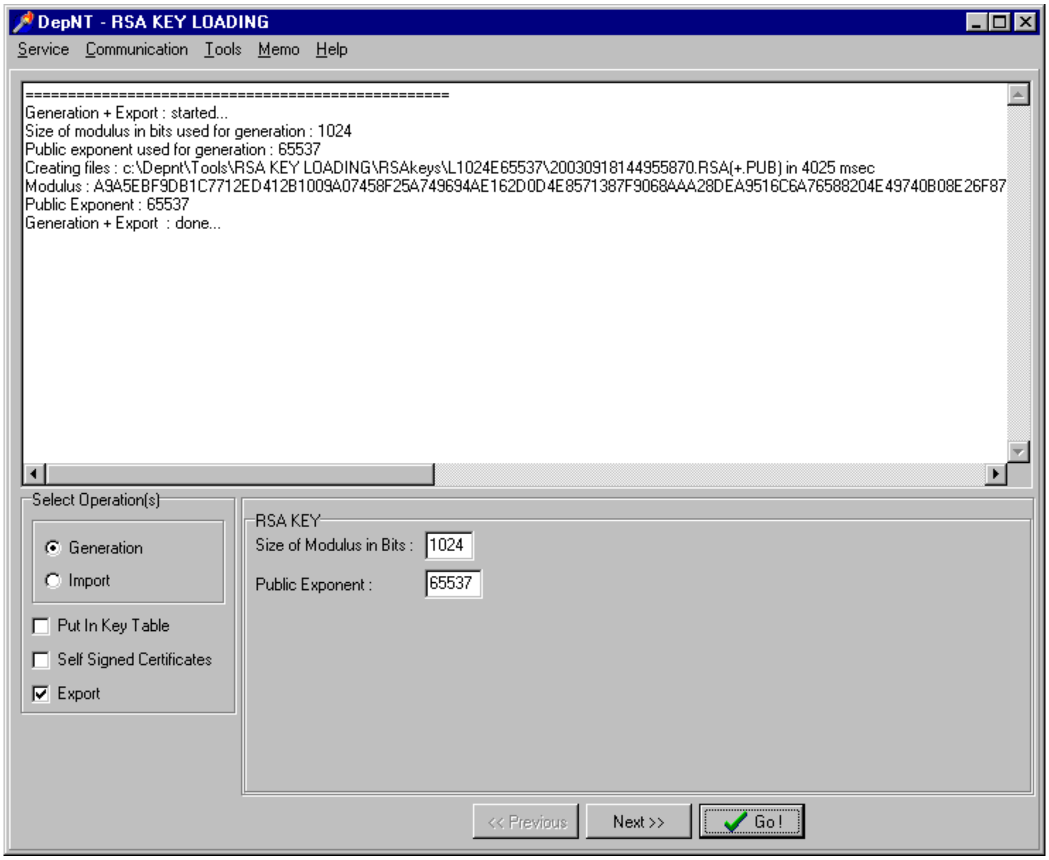
6.4. EXPORT RSA KEY

When selecting the "*Export RSA Key*" operation (Checkbox **Export**), the user can have a copy of the generated RSA key pair into files: one .RSA file and one .PUB file

The .RSA file contains the RSA key pair, where the public part of the key is recorded in clear, and the secret part of the key is recorded encrypted.

The .PUB file only contains the public part of the key, in clear.

"*Export RSA Key*" is an associated operation. Hence, it must be selected in combination with the *"Generate RSA Key"* operation.



```
==================================================
Generation + Export : started...
Size of modulus in bits used for generation : 1024
Public exponent used for generation : 65537
Creating files : c:\Depnt\Tools\RSA KEY LOADING\RSAkeys\L1024E65537\20030918144955870.RSA(+.PUB) in 4025 msec
Modulus : A9A5EBF9DB1C7712ED412B1009A07458F25A749694AE162D0D4E8571387F9068AAA28DEA9516C6A76588204E49740B08E26F87
Public Exponent : 65537
Generation + Export  : done...
```

Select Operation(s)

● Generation
○ Import

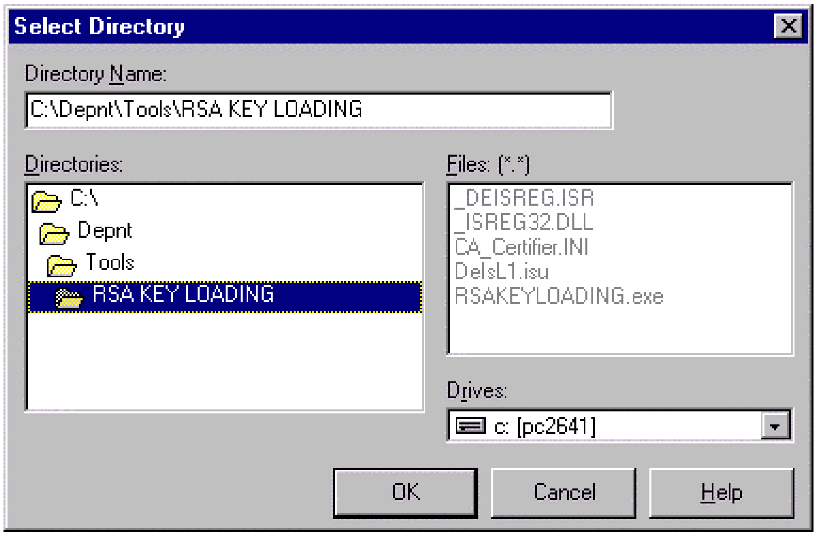
☐ Put In Key Table
☐ Self Signed Certificates
☑ Export

RSA KEY
Size of Modulus in Bits : 1024
Public Exponent : 65537

6.4.1. Directory Path

The user must select a directory to store the RSA keys that will be generated by the DEP Crypto Module, he/she can do that:

- either by choosing a complete path and writing it into the **Edit box**
- or selecting one by clicking on the BROWSE button. In this case a *Select Directory* window will appear. Click on the OK button to confirm the choice of the selected directory.



Select Directory

Directory Name: C:\Depnt\Tools\RSA KEY LOADING

Directories:
C:\
Depnt
Tools
RSA KEY LOADING

Files: (*.*)
_DEISREG.ISR
_ISREG32.DLL
CA_Certifier.INI
DelsL1.isu
RSAKEYLOADING.exe

Drives: c: [pc2641]

In the memo, the user can find the public part of the generated RSA keys (length of Modulus in Bits, Modulus, Public Exponent) and the name of the generated files ***.RSA** and ***.PUB**.

These two files have the structure of windows *.INI* files.

Here is an example of an *.RSA* file:

```
[Infos]
Version=01
Datim=18/02/2002 15:58:20

[Crypto]
nlen=1024
n=A4D360CA19B98B9BA05C32E3281784296A791960AFED19073AA0D014B8EA79676A9B9862B70F716BCE50
8646E89DE189D8B79BAD457E48CDD2B44B634B97334E2373977C05D9DAB19C525CFB850CC71A45E2F8A891
62DCF903D1E3628A9CE2D0AE123FB1075FB6913C9A07AA0C825303872D9DBD2DF1CA0220BF925EAF0602CB
e=65537
enckey=028F8000010881001406F8109FE9359F15AB63B2CD90F349E0A9AF80758200014283000742414E4
B535953840008DFA7BD4125C6333E850080A4D360CA19B98B9BA05C32E3281784296A791960AFED19073AA
0D014B8EA79676A9B9862B70F716BCE508646E89DE189D8B79BAD457E48CDD2B44B634B97334E2373977C0
5D9DAB19C525CFB850CC71A45E2F8A89162DCF903D1E3628A9CE2D0AE123FB1075FB6913C9A07AA0C82530
3872D9DBD2DF1CA0220BF925EAF0602CB8600030100018700 80EBE4FC334ECF412FADD24F88ED2A559CA6E
26BF461DEF0CB09BBE61CCE1DEB4595C21C4C07B227FA12FBD10E07751472E53E41DEEFF68DFE0969B7B9D
0A32BEBB9B8F5DBDA8A6DC0F673C5F65BA11EA0A89AEF43F9E21BD240439B1D8A95D180584E16B7C5B8388
20F3BCEABFA60716A8FFB518569B0B78F6AD833DBC186F1E5880040088E63D34BF2F5744DB3B4213D92F9F
FB50202056E5210456F2F020F413655F5C7DB9483A8E4C8B1E9820501DA47729E06D72FE1669806ADEF843
14EC82E456A8900407219A6DC3B4497EF922492847074DC84249FED0D196CCFE73FAC3DB83D25ADD37B8B9
B809260054500995BE0757AA1A4D36AE5ABBF04BFC897D60869BC82DBD58A0040030C529A81DBC1CE93099
88430337E96DFE75FCEB9E89FFFD530BD88E213BA1F51F10994661EE9750AD7CAB79FF0C1AB46C0836A3E5
50B1322F986D6A2C7AE538B00401CFA9815FA97ED6509CB7A6120B11740FA4357F1656D18B09BBC203B109
C27A98171D0329F6F1EA2E98707834B34FE9FEF647DEEDF1968121490F2B24B0DC82C8C0040D2A4E90E65F
9B7D0C490107D8E07222F7FCA6020B29D3879DF64AFB678D2AA5E9A949F4F9BC6A4F1FD06A63D16CB36DC1
114056099418577060BDD08766049A8
```

The meanings of the different fields are:

- Version: version of the file structure of the **.RSA** and **.PUB** files
- Datim: date and time of creation of the file
- Nlen: length in bits of the modulus
- N: modulus

- E: public exponent
- Enckey: secret part of the key encrypted with the transport key.

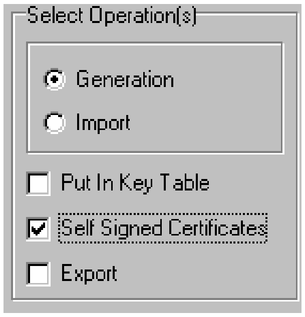
This is an example of the corresponding **.PUB** file with the public part only:

```
[Infos]
Version=01
Datim=18/02/2002 15:58:20

[Crypto]
nlen=1024
n=A4D360CA19B98B9BA05C32E3281784296A791960AFED19073AA0D014B8EA79676A9B9862B70F716BCE50
8646E89DE189D8B79BAD457E48CDD2B44B634B97334E2373977C05D9DAB19C525CFB850CC71A45E2F8A891
62DCF903D1E3628A9CE2D0AE123FB1075FB6913C9A07AA0C825303872D9DBD2DF1CA0220BF925EAF0602CB
e=65537
```

6.5. GENERATE CERTIFICATE REQUESTS BASED ON SELF-SIGNED CERTIFICATES (SSC)

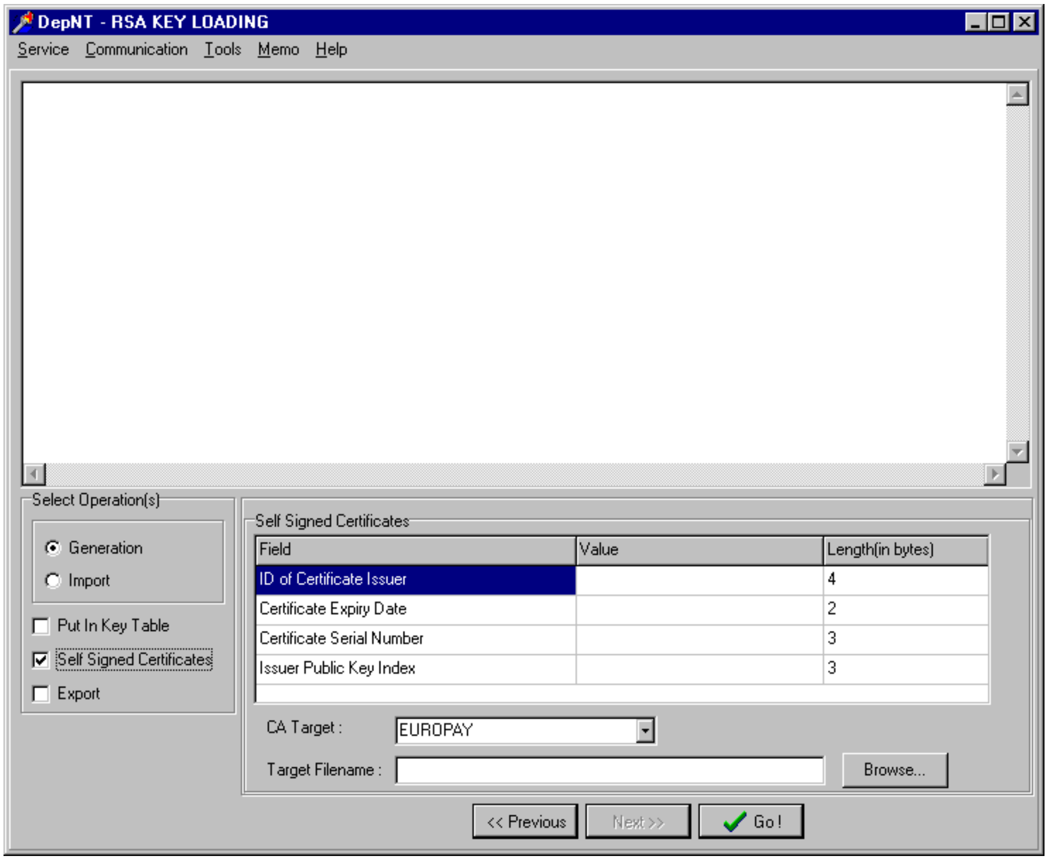
In this case, the user must activate the checkbox **Self Signed Certificates**. Immediately, the *Parameters* panel adjusts its content to show the data needed for the get SSC operation.



Select Operation(s)

● Generation
○ Import

☐ Put In Key Table
☑ Self Signed Certificates
☐ Export



Select Operation(s)

● Generation
○ Import

☐ Put In Key Table
☑ Self Signed Certificates
☐ Export

Self Signed Certificates

Field	Value	Length(in bytes)
ID of Certificate Issuer		4
Certificate Expiry Date		2
Certificate Serial Number		3
Issuer Public Key Index		3

CA Target : EUROPAY

Target Filename :

6.5.1. Choosing a CA Certifier

Selecting the CA Certifier in the scroll box **CA Target** automatically adjust the panel in order to show the necessary data to get the certificate requests files for that CA Certifier. Only the second column ("Value") can be modified with a given value, the two other columns ("Field" and "Length (in bytes)") are not to modify.

With the version 1.2.04 of the program, three values are defined for **CA Target**:

1. EUROPAY:

Selecting this value allows generating an EMV Certificate Request to submit to EUROPAY according to the following specifications:

- *EUROPAY, Registration Authority (RA) Interface Specification, version 2.1, November 2000*
- *EUROPAY, Registration Authority (RA), member Procedures, version 5.3, November 2000.*

2. VISA

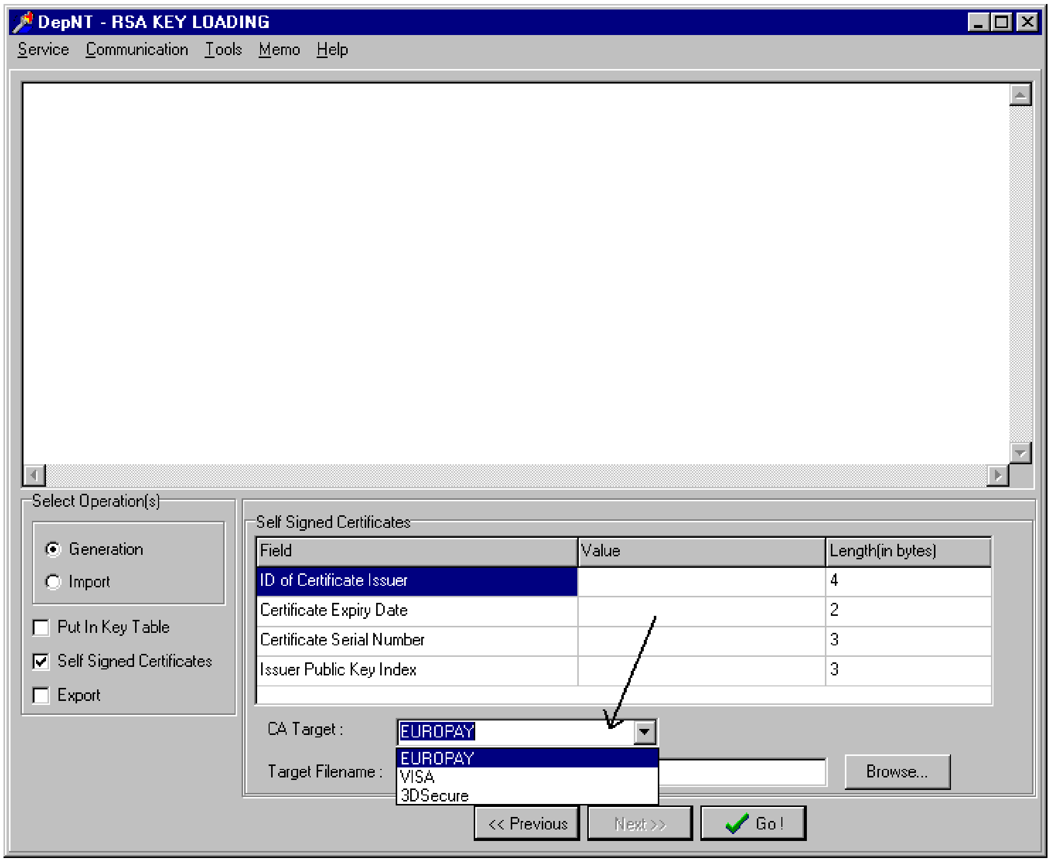
Selecting this value allows generating an EMV Certificate Request to submit to VISA according to the following specification:

- *VISA, Visa CA User's Guide, version 2.1, 31 March 2001.*

3. 3DSecure

Selecting this value allows generating a PKCS#10 Certificate Request to submit to VISA within the framework of 3-D Secure protocol, according to the following specification:

- *VISA, 3-D Secure Certificates. Certificate Request Process for Protocol 1.0.1, Member Version 1.2.5, June 24, 2002.*



Select Operation(s)

● Generation
○ Import

□ Put In Key Table
☑ Self Signed Certificates
□ Export

Self Signed Certificates

Field	Value	Length(in bytes)
ID of Certificate Issuer		4
Certificate Expiry Date		2
Certificate Serial Number		3
Issuer Public Key Index		3

CA Target : EUROPAY
EUROPAY
VISA
3DSecure

Target Filename :

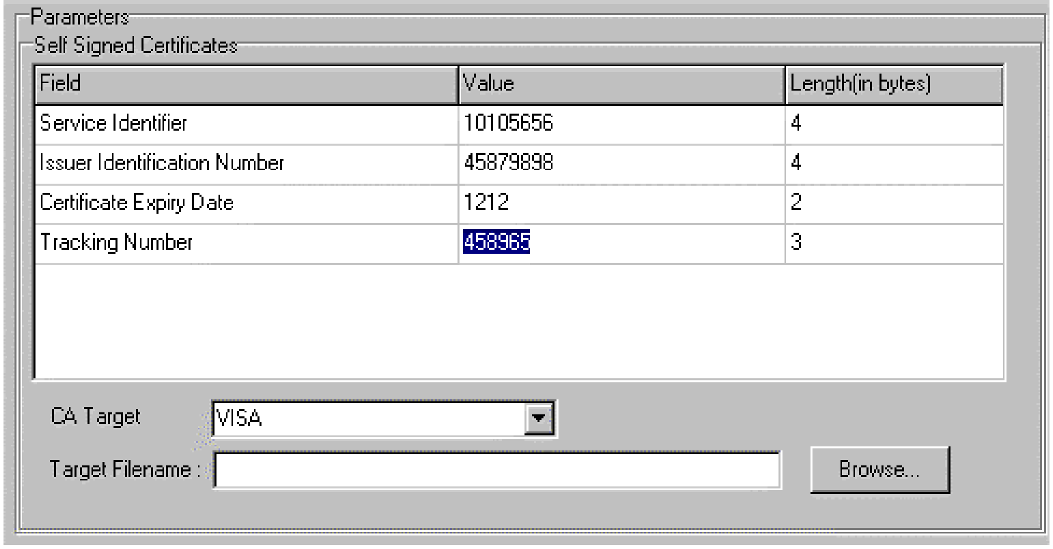
The number of CA Certifiers and their corresponding fields (name, length…) are defined in an .INI file (*CA_Certifier.INI*) that is delivered by *banksys*. It must be present in the same directory than the *DEP RSA Key Loading Program*.

The user can have a look at an example of such a file in the paragraph 8 on page 34.

6.5.2. Filling the Values in the Grid

The user must fill the grid to define the necessary parameters to get the certificate request files. Please note that the *DEP RSA Key Loading Program* does not verify these fields, it only verifies the lengths of all the values. A null value in the length column means no check is performed on the length of the corresponding field.

The meaning of the fields in the panel is given in ANNEX D: remarks on the data used for defining the Certificate request.



Parameters

Self Signed Certificates

Field	Value	Length(in bytes)
Service Identifier	10105656	4
Issuer Identification Number	45879898	4
Certificate Expiry Date	1212	2
Tracking Number	458965	3

CA Target: VISA

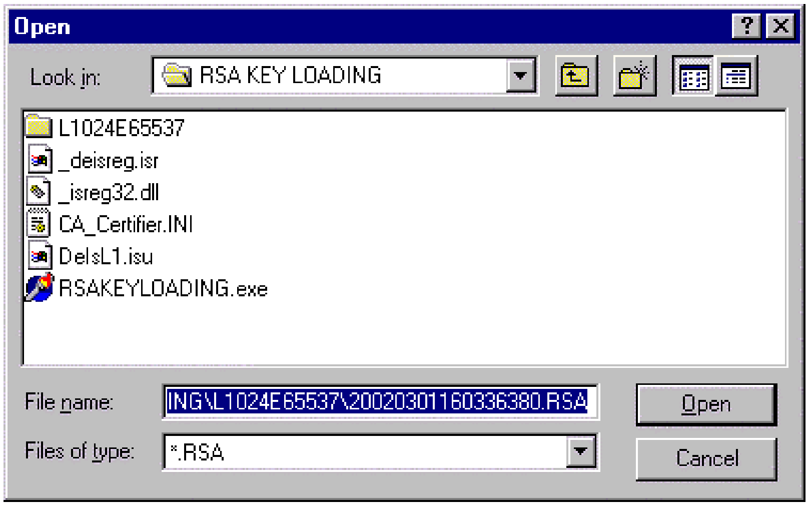
Target Filename :

When a field is optional and no value is entered, the corresponding information is not sent to the DEP Platform (i.e. Common Name for the 3Dsecure certificate).

6.5.3. Defining the Target file name

The result of "*Generate Certificates*" operation consists in one or more files. Files that are created are named *filename.ext*, where *filename* is the name indicated as **Target Filename** (if an extension is present, it will be ignored). The program, depending on the CA Certifier indicated in **CA Target**, automatically adds the correct extension "*ext*" (see paragraph 8 on page 34) for each created file. Indicating the name of the file in **Target Filename** can be done either by giving it explicitly or by clicking on the BROWSE button. A confirmation must be done to activate the selection.

If no directory is specified in the field of the **Target Filename**, the default directory where the *DEP RSA Key Loading Program* is will be used.



Look in: RSA KEY LOADING

L1024E65537
_deisreg.isr
_isreg32.dll
CA_Certifier.INI
DelsL1.isu
RSAKEYLOADING.exe

File name: ING\L1024E65537\20020301160336380.RSA

Files of type: *.RSA

6.5.4. Generating the Certificate Request based on SSC

The user must click on the GO! button to launch the operation. Remember that it is not a stand-alone operation: it must be associated with "*Generate RSA key*" operation or "*Import RSA key"* operation.

6.5.4.1. Generating PKCS #10 Certificates

The generation of this type of certificate can only be done in 2 steps, because the input data needed for the generation of the **PKCS #10** certificate need public parts of the newly generated RSA key:

- First step: "*Generate*" operation (i.e. **Generate** radio button) and "*Export*" operation (i.e. **Export** radio button),
- Final step: "*Import RSA Key*" operation (i.e. **Import** radio button) and "*Generate Certificates*" (i.e. **Self Signed Certificates** checkbox).

6.5.4.2. Generating EMV Certificates

For the generation of EMV Certificates (i.e. CA Target = EUROPAY or VISA), two options exist:

- A simultaneous selection of

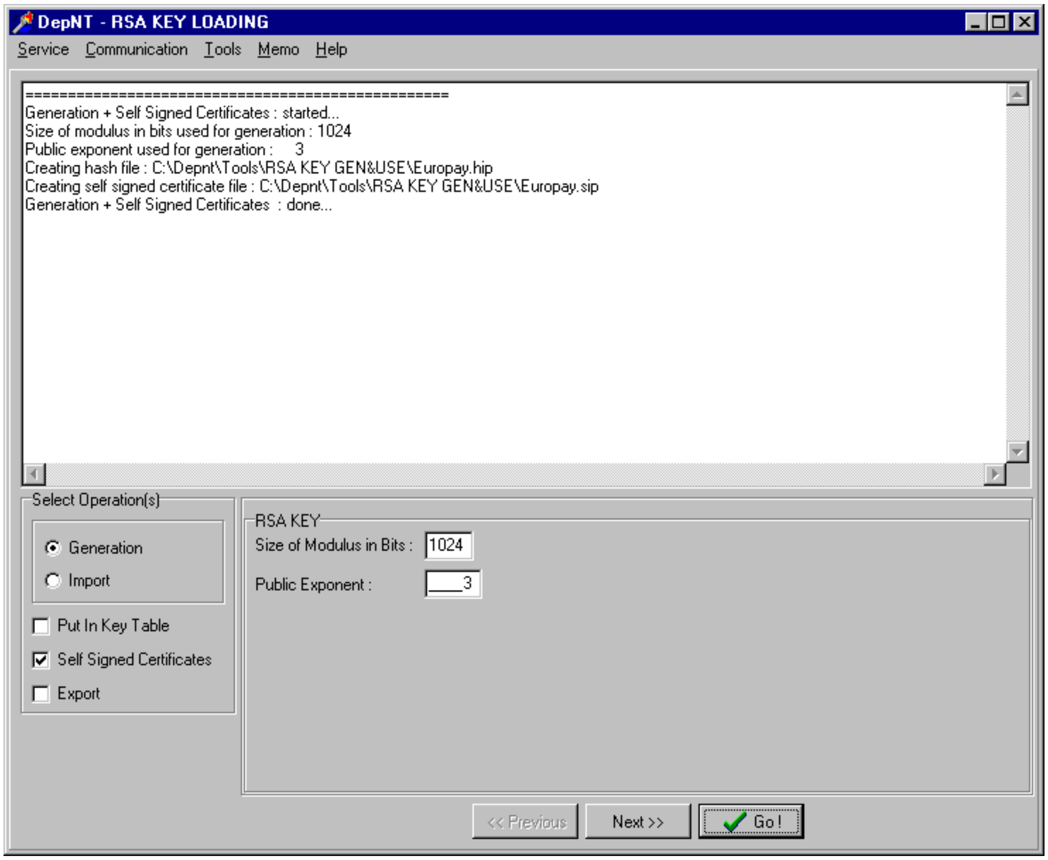
  - "*Generate RSA Key*" operation (i.e. **Generation** radio button)
  - "*Generate Certificates*" (i.e. **Self Signed Certificates** checkbox)

- Or a simultaneous selection of

  - "*Import RSA Key*" operation (i.e. **Import** radio button)

- "*Generate Certificates*" (i.e. **Self Signed Certificates** checkbox)

A summary of the process is given in the memo.



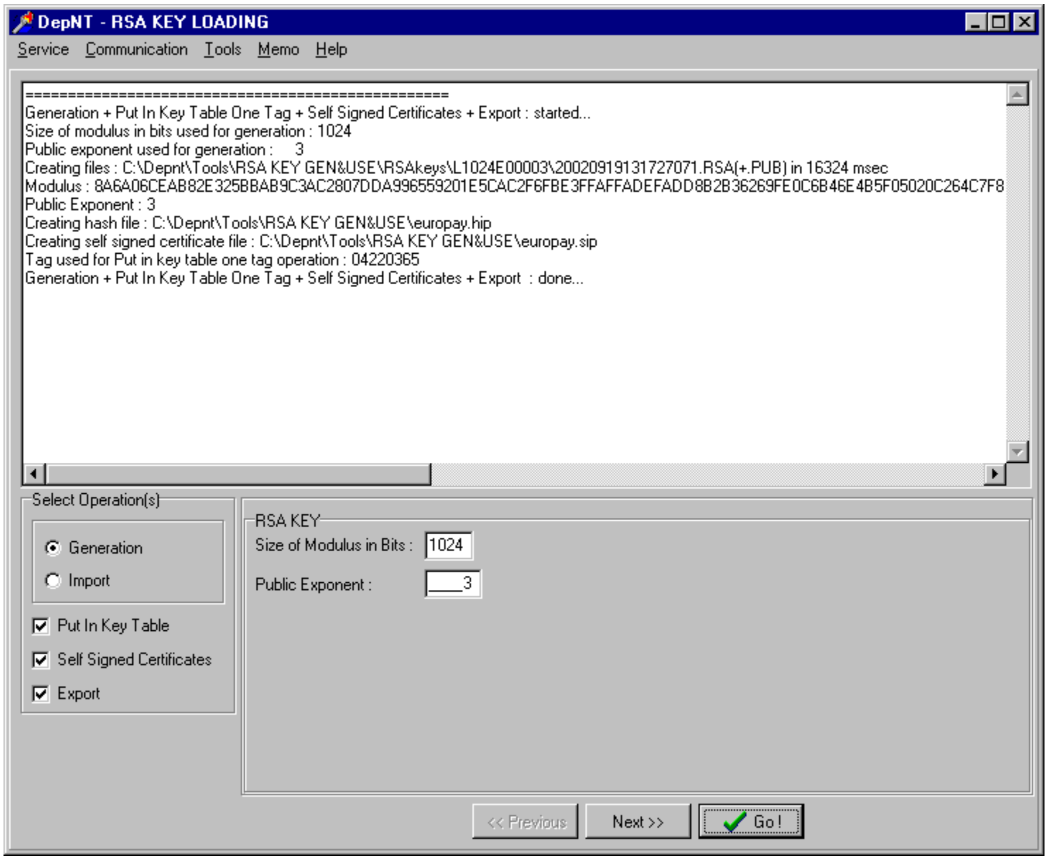
```
==================================================
Generation + Self Signed Certificates : started...
Size of modulus in bits used for generation : 1024
Public exponent used for generation :     3
Creating hash file : C:\Depnt\Tools\RSA KEY GEN&USE\Europay.hip
Creating self signed certificate file : C:\Depnt\Tools\RSA KEY GEN&USE\Europay.sip
Generation + Self Signed Certificates  : done...
```

● Generation
○ Import
□ Put In Key Table
☑ Self Signed Certificates
□ Export

Size of Modulus in Bits : 1024
Public Exponent : ____3

6.6. RUNNING ALL IN ONCE

The program also allows selecting one operation and all the three associated operations



```
==================================================
Generation + Put In Key Table One Tag + Self Signed Certificates + Export : started...
Size of modulus in bits used for generation : 1024
Public exponent used for generation :     3
Creating files : C:\Depnt\Tools\RSA KEY GEN&USE\RSAkeys\L1024E00003\20020919131727071.RSA(+.PUB) in 16324 msec
Modulus : 8A6A06CEAB82E325BBAB9C3AC2807DDA996559201E5CAC2F6FBE3FFAFFADEFADD8B2B36269FE0C6B46E4B5F05020C264C7F8
Public Exponent : 3
Creating hash file : C:\Depnt\Tools\RSA KEY GEN&USE\europay.hip
Creating self signed certificate file : C:\Depnt\Tools\RSA KEY GEN&USE\europay.sip
Tag used for Put in key table one tag operation : 04220365
Generation + Put In Key Table One Tag + Self Signed Certificates + Export  : done...
```

Select Operation(s)

● Generation
○ Import

☑ Put In Key Table
☑ Self Signed Certificates
☑ Export

RSA KEY
Size of Modulus in Bits : 1024
Public Exponent : ____3

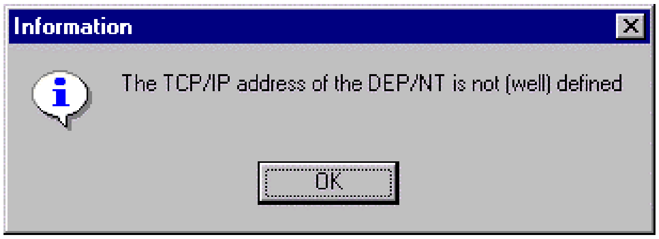
<u>Please note that it is only allowed for EMV certificates.</u>

7. TROUBLESHOOTING

In this chapter we show some of the troubleshooting that can occur when using the *DEP RSA Key Loading Program*.

7.1. PARAMETERS NOT/NOT WELL DEFINED

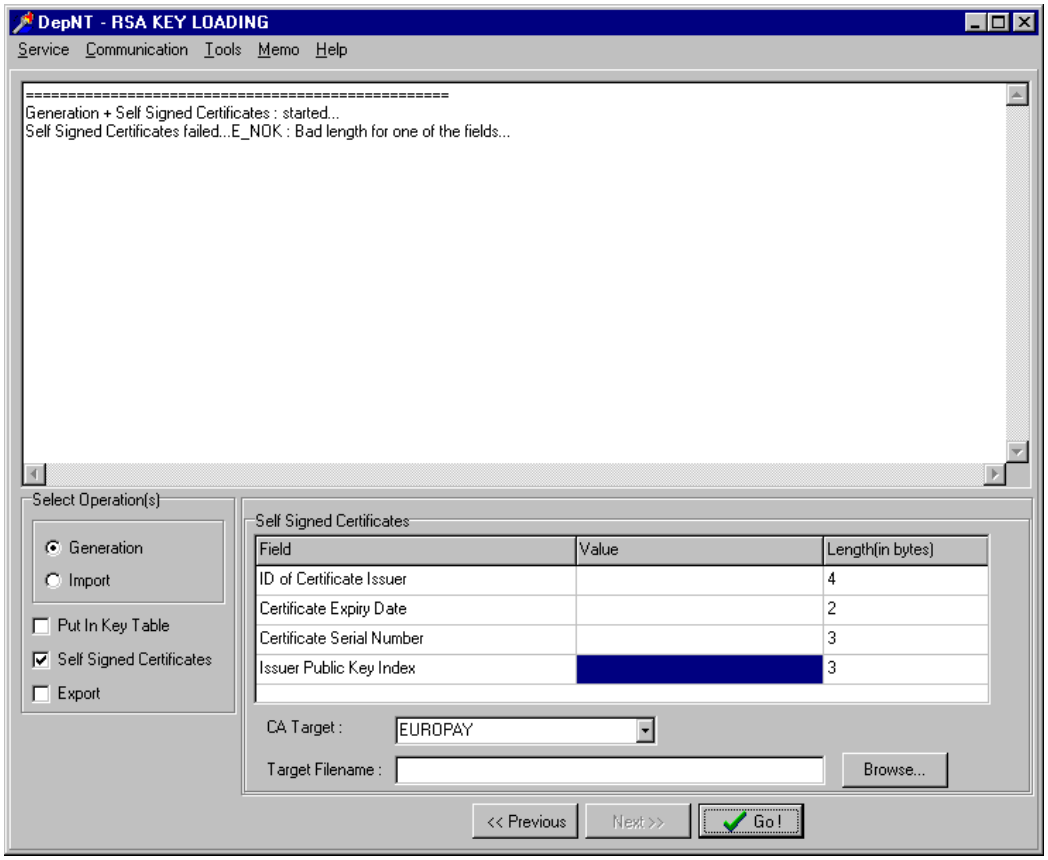
For example, when the user does not enter the TCP/IP address of the DEP Platform but nevertheless clicks on the GO! button, an *information* window appears.



The TCP/IP address of the DEP/NT is not (well) defined

In this case, one must click on the OK button, define correctly the needed parameters and launch again the operation. When an error occurs, the current operation is stopped and the whole process ends.

Some verification on parameters is also realized : e.g. when trying to get the SSC and the fields do not have the good length, an error is generated. Here the import has been realized but the operation of getting the SSC has aborted. The user must correct the data and launch the process again.



```
==================================================
Generation + Self Signed Certificates : started...
Self Signed Certificates failed...E_NOK : Bad length for one of the fields...
```

● Generation
○ Import

□ Put In Key Table
☑ Self Signed Certificates
□ Export

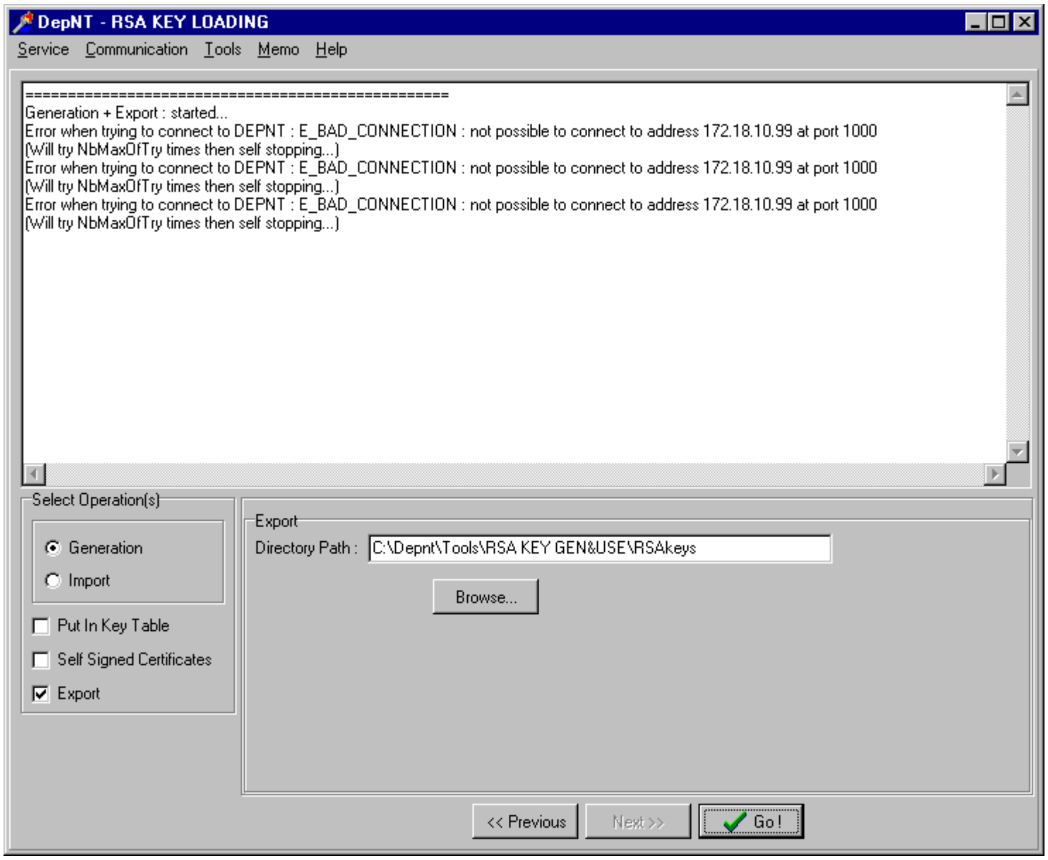
Field	Value	Length(in bytes)
ID of Certificate Issuer		4
Certificate Expiry Date		2
Certificate Serial Number		3
Issuer Public Key Index		3

CA Target : EUROPAY

Target Filename :

7.2. Problem of Connection

If there is a problem with the connection, or with the answer of the DEP Platform to a message, some information is reported in the memo. Usually, when a problem occurs, the *DEP RSA Key Loading Program* will try several times to do its current action before eventually stopping (see paragraph 5.3.3 on page 11).



```
==================================================
Generation + Export : started...
Error when trying to connect to DEPNT : E_BAD_CONNECTION : not possible to connect to address 172.18.10.99 at port 1000
(Will try NbMaxOfTry times then self stopping...)
Error when trying to connect to DEPNT : E_BAD_CONNECTION : not possible to connect to address 172.18.10.99 at port 1000
(Will try NbMaxOfTry times then self stopping...)
Error when trying to connect to DEPNT : E_BAD_CONNECTION : not possible to connect to address 172.18.10.99 at port 1000
(Will try NbMaxOfTry times then self stopping...)
```

Select Operation(s)

● Generation
○ Import

☐ Put In Key Table
☐ Self Signed Certificates
☑ Export

Export
Directory Path : C:\Depnt\Tools\RSA KEY GEN&USE\RSAkeys

7.3. MISSING KEY/CAPABILITY

In order to use the interfaces of the library PKI, the user must not forget to introduce the transport key into the key table of the DEP Crypto Module and to activate the required capability to insert a generated RSA key pair in the key table (*see PKI Library for DEP - Reference DFS Manual*).

8. ANNEX A: *CA_CERTIFIER.INI* FILE

The content of the file is as follows:

```
[NbCertifier]
Nb=3

[Certifier_1]
name=EUROPAY
NbField=4
Extension_hash=.hip
Extension_selfcertisspk=.sip
Extension_CAPublic=.sep
Extension_CACertificate=.CFE
Extension_CAhash=.hep
CA_Target=01
Field_1=ID of Certificate Issuer
Length_1=4
Tag_1=81
Field_2=Certificate Expiry Date
```

```
Length_2=2
Tag_2=82
Field_3=Certificate Serial Number
Length_3=3
Tag_3=84
Field_4=Issuer Public Key Index
Length_4=3
Tag_4=83

[Certifier_2]
name=VISA
NbField=4
Extension_hash=.vis
Extension_selfcertisspk=.inp
Extension_CAPublic=.v*
Extension_CACertificate=.i*
Extension_CAhash=.*
CA_Target=02
Field_1=Service Identifier
Length_1=4
Tag_1=94
Field_2=Issuer Identification Number
Length_2=4
Tag_2=91
Field_3=Certificate Expiry Date
Length_3=2
Tag_3=92
Field_4=Tracking Number
Length_4=3
Tag_4=93

[Certifier_3]
name=3DSecure
NbField=9
Extension_hash=.*
Extension_selfcertisspk=*.bin;*.b64
Extension_CAPublic=.*
Extension_CACertificate=.*
Extension_CAhash=.*
Extension_SELFPKCS10=.bin;.b64
CA_Target=03
Field_1=Hash Algo ID
Length_1=1
Tag_1=67
Field_2=Version
Length_2=1
Tag_2=67
Field_3=Signature Algo
MaxLength_3=2
Length_3=1
Tag_3=69
Field_4=Country Name
Length_4=0
Tag_4=69
Field_5=Organization Name
MaxLength_5=64
Length_5=0
Tag_5=69
Field_6=Organizational Unit Name
MaxLength_6=64
Length_6=0
Tag_6=69
Field_7=Locality Name
MaxLength_7=64
Length_7=0
Tag_7=69
Field_8=State/Province
MaxLength_8=128
Length_8=0
Tag_8=69
Field_9=Common Name
MaxLength_9=64
Length_9=0
Tag_9=69
```

The meanings of the different fields are:

- Nb: number of certifier authorities
- Name: name of the certifier number x
- Nbfield: number of field for SSC for certifier number x
- Extension_hash: extension of the file for will contain the hash of the Issuer
- Extension_selfcertisspk: extension of the file that will contain the SSC of the Issuer
- Extension_CAPublic : extension of the Public Key File coming from the CA
- Extension_CACertificate : extension of the certificate that comes from the CA
- Extension_CAHash : extension of the hash file coming back from the CA
- Extension_SELFPKCS10 : extensions for the PKCS10 self signed certificate request
- CA_Target: sequence number of the CA
- Field_y: signification of field y of certifier x
- Length_y: length of field y of certifier x
- Tag_y: internal tag of field y of certifier x

9. ANNEX B: SAVING OF A MEMO

This is an example of file that contains the information that was given in the memo:

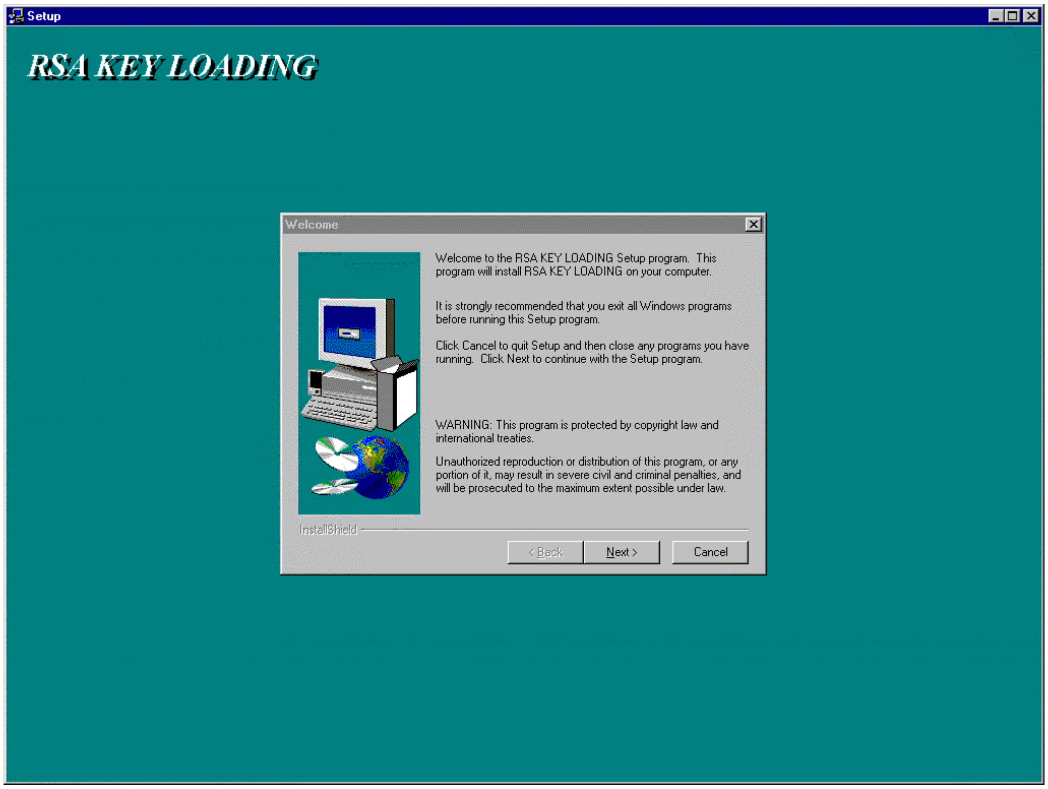
```
============================================
Generation + Put In Key Table One Tag + Self Signed Certificates + Export : started...
Size of modulus in bits used for generation : 1024
Public exponent used for generation :     3
Creating files : C:\Depnt\Tools\RSA KEY
GEN&USE\RSAkeys\L1024E00003\20020919141046209.RSA(+.PUB) in 17305 msec
Modulus :
87C8E27ADC137F58442FCDF19B8321B769C6060CAE939ED4FDCCBDEEB3948F6E6A45B9F6EA99335798F1A8
CE2DDE403B9D0EA855B40CE040A316A36D9A03C153A66183FF35F78BA13F107AEB124CB0EE88CE4ABB9CD3
442918342F4B1B22BA4684BCE7F24700B297E3F9860059068A12F4B96A621035E985C96FA6E60F423573
Public Exponent : 3
Creating hash file : C:\Depnt\Tools\RSA KEY GEN&USE\europay.hip
Creating self signed certificate file : C:\Depnt\Tools\RSA KEY GEN&USE\europay.sip
Tag used for Put in key table one tag operation : 04220355
Generation + Put In Key Table One Tag + Self Signed Certificates + Export  : done...
```

10. ANNEX C: INSTALLATION PROCEDURE

There exists an installation procedure for the *DEP RSA Key Loading Program*. To begin the installation wizard of the program, insert first the installation diskette or the CD-ROM and start the **Setup.exe**.

10.1. Welcome

The execution of the ***Setup.exe*** launches the installation by Install Shield of the *Program RSA KEY LODADING*. A Welcome window appears at the center of a Setup screen. It contains general recommendations and warnings about copyright laws and international treaties.



Welcome to the RSA KEY LOADING Setup program. This program will install RSA KEY LOADING on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program.

Click Cancel to quit Setup and then close any programs you have running. Click Next to continue with the Setup program.

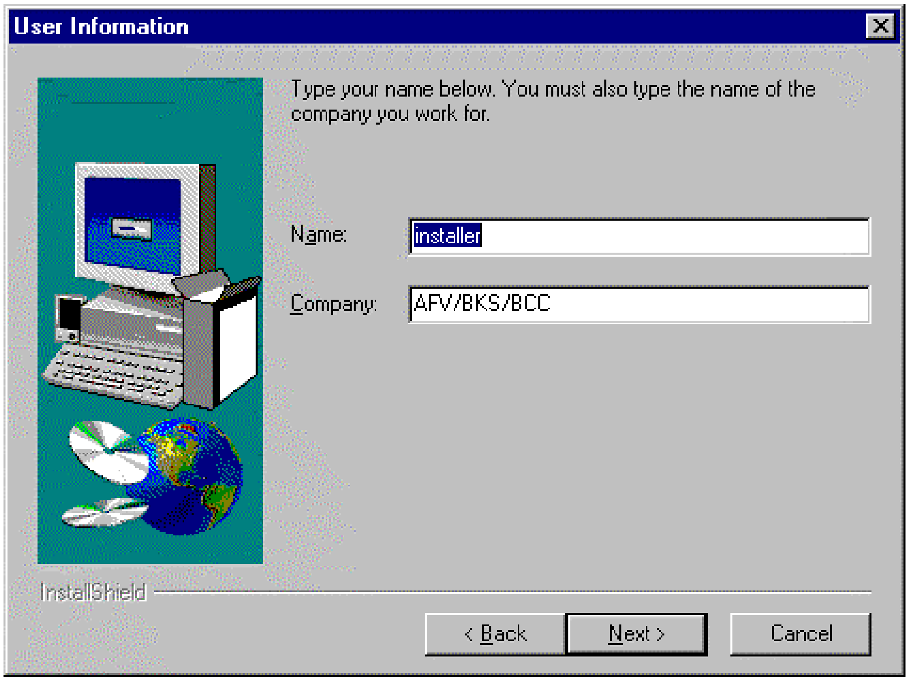
WARNING: This program is protected by copyright law and international treaties.

Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under law.

Click the NEXT button to continue the installation or click CANCEL to abort it.

10.2. User Information

The **User Information** screen allows the user to enter the names of the person and the company that performs the installation.



Type your name below. You must also type the name of the company you work for.

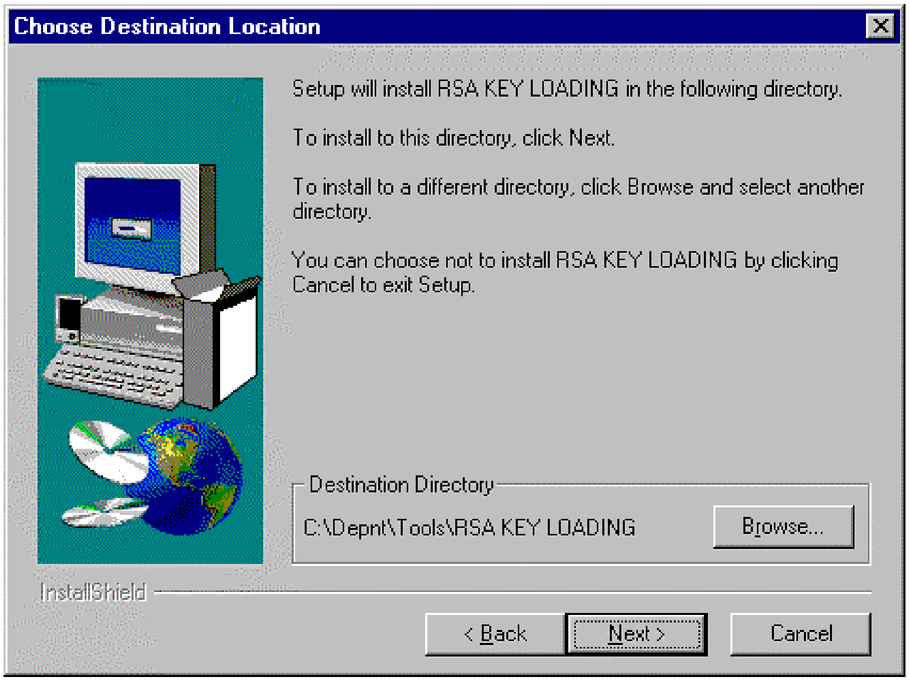
Name: installer

Company: AFV/BKS/BCC

Enter the name and the company, and click NEXT to continue, BACK to return to previous screen and CANCEL to abort the installation procedure.

10.3. Choose Destination Location

The **Choose Destination Location** window allows defining the path where the *DEP RSA Key Loading Program* is installed. The default path is **C:\Depnt\Tools\RSA KEY LOADING**.



Choose Destination Location

Setup will install RSA KEY LOADING in the following directory.

To install to this directory, click Next.

To install to a different directory, click Browse and select another directory.

You can choose not to install RSA KEY LOADING by clicking Cancel to exit Setup.

Destination Directory

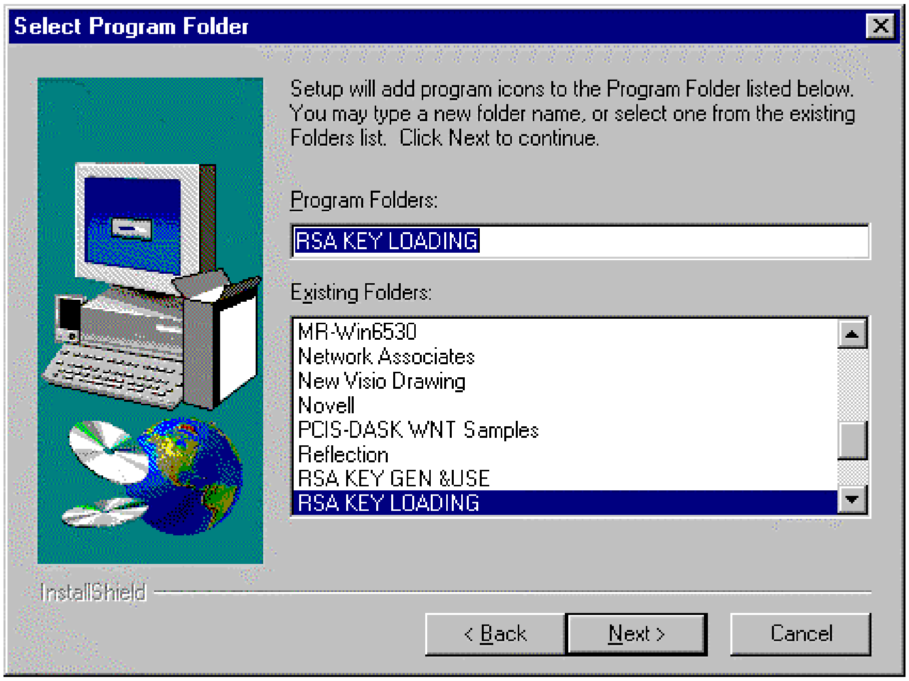
C:\Depnt\Tools\RSA KEY LOADING

Although it is recommended to use the default path, click the BROWSE button to select another directory for the installation of the *DEP RSA Key Loading Program* software.

Click the NEXT button to continue, BACK to return to previous screen or CANCEL to abort the installation procedure.

10.4. Select Program Folder

The **Select Program Folder** window is used to define the Program Folder where the program icons will be created.


Select Program Folder

Setup will add program icons to the Program Folder listed below. You may type a new folder name, or select one from the existing Folders list. Click Next to continue.

Program Folders:
RSA KEY LOADING

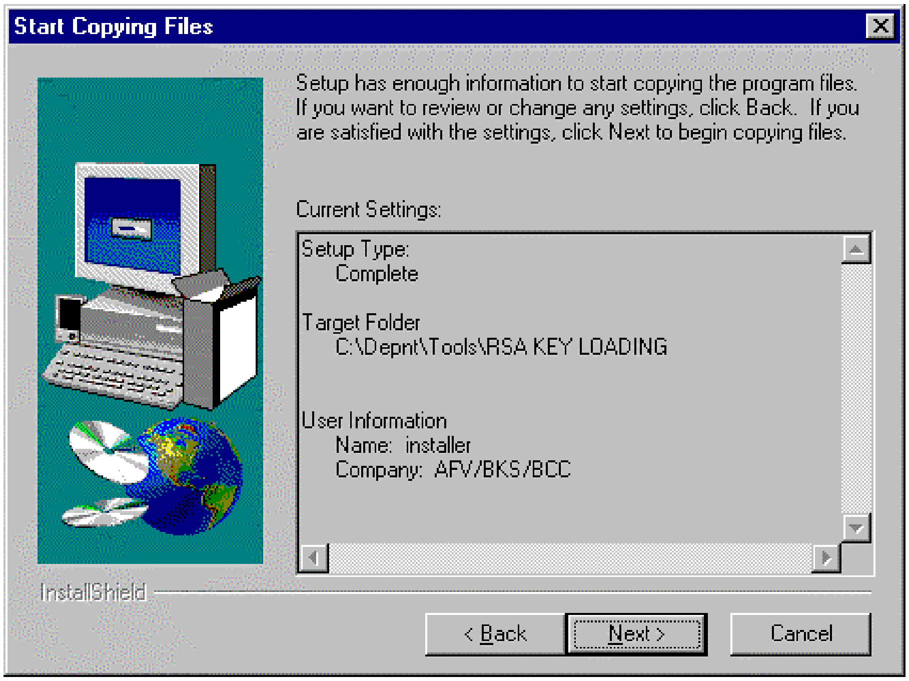
Existing Folders:
MR-Win6530
Network Associates
New Visio Drawing
Novell
PCIS-DASK WNT Samples
Reflection
RSA KEY GEN &USE
RSA KEY LOADING

By default, the Program Folders field contains **RSA KEY LOADING**. Although it is recommended to leave the default setting, it is possible to select another existing program folder or to enter a new one.

Click the NEXT button to continue, BACK to return to previous screen or CANCEL to abort the installation procedure.

10.5. START COPYING FILES

The **Start Copying Files** window gives an overview of the settings selected during the installation procedure.



Setup has enough information to start copying the program files. If you want to review or change any settings, click Back. If you are satisfied with the settings, click Next to begin copying files.

Current Settings:

Setup Type:
Complete

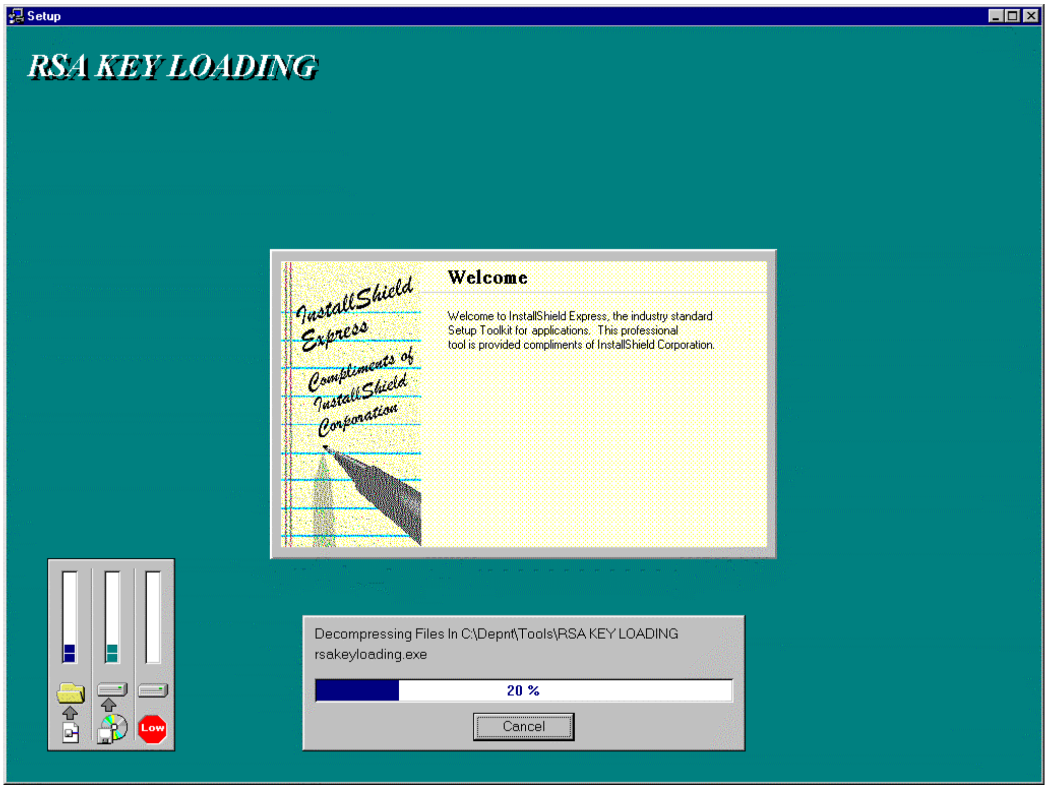
Target Folder
C:\Depnt\Tools\RSA KEY LOADING

User Information
Name: installer
Company: AFV/BKS/BCC

When the information is correct, click the NEXT button to continue, go BACK to modify some settings or use CANCEL to abort the installation procedure.

10.6. INSTALLING…

After having clicked on the NEXT button of the **Start Copying Files** window, all the required installations are executed.



Welcome to InstallShield Express, the industry standard Setup Toolkit for applications. This professional tool is provided compliments of InstallShield Corporation.

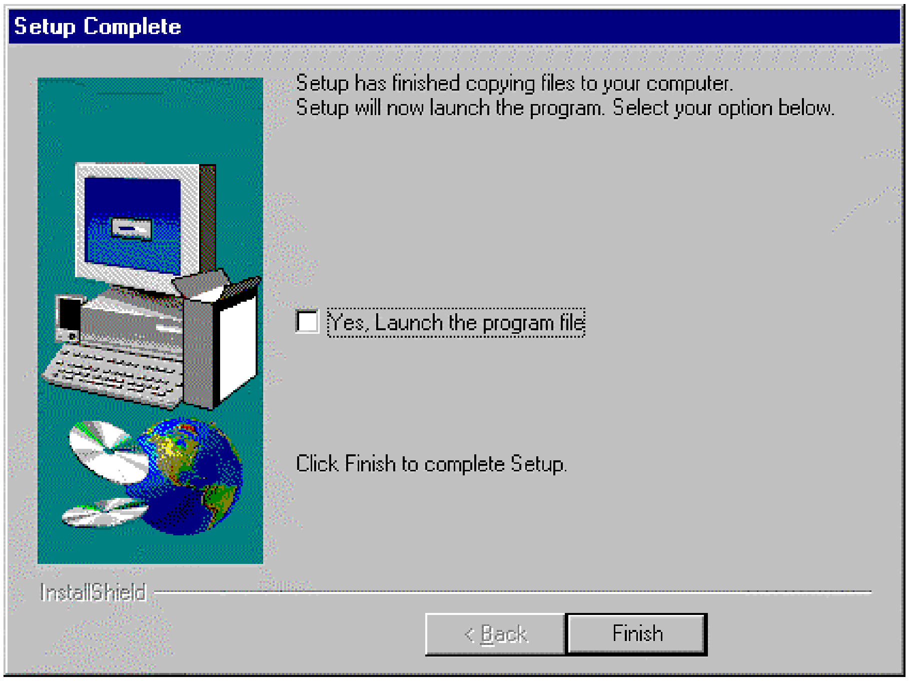
Decompressing Files In C:\Depnt\Tools\RSA KEY LOADING
rsakeyloading.exe

20 %

A progress bar and one or more status messages shall appear during the installation of the files.

10.7. Set-up Complete

When all the files and information are copied, a **Set-up Complete** window appears to confirm a successful installation.



Setup Complete

Setup has finished copying files to your computer.
Setup will now launch the program. Select your option below.

□ Yes, Launch the program file

Click Finish to complete Setup.

Click FINISH to end the installation.

11. ANNEX D: REMARKS ON THE DATA USED FOR DEFINING THE CERTIFICATE REQUEST

11.1. REFERENCES

This annex contains references to some standards documents:

ref 1: EUROPAY, Registration Authority (RA) Interface Specification, version 2.1, November 2000

ref 2: EUROPAY, Registration Authority (RA), member Procedures, version 5.3, November 2000For the purpose of this annex, some references are

ref 3: VISA, Visa CA User's Guide, version 2.1, 31 March 2001

ref 4: VISA, 3-D Secure Certificates. Certificate Request Protocol 1.0.1, member version 1.2.5, June 24, 2002.

11.2. CA TARGET = EUROPAY

Field	Length/Format	Description
ID of Certificate Issuer	8 hexadecimal digits, consisting in numeric digits left justified and padded on the right with hex. 'F's	Left-most 3-8 digits from the PAN, right-padded with hexadecimal'F'
Certificate Expiry Date	4 numeric digits	Month and Year (MMYY) after which this certificate is invalid
Certificate Serial Number	6 hexadecimal digits	Number chosen by the issuer
Issuer Public Key index	6 hexadecimal digits	Number chosen by the issuer, to uniquely identify the public key

Remarks:

1. In the table above, 'issuer' also means 'the requester of the certificate'.
2. Fields that have fixed value are not mentioned here. These fixed values are hard-coded in the program, and are according to the specifications ref 1 and ref 2.

11.3. CA TARGET= VISA

Field	Length/ Format	Description
Service Identifier	8 hexadecimal digits	Identifies a Visa Service. The proprietary Application Identifier Extension (PIX) is left justified and padded on the right with hex zeroes. Valid values for VSDC: 1010 = Credit/Debit 2010= Electron 3010 = Interlink 8010 = PLUS 999910 = Proprietary ATM
Issuer Identification Number	8 hexadecimal digits, consisting in numeric digits left justified and padded on the right with hex. 'F's	Issuer BIN, left justified and padded on the right with hex. 'F's
Certificate Expiry Date	4 numeric digits	Month and Year (MMYY) after which this certificate is invalid
Tracking Number	6 numeric digits	Tracking number, to identify a request for a Public Key Certificate

Remark:

1. In the text above, 'issuer' also means 'the requester of the

certificate'.

2. Fields that have a fixed value are not mentioned here. These fixed values are hard-coded in the program, and are according to the specification ref 3.

11.4. CA TARGET=3DSECURE

Field	Length/Format	Description
Hash Algo ID	2 digits. Possible values are 33: SHA1 34: SHA256 35: MD5 40: ISO 10118-2 (MDC2)	To indicate the hash algorithm used. **It shall be 33** in order to comply with the VISA's 3-D Secure Certificate Request Procedure [ref 4].
Version	2 digits.	Version number related to the version of PKCS#10 defining the syntax of the certificate request. **It shall be 00** for the version 1.7 of PKCS#10 standard.
Signature Algo	2 digits	To comply with the VISA's 3-D Secure Certificate Request Procedure [ref 4] **it shall be 05** to indicate the PKCS #1's **sha1WithRSAEncryption**
Country Name	2 characters ISO 3166-1 country codes	**It shall be BE** for Belgium
Organization Name	Max 32 characters	Organization Name
Organizational Unit Name	Max 32 characters	Organizational Unit Name
Locality Name	Max 64 characters	Locality Name
State/Province	Max 64 characters	State/Province
Common Name	Max 32 characters	Common Name

The table below gives the only permitted character set to use for 'Organization Name', 'Organization Unit Name', 'Locality Name', 'State/Province', 'Common Name' in order to comply with the VISA's 3-D Secure Certificate Request Procedure [ref 4]

Permitted Character Set	
A..Z	Uppercase alphabetic letters
a..z	Lowercase alphabetic letters
0..9	Decimal digits
_	Underscore
	Space character
'	Apostrophe
-	Hyphen
.	Period