Citigroup® Global Transaction Services

CitiDirect® Online Banking

Automated File and Report Delivery (AFRD) Utility Guide

December 2005

Proprietary and Confidential

citigroup

Table of Contents

SMIMETool version2: Highlights

Version 2.0 of the SMIMETool has some significant enhancements that meet audit standards and certain customer requirements. The enhancements include:

1) Generation of encrypt.ini and the decrypt.ini file via batch files.
2) The password for the customer's digital profile is encrypted. The integrity of the password is protected by a hash value.
3) Files can optionally be signed, or signed and encrypted for one or two recipients.

Warning: Please read this document carefully before unzipping the tool in your production environment, as it would overwrite any existing setup. If you wish to preserve the existing version of the tool, please create a back up of the existing version, or install the new version in a new environment.

Automated File and Report Delivery (AFRD) Utility Overview

The purpose of the Automated File and Report Delivery (AFRD) Utility is to provide CitiDirect® Online Banking clients with the means to facilitate the end-to-end automation of their File Import and/or Export processes. The Utility is incorporated into the client's existing automation routines as a means of automating the following File Import and File Export security functions:

- Digitally signing and/or encrypting data files intended for import into CitiDirect;
- Decrypting and/or verifying the digital signatures on reports or files exported from CitiDirect.

The AFRD Utility performs the previously mentioned functions according to the S/MIME industry specification (version 3) for encryption/decryption and digital signature processing.

The Utility works in combination with a personal digital certificate for decryption (of incoming files) and digital signature (for outgoing files) processing. (Please refer to the *AFRD S/MIME Security* section of this Guide for general PKI and digital certificates information.)

Note:

- The AFRD Utility does not perform file copying or deletion.
- Additional information on AFRD, such as Training Guides, FAQs and Quick Reference Cards, can be found in the Learning Center at www.citidirect.com.

This Guide outlines the installation, configuration and use of the AFRD Utility.

Requirements for the AFRD Utility

System Requirements

The AFRD Utility is compatible with Sun Microsystems™ Java™ RunTime Environment (JRE) version **1.3.x or 1.4.x** with the following operating systems:

- Windows® 2000 Professional
- UNIX® (Solaris 5.8)
- Red Hat® Linux 8.0

Note: To download JRE, go to the following URL: http://java.sun.com/j2se/downloads/index.html

Other Requirements

- A personal digital x509 certificate with private key. (Please refer to the *Digital Certificates* section of this Guide.)
- A Digital Certificate (public key) either in binary DER (.cer) format or PKCS#7 DER encoded (.p7c) format. This is necessary for running both the encryption and decryption functions. If the utility were to be used with **CitiDirect**, this would be the certificate downloaded from the **CitiDirect S/MIME Library.**
- If you want to execute the Utility from an application (as opposed to executing it from the command line), a program/script that triggers the AFRD Utility is necessary.

Note: The scheduling program is not required to reside on the same machine where the AFRD Utility executes.

Downloading, Configuring and Installing the AFRD Utility

The six steps involved in downloading, configuring and installing the AFRD utility are provided below. You must perform these steps before you can run the AFRD Utility.

If you are not familiar with the Sun Java Software (JRE) installation, classpath set up, etc., please refer to the section of the this Guide referenced in each step for detailed information.

Steps for AFRD Utility Setup

1. Download the AFRD Utility zip/tar file locally. (*Downloading the AFRD Utility*)

2. Extract the contents of the downloaded .zip or .tar file locally (e.g. to the C:/ drive). (E*xtracting the AFRD Utility Files)*

Warning: This will replace your existing setup. Make a backup of the existing setup before extracting the content from the tool if the existing version is needed.

3. ***For JRE: Version 1.4.x Only:***
   Determine the version of Java Software (JRE) on your machine. If the JRE version is **1.4.x,** copy these two files:

   local_policy.jar
   US_export_policy.jar

   from: **SmimeTool/Jar/ext** folder
   to **java/jre/lib/security** folder

   Example of the security folder location in Windows environment:

   C:/Program Files/Java/j2re1.4.x_x/lib/security
   These are Sun Microsystems' proprietary files and can also be downloaded from the Sun Microsystems' Web site. (*Setting Up the Classpath)*

4. Update the classpath to include Entrust™ Toolkit related jars in the classpath. (*Setting Up the Classpath)*
5. Edit **genINI.bat** file to add parameters and run to generate the .ini files. (*Generating/*Modifying the Encryption Initialization File)

6. Set parameters in .ini files (C:**/**SMimeTool**/**Run **/**encrypt.ini). Do not change auto generated values for the "epf", "pass" and "hash" keys.
   Note: Ensure **forward slash ("/")** is used in defining the folders/file path.
   (Modifying the Encryption Initialization File)

7. Execute instructions from the command prompt, or via a batch file (for Windows) or shell script (for UNIX). (*Running the AFRD Utility*)

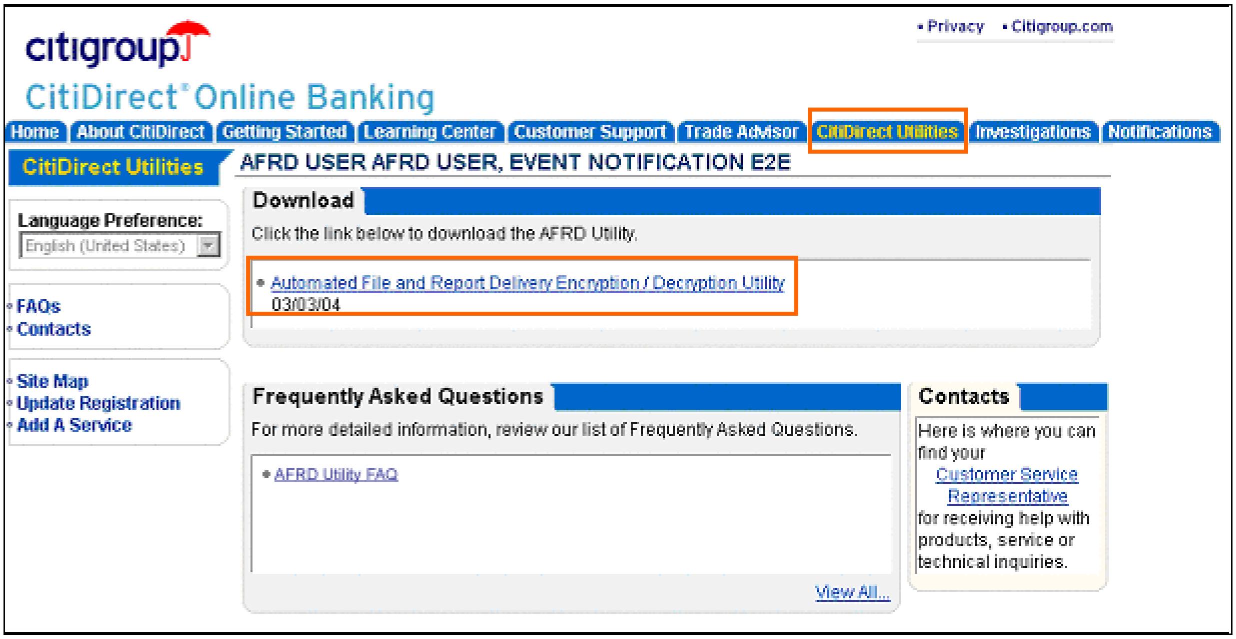
Note: Steps 1, 3 and 4 vary depending on which operating system is used to run the Utility: Windows or UNIX as indicated in the following sections.

Downloading the AFRD Utility

Follow the steps below to download the CitiDirect® Online Banking Automated File and Report Delivery (AFRD) Utility for Windows platforms. For UNIX platforms (.tar version), please contact your account representative.

1. From the CitiDirect Online Banking Web site (www.citidirect.com), click the **CitiDirect Utilities** tab.

Note: You must have entitlements to the AFRD service class in order to access this tab. If you do not have AFRD entitlements, this tab will not appear on the Web site. If you do not see the CitiDirect Utilities tab, please contact your Security Manager for assistance.



AFRD USER AFRD USER, EVENT NOTIFICATION E2E

Click the link below to download the AFRD Utility.

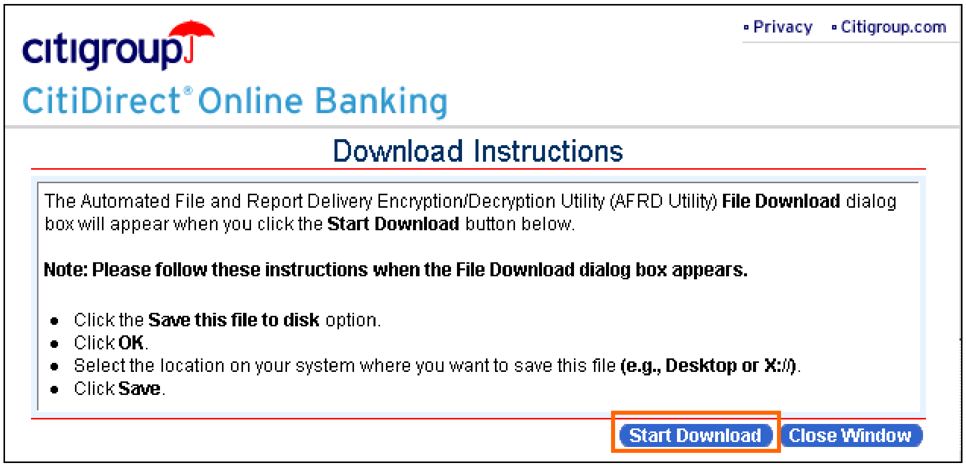
• Automated File and Report Delivery Encryption / Decryption Utility
03/03/04

For more detailed information, review our list of Frequently Asked Questions.

• AFRD Utility FAQ

Here is where you can find your Customer Service Representative for receiving help with products, service or technical inquiries.

2. In the **Download** section, click the **Automated File and Report Delivery Encryption/Decryption Utility** link. A **Download Instructions** dialog box appears.



Download Instructions

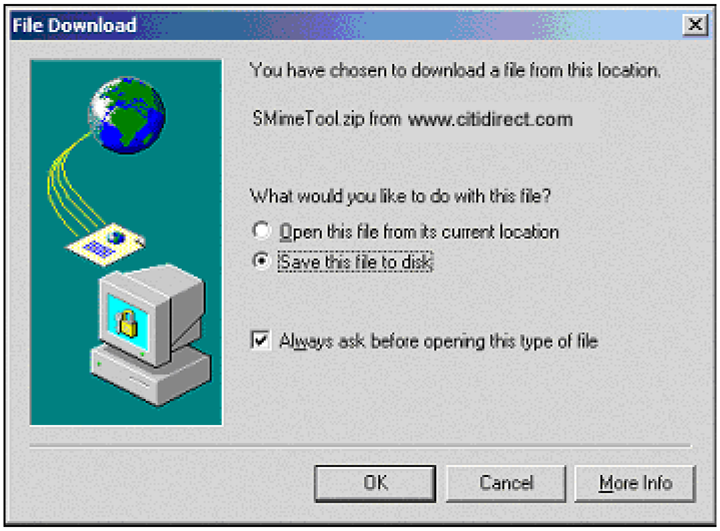
The Automated File and Report Delivery Encryption/Decryption Utility (AFRD Utility) **File Download** dialog box will appear when you click the **Start Download** button below.

Note: Please follow these instructions when the File Download dialog box appears.

- Click the **Save this file to disk** option.
- Click **OK**.
- Select the location on your system where you want to save this file **(e.g., Desktop or X:/)**.
- Click **Save**.

3. Review the instructions and then click the **Start Download** button. The **File Download** dialog box appears. (The dialog box depicts a download in a Windows NT environment for illustration purposes. The download process and screens that appear on your computer depend on the version of Windows/other operating systems on your computer.)

   A **Warning Message** appears to notify users that the extraction of the files would overwrite any existing files of the version that might exist at the location to which the files are being extracted.



You have chosen to download a file from this location.

SMimeTool.zip from www.citidirect.com

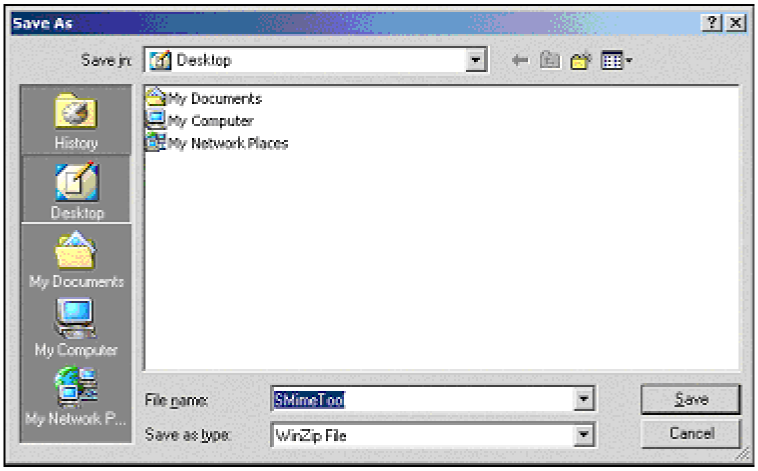
What would you like to do with this file?

○ Open this file from its current location

● Save this file to disk

☑ Always ask before opening this type of file

4. Select the **Save this file to disk** option, and then click **OK**. The **Save As** dialog box appears.

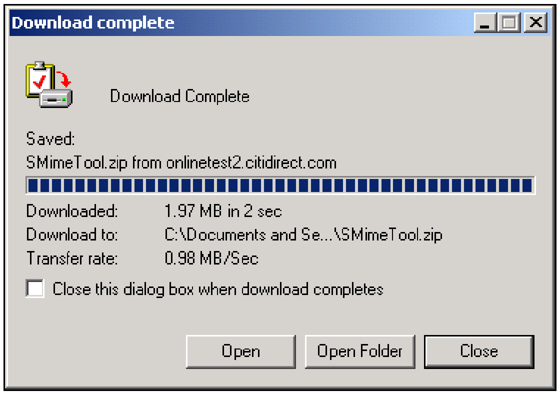


Save in: Desktop

File name: SMimeTool

Save as type: WinZip File

5. Select the save location for the AFRD Utility, and then click **Save** to begin downloading the Utility. When the download is finished, the **Download Complete** dialog box appears.



Download complete

Download Complete

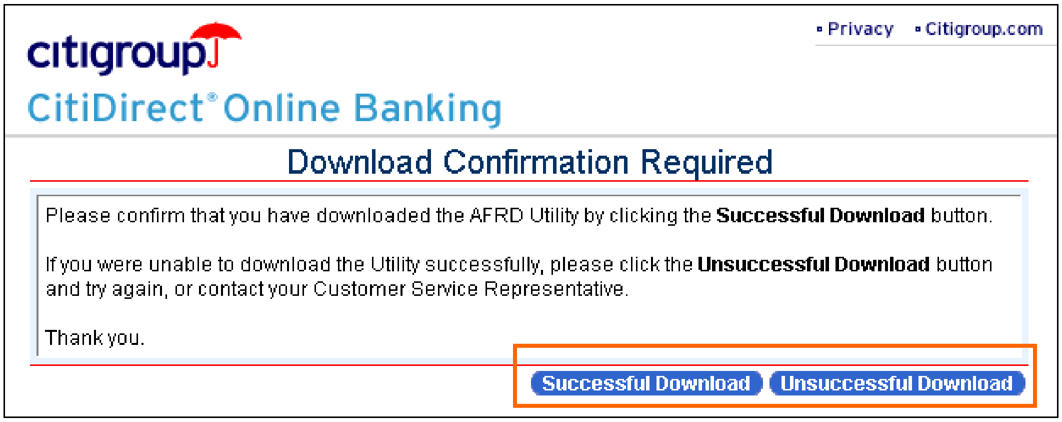
Saved:
SMimeTool.zip from onlinetest2.citidirect.com

Downloaded: 1.97 MB in 2 sec
Download to: C:\Documents and Se...\SMimeTool.zip
Transfer rate: 0.98 MB/Sec

☐ Close this dialog box when download completes

6. Click **Close**.

   The **Download Confirmation Required** dialog box appears.



Download Confirmation Required

Please confirm that you have downloaded the AFRD Utility by clicking the **Successful Download** button.

If you were unable to download the Utility successfully, please click the **Unsuccessful Download** button and try again, or contact your Customer Service Representative.

Thank you.

7. In the **Download Confirmation Required** dialog box, click the appropriate button as follows:

   - If you have successfully downloaded the AFRD Utility, click **Successful Download**.
   - If you were unable to download, click **Unsuccessful Download**.

Note: Clicking these buttons provides CitiDirect Online Banking with statistical information; it does not alert support personnel if there are download issues. If you have any issues when downloading the AFRD Utility, please contact your CitiDirect support personnel.

Extracting the AFRD Utility Files

Warning: This process would overwrite your existing setup. Please review this complete document before extracting the files.

After you have downloaded the compressed file, follow the steps below to extract the necessary AFRD Utility files. Procedures are provided for both Windows® and UNIX® operating systems.

For Windows® Operating Systems

1. Unzip the contents of the zip file, SMIMETOOL.ZIP to a local directory (e.g. C:/).

   Note: "**SMimeTool**" is the default folder name. You can choose to extract contents to a different folder using "WinZip" options.

2. If the zip file has been extracted under the C:/ directory, after unzipping, the /SmimeTool/ directory will contain the following folders:

FOLDER	CONTENTS
C:/SMimeTool/jar	Entrust™ Java Toolkit 6.0 SP2 .JAR files
C:/SMimeTool/jar/ext	JRE 1.3.x and 1.4.x files
C:/SMimeTool/Run	AFRD Utility executable files

For UNIX® Operating Systems

1. Expand and extract the files from the SMimeTool.tar file into a directory of your choice. A new directory called SMimeTool is created.
2. If the contents of the .tar file have been extracted under HOME, after extracting, the HOME directory will contain the following folders:

FOLDER	CONTENTS
HOME/SmimeTool/jar	Entrust™ Java Toolkit 6.0 SP2 .JAR files
HOME /SmimeTool/jar/ext	JRE 1.3.x and 1.4.x files
HOME/SmimeTool/Run	AFRD Utility executable files

Setting Up the Classpath

Follow the steps below to set up the required classpath. Procedures are provided for both Windows and UNIX operating systems.

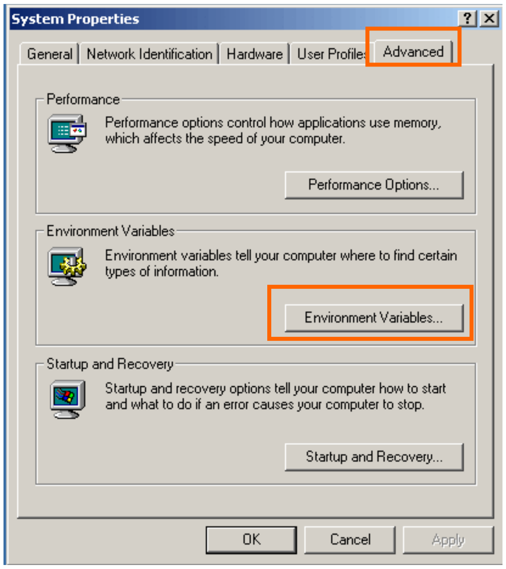
Note: All entries in the classpath **must** end with a value separator as follows:

Semicolon (;) *For Windows Operating Systems*

Colon (:) *For Unix/Linux Operating Systems*

For Windows® Operating Systems

1. From your computer's Control Panel, open the **System Properties** window and click the **Advanced** tab.

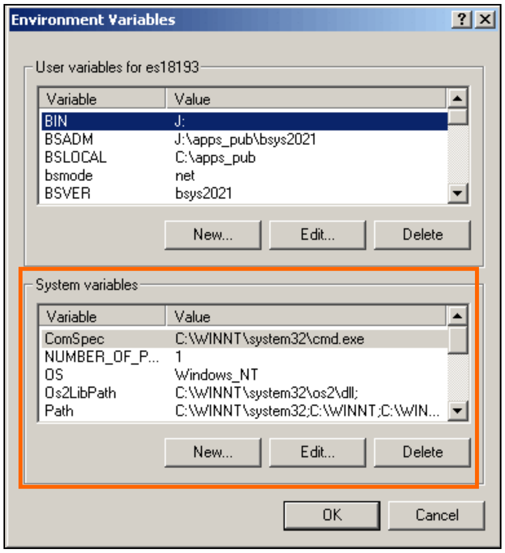


Performance options control how applications use memory, which affects the speed of your computer.

Environment variables tell your computer where to find certain types of information.

Startup and recovery options tell your computer how to start and what to do if an error causes your computer to stop.

2. Click **Environment Variables**. The **Environment Variables** dialog box appears.



User variables for es18193

Variable	Value
BIN	J:
BSADM	J:\apps_pub\bsys2021
BSLOCAL	C:\apps_pub
bsmode	net
BSVER	bsys2021

System variables

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
NUMBER_OF_P...	1
OS	Windows_NT
Os2LibPath	C:\WINNT\system32\os2\dll;
Path	C:\WINNT\system32;C:\WINNT;C:\WIN...

3. Under System Variables, search for the variable "CLASSPATH."

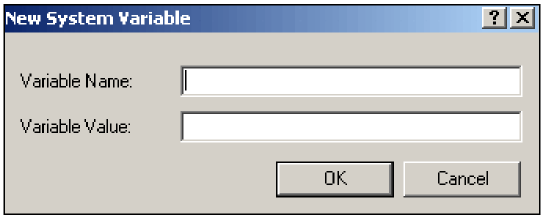
- If the variable "CLASSPATH" **is listed**:

  In the **System Variables** section, select the CLASSPATH variable and click **Edit**. Add the block of values listed below at the end of the already existing value; the value you add depends on the version of JRE that you are running.

  Note: Make sure to add a Semicolon after the existing values in the classpath. Each value in the classpath must end with a semicolon (" **;** ").

- If the variable "CLASSPATH" is **not listed**:

  In the **System Variables** section**,** click **New**. The **New System Variable** dialog box opens.



- In the **Variable Name** field, type "CLASSPATH." The **Variable Value** depends on which version of JRE you are running.

Note: **SMimeTool** is the directory where the tool is installed.

For JRE version 1.3.x the following must be added to the classpath:

```
.;C:\SMimeTool\Jar\entbase.jar;C:\SMimeTool\Jar\entcertlist.jar;
C:\SMimeTool\Jar\entcms.jar;C:\SMimeTool\Jar\entjsse.jar;
C:\SMimeTool\Jar\entp12.jar;C:\SMimeTool\Jar\entp7.jar;
C:\SMimeTool\Jar\entsmime.jar;C:\SMimeTool\Jar\entsmimev3.jar;
C:\SMimeTool\Jar\enttunnel.jar;C:\SMimeTool\Jar\entuser.jar;
C:\SMimeTool\Jar\entp5.jar;C:\SMimeTool\Jar\ext\jce1_2_2.jar;
C:\SMimeTool\Jar\ext\local_policy.jar;C:\SMimeTool\Jar\ext\US_export_policy.jar;
C:\SMimeTool\Jar\activation.jar;C:\SMimeTool\Jar\mail.jar;C:\SMimeTool\Jar\provid
erutil.jar;
```

For JRE version 1.4.x the following must be added to the classpath:

```
.;C:\SMimeTool\Jar\entbase.jar;C:\SMimeTool\Jar\entcertlist.jar;
C:\SMimeTool\Jar\entcms.jar;C:\SMimeTool\Jar\entjsse.jar;
C:\SMimeTool\Jar\entp12.jar;C:\SMimeTool\Jar\entp7.jar;
C:\SMimeTool\Jar\entsmime.jar;C:\SMimeTool\Jar\entsmimev3.jar;
C:\SMimeTool\Jar\enttunnel.jar;C:\SMimeTool\Jar\entuser.jar;
C:\SMimeTool\Jar\entp5.jar;C:\SMimeTool\Jar\activation.jar;C:\SMimeTool\Jar\mail.
jar;C:\SMimeTool\Jar\providerutil.jar;
```

Notes:

- **For JRE version 1.4.x,** copy the local **local_policy.jar, US_export_policy.jar**, from SMimeTool\Jar\ext folder to the **java\jre\lib\security folder**. For Windows, if the java has been installed under a default setting, following is an example of where the security folder can be located:

  C:\Program Files\Java\j2re1.4.2_06\lib\security
- These policy jars have been provided by Sun Microsystems and can also be downloaded from Sun Microsystems' Web site.
- Ensure that there is a semicolon between the existing value and the new being entered. All the described .jar files should be present in the directory (e.g., c:\SMimeTool\Jar) after you unzip the zip file.

For UNIX® Operating Systems

The classpath value depends on which version of JRE you are running. Classpath values for JRE versions 1.3.x and 1.4.x are provided below.

Classpath for JRE version 1.3.x:

```
.:JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/home/SMimeT
ool/Jar/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTool/Jar/entp12.jar:/hom
e/SMimeTool/Jar/entp7.jar:/home/SMimeTool/Jar/entp5.jar:/home/SMimeTool/Jar/entsmime.j
ar:/home/SMimeTool/Jar/entsmimev3.jar:/home/SMimeTool/Jar/enttunnel.jar:/home/SMimeToo
l/Jar/entuser.jar:/home/SMimeTool/Jar/ext/local_policy.jar:/home/SMimeTool/Jar/ext/US_
export_policy.jar:/home/SMimeTool/Jar/ext/jce1_2_2.jar:/home/SMimeTool/Jar/activation.
jar:/home/SMimeTool/Jar/mail.jar:/home/SMimeTool/Jar/providerutil.jar; export JAR
```

Classpath for JRE version 1.4.x:

```
.:JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/home/SMimeT
ool/Jar/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTool/Jar/entp12.jar:/hom
e/SMimeTool/Jar/entp7.jar:/home/SMimeTool/Jar/entp5.jar:/home/SMimeTool/Jar/entsmime.j
ar:/home/SMimeTool/Jar/entsmimev3.jar:/home/SMimeTool/Jar/enttunnel.jar:/home/SMimeToo
l/Jar/entuser.jar:/home/SMimeTool/Jar/activation.jar:/home/SMimeTool/Jar/mail.jar:/hom
e/SMimeTool/Jar/providerutil.jar; export JAR
```

Notes:

- **For JRE Vervsion 1.4.x:** Copy the **local_policy.jar** and **US_export_policy.jar** from the SMimeTool\Jar\ext folder to the java/jre/lib/security folder. If the JRE version 1.4.2 is installed, the path would be the following under the directory in which it is installed:

  Java/j2re1.4.2_06/lib/security
- These .jar files are provided by Sun Microsystems and can also be downloaded from Sun Microsystems' Web site.
- The .jar file path is already set in signandencrypt.sh and decryptandverify.sh. You must change the SMimeTool home directory in order for the script to work properly, or, if the classpath has been set in the environment and not through script file, you must delete the classpath set up in the script files.
- All entries in classpath must end with a colon **(":")** as a separator.

Generating/Modifying the Initialization Files

In order to encrypt or decrypt files the AFRD Utility reads the input parameters from initialization files (**encrypt.ini** and **decypt.ini**). In the previous version of the tool, these files were included in the contents of the zip file. In the current version, these files are created by editing and running the **genINI.bat** file that is included in the zip/tar file. (**getINI.bat** creates .ini files with an encrypted password for your security profile which is an enhancement to the current version).

The initialization files - ”**encrypt.ini**” and **“decrypt.ini”** - are generated by executing the **genINI.bat** file located in the /SMIMETOOL/RUN/ sub-directory. Once generated, **encrypt.ini** can be found in the AFRD /SMIMETOOL/RUN/ subfolder. These files once generated, must be modified **(**before the Utility can be used**)** to add additional parameters to ensure that it includes the specific file location on your computer for input, output and log files, etc.

Generating .ini file/s

You must following the steps below to generate encrypt.ini and decrypt.ini files.

1. Extract the contents of the zip/tar file to a local directory
2. Locate the **genINI.bat** in the folder ...**\SMimeTool\run.**
3. Right click on the **genINI.bat** and click **Edit.**
4. After **java**, enter the location and file name of your security profile.
5. After **%1** enter **no**, if there is only one recipient; enter **second**, if the file is to be encrypted for two recipients. Do not change the next parameters as this is the location in which .ini files would be created.

   Following is an explanation of genINI.bat file:

   Usage:<**epf/pfx** file location> %1 <for second recipient **second/no**><directory to create .ini file and key file")

   Example of the getINI.bat file after editing:

   ```
   java GenerateINI C:/cert/myprofile.pfx  %1 second C:/SmimeTool/run/
   ```

6. Once the **genINI.bat** is edited, execute the batch file from the DOS prompt followed by the password to your security profile. This execution would generate following files in **run** directory:
   - **Encrypt.ini** file with masked password and hash key, etc.
   - **Decrypt.ini** file with masked password and hash key, etc.
   - Key file.

   Below is a sample command provided when the tool is installed under C:/ drive:

   ```
   C:\SMimeTool\run>genINI mypassword01
   ```

Important: The key file contains the key to encrypt and decrypt the password in the ini file or in the command line option. The value is generated programmatically through the **genINI.bat** process. Any changes to values inside this file would break the encryption and decryption process.

Sample **Key File:**

```
69
109
-62
22
-99
35
-71
93
-105
-8
-38
-39
88
100
-98
-89
-15
127
-116
1
104
-15
-8
127
```

EDITING .INI FILES

Once .ini files are generated and dependent upon if the tool is utilized to encrypt, or decrypt the files, you would need to edit **encrypt.ini** and/or **decrypt.ini** files accordingly to provide values for the:

- inbound folder where the files need to picked up for processing.
- outbound folder where the files are to be placed after processing, location of recipient/s public keys, etc.

ENCRYPT.INI file – Example

Below is an example of the contents of the encrypt.ini file. Directory location/s in this example are for illustration only. The exact contents are based on the information from your computer, and actual locations would depend on the directories/file names on your computer.

```
===========================================================================
; ini file

profile=C:/cert/myprofile.pfx
pass=QEhiWdI20Gy0QFpK7Ze0kg==
hash=i+PJQ7Fgn/+/xRqtZm0KBK34PJ0=
InBoundFolder= C:/data/clear/mydatafile.txt
OutBoundFolder= C:/data/encrypted/
RcptCert=C:/CitiDirect/Certificate/citidirect.cer
SecondRcptCert=C:/Othersystem/certificate/OtherSystemCert.cer
LogPath= C:/SmimeTool/TOOL_LOG_FILES/<log directory with a '/' at end>
Encrypt=<true to sign and encrypt, false to sign only>

============================================================================
```

The above example assumes:

1. You have a certificate /security profile named **myprofile.pfx** that is located at:
   - **C:/security/profile/myprofile.pfx** (Your cert location/vendor could be different e.g. a **.epf** file instead of **.pfx**)

2. The Password to your certificate is entered through genINI.bat file.
3. The file is intended to be secured for two different recipients as follows: RcptCert and SecondRcptCert are the recipients whose certs should be in the values for these keys. If the file is intended for CitiDirect Online Banking use only, there should be only one recipient. This would need to be indicated in **generateINI.bat** file.
   - One of the recipient's (e.g. CitiDirect) public key/certificate is located at: **C:/CitiDirect/Certificate/citidirect.cer**
   - Second recipient's (e.g. some other system) public key/certificate is located at: **C:/Othersystem/certificate/OtherSystemCert.cer**

4. Files that need to be signed/encrypted for CitiDirect/all recipients is/are located at:
   - **C:/data/clear/mydatafile.txt**

5. The following directory has been created and exists for files to be saved after they are signed and encrypted:
   - **C:/data/encrypted/**

The following directory has been created and exists for the log files to be saved after the tool is executed:

- **C:/SmimeTool/TOOL_LOG_FILES/**

DECRYPT .INI file – Example

Below is an example of the contents of the **decrypt.ini** file. Directory location/s in this example are for illustration only. The exact contents are based on the information from your computer, and actual locations would depend on the directories/file names on your computer.

```
===========================================================================

; ini file
profile= C:/cert/myprofile.epf

pass=QEhiWdI20Gy0QFpK7Ze0kg==

hash=i+PJQ7Fgn/+/xRqtZm0KBK34PJ0=

InBoundFolder=<your inbound file here>

OutBoundFolder=<your outbound folder here>

SenderCert=<sender's signing certificate in .cer of .p7c>

LogPath=<log directory with a '/' at end>
===========================================================================
```

The above example assumes:

1 You have a Verisign® certificate named **myprofile.pfx** that is located at:

- **C:/cert/myprofile.pfx**

2. Encrypted password for the digital profile file entered through genINI.bat file.
3. Hash value of the password.
4. Inbound directory from which the decryption and verification of the encrypted file is done.
5. Output directory to which the decrypted file is copied.
6. Sender’s cert, which is used in signature verification process.
7. Logpath to which the log files for the decryption verification process is written.

Explanation of ENCRYPT.INI File Parameters

The table below provides descriptions of each of the ENCRYPT.INI file parameters.

PARAMETER	DESCRIPTION
Profile	Specifies the full file path and file name of your personal digital certificate (.epf or .pfx etc) file.
Pass	The Profile password in encrypted form.
Hash	Hash value of the encrypted password.
InBoundFolder	The initial location where the AFRD Utility will find import files for encryption. Multiple files can be encrypted simultaneously. A single file name can be stated as well, in which case the complete path and filename for the import file should be provided.
OutBoundFolder	The output location where the Utility will place the encrypted import files. After the files have been protected, you can decide to leave the encrypted import files in the same output location for retrieval by CitiDirect Online Banking or move the secured import files to another location for CitiDirect to retrieve.
RcptCert	Specifies the full file path for recipient encryption certificate. This will be required if **Encrypt** parameter is set to **True**. This certificate is the CitiDirect Public Key retrieved from CitiDirect Online Banking.
SecondRcptCert	Specifies the full file path for second recipient encryption certificate. This will be required if **Encrypt** parameter is set to **True** and the genINI is run with the "second" option. This certificate is the CitiDirect Public Key retrieved from CitiDirect Online Banking. NOTE: If the user does not want to have a second recipient then the option in **generateINI.bat** should be set to "no" instead of "second". This would generate an ini file for a single recipient only.
Encrypt	If **True**, the Utility encrypts and signs the file. If **False**, the Utility only signs it.

Notes:

- Use the **forward slash** (/) when specifying the parameters.
- Regardless of file name and/or extensions, the AFRD Utility attaches a **.p7m** file extension to the file name after it is processed. **.p7m** is the standard extension for S/MIME Format Message Files, recommended by the PKCS#7 specification.
- Because the .ini file contains a password, ensure that you place it in a secure location.

Running the AFRD Utility

The sections below provide instructions for running the AFRD Utility for each AFRD event type.

Signing and Encrypting (For File Import Files for CitiDirect)

The SignandEncrypt.bat file (signandencrypt.sh for UNIX), the actual encryption program, is located within the /SMIMETOOL/RUN/ sub-directory.

After the proper parameters have been saved in the ENCRYPT.INI file, the program SignandEncrypt can be executed in one of two ways:

- from an application (for example, the Windows Scheduler Utility); or
- from the command line.

Additionally, the command line parameters can be added to the batch file, which can be executed through a third program.

Further instructions for the command line option are provided below.

Command Line Instructions for Windows® Operating System

Below is an example of the command line instruction for running the Sign and Encrypt function within the Windows® operating system:

```
java CreateSMime
```

Example: **C:\SMimeTool\Run> java CreateSMime**

In this example, the AFRD Utility (SMIME Tool) has been installed on the C:\ drive.

When the command is executed as above the tool would read the parameters, e.g. users' profile, CitiDirect certificate, file folders, etc. from the .ini file. Alternately, the parameters can be provided at the command line after the above command. The –p <password> and the –h (hash) values would need to be copied from the ini files generated by genINI.bat. The tool would read the parameters from the command line. In this case the parameters from the **"encrypt.ini"** would not be taken into account.

Example:

```
C:\SMimeTool\Run>
C:\SMimeTool\Run>java CreateSMime -e "C:/security/profile/myprofile.pfx" -p 'QEhiWdI20Gy0QFpK7Ze0kg=='-h i+PJQ7Fgn/+/xRqtZm0KBK34PJ0= ' –i "c:/data/clear/*.*" –o "C:/data/encrypted/" –c "C:/entrust/Citidirect.cer" -L "C:/SMimeTool/TOOL_LOG_FILES/" y
```

Notes:

- The command line is case sensitive.
- After the command **java CreateSMime**, all path/s must be forward slashes ("/"), not backslash ("\").
- There is **NO** difference between the execution if the parameters have been passed at command prompt, or through **"encrypt.ini file".** Parameters/values passed through the command line override the values entered in the **"encrypt.ini"** file.

Command Line Instructions for UNIX® Operating System

Below is an example of the command line instruction for running the Sign and Encrypt function (signandencrypt.sh) within a UNIX® Operating system.

```
#!/bin/sh

#setup the classpath

JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/home/SMimeTool/Jar
/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTool/Jar/entp12.jar:/home/SMimeTool/
Jar/entp7.jar:/home/SMimeTool/Jar/entp5.jar:/home/SMimeTool/Jar/entsmime.jar:/home/SMimeTo
ol/Jar/entsmimev3.jar:/home/SMimeTool/Jar/enttunnel.jar:/home/SMimeTool/Jar/entuser.jar:/home
/SMimeTool/Jar/ext/local_policy.jar:/home/SMimeTool/Jar/ext/US_export_policy.jar:/home/SMime
Tool/Jar/ext/jce1_2_2.jar:/home/SMimeTool/Jar/activation.jar:/home/SMimeTool/Jar/mail.jar:/home
/SMimeTool/Jar/providerutil.jar:;

CLASSPATH=$JAR; export CLASSPATH
#run the shell
java CreateSMime
```

Signandencrypt.sh can also include parameters at the command line:

```
#!/bin/sh

# setup the classpath
JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/h
ome/SMimeTool/Jar/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTo
ol/Jar/entp12.jar:/home/SMimeTool/Jar/entp7.jar:/home/SMimeTool/Jar/entp5.
jar:/home/SMimeTool/Jar/entsmime.jar:/home/SMimeTool/Jar/entsmimev3.jar:/h
ome/SMimeTool/Jar/enttunnel.jar:/home/SMimeTool/Jar/entuser.jar:/home/SMim
eTool/Jar/ext/local_policy.jar:/home/SMimeTool/Jar/ext/US_export_policy.ja
r:/home/SMimeTool/Jar/ext/jce1_2_2.jar:/home/SMimeTool/Jar/activation.jar:
/home/SMimeTool/Jar/mail.jar:/home/SMimeTool/Jar/providerutil.jar:;

CLASSPATH=$JAR; export CLASSPATH

#run the shell
java CreateSMime -e C:/CERTS/<User’s Profile> -p <password key from the .ini file after
running genINI.bat> -h < hash key from the .ini file after running genINI.bat > -i
C:/SMimeTool/Files/<put here the name of file or file pattern> -o
C:/SMimeTool/Files/Encrypted -c <recipient’s digital certificate> -L <log folder path> y <”y”
for verification; “n” for signing only>
```

Note: The command line is case sensitive.

File Exports and Reports: Decrypting and Verifying

The DecryptandVerify.bat file (decryptandverify.sh for UNIX), the actual decryption program, is located within the /SMIMETOOL/RUN/ sub-directory.

After the proper parameters have been saved in the DECRYPT.INI file, the program DecryptandVerify can be executed in one of two ways:

- from an application (for example, the Windows Scheduler Utility); or
- from the command line.

Command Line Instructions for Windows® Operating Systems

Below is an example of command line instructions for running the Decrypt and Verify function within a Windows Operating System:

```
java ShowSMime

Example:
C:\SMimeTool\Run> java ShowSMime
```

In this example, the AFRD Utility (S/MIME Tool) has been installed on the C:\ drive.

When the command is executed as above, the tool would read the parameters e.g. users profile, CitiDirect certificate, file folders, etc. from the .ini file. Alternately, the parameters can be provided at the command line after the above command. The tool would read the parameters from the command line. In this case the parameters from the **“decrypt.ini”** would not be taken into account.

java ShowSMime -e <location and name of your certificate/private key> -p <password for your certificate/private key generated by genINI.bat in .ini file> -h <hash generated by genINI.bat in .ini file > -i < folder where encrypted the files are to be picked up> -o < folder where decrypted the files are to be places> –c <sender certificate/CitiDirect® certificate>

Example:

```
C:\SMimeTool\Run>
C:\SMimeTool\Run>java ShowSMime -e “C:/security/profile/myprofile.pfx” -p ‘QEhiWdI20Gy0QFpK7Ze0kg==’ –h i+PJQ7Fgn/+/xRqtZm0KBK34PJ0= –i “C:/data/clear/*.*” –o “c:/data/encrypted/” –c “C:/entrust/Citidirect.cer” -L "C:/SMimeTool/TOOL_LOG_FILES/"
```

Notes:

- The command line is case sensitive.
- After the command **java ShowSMime**, all path/s must be forward slashes(**“/”),** not back slashes (**“\”).**
- There is **NO** difference between the execution if the parameters have been passed at the command prompt, or through **“encrypt.ini file”.** Parameters/values passed through command line over-ride the values entered in the **“encrypt.ini”** file.

Command Line Instructions for UNIX Operating Systems

Below is an example of command line instructions for running the Decrypt and Verify functions (decryptandverify.sh) within a UNIX Operating System:

```
#!/bin/sh

# setup the classpath
JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/h
ome/SMimeTool/Jar/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTo
ol/Jar/entp12.jar:/home/SMimeTool/Jar/entp7.jar:/home/SMimeTool/Jar/entp5.
jar:/home/SMimeTool/Jar/entsmime.jar:/home/SMimeTool/Jar/entsmimev3.jar:/h
ome/SMimeTool/Jar/enttunnel.jar:/home/SMimeTool/Jar/entuser.jar:/home/SMim
eTool/Jar/1.4/local_policy.jar:/home/SMimeTool/Jar/ext/US_export_policy.ja
r:/home/SMimeTool/Jar/ext/jce1_2_2.jar:/home/SMimeTool/Jar/activation.jar:
/home/SMimeTool/Jar/mail.jar:/home/SMimeTool/Jar/providerutil.jar:;

CLASSPATH=$JAR; export CLASSPATH

#run the shell
java ShowSMime

#!/bin/sh

# setup the classpath
JAR=/home/SMimeTool/Jar/entbase.jar:/home/SMimeTool/Jar/entcertlist.jar:/h
ome/SMimeTool/Jar/entcms.jar:/home/SMimeTool/Jar/entjsse.jar:/home/SMimeTo
ol/Jar/entp12.jar:/home/SMimeTool/Jar/entp7.jar:/home/SMimeTool/Jar/entp5.
jar:/home/SMimeTool/Jar/entsmime.jar:/home/SMimeTool/Jar/entsmimev3.jar:/h
ome/SMimeTool/Jar/enttunnel.jar:/home/SMimeTool/Jar/entuser.jar:/home/SMim
eTool/Jar/ext/local_policy.jar:/home/SMimeTool/Jar/ext/US_export_policy.ja
r:/home/SMimeTool/Jar/ext/jce1_2_2.jar:/home/SMimeTool/Jar/activation.jar:
/home/SMimeTool/Jar/mail.jar:/home/SMimeTool/Jar/providerutil.jar:;

CLASSPATH=$JAR; export CLASSPATH
```

Parameters can also be specified at command line.

Example:

```
#run the shell (with parameters passed in command line over-riding the
#parameters in ini file)
java ShowSMime -e C:/CERTS/<YourProfile ".epf" or ".pfx" file> -p <password  key from the
.ini file after running genINI.bat> -h < hash key from the .ini file after running genINI.bat > for
the profile file> -i C:/SMimeTool/Files/Encrypted -o C:/SMimeTool/Files /Decrypted –c
<sender certificate>
```

Notes:

- The command line is case sensitive.
- After the command **java ShowSMime**, all path/s must be forward slashes (**“/”),** not back slashes (**“\”).**
- There is **NO** difference between the execution if the parameters have been passed at command prompt, or through **“decrypt.ini file”.** Parameters/values passed through the command line over-ride the values entered in the **“decrypt.ini”** file.

File Selection Wildcards for Encryption and Decryption

When setting the value for file location (both in the encryption and decryption instances), you can use the types of wildcards listed in the table below.

WILDCARD EXAMPLE	DESCRIPTION
InBoundFolder=C:/decryptiontool/1.3/encrypted/d*.ini.p7m	Only for .ini files starting with “d” (case sensitive)
InBoundFolder=C:/decryptiontool/1.3/encrypted/*.*	For all files in folder
InBoundFolder=C:/decryptiontool/1.3/encrypted/*.ini.p7m	For all file ending with “.ini.p7m”
InBoundFolder=C:/apps/SMimeTooljdk131/Run/trial/a*	For all files starting with “a”
InBoundFolder=C:/apps/SMimeTooljdk131/Run/trial/a*.*	For all files starting with “a”

Note: The following wildcard options are not supported:

<string> (no extension)
***<string>*.ext**
***<string>. ***

Check the Java Version on Your Machine

Check the version of java on your machine as follows:

Type the following text at the command prompt of the DOS window:

java – version

The output indicates the version of java on your machine, e.g. a version 1.4 is seen in the example below.

Example:

```
=========================================================
C:\>java -version
java version "1.4.2_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_0
Java HotSpot(TM) Client VM (build 1.4.2_03-b02, mixed mode)
==========================================================
```

Check the Classpath on Your Machine

Check if the classpath has been set as follows:

Below is an example of command to display classpath.

Open up a new DOS window and type:

set classpath

Output should show the values in classpath for JRE 1.4.x version

Example:

```
========================================================================
C:\>
C:\>set classpath
<below is an example of output>
classpath=.;C:\SMimeTool\Jar\entbase.jar;C:\SMimeTool\Jar\entcertlist.jar;C:\SMimeTool\
Jar\entcms.jar;C:\SMimeTool\Jar\entjsse.jar;C:\SMimeTool\Jar\entp12.jar;C:\SMimeTool\J
ar\entp7.jar;MimeTool_HOME\Jar\entsmime.jar;C:\SMimeTool\Jar\entsmimev3.jar;MimeTo
ol_HOME\Jar\enttunnel.jar;C:\SMimeTool\Jar\entuser.jar;MimeTool_HOME\Jar\entp5.jar;C:
\SMimeTool\Jar\activation.jar;C:\SMimeTool\Jar\mail.jar;C:\SMimeTool\Jar\providerutil.jar;
========================================================================
```

Setup Classpath

For instructions on setting up the classpath , please refer to the *Setting Up the Classpath* section of this Guide.

Add Updated Policy .jars to Java Directory

Please refer to the *Setting Up the Classpath* section of this Guide.

Obtain Recipient/Sender (CitiDirect) Public Certificate

The CitiDirect Online Banking certificate can be downloaded by logging into the CitiDirect S/MIME security library. Please contact your service representative for more information.

If the files are to be exchanged with a different system, the public certificate of that system should be used.

Obtain Your Certificate/Private Key

There are several vendors that issue encryption certificates including Verisign, Entrust, and Thawte, etc. Please contact these vendors directly.

AFRD S/MIME Security

This section provides a high level overview of the Public Key Infrastructure (PKI) utilized in Automated File and Report Delivery through CitiDirect® Online Banking and the AFRD Utility (S/MIME Tool). You can also find more information on PKI, and digital certificates and how they work on the Entrust and Verisign Web sites.

PKI

PKI (Public_Key Infrastructure) refers to the combination of software, encryption technology and services that enables organizations to protect the security of the information they exchange. PKI uses *"Public Key Encryption"* to secure files/data.

Public-key Encryption

Public-key encryption uses a pair of keys termed a **private key** and a **public key** to securely exchange data between two entities (in this case between CitiDirect Online Banking and the client).

The **sender uses the public key**, whereas the corresponding **private key** is used to decrypt the data after receiving it from the sender.

The **private key** is known only to the profile owner and is generally secured via a password. The private key must be kept secret.

The **public key** is distributed to any user/entity that wants to communicate securely with the owner of **private key**. The sender encrypts the file using the recipient's **public key**. Once the file is encrypted, it is illegible and can be viewed only after the recipient using the private key decrypts it**.** It cannot be viewed/decrypted by a third party.

Authentication/Signature Validation

Authentication is used to verify that the information comes from a trusted entity and is authentic. The recipient of the data knows who created it and that it has not been tampered with or altered in any other way after it was created.

Digital Signatures

A digital signature is basically a way to ensure that an electronic document (e-mail, spreadsheet, text file) is authentic. The Digital Signature Standard (DSS) is based on a type of public-key encryption method that uses the Digital Signature Algorithm (DSA). DSS is the format for digital signatures endorsed by the U.S. government. The DSA algorithm consists of a private key, known only by the originator of the document (the signer), and a public key.

Signature Validation

Signature validation includes the following:

- Contents of the data have not been altered after it was signed.
- Signing certificate is trusted by the recipient.
- Signing certificate is not expired.
- Signing certificate has not been revoked/ added to CRL-certificate revocation list. (For on-line validation only.)

AFRD / “S/MIME Tool”

AFRD and the “S/MIME Tool” utilize public key encryption. Users have three security options available in the *Delivery Options Library*:

- **Clear** – no encryption/digital signatures
- **Signed Only** – digitally signed files, no encryption
- **Signed and Encrypted** – files digitally signed and encrypted

Files Sent to CitiDirect

Files are transferred via a Secure Sockets Layer (SSL) encrypted sessions and data are secure during transmission.

Clear: No security is applied for files sent to CitiDirect Online Banking, however, files are received through https and file transmission is secure.

Signed Only:

1. Customers upload the public key/public certificate into the **S/MIME Security Library** in CitiDirect Online Banking and associate this certificate to the AFRD job through the Delivery Option Library.

2. Files are digitally signed by customers using their private key. This is accomplished via CitiDirect **S/MIME Tool**, **Entrust Entelligence™**, or any other tool that meets PKCS#7 message standards requirements.

3. When the file is received by CitiDirect, signature validation is done to ensure:

   - File contents have not been altered after the file was signed.
   - Signature on the file matches the certificate selected by the user in the delivery option for the job.
   - Certificate uploaded into CitiDirect has not expired.

If any of the conditions are not met, the job is set to “**FAILED**” and file is not processed.

Signed and Encrypted

1. File/s are:

   - Digitally signed by customers using their private key; and
   - Encrypted using **CitiDirect public key** for confidentiality.

This can be accomplished via the CitiDirect "**S/MIME TOOL**", Entrust "**Entelligence**", or any other tool. Once the file is encrypted, contents cannot be viewed

2. Customers upload the **public key/public certificate** into the "**S/MIME Security Library**" in CitiDirect and associate this certificate to the AFRD job through the **Delivery Option** library.

3. When CitiDirect receives the file, it is decrypted using **CitiDirect Private Key**.

4. Signature validation is done to ensure:
   - File contents have not been altered after it was signed.
   - Signature on the file matches the certificate selected by the user in the delivery option for the job.
   - Certificate uploaded into CitiDirect has not expired.

If decryption/signature validation fails, the job is set to "**FAILED**" and file is not processed.

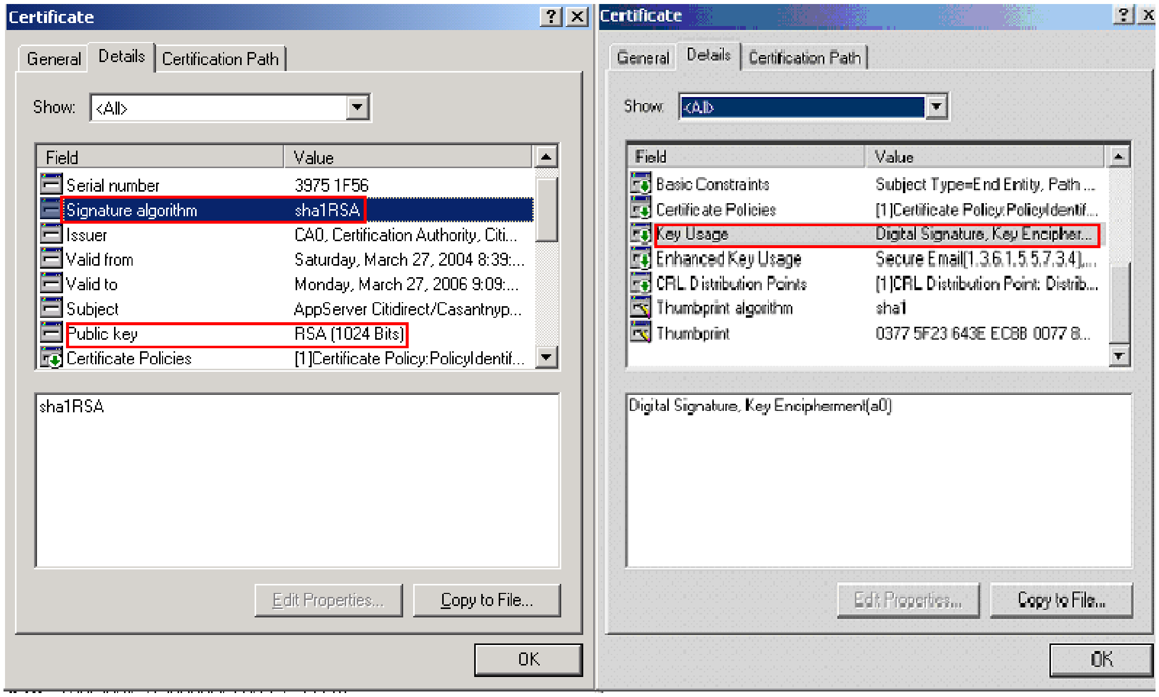
Digital Certificates

Digital Certificates can be requested from any Certificate Authority (CA). However, certificates obtained by the client for digital signing must meet the following minimum criteria:

- A valid, (not expired), X.509 signing certificate must be used (email certificate).
- RSA public key (signature) algorithms using key lengths of 1024 bits (or larger).
- Message Digest Algorithm must be SHA1-RSA.
- Encryption algorithm is DES3 (triple DES).
- Key usage must include digital signature (for signing files) and key encipherment (for encrypting the files).
- Certificate (Public Key Only) should be exported in the following format for upload to the CitiDirect "S/MIME Library":

  - **.p7c** (* PKCS7 cryptographic message syntax standard certificate)
  - **.cer** (DER encoded binary X.509 Certificate)
  - **.cer** (Base64 encoded X.509 Certificate)

Example:



Field	Value
Serial number	3975 1F56
Signature algorithm	sha1RSA
Issuer	CA0, Certification Authority, Citi...
Valid from	Saturday, March 27, 2004 8:39:...
Valid to	Monday, March 27, 2006 9:09:...
Subject	AppServer Citidirect/Casantnyp...
Public key	RSA (1024 Bits)
Certificate Policies	[1]Certificate Policy:PolicyIdentif...

sha1RSA

Field	Value
Basic Constraints	Subject Type=End Entity, Path ...
Certificate Policies	[1]Certificate Policy:PolicyIdentif...
Key Usage	Digital Signature, Key Encipher...
Enhanced Key Usage	Secure Email(1.3.6.1.5.5.7.3.4),...
CRL Distribution Points	[1]CRL Distribution Point: Distrib...
Thumbprint algorithm	sha1
Thumbprint	0377 5F23 643E ECBB 0077 8...

Digital Signature, Key Encipherment(a0)

Frequently Asked Questions and Answers (FAQs)

How do I know which version of java (JRE) is on my machine?

You can determine which version of Sun Microsystems Java Software (JRE) is on your system by executing the command “**java - version**” at the command prompt. The output indicates the version of java on your machine, e.g. a version of java 1.4 is seen in the example below.

Example:

```
=========================================================
C:\>java -version
java version "1.4.2_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_0
Java HotSpot(TM) Client VM (build 1.4.2_03-b02, mixed mode)
==========================================================
```

Can the Automated File and Report Delivery (AFRD) Utility run on the same computer on which CitiDirect® Online Banking runs? I cannot download CitiDirect program files (.cabs/.jars) after installing java.

The AFRD Utility (S/MIME tool) can be installed on the same computer as CitiDirect. If you are having problems downloading CitiDirect .cabs/.jar files after installation of java (e.g. your screen is frozen at “**Please Wait While We Check for updates…”),** please follow these steps:

1. Close the CitiDirect Online Banking window.
2. Open an Internet Explorer window.
3. From the menu, click Tools.
4. Select Internet Options.
5. Click the Advanced tab.
6. Scroll down to Java(SUN).
7. Uncheck the checkbox “Use java 2 v1.x.x . . .”
8. Click Apply.
9. Click OK.
10. Restart your browser, and sign-on to CitiDirect Online Banking.

How do I obtain the Automated File and Report Delivery Utility (S/MIME Tool)?

A zip version of the AFRD Utility tool for Win32 (Windows Operating system) can be obtained from the CitiDirect Online Banking Web site home page after you sign-on to CitiDirect. You must be a registered user entitled to AFRD in order to access the download site. Alternatively, you can contact your CitiDirect service representative for a zip file, or a **“.tar”** version for UNIX/Linux platforms.

How do I obtain the CitiDirect certificate/public key?

The CitiDirect Online Banking public key can be downloaded from the CitiDirect S/MIME security library.

For more information, please refer to the AFRD documentation in the Learning Center at www.citidirect.com or contact your CitiDirect service representative.

What is the use of the CitiDirect public key?

The CitiDirect public key has a dual use depending on whether you are sending/uploading files to CitiDirect Online Banking or receiving/downloading secured (signed/signed and encrypted) files from CitiDirect.

If you were sending Import files to CitiDirect, this key would be used to encrypt the files for CitiDirect. Files can only be decrypted by using the private key in the CitiDirect system.

If you were receiving (via https or downloading) secured files from CitiDirect, the files would be signed by the CitiDirect private key. The CitiDirect public key would be used to verify the signature on the files received from CitiDirect.

Can I use the CitiDirect public key to encrypt files for other systems?

No. The CitiDirect Online Banking public key can be used only for files transferred between your system and CitiDirect. Files encrypted by this public key can be decrypted using only the private key owned by CitiDirect.

How do I obtain my security profile/certificate? What type of certificate should I obtain?

There are several vendors/Certificate Authorities (CAs) that issue digital certificates. Digital Certificates can be requested from any CA. However, certificates obtained by the client for digital signing must meet the following minimum criteria:

- A valid, (not expired), X.509 signing certificate must be used (email certificate).
- RSA public key (signature) algorithms using key lengths of 1024 bits (or larger).
- Message Digest Algorithm must be SHA1-RSA.
- Encryption algorithm is DES3 (triple DES).
- Key usage must include digital signature (for signing files) and key ???? (for encrypting the files).
- Certificate (Public Key Only) should be exported in the following format for uploaded to CitiDirect “S/MIME Library”:
    - **.p7c** (* PKCS7 cryptographic message syntax standard certificate)
    - **.cer** (DER encoded binary X.509 Certificate)
    - **.cer** (Base64 encoded X.509 Certificate)

Vendors that issue these certificates include Verisign, Entrust, and Thawte. Please contact vendors directly for service and support.

Please refer to *AFRD SMIME Security* section of this Guide or the AFRD documentation on www.citidirect.com for more details.

How do I check the classpath? How do I create a new system variable "CLASSPATH"? Should the existing classpath be deleted? How do I edit the existing classpath?

For Windows® platforms, the classpath can be checked by running the "set classpath" command or by going to system properties in Windows and checking the value for "classpath" under system variables.

If a system variable "classpath" already exists, classpath values specified in this document can be added to the existing values. All values must be separated by a semi colon **(";").**

Please refer the *Setting Up the Classpath* section of this Guide for more information.

Can the AFRD Utility be executed using a batch file instead of the command line? Can I specify parameter values (certificate name, directory location, etc.) in the batch file? If parameter values were specified in the .ini file and batch file/command prompt, which values would take precedence?

The AFRD Utility can be executed from the command line, or via a batch file that can be called through a third program. Parameter values can be added to the batch file after the command, or can be part of the .ini file. If the values are specified in the .ini file and entered after the command in the batch file, or at the command prompt, the tool considers values in the batch file/command prompt and not the values in the .ini file.

Why do I need to copy the security files local_policy.jar and US_export_policy.jar for java (JRE) version 1.4.x to a specific folder? Why is this process not required for java (JRE) version 1.3.x? Can these files be obtained directly from Sun Microsystems®?

These files are required by Sun Microsystems Java Software (JRE) in compliance with U.S. Export Policy. You can find more details on the Sun Microsystems' Web site.

For JRE version 1.3.x, these files are provided with the AFRD Utility tool and pointing to these files in the **classpath** is sufficient. (Please see the *Setting the Classpath for JRE .3.x* section of this Guide for more information.)

For JRE version 1.4.x, the files must be in the specified **Java/j2re1.4.x_x/lib/security** Directory. These files are from Sun Microsystems and can also be downloaded directly from Sun Microsystems' Web site.

Troubleshooting

If you experience any issues in executing the AFRD Utility, the following resources can be used for troubleshooting:

- Encryption and/or Decryption Log Files
- Command-Line Error Messages
- “Exception” Error Messages

These resources are described below.

Note: Additional information on AFRD, such as Training Guides, FAQs and Quick Reference Cards can be found in the Learning Center at www.citidirect.com.

Log Files

The first source for troubleshooting file encryption/decryption processing problems is the log file. Separate logs are generated for the encryption and decryption processes. These log file types are described below.

Encryption Log File

The AFRD Utility generates one encryption log file per day regardless of how many events have been executed. This log file is created in the /SMIMETOOL/RUN/ sub-directory and named using the following convention:

CitiDirectEncryptDecrypt-<Current Date>.txt (e.g., CitiDirectEncryptDecrypt-2003-31-01.txt)

You can archive the log files for audit-tracking purposes or remove them periodically.

In the event of encryption problems, a copy of this log file should be provided to Citigroup for review.

Sample Encryption Log File

Below is an example of the contents of an encryption log file that was created after the successful encryption of a file named Imp-file.txt:

```
=========================================================================
CreateSMime:: New Schedule started at Sat Nov 01 20:31:42 EST 2003

CreateSMime::Sat Nov 01 20:31:45 EST 2003::INFO-> The File
c:/tool/decrypted/imp-file.txt has been successfully processed

CreateMessage::INFO -> The schedule has completed at Sat Nov 01 20:31:45 EST
2003
```

Encryption Log File Messages

A list of encryption log file messages is provided in the table below.

SERIAL NUMBER	MESSAGE	DESCRIPTION
1	<Recipient certificate> is absent	Encryption flag can have either **true or false** as its value if you are using .ini file or **y/n** if you are entering values from the command prompt.
2	Cert absent in path	Encryption set to **True**, Recipient Certificate is Absent.
3	Profile does not exist	Profile <profile name> does not exist or incorrect profile.
4	Password Absent	Profile password not provided.
5	value for Input folder, output folder, logfile is not provided in parameters	Input Folder: <InputFolder> or Output Folder: <Output Folder> or Log Folder: <logfile> not specified.
6	File without extension in input folder path	File pattern is not supported.
7	Input Folder does not exist	Input Folder does not exist - Application will exit.
8	Output Folder does not exist	Output Folder does not exist - Application will exit.
9	Log file is not created	Logger Problem: exception message.
10	Profile file is neither .epf nor .pfx	The Profile format is not valid- should be .pfx,.p12 or .epf.
11	Certificate is not X509format	Invalid Certificate Format.
12	Certificate Expiration	Exception is appended.
13	Signature Exception	Exception is appended.
13	PKCS7 Exception	Exception is appended.
14	Certificate validity	Exception is appended.
15	Key length is not equal to 1024	Error message is appended.
16	Not sha1withRSA	Algorithm not supported exception is appended.
17	Profile expires within 3 months	Profile key update required.
18	Certificate expires within 3 months	Certificate is going to expire within 3 months.

Decryption Log File

The AFRD Utility generates one decryption log file per day regardless of how many events have been executed. This log file is created in the /SMIMETOOL/RUN/ sub-directory and is named using the following convention:

CitiDirectDecryptVerify-<current date>.txt (e.g., SMIMETool-2003-06-25 AD.txt)

You can archive the log files for audit-tracking purposes or remove them periodically. In the event of decryption problems, a copy of this log file should be provided to Citigroup for review.

Sample Decryption Log File

Below is an example of a decryption log file that was created after the successful decryption of a file named Exportqc210935.txt.p7m:

```
=============================================================
ShowSMime:: New Schedule started at 2003-07-25 11:43:49
=============================================================
ShowSMime:: INFO > The File being Processed is
c:/CitiDirectDecryptVerify/Exportqc210935.txt.p7m

ShowSMime::INFO > RFC822 content type in [Message-ID:
<2347637.1058448959802.JavaMail.casaftcqa@casantqa26>
Date: Thu, 17 Jul 2003 09:35:59 -0400 (EDT)
Mime-Version: 1.0
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=Exportqc210935.txt
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=Exportqc210935.txt

MIAGCSqGSIb3DQEHA6CAMIACAQAxggHsMIHzAgEAMFwwVDELMAkGA1UEBhMCVVMxEjAQBg
CUNpdGlncm91cDEgMB4GA1UECxMXQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkxDzANBgNVBAsTBkN
MCBRQQIEOXS1bDANBgkqhkiG9w0BAQEFAASB]

ShowMessage::INFO -> isEnvelopedMimeType [true]

ShowSMime::INFO > Opening mime envelope

ShowSMime::INFO > Mime envelope opened

ShowSMime:: INFO -> encrypted content type [application/pkcs7-mime;
name=smime.p7m]
ShowSMime:: INFO -> encryption algorithm [DES-EDE3-CBC with parameter]
ShowSMime:: INFO -> Recipient information index found at [1]
ShowSMime:: INFO -> content decrypted successfully
ShowSMime:: INFO -> signed content type [application/pkcs7-mime;
name=smime.p7m]
ShowSMime:: INFO -> Signed content extracted successfully
ShowSMime:: SUCCESS -> Signer Certificate Verified

ShowSMime::INFO > Data written to outputstream from Input Stream

ShowSMime::Fri Jul 25 11:43:53 EDT 2003::INFO-> The File
c:/ CitiDirectDecryptVerify /Exportqc210935.txt.p7m has been successfully
processed
=============================================================
```

Decryption Log File Messages

A list of decryption log file messages is provided in the table below.

SERIAL NUMBER	MESSAGE	DESCRIPTION
1	Profile is wrong	ShowSMime:: Wed Mar 17 17:33:42 EST 2004::ERROR-> <epf file name> (The system cannot find the file specified.)
2	Profile does not exist	ShowSMime::Wed Mar 17 17:34:58 EST 2004::ERROR-> The Profile format is not valid- should be .pfx,.p12 or .epf.
3	Any parameter missing	Profile:C:/CERTS/epfNew/citifile2.epf or Password: or Input Folder: C:/TC/decryptiontool/1.3/encrypted/S*.txt.p7m or Output Folder: c:/TC/decryptiontool/1.3/decrypted/ or Log Folder: C:/SMimeLogs/not specified.
4	File without extension in input folder path	File pattern is not supported.
5	Input Folder does not exist	Input Folder does not exist – Application will exit.
6	Output Folder does not exist	Output Folder does not exist - Application will exit.
7	Log file is not created	Logger Problem: exception message.
8	Profile file is neither .epf nor .pfx	Profile<profile>does not exist or incorrect profile.
9	Profile expires within 3 months	Profile key update required.
10	Sender's certificate given	Sender certificate not provided.

Command-Line Error Messages

Manually executing the .BAT file (DecryptandVerify.bat or SignandEncrypt.bat) from the command line from within the /SMIMETOOL/RUN/ sub-directory may provide additional messages/errors that are not recorded in the log file.

A list of possible error messages is provided in the table below.

ERROR MESSAGE	DESCRIPTION/CORRECTIVE ACTION
"ERROR> Incorrect password! File is not placed in the outbound folder"	• The Profile (Personal Digital Certificate) password was input incorrectly in the ENCRYPT.INI file. • Check the password value in the .INI file.
"C:\*.EPF (The filename, directory name, or volume label syntax is incorrect)"	• An invalid value for the Profile was provided in the ENCRYPT.INI file. • Verify the full filepath (including the filename) location of the import file. • Verify an import file exists in the specified location.
"Input Folder does not exist – Application will exit"	• The values in the ENCRYPT.INI file contain a wild card (Example: *.txt) in the INBOUND folder filename.
"Output Folder does not exist – Application will exit"	• The values in the ENCRYPT.INI file contain an OUTBOUND folder directory structure that does not exist. • Verify the existence and full path to the OUTBOUND folder.
"Input Folder does not exist – Application will exit"	• The values in the ENCRYPT.INI file contain an INBOUND folder directory structure that does not exist. • Verify the existence and full path to the INBOUND folder.

If the error displayed is not listed in the table above, please refer to the **Common Exception Errors Related to Setup Problems** table on the following page.

Common Exception Errors Related to Setup Problems

The table below contains a list of common exception errors that can be displayed as a result of incorrect setup, the probable cause of the errors, and corrective actions that can be taken. For additional information, please refer to the *Setting Up the Classpath* section of this Guide.

Error Description	Probable Cause	Corrective Actions

C:\>java CreateSMime
 	
Exception in thread "main" **java.lang.NoClassDefFoundError**: CreateSMime
	Command line instruction to encrypt/decrypt files (“java CreateSMime” / java “ShowSMime”) has not been executed from the directory in which it has been installed.	Change directory to where the CreateSMime class resides.
		
		(DOS window Example:
		C:\> cd C:\SMimeTool\Run
		C:\SMimeTool\Run>java CreateSMime)

C:\SMimeTool\Run>java CreateSMime Exception in thread "main" java.lang.NoClassDefFoundError: **javax/activation/**Data Source (or any error starting with java.lang.**NoClassDefFoundError: javax/……**	**“Classpath”** does not include activation.jar (or other jar/s specified after javax/ in error message).	Ensure that: - **classpath** has been set correctly - All values are separated with a semicolon **“;”** - Last value in the classpath has **“ ; ”** at the end of the value.
C:\SMimeTool\Run>java CreateSMime java.lang.Exception: CreateSMime::Fri Jan 14 16:47:56 EST 2005::ERROR-> File(s) not found ->CreateSMime::Fri Jan 14 16:47:56 EST 2005::ERROR-> The File C:/ENCRYPTION_FILES/encrypt.txt could not be processed - javax.mail.MessagingException: java.lang.SecurityException: **Unsupported keysize or algorithm parameters** at CreateSMime.createSMimeMessage(CreateSMime.java:823) ………	Jre/java version installed is 1.4.x and the latest **local_policy.jar,** and **US_export_policy.jar** have not been copied to the Java/ j2re 1.x.x/security/lib folder.	Copy **local_policy.jar,** and **US_export_policy.jar** from SMimeTool\jar\ext\ to …\java\j2re1_x_x\lib\security\ folder
C:\SMimeTool\Run>java CreateSMime com.entrust.toolkit.exceptions.UserFatalException: Incorrect password. at com.entrust.toolkit.credentials.q.b(Unknown Source)………	Password provided for your certificate in .ini file is incorrect .	Check the value for password in the .ini file. Ensure this is the correct password for the security Profile/Certificate in the .ini file.

Error Description	Probable Cause	Corrective Actions
C:\SMimeTool\Run> java CreateSMime java.lang.Exception: Encryption set to true, **Recipient CertificateC:/entrust/Citidirect.p7cis absent** at CreateSMime.init(CreateSMime.java:277) at CreateSMime.<init>(CreateSMime.java:314) at CreateSMime.main(CreateSMime.java:1043)	1- Certificate/public key to encrypt the file is missing, or the value in the .ini file for **“RcptCert”** is incorrect. 2- A back slash **“\”**instead of forward slash **“/”** has been used to specify the location for the profile.	1- Check the value for **“RcptCert.”** Verify the public certificate (Downloaded from CitiDirect) is present at the specified location. 2- Ensure a forward slash **“/”** (not a back slash **“\”**) has been used to specify the location for the profile.
C:\SMimeTool\Run>java CreateSMime java.lang.Exception: Profile: c:/security/profile/**myprofile.pfx does not exist or wrong profile** at …..	1- Security Profile mentioned in the “**.ini”** file cannot be located. 2- A back slash **“\”**instead of forward slash **“/”** has been used to specify location for the profile.	1- Ensure the value for “Profile” parameter in the .ini file. Ensure the profile/certificate with correct name exists in the specified location. 2- Ensure a forward slash **“/”** (not a back slash **“\”**) has been used to specify location for the profile.
C:\SMimeTool\Run>java CreateSMime iaik.pkcs.PKCSParsingException: ASN.1 creation error:ASN.1 creation error:No known implementation for ASN.1 Type [7,Decoded value,UNIVERSAL] at iaik.pkcs.pkcs12.PKCS12.<init>(Unknown Source)………. at CreateSMime.login(CreateSMime.java:465) at ……	Security Profile/Certificate has either been corrupted, or is not accessible for other reasons.	Contact your security Profile issuer (e.g., Verisign/Entrust) to refresh the certificate.
C:\SMimeTool\Run>java CreateSMime java.lang.Exception:….. …….::ERROR-> The File C:/SMimeTool/TO_BE_ENCRYPTED_FILES/FILE_for_citi.txt could not be processed - CreateSMime::Mon Jan 17 21:50:37 EST 2005::ERROR-> in Creating EncryptedContent - null at ……	Folder specified for log files is not correct/folder does not exist. Encryption certificate. (CitiDirect certificate does not exist or is invalid.) 2- A back slash **“\”** instead of forward slash **“/”** has been used to specify the location for the profile.	Check value for “**LogPath”** in .ini fie. Ensure the folder exists at the specified location. 2- Ensure a forward slash **“/”** (not a back slash **“\”**) has been used to specify the location for the profile.

Error Description	Probable Cause	Corrective Actions
C:\SMimeTool\Run>java CreateSMime java.io.FileNotFoundException: C:\SMimeTool\TOOL_LOG_FILES\CitiDirectSignEncrypt-01-19-05.txt **(Access is denied**) at java.io.FileOutputStream.openAppend(Native Method) at java.io.FileOutputStream.<init>(Unknown Source) at java.io.FileOutputStream.<init>(Unknown Source) at ….. CreateSMime.main(CreateSMime.java:1043) java.lang.Exception: null Logger Problem::null at …	1- Folder specified for log files is not correct/folder does not exist.	1- Check value for “**LogPath”** in the .ini file. Ensure the folder exists at the specified location.
C:\SMimeTool\Run>java CreateSMime java.lang.Exception: Log Folder not specified or missing '/' at the end of path at CreateSMime.init(CreateSMime.java:296) at CreateSMime.<init>(CreateSMime.java:314) at CreateSMime.main(CreateSMime.java:1043)	Value for log files folder is not correct/folder does not exist.	Check value for “**LogPath”** in the .ini file. Ensure the folder exists at the specified location.
C:\SMimeTool\Run>java CreateSMime java.lang.Exception: CreateSMime::Mon Jan 17 21:36:37 EST 2005::ERROR-> null at CreateSMime.checkCertificate(CreateSMime.java:535) at CreateSMime.<init>(CreateSMime.java:337) at CreateSMime.main(CreateSMime.java:1043)	Encryption Certificate (CitiDirect certificate) is invalid/missing, or inaccessible.	Ensure value for **“RcptCert”** in the .ini file (CitiDirect public certificate) is correct. Re-download public key from “CitiDirect S/MIME Library” and replace the existing key with the new key.
Usage: CreateSMime [-options] [args...] where options include: -e the profile -p the password -i the file input folder -o the file output folder -c the certificate -L the logfolder y/n 'y' to encypt, 'n' to sign (no encryption)	Parameters for folder locations etc are not in quotes (**“**). OR One, or more parameters are missing. (Instruction to run the tool has been issued from command line in this case.)	Provide all file location parameters in double quotes.

Disclaimer

The authoritative and official text of this CitiDirect® Online Banking documentation shall be in the English language as used in the United States of America. Any translation of any CitiDirect documentation from English to another language is done solely for the convenience of the reader, and any inconsistencies, or inaccuracies between the English text and that translation shall be resolved in favor of the English text.

These materials are proprietary and confidential to Citibank, N.A., and are intended for the exclusive use of CitiDirect Online Banking customers. The foregoing statement shall appear on all copies of these materials made by you in whatever form and by whatever means, electronic or mechanical, including photocopying or in any information storage system. In addition, no copy of these materials shall be disclosed to third parties without express written authorization of Citibank, N.A.

Customer shall be solely responsible for the use of any User identifications, passwords and authentication codes that may be provided to it, from time to time, in connection with CitiDirect Online Banking (collectively, "User IDs"). Customer agrees to keep all User IDs strictly confidential at all times. Customer shall immediately cease use of CitiDirect Online Banking if it receives notification from Citibank, or otherwise becomes aware of, or suspects, a technical failure or security breach. Customer shall immediately notify Citibank if it becomes aware of, or suspects, a technical failure or security breach.

December 2005