



ANTEI FTP API 3.50

User Manual

Content

| | |
|--|----|
| Introduction | 3 |
| What is new in version 3.50..... | 4 |
| What is new in version 3.01..... | 4 |
| What is new in version 2.00..... | 4 |
| Installation | 5 |
| FTP..... | 7 |
| Uploading files to ftp location..... | 8 |
| Downloading files from FTP Location | 9 |
| Create the directory..... | 10 |
| Remove the directory | 11 |
| Delete the file..... | 12 |
| List the folder | 13 |
| SFTP..... | 15 |
| Uploading files to SFTP location..... | 15 |
| Downloading files from SFTP location | 17 |
| Create the directory..... | 18 |
| Remove the directory | 19 |
| Delete the file..... | 20 |
| List the folder | 21 |
| FAQ..... | 23 |

Introduction

The ANTEI FTP API 3.50 provides a secure, simple means of transferring files to and from ftp servers. ANTEI FTP API is a COM control which allows easy integration with any ERP system.

ANTEI FTP API is generally accessible to private and commercial users with one off charge £34.99 per machine. ANTEI FTP API Demo version is restricted to the maximum file size of 10kB. Demo version does not require any fee-based subscriptions or other fee-based restricted access paid to ANTEI Computers Ltd.

Should you have any questions please visit our website <http://www.antei-computers.com> or contact us by email: office@antei-computers.com.

What is new in version 3.50

- SSH FTP
 - Multi port support.
 - ListDirectory returns file modification time and date
 - Improved error handling

What is new in version 3.01

- FTP and SFTP split to two separate classes
- FTP
 - New function CreateDirectory – creates folder in the ftp location
 - New function ListDirectory – returns the list of files and subfolders in the directory
 - New function DeleteFile – deletes the file from the ftp folder
 - New function DeleteDirectory – deleted the folder from the ftp location
- SSH FTP
 - Support of certificates larger than 1024 b.
 - New function CreateDirectory - creates folder in the ftp location
 - New function DeleteFile – deletes the file from the ftp folder
 - New function DeleteDirectory – deleted the folder from the ftp location
 - New function ListDirectory – returns the list of files and subfolders in the directory

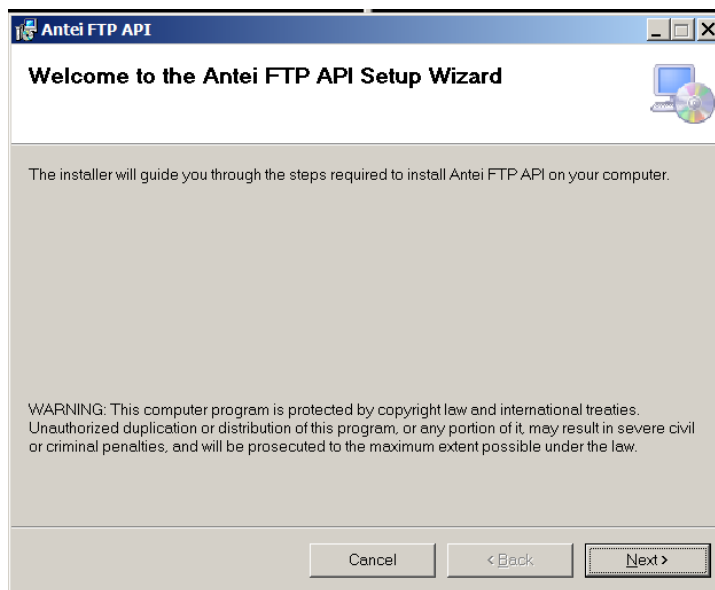
What is new in version 2.00

- Passive and active modes
- Supports SFTP protocol including both certificate and non-certificate connections

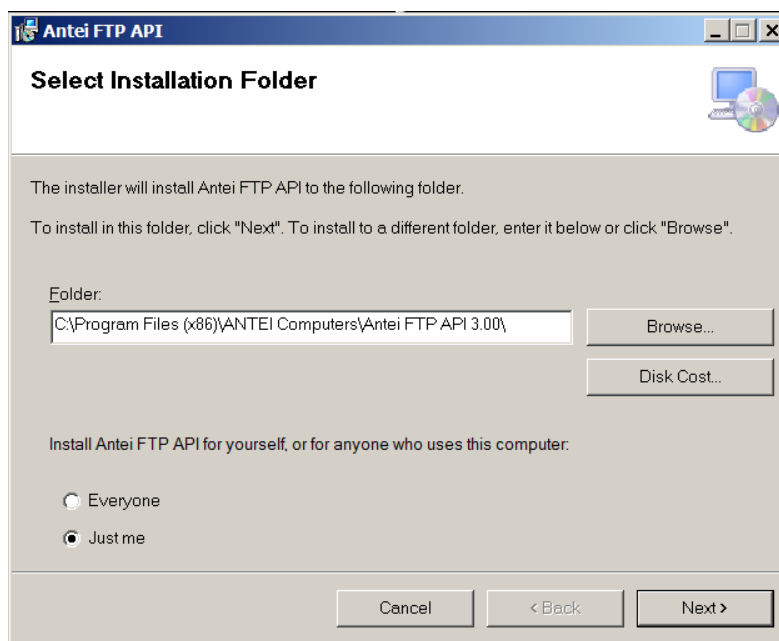
Installation

Please, follow the steps to install the ANTEI FTP API 3.50.

1. If you have previous versions of ANTEI FTP API installed please remove it. In order to do this run setup.exe and select Remove option or uninstall from *Control Panel, Programs and Features*.
2. Extract setup files from 'ANTEI FTP API.zip' file
3. Run Setup.exe and press Next



4. Change or confirm destination folder, user option and press Next



5. Press Next
6. Confirm EULA Licence and press Next

To uninstall or repair ANTEI FTP API 3.50 please run installation wizard and select Remove or Repair option.

In case of failure please follow the steps:

1. Run Command Prompt as Administrator
2. Navigate to installation directory, by default it is "c:\Program Files\ANTEI Computers\Antei FTP API 3.50\" or "c:\Program Files (x86)\ANTEI Computers\Antei FTP API 3.50\"
3. Run the following command to register the dlls e.g.:
`<path_to_the_ftp_api>\dllreg.exe <path_to_the_ftp_api>\AnteiFTPAPI.dll`

E.g.

"c:\Program Files\ANTEI Computers\Antei FTP API 3.50\dllreg.exe" "c:\Program Files\ANTEI Computers\Antei FTP API 3.50\ AnteiFTPAPI.dll"

4. Run the following command to unregister the dll:
`<path_to_the_ftp_api >\dllunreg.exe <path_to_the_ftp_api>\ AnteiFTPAPI.dll`

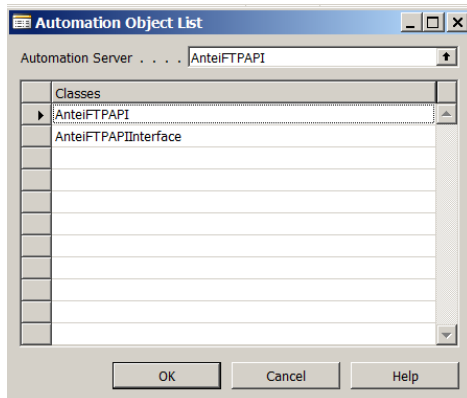
E.g.

"c:\Program Files\ANTEI Computers\Antei FTP API 3.50\dllunreg.exe" "c:\Program Files\ANTEI Computers\Antei FTP API 3.50\AnteiFTPAPI.dll"

The same steps need to be repeated for AnteiSFTPAPI.dll.

FTP

To use FTP API please add the reference to 'AnteiFTPAPI'.AnteiFTPAPI class.



Uploading files to ftp location

Use **UploadToFTP** function to upload a file to the FTP. This function does not support SFTP.

```
ErrorMsg = AnteiFTPAPI.UploadToFTP(FTPAddress, UserName, Password, SourceFileFolder,
FileName, Passive);
```

| Description | Field Type | Comment |
|------------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| SourceFileFolder | string | Path to the file location |
| FileName | string | Name of the file to be uploaded |
| Passive | bool | True – passive mode False – active mode |
| ErrorMsg | string | Empty string if the upload has been completed successfully. Otherwise detailed error message. |

Downloading files from FTP Location

Use **DownloadFromFTP** function to download the file from the ftp site. This function does not support SFTP.

```
ErrorMsg = AnteiFTPAPI.DownloadFromFTP(FTPAddress, UserName, Password, SourceFileName,  
TargetFileName, Passive);
```

| Description | Field Type | Comment |
|----------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| SourceFileName | string | Name of the file to be downloaded |
| TargetFileName | string | Path and the file name |
| Passive | bool | True – passive mode False – active mode |
| ErrorMsg | string | Empty string if the download has been completed successfully. Otherwise detailed error message. |

Create the directory

Function **CreateDirectory** adds new directory.

```
ErrorMsg = AnteiFTPAPI.CreateDirectory(FTPAddress, UserName, Password, DirName, Passive);
```

| Description | Field Type | Comment |
|-------------|------------|--|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| DirName | string | Name of the directory to be created |
| Passive | bool | True – passive mode False – Active mode |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |

Remove the directory

Function **DeleteDirectory** removes the existing directory.

```
ErrorMsg = AnteiFTPAPI.DeleteDirectory(FTPAddress, UserName, Password, DirName, Passive);
```

| Description | Field Type | Comment |
|-------------|------------|--|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| DirName | string | Name of the directory to be deleted |
| Passive | bool | True – passive mode False – Active mode |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |

Delete the file

Function **DeleteFile** removes the file from the given folder.

```
ErrorMsg = AnteiFTPAPI.DeleteFile(FTPAddress, UserName, Password, FileName, Passive);
```

| Description | Field Type | Comment |
|-------------|------------|--|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| FileName | string | Name of the file to be deleted |
| Passive | bool | True – passive mode False – Activ mode |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |

List the folder

Function **ListDirectory** returns the list of files and subdirectories in the given location. Data is returned in xml format. The function returns only files and folders in the current directory. It does not check subfolders. To check the subfolders a separate call of ListDirectory function must be made.

```
ErrorMsg = AnteiFTPAPI.ListDirectory(FTPAddress, UserName, Password, Passive, OutputFileName);
```

| Description | Field Type | Comment |
|----------------|------------|--|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| Passive | Bool | True – passive mode False – Active mode |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |
| OutputFileName | string | Name of the file with xml output |

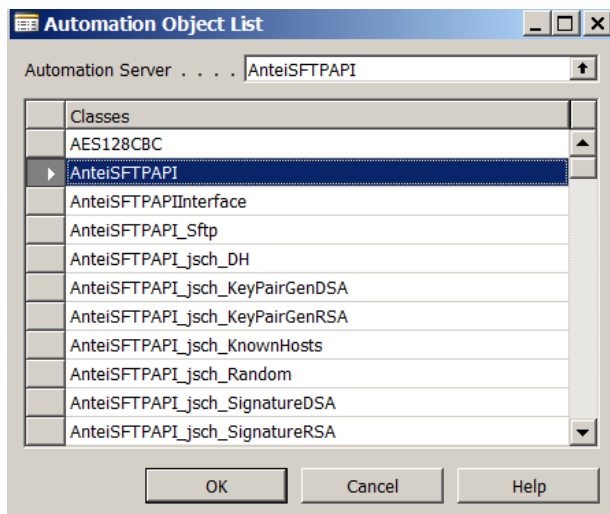
Sample xml output generated by ListDirectory function:

```
<?xml version="1.0" encoding="utf-8"?>
<Structure>
  <Detail>
    <URL>ftp://ftp.com</URL>
    <Type>file</Type>
    <Name>test2.txt</Name>
    <DateTime></DateTime>
  </Detail>
  <Detail>
    <URL>ftp://ftp.com</URL>
    <Type>file</Type>
    <Name>test.xml</Name>
    <DateTime></DateTime>
  </Detail>
  <Detail>
    <URL>ftp://ftp.com </URL>
    <Type>dir</Type>
    <Name>nav2</Name>
    <DateTime></DateTime>
  </Detail>
  <Detail>
    <URL>ftp://ftp.com </URL>
    <Type>file</Type>
```

```
<Name>du.xlsx</Name>  
<DateTime></DateTime>  
</Detail>  
</Structure>
```

SFTP

To use SFTP API please add the reference to 'AnteiSFTPAPI'.AnteiSFTPAPI class.



ANTEI SFTP API supports Open SSH certificates only. Please, read FAQ section to learn how to convert a commercial SSH certificate to Open SSH format.

Uploading files to SFTP location

Use **UploadToSFTP** function to upload a file to the Secure FTP.

ErrorMsg = AnteiFTPAPI.UploadToSFTP(FTPAddress, Port, UserName, Password, SourceFile, TargetFileName, CertFile);

| Description | Field Type | Comment |
|----------------|------------|---|
| FTPAddress | String | Address of the ftp site. Eg.: ftp://site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | String | Name of the user authorised to upload files |
| Password | String | User password or empty string if your server uses certificates. |
| SourceFile | String | Path and name of the file to be uploaded |
| TargetFileName | String | Destination folder and file name |
| CertFile | String | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |
| ErrorMsg | String | Empty string if the upload has been completed successfully. Otherwise detailed error message. |

Example:

The example uploads test.txt file from main folder on c drive to directory 'dir'. The ftp server in below example does not require a certificate and uses SFTP protocol.

```
AnteiFTPAPI.UploadToSFTP('server.domain', 0, 'user_name', 'password', 'c:/test.txt', '/dir/test.txt',  
");
```


Downloading files from SFTP location

Use **DownloadFromSFTP** function to download the file from the Secure FTP site.

```
ErrorMsg = AnteiFTPAPI.DownloadFromSFTP(FTPAddress, Port, UserName, Password,  
SourceFileName, TargetFileName, CertFile);
```

| Description | Field Type | Comment |
|----------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: ftp://site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password. Empty when certificate is used. |
| SourceFileName | string | Name of the file to be downloaded |
| TargetFileName | string | Path and the file name |
| CertFile | string | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |
| ErrorMsg | string | Empty string if the download has been completed successfully. Otherwise detailed error message. |

Example:

The example downloads test.txt file from 'dir' folder and saves it in FTPFile folder. The ftp server in below example does not require a certificate and uses SFTP protocol.

```
AnteiFTPAPI.DownloadFromSFTP('server.domain', 0, 'user_name', 'password', 'dir/test.txt',  
'f:\FTPFile\');
```

Create the directory

Function **CreateDirectory** adds new directory.

```
ErrorMsg = AnteiSFTPAPI.CreateDirectory(FTPAddress, Port, UserName, Password, DirName, CertFile);
```

| Description | Field Type | Comment |
|-------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password. Empty when certificate is used. |
| DirName | string | Name of the directory to be created |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |
| CertFile | string | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |

Remove the directory

Function **DeleteDirectory** removes the existing directory.

```
ErrorMsg = AnteiSFTPAPI.DeleteDirectory(FTPAddress, Port, UserName, Password, DirName, CertFile);
```

| Description | Field Type | Comment |
|-------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| DirName | string | Name of the directory to be deleted |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |
| CertFile | string | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |

Delete the file

Function **DeleteFile** removes the file from the given folder.

```
ErrorMsg = AnteiSFTPAPI.DeleteFile(FTPAddress, Port, UserName, Password, FileName, CertFile);
```

| Description | Field Type | Comment |
|-------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| FileName | string | Name of the file to be deleted |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |
| CertFile | string | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |

List the folder

Function **ListDirectory** returns the list of files and subdirectories in the given location. Data is returned in xml format. The function returns only files and folders in the current directory. It does not check subfolders. To check the subfolders a separate call of ListDirectory function must be made.

```
ErrorMsg = AnteiSFTPAPI.ListDirectory(FTPAddress, Port, UserName, Password, OutputFileName, CertFile);
```

| Description | Field Type | Comment |
|----------------|------------|---|
| FTPAddress | string | Address of the ftp site. Eg.: site.com |
| Port | Int | Number of port, if zero then default value 22 will be used. |
| UserName | string | Name of the user authorised to upload files |
| Password | string | User password |
| ErrorMsg | string | Empty string if the operation has been completed successfully. Otherwise detailed error message. |
| OutputFileName | string | Name of the file with xml output |
| CertFile | string | Path and file name with certificate. The certificate file needs to be in OpenSSH format (see FAQ section how to convert a certificate to OpenSSH). Use empty string if your server does not require certificates. |

Sample xml output generated by ListDirectory function:

```
<?xml version="1.0" encoding="utf-8"?>
<Structure>
  <Detail>
    <URL>ftp://ftp.com</URL>
    <Type>file</Type>
    <Name>test2.txt</Name>
    <DateTime></DateTime>
  </Detail>
  <Detail>
    <URL>ftp://ftp.com</URL>
    <Type>file</Type>
    <Name>test.xml</Name>
    <DateTime></DateTime>
  </Detail>
  <Detail>
    <URL>ftp://ftp.com </URL>
    <Type>dir</Type>
    <Name>nav2</Name>
```

```
<DateTime></DateTime>
</Detail>
<Detail>
  <URL>ftp://ftp.com </URL>
  <Type>file</Type>
  <Name>du.xlsx</Name>
  <DateTime></DateTime>
</Detail>
</Structure>
```

FAQ

1. Why do I receive an error message that the control has not been activated in the Classic client?

If the COM component is going to be used in NAV classic client please make sure User Account Control has been deactivated. The RTC client asks if you want to allow connection to the FTP API. Classic client does not have this availability.

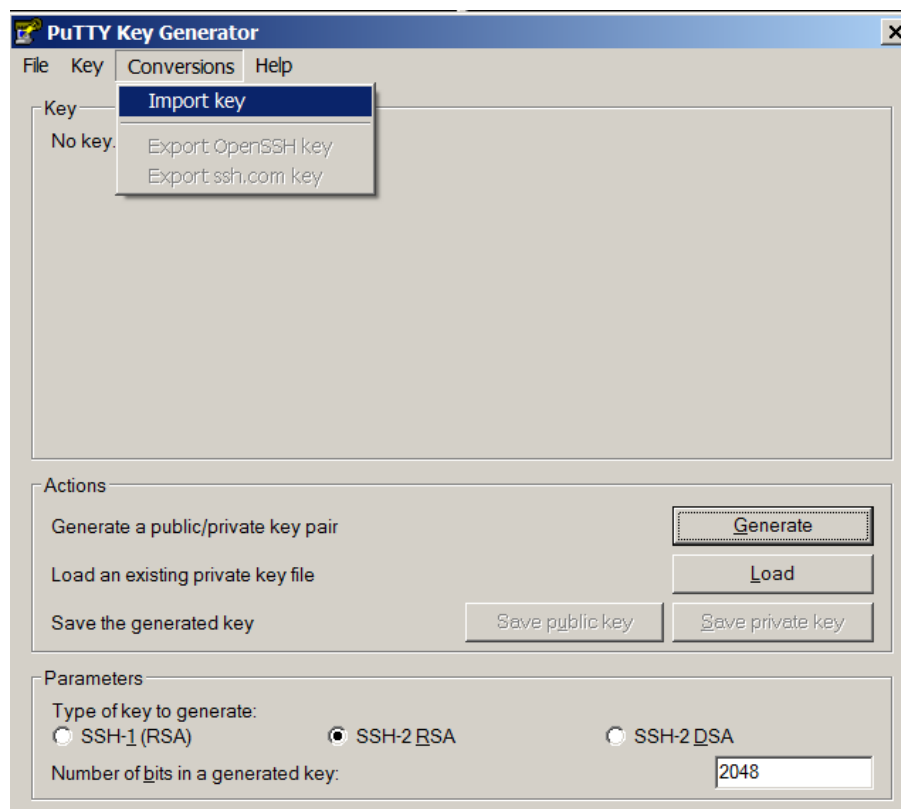
2. Why do I receive an error message that the control has not been activated when connected to NAV via Citrix Remote Desktop?

In some Citrix installations local resources are used (e.g. local desktop, printers, etc). In this case the ANTEI FTP API needs to be installed on the local machine as well.

3. How can I convert a commercial certificate to Open SSH?

There is plenty of software available on the market to convert SSH certificates to different format. Below example shows how to use freeware PuttyGen.

Import your key:



Having imported the key press "Export OpenSSH key in menu Conversion.