EUROPEAN ORGANISATION
FOR THE SAFETY OF AIR NAVIGATION

EUROCONTROL

Electronic AIP Developer's Manual

Edition Number	:	1.0.3
Edition Date	:	16 FEB 2004
Status	:	Released Issue
Intended for	:	General Public

EUROPEAN AIR TRAFFIC MANAGEMENT

DOCUMENT CHARACTERISTICS

TITLE

Electronic AIP Developer's Manual

	EATMP Infocentre Reference:	040216-04
Document Identifier	**Edition Number:**	1.0.3
	Edition Date:	16 FEB 2004

Abstract

The Electronic AIP (eAIP) Specification developed by EUROCONTROL provides a standard way to:

- publish the content of an AIP (including AMDT, SUP and AIC) in a structured electronic format;
- visualise the content of an AIP on a computer screen, using Web technology.

This manual is complementing the eAIP Specification by providing detailed instructions for eAIP system developers, implementers and IT departments. It provides documentation for the XSLT stylesheets and covers topics such publishing an eAIP, using the tools provided by EUROCONTROL (makeAIP.bat) for generating HTML and PDF versions, setting-up a digital signing environment for the eAIP.

Keywords

Electronic AIP	XML	integrity
AIP	DTD	security
eAIP	stylesheet	
AIS	paperless	

Contact Person(s)	Tel	Unit
Benoit Maisonny	93067	
Eduard Porosnicu	93326	

STATUS, AUDIENCE AND ACCESSIBILITY

Status		Intended for		Accessible via	
Working Draft	☐	General Public	☑	Intranet	☐
Draft	☐	EATMP Stakeholders	☐	Extranet	☐
Proposed Issue	☐	Restricted Audience	☐	Internet (www.eurocontrol.int)	☑
Released Issue	☑	*Printed & electronic copies of the document can be obtained from the EATMP Infocentre (see page iii)*			

ELECTRONIC SOURCE

Path:	U:\AHEAD\PROJECTS\P4- e-AIP\eAIP ROLL OUT\Sources\formal	
Host System	Software	Size
Windows_NT	Microsoft Word 10.0	742 Kb

EATMP Infocentre
EUROCONTROL Headquarters
96 Rue de la Fusée
B-1130 BRUSSELS

Tel: +32 (0)2 729 51 51
Fax: +32 (0)2 729 99 84
E-mail: eatmp.infocentre@eurocontrol.int

Open on 08:00 - 15:00 UTC from Monday to Thursday, incl.

DOCUMENT APPROVAL

The following table identifies all management authorities who have successively approved the present issue of this document.

AUTHORITY	NAME AND SIGNATURE	DATE
Author	Benoit MAISONNY [illegible]	16 FEB 2004
Paperless AIS Project Manager	Eduard POROSNICU [illegible]	16 FEB 2004
AIS AHEAD Programme Manager	Conrad CLEASBY Conrad Cleasby	16 FEB 2004

DOCUMENT CHANGE RECORD

The following table records the complete history of the successive editions of the present document.

EDITION NUMBER	EDITION DATE	INFOCENTRE REFERENCE	REASON FOR CHANGE	PAGES AFFECTED
1.0.3	16 FEB 2004		First release	All

Table of Contents

Executive Summary

The Electronic AIP (eAIP) Specification developed by EUROCONTROL provides a standard way to:

- publish the content of an AIP (including AMDT, SUP and AIC) in a structured electronic format;
- visualise the content of an AIP on a computer screen, using Web technology.

This manual is complementing the eAIP Specification by providing detailed instructions for eAIP system developers and implementers in several technical areas. Technical people, who support eAIP Editors in producing eAIPs in HTML, XSL-FO and PDF formats, will find detailed usage instructions for generating HTML, XSL-FO and PDF formats. This includes instructions for using the "MakeAIP" demonstration (proof of concept) tool provided by EUROCONTROL. Also included are explanations on technological and procedural choices, which support the eAIP Security guidelines included in the eAIP Specification.

Chapter 1. Introduction

This manual is complementing the eAIP Specification by providing detailed instructions for eAIP system developers and implementers in several technical areas.

Many parts of this manual refer to the "MakeAIP" provided by EUROCONTROL. However, the MakeAIP tool provided by EUROCONTROL shall be considered as a proof of concept only. No effort was spent on fine-tuning the tool for better performance. There was no formal validation (software audit) of the HTML and PDF output, to prove that it corresponds at 100% to the content of the XML files. The tool should be revisited, improved and validated before being used in a production environment.

Chapter 2. eAIP Publication Process

2.1. Introduction

The content of this chapter is aimed at people who support eAIP Editors in producing eAIPs in HTML, XSL-FO and PDF formats. The publication process contained in the eAIP Specification is a good introduction to this document.

2.2. Production of HTML, XSL-FO and PDF formats

1. Open a terminal or command prompt window ("MS-DOS")
2. Change the current directory to where you unzipped the eAIP package (for example on Windows, type `cd C:\path\to\eAIP-pack`)
3. Transform the eAIP file into HTML using MakeAIP.bat (read the MakeAIP User Manual first) Usage examples:

   - simply type "makeaip" to display usage information
   - type "makeaip sections -f html" to generate all HTML files for the sections
   - type "makeaip GEN-0.1 -f html" to generate only the HTML file for the GEN-0.1 section
   - type "makeaip AD-2.EADD -f html" to generate only the HTML file for the EADD aerodrome
   - type "makeaip menu" to generate the navigation menu (ToC window)

4. Transform the eAIP file into XSL-FO by using the same commands as above, but replacing "-f html" by "-f fo"
5. Transform XSL-FO files to XEP's AT (Area Tree) by using the same commands as above, but replacing "-f html" by "-f at".
6. Transform AT files to PDF by using the same commands as above, but replacing "-f html" by "-f pdf".
7. Delete any section file in HTML which is "NIL", unless it becomes NIL in this amendment
8. Delete any section file in PDF which is "NIL" or becomes NIL in this amendment

Instead of makeaip.bat (which only runs on Windows), you can use your favourite XSLT processor and XSL-FO renderer directly.

The XSLT processing takes about 10 seconds per file, using a Pentium III processor at 866MHz with 128MB RAM.

2.2.1. How do I get a paper version of my eAIP?

1. Generate the PDF files
2. Print them.

MakeAIP does not contain any batch script to automate this yet.

2.2.2. Special procedure for GEN 0.4 (checklist of pages) in PDF format

GEN-0.4 needs AT files (XEP's Area Tree format) for all sections in order to be generated, including for itself. This creates an interesting chicken-and-egg issue: AT format are produced using FO format, but GEN-0.4.fo needs GEN-0.4.at. To solve it, we have a special procedure for section GEN-0.4:

1. Generate all sections in AT format (this will produce an empty GEN-0.4):

   a. `makeaip sections -f fo`
   b. `makeaip sections -f at`

2. Generate a better version of GEN-0.4 in AT format using MakeAIP's checklist command. Note that the checklist will include pages from the empty GEN-0.4 file.

   a. `makeaip checklist -f fo -s URI`
   b. `makeaip GEN-0.4 -f at`

   In this example, the URI would be "../at", because "at" is the default directory where AT files are built and the URI is relative to the default "fo" directory where FO files are built.
3. Generate GEN-0.4 AT format again, in the same way. This time, it will include the better GEN-0.4 pages, but the number of pages of GEN-0.4 may have changed (because the number of listed pages of GEN-0.4 may have increased, pushing other listed pages and eventually creating a new page for GEN-0.4). If so, you must generate GEN-0.4 once again, and check once again if the number of pages of GEN-0.4 has changed (at this time, it is very unlikely).
4. Finally, you can generate the PDF format: `makeaip GEN-0.4 -f pdf`
5. Repeat all this for each language of your eAIP.

For a concrete example, please read the Amendment Procedures, section "Paper eAIP".

2.2.3. Special procedure for "0.6" sections (tables of contents) in PDF format

Note

MakeAIP does not contain a complete tool to automate this procedure; it might be possible to provide it in future.

To have a correct table of contents in the 0.6 sections, a whole eAIP Part must be generated in a single FO and then AT file. Then, the AT format is used to move correct 0.6 pages from the whole Part file to the 0.6 section file. Only after that can we build the PDF format.

1. `makeaip GEN -f fo`
2. `makeaip GEN -f at`
3. Open the GEN Part AT file (preferably with an XML editor); locate the pages (xep:page elements) of section 0.6 (hint: count how many pages the previous sections are made of, using their AT or PDF format) and copy them into the GEN-0.6 section file, replacing existing pages
4. `makeaip GEN-0.6 -f pdf`
5. Repeat for ENR and AD

2.3. Internationalisation and Localisation

2.3.1. Locales file structure

The locales file is written in a simple XML format. Each translated item is represented by a text element with a unique id attribute. A text element contains at least one translation element, which contains the actual text of the translation. The translation element has an xml:lang attribute, which indicates the language tag of the translation. A text element may contain an optional note element to give information to translators.

2.3.2. Implementation of Internationalisation

When a style sheet must generate static text, it calls the "gettext" template, which is defined in xslt/g-gettext.xslt. This template takes a mandatory parameter "ID", containing the id of the text element to be output. An optional "c1" parameter allows to specify content that might be put at a different place in a translation item, depending on the language.

The "gettext" template looks in the locales file for a text element with the right id and, within it, for a translation element with the right language tag, as explained above.

2.3.3. Customising the eAIP menu

The eAIP menu should be customised to offer links to other language version. If no other language version exist, then the relevant lines in the file xslt/html-content.xslt can simply be removed.

The links to other language versions are quite easy to devise: depending on your files organisation, you can build the other file names by simply changing the language suffix. There is not (yet) any way to find out automatically what other languages are available for a given eAIP.

2.4. Software Requirements

Basically, all you need is an XSLT processor and an XSL-FO renderer. However, most of such software are developed in Java, so you will probably also need a Java Runtime Environment. Optionally, you may want to configure MakeAIP.bat.

2.4.1. XSLT processor

An XSLT processor is a software that transforms XML files into other formats, such as HTML and XSL-FO, using XSLT instructions. For more information about XSLT, see the references section of the eAIP Specification.

Famous XSLT processors include:

- Saxon [http://saxon.sourceforge.net], an Open Source Software written in Java, by Michael Key (recommended);
- Xalan [http://xml.apache.org/xalan-j/index.html], another Open Source Software written in Java by the Apache Software Foundation;
- MSXML [http://msdn.microsoft.com/xml], a closed source implementation by Microsoft, available free of charge as well.

There are many other processors available. Because of MSXML's particular implementation of namespaces, we do NOT recommend it, unless you are an XML expert and you know what you are doing. Note that MSXML only runs on Windows platforms, while Xalan runs on any Java 2 platform (Windows, Unix, Mac and others).

If you are using MSXML, check what version you want to use and specify this in the MakeAIP file (read MakeAIP.html). This allows you to use any version of MSXML that you have installed, even in "side-by-side mode". Please note that Internet Explorer will only use the version that is in "replace mode". See http://www.netcrucible.com/xslt/msxml-faq.htm for more details.

The "eAIP Splitter", one of the tools provided in the eAIP package, only works with Xalan. It would be easy to adapt it to Saxon, though. It looks as not being possible to make it work with MSXML (as of version 4).

2.4.2. XSL-FO renderer

An XSL-FO renderer is a software that transforms XSL-FO files into other formats, such as PDF (Portable Document Format, made popular by Adobe) and PostScript. For more information about XSL-FO, see the references section of the eAIP Specification.

Two famous XSL-FO renderers are:

- XEP [http://renderx.com/FO2PDF.html], a commercial product from RenderX (recommended).
- FOP [http://xml.apache.org/fop/index.html], an Open Source Software written in Java by the Apache Software Foundation, available free of charge;

There are a few other renderers available. Both FOP and XEP run on any Java 2 platform (Windows, Unix, Mac and others). FOP is free but not complete. XEP is (almost) complete but not free. A free evaluation version of XEP is available at their Web site. We recommend XEP because the eAIP needs a couple of specific features that FOP has not implemented yet. It is possible to work around FOP's lacking features, but it requires some development work on the XSLT style sheets to generate XSL-FO.

2.4.3. MakeAIP.bat

If you want to use the MakeAIP.bat batch file provided in the eAIP package, you should configure it first. It needs to know where are the XSLT and XSL-FO software. Please read MakeAIP.bat documentation for more information.

2.5. Software installation guidelines

Here are some guidelines you might want to follow in order to install all the software needed for the eAIP. If you don't feel comfortable with software installation in general, you might better call your IT support for help. Otherwise, these guidelines - if carefully followed - should lead you to a working eAIP production system based on MakeAIP.

Please note that these guidelines only deal with Java software. There are other options available. Installing Java and configuring Xalan may seem complex at first glance, even for computer specialists. However, you only need to do it once (except if you want to upgrade it later, of course)

Note

The following guidelines are a little bit outdated now, as they don't consider Sun's JVM version 1.4. If you prefer to use this version, please refer to this FAQ entry on Xalan Web site (even if you don't want to use Xalan): Issues running Xalan-Java on JDK 1.4 [http://xml.apache.org/xalan-j/faq.html#faq-N100C3]

2.5.1. Java

We suggest you install the IBM Java Runtime Environment (JRE) version 1.3.x. Other JRE's should work as well, but these guidelines might need to be adapted.

1. Check if you have it already: open a terminal or command prompt window ("MS-DOS") and type `java -version` and then enter. If you have Java correctly installed, it should display its version number (for example: 1.3.0) and some other information, including the provider (for example, IBM). If you don't have a version 1.3.0, we recommend upgrading, unless you know what you are doing. If you do have it, you can skip this section and read about Xalan directly.
2. Download: browse to http://www.ibm.com/java/jdk; register on the Web site (or log in if you are already registered); you will then be presented with a list of packages: click on the JRE 1.3.0 corresponding to your platform (Windows NT, Windows 98, Linux, etc.); you will then see another list: download the installer and the JRE (carefully follow the explanations on the site)
3. Install: carefully follow the instructions provided on the JRE download page; the installation itself is rather easy and you don't have much input to give.
4. Update your PATH: Add the java directory to your PATH environment variable: this depends on the platform you are working on, so you should again refer to the installation instructions (they are in the "docs" directory, where you installed the JRE).

2.5.2. Xalan

1. Download: browse to http://xml.apache.org/xalan-j/index.html. Under the title "How do I get it?", lower in the page, download the binary distribution.
2. Install: create a directory named "c:\dev" (on Unix, name it "~/dev") and unzip the package into this new directory. You can choose another directory, but make sure there is no space in the path.
3. Update your CLASSPATH: Add Xalan and its components to your CLASSPATH environment variable (create this variable if needed). Your CLASSPATH should contain: `c:\dev\xalan-j_2_4_0\bin\xercesImpl.jar;c:\dev\xalan-j_2_4_0\bin\xml-apis.jar;c:\dev\xalan-j_2_4_0\bin\xalan.jar` (under Unix, replace the paths as needed, replace ";" by ":"). CLASSPATH is like the PATH environment variable, but is reserved for Java. You set this variable in the same way as for PATH above.

Note

IBM's JRE comes with an old version of Xerces, which should be removed. Simply delete the file xerces.jar, located in [IBM JRE installation directory]\jre\lib\ext (merely renaming the file will not do it).

2.5.3. FOP

1. Download: browse to http://xml.apache.org/fop/index.html and click on "Download" in the menu. Click on "distribution directory". Download the latest version (currently, Fop-0.20.4-bin.tar.gz).
2. Install: Unzip the package, for example in "c:\dev\fop".
3. Read the documentation of MakeAIP.bat from our eAIP package to configure MakeAIP.bat for your environment.

Chapter 3. MakeAIP.bat documentation

3.1. Introduction

MakeAIP.bat is a batch script for Windows NT/2000. It eases production of eAIP related files, that is conversion of eAIP in XML format to HTML, XSL-FO and PDF.

Note

MakeAIP.bat shall be considered as a proof of concept only. No effort was spent on fine-tuning the tool for better performance. There was no formal validation (software audit) of the HTML and PDF output, to prove that it corresponds at 100% to the content of the XML files. The tool should be revisited, improved and validated before being used in a production environment

This documentation applies to MakeAIP.bat version 2.5, revision 1.12, but it is not complete: more parameters are explained in AMDT procedures.

3.2. Files

Table 3.1.

MakeAIP.bat	Main file, which you use for any command
doc\MakeAIP.html	This file: the documentation
tools\MakeAIP\FOP.bat	Batch file for FOP
tools\MakeAIP\msxsl.bat	Batch file for msxsl.exe
tools\MakeAIP\saxon.bat	Batch file for Saxon
tools\MakeAIP\xalan.bat	Batch file for Xalan
tools\MakeAIP\XEP.bat	Batch file for XEP
tools\MakeAIP\xt.bat	Batch file for XT
html\	Directory where HTML files are generated
fo\ (must be created)	Directory where XSL-FO files are generated
pdf\ (must be created)	Directory where PDF files are generated

The file saxon.bat is pre-configured to work with Saxon 6.5.2, provided with the eAIP package.

3.3. Usage

1. Open a command prompt window ("MS-DOS")
2. Change directory to where you installed the eAIP package

Then you can simply type makeaip.bat to display a summary of usage information.

Note

Tip: Microsoft offers several "power toys" for their various Windows versions. One of them, called "Command Prompt Here" will prove very useful, as it allows to start a command prompt window from File Explorer, directly in a specific directory, by right-clicking on a directory entry. It is available at http://www.microsoft.com/ntworkstation/downloads/PowerToys/Networking/NTComPrompt.asp, for Windows NT4. It exists for other Windows versions as well.

3.3.1. Command-line parameters

MakeAIP.bat accepts several parameters, but only the first one is mandatory. A complete command line looks like this:

```
makeaip target -f format  -d directory -l language -c country
-p XSLT_processor -r XSL-FO_renderer
```

where:

- target is mandatory and is one of the following:
  - sections: generate all eAIP sections such as GEN-2.2, ENR-3.1 and AD-2.EADD
  - menu: generate the navigation menu in HTML (the left frame), for html format only
  - all: generate all the above, for html format only
  - a part, chapter or section name such as GEN-2.2: generate only the specified section
  - an eAIC or eSUP name such as eSUP-2001-01-01 or eAIC-2001-01-01
  - validator: report any DTD validation errors and run MakeAIP schematron (see below)
  - schematron: generates a Schematron report of Additional Rules validation (see eAIP Specification)
  - checklist: generates a complete GEN-0.4 in XSL-FO format (needs -f fo and -s with URI of AT directory)
  - amdt_pages: used for paper AMDT production (see AMDT procedures)
- format is one of the following:
  - html: convert eAIP from XML to HTML
  - fo: convert eAIP from XML to XSL-FO; see notes below
  - pdf: convert eAIP from XSL-FO to PDF; see notes below
- directory is the directory where the eAIP XML file to convert resides
- language is the target language (your eAIP can be in several language, identified by the suffix to the eAIP XML file, as in EC-eAIP-en-GB.xml)
- country is the ICAO country code (in case you have several eAIPs from different countries; identified by the prefix to the eAIP XML file, as in EC-eAIP-en.xml)
- XSLT_processor is one of the following: "saxon", "xalan", "msxsl", "xt" (without the quotes)
- XSL-FO_renderer is one of the following: "FOP", "XEP" (without the quotes)

Note

1. Validator: the validator does not convert the eAIP in any format. Instead, it tries to validate the eAIP, first against the declared DTD, then against the "additional rules". The directory, language and country parameters are recognised by the Validator, not the others.
2. The PDF format is generated using the XSL-FO format. If the XSL-FO does not exist yet, MakeAIP.bat will generated it for you. Note: there is currently a bug which prevents this from working properly, so you will need to run MakeAIP separately for each format: fo, at and pdf successively. If it does exist, then it will be used. This means that if you update the XML source, you will have to

either delete previously generated XSL-FO files, or re-generate them yourself with the "-f fo" option.

3. When generating XSL-FO and PDF formats, you should take care of specifying your renderer for both formats, as the XSL-FO files will be slightly different depending on the -r option.
4. When we write about "sections" in this document, we always mean a 3rd-level element in the eAIP document structure. Parts (GEN, ENR, AD) are the first level, chapters (GEN-2, AD-3, etc.). The third level includes all sections in GEN, ENR, AD-0 and AD-1, and all aerodromes and heliports. It does not include sections in aerodromes and heliports

3.3.2. Examples

3.3.2.1. Generate all HTML files

Command:

```
makeaip all
```

Result: all chapter files in HTML are created, plus the navigation menu, in the directory html\

3.3.2.2. Generate the section ENR-0.1 in HTML

Command:

```
makeaip ENR-0.1
```

Result: the file EC-ENR-0.1-en-GB.html is created in the directory html\

3.3.2.3. Generate the section ENR-0.1 in PDF

Command:

```
makeaip ENR-0.1 -f pdf
```

Result: the file EC-ENR-0.1-en-GB.fo is created in the directory fo\ and the file EC-ENR-0.1-en-GB.pdf is created in the directory pdf\

3.3.2.4. Generate the section ENR-0.1 in PDF for the Dutch version of the Dutch AIP

Command:

```
makeaip ENR-0.1 -f pdf -l nl -c EH
```

Result: the file EC-ENR-0.1-en-GB.pdf is created in the directory pdf\

3.3.2.5. 5. Generate the section ENR-0.1 in PDF using msxsl.exe as XSLT processor and XEP as XSL-FO renderer

Command:

```
makeaip ENR-0.1 -f pdf -p msxsl -r XEP
```

Result: the file EC-ENR-0.1-en-GB.fo is created in the directory fo\ and the file EC-ENR-0.1-en-GB.pdf is created in the directory pdf\

3.3.2.6. Validate the eAIP

Command:

```
makeaip validator
```

Result: MakeAIP calls an XML parser to validate the eAIP against the DTD and then it calls an XSLT processor to validate against the additional rules

3.3.2.7. Circulars and AIP Supplements

Command:

```
makeaip eAIC-2001-02 -d eAIC
```

Result: the file EC-eAIC-2001-02-en-GB.html is created in the directory html\, using the source file eAIC\EC-eAIC-2001-02-en-GB.xml

3.4. Errors during execution

When a program called by MakeAIP terminates with a proper error code, MakeAIP will ask the user if it should continue. At that time, one can press ctrl-C to stop MakeAIP or any other key to continue. This can be useful when generating all sections: the generation can take a couple of minutes, and a lot of information is output to the screen, so it is difficult to know if each file were generated without error.

This behaviour can be switched off: look at the last few lines of MakeAIP.bat.

3.5. Configuration

Before using MakeAIP.bat, you should certainly configure it. There are 2 aspects to configure: the defaults parameters and the location and parameters of XSLT and XSL-FO processors.

3.5.1. Default parameters

MakeAIP.bat command line parameters can be set to default values, so that you don't need to specify them at each use:

Table 3.2.

Command line option	Signification	Initial value	Other possible values
-f	output format	html	fo, pdf
-d	eAIP directory	eAIP	any
-l	language	en-GB	any
-c	ICAO country code	EC	any
-p	XSLT processor	saxon	xalan, msxsl, xt
-r	XSL-FO renderer	XEP	FOP

To change the default values, edit MakeAIP.bat in a text editor such as Notepad, locate the section "Set defaults" near the top of the file and edit the parameters as you wish. Note that some parameters only accept a limited list of values.

3.5.2. XSLT and XSL-FO processors

We provide ready-made batch files for Saxon, Xalan, msxsl.exe, xt, FOP and XEP.

Saxon.bat is already configured to work with the supplied saxon.jar binary. Please note that we only provide this binary file and not the whole Saxon distribution, in order to save space. The complete distribution is available at http://saxon.sourceforge.net/, including documentation and source code.

You need to edit these files (or at least those for the processors/renderers that you will use) with a text editor in order to specify the location of these programs, or more generally, how you want them to be executed. If you followed a standard installation procedure for each of them, the only thing you probably need to change is their home directory. To configure XEP, you might also need to change the name of XEP's jar file, which depends on the version you are using (client, server, etc.). Refer to XEP's own batch files for an example.

These batch scripts always take the same set of parameters:

- For XSLT processors (xalan.bat, msxsl.bat, xt.bat):

  ```
  xxx.bat XML XSLT OUTPUT param1-name param1-value param2-name
  param2-value param3-name param3-value
  ```

  where:

  - XML is the source XML file;
  - XSLT is the style sheet to use;
  - OUTPUT is the resulting file (html or fo);
  - paramX-name and -value are optional XSLT parameters name/value pairs

- For XSL-FO renderers (FOP.bat, XEP.bat):

  ```
  xxx.bat FO OUTPUT
  ```

  where:

  - FO is the source XSL-FO file;
  - OUTPUT is the resulting file (pdf)

3.5.3. Validator

The validator.bat file is different from the others. It first calls a parser to validate the eAIP XML file. This validation is done in the classic XML meaning: the XML file is validated against the DTD that it declares.

Then, validator.bat calls MakeAIP schematron, which runs an XSLT processor to produce a Schematron validation report. This validates the eAIP against the Additional Rules.

3.5.4. Advanced configuration

You can also use other processors: there are many other XSLT processors and FO renderers. To do that, simply copy an existing batch file, rename it and edit it following the program's command line syntax. Then, edit MakeAIP.bat and add your program's name to the variable xsltproc_list (for an XSLT processor) or foproc_list (for an XSL-FO renderer) in the section "CONFIGURATION". Note that the program name you choose should only contain letters and numbers (no space or other special character).

Chapter 4. How to create a new eAIP

4.1. Introduction

This chapter is aimed at people who support eAIP Editors in order to create their first eAIP.

4.2. Procedure

For the sake of this example, let's assume that we are creating a new eAIP for Belgium in French. The ICAO country code is EB and the ISO language code is fr-BE. Also, the eAIP package has been installed in a directory named eAIP-pack and all commands are issued from that directory.

4.2.1. First method: the hard (but clean) way

In this method, we start a new eAIP from scratch.

1. Create a directory named EB-eAIP
2. Create a new XML file named EB-eAIP-fr-BE.xml (we assume you use a good XML editor software from now on)
3. Assign the DTD: the system path should be "../dtd/eAIP.dtd" or the official URL of the DTD (for example: (http://www.eurocontrol.int/ais/ahead/eAIP/dtd/1.0.3/eAIP.dtd"); note that setting a complete path such "C:\eAIP-pack\dtd\eAIP.dtd", but this is wrong, because the system path must be a valid URL, not a Windows path.
4. Hopefully, at this point, your software will have automatically generated the mandatory structure of an eAIP for you in your XML file. However, the eAIP will not be valid yet, and you have a lot of work to make it valid.
5. Edit MakeAIP.bat to change the deault parameters and add reference to the new aerodromes and heliports.
6. Edit EB-eAIP-fr-BE.xml to update all attributes on the "e:eAIP" element;
7. Edit the other XML files to replace text from the Specimen AIP with your AIP text. At this point, you will most certainly need help. Visit EUROCONTROL's Web site [http://www.eurocontrol.int/ais/eaip] and read the eAIP Manuals for more information.
8. When your eAIP is valid, you can split it using the eAIP Splitter (read below for an example).

4.2.2. Alternative method using the Specimen eAIP as a start

This method has the advantage of having complete eAIP content from the beginning. the inconvenient is that this content will eventually need to be replaced or removed.

1. Copy eAIP-pack/eAIP to a new directory named eAIP-pack/EB-eAIP; from now on, we will only work in this new directory;
2. Rename all EC-*-en-GB.xml files, replacing "EC" by "EB" and "en-GB" by "fr-BE";
3. Copy EB-AD-2.EADD-fr-BE.xml for each of your airports, renaming the files with the relevant ICAO airport code;
4. Repeat previous step with EB-AD-3.EADH-fr-BE.xml for each heliport;
5. Edit EB-eAIP-fr-BE.xml to add references to your new airports and heliports files (based on the example with EADD and EADH), in 2 places:

    - At the end of the DOCTYPE section;
    - At the end of the file, within e:AD-2 and e:AD-3 elements

6. Edit MakeAIP.bat to change the deault parameters and add reference to the new aerodromes and heliports.
7. Edit EB-eAIP-fr-BE.xml to update all attributes on the "e:eAIP" element;
8. Edit the other XML files to replace text from the Specimen AIP with your AIP text. At this point, you will most certainly need help. Visit EUROCONTROL's Web site and read the eAIP Manuals for more information.

4.2.2.1. Alternative method using XSLT

This method makes use of the XSLT identity transformation provided within the eAIP package, in the tools/Splitter directory.

1. Create a new directory named eAIP-split in the eAIP package directory.
2. From within the eAIP package directory, run an identity transformation on the Specimen eAIP: "`java -jar tools/saxon/saxon.jar -o eAIP-split/eAIP-full.xml eAIP/EC-eAIP-en-GB.xml tools/Splitter/ns-identity-dtd.xslt`". Note: the new eAIP will declare the official eAIP DTD on EUROCONTROL's Web site in its DOCTYPE section; if you don't have direct access to the internet, you should change it to "../dtd/eAIP.dtd".
3. Edit eAIP-full.xml:

    a. update the e:eAIP element (root element): set ICAO-country-code and xml:lang attributes to your country code and AIP language.
    b. update the content of the e:GEN-2.4 element: remove all existing Location-definition elements and add new ones for all your aerodromes and heliports.
    c. copy the e:Aerodrome and e:Heliport elements of EADZ and EADH as many times as needed for all your aerodromes and heliports. Remove the unneeded ones (EADD, EAZZ). You don't need to have real content in these elements: only 2 things are important: (1) the Ref attribute must refer to the id attribute of the corresponding Location-definition element, and (2) eAIP-full.xml must be valid.

4. Split eAIP-full.xml: "`java -jar tools/saxon/saxon.jar eAIP-split/eAIP-full.xml tools/Splitter/ns-eAIP-splitter.xslt`". eAIP-split directory now contains your split eAIP. Rename this directory to EB-eAIP, for example. Edit the default parameters of MakeAIP.bat to use this directory by default.
5. Edit MakeAIP.bat to change the deault parameters and add reference to the new aerodromes and heliports.
6. Edit EB-eAIP-fr-BE.xml to update all attributes on the "e:eAIP" element;
7. Edit the other XML files to replace text from the Specimen AIP with your AIP text. At this point, you will most certainly need help. Visit EUROCONTROL's Web site and read the eAIP Manuals for more information.

Chapter 5. XSLT style sheet documentation

5.1. Introduction

The EUROCONTROL Electronic AIP Specification is provided with a set of XSLT style sheets meant as an example of how an eAIP XML document can be converted into HTML and XSL-FO formats. The HTML style sheet is designed to present an AIP on a Web site, be it an intranet or a public site; it follows most of the eAIP Usability Guidelines [usability_study.pdf]. The XSL-FO style sheet was developed in order to generate a paper version of an eAIP. This document describes how the XSLT style sheets are structured and explains how the "content logic" was separated from the presentation logic.

5.1.1. Pre-requisites

Readers are expected to have at least a basic knowledge of XSLT. The following resources might be a good place to start: XSLT 1.0 W3C Recommendation [http://www.w3.org/TR/xslt]; D. Pawson's XSLT FAQ. [http://www.dpawson.co.uk/xsl/sect2/sect21.html].

5.2. Files and Templates Organisation

5.2.1. Content logic vs. presentation logic

By content logic, we mean the XSLT code that produces the actual text and data (the content) which compose an AIP. For example, a section title is marked up as such in the XML eAIP document and is copied into the generated HTML and XSL-FO documents, possibly with an automatically-generated numbering. In this case, some content logic may be needed to compute the title numbering.

By presentation logic, we mean the XSLT code that produces the formatting and styling necessary to present the content on the target format (HTML, XSL-FO and others). Continuing with our title example, the title content is typically surrounded by formatting mark-up such as a h1-h6 element in HTML or a block element in XSL-FO. These elements would possibly contain styling mark-up such as a class in HTML and font-related attributes in XSL-FO. Output of these elements and attributes is part of the presentation logic.

Presentation logic is specific to each target format, while content logic can generally be reused for any format. Still with our title example, the numbering computed by the content logic is exactly the same for HTML and XSL-FO. In fact, it should probably be the same for any target format, because numbering is considered to be part of the content. Even though the AIP editors marked the title to be automatically numbered, the number produced must be the same for all target formats.

5.2.2. Files organisation

The style sheets are split in 3 groups:

- Generic style sheets are prefixed with "g-"; example: g-block.xslt;
- HTML-specific style sheets are prefixed with "html-"; example: html-block.xslt;
- XSL-FO-specific style sheets are prefixed with "fo-"; example: fo-block.xslt.

5.2.2.1. Generic style sheets

Generic style sheets contain the "content logic". They are normally imported by formatting-specific style sheets. Generic style sheets do not generate any element. They generate the content (the text) that must be produced automatically (such as for auto-numbering) and combine the various pieces of content together where needed.

It is mostly within generic style sheets that eAIP elements are matched by templates. These templates deal with content logic that need to be applied before calling a formatting-specific template. The formatting template should be overridden by a calling style sheet if any formatting is needed, but it is not mandatory.

5.2.2.2. HTML-specific style sheets

These style sheets contain formatting templates specific to HTML. Templates in these style sheets override named presentation templates defined in generic style sheets.

5.2.2.3. XSL-FO-specific style sheets

These style sheets contain formatting templates specific to XSL-FO. Just like in HTML-specific style sheets, templates in these style sheets override named presentation templates defined in generic style sheets.

5.2.2.4. Table of XSLT files

This table list all XSLT files alphabetically with a short description of their purpose. See also the elements to XSLT files mapping table, which indicate where each element is processed.

Table 5.1.

Generic	HTML	XSL-FO	Description
-	html-amdt-list.xslt	-	Generates a list of all changes in a given AMDT (in development)
g-amendments.xslt	html-amendments.xslt	fo-amendments.xslt	Amendment elements and attributes, including 0.2 sections
-	-	fo-att-set.xslt	Generic templates for dynamic class name/attribute-set processing (see below, Dynamic Attribute-set Processing for details)
g-block.xslt	html-block.xslt	fo-block.xslt	Processing of block elements such as Address and div
-	-	fo-checklist.xslt	Generation of the checklist of pages in GEN-0.4
g-content.xslt	html-content.xslt	fo-content.xslt	Initialisation of common variables and parameters, plus eAIP element processing for HTML
g-eAIC-eSUP.xslt	html-eAIC-eSUP.xslt	fo-eAIC-eSUP.xslt	Elements and attributes related to Circulars and Supplements documents (but not the Supplement element used in an eAIP document
g-generic.xslt	html-generic.xslt	fo-generic.xslt	Common templates used for many elements; default templates
g-gettext.xslt	-	-	A generic set of templates to retrieve the localisation of words and sentences
g-graphics.xslt	html-graphics.xslt	fo-graphics.xslt	Templates related to Figure and Graphic-file elements
g-inline.xslt	html-inline.xslt	fo-inline.xslt	Processing of inline elements such as Abbreviation and strong

Generic	HTML	XSL-FO	Description
g-links-s.xslt	html-links-s.xslt	fo-links-s.xslt	Generation of links within an eAIP document
g-lists.xslt	html-lists.xslt	fo-lists.xslt	Processing of list elements such as Location-definition and ul
-	html-menu.xslt	-	Generation of the table of contents for the HTML navigation interface
g-numbering.xslt	html-numbering.xslt	fo-numbering.xslt	Auto-numbering templates
-	-	fo-pages.xslt	Page layouts, page headers and footers
g-structure.xslt	html-structure.xslt	fo-structure.xslt	Processing of structural elements such as Sub-section, sections, chapters and Title elements
-	-	fo-styles.xslt	Definition of style classes for XSL-FO; conceptually equivalent to the eAIP.css file in HTML
g-supplements.xslt	html-supplements.xslt	fo-supplements.xslt	The Supplement in an eAIP document and related templates
-	html-table.xslt	fo-table.xslt	Processing of tables-related elements (especially for XSL-FO)
g-tables-ENR.xslt	html-tables-ENR.xslt	fo-tables-ENR.xslt	All ENR part specific elements, such as Route, Designated-point and Navaid
g-toc.xslt	html-toc.xslt	fo-toc.xslt	Generation of tables of contents (0.6 sections)

5.2.3. Templates organisation

In order to separate as much as possible content from presentation, the XSLT code was split in three parts. Most elements are handled by two or three templates. For the sake of this documentation, let's call them "the matching template", "the presentation template" and "the content template".

In general, templates will only process one element or one attribute (though presentation templates might generate several elements and attributes as needed). Therefore, attributes may have their own set of three templates.

5.2.3.1. Matching template

Matching templates are generally defined in generic style sheets. Most often, they will simply call the relevant presentation template. In some cases, they contain some content logic that needs to be applied before any formatting. The table of content is an example of such case. Matching templates are normally re-used for any target format.

When content cannot be fully separated from presentation, matching templates are defined in formatting-specific style sheets as well and they then override generic matching templates. Two XSLT mechanisms can then be used: the fact that templates defined within the importing style sheet will always override an imported template with the same priority; or by using a more specific matching expression in the importing style sheet.

5.2.3.2. Presentation template

In most cases, presentation templates generate a formatting-specific element or attribute, which depends on the specific formatting targeted. They call the content template within the generated element. Default presentation templates (in generic style sheets) do nothing but call the relevant content template.

5.2.3.3. Content template

Content templates only generate the actual AIP content: text and other data. They are normally re-used for any target format.

5.2.4. Example

The Designated-point element gives us a complete example:

5.2.4.1. Matching template

A Designated-point can be listed or not in the Designated-points table. If the Listed attribute is set to "Yes", then the Designated-point's entry in the table is generated. This if-clause is common to both HTML and PDF presentations, so the relevant code is located in the generic style sheet (g-tables-ENR.xslt), in this matching template.

```
<xsl:template match="e:Designated-point">
    <xsl:if test="@Listed='Yes'"><!-- foreign points are not listed -->
        <xsl:call-template name="Designated-pointStyle"/>
    </xsl:if>
</xsl:template>
```

5.2.4.2. Presentation template

In HTML as in XSL-FO, a table row is generated for each (listed) designated-point. However, the table row element in HTML is not the same as the one for XS-FO. Also, the XSLT code to show amendments in HTML is different to the one in XSL-FO. Hence, these two quite different templates, first for HTML (html-tables-ENR.xslt):

```
<xsl:template name="Designated-pointStyle">
    <tr>
        <xsl:apply-templates select="@*"/>
        <xsl:call-template name="Designated-pointContent"/>
```

```
5.      </tr>
6.  </xsl:template>
```

And then for XSL-FO (fo-tables-ENR.xslt):

```
1.  <xsl:template name="Designated-pointStyle">
2.      <xsl:if test="@Updated != 'Deleted'">
3.          <fo:table-row>
4.              <xsl:apply-templates select="@*[not(starts-with(local-name(), 'Updated'))]"/>
5.              <fo:table-cell xsl:use-attribute-sets="Amdt-Cell">
6.                  <xsl:apply-templates select="@Updated"/>
7.                  <xsl:apply-templates select="e:Designated-point/@Updated-ref"/>
8.                  <fo:block><xsl:call-template name="Deleted-MarkRoute"/>&#xa0;</fo:block>
9.              </fo:table-cell>
10.         <xsl:call-template name="Designated-pointContent"/>
11.         </fo:table-row>
12.     </xsl:if>
13. </xsl:template>
```

While being quite different, they have one mandatory line in common (line 4 for HTML and line 10 for XSL-FO): the call to the content template, which contains the rest of common XSLT code.

5.2.4.3. Content template

Back to the generic style sheet, where the "Designated-pointContent" template calls the relevant templates to process the content of a Designated-point element:

```
1.  <xsl:template name="Designated-pointContent">
2.      <xsl:apply-templates select="e:Designated-point-ident"/>
3.      <xsl:apply-templates select="e:Latitude"/>
4.      <xsl:call-template name="Designated-pointRouteStyle"/>
5.  </xsl:template>
```

5.3. Elements organisation

5.3.1. Where does it start?

There are three "main" style sheets: html-content.xslt for HTML pages, html-menu.xslt for the HTML Navigation menu (ToC) and fo-content for FO files. These 3 files import many other files, which handle specific groups of eAIP elements.

5.3.2. Elements to Files Mapping

The following table shows which XSLT file handle each element.

Table 5.2.

Element	For XHTML	For XSL-FO	Element	For XHTML	For XSL-FO
/ (document root)	-	pages	Abbreviation	inline	inline
Abbreviation-description	lists	lists	Abbreviation-details	lists	lists
Abbreviation-ident	lists	lists	AD, AD-x, AD-x.y	structure	structure
Address	block	block	Address-part	block	block
Aerodrome	structure	structure	Affects	supplements	supplements
Amendment	supplements	supplements	Date-time	inline	inline
Deleted	amendments	amendments	Designated-point	tables-ENR	tables-ENR
Designated-point-ident	tables-ENR	tables-ENR	Designated-point-table	tables-ENR	tables-ENR
eAIC	eAIC-eSUP	-	eAIP	content	-
eAIP-reference	eAIC-eSUP	eAIC-eSUP	ENR, ENR-x, ENR-x.y	structure	structure
eSUP	eAIC-eSUP	-	Figure	graphics	graphics
GEN, GEN-x, GEN-x.y	structure	structure	Generated (in AD 2.1 & 3.1)	structure	structure
Generated (in sections 0.6)	toc	toc	Graphic-file	graphics	graphics
Heliport	structure	structure	Inserted	amendments	amendments
Latitude (Designated-point)	tables-ENR	tables-ENR	Latitude (Navaid)	tables-ENR	tables-ENR
Location	inline	inline	Location-definition	lists	lists
Location-ident	lists	lists	Location-name	lists	lists
Location-table	lists	lists	Longitude (Designated-point)	tables-ENR	tables-ENR
Longitude (Navaid)	tables-ENR	tables-ENR	Navaid	tables-ENR	tables-ENR
Navaid-declination	tables-ENR	tables-ENR	Navaid-elevation	tables-ENR	tables-ENR
Navaid-frequency	tables-ENR	tables-ENR	Navaid-hours	tables-ENR	tables-ENR
Navaid-ident	tables-ENR	tables-ENR	Navaid-indication	tables-ENR	tables-ENR

Element	For XHTML	For XSL-FO	Element	For XHTML	For XSL-FO
Navaid-indication-distance	tables-ENR	tables-ENR	Navaid-indication-radial	tables-ENR	tables-ENR
Navaid-magnetic-variation	tables-ENR	tables-ENR	Navaid-name	tables-ENR	tables-ENR
Navaid-remarks	tables-ENR	tables-ENR	Navaid-table	tables-ENR	tables-ENR
Navaid-type	tables-ENR	tables-ENR	NIL	structure	structure
Not-applicable	block	block	References	eAIC-eSUP	eAIC-eSUP
Route	tables-ENR	tables-ENR	Route-designator	tables-ENR	tables-ENR
Route-remark	tables-ENR	tables-ENR	Route-RNP	tables-ENR	tables-ENR
Route-segment	tables-ENR	tables-ENR	Route-segment-airspace-class	tables-ENR	tables-ENR
Route-segment-ATC	tables-ENR	tables-ENR	Route-segment-COP	tables-ENR	tables-ENR
Route-segment-length	tables-ENR	tables-ENR	Route-segment-lower	tables-ENR	tables-ENR
Route-segment-lower-override	tables-ENR	tables-ENR	Route-segment-mag-track	tables-ENR	tables-ENR
Route-segment-minimum	tables-ENR	tables-ENR	Route-segment-remark	tables-ENR	tables-ENR
Route-segment-remark-reference	tables-ENR	tables-ENR	Route-segment-reverse-mag-track	tables-ENR	tables-ENR
Route-segment-reverse-true-track	tables-ENR	tables-ENR	Route-segment-RNP	tables-ENR	tables-ENR
Route-segment-true-track	tables-ENR	tables-ENR	Route-segment-upper	tables-ENR	tables-ENR
Route-segment-usage	tables-ENR	tables-ENR	Route-segment-usage-direction	tables-ENR	tables-ENR
Route-segment-usage-level-type	tables-ENR	tables-ENR	Route-segment-usage-reference	tables-ENR	tables-ENR
Route-segment-width	tables-ENR	tables-ENR	See-supplement	supplements	supplements

Element	For XHTML	For XSL-FO	Element	For XHTML	For XSL-FO
Significant-point-ATC	tables-ENR	tables-ENR	Significant-point-description	tables-ENR	tables-ENR
Significant-point-reference	tables-ENR	tables-ENR	Significant-point-remark	tables-ENR	tables-ENR
Significant-point-remark-reference	tables-ENR	tables-ENR	Sub-section	structure	structure
Supplement	supplements	supplements	SUP-section	structure	structure
Title	structure	structure	a	generic	inline
br	generic	inline	caption	generic	table
cite	generic	inline	col	generic	table
colgroup	generic	table	div	block	block
em	generic	inline	li	generic	lists
ol	generic	lists	p	generic	block
span	generic	inline	strong	generic	inline
table	table	table	tbody	generic	table
td	generic	table	tfoot	generic	table
th	generic	table	thead	generic	table
tr	generic	table	ul	generic	lists

5.4. Customisation

How to integrate your own templates with the provided ones? If you want to re-use most of the provided style sheets but need customisation for a few templates, you can avoid modifying the provided style sheets. Here is how:

1. Create your own XSLT file and write your templates (matching or named) in it;
2. In your XSLT file, import one of the provided main files (html-content.xslt, html-menu.xslt or fo-content.xslt), depending on your needs; by that way, your templates will override the imported ones;
3. In your templates, take care to call the right provided templates or to "apply-templates", where applicable, to make the link to other, provided templates;
4. Alternatively, if your templates allows it, use "apply-imports" to call the default processing for the node your template matches.

5.5. Content Logic Implementation

The objective of this section is to explain how the eAIP content logic is implemented, when it may see awkward.

5.5.1. Route Tables

The XSLT code which builds route tables in ENR 3.1 to ENR 3.5 was originally developed "by example", i.e. by closely following the ICAO AIP Specimen. This may seem too restrictive sometimes, but can be changed: These restrictions are only present in the XSLT transformations and not in the DTD. Specifically, the following rules are implemented:

- Route-segment-width and Route-segment-COP are not considered for ENR-3.3 and 3.5
- Route-segment-minimum and Route-segment-lower-override are not considered for 3.2, 3.3 and 3.5

5.6. Dynamic Attribute-set Processing

5.6.1. The Issue

The eAIP DTD features a class attribute on every elements, with the objective of separating presentation from content. When an element needs a specific styling, the editor gives a name to this styling and use it in the class attribute. This name is then used in HTML and XSL-FO transformations to look for the corresponding style definition.

For XHTML, the transformation is rather straightforward: the class name is simply copied to the XHTML file generated. The style definition are located in a CSS file named eAIP.css.

In XSL-FO, there is no way to define classes externally: all styling has to be defined as attributes on XSL-FO elements. So, our transformation converts class names to XSLT attribute-sets elements, which are applied to XSL-FO elements. Attribute-sets are defined in the file fo-styles.xslt.

However, a limitation of XSLT imposes that attribute-sets are used statically. That is, a transformation is supposed to know in advance the name of an attribute-set it uses. We sometimes have static attribute-sets, when we use the default class name for a given element. For example, an Address element is formatted using the Address attribute-set. Here is its presentation template (amendment processing removed for clarity):

```
<xsl:template name="AddressStyle">
    <fo:block xsl:use-attribute-sets="Address">
        <xsl:call-template name="AddressContent"/>
    </fo:block>
</xsl:template>
```

But, when we need to convert the class name declared in a class attribute, this is of course not possible and we then need to apply an attribute-set dynamically. The following is illegal XSLT code:

```
<xsl:template name="AddressStyle">
    <fo:block xsl:use-attribute-sets="Address {@class}"><!-- illegal -->
        <xsl:call-template name="AddressContent"/>
    </fo:block>
</xsl:template>
```

5.6.2. The Solution

There are three known workarounds, as explained by Jeni Tennison in an email on the XSL-list mailing list titled "Re: [xsl] using variable as a value for "use-attribute-sets" [http://www.biglist.com/lists/xsl-list/archives/200201/msg00189.html]". We have chosen the second solution, which basically consists in parsing our fo-styles.xslt file from within our transformation, just like any other XML file, and use it as our data source for class names to attribute-sets conversion. This trick is implemented in the file fo-att-set.xslt. We expanded Jeni Tennison's example so that variables and parameters are processed and attribute-sets can use other attribute-sets.

The following template, taken from fo-att-set.xslt, shows the generic handling of the class attribute:

```
<xsl:template match="@class">
    <xsl:call-template name="use-att-set">
        <xsl:with-param name="CLASS" select="string(.)"/>
    </xsl:call-template>
</xsl:template>
```

Note that in most cases, this template is not actually used, because the value of the class attribute generally needs to be combined with other classes.

A class name is defined as follows in fo-styles.xslt:

```
1.  <xsl:attribute-set name="region-common">
2.      <xsl:attribute name="margin-top">1cm</xsl:attribute>
3.      <xsl:attribute name="margin-bottom">1.5cm</xsl:attribute>
4.  </xsl:attribute-set>
5.
6.  <xsl:variable name="outside-margin">1cm</xsl:variable>
7.  <xsl:variable name="inside-margin">2cm</xsl:variable>
8.
9.  <xsl:attribute-set name="Left-region-body" use-attribute-sets="region-common">
10.     <xsl:attribute name="margin-left"><xsl:value-of select="$outside-margin"/></xsl:attribute>
11.     <xsl:attribute name="margin-right"><xsl:value-of select="$inside-margin"/></xsl:attribute>
12. </xsl:attribute-set>
```

The class Left-region-body is implemented by the attribute-set of the same name. This attribute-set uses another attribute-set, region-common, which sets more attributes, common to Left-region-body and others. Left-region-body's attributes make use of variables, also defined above.

5.6.3. Limitations

There are some limitations in our implementations:

- A variable or parameter must be defined by its content (as in the variable outside-margin, line 6 above), not by its select attribute, if it is called in an attribute inside an attribute-set that may be used dynamically;
- In such a variable or parameter, only text nodes will be used; specifically, value-of elements will not be resolved (they are only resolved in an attribute, as in line 10 above).

It is possible to overcome these limitations, but we don't need it so far.

Chapter 6. eAIP Security - Technical and procedural choices

6.1. Introduction

This chapter explains technological and procedural choices, which have been tested when developing the eAIP Security guidelines. See also the eAIP Security Risks and Mitigation Strategies in the eAIP Specification.

6.2. Possible questions

6.2.1. List of questions

1. Why sign only packages and not individual documents?
2. Why cover only transport integrity and authenticity?
3. Why choose x509?
4. Why choose PGP?
5. PGP and x509. Which on to choose?
6. Why not choose XML Signature?

6.2.2. Why sign only packages and not individual documents?

Signing can be a lengthy process: an eAIP can be composed of more than 100 files, including charts. Depending on the software and procedure used, the signing party might have to enter his password for each file to sign. On slower hardware, signing a large number of file can also be long, depending on the algorithms chosen.

6.2.3. Why cover only transport integrity and authenticity?

The other possible areas to cover could be:

- Confidentiality: making sure unwanted third parties cannot obtain the information;
- Availability: guarantee that the end-user has access to the information, all the time;
- Non-repudiation: making sure the end-user cannot reject having received the information.

6.2.3.1. Confidentiality

The content of an eAIP is public information and as such we have not looked into methods for ensuring confidentiality. However, the publishing party can set up encrypted channels to distribute the eAIP, e.g. using an SSL-enabled website (with two way SSL) or using encrypted email.

6.2.3.2. Availability

As the eAIP Security guidelines are independent of the ways the eAIP is distributed, it is outside the scope of these documents to specify availability requirements regarding these distribution channels. It is up to the publisher and the end-user to agree on availability.

6.2.3.3. Non-repudiation

Non-repudiation is the ability to guarantee that a person has sent a message and that the receiving party has indeed received it. This is done typically by a trusted third party, which holds arbitration powers.

The current paper AIP distribution does not have a non-repudiation mechanism. Therefore, it was not included in the scope of the eAIP Specification.

6.2.4. Why choose x509?

x509, the ITU standard for public key cryptography, is widely used in PKI, in SSL and other security-related protocols. By its widespread use, it is natural to adopt this standard for eAIP security.

6.2.5. Why choose PGP?

PGP, a public key system devised by Phill Zimmerman, is also used extensively to secure communications via email, instant messenger, etc. Its ease of use and decentralised management make it ideal for small organisations and tight IT budgets. Therefore, these two methods have been chosen, as they could ease the adoption and implementation of eAIP signature mechanisms.

6.2.6. PGP and x509: Which one to choose?

Choosing between x509 and PGP is a choice which has to be carefully taken. Here are a few elements to help deciding:

- Existing or planned use of x509: If your organisation already has x509 PKI deployed and has management procedures in place, then it is trivial to integrate the signing of eAIP into it. In case you are planning a x509 PKI roll-out, you could use the eAIP signing as a test case. You can also use the x509 tutorial to get acquainted with x509 concepts before plunging in to PKI.
- Limited security budget: creating and managing an x509 PKI is costly, both on the software and personnel. PGP is significantly less costly and simple to set up.
- Plan for organisational-wide security: If your organisation does not plan to roll out PKI, PGP is a good choice as it is decentralised by nature. Indeed, any individual or organisational unit may create a PGP key. On the other hand, x509 requires central administration of a CA, and security procedures for signing and certifying requests. This fits well in the centralised security model used by many enterprises.
- Simple test of eAIP security: To simply test the signature of the eAIP, we recommend you use PGP, as it is easier to set up. Use x509 if you are interested in this technology and are planning a PKI roll-out, or if you already have a PKI.

6.2.7. Why not choose XML Signature?

The XML Signature is a W3C [http://www.w3.org] recommendation defining the integration of digital signatures within XML documents. Using this recommendation, it is possible to sign a complete document or some fragments of it, to embed the signatures inside the XML document (as well as outside, like X.509 and PGP).

Signing a fragment of a document allows several authors to separately sign their part of the same document. It would therefore be possible for each editor to sign individually his/her work (including charts, and modifications linked to amendments).

Since the eAIP is seen from the user's standpoint as being a single publication coming from the official publishing organisation, we have concluded that signing parts of a document is not an eAIP security requirement.

6.3. Further reading

6.3.1. Public key cryptography

- SSL Certificates HOWTO [http://www.gtlib.cc.gatech.edu/pub/linux/docs/HOWTO/other-formats/html_single/SSL-Certificates-HOWTO.html]; Franck Martin
- The Great List - SSL - Encryption [http://www.greatlist.com/?ssl]; Caribe WWW Research

6.3.2. x509

- Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks [http://www-t.zhwin.ch/it/ksy/Block08/ITU/X509_4thEditionDraftV8.pdf]; International Telecommunication Union
- An Introduction to XML Digital Signatures [http://www.xml.com/pub/a/2001/08/08/xmldsig.html]; Ed Simon, Paul Madsen, Carlisle Adams
- XML-Signature Syntax and Processing [http://www.w3.org/TR/xmldsig-core/]; W3C

6.3.3. PGP

- Gnu Privacy Guard (GnuPG) Mini Howto [http://webber.dewinter.com/gnupg_howto/english/GPGMiniHowto.html]; Brenno J.S.A.A.F. de Winter, Michael Fischer v. Mollard, Arjen Baart
- Corporate use of PGP & Key Management/recovery [http://biphome.spray.se/laszlob/pgp/Paper.doc]; Laszlo Baranyi

Chapter 7. eAIP Security - How to setup up a x509 signing environment

7.1. Introduction

This chapter describes the steps to:

1. Set up a CA (Certification Authority)
2. Create a signing certificate
3. Distribute the certificate to the end-users

Setting up a x509 CA is a complex operation. It is recommended that the user gets familiar with x509 and public key cryptography in general by reading the links provided in the Technical and procedural choices chapter. See also the eAIP Security Risks and Mitigation Strategies in the eAIP Specification.

7.2. Necessary software

Please download and install the following software. You might need administrative privileges or ask your system administrator to perform the installation.

- XCA [http://www.hohnstaedt.de/xca/xcawin.html]: graphical interface to OpenSSL for management of x509 certificates. This tutorial covers version 0.4.4.

7.3. Description of the steps required

7.3.1. Overview

The steps to set up a Certification Authority for an organisation are the following:

1. Generate a private key / public key pair used by the CA
2. Generate a self-signed CA certificate

The CA is then able to sign certificates.

To create a certificate for the person or entity signing the eAIP package:

1. The signing person generates a private key / public key pair
2. The signing person generates a certificate signing request
3. The signing person sends the CSR to the CA
4. The CA signs the request, creates a certificate, and sends it back to the person or entity
5. The signing person imports the certificate and verifies it

The person or entity signing the eAIP package has a certificate and is able to sign eAIP packages, as described in How to sign an eAIP with x509

7.4. Setting up a CA

7.4.1. Introduction

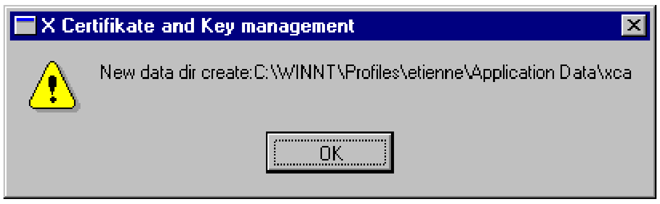
The steps are

1. Initial software configuration
2. Creation of CA public and private key pair
3. Creation of CA certificate
4. Export and distribution of CA certificate

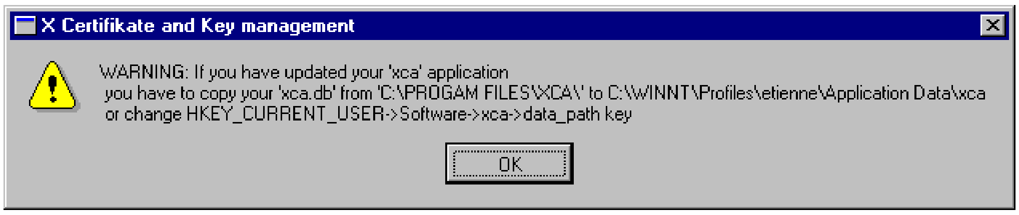
7.4.2. Initial software configuration

Run the installer package, leaving the default options

After installation, run the XCA program from the Start Menu. You might be notified with the following messages which you may safely ignore.

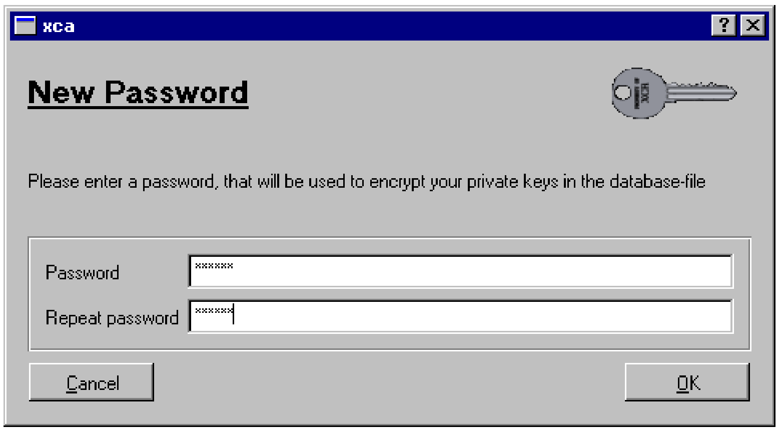

X Certifikate and Key management

New data dir create:C:\WINNT\Profiles\etienne\Application Data\xca


X Certifikate and Key management

WARNING: If you have updated your 'xca' application
you have to copy your 'xca.db' from 'C:\PROGAM FILES\XCA\' to C:\WINNT\Profiles\etienne\Application Data\xca
or change HKEY_CURRENT_USER->Software->xca->data_path key

You will be asked to enter a password to protect your keys and your configuration:


xca

New Password

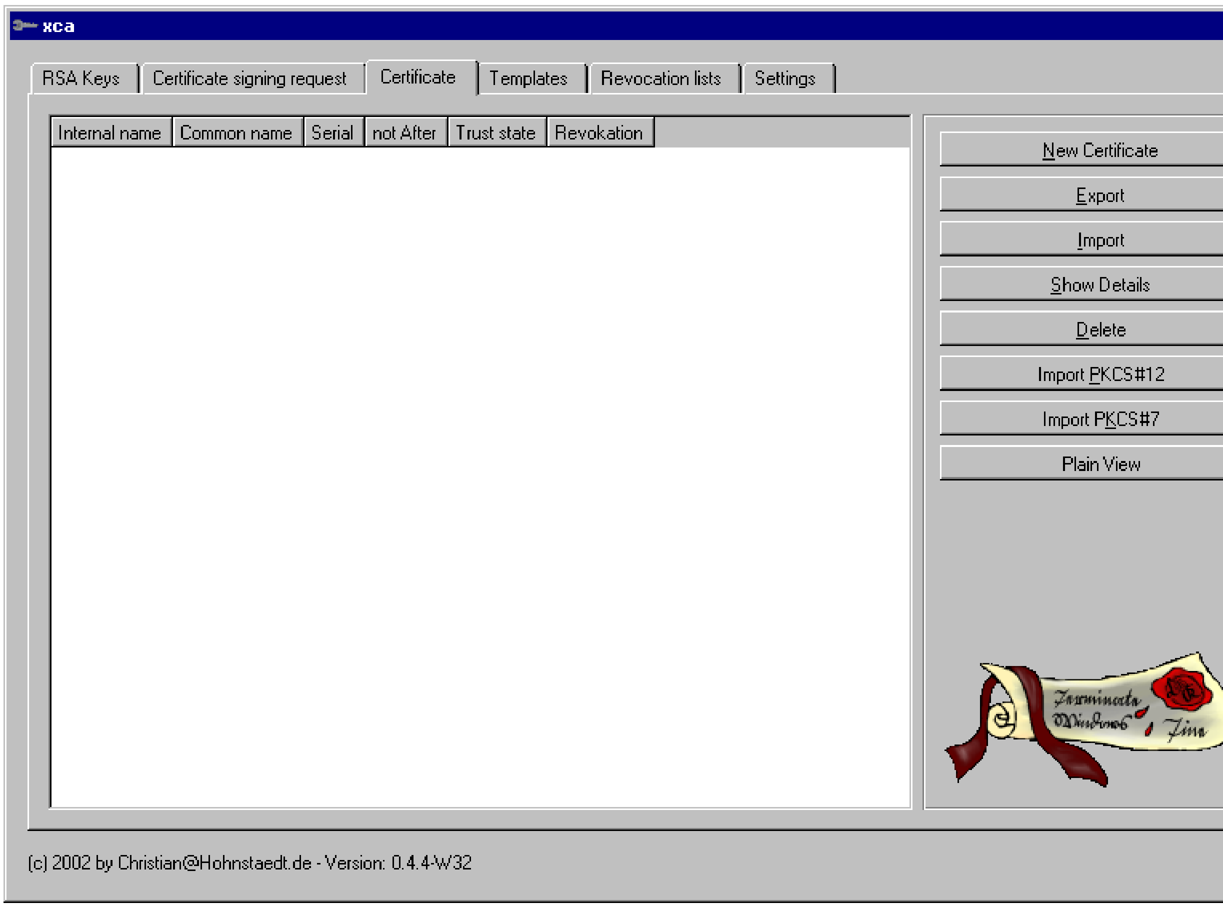
Please enter a password, that will be used to encrypt your private keys in the database-file

Password ******
Repeat password ******

Enter a strong password twice and click OK. Please remember this password, as there is no way to retrieve it.

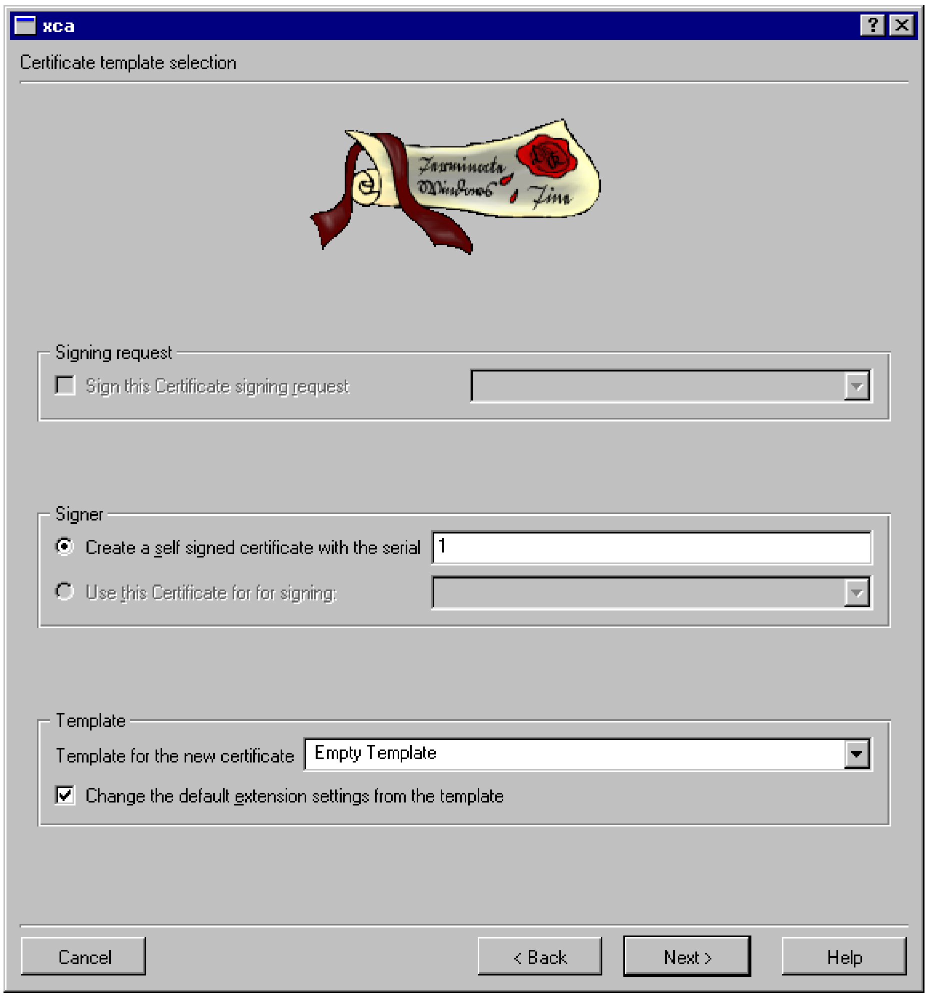
You are presented with the main interface of the application.

7.4.3. Creation of CA public and private key pair



(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Click on the Certificate tab on top, then on the New certificate button. A wizard opens, click Next.



Certificate template selection

Signing request

☐ Sign this Certificate signing request

Signer

● Create a self signed certificate with the serial: 1

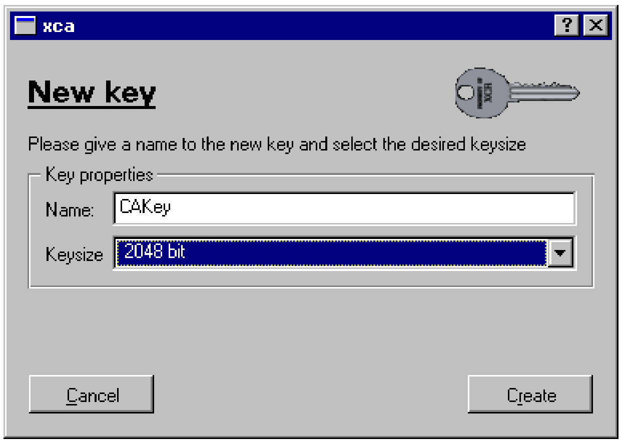
○ Use this Certificate for for signing:

Template

Template for the new certificate: Empty Template

☑ Change the default extension settings from the template

Leave the values to their default. Click Next.



New key

Please give a name to the new key and select the desired keysize

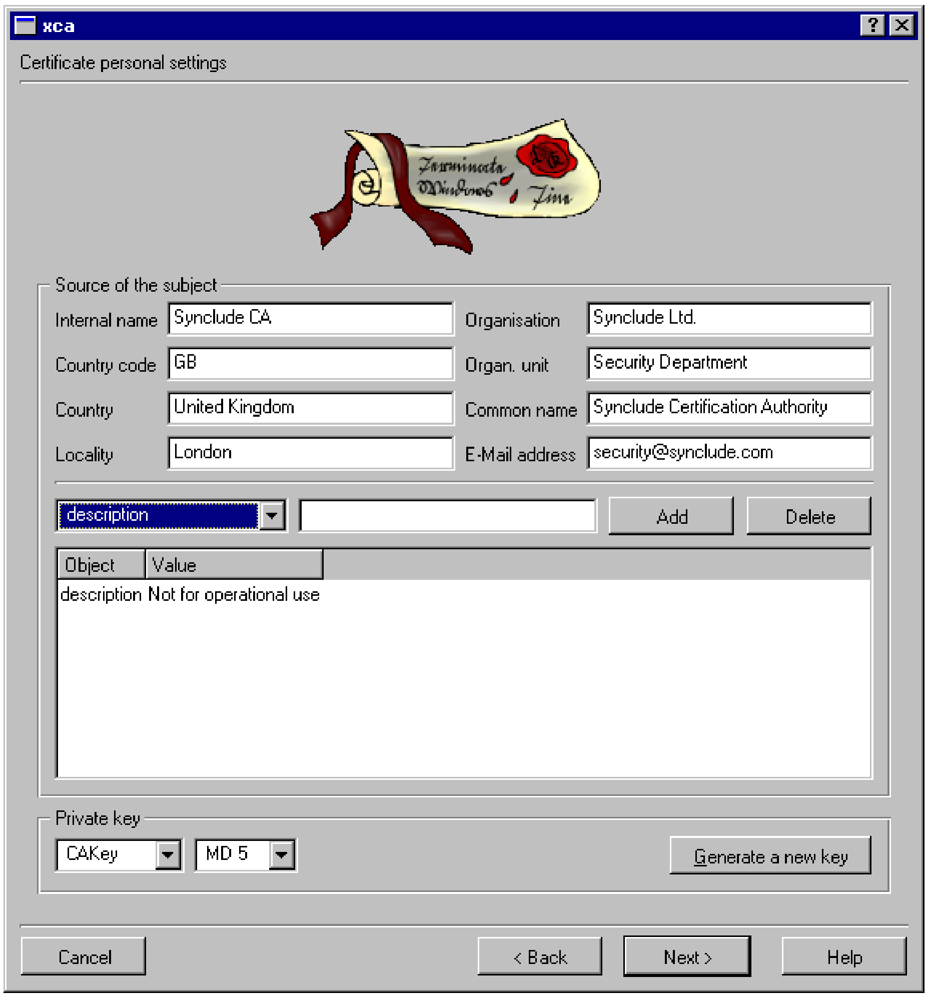
Key properties

Name: CAKey

Keysize: 2048 bit

Enter the CA key name you wish, and choose the key strength you need (at least 2048 is recommended). Click Create to start the key generation.

7.4.4. Creation of CA certificate



Source of the subject

Internal name	Synclude CA	Organisation	Synclude Ltd.
Country code	GB	Organ. unit	Security Department
Country	United Kingdom	Common name	Synclude Certification Authority
Locality	London	E-Mail address	security@synclude.com

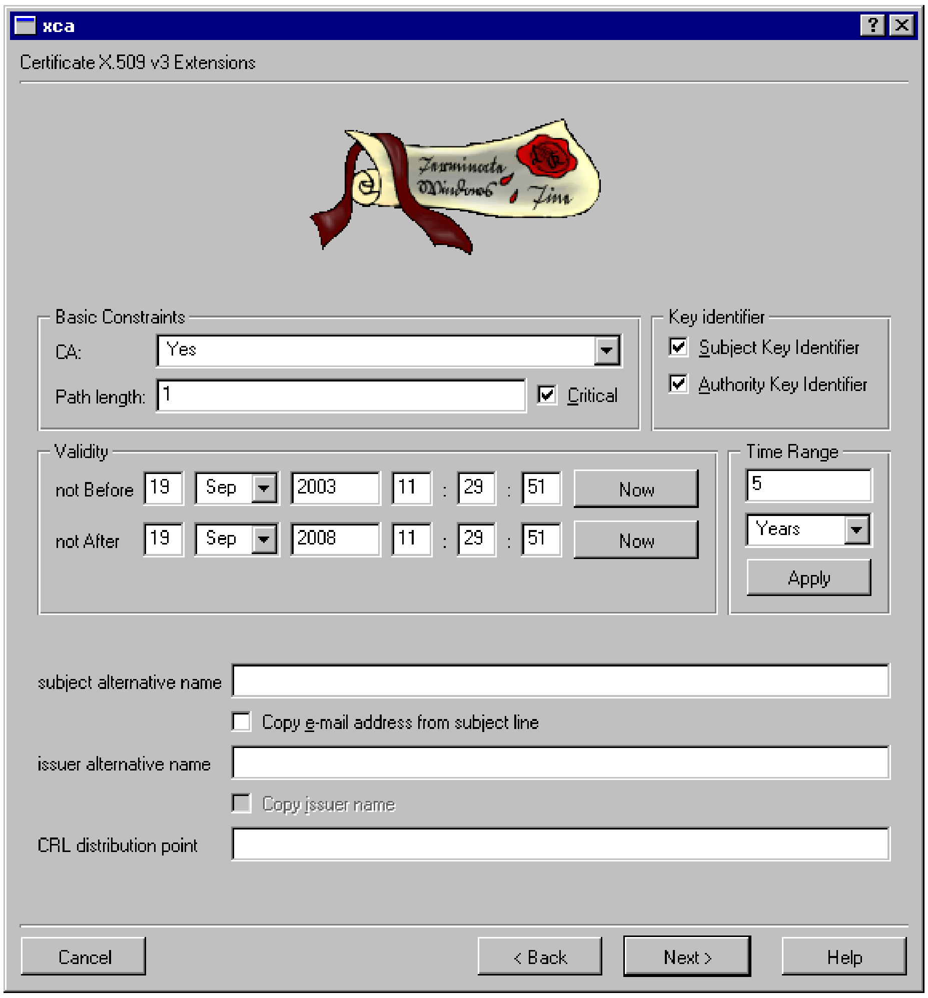
Object	Value
description	Not for operational use

Private key: CAKey, MD 5

Enter the information relative to your organisation:

- Internal name: A name by which you will recognise this certificate in the XCA software.
- Organisation: The full name of your organisation.
- Country Code: The ISO country code of your organisation.
- Organ. Unit: The unit of the organisation which is responsible for the management of the certificates,
- Country: The country of your organisation.
- Common Name: The name by which this certificate will be known.
- Locality: The city of your organisation.
- E-Mail address: The email address at which the unit responsible for the management of certificates can be contacted.

At the bottom, in Private Key, make sure the name of the key you just generated is selected. Click Next.


xca

Certificate X.509 v3 Extensions

Basic Constraints

CA: Yes

Path length: 1 ✓ Critical

Key identifier

✓ Subject Key Identifier

✓ Authority Key Identifier

Validity

not Before 19 Sep 2003 11 : 29 : 51 Now

not After 19 Sep 2008 11 : 29 : 51 Now

Time Range

5

Years

Apply

subject alternative name

☐ Copy e-mail address from subject line

issuer alternative name

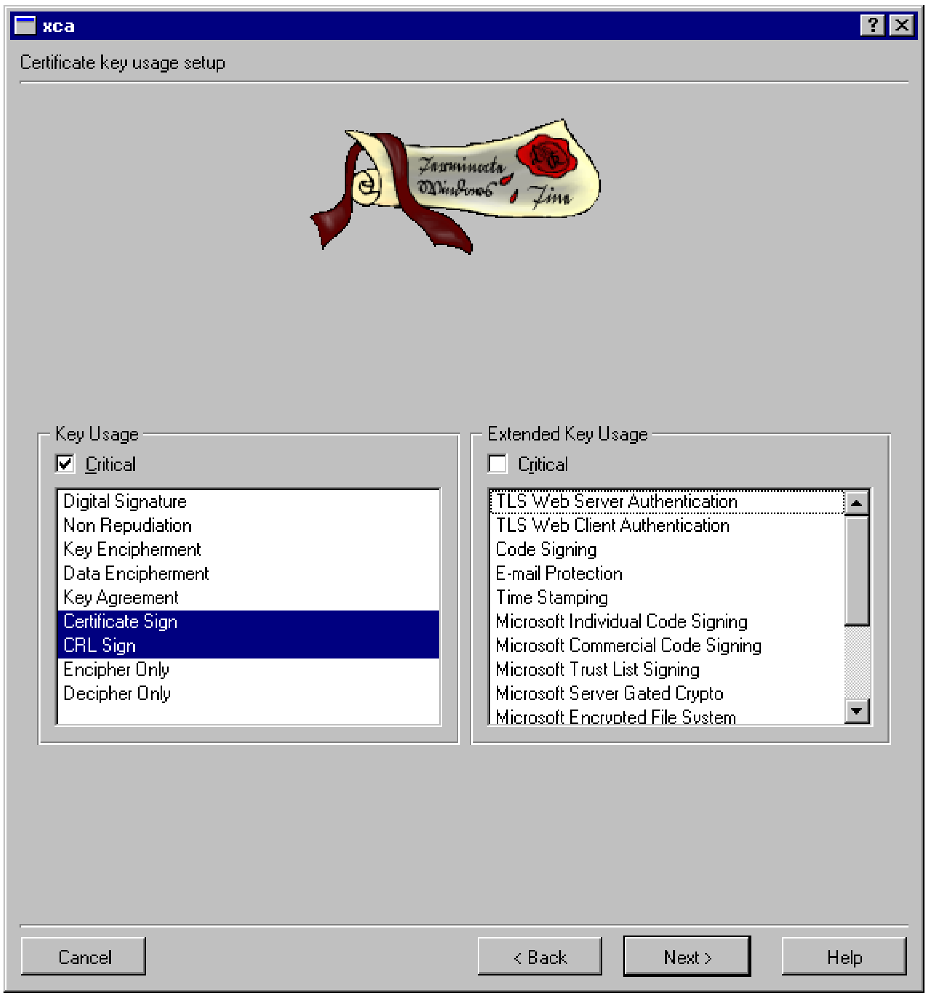
☐ Copy issuer name

CRL distribution point

Enter the following:

- CA: Choose Yes, as this is a CA certificate.
- Path Length: Enter 0 unless you wish to create sub CAs.
- Key Identifier: Select both options to identify uniquely this key and certificate.
- Validity: Enter the start and end dates of the validity of this certificate. Keep it long enough, as all certificates issued by this CA will no longer be considered valid after the expiration of the CA certificate.
- CRL distribution point: Enter the URL precede by URI: at which you will post the Certificate Revocation List (list of certificates which are cancelled).

Click Next.



Certificate key usage setup

Key Usage

[✓] Critical

- Digital Signature
- Non Repudiation
- Key Encipherment
- Data Encipherment
- Key Agreement
- [✓ Certificate Sign]
- [✓ CRL Sign]
- Encipher Only
- Decipher Only

Extended Key Usage

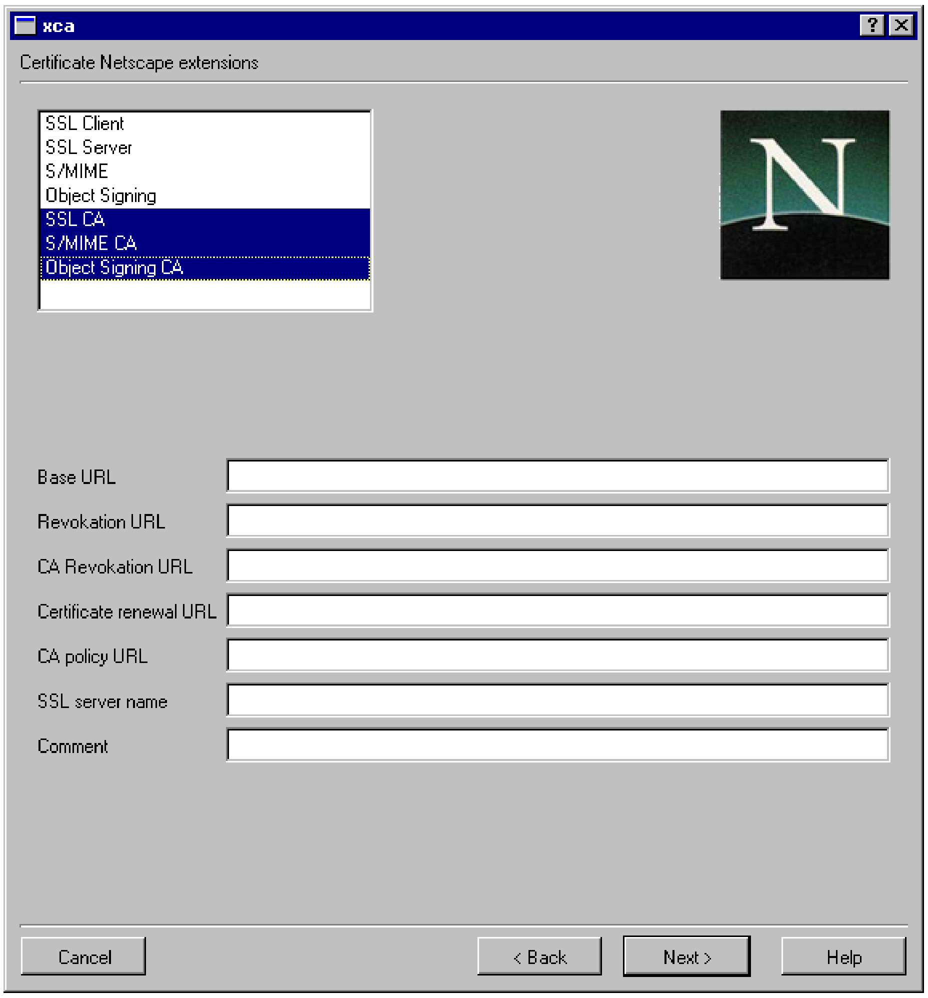
[ ] Critical

- TLS Web Server Authentication
- TLS Web Client Authentication
- Code Signing
- E-mail Protection
- Time Stamping
- Microsoft Individual Code Signing
- Microsoft Commercial Code Signing
- Microsoft Trust List Signing
- Microsoft Server Gated Crypto
- Microsoft Encrypted File System

Specify in the Key Usage:

- Critical
- Certificate Sign: Enable this certificate to sign other certificates (primary purpose of a CA certificate).
- CRL Sign: Enable this certificate to sign the Certificate Revocation List, defining the cancelled certificates.

Click Next.


xca

Certificate Netscape extensions

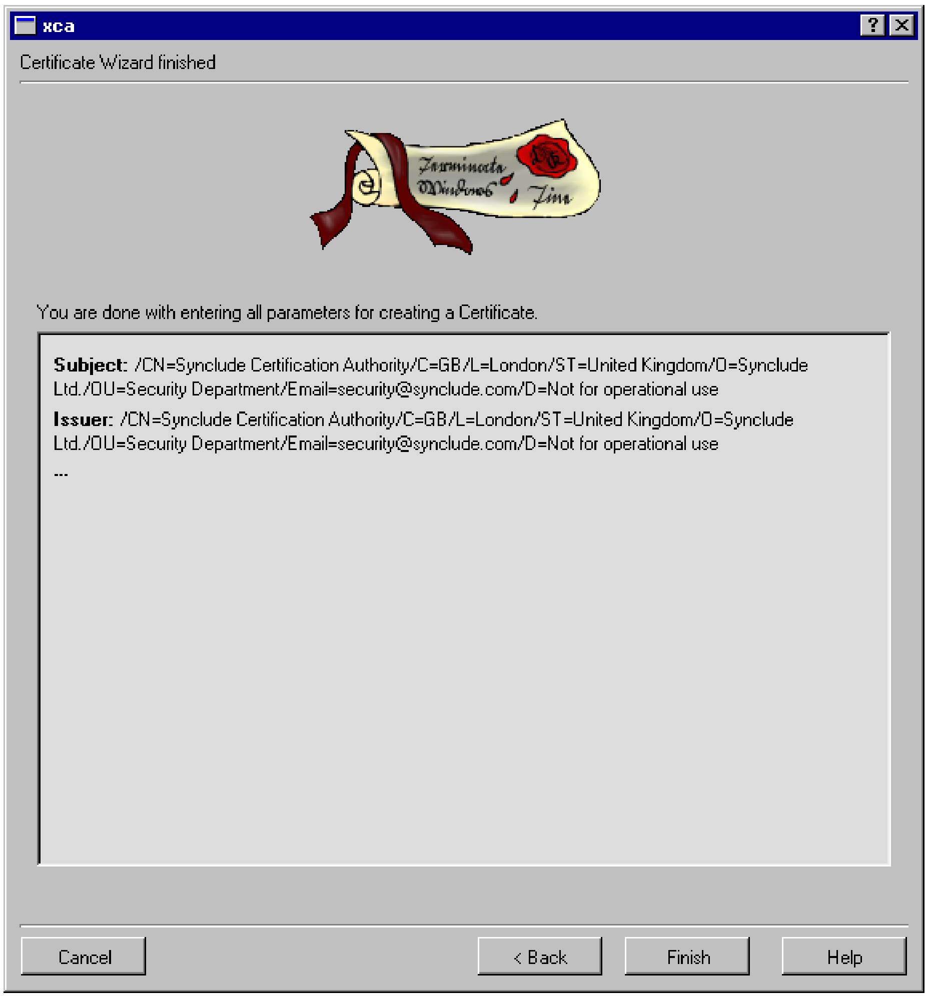
SSL Client
SSL Server
S/MIME
Object Signing
SSL CA
S/MIME CA
Object Signing CA

Base URL	
Revokation URL	
CA Revokation URL	
Certificate renewal URL	
CA policy URL	
SSL server name	
Comment	

Select in the Netscape Extensions:

- SSL CA: If you want to use the same CA certificate to generate other certificates used for SSL communication.
- S/MIME CA: Select this to enable this certificate to sign S/MIME certificates.
- Object Signing CA: Select this to enable this certificate to sign mobile code (e.g. Java applets).

You may leave the rest blank, or fill the entries depending on your security policy. Click Next.



Certificate Wizard finished

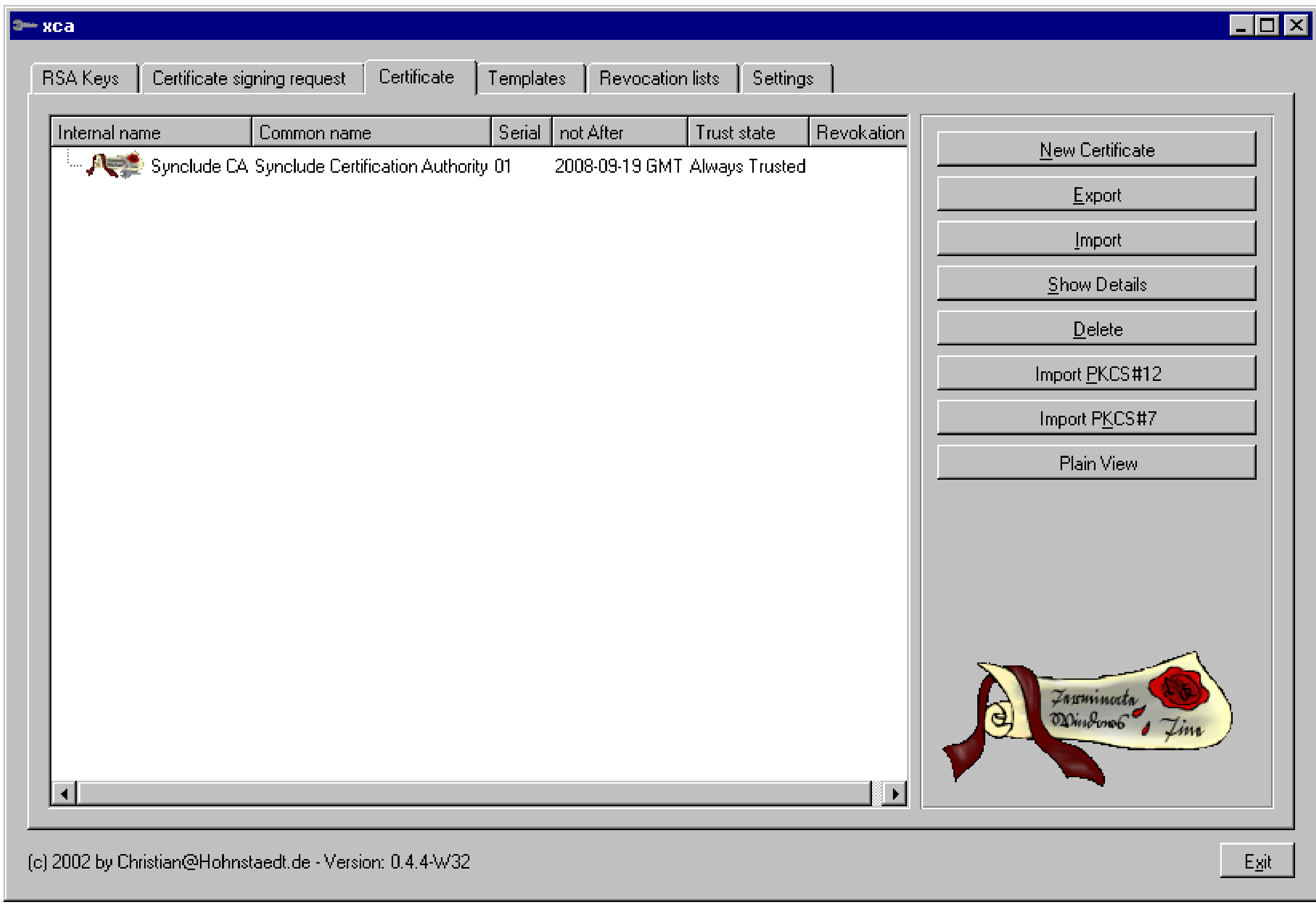
You are done with entering all parameters for creating a Certificate.

Subject: /CN=Synclude Certification Authority/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=Security Department/Email=security@synclude.com/D=Not for operational use

Issuer: /CN=Synclude Certification Authority/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=Security Department/Email=security@synclude.com/D=Not for operational use

...

Review the information you have entered. If there are any mistakes, you may use the Back button to correct them. Click Finish.

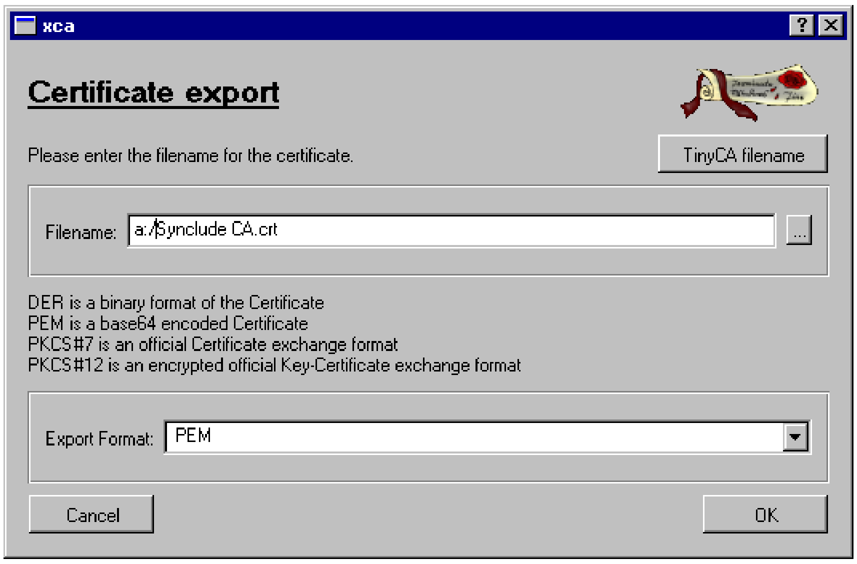


Internal name	Common name	Serial	not After	Trust state	Revokation
Synclude CA	Synclude Certification Authority	01	2008-09-19 GMT	Always Trusted	

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

You have created a CA certificate capable of signing Certificate Signing Requests and creating certificates for people in your organisation.

7.4.5. Export of CA certificate



Please enter the filename for the certificate.

Filename: a:/Synclude CA.crt

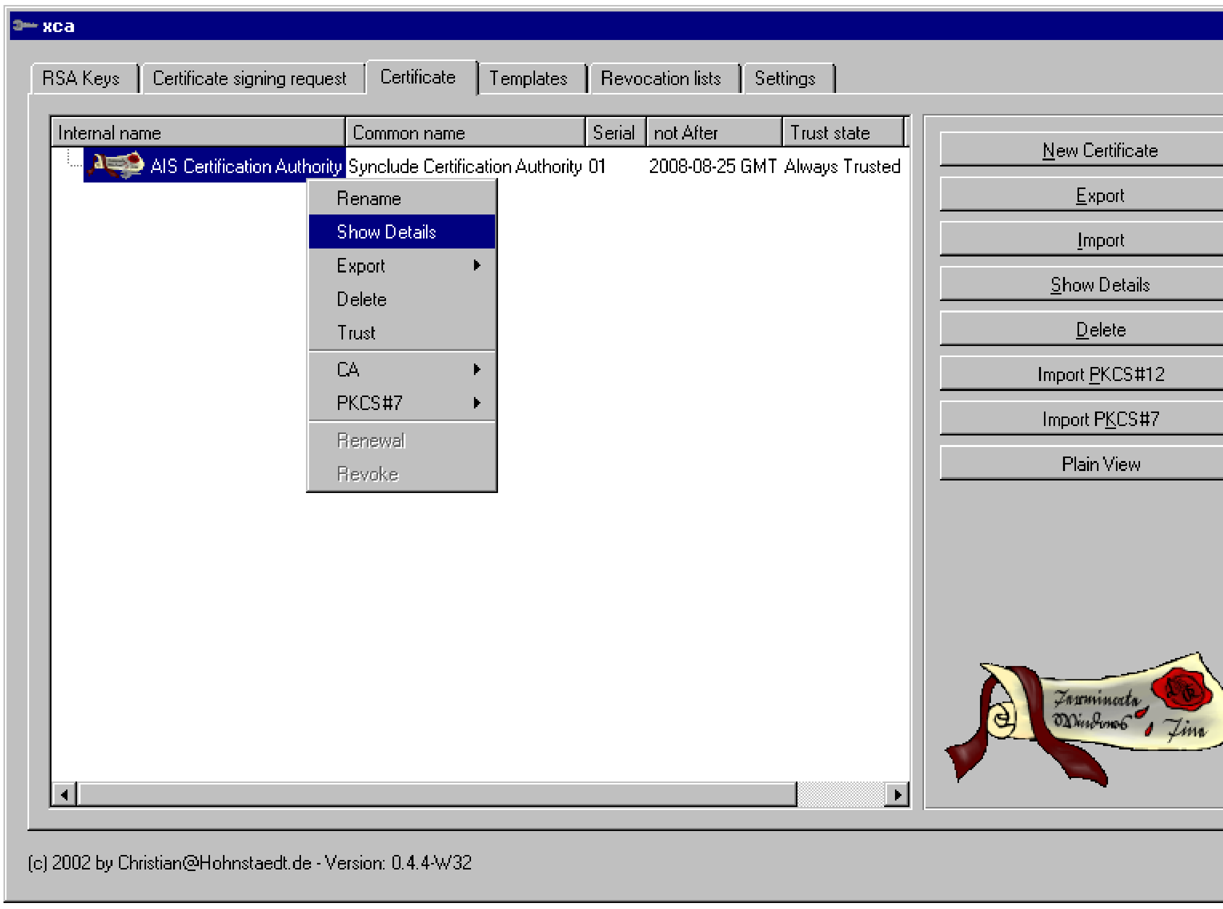
DER is a binary format of the Certificate
PEM is a base64 encoded Certificate
PKCS#7 is an official Certificate exchange format
PKCS#12 is an encrypted official Key-Certificate exchange format

Export Format: PEM

You need to export the CA certificate in order for other people to use it. Click on the certificate, and click on the Export button on the right. You are presented with the export interface. Enter:

- Destination directory and name of the exported certificate file.
- Export format: choose PEM.

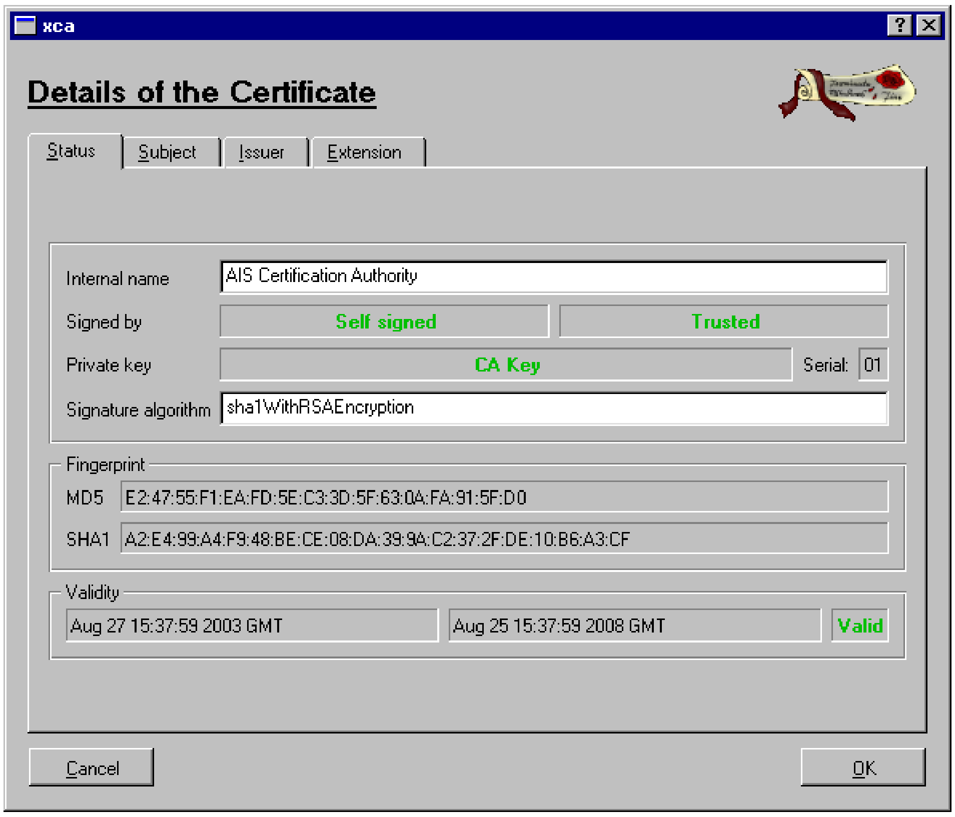
You also need the associated key fingerprint for end-users to be able to check the validity of the certificate.



Internal name	Common name	Serial	not After	Trust state
AIS Certification Authority	Synclude Certification Authority	01	2008-08-25 GMT	Always Trusted

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Right-click on the CA certificate and select Show details.



Internal name: AIS Certification Authority
Signed by: Self signed | Trusted
Private key: CA Key | Serial: 01
Signature algorithm: sha1WithRSAEncryption

Fingerprint
MD5: E2:47:55:F1:EA:FD:5E:C3:3D:5F:63:0A:FA:91:5F:D0
SHA1: A2:E4:99:A4:F9:48:BE:CE:08:DA:39:9A:C2:37:2F:DE:10:B6:A3:CF

Validity
Aug 27 15:37:59 2003 GMT | Aug 25 15:37:59 2008 GMT | Valid

Note down the MD5 fingerprint.

Send the exported CA certificate to the person creating the signing key. You now have to wait for that person to send their Certificate Signing Request. This file will also be sent to all end-users to validate the signed eAIP packages sent by your organisation, as well as its associated fingerprint.

7.5. Creating a Certificate Signing Request

7.5.1. Introduction

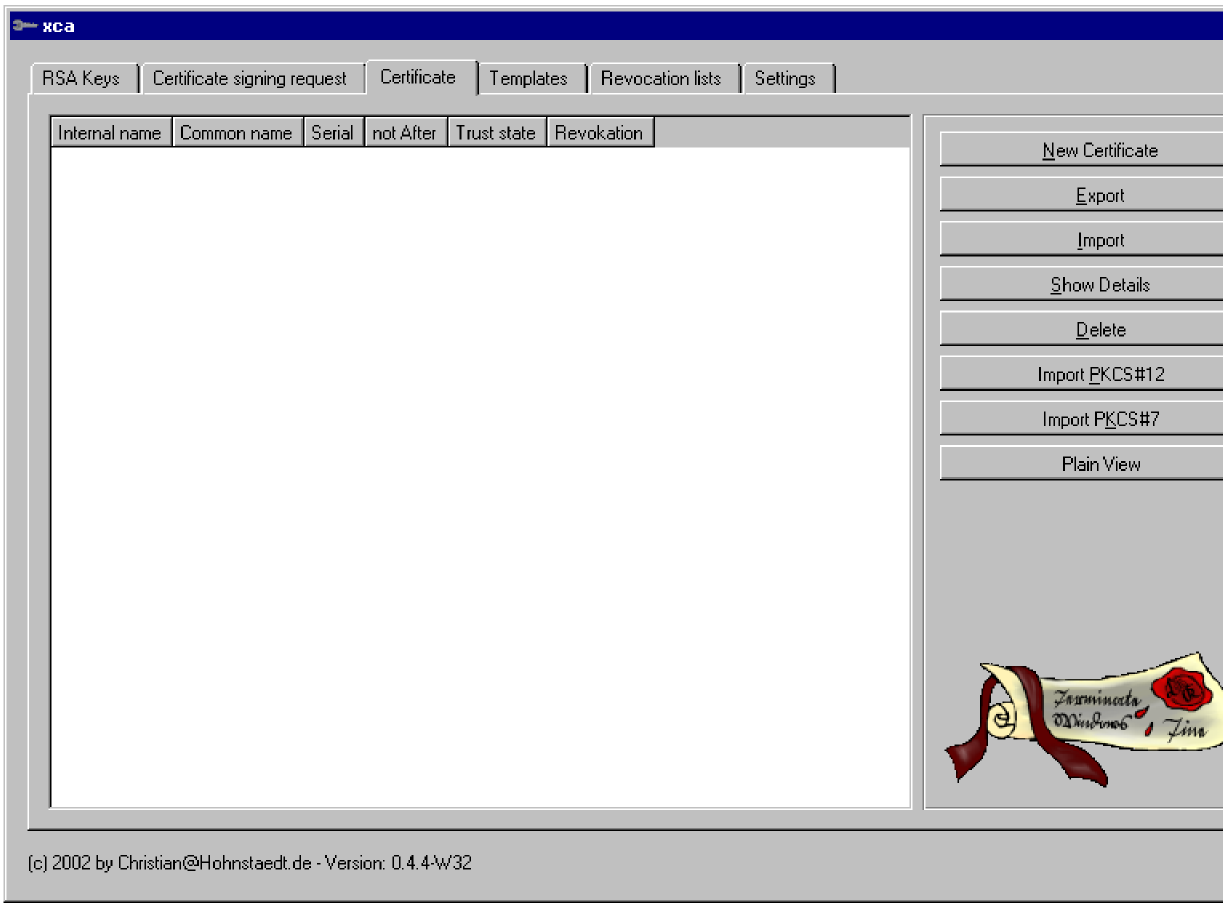
Creating a Certificate Signing Request is done by the actual person doing the signing of the eAIP packages. The steps are:

1. Install necessary software
2. Import CA certificate
3. Generate Certificate Signing Request
4. Export Certificate Signing Request, send to CA

7.5.2. Install necessary software

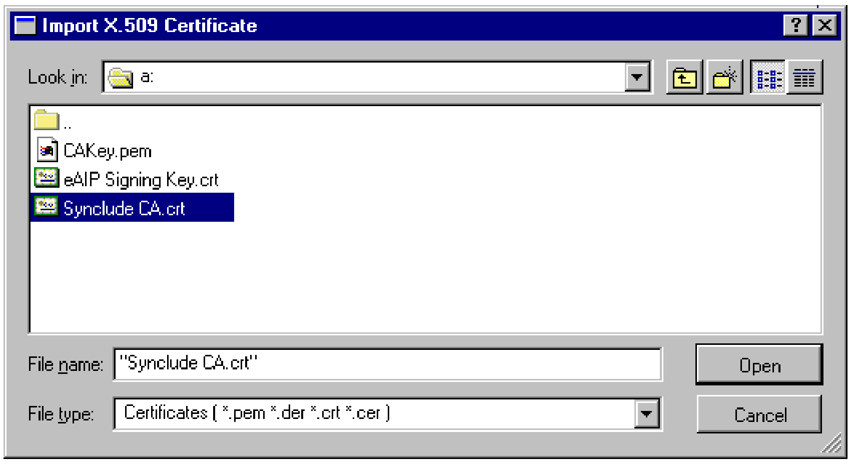
The steps are similar to those for the security administrator.

7.5.3. Import CA certificate



(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

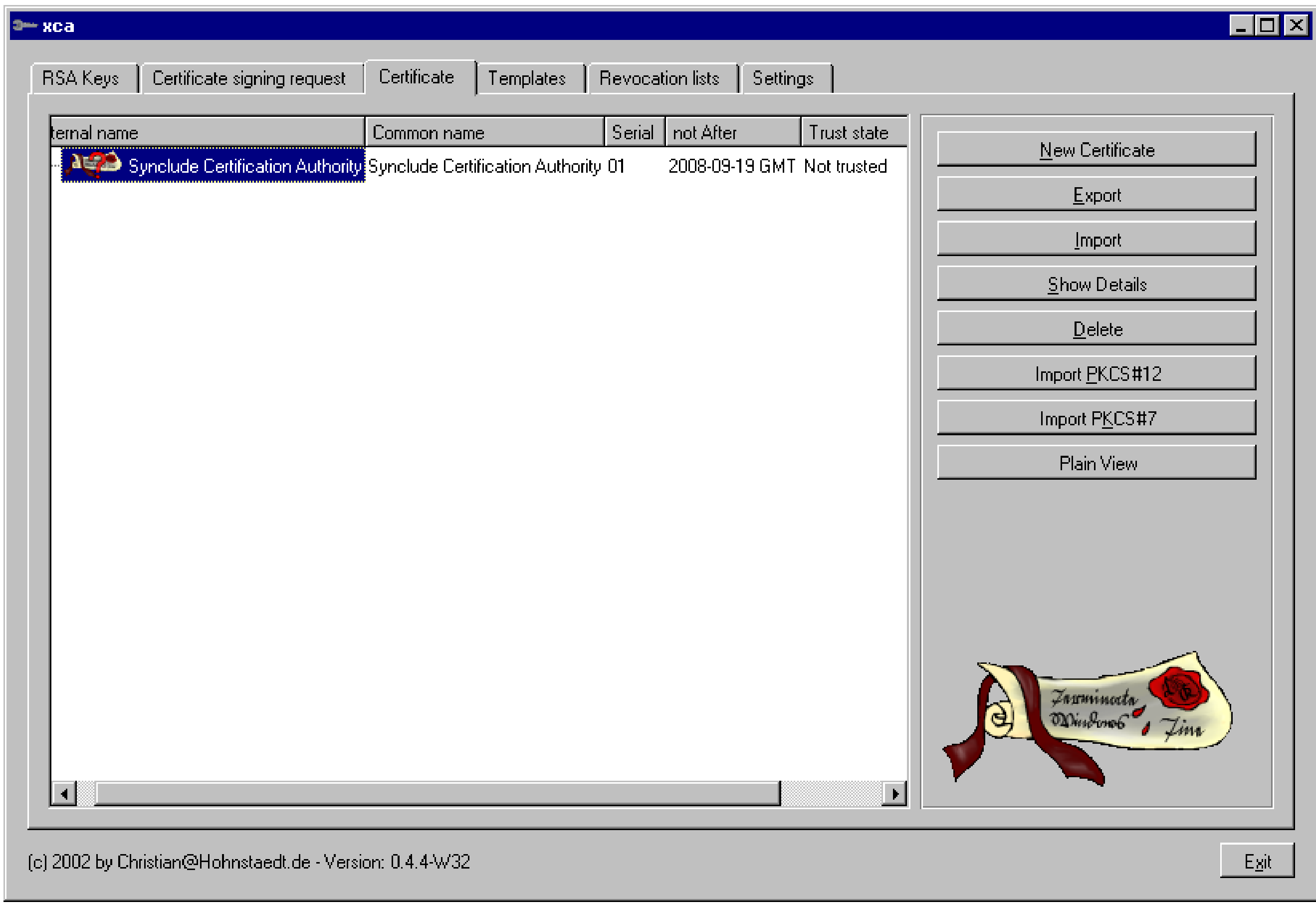
On the main window, click on the Certificate tab at the top, then on the Import button.



File name: "Synclude CA.crt"

File type: Certificates (*.pem *.der *.crt *.cer)

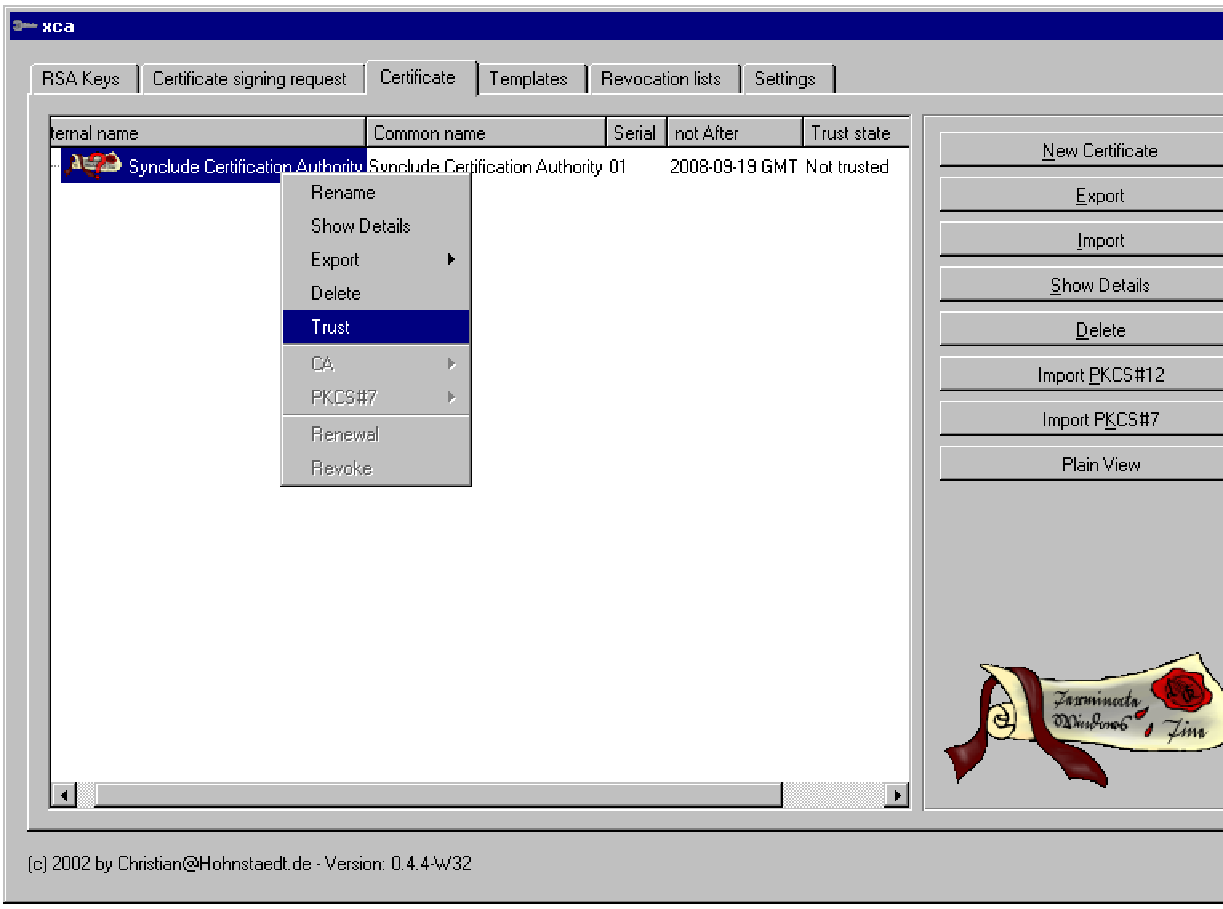
Navigate to the location where the certificate file sent to you is stored. Click Open.



ternal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Not trusted

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

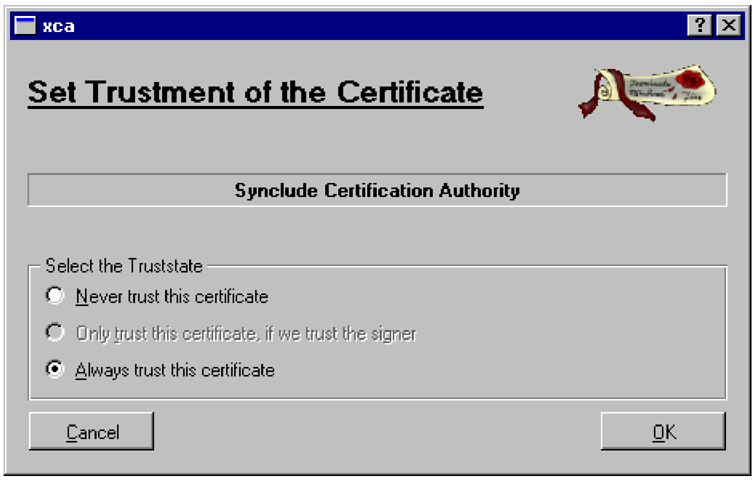
The certificate appears in the list. Notice that its trust state is Not Trusted.



ternal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Not trusted

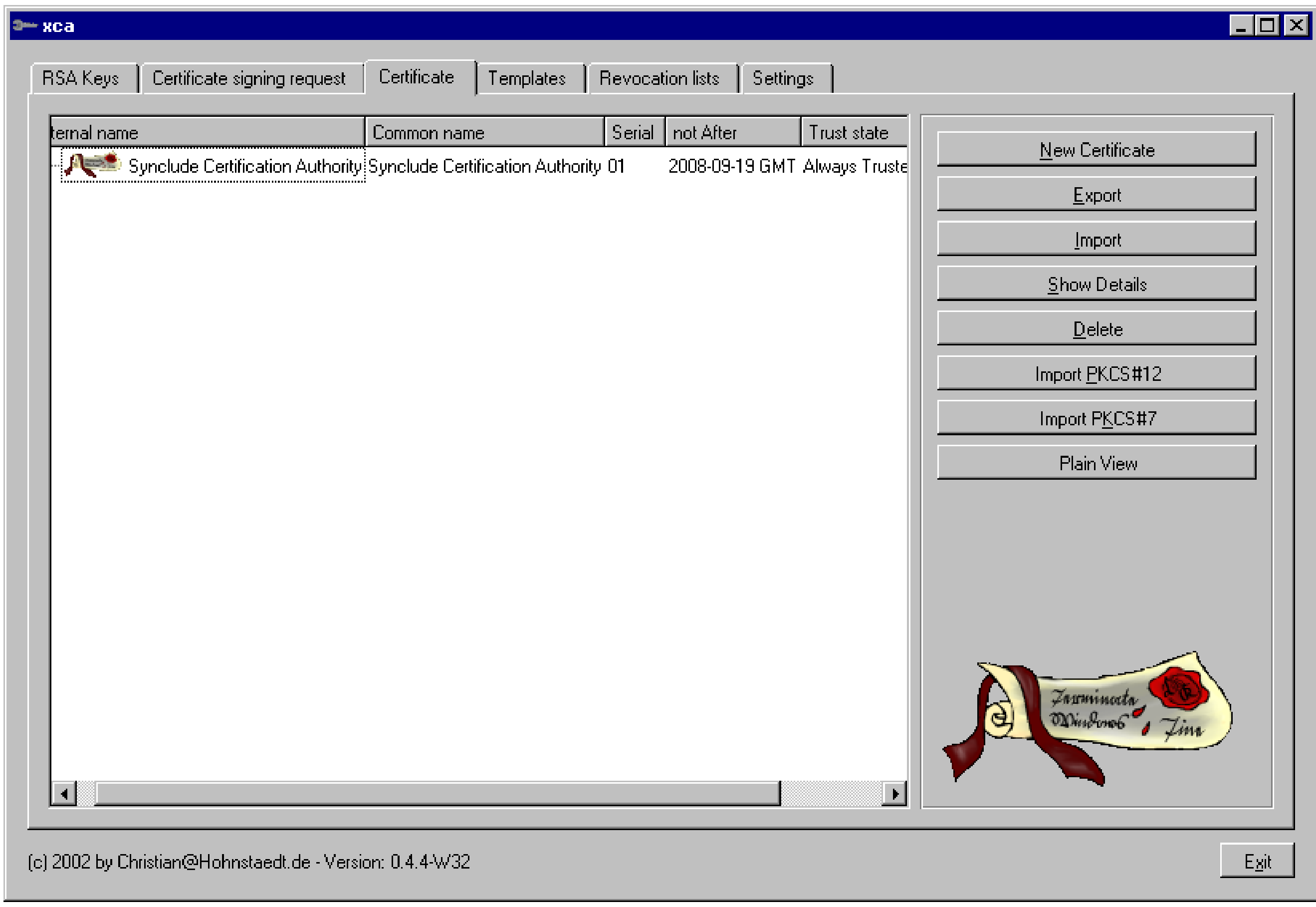
(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Right-click on the certificate, and select the Trust option.


xca
Set Trustment of the Certificate
Synclude Certification Authority
Select the Truststate
Cancel
OK

( ) Never trust this certificate
( ) Only trust this certificate, if we trust the signer
[✓ Always trust this certificate]

The trust setting dialogue appears. Select Always trust this certificate. Click OK.

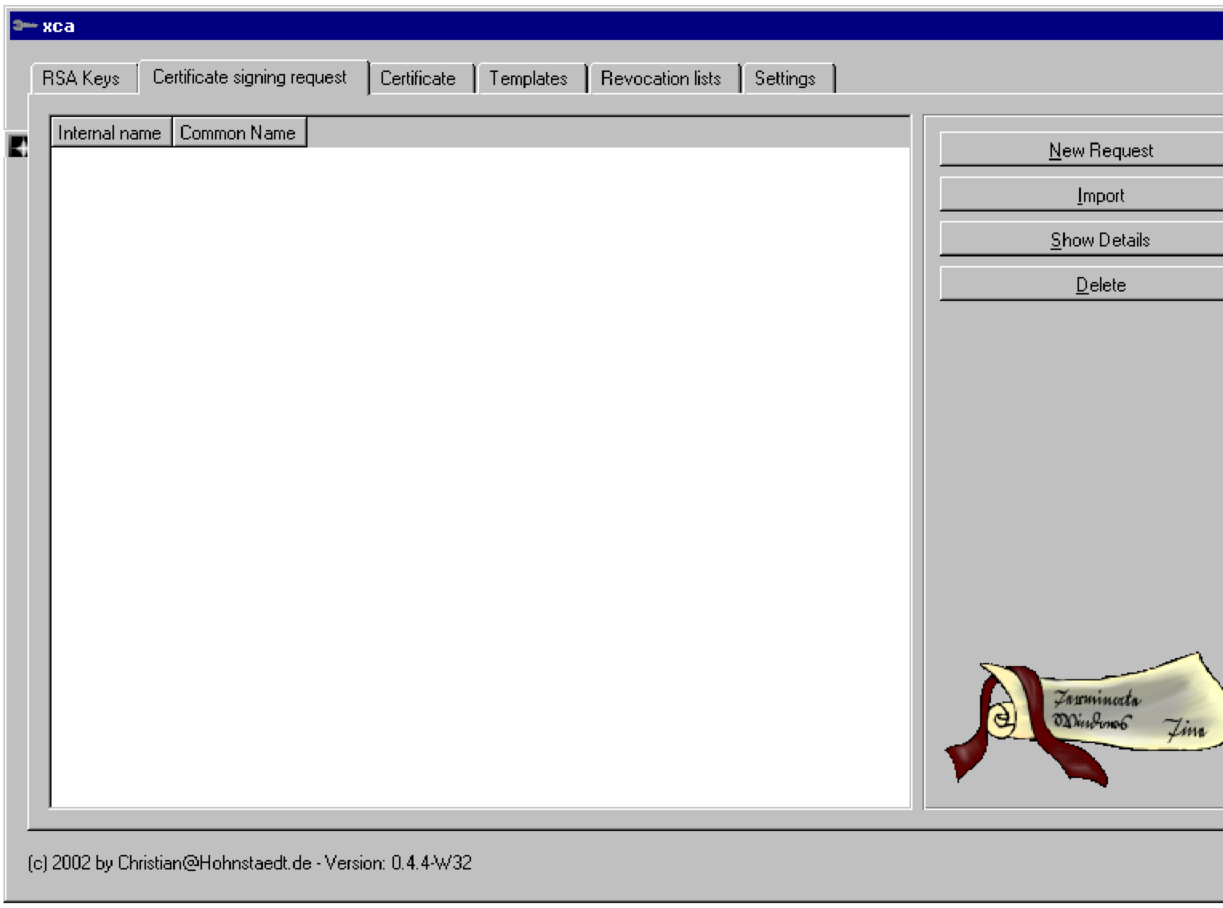


ternal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Always Truste

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

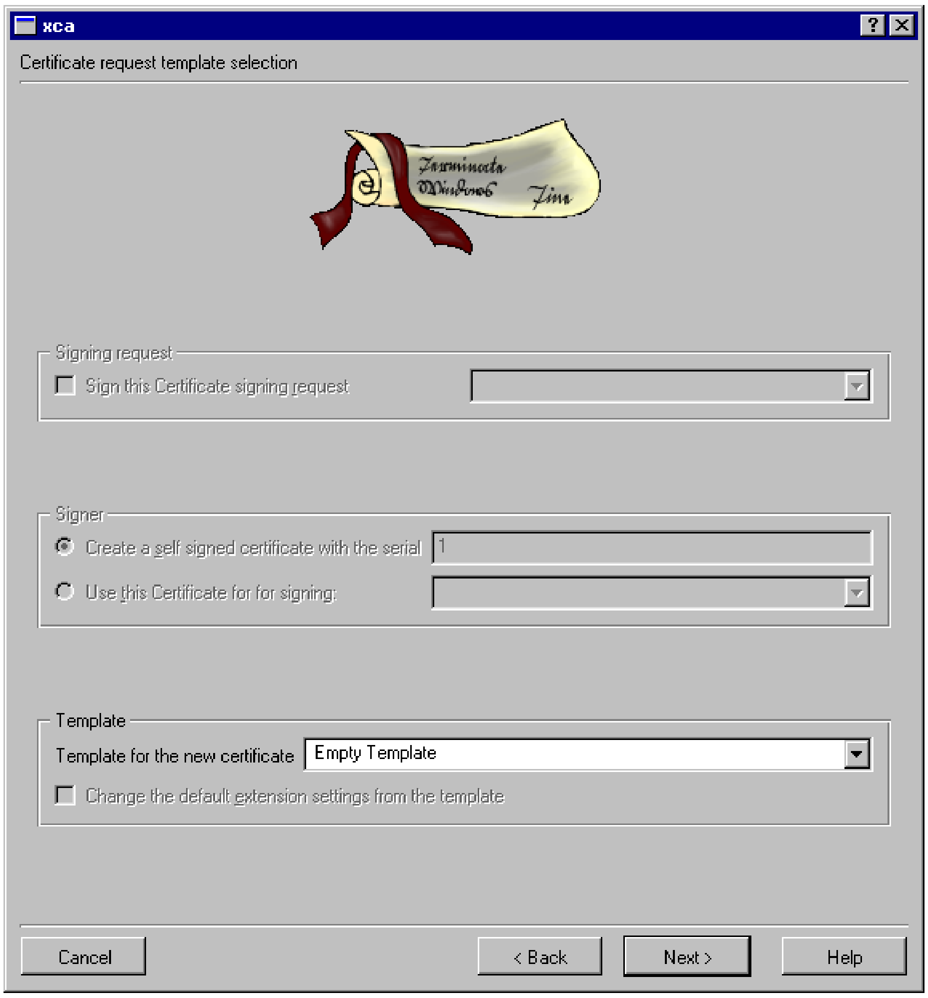
You notice that the trust setting of the certificate has changed.

7.5.4. Generate Certificate Signing Request



(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Click on the Certificate signing request tab on top, then on New Request. The Certificate request wizard opens up. Click Next.


xca

Certificate request template selection

Signing request

☐ Sign this Certificate signing request

Signer

● Create a self signed certificate with the serial 1

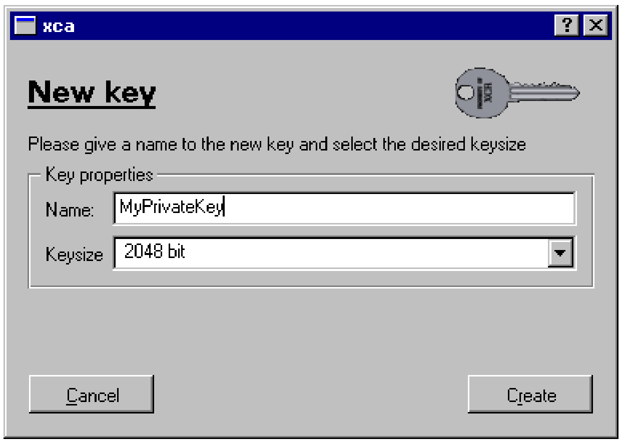
○ Use this Certificate for for signing:

Template

Template for the new certificate: Empty Template

☐ Change the default extension settings from the template

Leave the default values. Click Next.


xca

New key

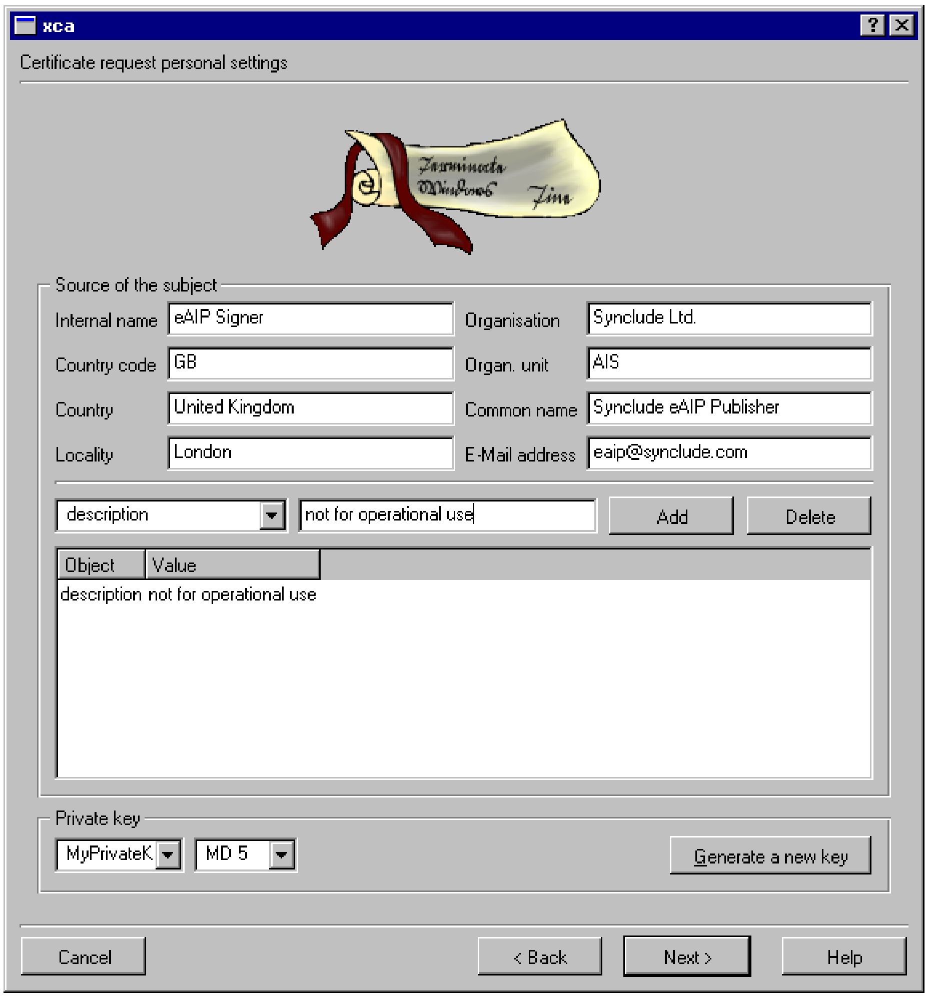
Please give a name to the new key and select the desired keysize

Key properties

Name: MyPrivateKey

Keysize: 2048 bit

Enter the key name you wish, and choose the key strength you need (at least 2048 recommended). Click Create to start the key generation.



Internal name: eAIP Signer
Organisation: Synclude Ltd.
Country code: GB
Organ. unit: AIS
Country: United Kingdom
Common name: Synclude eAIP Publisher
Locality: London
E-Mail address: eaip@synclude.com

description: not for operational use

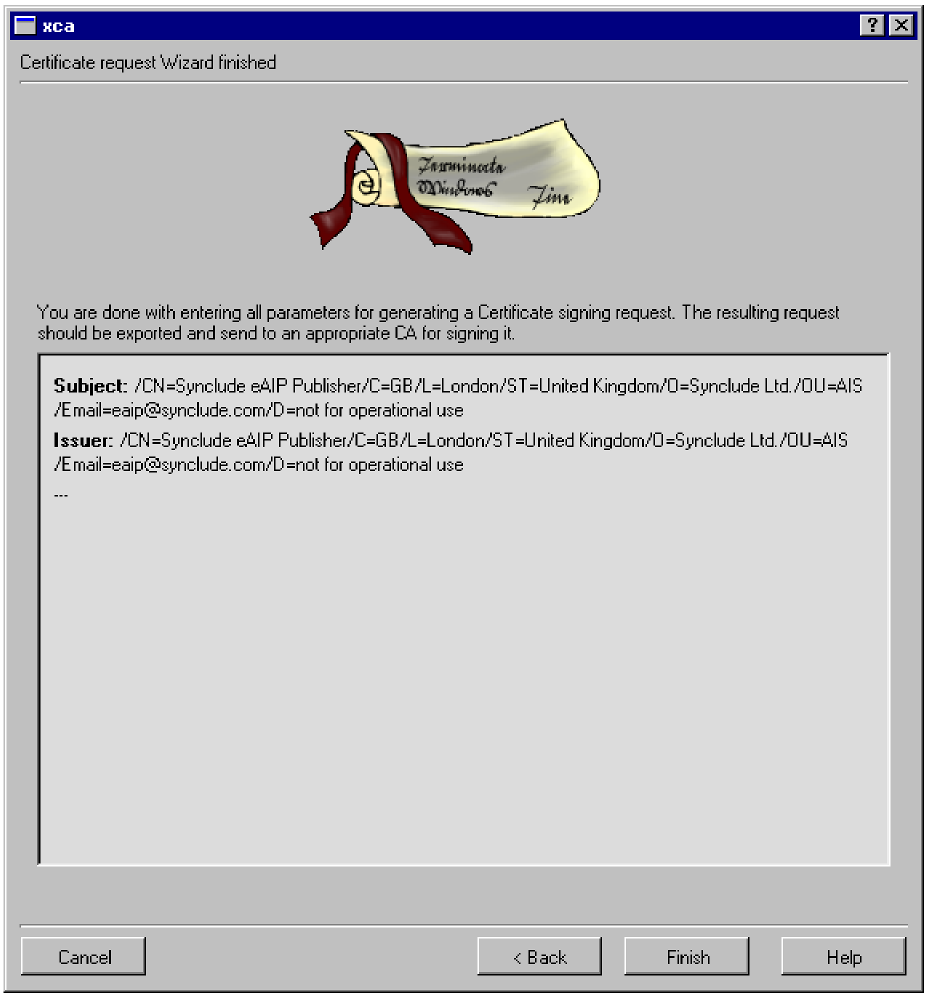
Object	Value
description	not for operational use

Private key: MyPrivateK, MD 5

Enter the information relative to your organisation:

- Internal name: A name by which you will recognise this certificate in the XCA software.
- Organisation: The full name of your organisation.
- Country Code: The ISO country code of your organisation.
- Organ. Unit: The unit of the organisation which is responsible for the management of the certificates,
- Country: The country of your organisation.
- Common Name: Your name, or the organisation unit if this certificate will be shared.
- Locality: The city of your organisation.
- E-Mail address: Your email address.

At the bottom, in Private Key, make sure name of the key you just generated is selected. Click Next.



Certificate request Wizard finished

You are done with entering all parameters for generating a Certificate signing request. The resulting request should be exported and send to an appropriate CA for signing it.

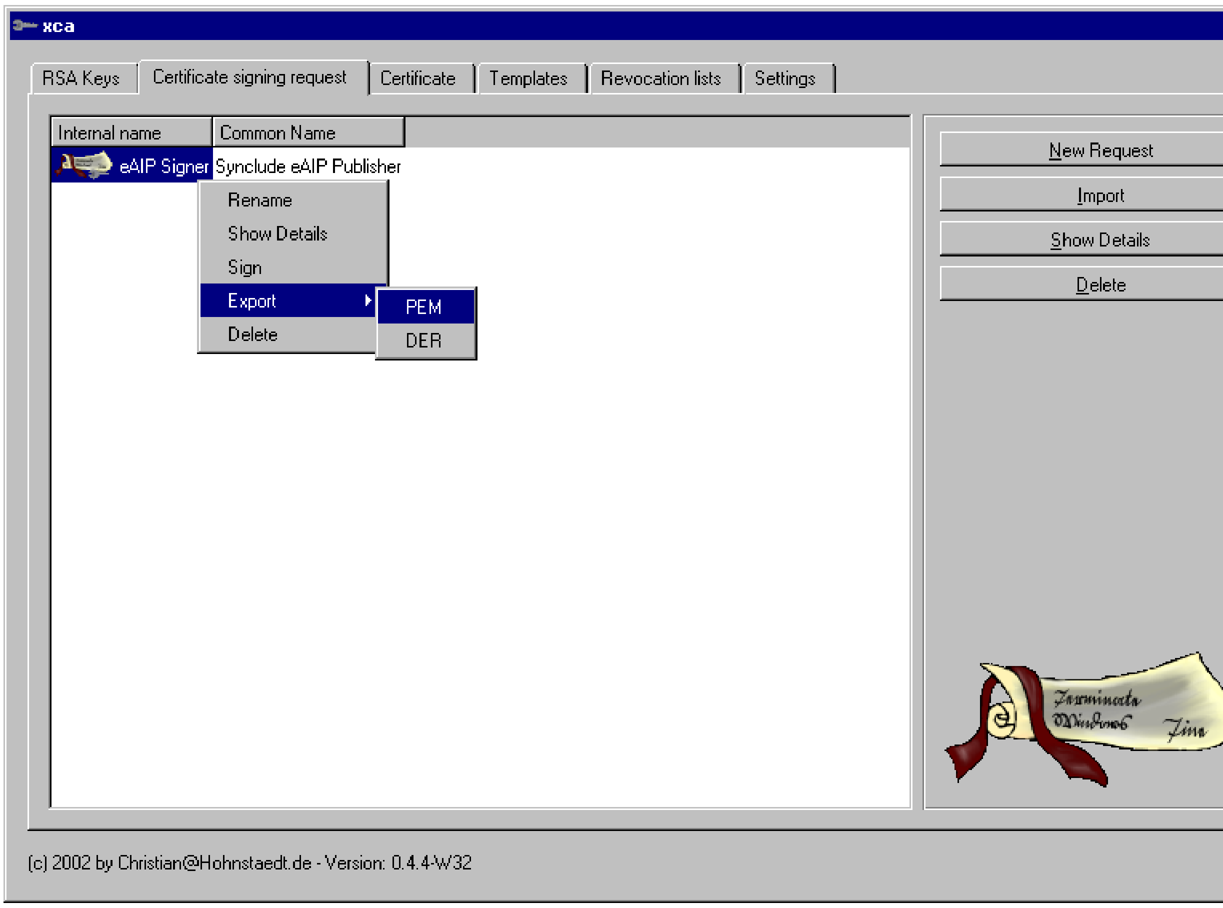
Subject: /CN=Synclude eAIP Publisher/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=AIS /Email=eaip@synclude.com/D=not for operational use

Issuer: /CN=Synclude eAIP Publisher/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=AIS /Email=eaip@synclude.com/D=not for operational use

---

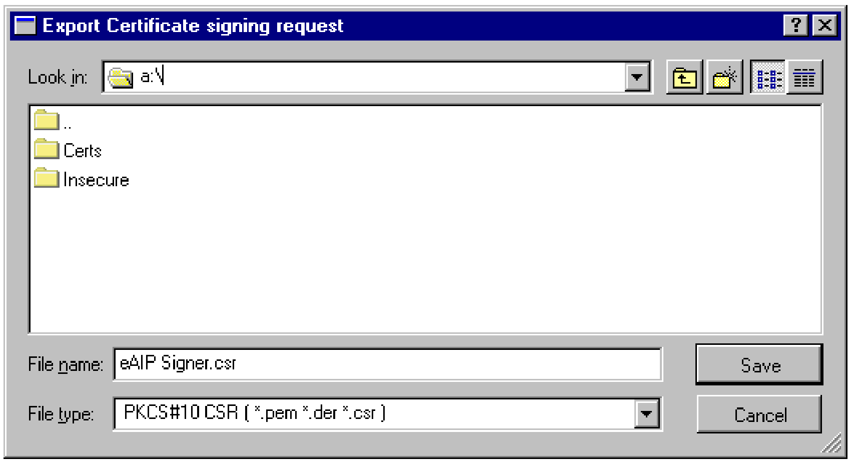
Review carefully the information you have entered. If there is any mistake, use the Back button to correct them. Click Finish.

7.5.5. Export Certificate Signing Request, send to CA



(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

You can see the Signing request you just generated in your list. Right-click on it and select Export, and PEM as format.



File name: eAIP Signer.csr

File type: PKCS#10 CSR (*.pem *.der *.csr)

Navigate to the location where you want to save this file. Click Save. You send this file to the person responsible for the CA.

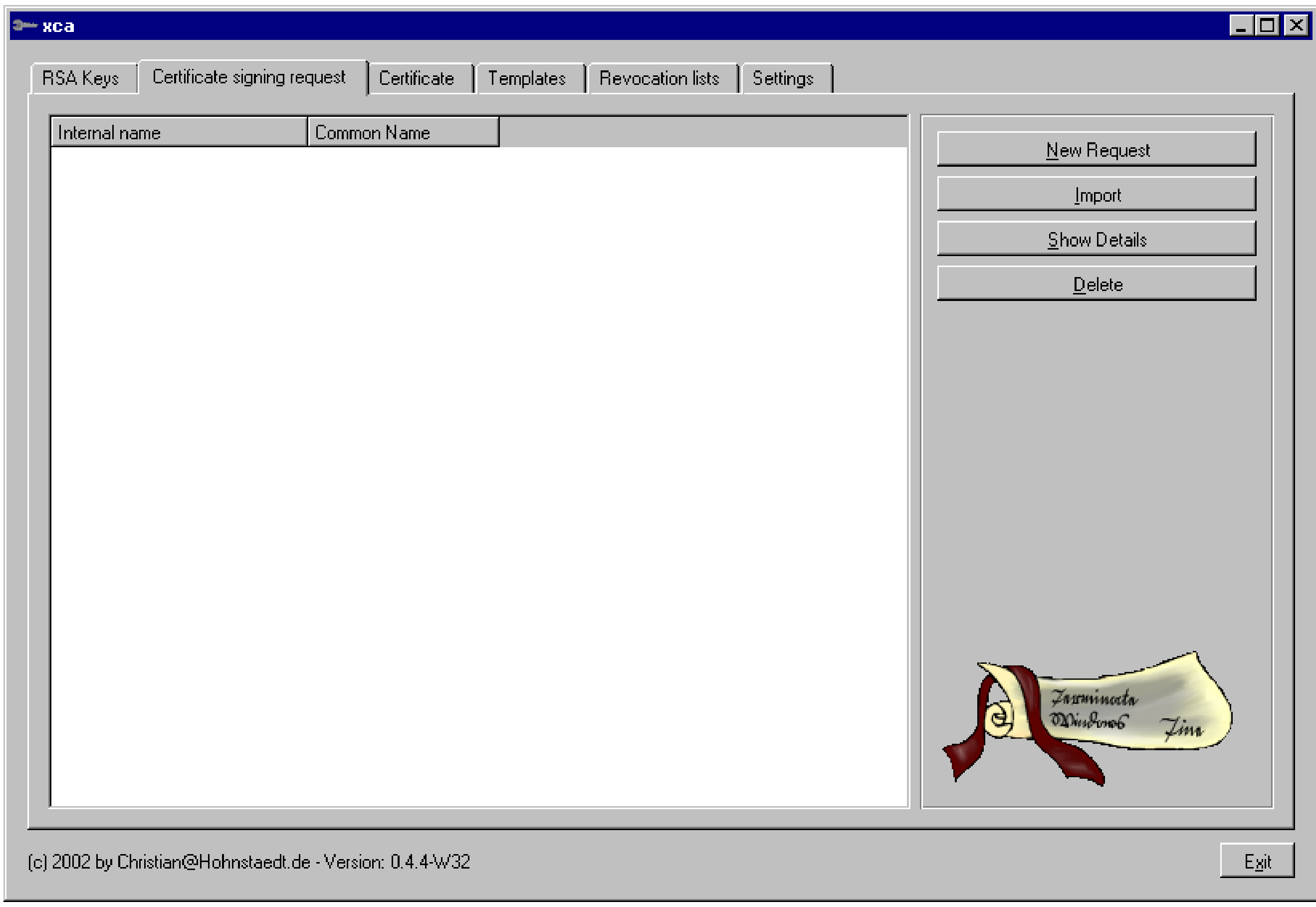
7.6. Signing the Certificate Signing Request to create Certificate

7.6.1. Introduction

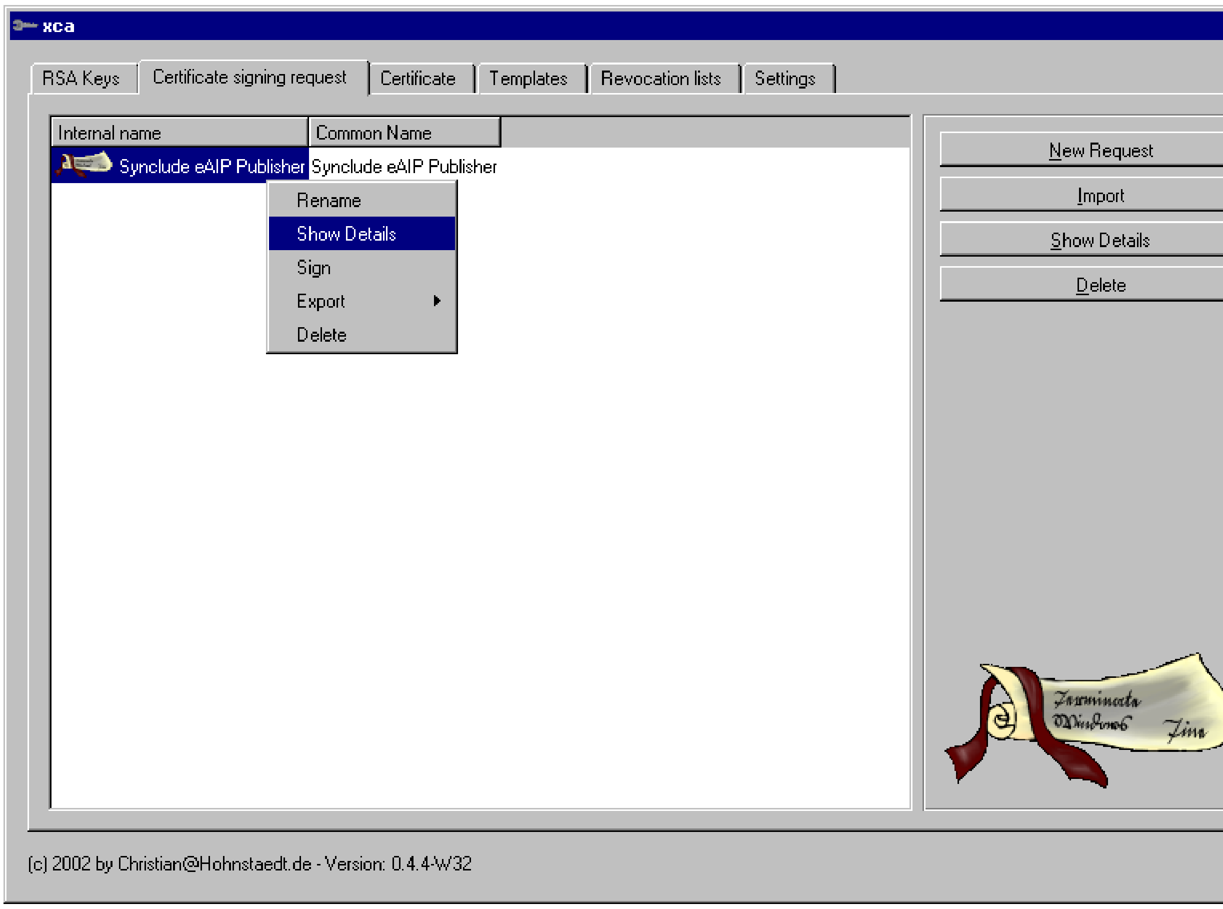
Signing the Certificate Signing Request to create Certificate step is done by the CA. The steps are:

1. Import the Certificate Signing Request
2. Sign the Certificate Signing Request to create Certificate
3. Export and distribute certificate

7.6.2. Import the Certificate Signing Request



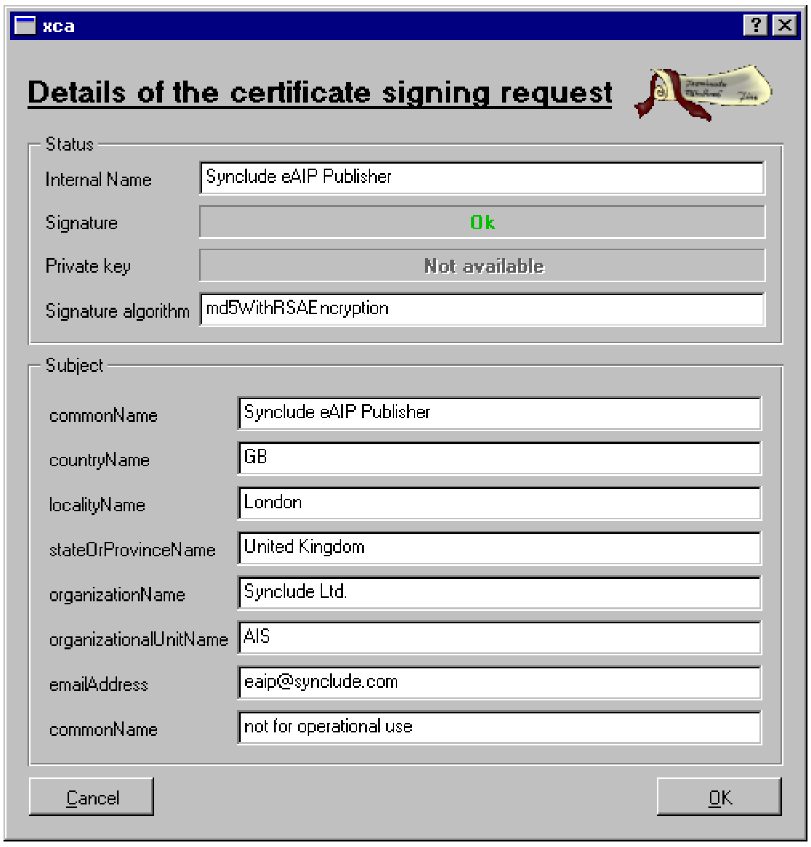
In the main interface, click on the Certificate signing request tab on the top, then on the Import button on the right. You will be presented with a file dialogue. Navigate to the file that the signing party sent to you.



Internal name	Common Name
Synclude eAIP Publisher	Synclude eAIP Publisher

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

The Signing Request appears in the list. Right-click on it and select Show Details.



Details of the certificate signing request

Status

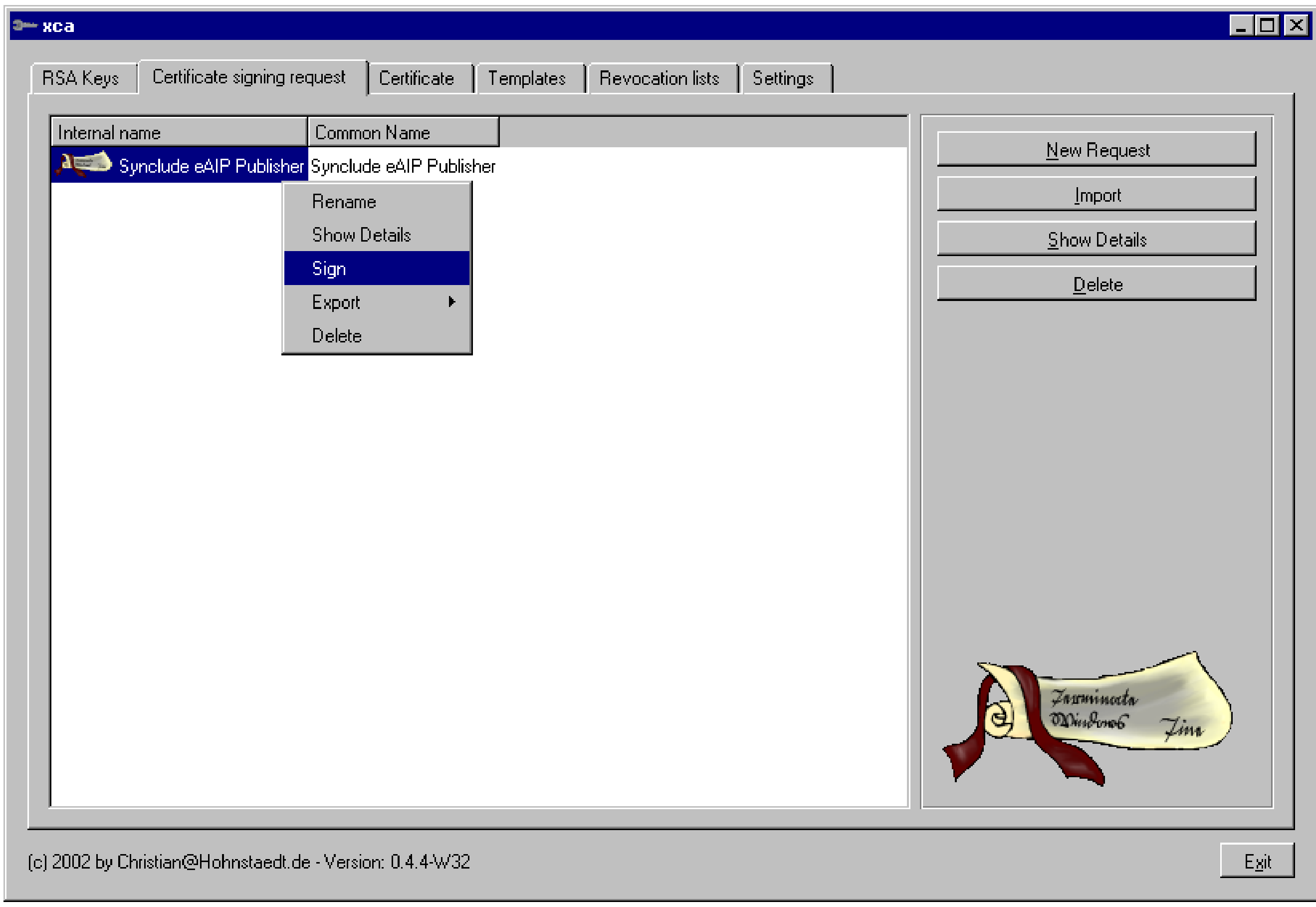
Internal Name	Synclude eAIP Publisher
Signature	Ok
Private key	Not available
Signature algorithm	md5WithRSAEncryption

Subject

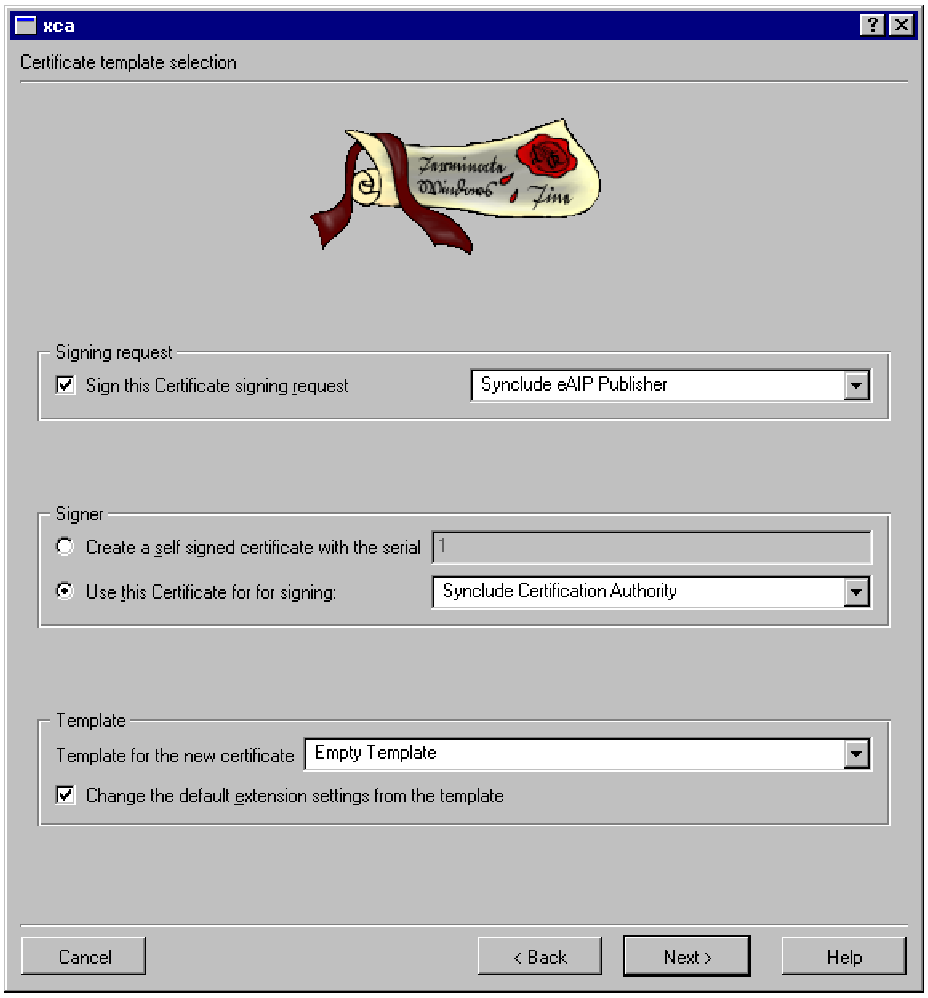
commonName	Synclude eAIP Publisher
countryName	GB
localityName	London
stateOrProvinceName	United Kingdom
organizationName	Synclude Ltd.
organizationalUnitName	AIS
emailAddress	eaip@synclude.com
commonName	not for operational use

Double-check that the information entered is valid. If there is any errors, have the person re-submit a new Signing Request. Click OK when satisfied.

7.6.3. Sign the Certificate Signing Request to create Certificate



Right-click on the signing request, select Sign. You are presented with the signing wizard. Click Next.



xca

Certificate template selection

Signing request

[✓] Sign this Certificate signing request: Synclude eAIP Publisher

Signer

( ) Create a self signed certificate with the serial: 1

(●) Use this Certificate for for signing: Synclude Certification Authority

Template

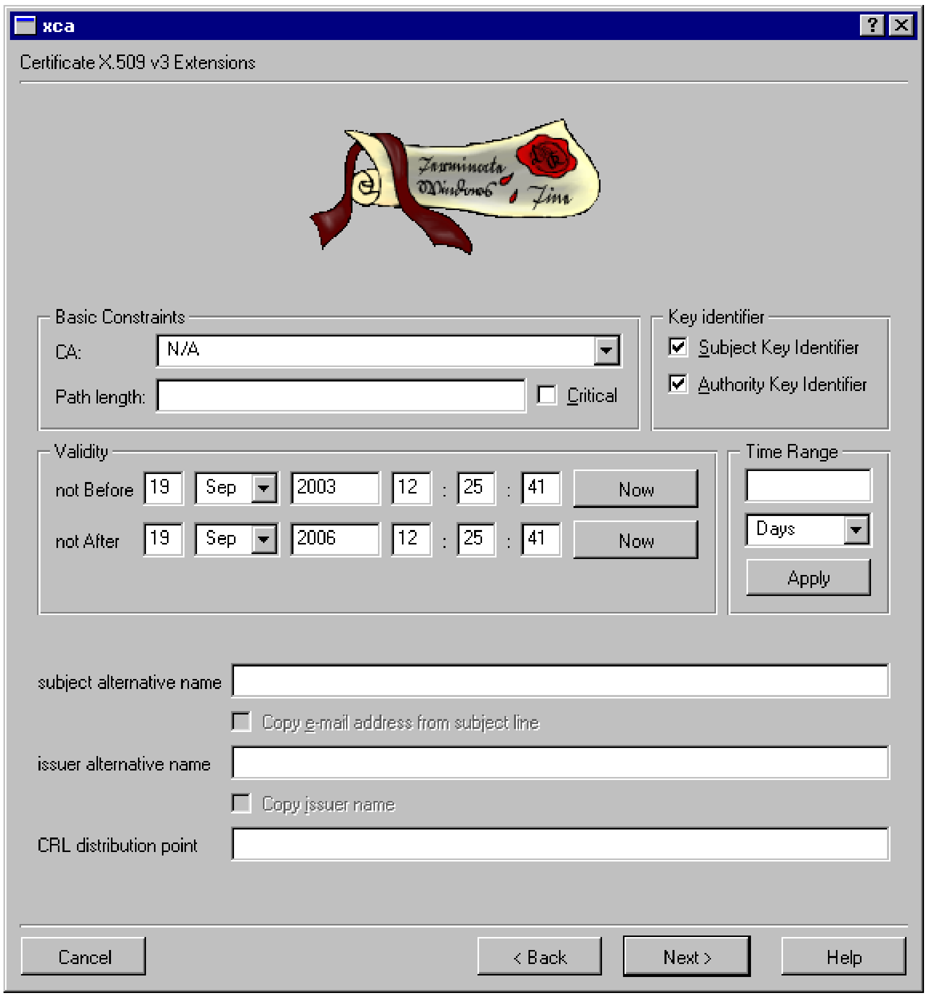
Template for the new certificate: Empty Template

[✓] Change the default extension settings from the template

Have the following selected:

- Sign this certificate request: Leave the selected request in the drop-down menu
- Use this certificate for signing: Select the CA certificate which is used for signing

Click on Next.



Certificate X.509 v3 Extensions

Basic Constraints

CA:	N/A
Path length:	

☐ Critical

Key identifier

☑ Subject Key Identifier
☑ Authority Key Identifier

Validity

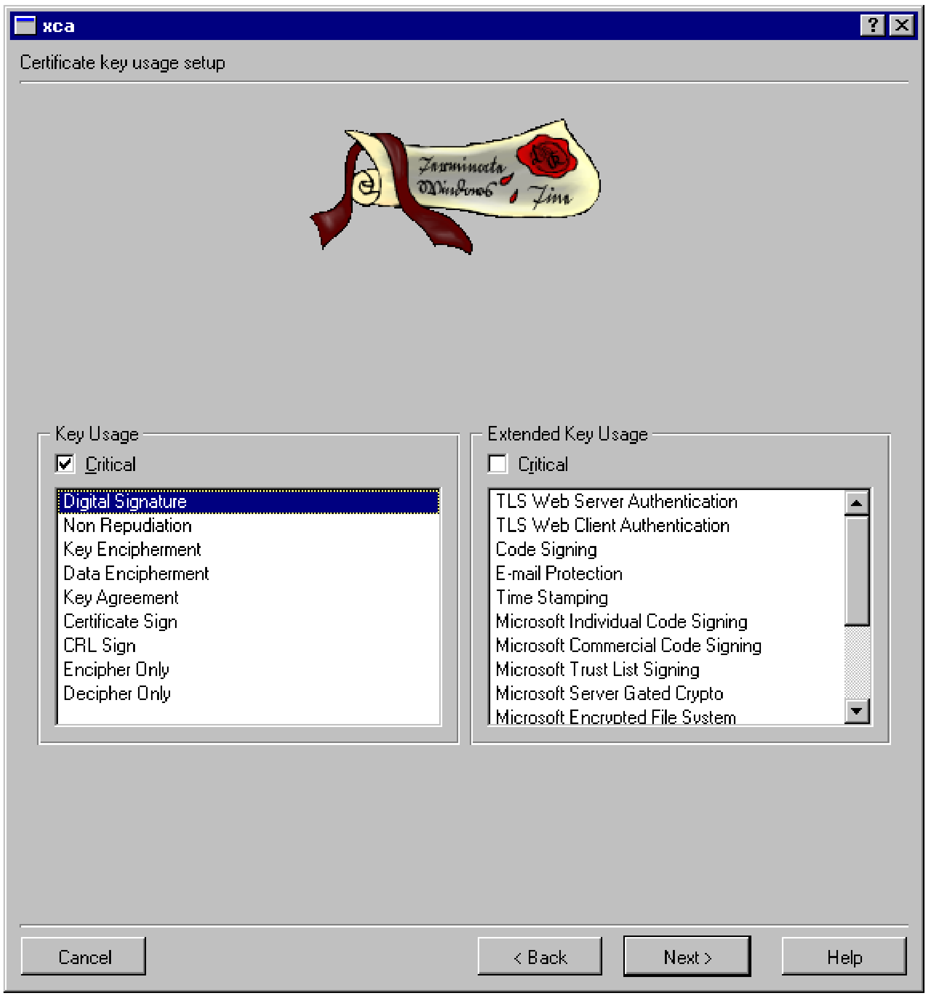
not Before	19	Sep	2003	12	25	41
not After	19	Sep	2006	12	25	41

Time Range

Days

subject alternative name	
	☐ Copy e-mail address from subject line
issuer alternative name	
	☐ Copy issuer name
CRL distribution point	

Fill in the Key identifier and the validity period. Please note that after a certificate has expired, another one must be regenerated. Click Next.


xca

Certificate key usage setup

Key Usage

[✓] Critical

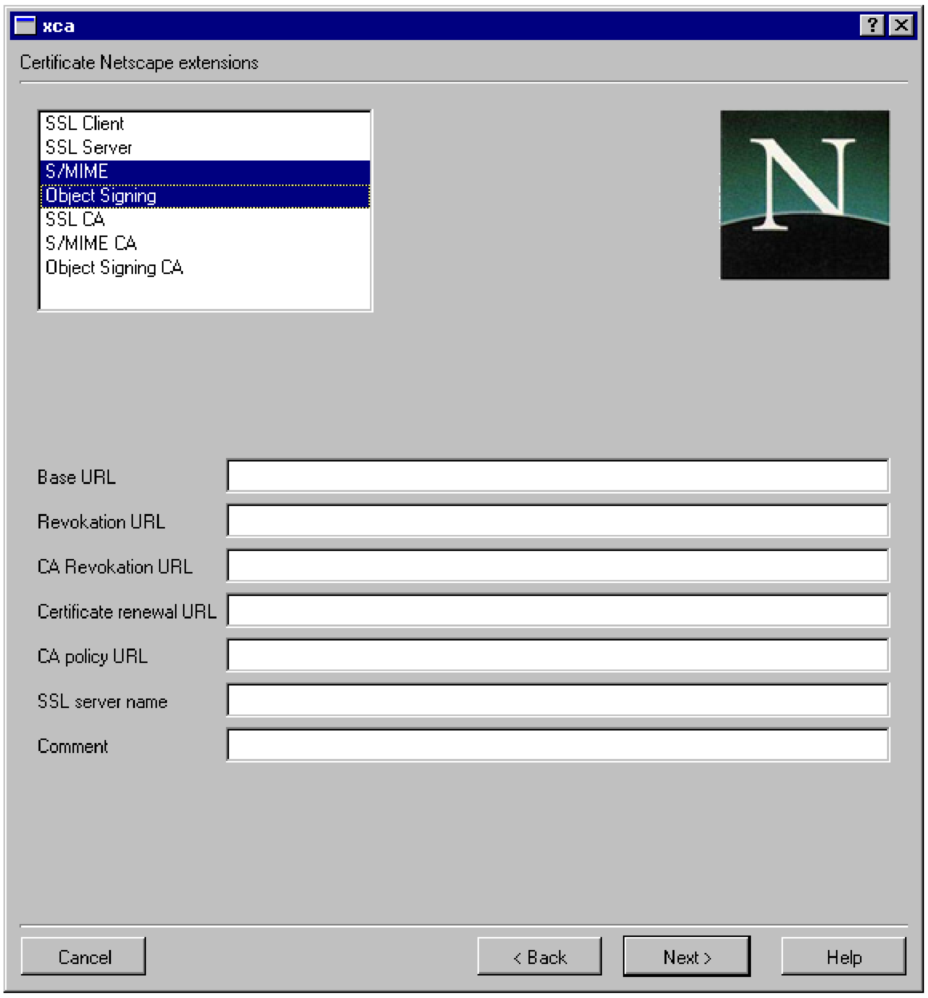
- **Digital Signature**
- Non Repudiation
- Key Encipherment
- Data Encipherment
- Key Agreement
- Certificate Sign
- CRL Sign
- Encipher Only
- Decipher Only

Extended Key Usage

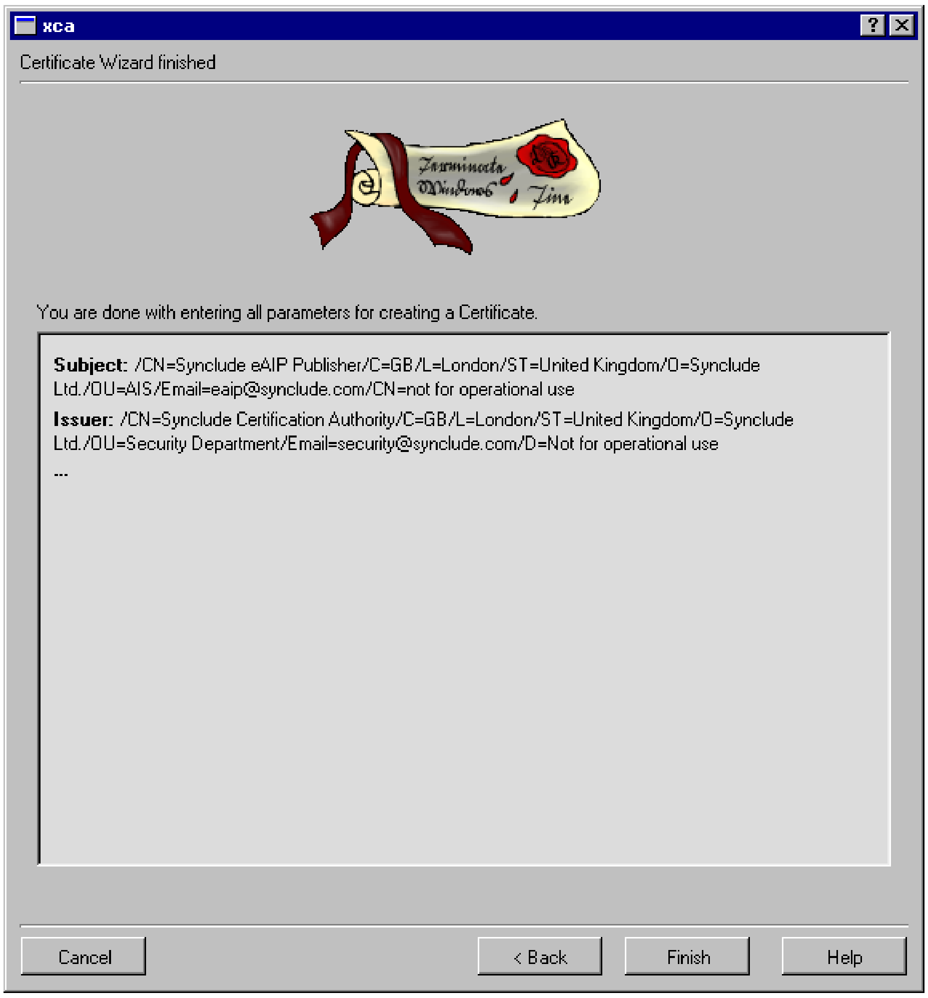
[ ] Critical

- TLS Web Server Authentication
- TLS Web Client Authentication
- Code Signing
- E-mail Protection
- Time Stamping
- Microsoft Individual Code Signing
- Microsoft Commercial Code Signing
- Microsoft Trust List Signing
- Microsoft Server Gated Crypto
- Microsoft Encrypted File System

Select in the Key Usage Critical and Digital Signature. Click Next.



If you wish to allow the use of this certificate for other purposes (e.g. SSL or secured email using S/MIME) you may select other options, depending on your security policy. Click Next.



You are done with entering all parameters for creating a Certificate.

Subject: /CN=Synclude eAIP Publisher/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=AIS/Email=eaip@synclude.com/CN=not for operational use

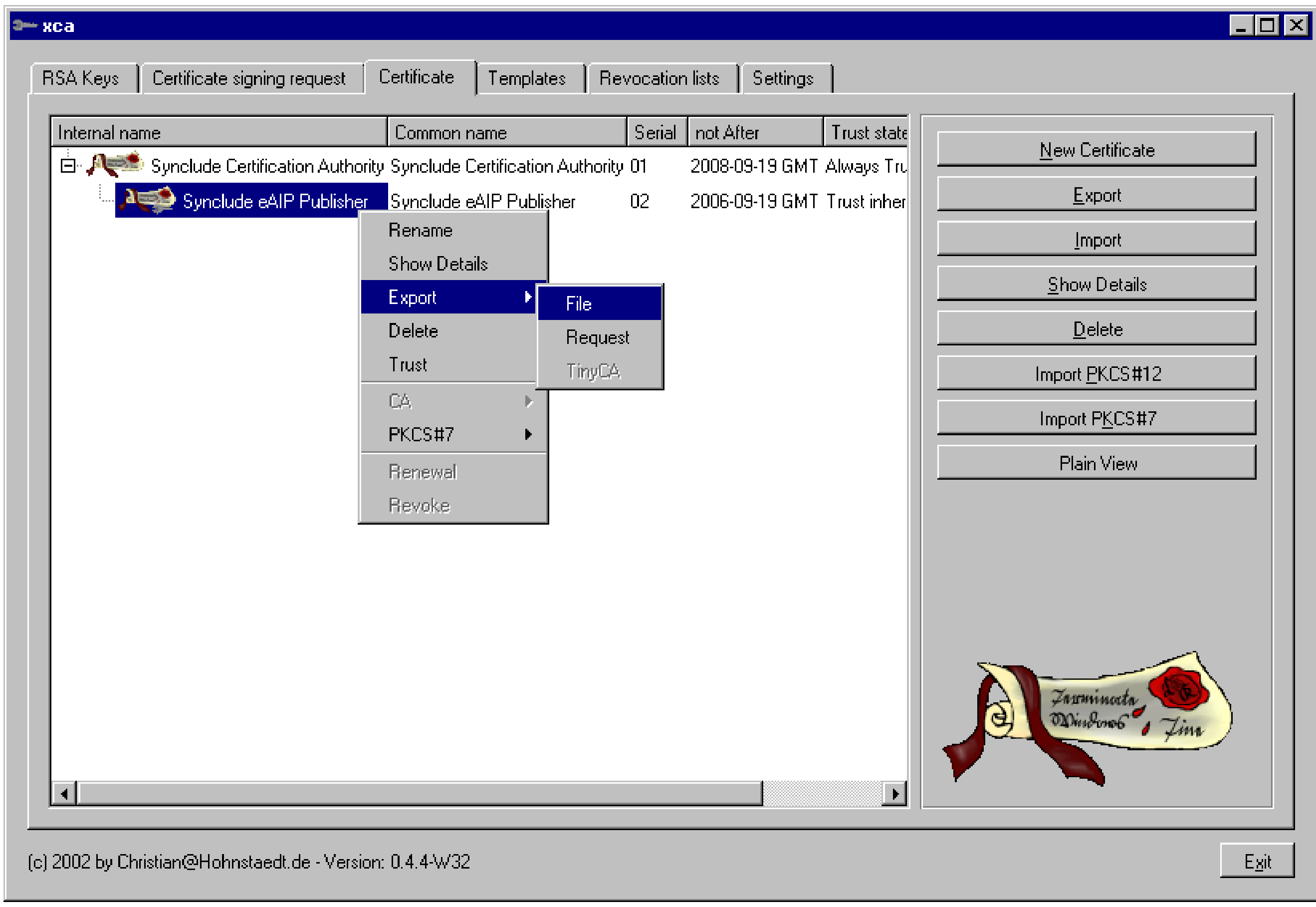
Issuer: /CN=Synclude Certification Authority/C=GB/L=London/ST=United Kingdom/O=Synclude Ltd./OU=Security Department/Email=security@synclude.com/D=Not for operational use

...

Check again the validity of the information. When satisfied, click Finnish.

7.6.4. Export the signing certificate

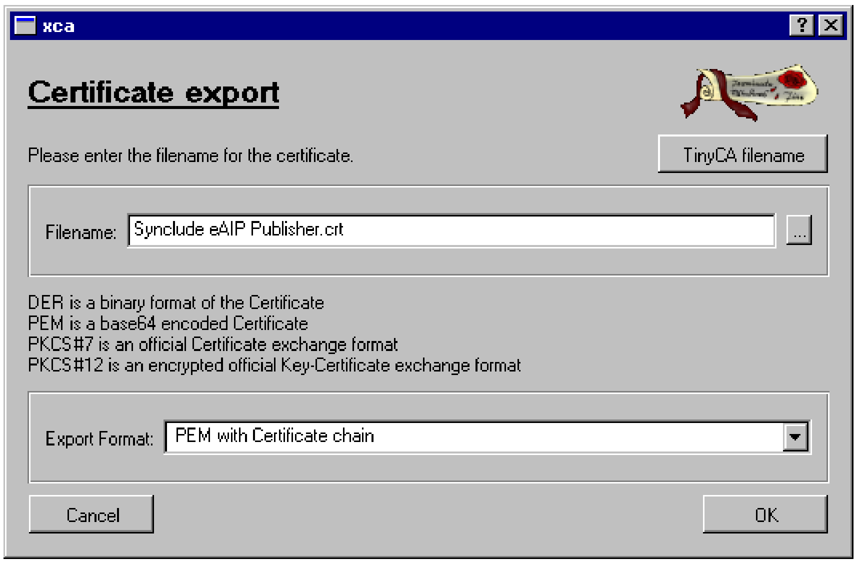
Export the signing certificate from XCA by saving it to a file:



Internal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Always Tru
Synclude eAIP Publisher	Synclude eAIP Publisher	02	2006-09-19 GMT	Trust inher

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Select the Certificate tab on top. Right-click on the signing certificate and select the Export / File option.



Please enter the filename for the certificate.

Filename: Synclude eAIP Publisher.crt

DER is a binary format of the Certificate
PEM is a base64 encoded Certificate
PKCS#7 is an official Certificate exchange format
PKCS#12 is an encrypted official Key-Certificate exchange format

Export Format: PEM with Certificate chain

In the Filename box, enter the path where you want to store the certificate, followed by the filename. For example: `A:\Synclude eAIP Publisher.crt` will save it on a floppy disk.

You can now send this file to the person signing the eAIP packages.

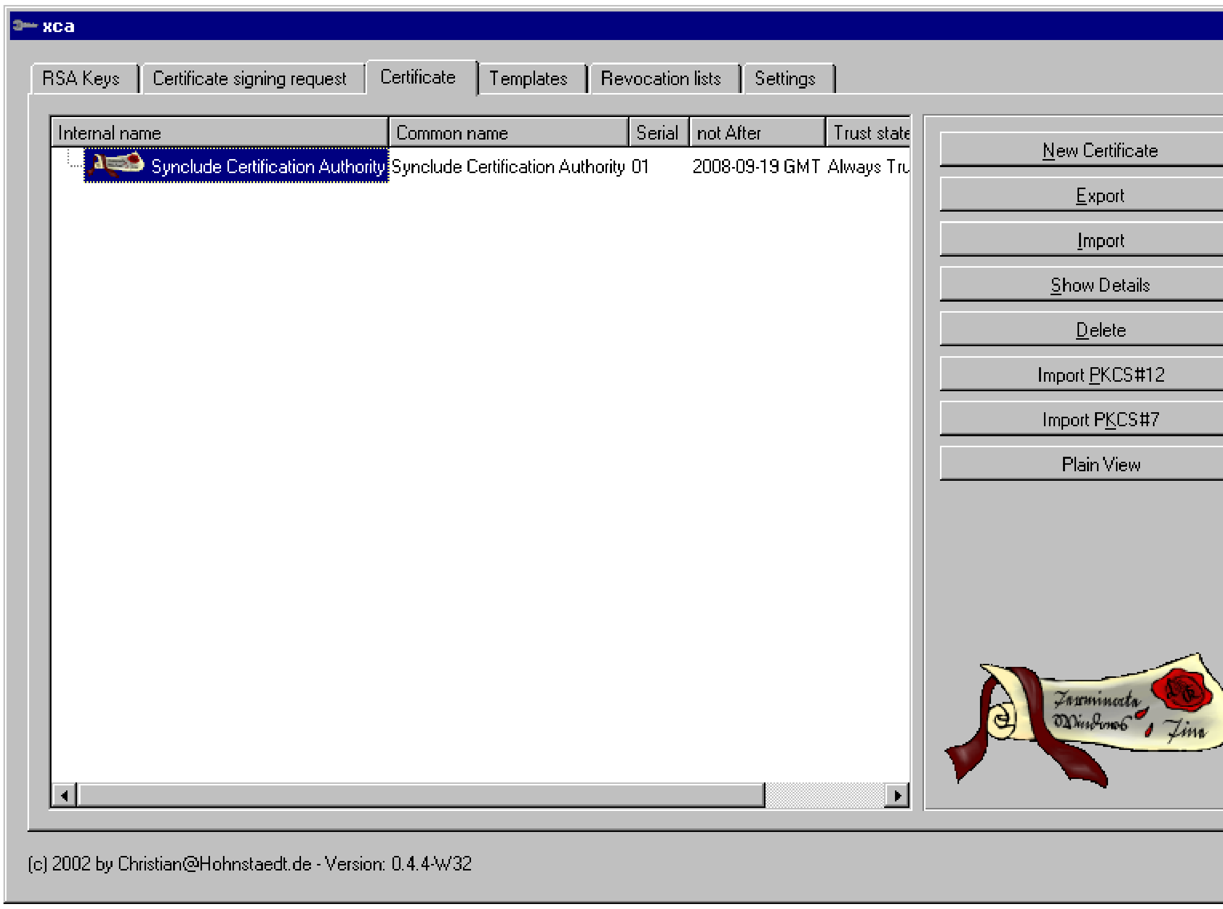
7.7. Import the Certificate

7.7.1. Introduction

This step is done by the person signing the eAIP.

7.7.2. Import certificate

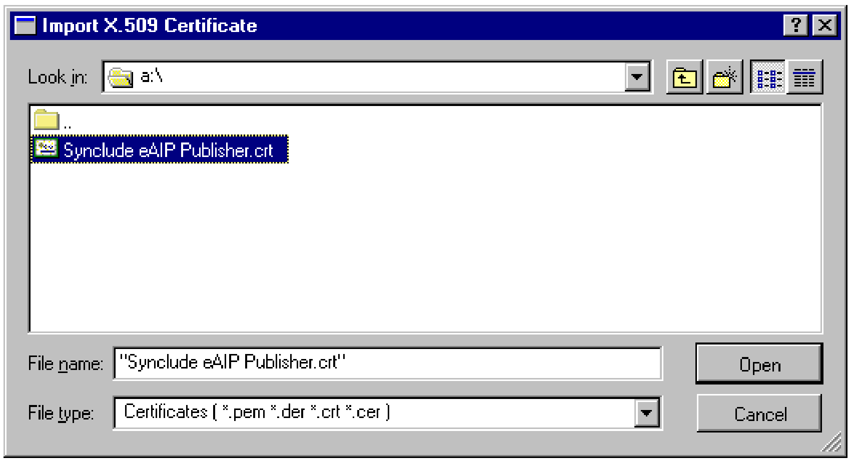
Open the XCA application.



Internal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Always Tru

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

Click on the Certificate tab on top. Click on the Import button on the right.



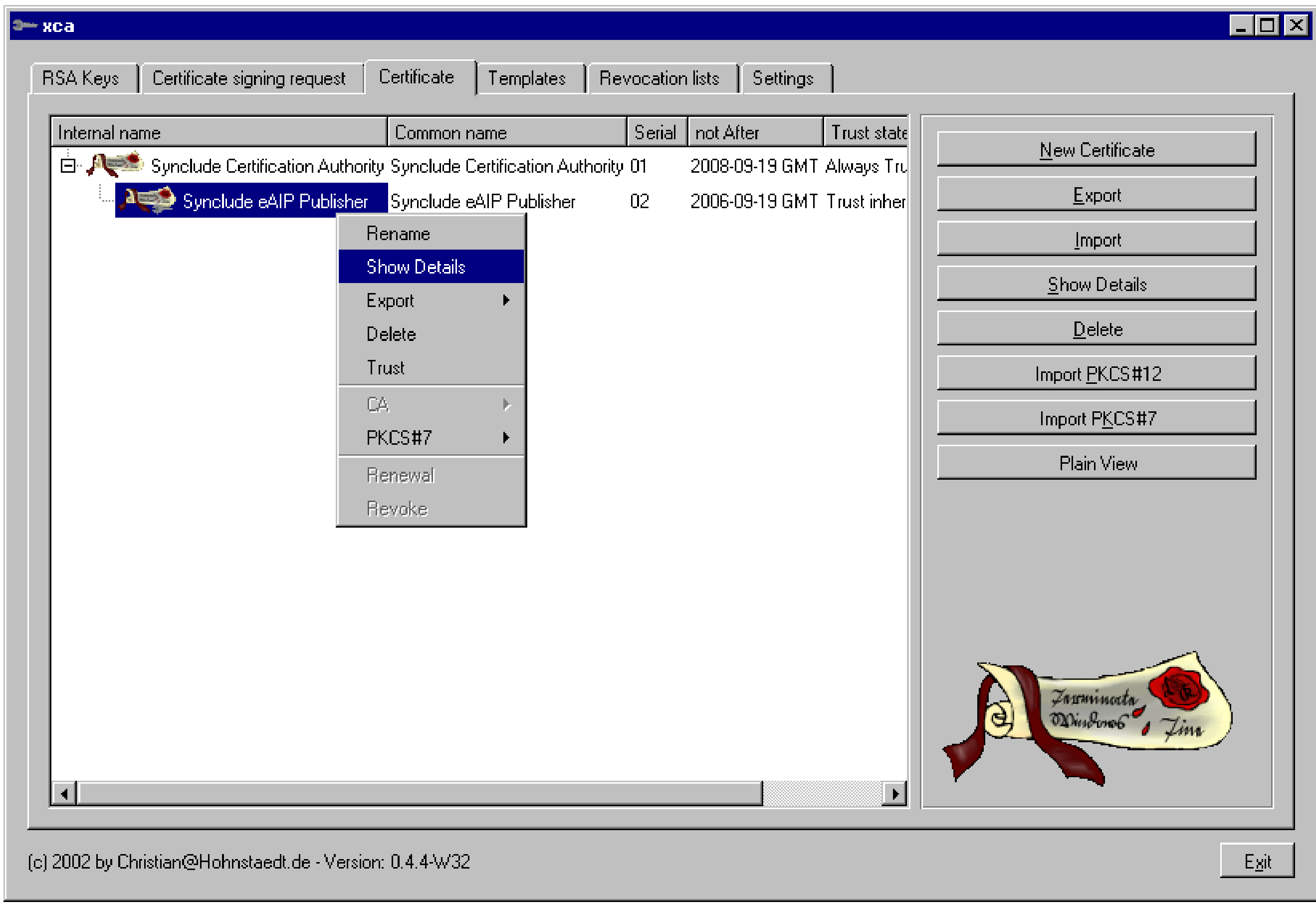
Look in: a:\

..
Synclude eAIP Publisher.crt

File name: "Synclude eAIP Publisher.crt"

File type: Certificates (*.pem *.der *.crt *.cer)

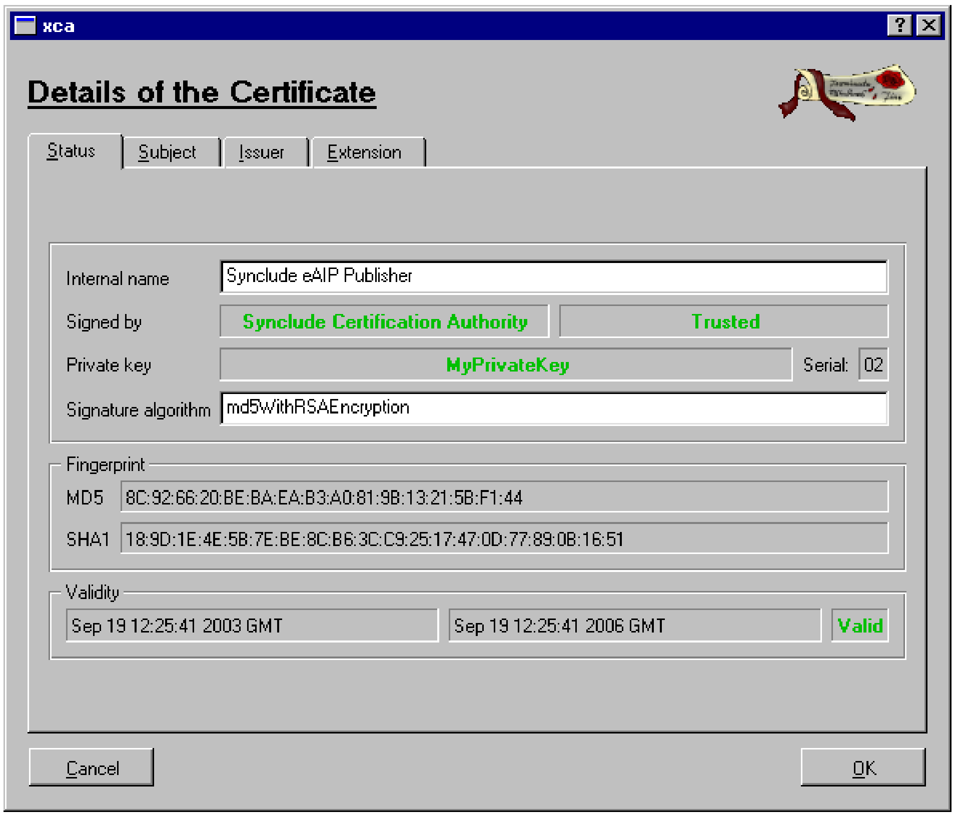
Navigate to the location of the certificate received from the CA. For example, if received on a floppy, navigate to `A:\Synclude eAIP Publisher.crt`.



Internal name	Common name	Serial	not After	Trust state
Synclude Certification Authority	Synclude Certification Authority	01	2008-09-19 GMT	Always Tru
Synclude eAIP Publisher	Synclude eAIP Publisher	02	2006-09-19 GMT	Trust inher

(c) 2002 by Christian@Hohnstaedt.de - Version: 0.4.4-W32

You will see the signing certificate under the CA certificate. To verify the import stage was done properly, right-click on the signing certificate and select Show Details.



Details of the Certificate

Internal name: Synclude eAIP Publisher

Signed by: Synclude Certification Authority | Trusted

Private key: MyPrivateKey | Serial: 02

Signature algorithm: md5WithRSAEncryption

Fingerprint

MD5: 8C:92:66:20:BE:BA:EA:B3:A0:81:9B:13:21:5B:F1:44

SHA1: 18:9D:1E:4E:5B:7E:BE:8C:B6:3C:C9:25:17:47:0D:77:89:0B:16:51

Validity

Sep 19 12:25:41 2003 GMT | Sep 19 12:25:41 2006 GMT | Valid

Verify that the Signed by is correct and that the status is trusted. Check that the Private Key field matches your private key.

Chapter 8. eAIP Security - How to setup up a PGP signing environment

8.1. Introduction

This chapter describes the steps to:

1. Create a signing certificate
2. Distribute the certificate to the end-users

It is recommended that the user gets familiar with PGP and public key cryptography in general by reading the links provided in the Technical and procedural choices chapter. See also the eAIP Security Risks and Mitigation Strategies in the eAIP Specification.

8.2. Necessary software

Please download and install the following software. You might need administrative privileges or ask your system administrator to perform the installation.

- WinPT [http://winpt.sourceforge.net/en/]: Windows Privacy Tray

8.3. Description of the procedure

The procedure to create a PGP key usable for signing the eAIP is:

1. Generate a PGP public/private key pair
2. Export the public key
3. Distribute the public key to the receiving parties

This is done by the person which will be responsible for the signing of the eAIP package.

8.4. Generate a PGP key

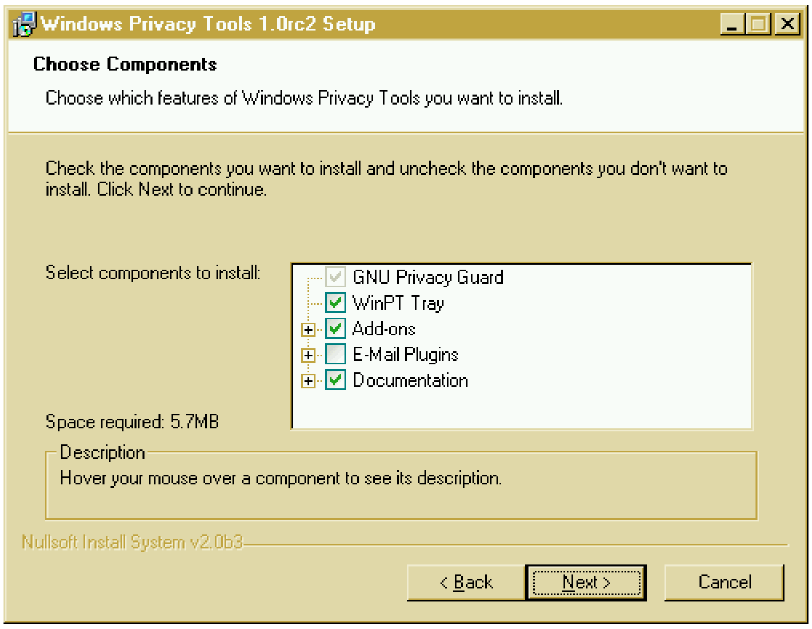
8.4.1. Introduction

The steps are:

1. Initial software installation
2. Generate a PGP key pair

8.4.2. Initial software installation

Run the WinPT installer which has been downloaded. Follow the instructions on screen.



Choose Components

Choose which features of Windows Privacy Tools you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- [✓] GNU Privacy Guard
- [✓] WinPT Tray
- [✓] Add-ons
- [ ] E-Mail Plugins
- [✓] Documentation

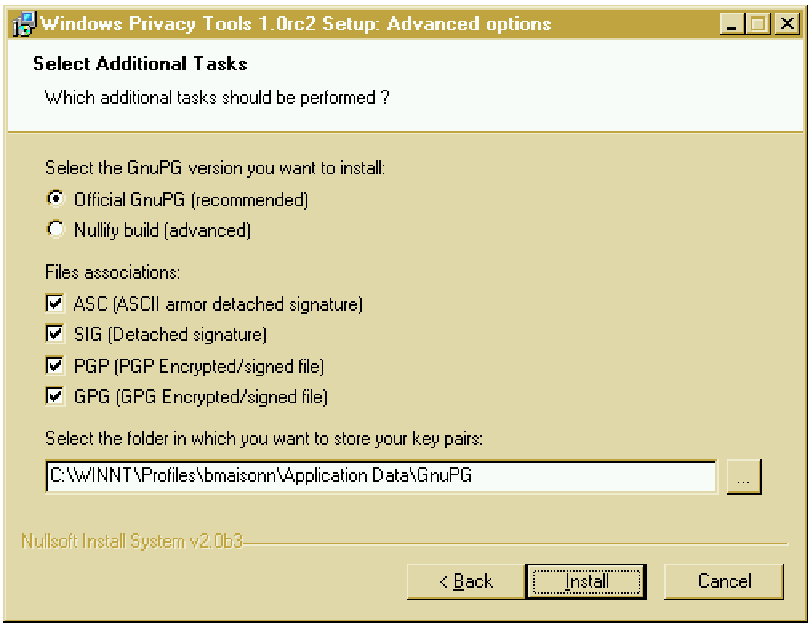
Space required: 5.7MB

Description

Hover your mouse over a component to see its description.

Nullsoft Install System v2.0b3

Install the required modules, as depicted above. This document does not cover all possible use of the software. Click Next to pursue installation.



Select Additional Tasks

Which additional tasks should be performed ?

Select the GnuPG version you want to install:

- ● Official GnuPG (recommended)
- ○ Nullify build (advanced)

Files associations:

- [✓] ASC (ASCII armor detached signature)
- [✓] SIG (Detached signature)
- [✓] PGP (PGP Encrypted/signed file)
- [✓] GPG (GPG Encrypted/signed file)

Select the folder in which you want to store your key pairs:

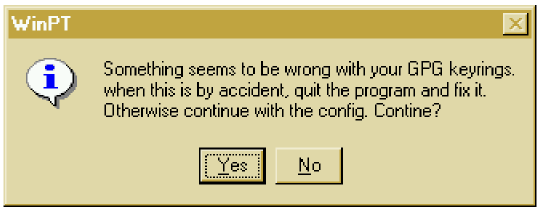
C:\WINNT\Profiles\bmaisonn\Application Data\GnuPG

Nullsoft Install System v2.0b3

In the advanced options, select the options as depicted above. Click Install to proceed with installation.

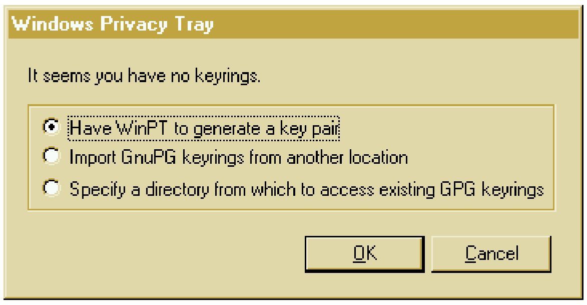
8.4.3. Generate a PGP key pair

Once installed, run the WinPT application. You will be prompted by the following:



Something seems to be wrong with your GPG keyrings. when this is by accident, quit the program and fix it. Otherwise continue with the config. Contine?

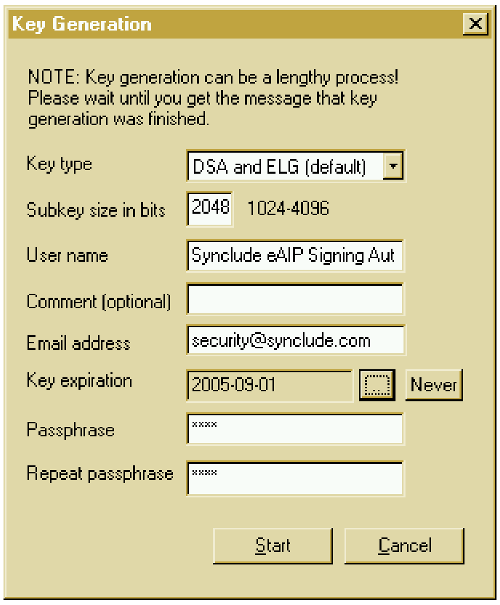
Click on Yes to have WinPT generate your personal key repository (keyring).


Windows Privacy Tray

It seems you have no keyrings.

- [✓ Have WinPT to generate a key pair]
- Import GnuPG keyrings from another location
- Specify a directory from which to access existing GPG keyrings

Select Have WinPT to generate a key pair, and click OK.


Key Generation

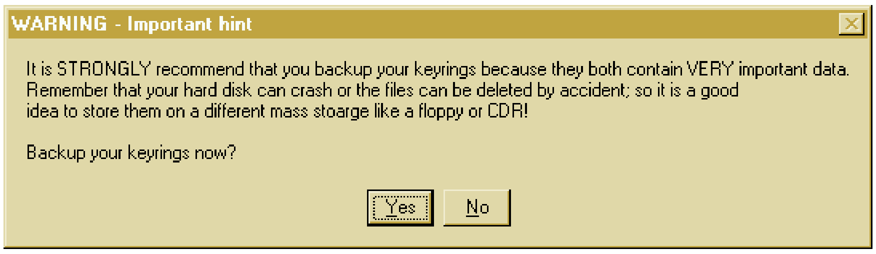
NOTE: Key generation can be a lengthy process!
Please wait until you get the message that key
generation was finished.

Key type	DSA and ELG (default)
Subkey size in bits	2048 1024-4096
User name	Synclude eAIP Signing Aut
Comment (optional)	
Email address	security@synclude.com
Key expiration	2005-09-01 Never
Passphrase	****
Repeat passphrase	****

Enter the following information:

- Key type: the public key algorithms used by PGP. Use combined DSA and ELG.
- Sub-key size in bits: the size of the key. We recommend at least 2048.
- User name: the name of the signing entity.
- Comment: information complementing the User name.
- E-mail address: an e-mail address where signing entity can be reached.
- Key expiration: the date until which this key is valid.
- Pass-phrase: select a strong password to protect this key. Repeat it to validate it.

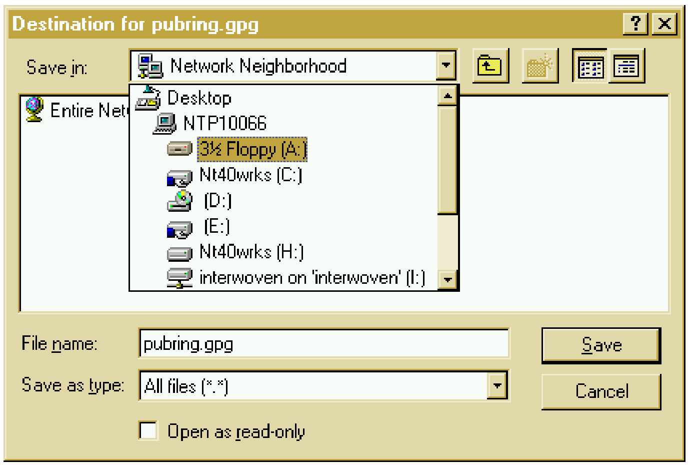
Click on Start to generate the key pair.


WARNING - Important hint

It is STRONGLY recommend that you backup your keyrings because they both contain VERY important data.
Remember that your hard disk can crash or the files can be deleted by accident; so it is a good
idea to store them on a different mass stoarge like a floppy or CDR!

Backup your keyrings now?

You may choose to follow the advice of the software, and proceed with a backup.



Select a safe location for your keyring.

8.5. Export the public key

8.5.1. Introduction

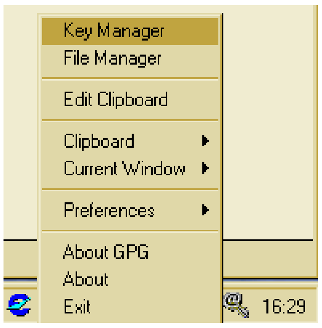
To verify packages you will sign, end users must have a copy of your public key. Therefor you must export it and distribute it to your audience.

The steps are:

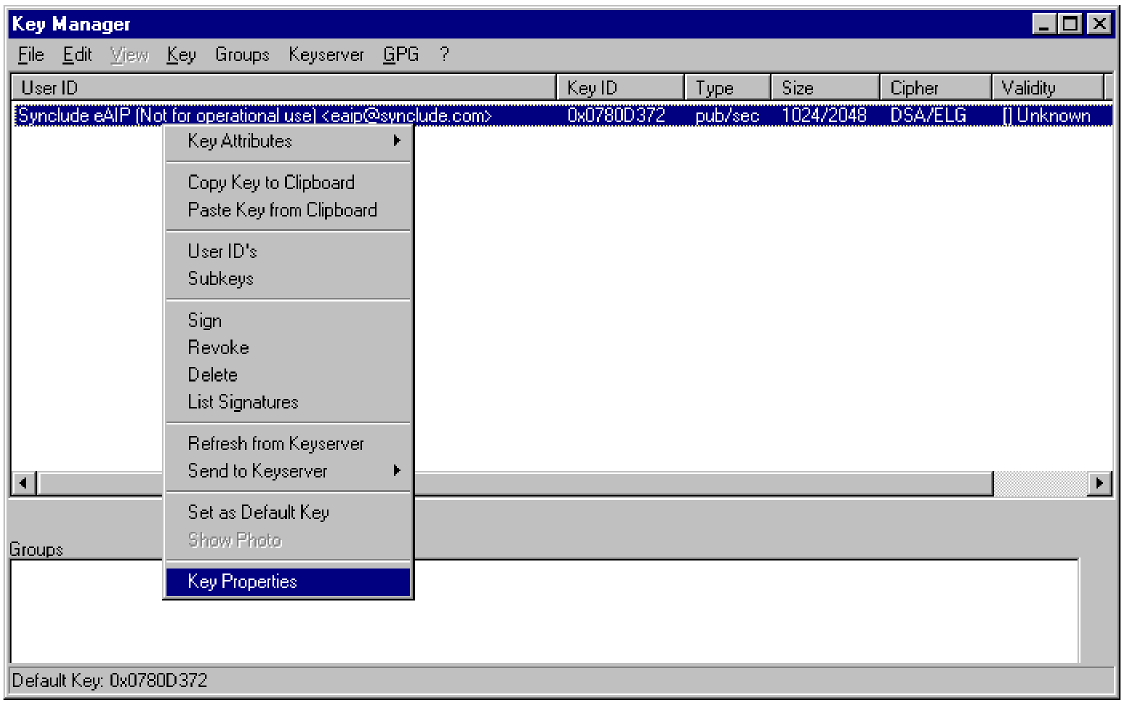
1. Note down the key fingerprint
2. Export public key from the PGP key pair
3. Distribute the public key and the fingerprint to end-users

8.5.2. Note down the key fingerprint

To export your private key, you must be running WinPT.



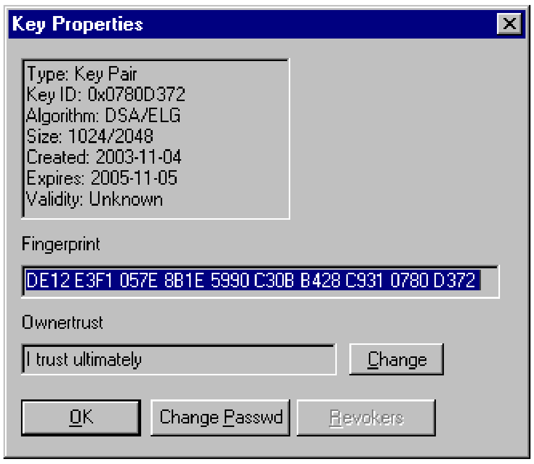
Right-click on the WinPT tray box icon (bottom right of your screen), and select Key Manager.



Synclude eAIP (Not for operational use) <eaip@synclude.com> 0x0780D372 pub/sec 1024/2048 DSA/ELG [] Unknown

Default Key: 0x0780D372

Right-click on the key and select Key Properties.



Type: Key Pair
Key ID: 0x0780D372
Algorithm: DSA/ELG
Size: 1024/2048
Created: 2003-11-04
Expires: 2005-11-05
Validity: Unknown

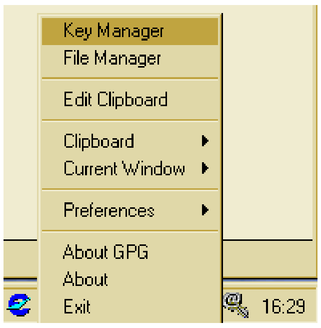
DE12 E3F1 057E 8B1E 5990 C30B B428 C931 0780 D372

I trust ultimately

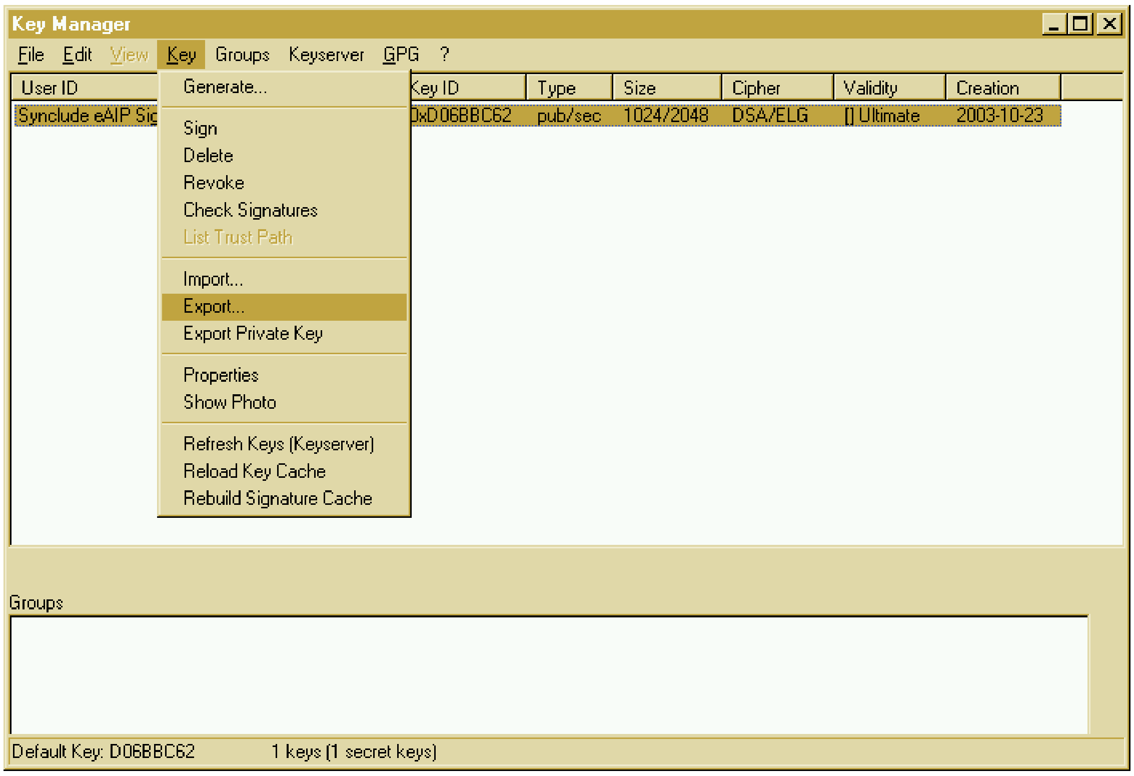
Note down the fingerprint.

8.5.3. Export public key from the PGP key pair

To export your private key, you must be running WinPT.



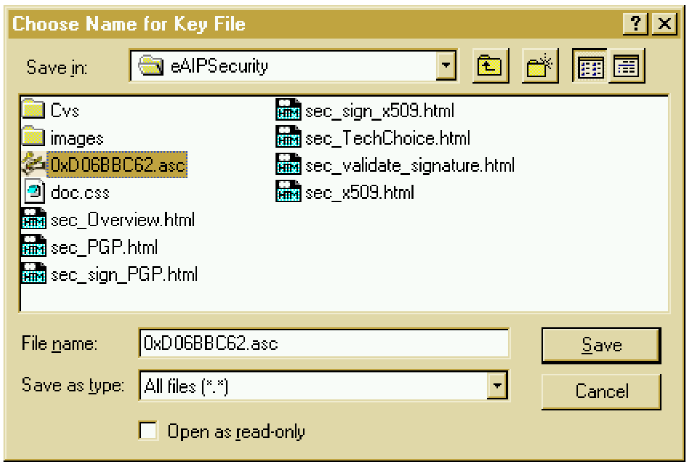
Right-click on the WinPT tray box icon (bottom right of your screen), and select Key Manager.



User ID	Key ID	Type	Size	Cipher	Validity	Creation
Synclude eAIP Sig	0xD06BBC62	pub/sec	1024/2048	DSA/ELG	[] Ultimate	2003-10-23

Default Key: D06BBC62 1 keys (1 secret keys)

In the Key Manager window, select in the Key menu the Export... option.



File name: 0xD06BBC62.asc

Save as type: All files (*.*)

Choose a location where to save the key. Leave the filename to the default value. This file will then be distributed to all the recipients of the signed eAIP packages.

8.5.4. Distribute the public key and the fingerprint to end-users

Send the file save above via any digital transmission channel to your end-users.

Send the key fingerprint via another transmission channel.