

OpenSBR XBRL signing manual

Version: 0.1 – May 5, 2020

Contents

1	Introduction	2
1.1	SBR Assurance.....	2
1.2	GLEIF Inline XBRL signature.....	3
2	Software overview	3
3	Manual	4
3.1	Using a certificate	4
3.2	SBR Assurance desktop tool.....	4
3.3	GLEIF signature desktop tool	5
3.4	GLEIF signature command-line tool.....	7
3.5	GLEIF Google Chrome extension.....	8
3.6	GLEIF Firefox extension (to be released)	8
4	Installation	9
4.1	SBR Assurance desktop tool.....	9
4.2	GLEIF signature desktop tool	9
4.3	GLEIF signature command-line tool.....	10
4.4	GLEIF Google Chrome extension.....	10
4.5	GLEIF Firefox extension (to be released)	11
5	Using the source code.....	12
6	Disclaimer.....	12

1 Introduction

XBRL is an important open standard for the exchange of (financial) reporting information. With an increasing number of regulators mandating XBRL and Inline XBRL, being able to prove the integrity is important.

OpenSBR.org, a non-profit initiative, created open-source solutions to sign and validate XBRL and Inline XBRL reports.

The solutions are developed for two different scenarios:

- The Dutch SBR Assurance: accountants sign off on XBRL documents, which is mandatory.
- GLEIF started a proof of concept, together with XBRL International and OpenSBR to sign off on Inline XBRL documents.

1.1 SBR Assurance

The specification is based on the Dutch XBRL signature framework (SBR Assurance). The only difference is the object of signing and the location of the signature:

- SBR Assurance¹ was designed to create a detached signature for XBRL files (and optional related files): the signature is created in a separate file.
- The GLEIF Inline XBRL signature is designed to create an enveloped signature for an Inline XBRL file (and optional related files): the signature is included in the object.

Standard Business Reporting (SBR) is a Dutch public-private partnership which aims to reduce the administrative burden, improve transparency, and foster innovation in regulatory reporting. SBR is about standardization of data definitions, processes, and technology across reporting domains, creating a level playing field for software vendors and service providers, encouraging competition.

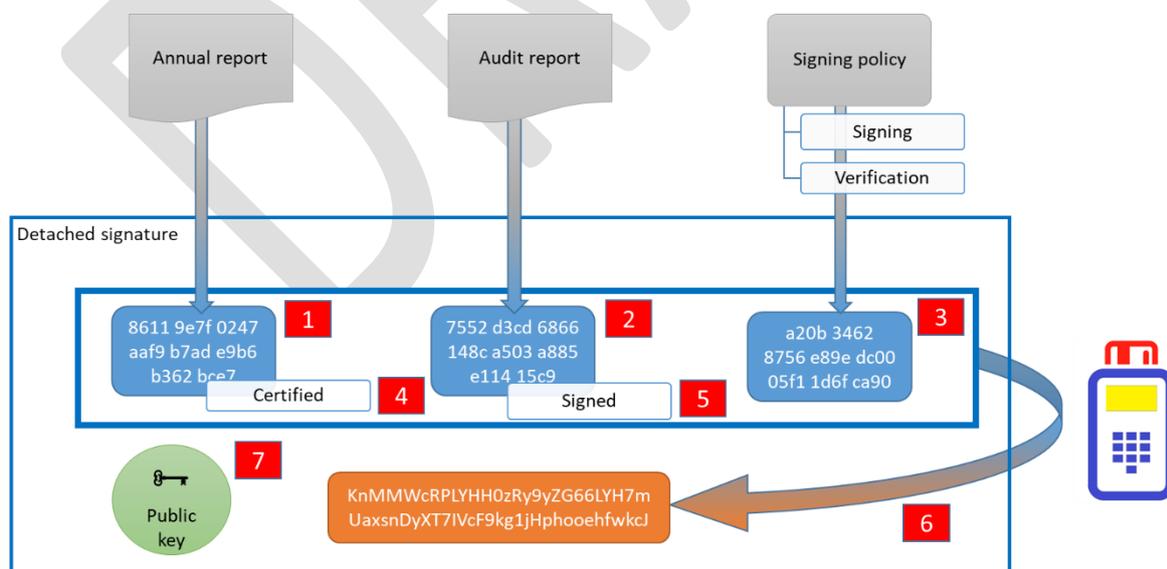


Figure 1: Linking and signing process (source: NBA.nl, used with permission)

¹ The SBR Assurance specification was created by The Royal Netherlands Institute of Chartered Accountants, or Koninklijke Nederlandse Beroepsorganisatie van Accountants (NBA). More information on <https://www.nba.nl/over-de-nba/english-information/>

The Dutch business register (Kamer van Koophandel) takes part in SBR and requires businesses to file their annual report in a digital format (eXtensible Business Reporting Language). Per January 2018, medium-sized businesses must report with SBR, including the audit report.

SBR Assurance is the specification for the digital equivalent of the auditor’s opinion on an annual report. With SBR Assurance, an auditor can sign off on an annual report with a qualified digital signature. The resulting file, a *detached signature*, can be used by recipients of the report to validate that the signature, annual report and audit report were not changed since signed by the auditor, and to verify which auditor signed off on the files.

OpenSBR provided the source code, and a proof of concept computer program to generate and verify the detached signature. Both the source code and the program can be used under a permissive MIT open source license.

1.2 GLEIF Inline XBRL signature

Together with GLEIF and XBRL International, OpenSBR drafted a specification for a digital signature on Inline XBRL, based on existing open standards and specifications.

Documents can be signed with X.509 certificates, including GLEIF certificates (tying a signed document to the Legal Entity Identifier of an organization). The specification relies on open standards such as X.509, XML-DSig and XAdES.

OpenSBR also created draft reference implementations for signing and verification of electronic signatures.

2 Software overview

OpenSBR created different software packages for different usage scenarios:

- Signing tools for XBRL and Inline XBRL, for different platforms
- Validation tools for web browsers

The specifications of the software tools are listed in the table below.

Software	Goal	Platform	Technology
SBR Assurance desktop tool	Sign and validate XBRL documents	Windows	.NET 4.6.2 and later
GLEIF signature desktop tool	Sign, countersign and validate Inline XBRL documents	Windows	.NET 4.7.2 and later
GLEIF signature command-line tool	Sign, countersign and validate Inline XBRL documents	Windows, macOS, Linux	Dotnet Core 3.1
GLEIF Google Chrome extension	Validate Inline XBRL documents	Google Chrome, Microsoft Edge (Windows, macOS, Linux)	JavaScript
GLEIF Firefox extension (to be released)	Validate Inline XBRL documents	Firefox (Windows, macOS, Linux)	JavaScript

The installation instructions for the tools can be found in chapter 0.

3 Manual

3.1 Using a certificate

To be able to sign documents, a digital computer certificate is required. It is possible to use test certificates. In order to send in qualifying audit reports to the Dutch business register, an auditor must use his/her professional government-trusted certificate (PKIOverheid certificate²). A certificate can be obtained from one of the trusted service providers.

3.2 SBR Assurance desktop tool

The tool works in two different modes. A detached signature can be created for one or more documents, or a detached signature can be verified.

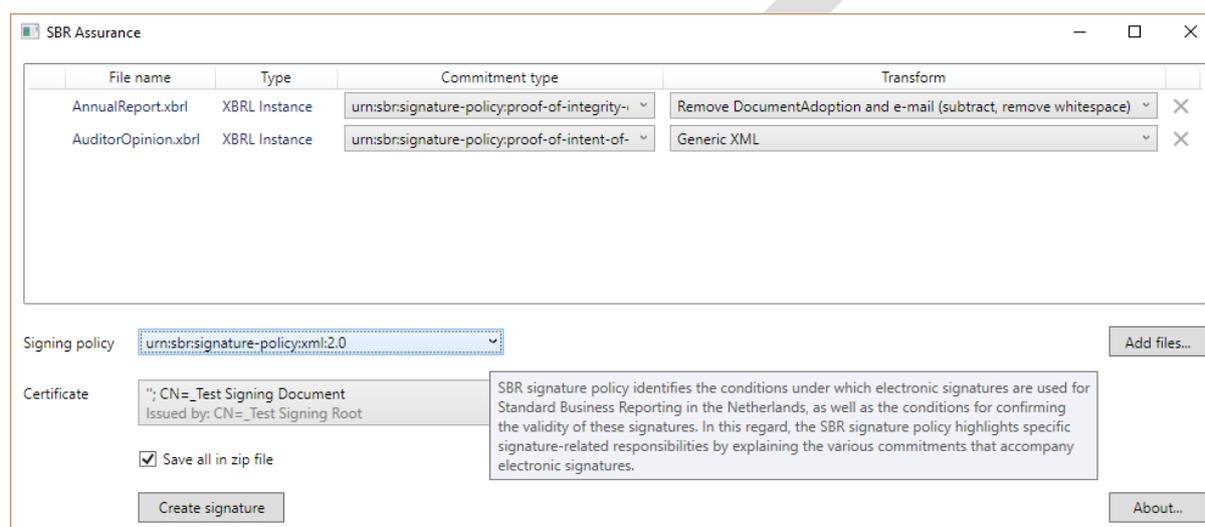
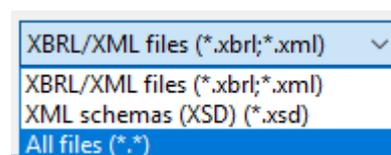


Figure 2: proof of concept desktop tool (OpenSBR Assurance) – creating a signature

Verification of documents is possible without a certificate.

3.2.1 Creating a detached signature (linking and signing)

Any type of file can be added to the tool by clicking the "Add files..." button, or by dragging and dropping files. Typically, an auditor chooses an XBRL annual report and an XBRL audit report, but any document type can be included, such as XBRL extension files, XML files, PDF files, spreadsheets (e.g. ods, xlsx), etc. Other files can be added when selecting "All files (*.*)" in the file dialog.



For each document, the following options are available:

- Commitment type. The NBA defined three types of commitment an auditor can express on the document:
 - Proof of integrity of the object for which the practitioner expresses an opinion. Used on an object of assurance.
 - Proof of intent of practitioner to express an opinion. Used for the audit report or other types of opinion.
 - Proof of intent of practitioner to add a copy of the opinion.

² More information is available on <https://www.logius.nl/english/pkioverheid/>

- Transform. This option applies to XML (or XBRL) documents, which allows certain XML information to be excluded from the signature scope. A basic transform rule, which excludes the document adoption date³, is included in the settings; but more transform rules can be added by editing the Settings.xml file.

With a correct certificate selected, clicking the “Create signature” button will create a ZIP file containing copies of the original files, and the detached signature.

3.2.2 Verifying a digital signature

The integrity of a signed document set can be tested by opening the detached signature and original files in the tool; either by adding files one by one, or by adding a ZIP file containing all files.

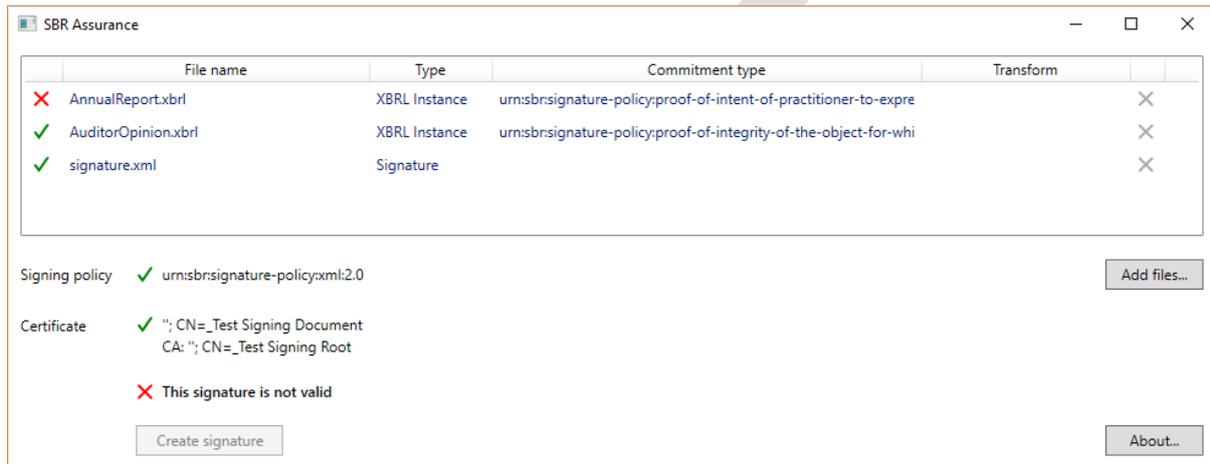


Figure 3: proof of concept desktop tool (OpenSBR Assurance) – verification of a signature

The tool shows which signing policy and which certificate were used. If any of the files has been altered, the tool will detect that the detached signature file does not match and shows a red cross.

3.3 GLEIF signature desktop tool

Start the tool by clicking on SignXBRL.exe. An Inline XBRL document can be “dragged and dropped” onto the main window.

If the Inline XBRL document was already signed, the signature is validated (could take a couple of minutes for large documents) and the signatures and results are shown.

³ In the filing process, the business owner legally adopts the annual report only after the auditor has signed the document. The adoption date will be inserted after the signing process, so must be excluded from the cryptographic signing process.

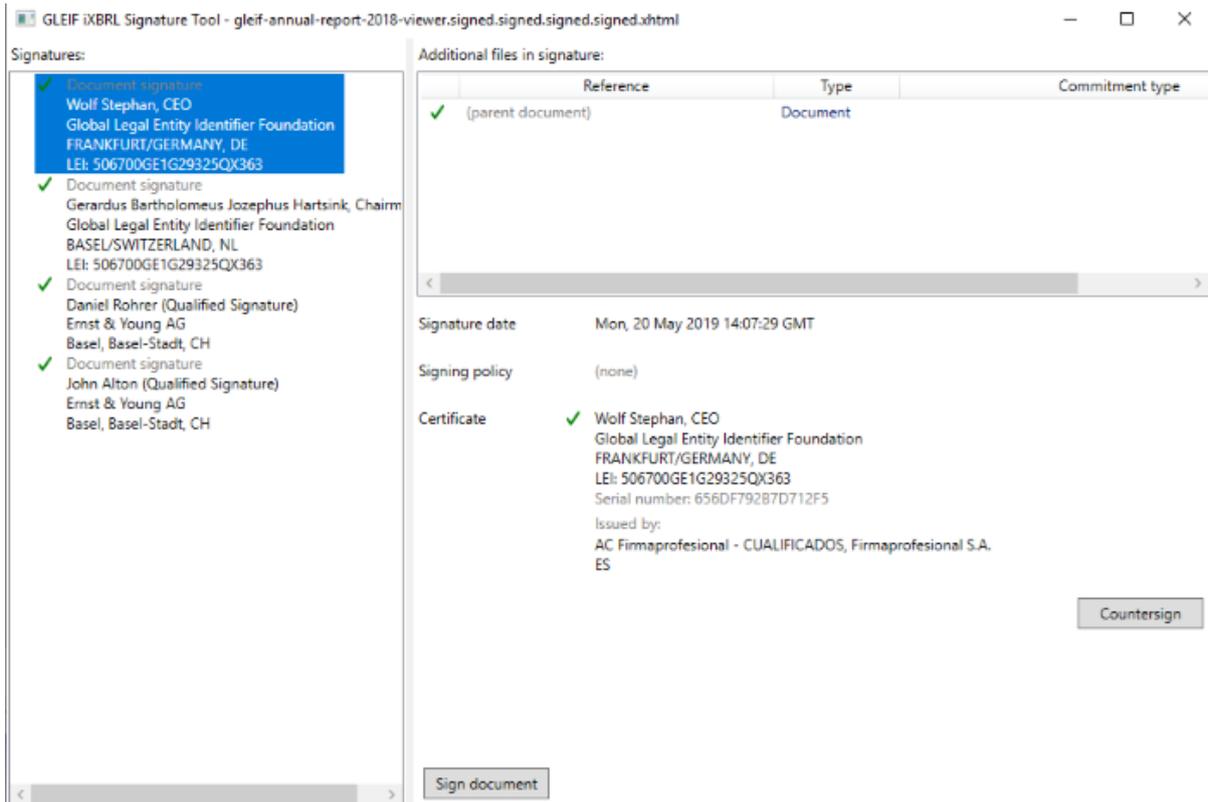


Figure 4: proof of concept desktop tool (GLEIF Inline XBRL signature) – verification of a signature

The document can be signed, or, after selecting an existing signature, counter-signed. In the signing interface, it is possible to include reference to external files, such as extension taxonomy packages or even pdf files, which will be included in the signature calculation.

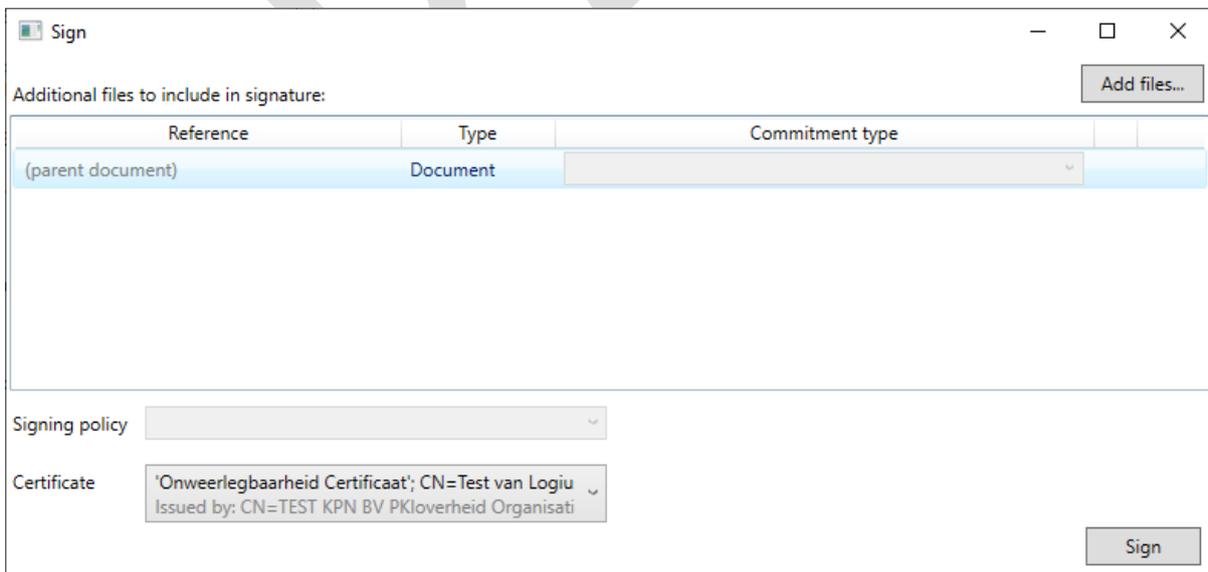


Figure 5: proof of concept desktop tool (GLEIF Inline XBRL signature) – signing a document

3.4 GLEIF signature command-line tool

The command-line tool is designed to work on multiple platforms, which means that not all platform-specific features (such as certificate stores) are fully incorporated.

Make sure the dotnet framework is installed, and the command-line tool is downloaded and unpacked.

Start a terminal window (Applications→Utilities→Terminal) and navigate to the folder containing the command-line tool.

[to do: explain how to navigate to the appropriate folder for the tool]

Type in “dotnet SignXBRL-cli.dll” and press Enter. The following should be visible on the screen:

```
SignXBRL-cli <command> [<options>] file
Commands:
  sign
  countersign
  list
  validate
  certificatestore      List available keys in certificate store
```

Signing requires multiple parameters. The options are displayed when typing: “dotnet SignXBRL-cli.dll sign”

```
SignXBRL-cli sign
[-pfx <p12 file> [-pw <password>]]      Read certificate from PKCS#12 file
[-cert <certificate file> -key <private key>]  Read certificate from PEM-encoded crt and
key files
[-subject <certificate store subject text>]  Read certificate from certificate store
[-thumbprint <certificate store thumbprint>]  Read certificate from certificate store
[-out <output file>] <file>
```

Currently, the option to select a certificate from the certificate store is not supported on every platform. [to do: explain how to export a certificate from the store and use it as a file]

An example to sign a document is:

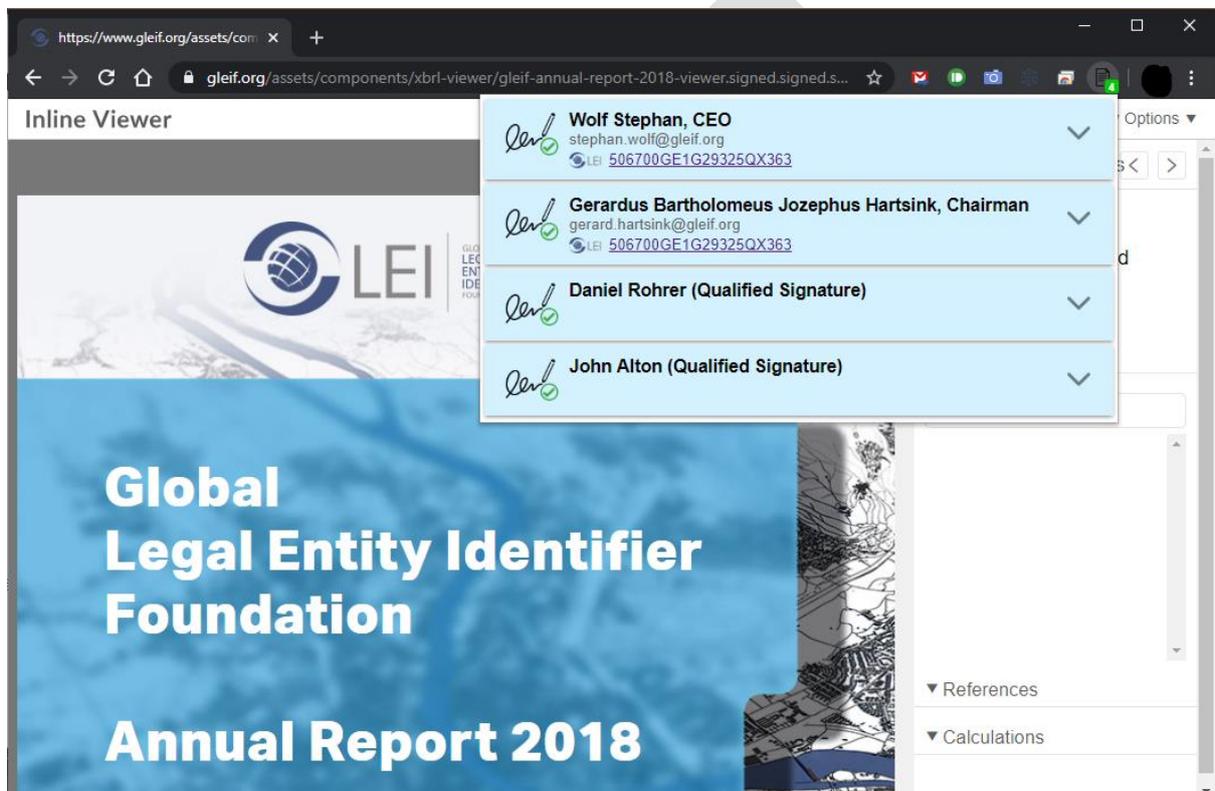
```
“dotnet SignXBRL-cli.dll sign -pfx certificate.p12 -pw abcd document.xhtml”
```

[to do: explain how to sign a document in a different folder]

3.5 GLEIF Google Chrome extension

The extension is visible as a button in the browser tool bar. The icon shows one of the following statuses:

Icon	Status
	No document with a signature loaded
	A signed document is loaded; verification in progress
	A signed document is loaded, and signed by (x) certificates
	Error encountered when validating the signature



3.6 GLEIF Firefox extension (to be released)

Refer to the instructions for Google Chrome.

4 Installation

4.1 SBR Assurance desktop tool

4.1.1 Requirements

The source code is written in the computer language C# and uses the Microsoft .NET framework. The applied cryptography requires version 4.6.2 (or later) of the .NET framework. This version is supported on computers running Windows 7 SP1, Windows 8.1, Windows 10, Windows Server 2008 R2 SP1, Windows Server 2012, Windows Server 2016. The software is not designed to run on Linux or macOS.

The proof of concept desktop tool requires the Microsoft Windows graphical user interface.

4.1.2 Installing OpenSBR Assurance

A ZIP file containing the desktop tool can be downloaded from <http://opensbr.org/>. After unpacking, the following files are available:

- SBRAssurance.exe, the executable to start OpenSBR Assurance
- OpenSBR.Xades.dll, a computer library containing the cryptographic functions
- Settings.xml, containing advanced settings for the signing policy

4.2 GLEIF signature desktop tool

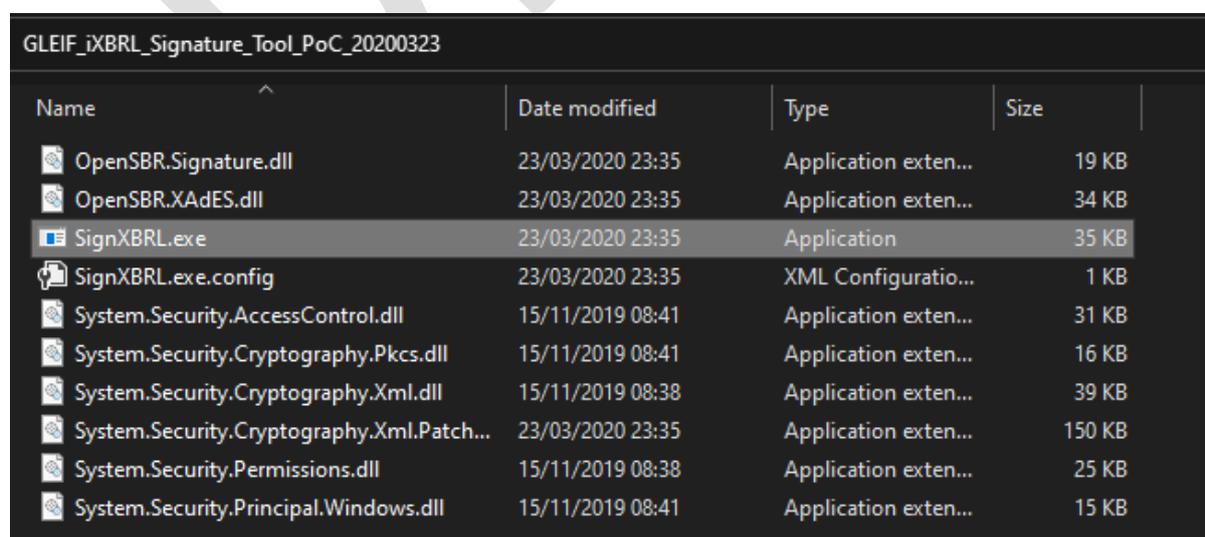
The source code is written in the computer language C# and uses the Microsoft .NET framework. The applied cryptography requires version 4.7.2 (or later) of the .NET framework. This version is supported on computers running Windows 7 SP1, Windows 8.1, Windows 10, Windows Server 2008 R2 SP1, Windows Server 2012, Windows Server 2016. The software is not designed to run on Linux or macOS.

The proof of concept desktop tool requires the Microsoft Windows graphical user interface.

4.2.1 Installing GLEIF signature tool

A ZIP file containing the desktop tool can be downloaded from <https://opensbr.org/inline/>. No installation is required, the ZIP file should be unpacked only.

After unpacking, 10 files should be visible. SignXBRL.exe is the executable file to start the application.



Name	Date modified	Type	Size
OpenSBR.Signature.dll	23/03/2020 23:35	Application exten...	19 KB
OpenSBR.XAdES.dll	23/03/2020 23:35	Application exten...	34 KB
SignXBRL.exe	23/03/2020 23:35	Application	35 KB
SignXBRL.exe.config	23/03/2020 23:35	XML Configuratio...	1 KB
System.Security.AccessControl.dll	15/11/2019 08:41	Application exten...	31 KB
System.Security.Cryptography.Pkcs.dll	15/11/2019 08:41	Application exten...	16 KB
System.Security.Cryptography.Xml.dll	15/11/2019 08:38	Application exten...	39 KB
System.Security.Cryptography.Xml.Patch...	23/03/2020 23:35	Application exten...	150 KB
System.Security.Permissions.dll	15/11/2019 08:38	Application exten...	25 KB
System.Security.Principal.Windows.dll	15/11/2019 08:41	Application exten...	15 KB

4.3 GLEIF signature command-line tool

The software was built using Microsoft dotnet cross-platform technology. This does require the installation of the dotnet framework.

4.3.1 Installation of the Microsoft dotnet framework

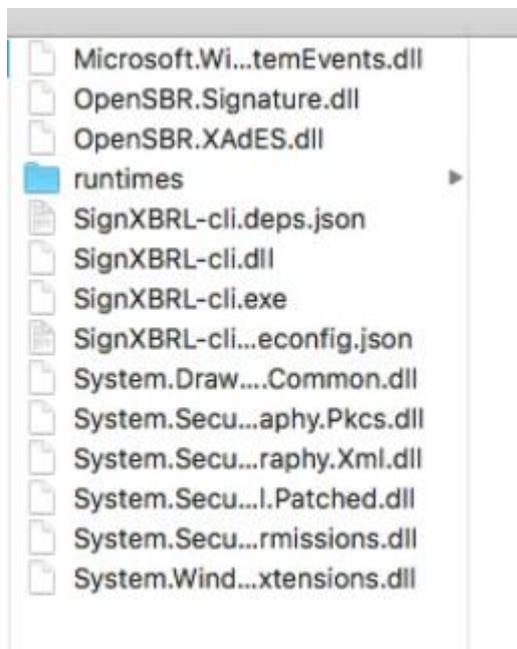
The dotnet framework 3.1 must be installed from the following link:

<https://dotnet.microsoft.com/download/dotnet-core/3.1>

Download the latest .NET Core Runtime 3.1.x for macOS (Installer x64), and follow the instructions to install.

4.3.2 Installation of the command-line tool

Download the latest version of the command-line tool from <https://opensbr.org/inline/> and unpack the zip file in a location of your choice.



4.4 GLEIF Google Chrome extension

The GLEIF Google Chrome extension can be downloaded from the Chrome Web Store:

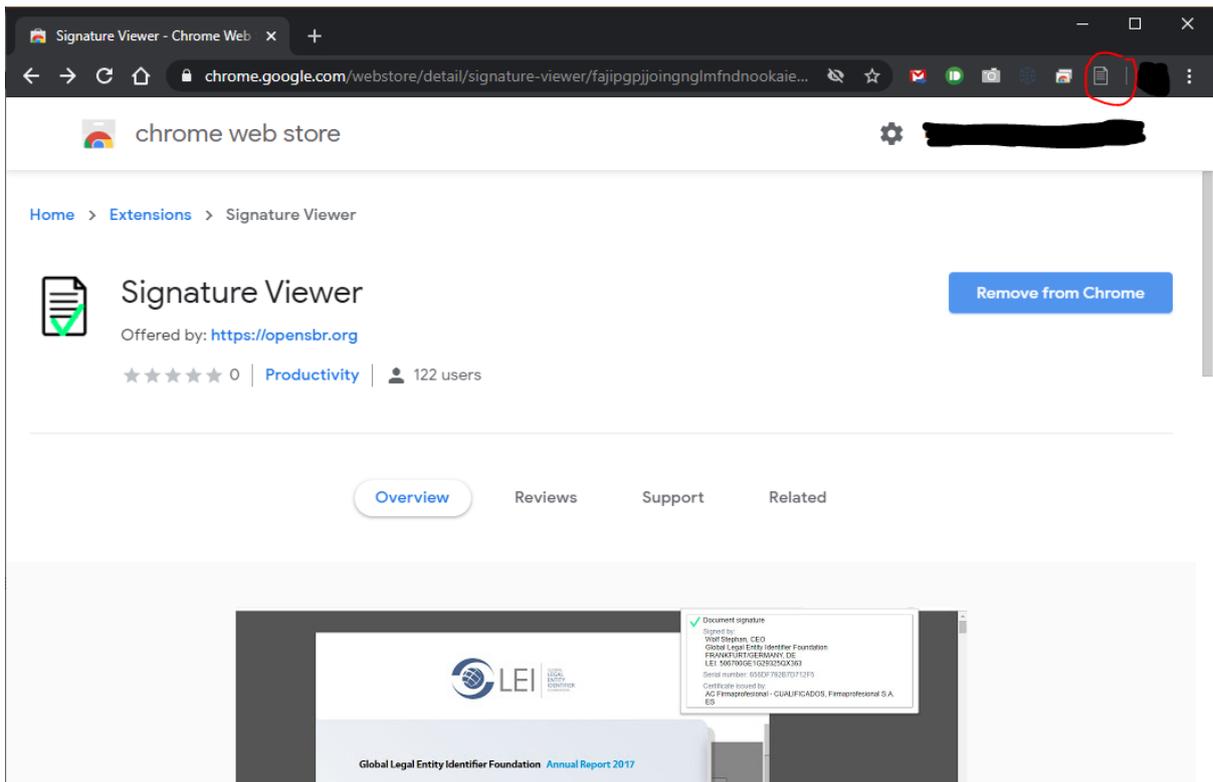
<https://chrome.google.com/webstore/category/extensions>

The specific location for the Signature Viewer can be found on:

<https://chrome.google.com/webstore/detail/signature-viewer/fajipgpjjoingnglmfndnookaiebecop>

Clicking the button “Add to Chrome” will install the extension in the browser.





4.5 GLEIF Firefox extension (to be released)

The add-on will be published on <https://addons.mozilla.org/>

5 Using the source code

The source code is available for download at GitHub. A link can be found on <http://opensbr.org/>. The source code contains both the OpenSBR Assurance library and the desktop tool. The desktop tool and library can be compiled from scratch. The library can be used in other project as well.

Programmers of any level will find it easy to open the source code with tools such as Visual Studio⁴.

6 Disclaimer

OpenSBR provides the source code and tool under MIT License conditions.

MIT License

Copyright © 2017 OpenSBR.org – <http://opensbr.org/>

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

⁴ The source code was create with the free edition of Microsoft Visual Studio 2017. The source code contains some expressions which are specific to this version (C# 7.0), but which can be replaced easily.