**개발자를 위한**

**SCORE PKI For OpenWeb**

**간편설치 매뉴얼**

**Ver 1.0**

한국무역정보통신

공인인증센터

2015.10.15



**목 차**

[1. 개 요 1](#_Toc498705492)

[2. 배 경 1](#_Toc498705493)

[3. 지원 플랫폼 1](#_Toc498705494)

[4. 프로그램 개요 2](#_Toc498705495)

[5. 제품의 구성 3](#_Toc498705496)

[6. 데모 설치 4](#_Toc498705497)

[7. 설정정보 7](#_Toc498705498)

[8. 설치 화면 예시 8](#_Toc498705499)

[9. 설치결과 확인 화면 10](#_Toc498705500)

[10. 제품삭제 방법 11](#_Toc498705501)

[11. 버젼 업그레이드 방법 12](#_Toc498705502)

[12. 동작전 환경 구성 (인증서 유효성 검증시 확인 사항) 12](#_Toc498705503)

# 개 요

㈜ 한국무역정보통신의 클라이언트 암호화 OpenWeb용 툴킷을 설치하고 적용하기 위한 매뉴얼 입니다.

# 배 경

IT 환경이 발달함에 따라 OS별 다양한 웹브라우저가 개발되고, 인터넷을 보다 편리하게 사용할 수 있도록 제공하고 있습니다.

따라서 과거에는 windows 환경의 IE 형태로만 제공되는 보안툴킷이, 오픈웹 환경을 지원하기 위해 IE, 파이어폭스(FireFox), 크롬(Chrome), 사라리(Safari), 오페라(Opera) 등의 브라우저에서도 연동 가능한 형태로 제작 되었습니다.

공인인증서를 이용하는 고객의 환경을 최대한 고려하여 로그인, 전자서명, 암/복호화 등의 보안기능을 수행할 수 있도록 하였습니다.

# 지원 플랫폼

SCORE PKI For OpenWeb 제품은 다양한 브라우저에 연동하게 하기 위해서 윈도우 프로그램으로 개발되어 있습니다.

* 지원 가능한 OS
* Microsoft Windows XP sp3 이상
* Microsoft Windows Vista
* Microsoft Windows Win 2003
* Microsoft Windows Windows 7
* Microsoft Windows Windows 8
* Microsoft Windows Win 10
* 지원 브라우저 종류
* IE 8.0 이상 ( XP + SP3 + IE8 이상)
* Firefox 41.0 이상
* Safari 5.1.7 이상
* Chrome 45.0 이상
* Opera 32.0 이상
* Edge
* 네트워크 인터페이스: Ethernet
* 네트워크 프로토콜: HTTPS over TCP/IP

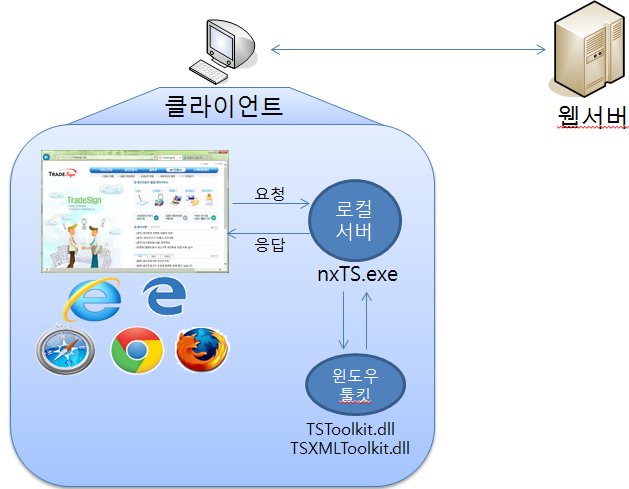
# 프로그램 개요

해당 제품은 고객의 서비스가 대부분 웹 형태로 제공되고 있는 환경에서 브라우져 즉 클라이언트에서 암/복호화 , 전자서명, 인증서 유효성 검증등을 처리 할 수 있는 형태의 제품 입니다.

* 동작 프로세스

브라우저를 통해서 데모화면을 접속하면 PC에 해당 제품의 설치여부를 체크 합니다. 만일 해당 프로그램이 PC에 설치 되어 있지 않다면 설치페이지로 이동 합니다. 설치페이지에서 “설치프로그램 다운로드” 를 클릭하여 설치프로그램을 PC 로 내려받고 고객이 직접 설치를 진행합니다.

이후에 다시 브라우저로 데모화면을 접속하여 해당기능을 사용할 수 있습니다.

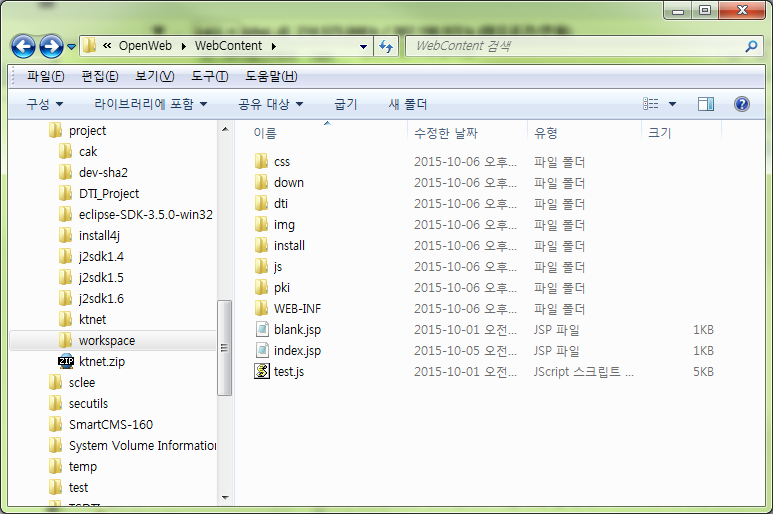


제품 구성도

* OpenWeb 제품은 클라이언트 단에서 제공되는 툴킷으로 서버 버전은 서비스에 사용되는 언어에따라 별도 구매하여 사용하셔야 합니다.
* PC에 설치되는 “로컬서버"는 브라우저의 html, javascript 소스와 통신을 하면서 Activex에서 제공하던 암/복호화, 전자서명 등의 기능을 동작하는 서버 입니다.
* 브라우저에서 필요한 값을 파라메터로 로컬서버에 전달하여 해당기능을 동작하고 응답 결과를 callback 함수로 리턴 받아서 화면에 뿌려주는 형태로 동작이 됩니다.
* Activex 제품에서는 API 끼리 메모리로 값을 공유 할 수 있지만 해당 제품은 API끼리 값을 공유할 수 없는 구조로 되어 있습니다. 따라서 API를 호출할 때 마다 필요한 정보를 모두 전달해야합니다. 리턴값에서도 얻을 수 있는 가능 정보를 모두 포함하여 수신받도록 되어 있어서 이후에 필요한 정보만 뽑아서 사용하는 형태로 범용적으로 API를 설계하였습니다.
* OpenWeb 제품은 비동기식 방식형태로 코딩이 되어야 합니다. 인증서 선택창 및 암/복호화 기능을 하는동안 스크립트는 기다리지 않고 하던 동작을 진행하게 되어 있습니다. 이후에 로컬서버에서 동작을 마치면 callback 함수로 결과 값이 넘어와 이때 결과에 따른 프로세스를 진행하여야 합니다.
* 로컬서버는 요청메시지를 받아서 윈도우 툴킷으로 해당 기능을 요청합니다. 로컬서버는 윈도우 툴킷을 사용하기 위한 GateWay 형태의 제품으로 모든 기능은 윈도우 툴킷에서 담당합니다.

# 제품의 구성

제품을 받아서 압축을 풀어보면 다음과 같은 형태의 파일을 볼 수 있습니다.



* 디렉토리별 설명

|  |  |  |
| --- | --- | --- |
| 디렉토리 | 설명 | |
| down/\*\* | nxtssetup.exe : nxTS서버 설치 파일(PKI)  TSToolkit.exe : 기존의 Activex 설치 파일 | |
| install/\*\* | 홈페이지의 소스에 include 해서 모듈을 설치 할 수 | |
| js/jquery/\*\* | jquery 1.11.3 (데모 테스트용이며 필요시 업무서버에 사용중인 Jquery를 사용) | |
| js/nxts/\*\* | nxts.min.js | nxts공통 js |
| nxtxpki\_config.js | PKI 설정 파일 |
| nxtspki.js | PKI js |
| blank.jsp | iframe 전송시 기본 src로 지정할 빈 페이지 | |
| pki/\*\* | PKI 클라이언트 데모 샘플 | |
| Img/\*\* | processing.gif 요청중 이미지 | |

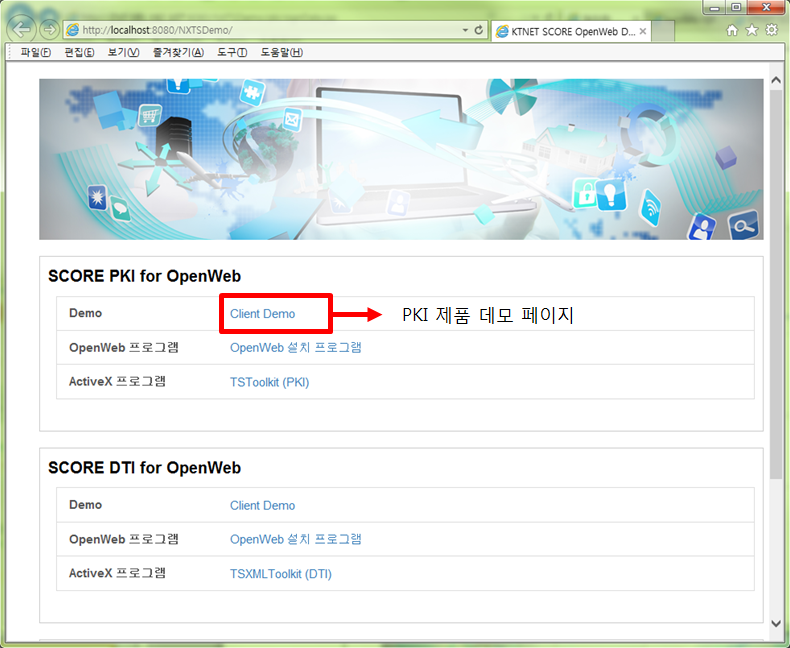
# 데모 설치

PC에 WAS의 DocumentRoot 디렉토리에 NXTSDemo 디렉토리를 생성하고 해당 디렉토리에 압축을 풀어 파일을 복사합니다.

WAS를 실행한 후 PC에서 웹브라우저로 URL을 입력하여 데모프로그램을 실행합니다.

예) <http://localhost:8080/NXTSDemo>/

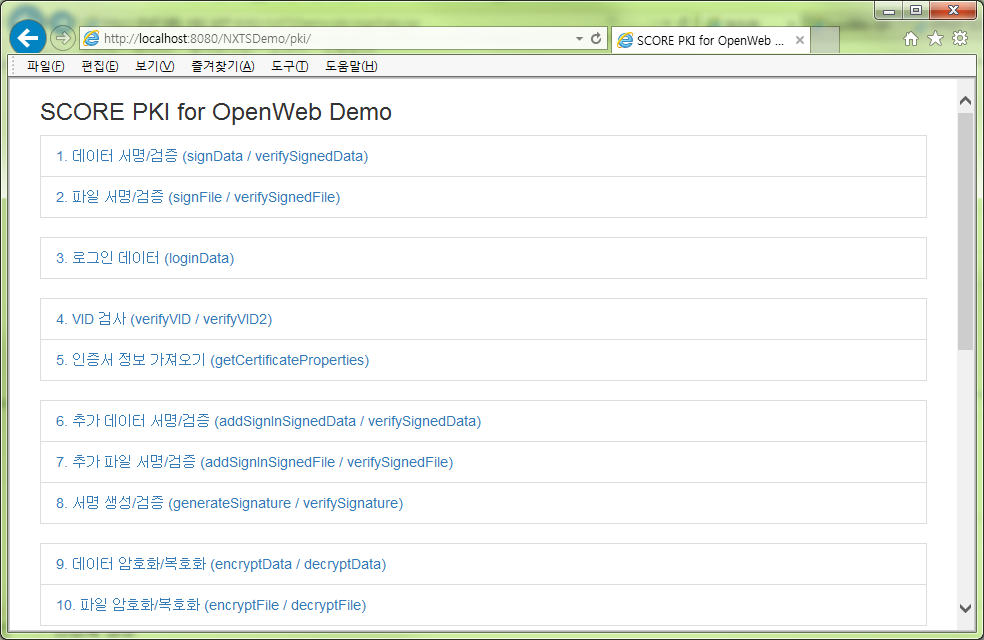
※샘플 소스에 NXTSDemo 라는 이름이 고정되어 있어서 디렉토리명을 변경하지 말아 주세요.!!

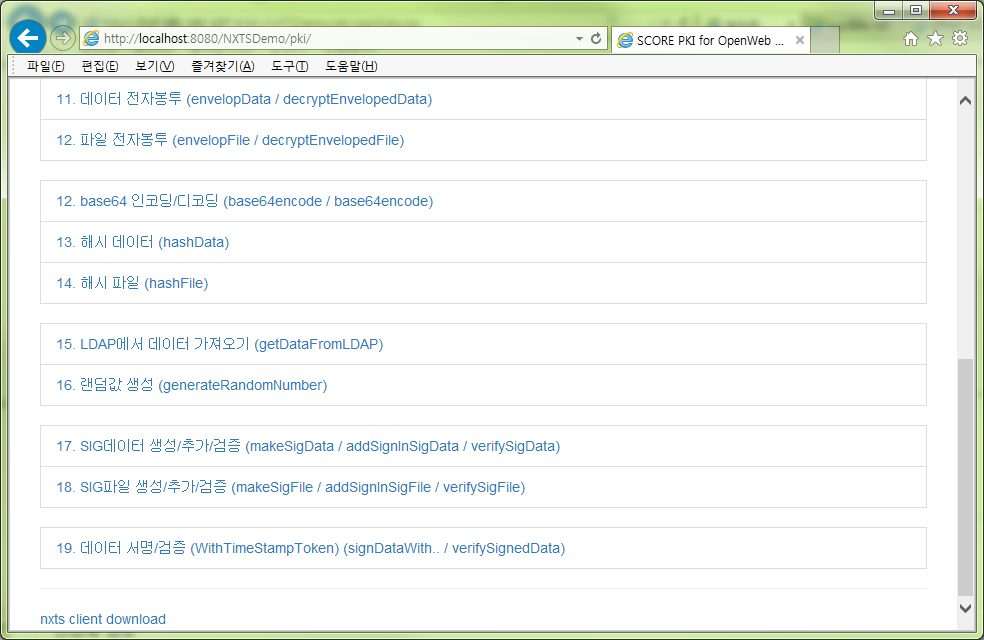


* 데모 실행 화면

Client Demo 를 클릭하면 아래처럼 다양한 샘플의 화면을 볼 수 있습니다.

각각의 버튼을 클릭하면 해당 기능의 동작을 테스트 할 수 있는 화면을 볼 수 있습니다.





제일먼저 “1. 데이터 서명/검증” 항목을 클릭하면 다음과 같은 화면이 나옵니다.

여기서 데이터 항목에 원문을 입력하고 “데이터 서명” 버튼을 클릭하면 인증서 선택창을 볼 수 있습니다. 인증서를 선택하고 패스워드를 입력하면 서명된값이 ‘서명결과” 항목에 복사가 되어 결과값을 확인 할 수 있습니다.





서명된 결과 값이 “서명결과” 항목에 복사되고 이후로는 서명한 인증서의 정보를 각 항목별로 복사되어 인증서에서 필요한 값을 가지고 갈 수 있도록 되어 있습니다.

# 설정정보

down/ 디렉토리에 있는 각 모듈의 클라이언트의 버전에 따라 각각의 nxtspki\_config.js 파일 상단에 있는 해당 모듈 (nx와, 기존 ts모듈)의 버전을 적절히 수정해야 합니다.

그렇지 않을경우, 버전 체크 오류로 업데이트 되더라도 설치 페이지로 이동하지 않거나, 반복적으로 설치페이지로 이동하는 현상이 발생 할 수도 있습니다.

설정 가능한 항목

* version.nx
  + nx클라이언트의 버전
* version.tsxmltoolkit
  + TSToolkit (PKII)의 버전
* installPage
  + 설치페이지 경로
* processingImageUrl
  + 진행중 상태를 표시할 아이콘의 경로
* nxtsdti\_config.js 수정. 예

|  |
| --- |
| nxTSPKIConfig.version.nx = "1,0,0,1";  nxTSPKIConfig.version.tsxmltoolkit = "2,0,6,4";  nxTSPKIConfig.installPage = "../install/dti/";  nxTSPKIConfig.processingImageUrl = "../img/processing.gif" |

# 설치 화면 예시

로컬서버의 설치여부를 판단하는 방법은 두가지가 있습니다.

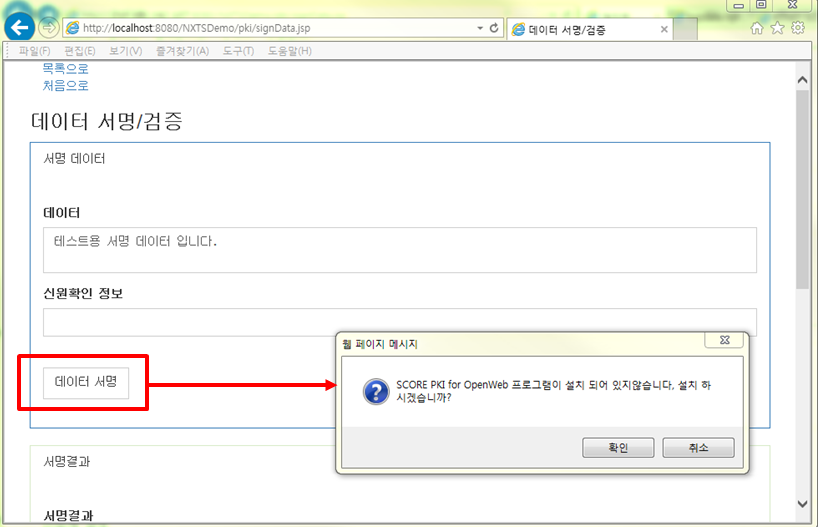
* 로컬서버의 설치여부를 확인할 수 있는 스크립트를 추가해 넣어서 페이지가 열릴 때 로컬서버의 설치여부를 체크하여 설치페이지로 포워딩 한다.

|  |
| --- |
| $(document).ready(**function**(){  nxTSPKI.installCheck();  }); |

* 툴킷 호출함수 내부에 로컬서버의 설치여부를 체크하는 기능이 기본적으로 들어 있습니다. JSP 페이지에서 별도의 체크 함수 없이 툴킷함수만 사용하여도 로컬서버가 없을 경우 자동으로 설치페이지로 포워딩 됩니다.

고객의 적용 환경에 따라 위의 두가지 환경에서 적절한 방법을 적용하시면 됩니다.

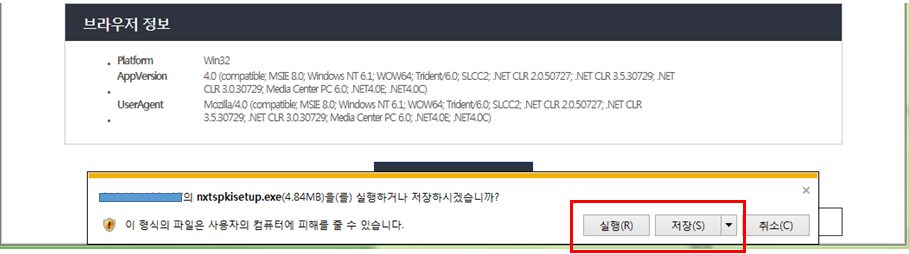
먼저 두번째 형태의 기능 동작화면을 보면 다음과 같습니다.



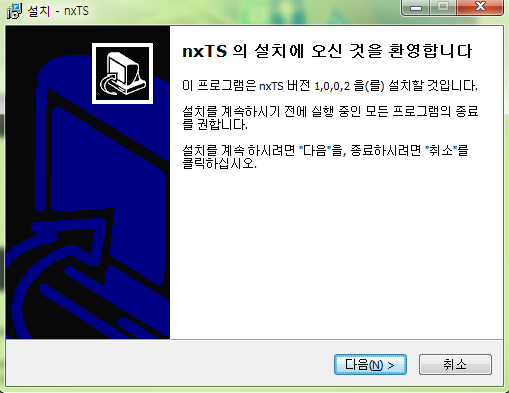
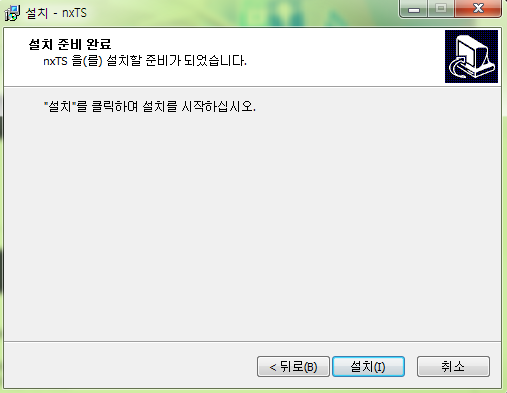
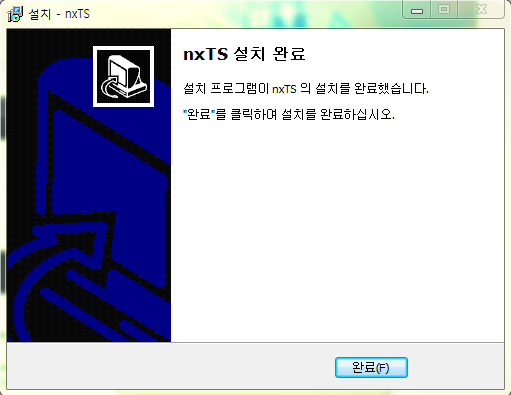
확인 버튼을 누르면 설치 페이지로 이동합니다.



“설치파일 다운로드” 버튼을 클릭을 해서 직접 exe 프로그램을 다운로드 받습니다.



“실행” 또는 ‘저장” 버튼을 눌러서 파일을 다운 받은 후 실행하시면 제품이 수동으로 설치가 됩니다.

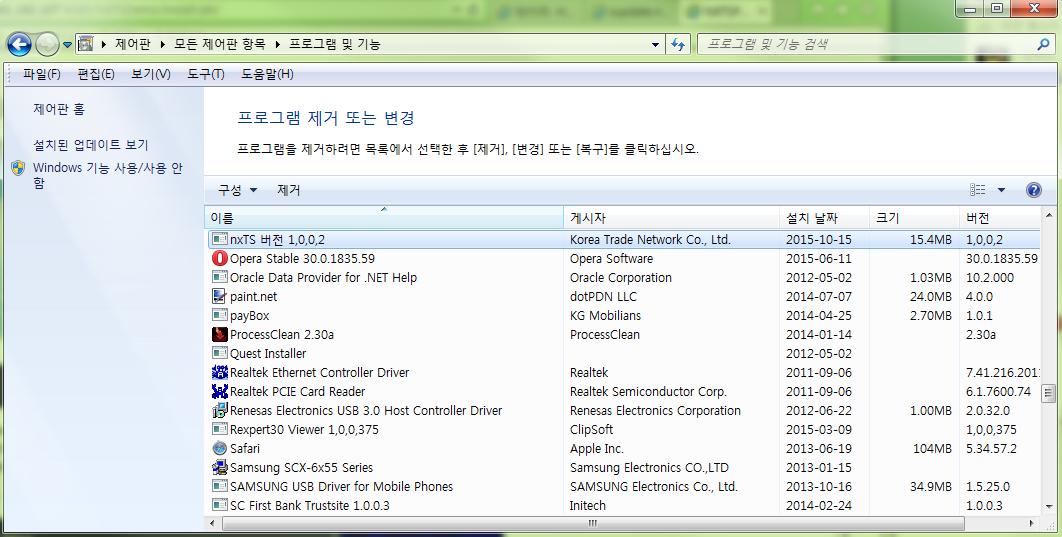
  

설치가 완료되면 이전 페이지로 다시 돌아 갑니다.

# 설치결과 확인 화면

툴킷의 정상 설치 여부를 확인하기 위해서는 제어판 -> 프로그램 및 기능 화면을 열어서 설치 제품별로 해당 항목이 있는지 확인 하시면 됩니다.

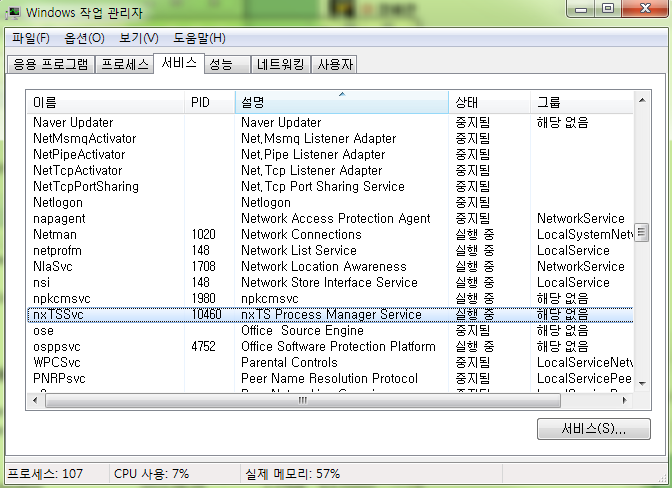
* 제어판에서 확인 방법



위의 화면은 ScorePKI for OpenWeb 제품을 설치하면 위의 2가지 항목으로 제어판에서 확인 가능합니다.

* 작업관리자에서 확인 방법

데스트탑 -> 작업관리자 시작 을 클릭하여 작업관리자 화면을 열어서 해당 프로세스를 확인 합니다

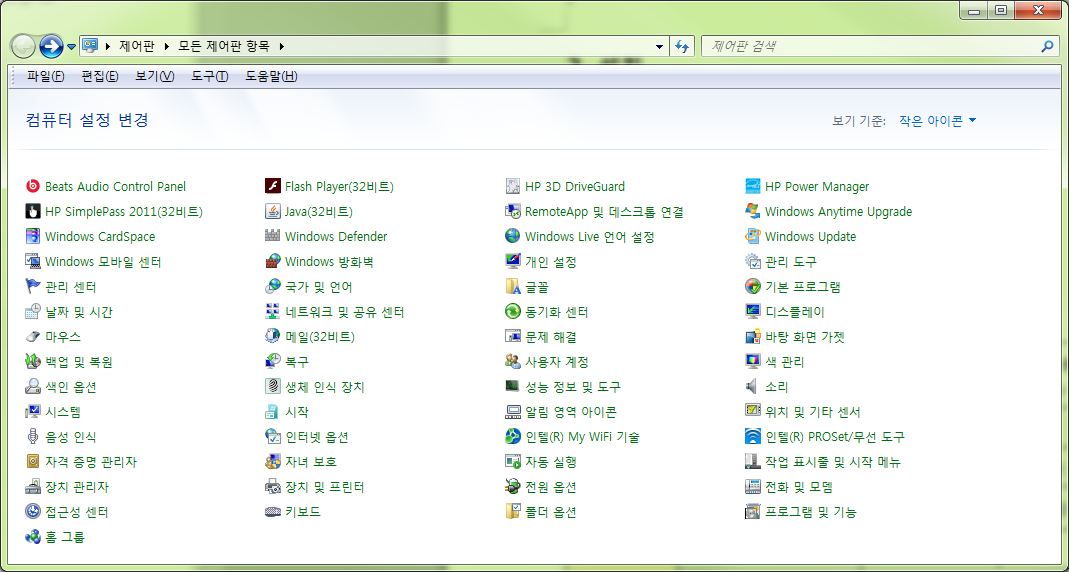


서비스 탭에서 nxTSSvc 이름으로 검색하여 해당 프로세스가 동작중인지 확인합니다. 동작중이면 설치가 정상적으로 잘 이루어진 것입니다.

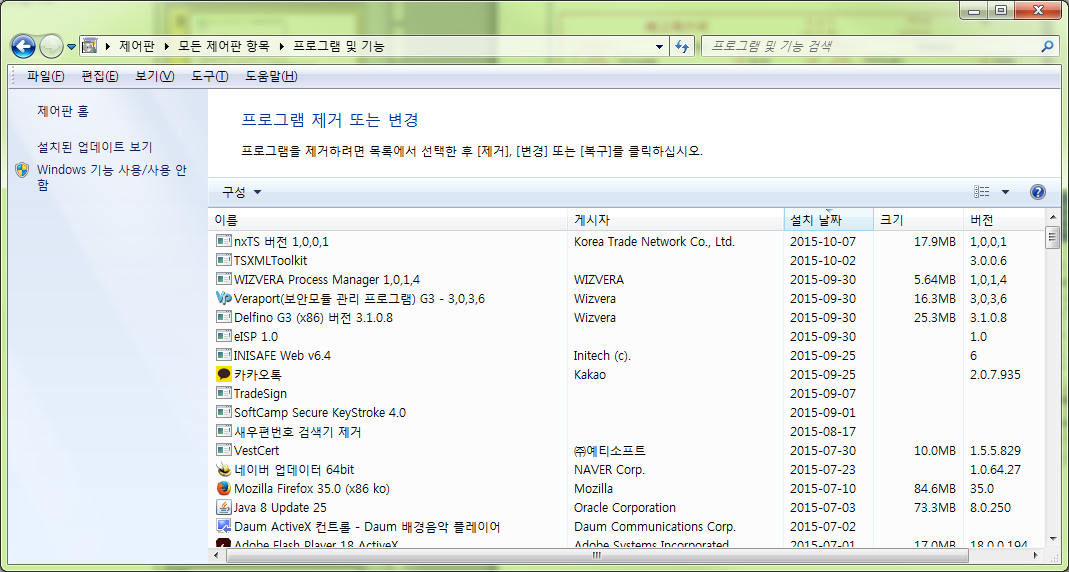
# 제품삭제 방법

설치된 서버를 삭제하기 위해서는 제어판으로 들어 갑니다.

제어판 -> 프로그램 및 기능을 클릭하면 아래의 화면을 확인할 수 있습니다.



nxTS 제품 및 TSToolkit을 선택하여 오른쪽 마우스 클릭을 하여 “제거/변경” 을 선택하면 PC에 설치된 로컬서버 및 PKI 툴킷이 삭제 됩니다.



고객사에서 적용한 툴킷에 따라 “PKI 툴킷” 및 “DTI 툴킷“의 두가지 제품이 있습니다.

적용된 제품에 따라 맞는 프로그램을 삭제하시면 해당 PC에서 관련 제품을 깨끗하게 삭제할 수 있습니다.

* PKI 툴킷 삭제 프로그램 : nxTS 와 TSToolkit 두개의 프로그램 삭제

# 버젼 업그레이드 방법

해당 제품의 업그레이드는 총 3가지로 나눌 수 있습니다.

각각의 파일별로 버전이 따로 존재합니다. 패치를 하게 되는 경우는 아래의 3가지 파일의 호환성을 고려하여 패치를 해 주셔야 합니다. 가급적이면 패치를 할 경우 3가지 종류의 파일을 모두 복사하셔서 버전별 차이로 인해 발행하는 오류가 없도록 패치 부탁 드립니다. ( 패치기 설정파일의 설정 정보는 동일하게 복사하셔서 적용 부탁 드립니다. )

* 웹서버에 올라가는 스크립트 소스

/js/demo.js

/js/nxts/\*\* -> 모든파일 (nxts.min.js, nxtspki\_config.js 등)

/install/pki/index.js -> 수동설치다운로드 화면 파일

* 웹서버에 올라가는 클라이언트 설치 프로그램

/down/nxtsdtisetup.exe -> nxTS 로컬서버와 TSToolkit 제품의 통합 설치 파일

* 툴킷 파일

/down/ktnet/TSToolkit.exe -> PKI 툴킷 수동설치 파일

# 동작전 환경 구성 (인증서 유효성 검증시 확인 사항)

데모를 동작하는 PC에서 인증서 유효성 검증 소스를 테스트 하기 위해서는 고객의 PC 환경에서 아래의 ldap 서버로 접속이 가능한지를 확인한 후 테스트를 진행하여야 합니다. ( 대부분 서버에서 검증을 하는 형태로 샘플이 제공되니 소스를 확인해 보시고 적용 부탁 드립니다. )

인증서 관련 테스트에는 해당 인증서가 유효한지 여부를 확인하기 위해 인증기관으로 데이터를 요청하여 처리하는 프로세스가 있습니다. 따라서 데모 페이지를 오류없이 동작하기 위해서는 아래의 환경을 미리 확인하고 설정하셔야 합니다.

* DNS 등록

프로그램 내에서 URL 정보로 인증기관에 접속해야 하므로 DNS 서버에 해당 URL 과 IP를 등록해야 합니다.

- DNS 등록 정보

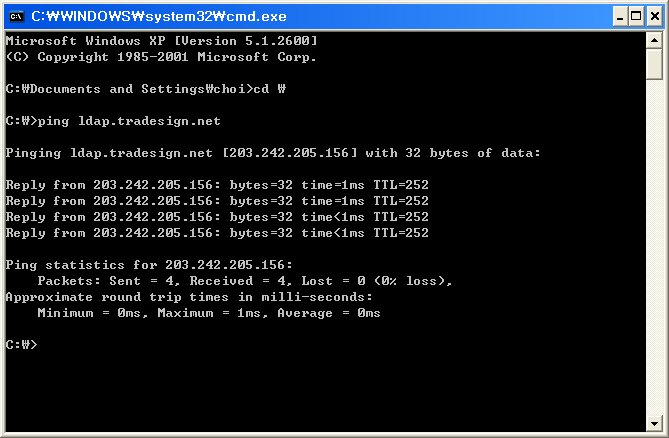
|  |
| --- |
| ldap.signgate.com 211.35.96.26 389  dir.signkorea.com 210.207.195.77 389  ds.yessign.or.kr 203.233.91.35 389  dir.crosscert.com 211.192.169.180 389  ldap.tradesign.net 203.242.205.156 389 |

\* UNIX 및 Windows 서버에 따라 등록 방법이 틀리니 해당 정보를 보시고 OS에 맞게 설정해 주세요. ( DNS 서버에 등록하기 어려우면 해당 서버의 hosts 파일에 등록해 주세요)

- 확인방법

윈도우 에서는 CMD 창을 열어서 ping 또는 telnet을 URL로 접속해서 IP 가 나오면 됩니다.

> ping ldap.tradesign.net



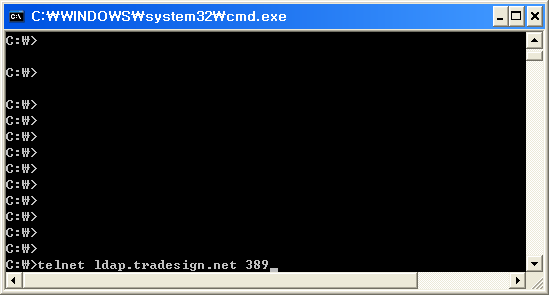
* 방화벽 열기

고객사 마다 방화벽 정책이 다릅니다. 따라서 방화벽의 정책을 변경하셔서 인증기관의 LDAP 서버로 접속해서 파일을 가져올 수 있도록 설정해야 합니다.

Tradesign.jar 파일이 설치된 서버에서 5대 인증기관으로 389 포트로 접속이 가능할 수 있도록 outbound 정책을 허용하여야 합니다.

- 확인 방법

커맨드 창에서 > telnet ldap.tardesign.net 389 를 입력한 후 엔터를 치면 화면이 변경되면서 접속된 형태로 창이 보여야 합니다.



**Q&A**

Q1. 제품을 다운로드 할 수 있는 페이지의 경로를 변경하고 싶습니다. 어느곳을 수정해야 하나요?

A1. \js\nxts\nxtspki\_config.js 파일을 열어서 아래의 변수의 값을 수정하여 적용하세요.

|  |
| --- |
| nxTSPKIConfig.installPage =  nxTSPKIConfig.installPage = "/NXTSDemo/install/pki/index.jsp";  // 제품 다운로드 페이지 (제품이 설치되지 않았으면 설정된 페이지로 이동함.)  nxTSPKIConfig.installMessage = "SCORE PKI for OpenWeb 프로그램이 설치 되어 있지 않거나, 이전 버전이 설치되어 있습니다. \n\n설치페이지로 이동하시겠습니까?";  // 제품 다운로드 페이지에서 alert 창으로 뿌려질 메시지 내용 |

##### Q2. 제품에서 js 파일을 수정해서 사용해도 될까요?

A2. 제품에 3가지의 js 파일이 포함됩니다. common.jsp 를 모든 페이지에서 추가하여 툴킷이 호출되는 모든 페이지에서 아래의 3가지 js를 추가해야 합니다. 파일도 순서대로 추가 하여야 오류 발생하지 않습니다.

|  |
| --- |
| <script src="../js/nxts/nxts.min.js"></script>  <script src="../js/nxts/nxtspki\_config.js"></script>  <script src="../js/nxts/nxtspki.js"></script> |

주의할 사항은 위의 js 파일을 열어보면 각각의 함수가 정의되어 있으며 내부적으로 필요한 여러 함수를 사용하고 있습니다.

향후 제품이 패치가 되면 파일 단위로 패치를 하게 되기 때문에 기존의 파일을 업데이트 하라고 가이드를 제공하게 될 것입니다. 따라서 js 파일을 변경하지 말아 주시기 바랍니다.

※ nxtspki\_config.js 파일에 설정하는 값들은 변경하여 설정 하여야 합니다.

##### Q3. 제품이 설치되어 있는지 여부를 확인하고 싶은데 어떻게 하나요?

A3. 제품의 설치 여부를 판단하기 위해서는 로컬서버에 요청 메시지를 보내서 응답이 없으면 설치가 되어 있지 않은 것으로 판단하여 설치 페이지로 이동하도록 되어 있습니다. 설치체크 함수는 초기화 함수에 포함되어 있습니다.

|  |
| --- |
| \install\pki\installcheck.jsp |

##### Q4. 제품의 초기화 방법

A4. 초기를 진행하는 방법은 2가지 방법이 있습니다.

해당 페이지가 열릴 때 초기화를 진행하는 방법이 있고

해당 페이지에서 사용시점에 초기화를 진행하는 방법이 있습니다.

두가지 방법 모두다 자유롭게 적용 가능하지만 팁을 드리면 한 페이지에서 여러 번 nxts함수가 불려지면 $(document).ready 에 넣어서 해당 페이지가 불려질 때 초기화를 하는 것이 낫고, 해당페이지에서 한번 nxts 함수가 불려지면 해당 함수가 불려지는 버튼에 초기화 함수를 넣는 것을 추천합니다.

※ 참고로 초기화 함수는 onInit 함수로 설정값들을 설정하고, init 함수로 설치여부를 체크하고 세션을 맺는 것 까지 가 초기화 기능 입니다.

|  |
| --- |
| //초기화  $(document).ready(function(){  nxTSPKI.onInit(function(){  //nxTSPKI.init 함수 완료 후 실행해야 하는 함수나 기능 작성 위치  //alert("Init 완료");  });  nxTSPKI.init(true);  }); |

|  |
| --- |
| nxTSPKI.**init** ( false, 함수명 ) : 사용법 설명  첫번째 파라메터는 true : 화면에 아무 표시 안함. ( 고객이 동작 여부를 알지 못함 )  false : 화면이 회색으로 보임 ( 사용자가 동작 여부를 인지하게 함 )  두번째 파라메터 는 초기화가 진행된 이후에 실행할 함수를 설정함 ( 즉 callback 함수에 들어갈 내용 이 여기에 들어감 ) |

|  |
| --- |
| 설치 여부를 판단할 때 최대 8초 정도를 기다려서 응답이 없으면 설치가 안된것으로 판단하고 있습니다.  해당 시간은 이미 지정되서 ./nxts/nxts.min.js 파일에 작성이 되어 있지만 수정은 불가 합니다.  ---- nxts.min.js 파일에 아래의 형태로 설정되어 있습니다. ----  nxTSConfig.getVersionTimeout = 5000;  nxTSConfig.scInitTimeout = 3000;  ----------------------------------------------------------- |

※ 중요

① Init 함수는 동작중에 다시 init 함수가 불려져서는 안됩니다.

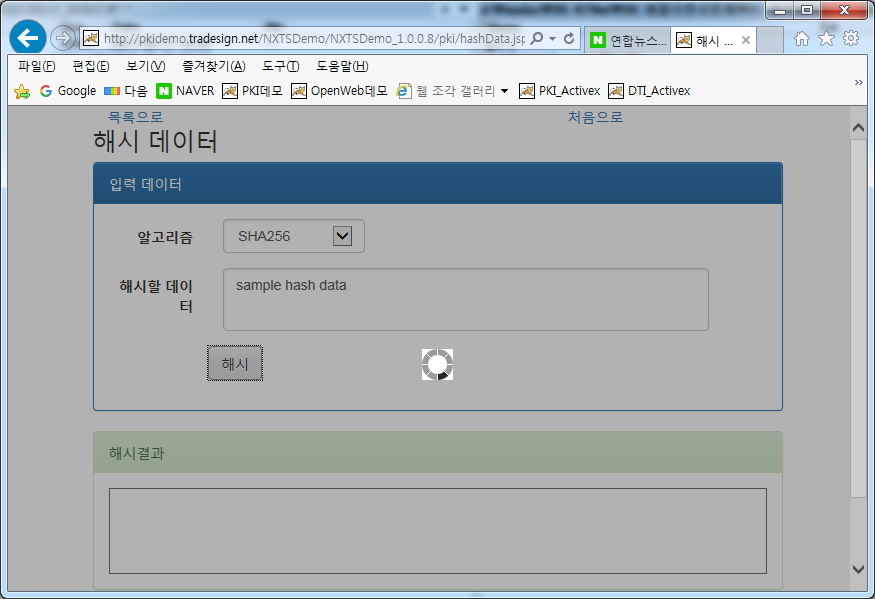
( init 함수는 약 2초 정도의 시간이 소요 됩니다. 따라서 반드시 callback 함수에서 초기화 이후의 기능을 동작하게 하도록 처리하셔야 합니다. )

② Init 함수가 끝난 후에 nxts 함수를 불러야 합니다. ( nxts 함수 내부에 init 함수가 포함되어 있음 )

##### Q5. 인증서 선택창이 뜰 때 화면이 선택창 뒤 화면의 동작 여부

인증서 선택창이 뜨면 바닥의 브라우저가 선택되거나 수정되지 않도록 하기 위해서 화면을 아래처럼 Modal 방식으로 뜨도록 하고 있습니다.

만일 선택창이 뜨지 않는 hash 함수 같은 것을 사용할 때 Modal 방식이 되지 않도록 설정하는 방법



A5. Modal 방식만 제공합니다.

고객이 화면에서 리턴값을 받기 전에 이런저런 동작을 할 수 있으면 이로 인해서 여러가지 문제가 발생할 수 있습니다. 따라서 저희 툴킷에서는 클라이언트가 동작할 동안에는 Modal 방식만 제공 가능합니다.

##### Q6. 서버에서 한글을 넣었는데 화면에서 한글이 깨어져 보여요

한글 인코딩 방법은 UTF-8 방식과 EUC-KR 방식의 두가지 방식으로 나눕니다.

고객의 프로세스에서 서버 모듈과 클라이언트모듈을 함께 연동해서 동작한다고 하면 한글을 입력할 때 인코딩 타입을 잘 맞추어 사용하여야 깨어지지 않습니다. 대부분의 샘플이 이미 잘 맞추어져 있지만 고객사 별로 특별한 인코딩 방법을 사용하여야 하는 경우에는 명시적으로 선언을 하여 한글이 깨어지 않도록 처리 할 수 있습니다.

* 한글 인코딩의 기본 개념

|  |  |  |
| --- | --- | --- |
| 종류 | 구분 | 특징 |
| UTF-8 | 조합형 | 대표적인 유니코드 인코딩 방식,  ASCII 문자들을 표현할 수 있음.  한글 3바이트로, 영문 및 공백은 1바이트로 처리 |
| EUC-KR | 완성형 | 윈도우 운영체계에서 처리하는 방식  한글을 2바이트 처리, |

* 제품 내부에서 한글 인코딩 처리 방법

저희 제품은 가장 많이 동작하는 환경이 윈도우 환경이므로 모듈 내부에서 한글 처리 방법은 멀티바이트 형태(euc-kr)로 처리하고 있습니다.

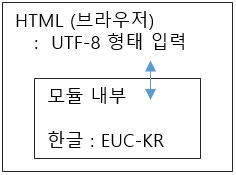
여기서 주의할 사항은 보안모듈 내부에서는 한글을 euc-kr 형태로 처리하기는 하지만, 고객사 및 지원환경에서 API를 처리하는 환경은 대부분 UTF-8을 사용하고 있기 때문에 외부와의 인터페이스는 한글을 UTF-8로 처리하고 있습니다.

* 서버 java 제품에서 사용 예

Java 제품은 jar 파일을 바로 호출하여 직접 사용하는 형태 입니다. 따라서 한글을 아래처럼 멀티바이트 처리하여 입력해 주시면 됩니다.

|  |
| --- |
| \*\* 원문을 입력할 때  String plainText = new String(“한글나라123abc”, "euc-kr") ; // 멀티 바이트로 한글 입력 |

* OpenWeb / Activex 제품에서 사용 예



먼저 javaScript 나 html 단에서의 처리는 모두 UTF-8로 처리를 하셔야 합니다.

|  |
| --- |
| <%@ page contentType="text/html; charset=utf-8" %> |

윈도우 모듈에서는 한글을 입력 받으면 UTF-8을 멀티바이트로 변환한 다음에 처리하고 있습니다. 응답 결과도 Dll 내부에서는 멀티바이트 처리를 하고 이후에 응답결과는 UTF-8 인코딩 형태로 변환해서 전달하고 있습니다.