|  |
| --- |
|  |
| **NonActiveX 설계 방식 참고** |
| 인지소프트 최하영 연구원 |

|  |
| --- |
|  |

[목록 .](#_Toc527479393)

[1 Inzi NonActiveX개요 3](#_Toc527479393)

2. [AJAX](#_Toc527479393) 3

[- 기술개요 3](#_Toc527479393)

[- 장 / 단점 3](#_Toc527479393)

[­- 적용방법 3](#_Toc527479393)

[- NonActiveX 적용 3](#_Toc527479393)

[- Web 예제 3](#_Toc527479393)

[3. WebSocket 3](#_Toc527479393)

[- 기술개요 3](#_Toc527479393)

[- 장 / 단점 3](#_Toc527479393)

[­- 적용방법 3](#_Toc527479393)

[- NonActiveX 적용 3](#_Toc527479393)

[- Web 예제 3](#_Toc527479393)

[4. 고려해볼 사항 3](#_Toc527479393)

[- 이미지를 어떻게 받을것인가 3](#_Toc527479393)

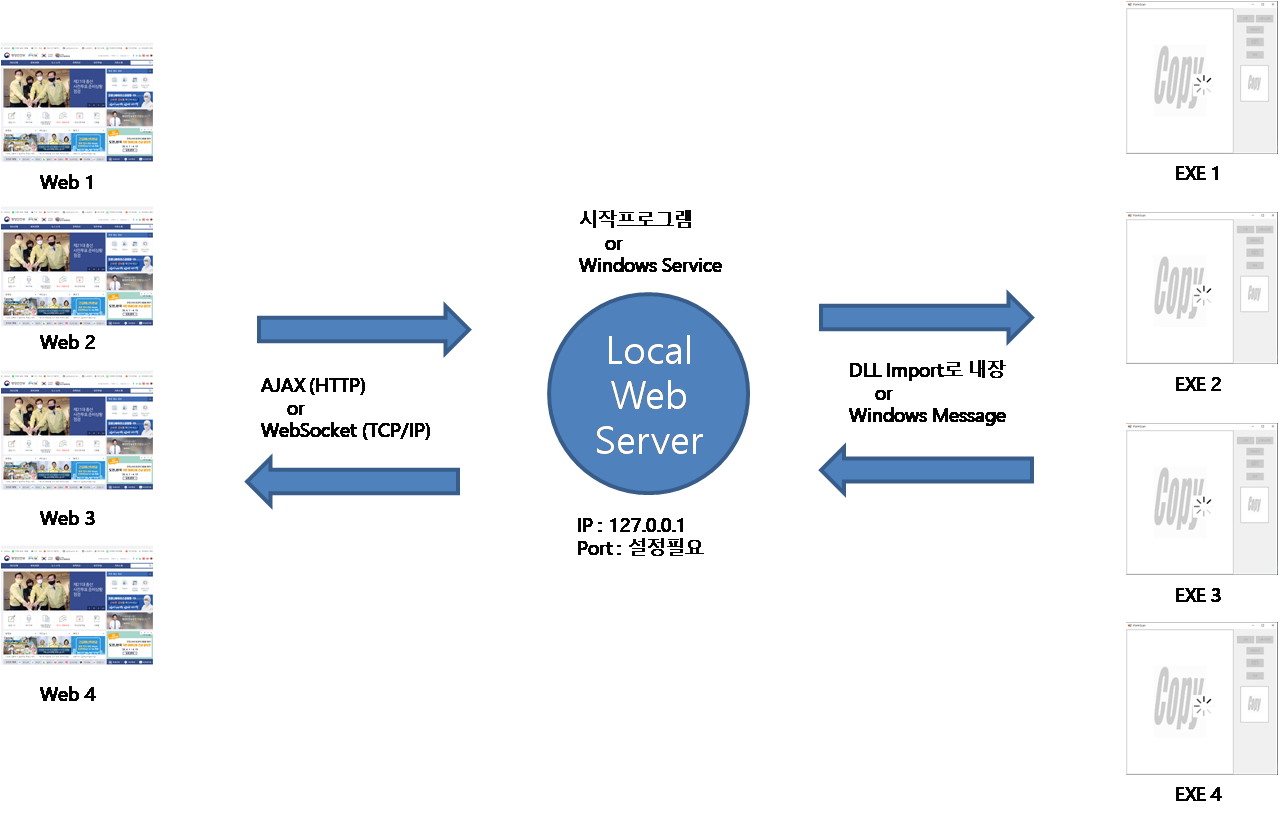
# inzi NonActiveX 개요

inzi\_iFormScan NonActiveX는 기존의 ActiveX(OCX)를 모두 제거 하고 기존 웹 화면과 같은 기능을 수행하는 EXE(실행파일)를 직접 실행시키고, 로컬 웹 서버를 이용해 브라우저와 스캔 EXE간의 통신, I/O를 가능하게 하여 웹에서 호출 한 EXE 수행 결과 값을 브라우저로 리턴 받을 수 있습니다.

브라우저 단에서는 EXE를 실행시키기 위해 custom URI를 사용하고 로컬 웹 서버와 Ajax 통신 또는 을 통하여 EXE와 통신 및 I/O작업을 할 수 있습니다.

제약사항 : OS - Windows 계열 Windows 7이상 사용가능. (상세 환경은 EXE의 .NET 개발 버전에 따름)  
 브라우저 – AJAX, WebSocket 웹 표준을 지원하는 모든 브라우저 사용가능

1. NonActiveX 기본 구조



WEB과 EXE를 통신시키기 위해서는 사용자 PC에 Local Web Server가 필요하며,  
이 Local Web Server는 별도 Service로 존재하거나 EXE가 실행되면 동시에 사용자 PC에 올라갑니다.  
  
WEB과 EXE의 통신은 이 Local Web Server를 통해서 이루어지며,   
Web에서 AJAX(HTTP)통신 혹은 WebSocket(TCP/IP)의 웹표준 기술을 이용하여 EXE에 값 요청 및   
응답이 가능합니다.

# AJAX

1. 기술 개요  
   AJAX는 Asynchronous Javascript And XML의 약자로써 기존의 Form으로부터 Submit하여 새로고침되어 모든 정보를 다시 받아오는 HTTP Submit 방식에서 필요한 데이터만을 비동기로 웹서버에 요청해서 받은 후 클라이언트에서 데이터에 대한 처리를 할 수 있다.   
     
   통신 방법 : TCP/IP - HTTP (Request / Response)
2. **장점/단점**▪ 장점

* 페이지 이동없이 고속으로 화면을 전환할 수 있다.
* 서버 처리를 기다리지 않고, 비동기 요청이 가능하다.
* 수신하는 데이터 양을 줄일 수 있고, 클라이언트에게 처리를 위임할 수도 있다.
* **WebSocket보다 과거, 버전의 브라우저까지 지원이가능하다. (IE8, 9)**

▪ 단점

* Ajax를 쓸 수 없는 브라우저에 대한 문제가 있다.
* HTTP 클라이언트의 기능이 한정되어 있다.
* 페이지 이동없는 통신으로 인한 보안상의 문제
* 지원하는 Charset이 한정되어 있다.
* 스크립트로 작성되므로 디버깅이 용이하지 않다.
* 요청을 남발하면 역으로 서버 부하가 늘 수 있음.
* 동일-출처 정책으로 인해 다른 도메인과는 통신이 불가능하다.
* **HTTP 통신이므로 클라이언트와 지속 통신을 하기 위해서는 Request를 지속 요청해야 된다.**

1. **NonActiveX 적용 방법**

1. EXE 실행 (로컬 웹서버를 통해 EXE 실행)  
**2. Connect : 로컬 웹 서버를 통해 EXE에 Request 신호 보내기   
 (AJAX – HTTP Request)**3. EXE 작업 후 결과 값 **콜백함수**로 받아오기 (AJAX – HTTP Response)  
4. EXE로부터 받은 결과데이터 처리.   
 - 혹시 다시 EXE로부터 지속적으로 통신이 필요하면 2,3번 반복 (HTTP Request <-> Response)

※ 4단계에서 다시 Connect 신호(Request)를 보내는 이유는 구조상 EXE에서 WEB으로 일방적으로 데이터를  
전달 할 수 없어, WEB에서 EXE로 요청을 보내고 Response로 리턴(CallBack함수 이용)을 받는 방법 밖에 없습니다.   
따라서 Connect(Request)신호를 보낸 후 Ajax통신으로(HTTP Request) Callback을 걸어놓은 뒤, EXE에서 작업수행 후 결과값을 보내면 걸어놓은 Callback 함수로 리턴값을 받아서(HTTP Response) 처리 합니다. 만약 다시 EXE에서 결과값을 받아와야 되는 상황이라면, 다시 START신호를 통해 Callback을 걸어두어야 합니다.

1. **웹 예제  
   **

# WebSocket

1. 기술 개요  
   WebSocket은 말 그대로 웹브라우저에서 Socket통신을 한다는 것이며, Socket통신이기 때문에  
   TCP/IP 통신 프로토콜을 사용한다.  
   WebSocket은 기존에 존재하던 HTTP의 문제를 해결 하기위해 나왔는데, HTTP에서 원리적으로 해결할 수 없었던 문제는 “클라이언트의 요청이 없음에도, 그 다음 서버로부터 응답을 받는 상황”이었다. WebSocket은 HTTP가 해결할 수 없었던 이 문제를 해결하는 새로운 웹 표준이며,   
   브라우저가 서버에 데이터를 요청하고 서버가 웹브라우저에 데이터를 보내기 위해 별다른 제약이 없다. 하지만 AJAX와 동일하게 사용하는 웹브라우저가 WebSocket을 지원해야만 한다.  
     
   통신 방법 : TCP/IP - Socket
2. **장점/단점**▪ 장점

* **양방향 통신**
* **실시간 통신 / 성능**

▪ 단점

* 실시간 양방향 통신(Socket)이므로 많은 비용을 지불해야함.
* **오래된 버전의 웹 브라우저에서는 지원하지 않음.**
* 인터넷 익스플로어 같은 경우 10 버전부터 지원.
* 프로그램 구현에 보다 많은 복잡성 초래.
* 디버깅의 어려움.

1. **NonActiveX 적용 방법**

1. EXE 실행 (로컬 웹서버를 통해 EXE 실행)  
**2. Connect : 로컬 웹 서버를 통해 EXE에 Socket Connection 신호 보내기   
 (WebSocket – Socket Connect)**3. EXE 작업 후 결과 값 **onMessage 이벤트**로 받아오기 (Socket Receive Event)  
4. EXE로부터 받은 결과데이터 처리.   
5. WebSocket- **Send 메소드**를 사용하여 EXE에 다시 메시지 보내기  
6. 3~5 반복.

1. **웹 예제  
   **

# 고려해볼사항

1. **이미지를 어떻게 받을 것인가  
   방법 1. 스캔 및 작업된 이미지의 파일 경로를 받아 처리** -> 주고받는 데이터 크기가 작으나 로컬파일경로를 브라우저에서 처리하므로  
    권한 이슈가 발생 할 수 있다.  
   **방법 2. 바이트값 Base64 String으로 받기**  
    -> 권한이슈는 없으나 받는 데이터의 크기가 커서 퍼포먼스나 다른 문제가 발생 할 수 있다.