A method for replacing old ActiveX which is no longer supported in IE.

General approach: Install a daemon on the desktop and use a Websocket to communicate between a page and the server in order to launch local commands.

Beware of the potential risks of this method because we open a breach on the PC but we can still control the level of risk.
ie: Because a malicious js code could call the websocket, I would advise to improve this prototype by adding a token (One-Time Password) which will be shared between the server and the genuine page.
Never allow to send directly a windows/DOS command !

In the following case, the dameon is implemented in node.js

- Install Node.js
- Install modules for webSocket, FileSystem and Node Commands (ws, fs and node-cmd)
- Write the js script server which will listen for the commands to be executed
- Write the js script to be included in the html page. The js will send requests through a websocket.

Code of the Node.js server

```
// ------------------------------------------------------------------
// serverws.js
// v0.1
// GPM FACTORY
// Nov 2023
//
// Websocket server used for creating files and
```


```
// triggering a limited set of OS commands
// usage : node serversws.js
// ----------------------------------------------------------------

const WebSocket = require('ws')
const fs = require('fs');
const nodeCmd = require('node-cmd')
let content
let fidir = ''
const LOG = 1       // 1= Yes, 0= No
const PORT = '8088'
const U_DIR = 'C:/_main/Projets/chronoA/out/'
const C_DIR = 'C:/_main/Projets/chronoB/out/'
const C1 = 'DIR'   // 'C:/temp/print_acrobat.cmd'
//const ss = require('stream-string')  // not needed
const wss = new WebSocket.Server({ port: PORT })

// Each message from client is structured as :
// For a file:
//   #<FILE_TYPE>#<FILE_NAME>#<CONTENT>
//   with <FILE_TYPE> = U or C
// For an OS command:
//  !<COMMAND_NUMBER>

wss.on('connection', ws => {
    ws.on('message', function message(data) {
        content = '' + data
        if (content.substring(0,1) == '#') {
            // it's a file content
            // we must detect the type of content
            let fity = content.substring(1,2)
            if (fity == 'U'){

                fidir = U_DIR
```

```
               }
            if (fity == 'C'){

                fidir = C_DIR
            }

            let nend = content.indexOf("#",3)
            let finame = content.substring(3,nend)
            if (LOG == 1) {
                console.log("file name=" + finame)
            }
            let writeStream = fs.createWriteStream(fidir+finame)
            writeStream.write(content.substr(nend+1))
            writeStream.end()
            ws.send('File has been created')
        }
        if (content.substring(0,1) == '!') {
            // it's a command
            // we avoid to pass an arbitrary command because potential
securiy issues. Instead, we pass a command type (!TBD)
            cmd = content.substring(1)
            if (LOG == 1) {
                console.log("commande="+ cmd)
            }
            if (cmd == '1') {
                nodeCmd.run(C1, (err, data, stderr) =>
console.log(data));
            }
            else {
                ws.send('Unknown command')
            }
         }
        if (LOG == 1) {
            console.log('received: %s', data)
```

```
        }
    });


})
```

Code of the JS script in the HTML page

```
<SCRIPT LANGUAGE="JavaScript">
        function makefile(){
                var fso;var thefile;


    ws.onopen = () => {
      console.log('Send the chrono to ws');

       finame = 'etiq-<?php echo $_GET["ord_id"]?>.csv';
        // #C# means : CHRONO
      msg = '#C#'+ finame + '#' + document.tags.chrono.value;
      ws.send(msg)
    }
    ws.onmessage = (message) => {
        alert (message.data);
        console.log('message received', message.data)
    }
//    alert('Le fichier est crée.');
}
```