Tutorial For Synchronized Scanning

If you want to get some data from the scanner quickly , here's a small program that uses imaginary data types and functions to demonstrate how to use the scanner in Encoder Synchronized Mode:

1. Call `jsInitialize()` if the Joescan API is statically linked.
2. Open a connection using `jsOpenConnection()`.
3. Save the parameters using `jsSendParameterFileToScanner()` in case a scanner has been swapped out.
4. Put it into encoder mode using `jsEnterEncoderSyncMode()`.
5. Get profiles using `jsGetProfile()`. Be careful to not create a hard loop if no profiles are currently available.
6. End encoder mode using `jsExitSyncMode()`.
7. Close the connection using `jsCloseConnection()`.

```
#include <stdio.h>
#define JCAM_STATIC_LIB
#include "jcam_dll.h"

using namespace joescan;

int main(int argc, char **argv)
{
    if(jsInitialize() == FALSE)
    {
        printf("Could not initialize TCP\IP communications.\n");
        return -1;
    }

    char *ip = "192.168.1.205";
    if(argc > 1)
        ip = argv[1];

    JCONNECTION jc = jsOpenConnection(ip);
    if(jc == 0)
    {
        printf("Could not open connection to %s.\n", ip);
        return -1;
    }

    switch(jsSendParameterFileToScanner(jc, "param.dat"))
    {
        case SCANNER_FAILURE:
            printf("Lost connection.\n");
            jsCloseConnection(jc);
            jc = 0;
            return -1;
        case OPERATION_FAILURE:
            for(i = 0; i < jsGetNumberOfErrorMessages(jc); i++)
                printf("%s\n", jsGetErrorMessage(jc, i));
            jsCloseConnection(jc);
            jc = 0;
            return -1;
        default:
            printf("Sent parameters successfully.\n");
    }

    switch(jsEnterEncoderSyncMode(jc))
    {
        case 0:
            break;
        case INVALID_PARAMETER:
            printf("Invalid parameter to jsEnterEncoderSyncMode().\n");
            jsCloseConnection(jc);
            jc = 0;
            return -1;
        case SCANNER_FAILURE:
            printf("Scanner failure during jsEnterEncoderSyncMode().\n");
            jsCloseConnection(jc);
            jc = 0;
            return -1;
    }

    jsProfile profile;
    while(PLC.isLogVisible())
    {
        switch(jsGetProfile(jc, &profile))
        {
            case 0:
                logModel.addProfile(profile);
                break;
            case PROFILE_UNAVAILABLE:
```

```
                optimizer.earlyOptimize(logModel);
                break;
            case INVALID_PARAMETER:
                printf("Invalid parameter to jsGetProfile().\n");
                jsCloseConnection(jc);
                jc = 0;
                return -1;
            case SCANNER_FAILURE:
                printf("Scanner failure during jsGetProfile().\n");
                jsCloseConnection(jc);
                jc = 0;
                return -1;
        }
    }

    switch(jsExitSyncMode(jc))
    {
        case 0:
            break;
        case INVALID_PARAMETER:
            printf("Invalid parameter to jsExitSyncMode().\n");
            jsCloseConnection(jc);
            jc = 0;
            return -1;
        case SCANNER_FAILURE:
            printf("Scanner failure during jsExitSyncMode().\n");
            jsCloseConnection(jc);
            jc = 0;
            return -1;
    }

    jsCloseConnection(jc);
    jc = 0;
    optimizer.fullyOptimize(logModel);
    return 0;
}
```