

Example 4: On-Demand Scanning

After connecting to a scanner head and configuring it, you can now request data. There are several ways to do this, but the simplest is On-Demand Scanning. By default, the scanner head starts in this mode, and all you have to do to get data is to call [Scanner.GetProfile\(\)](#) .

There are a number of considerations when using On-Demand Scanning.

- **Speed:** The scan speed in this mode is lower than in Synchronized Mode, because the head cannot effectively parallelize readouts and scan requests.
- **Timing:** The timing is entirely dependent on the readout loop of your controlling computer. If the object in the scanner's view is moving, the distance between consecutive profiles may be different, depending on the processing time needed in the readout loop.
- **Synchronization:** In On-Demand mode, synchronizing multiple scan heads is not possible.

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Net;
using JoeScan.JCamNet;

namespace Examples
{
    class Program
    {
        static void Main(string[] args)
        {
            IPAddress baseIpAddress = new IPAddress(new byte[] { 192, 168, 1, 150 });
            Scanner scanner = null;
            try
            {
                scanner = Scanner.Connect(baseIpAddress, 0);
            }
            catch (Exception e)
            {
                Console.WriteLine("Failed to connect to scanner: {0}, Reason: {1}",
                    baseIpAddress.ToString(), e.Message);
                return;
            }
            TextReader parametersReader;
            using (parametersReader = File.OpenText("param.dat"))
            {
                string parametersString = parametersReader.ReadToEnd();
                try
                {
                    scanner.SetParameters(parametersString, true);
                }
                catch (Exception e)
                {
                    Console.WriteLine("Scanner {0}:{1} did not accept configuration parameters!",
                        scanner.IPAddress.ToString(), scanner.CableID);
                    return;
                }
            }
            // Now we're ready to scan. In this example, we retrieve a fixed number of Profiles from
            // the scan head. In practice, this would be done in a loop.
            List<Profile> profiles = new List<Profile>();
            for (int i = 0; i < 10; i++)
            {
                try
                {
                    Profile p = scanner.GetProfile();
                    Console.WriteLine("Profile {0}: measured {1} points.", i, p.Count);
                }
                catch(Exception e)
                {
                    Console.WriteLine("Error retrieving profile from scanner. Exiting.");
                    return;
                }
            }
        }
    }
}

```

Sample output is:

```
Profile 0: measured 218 points.  
Profile 1: measured 218 points.  
Profile 2: measured 218 points.  
Profile 3: measured 218 points.  
Profile 4: measured 217 points.  
Profile 5: measured 218 points.  
Profile 6: measured 218 points.  
Profile 7: measured 217 points.  
Profile 8: measured 217 points.  
Profile 9: measured 218 points.
```



The number of measured points in the example above fluctuates due to external influences (e.g., ambient light). If you don't receive any points, always check with JSdiag to see if your scan window is set up correctly and that LaserOn times are sufficient to produce good data.