public static void Main1()

            {

                UAEndpointDescriptor endpointDescriptor =

                    "opc.tcp://[192.168.1.2:4840/Amada\_Events](http://192.168.1.2:4840/Amada_Events)";

                //

                //opc.tcp://[opcua.demo-this.com:62544/Quickstarts/AlarmConditionServer](http://opcua.demo-this.com:62544/Quickstarts/AlarmConditionServer)

                // Instantiate the client object and hook events.

                UAMonitoredItemArguments myargument = new UAMonitoredItemArguments();

                myargument.AutoConditionRefresh = false;

                var client = new EasyUAClient();

                client.EventNotification += client\_EventNotification;

                Console.WriteLine("Subscribing...");

                client.SubscribeEvent(endpointDescriptor, UAObjectIds.Server, 1000);

                //client.SubscribeEvent(endpointDescriptor, UAObjectIds.ObjectsFolder, 1000);

                Console.WriteLine("Processing event notifications for 30 seconds...");

                System.Threading.Thread.Sleep(30 \* 1000);

                Console.WriteLine("Unsubscribing...");

                client.UnsubscribeAllMonitoredItems();

                Console.WriteLine("Waiting for 5 seconds...");

                System.Threading.Thread.Sleep(5 \* 1000);

                Console.WriteLine("Finished.");

            }

            static void client\_EventNotification(object sender, EasyUAEventNotificationEventArgs e)

            {

                Console.WriteLine();

                // Display the event.

                if (e.EventData is null)

                {

                    Console.WriteLine(e);

                    return;

                }

                Console.WriteLine("All fields:");

                foreach (KeyValuePair<UAAttributeField, ValueResult> pair in e.EventData.FieldResults)

                {

                    UAAttributeField attributeField = pair.Key;

                    ValueResult valueResult = pair.Value;

                    Console.WriteLine("  {0} -> {1}", attributeField, valueResult);

                }

                // Extracting a specific field using a standard operand symbol.

                Console.WriteLine($"Source name: {e.EventData.FieldResults[UABaseEventObject.Operands.SourceName]}");

                // Extracting a specific field using an event type ID and a simple relative path.

                Console.WriteLine(

                    $"Message: {e.EventData.FieldResults[UAFilterElements.SimpleAttribute(UAObjectTypeIds.BaseEventType, "/Message")]}");

            }