

The OPC-UA subscription publishing has stopped.

Performance repeat an error in the communicating by a "keep mechanism: Mechanism OPC UA". At this time contain some "tag" (read/write date) either the server or client is no longer communicating periodically while the connecting is still operational. After a certain time, automatically discover restored of the communication of the tags (read/write date).

I check some items:

1. The server is PLC (Bosch Rexroth) and client PC (HP I5) works not on the same. This hardware is connecting via a local network:
 - Bosch Rexroth PLC "IP 172.17.1.1" and PC "IP 17.17.0.110" the networks process environment is 0.04% in use this is/was monitor via the task manager of the PC.
2. There isn't a network connectivity losses:
 - Via the tool Wireshark we can follow that the message Publish Response PC and Publish Request PLC are continuously available and the read/write message are also in good ranges.
3. The processor use is between 3 till 14% depends of the activities this is/was monitor via the task manager of the PC.
4. The software is not running in debug but in release mode.

In are application software for the OPC UAClient connection I did changed the Opc Compliance parameter to "KeepAliveInterval = 60 * 1000; " look below:

UAClient.cs

```
public UAClient()
{
    //EasyUAClient.AdaptableParameters = EasyUAA adaptableParameters.OpcCompliance;
    uaClient = new EasyUAClient();
    //uaClient.Isolated = true;
    //uaClient.IsolatedParameters.Session.RetrialDelay = 5 * 1000;

    // AD 20150716 The OPC-UA subscription publishing has stopped
    uaClient.IsolatedParameters.Session.KeepAliveInterval = 60 * 1000;

    //uaClient.IsolatedParameters.Session.ServerShutdownRetrialDelay = 15 * 1000;
    //NodeDescriptorFormat = string.Format("{0}{0}", Settings.Default.UANodeDescriptor);
}

public static void SetOpcCompliance()
{
    EasyUAClient.AdaptableParameters = EasyUAA adaptableParameters.OpcCompliance;
    //EasyUAClient.AdaptableParameters.Session.RetrialDelay = 5 * 1000;

    // AD 20150716 The OPC-UA subscription publishing has stopped
    EasyUAClient.AdaptableParameters.Session.KeepAliveInterval = 60 * 1000;

    //EasyUAClient.AdaptableParameters.Session.ServerShutdownRetrialDelay = 15 * 1000;
    //EasyUAClient.AdaptableParameters.Session.SessionTimeout = 24 * 60 * 60 * 1000;
    //EasyUAClient.AdaptableParameters.Session.CloseTimeout = 10 * 1000;
    //EasyUAClient.AdaptableParameters.Session.HoldPeriod = 60 * 1000;
    //EasyUAClient.AdaptableParameters.Session.SessionConnectTimeout = 60 * 1000;
    //EasyUAClient.AdaptableParameters.Session.SessionTimeout = 60 * 1000;

    //EasyUAClient.SharedParameters.Engine.ApplicationCertificateSubject = "DGpS.WindowsService.PLCMessages";
    //EasyUAClient.SharedParameters.Engine.ApplicationName = "DGpS.WindowsService.PLCMessages";
    //EasyUAClient.SharedParameters.Engine.ApplicationUriString = "urn:DGPress-VM:DGpS.WindowsService.PLCMessages:1.0.0.0:neutral:null";
    //EasyUAClient.SharedParameters.Engine.ProductUriString = "DGpS.WindowsService.PLCMessages";
    //EasyUAClient.Install();
}
```

Example to subscribe:

```
public void Subscribe(SubscriptionParameter subscription)
{
    _subscription = subscription;
    if (_subscription != null)
    {
        if (!string.IsNullOrEmpty(subscription.Format))
        {
            Format = subscription.Format;
        }

        _handle = Opc.Subscribe(subscription.ServerUri, subscription.NodeId, subscription.Interval, OPC_MonitoredItemChanged);
        Enabled = true;
    }
}

private void OPC_MonitoredItemChanged(object sender, OpcLabs.EasyOpc.UA.OperationModel.EasyUAMonitoredItemChangedEventArgs e)
{
    if (e.Exception == null)
    {
        if (e.AttributeData.HasValue())
        {
            Text = string.Format(CultureInfo.InvariantCulture, FormatString, e.AttributeData.Value);
        }
    }
}
```

```
    }  
  }  
  else  
  {  
    Text = NO_LINK;  
    Logger.Write(string.Format("{0}: {1}", _subscription.NodeId, e.Exception.GetBaseException().Message), Category.Error);  
  }  
}
```

PopupOPCTextbox.cs

Example of subscribe and call back. When an error is available then we show a "NO LINK!" text in the text box and write the error to a log file.

```
public void Subscribe(SubscriptionParameter subscription)  
{  
  _subscription = subscription;  
  if (_subscription != null)  
  {  
    if (!string.IsNullOrEmpty(subscription.Format))  
    {  
      Format = subscription.Format;  
    }  
  
    _handle = Opc.Subscribe(subscription.ServerUri, subscription.NodeId, subscription.Interval, OPC_MonitoredItemChanged);  
    Enabled = true;  
  }  
}  
  
private void OPC_MonitoredItemChanged(object sender, OpcLabs.EasyOpc.UA.OperationModel.EasyUAMonitoredItemChangedEventArgs e)  
{  
  if (e.Exception == null)  
  {  
    if (e.AttributeData.HasValue())  
    {  
      Text = string.Format(CultureInfo.InvariantCulture, FormatString, e.AttributeData.Value);  
    }  
  }  
  else  
  {  
    Text = NO_LINK;  
    Logger.Write(string.Format("{0}: {1}", _subscription.NodeId, e.Exception.GetBaseException().Message), Category.Error);  
  }  
}
```

After different testing, we wish to solve the problems. We hope that you can help use. If you have any questions or remarks, please let me know.

Met vriendelijke groeten / Yours Sincerely / Mit freundlichen Grüßen,
Alex Derksen
Software Engineer



▪ Hallseweg 21 ▪ 6964 AJ Hall ▪ The Netherlands ▪ [tel:+31 313 671911](tel:+31313671911) ▪ fax:+31 313 671919
▪ e-mail: a.derksen@dgpsservices.com ▪ website: www.dgpsservices.com