



Tip of the Week :

- How to use “Grouped”

[Grouping Links]

- Client Program has **several** persistent links whose data somehow “**belong together**”.
- It only makes sense to “**examine the data**” or “**refresh the display**” when everything has been updated.
- Get only **one** asynchronous event notification instead of several !

[Grouping Links]

- Only makes sense if
 - Asynchronous reads!
 - All links use the same transfer mode!
 - Transfer modes:
 - SINGLE – okay !
 - REFRESH (DATACHANGE) - not a good idea !
 - POLL (TIMER) – okay !
 - EVENT – not a good idea !

[Grouping Links (C API)]

```
for (i=0; i<numInGroup; i++) {  
    id[i] = AttachLink(dev[i],prp[i],&dout,NULL,CA_READ,200,cbGrp,CM_POLL|CM_GROUPED);  
}  
...  
void cbGrp(int id, int cc)  
{  
    // id = id of the last link to come in  
    if (cc == 0) // everybody came in successfully  
    else loop over numInGroup and use either  
        status = GetCurrentLinkStatus(i);  
    or  
        status = GetCurrentLinkStatusFromCallbackId(i);  
    to get each link status.  
}
```

Optimistic programming ?



Can also use the “CM_USEONERROR” switch !!

- If link status != 0, data entries will be replaced by the error value given!

Grouping Links (ActiveX API)

```
Project1 - Form1 (Code)
(General) getCollectedCounts

Sub getCollectedCounts(ByVal starttime As Long, ByVal stoptime As Long)
    Dim i As Integer
    Dim TO
    Static lStartArray(1) As Long
    Static lStopArray(1) As Long

    collectionPending = True
    Form1.MousePointer = 13 'ccArrowHourglass
    With FecStatsCollected
        .AccessRate = 3000
        .CloseLink -1
        .Grouped = True
        .AccessMode = "READ"
        .deviceContext = gMachine
        .DeviceGroup = gStatsServer
        .deviceProperty = gMonitoredHists(StatsList.ListIndex) + ".HIST"
        For i = 0 To gNumMonitoredFecs - 1
            .DeviceName = gMonitoredFecs(i)
            startcnts(i).lval = -1: stopcnts(i).lval = -2
            lStartArray(0) = starttime
            lStartArray(1) = starttime
            .AttachLink startcnts(i).lval, 2, "Long/USE_ON_ERROR", lStartArray
            lStopArray(0) = stoptime - 900
            lStopArray(1) = stoptime
            .AttachLink stopcnts(i).lval, 2, "Long/USE_ON_ERROR", lStopArray
        Next
        DoEvents
    End With
    TO = timer
    Do
        Sleep (10)
        DoEvents
    Loop While collectionPending And (timer - TO) < 2
```

Grouping Links (Java API)

```
public static void main (String[] args)
{
    int id;
    TLinkTestCallback cb = new TLinkTestCallback();
    TDataType dout = new TDataType(curr);
    TLink curLink = new TLink("/HERA/HEEIDC/GEARO", "Strom", dout, null, TAccess.CA_READ);
    id = curLink.attach(TMode.CM_POLL | TMode.CM_GROUPED, cb, 1000);
    assert curLink.isGrouped();
    if (id < 0) System.out.println("attach error : " + curLink.getLinkStatus());
    dout = new TDataType(tau);
    TLink tauLink = new TLink("/HERA/HEEIDC/GEARO", "Tau", dout, null, TAccess.CA_READ);
    id = tauLink.attach(TMode.CM_POLL | TMode.CM_GROUPED, cb, 1000);
    if (id < 0) System.out.println("attach error : " + tauLink.getLinkStatus());
    dout = new TDataType(sin);
    TLink sinLink = new TLink("/TEST/SINE/#0", "SINE", dout, null, TAccess.CA_READ);
    id = sinLink.attach(TMode.CM_POLL | TMode.CM_GROUPED, cb, 1000);
    if (id < 0) System.out.println("attach error : " + sinLink.getLinkStatus());
}
```

Grouping Links (Java API)

```
public class TLinkTestCallback implements TLinkCallback
{
    public void callback(TLink link)
    {
        TDataType dout = null;
        System.out.println("callback at " + System.currentTimeMillis() + "\n");
        if (link.isGrouped())
        {
            TLink[] grplst = link.getGroup().getMembers();
            System.out.println("link is grouped (" + grplst.length + " members)");
            for (int i=0; i<grplst.length; i++)
            {
                System.out.println("link : " + grplst[i].linkId + " " + grplst[i].getLinkStatus());
                if ((dout=grplst[i].getOutputDataObject()) != null)
                    System.out.println(dout.toString());
            }
        }
        else
        {
            System.out.println("link : " + link.linkId + " " + link.getLinkStatus());
            if ((dout=link.getOutputDataObject()) != null && dout.getArrayLength() < 10)
                System.out.println(dout.toString());
        }
    }
}
```

Grouping Links (ACOP Transport Bean API)

- `.setGrouped(true)` as in ActiveX
 - All links bound to 'this' transport bean are then grouped !
 - Single 'Receive' event when all links have notified !
- `tbl = getAcopLinkTable()`

```
Vector tbl = a.getAcopTransport().getAcopLinkTable();
AcopTransportRequest atr;
for (int i=0; i<tbl.size(); i++)
{
    atr = (AcopTransportRequest)tbl.get(i);
    System.out.println("link " + atr.getLinkIdIdentifier() + " : " + atr.getStatusCode());
}
```


[Grouping Links in LabView]

- Not a critical display issue
 - due to the 'buffered' asynchronous listener API !
- Use the `IvTineWaitForData.Vi` for one member of the group ?
 - Do we need a `IvTineWaitForGroup.Vi` ?
 - => also need a way to set the members of a group

[Next Time :

- How/When to call the “Scheduler” !

