TINE Release 4.0 News

(Feb 11, 2011: That was the month that was !)

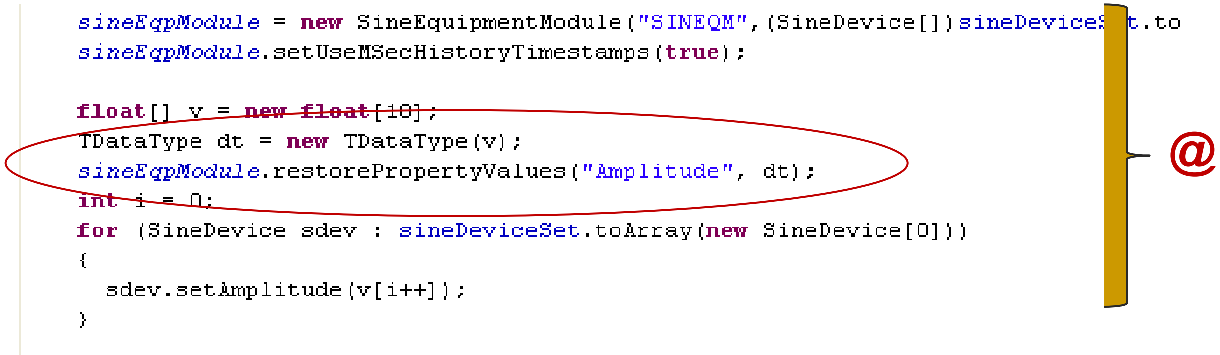
“What a long, strange trip it’s been ….”

Release 4.2.0

- Both C Lib and Java Lib at 4.2.0 !
  - ‘Save-and-Restore’ Features:
    - New routines
    - Automatic save/restore of so-registered properties
  - Callback group synchronization
  - ‘Hide’ Alarms feature

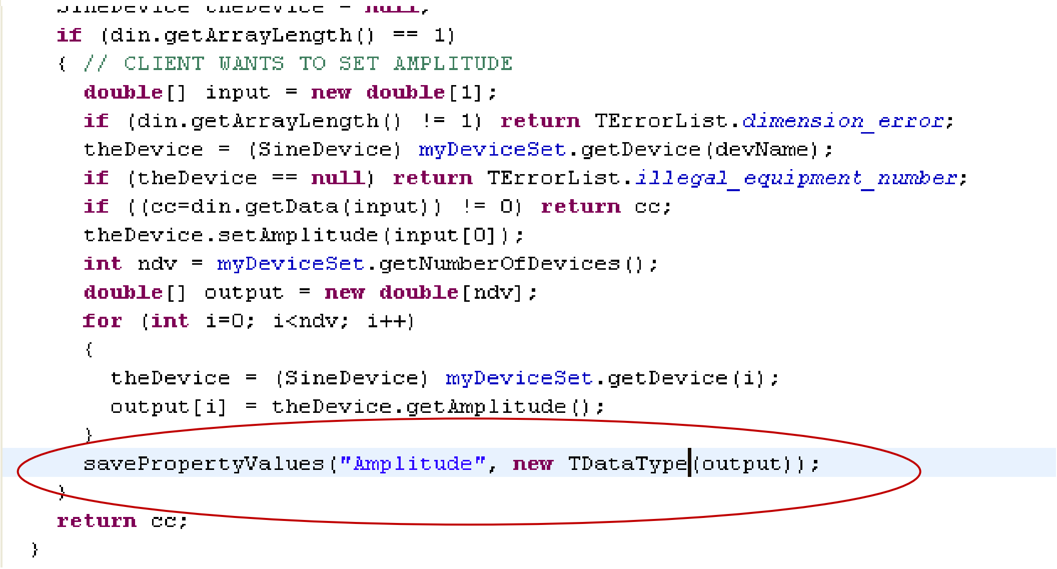
Release 4.2.0

Java: Save and Restore Routines



```
  sineEqpModule = new SineEquipmentModule("SINEQM",(SineDevice[])sineDeviceSet.to
  sineEqpModule.setUseMSecHistoryTimestamps(true);

  float[] v = new float[10];
  TDataType dt = new TDataType(v);
  sineEqpModule.restorePropertyValues("Amplitude", dt);
  int i = 0;
  for (SineDevice sdev : sineDeviceSet.toArray(new SineDevice[0]))
  {
    sdev.setAmplitude(v[i++]);
  }
```

@ Initialization



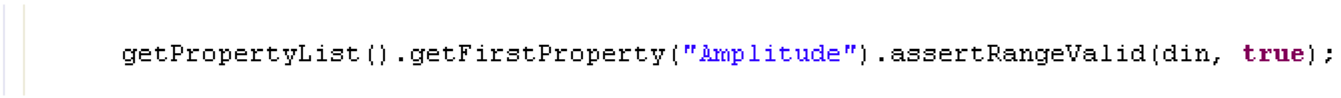
```
  SineDevice theDevice = null;
  if (din.getArrayLength() == 1)
  { // CLIENT WANTS TO SET AMPLITUDE
    double[] input = new double[1];
    if (din.getArrayLength() != 1) return TErrorList.dimension_error;
    theDevice = (SineDevice) myDeviceSet.getDevice(devName);
    if (theDevice == null) return TErrorList.illegal_equipment_number;
    if ((cc=din.getData(input)) != 0) return cc;
    theDevice.setAmplitude(input[0]);
    int ndv = myDeviceSet.getNumberOfDevices();
    double[] output = new double[ndv];
    for (int i=0; i<ndv; i++)
    {
      theDevice = (SineDevice) myDeviceSet.getDevice(i);
      output[i] = theDevice.getAmplitude();
    }
    savePropertyValues("Amplitude", new TDataType(output));
  }
  return cc;
}
```

@ property handler

Release 4.2.0

Java: Range Checking Routines

- Can also use assertRangeValid() in java !

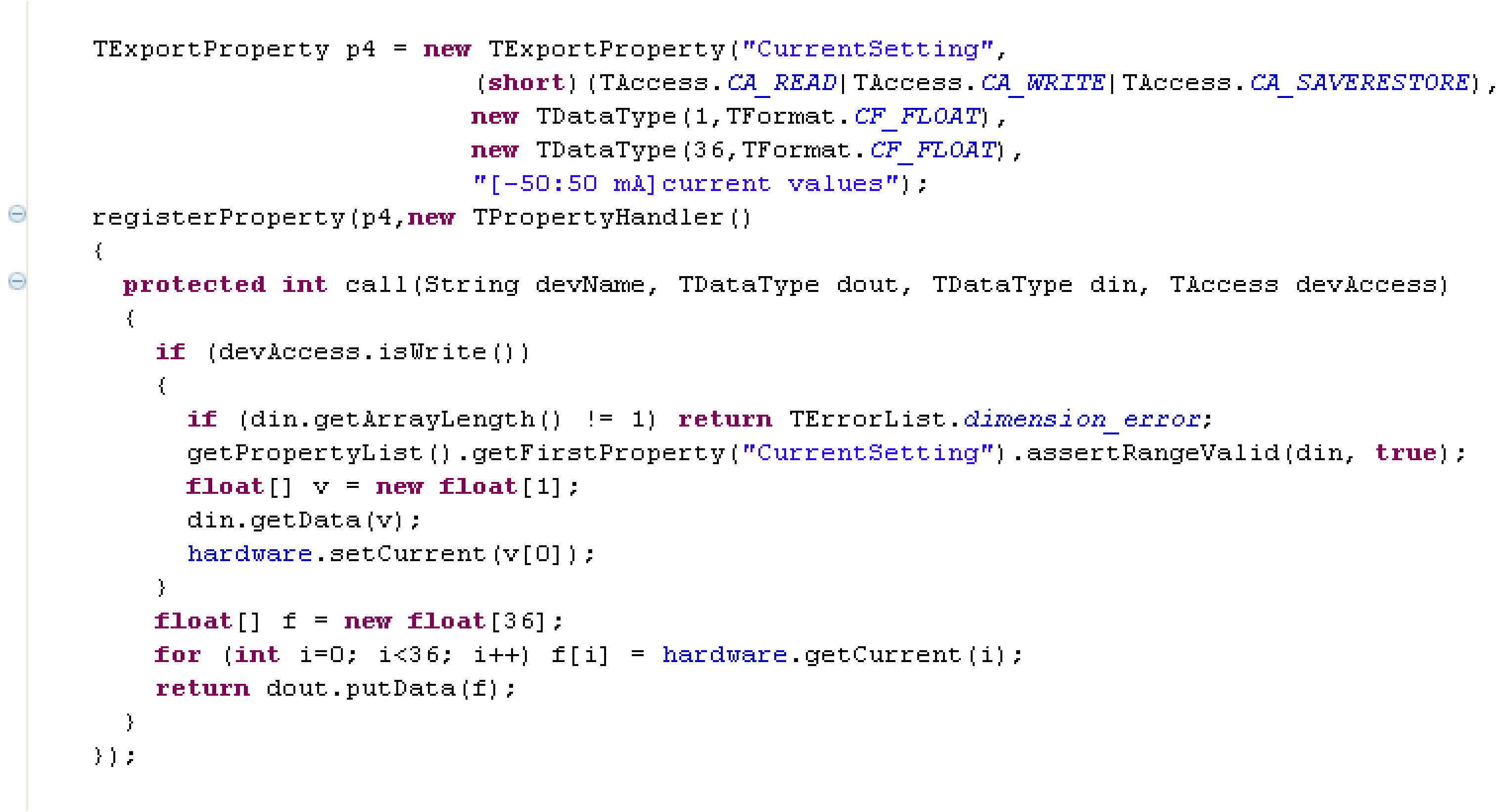


```
getPropertyList().getFirstProperty("Amplitude").assertRangeValid(din, true);
```

Release 4.2.0

- Automatic ‘Save and Restore’

Java: Property registered via API

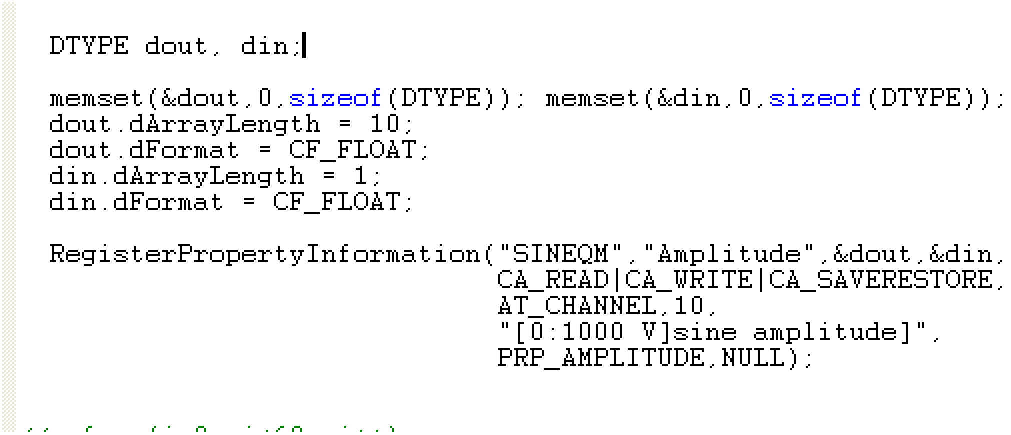


```
TExportProperty p4 = new TExportProperty("CurrentSetting",
                                         (short)(TAccess.CA_READ|TAccess.CA_WRITE|TAccess.CA_SAVERESTORE),
                                         new TDataType(1,TFormat.CF_FLOAT),
                                         new TDataType(36,TFormat.CF_FLOAT),
                                         "[-50:50 mA]current values");
registerProperty(p4,new TPropertyHandler()
{
  protected int call(String devName, TDataType dout, TDataType din, TAccess devAccess)
  {
    if (devAccess.isWrite())
    {
      if (din.getArrayLength() != 1) return TErrorList.dimension_error;
      getPropertyList().getFirstProperty("CurrentSetting").assertRangeValid(din, true);
      float[] v = new float[1];
      din.getData(v);
      hardware.setCurrent(v[0]);
    }
    float[] f = new float[36];
    for (int i=0; i<36; i++) f[i] = hardware.getCurrent(i);
    return dout.putData(f);
  }
});
```

Release 4.2.0

- Automatic ‘Save and Restore’

C: Property registered via API



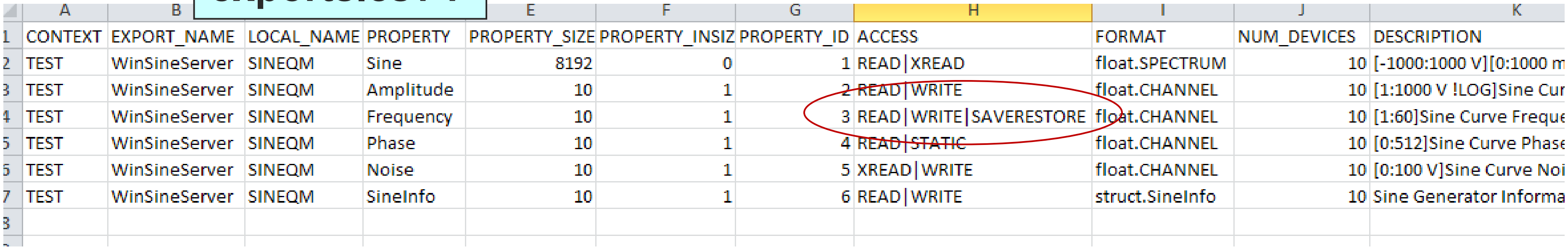
```
DTYPE dout, din;

memset(&dout,0,sizeof(DTYPE)); memset(&din,0,sizeof(DTYPE));
dout.dArrayLength = 10;
dout.dFormat = CF_FLOAT;
din.dArrayLength = 1;
din.dFormat = CF_FLOAT;

RegisterPropertyInformation("SINEQM","Amplitude",&dout,&din,
                            CA_READ|CA_WRITE|CA_SAVERESTORE,
                            AT_CHANNEL,10,
                            "[0:1000 V]sine amplitude]",
                            PRP_AMPLITUDE,NULL);
```

Release 4.2.0

- Automatic ‘Save and Restore’

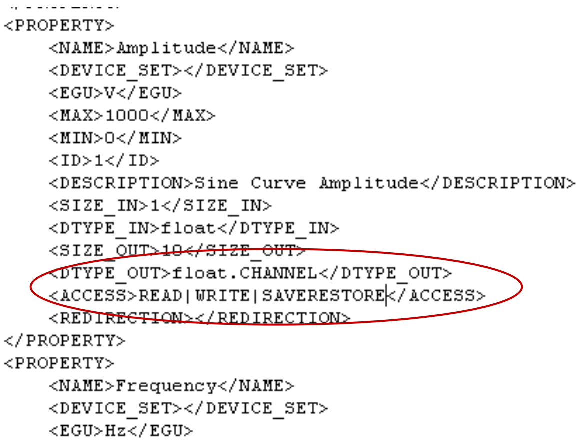
Property registered via configuration file

exports.csv :



	A	B			E	F	G	H	I	J	K
1	CONTEXT	EXPORT_NAME	LOCAL_NAME	PROPERTY	PROPERTY_SIZE	PROPERTY_INSIZ	PROPERTY_ID	ACCESS	FORMAT	NUM_DEVICES	DESCRIPTION
2	TEST	WinSineServer	SINEQM	Sine	8192	0	1	READ|XREAD	float.SPECTRUM	10	[-1000:1000 V][0:1000 m
3	TEST	WinSineServer	SINEQM	Amplitude	10	1	2	READ|WRITE	float.CHANNEL	10	[1:1000 V !LOG]Sine Cur
4	TEST	WinSineServer	SINEQM	Frequency	10	1	3	READ|WRITE|SAVERESTORE	float.CHANNEL	10	[1:60]Sine Curve Freque
5	TEST	WinSineServer	SINEQM	Phase	10	1	4	READ|STATIC	float.CHANNEL	10	[0:512]Sine Curve Phase
6	TEST	WinSineServer	SINEQM	Noise	10	1	5	XREAD|WRITE	float.CHANNEL	10	[0:100 V]Sine Curve Noi
7	TEST	WinSineServer	SINEQM	SineInfo	10	1	6	READ|WRITE	struct.SineInfo	10	Sine Generator Informa
8											

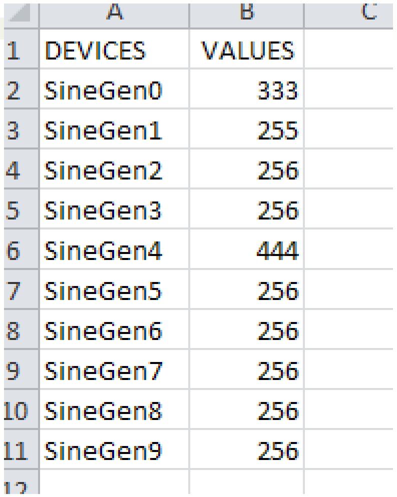
fec.xml :



```
<PROPERTY>
    <NAME>Amplitude</NAME>
    <DEVICE_SET></DEVICE_SET>
    <EGU>V</EGU>
    <MAX>1000</MAX>
    <MIN>0</MIN>
    <ID>1</ID>
    <DESCRIPTION>Sine Curve Amplitude</DESCRIPTION>
    <SIZE_IN>1</SIZE_IN>
    <DTYPE_IN>float</DTYPE_IN>
    <SIZE_OUT>10</SIZE_OUT>
    <DTYPE_OUT>float.CHANNEL</DTYPE_OUT>
    <ACCESS>READ|WRITE|SAVERESTORE</ACCESS>
    <REDIRECTION></REDIRECTION>
</PROPERTY>
<PROPERTY>
    <NAME>Frequency</NAME>
    <DEVICE_SET></DEVICE_SET>
    <EGU>Hz</EGU>
```

Release 4.2.0

- Behind the scenes:
  - Save-and-restore file:
    ‘Amplitude-settings.csv’
    (in <*local name*> sub-directory)



	A	B	C
1	DEVICES	VALUES	
2	SineGen0	333	
3	SineGen1	255	
4	SineGen2	256	
5	SineGen3	256	
6	SineGen4	444	
7	SineGen5	256	
8	SineGen6	256	
9	SineGen7	256	
10	SineGen8	256	
11	SineGen9	256	
12			

  - Values restored when the equipment module is initialized by calling the eqm.
    - one call for each device!
  - Values saved when a WRITE call is successful !

Release 4.2.0

- Restrictions:
  - Allowed '*save-and-restore*' properties :
    - Must support CA_WRITE access
    - Input format = output format
    - Either an '*attribute*'
      - Input size = Output size = 1
    - Or a '*multi-channel array*'
      - Input size = Output size = N
      - Input size = 1, Output size = N

- Otherwise: *Do it yourself!*

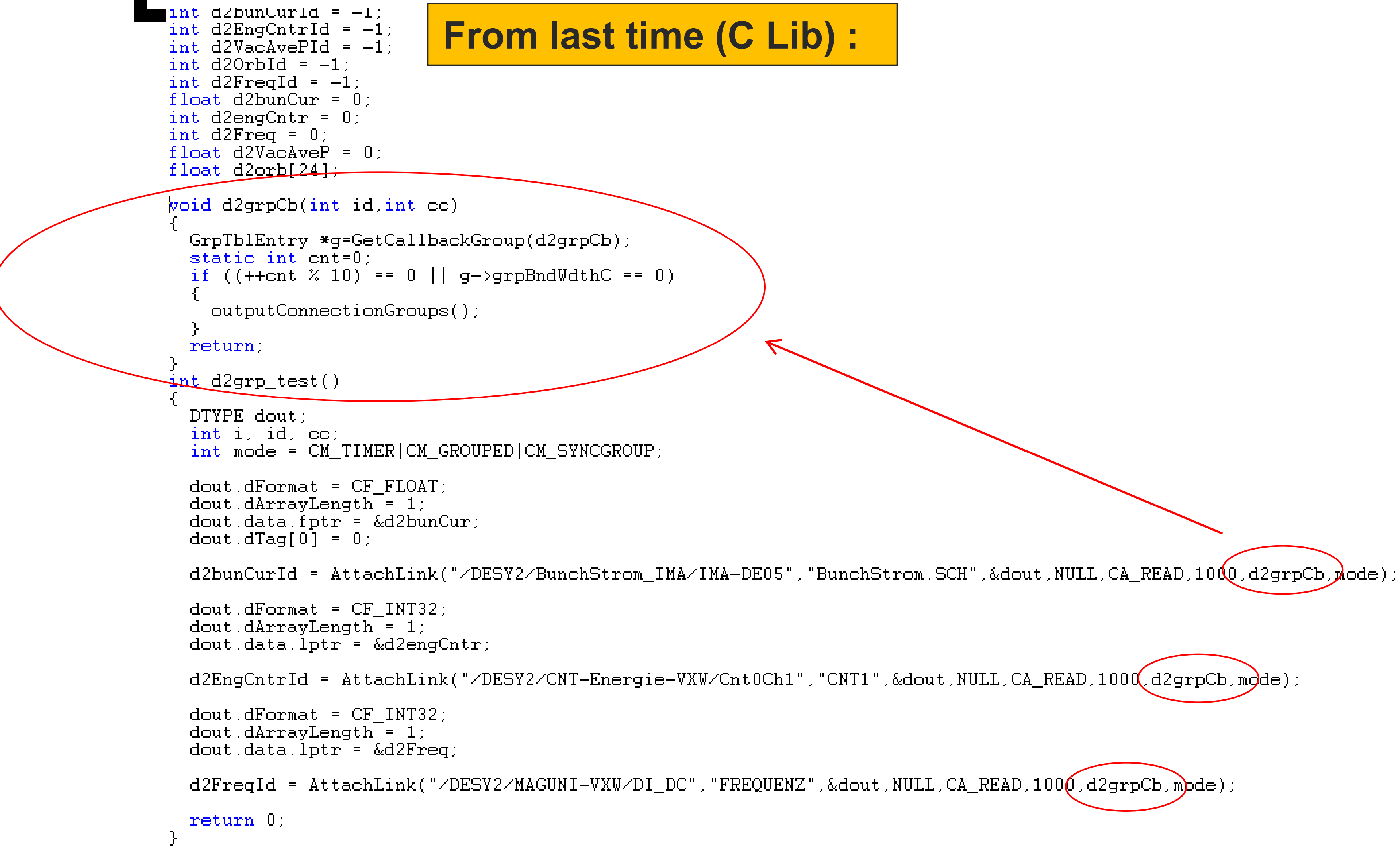
Release 4.2.0

From last time (C Lib) :

- Callback Synchronization
  - Currently:
    - Group notification achieved via CM_GROUPED bit in access mode
    - Wait for all group members prior to callback notification
  - In addition:
    - CM_SYNCGROUP bit
      - Minimize time ‘dispersion’ and cycle count ‘dispersion’
    - Retrieve cycle ‘offset’ and other group info
      - GetCallbackGroup()

Release 4.2.0

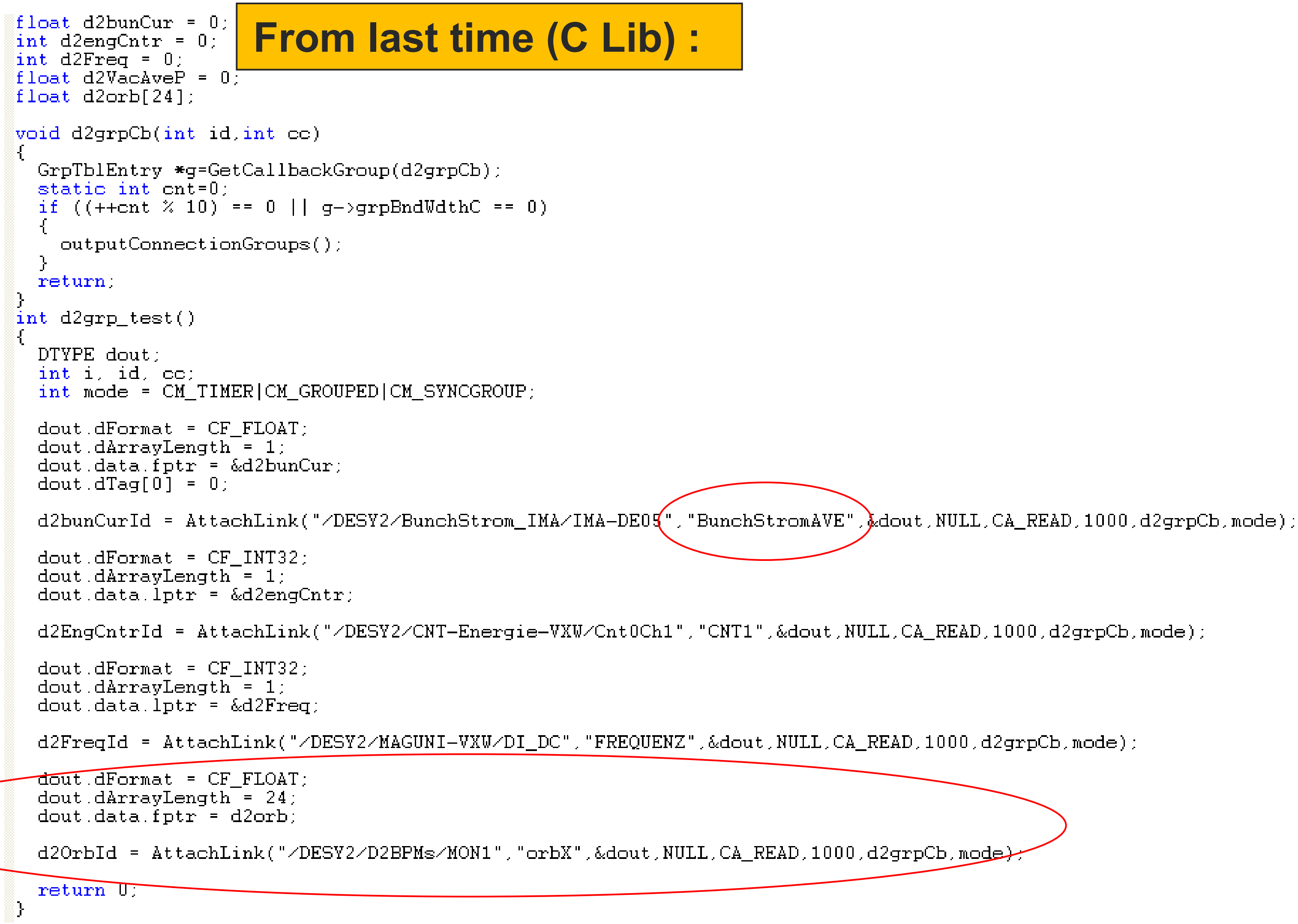
From last time (C Lib) :



```
int d2bunCurId = -1;
int d2EngCntrId = -1;
int d2VacAvePId = -1;
int d2OrbId = -1;
int d2FreqId = -1;
float d2bunCur = 0;
int d2engCntr = 0;
int d2Freq = 0;
float d2VacAveP = 0;
float d2orb[24];

void d2grpCb(int id,int cc)
{
  GrpTblEntry *g=GetCallbackGroup(d2grpCb);
  static int cnt=0;
  if ((++cnt % 10) == 0 || g->grpBndWdthC == 0)
  {
    outputConnectionGroups();
  }
  return;
}
int d2grp_test()
{
  DTYPE dout;
  int i, id, cc;
  int mode = CM_TIMER|CM_GROUPED|CM_SYNCGROUP;

  dout.dFormat = CF_FLOAT;
  dout.dArrayLength = 1;
  dout.data.fptr = &d2bunCur;
  dout.dTag[0] = 0;

  d2bunCurId = AttachLink("/DESY2/BunchStrom_IMA/IMA-DE05","BunchStrom.SCH",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  dout.dFormat = CF_INT32;
  dout.dArrayLength = 1;
  dout.data.lptr = &d2engCntr;

  d2EngCntrId = AttachLink("/DESY2/CNT-Energie-VXW/Cnt0Ch1","CNT1",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  dout.dFormat = CF_INT32;
  dout.dArrayLength = 1;
  dout.data.lptr = &d2Freq;

  d2FreqId = AttachLink("/DESY2/MAGUNI-VXW/DI_DC","FREQUENZ",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  return 0;
}
```

Release 4.2.0

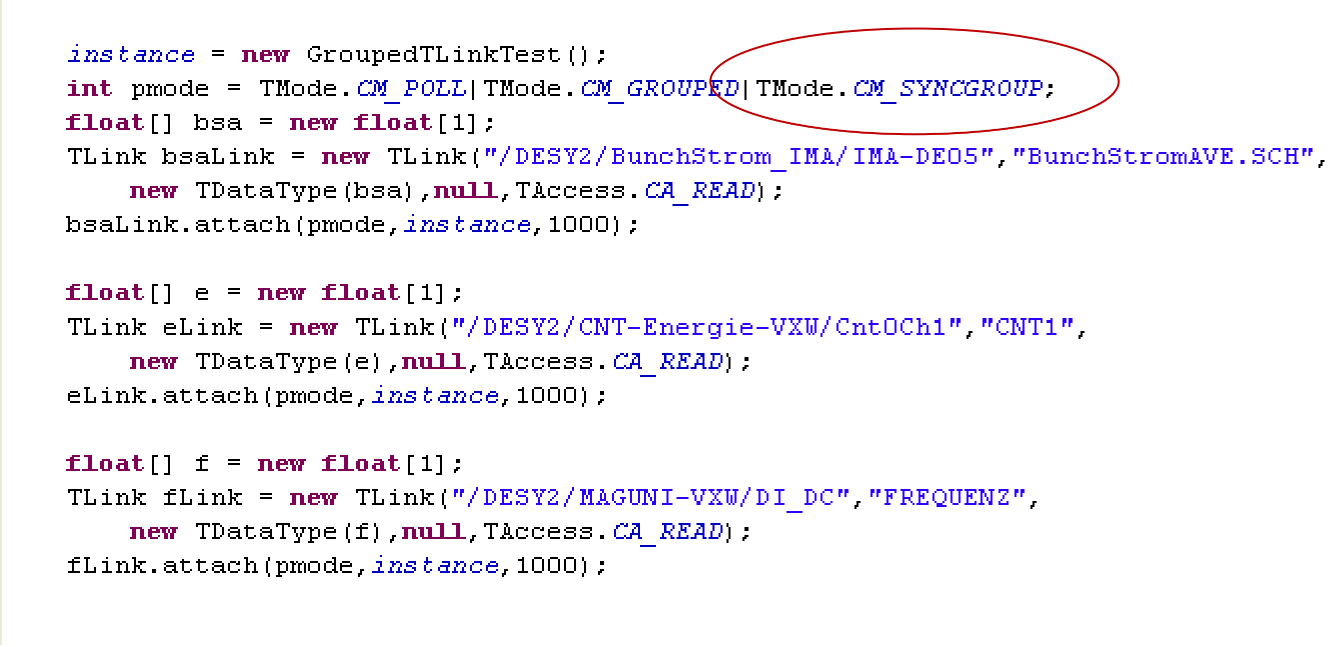
From last time (C Lib) :



```
float d2bunCur = 0;
int d2engCntr = 0;
int d2Freq = 0;
float d2VacAveP = 0;
float d2orb[24];

void d2grpCb(int id,int cc)
{
  GrpTblEntry *g=GetCallbackGroup(d2grpCb);
  static int cnt=0;
  if ((++cnt % 10) == 0 || g->grpBndWdthC == 0)
  {
    outputConnectionGroups();
  }
  return;
}
int d2grp_test()
{
  DTYPE dout;
  int i, id, cc;
  int mode = CM_TIMER|CM_GROUPED|CM_SYNCGROUP;

  dout.dFormat = CF_FLOAT;
  dout.dArrayLength = 1;
  dout.data.fptr = &d2bunCur;
  dout.dTag[0] = 0;

  d2bunCurId = AttachLink("/DESY2/BunchStrom_IMA/IMA-DE05","BunchStromAVE",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  dout.dFormat = CF_INT32;
  dout.dArrayLength = 1;
  dout.data.lptr = &d2engCntr;

  d2EngCntrId = AttachLink("/DESY2/CNT-Energie-VXW/Cnt0Ch1","CNT1",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  dout.dFormat = CF_INT32;
  dout.dArrayLength = 1;
  dout.data.lptr = &d2Freq;

  d2FreqId = AttachLink("/DESY2/MAGUNI-VXW/DI_DC","FREQUENZ",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  dout.dFormat = CF_FLOAT;
  dout.dArrayLength = 24;
  dout.data.fptr = d2orb;

  d2OrbId = AttachLink("/DESY2/D2BPMs/MON1","orbX",&dout,NULL,CA_READ,1000,d2grpCb,mode);

  return 0;
}
```

Release 4.2.0 (callback synchronization)

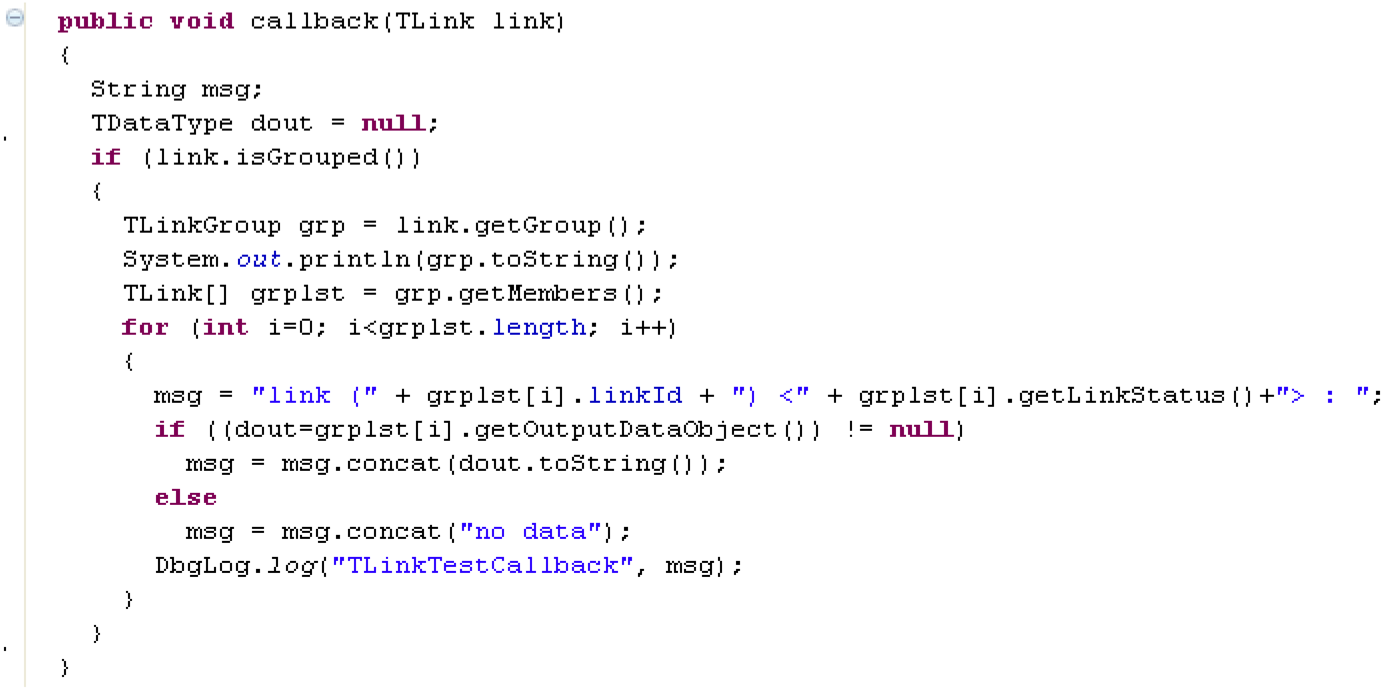
- Java Example:



```
instance = new GroupedTLinkTest();
int pmode = TMode.CM_POLL|TMode.CM_GROUPED|TMode.CM_SYNCGROUP;
float[] bsa = new float[1];
TLink bsaLink = new TLink("/DESY2/BunchStrom_IMA/IMA-DE05","BunchStromAVE.SCH",
    new TDataType(bsa),null,TAccess.CA_READ);
bsaLink.attach(pmode,instance,1000);

float[] e = new float[1];
TLink eLink = new TLink("/DESY2/CNT-Energie-VXW/Cnt0Ch1","CNT1",
    new TDataType(e),null,TAccess.CA_READ);
eLink.attach(pmode,instance,1000);

float[] f = new float[1];
TLink fLink = new TLink("/DESY2/MAGUNI-VXW/DI_DC","FREQUENZ",
    new TDataType(f),null,TAccess.CA_READ);
fLink.attach(pmode,instance,1000);
```

Release 4.2.0 (callback synchronization)

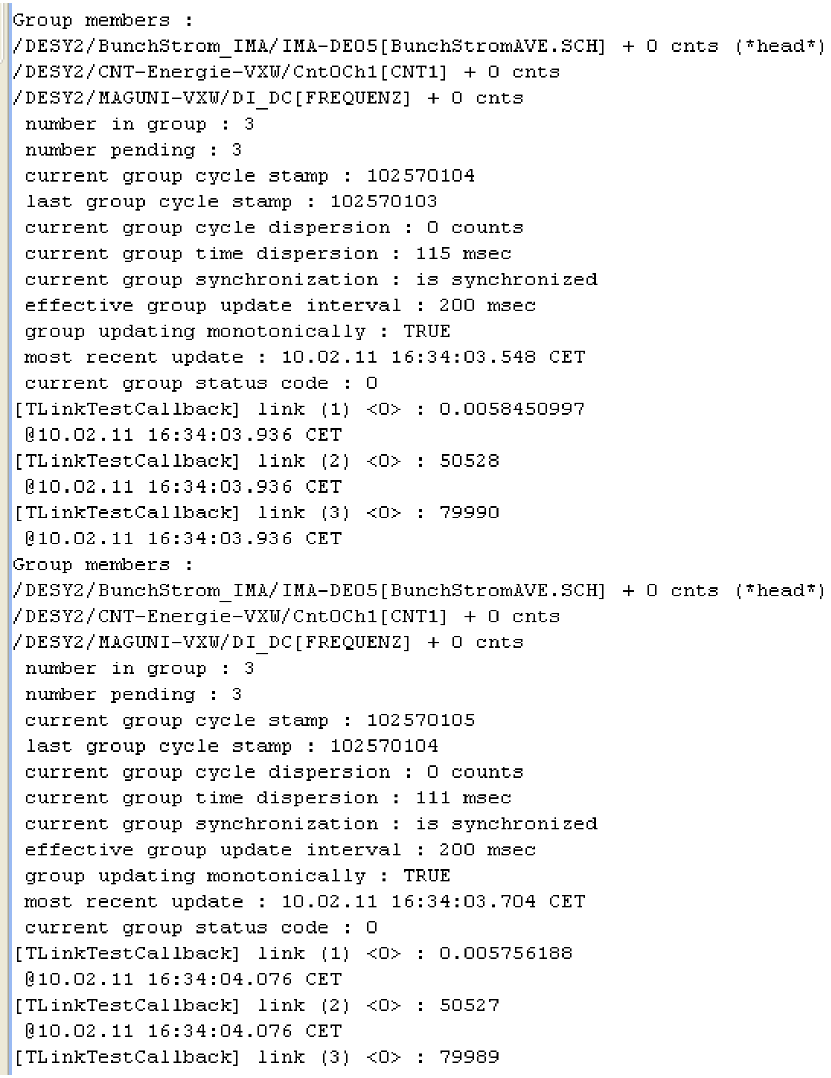
- Java Callback Example:



```java
public void callback(TLink link)
{
  String msg;
  TDataType dout = null;
  if (link.isGrouped())
  {
    TLinkGroup grp = link.getGroup();
    System.out.println(grp.toString());
    TLink[] grplst = grp.getMembers();
    for (int i=0; i<grplst.length; i++)
    {
      msg = "link (" + grplst[i].linkId + ") <" + grplst[i].getLinkStatus()+"> : ";
      if ((dout=grplst[i].getOutputDataObject()) != null)
        msg = msg.concat(dout.toString());
      else
        msg = msg.concat("no data");
      DbgLog.log("TLinkTestCallback", msg);
    }
  }
}
```

Release 4.2.0 (callback synchronization)

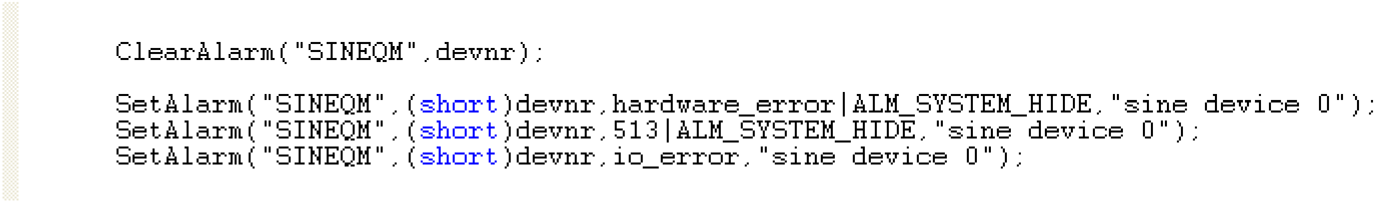
- Output:



```
Group members :
/DESY2/BunchStrom_IMA/IMA-DE05[BunchStromAVE.SCH] + 0 cnts (*head*)
/DESY2/CNT-Energie-VXW/Cnt0Ch1[CNT1] + 0 cnts
/DESY2/MAGUNI-VXW/DI_DC[FREQUENZ] + 0 cnts
 number in group : 3
 number pending : 3
 current group cycle stamp : 102570104
 last group cycle stamp : 102570103
 current group cycle dispersion : 0 counts
 current group time dispersion : 115 msec
 current group synchronization : is synchronized
 effective group update interval : 200 msec
 group updating monotonically : TRUE
 most recent update : 10.02.11 16:34:03.548 CET
 current group status code : 0
[TLinkTestCallback] link (1) <0> : 0.0058450997
 @10.02.11 16:34:03.936 CET
[TLinkTestCallback] link (2) <0> : 50528
 @10.02.11 16:34:03.936 CET
[TLinkTestCallback] link (3) <0> : 79990
 @10.02.11 16:34:03.936 CET
Group members :
/DESY2/BunchStrom_IMA/IMA-DE05[BunchStromAVE.SCH] + 0 cnts (*head*)
/DESY2/CNT-Energie-VXW/Cnt0Ch1[CNT1] + 0 cnts
/DESY2/MAGUNI-VXW/DI_DC[FREQUENZ] + 0 cnts
 number in group : 3
 number pending : 3
 current group cycle stamp : 102570105
 last group cycle stamp : 102570104
 current group cycle dispersion : 0 counts
 current group time dispersion : 111 msec
 current group synchronization : is synchronized
 effective group update interval : 200 msec
 group updating monotonically : TRUE
 most recent update : 10.02.11 16:34:03.704 CET
 current group status code : 0
[TLinkTestCallback] link (1) <0> : 0.005756188
 @10.02.11 16:34:04.076 CET
[TLinkTestCallback] link (2) <0> : 50527
 @10.02.11 16:34:04.076 CET
[TLinkTestCallback] link (3) <0> : 79989
```

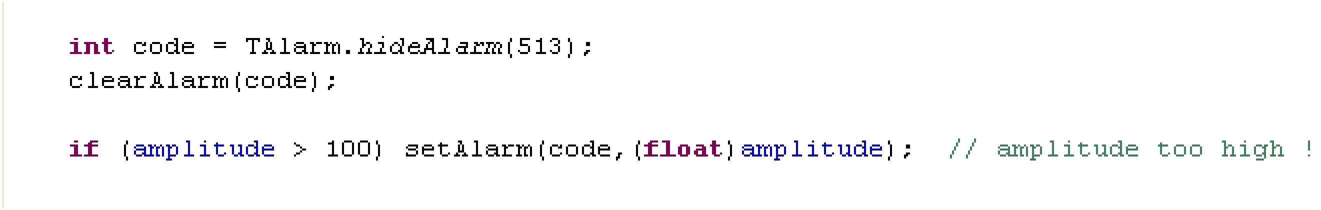
Release 4.2.0

- Hiding Alarms
  - C API example:



```
ClearAlarm("SINEQM",devnr);

SetAlarm("SINEQM",(short)devnr,hardware_error|ALM_SYSTEM_HIDE,"sine device 0");
SetAlarm("SINEQM",(short)devnr,513|ALM_SYSTEM_HIDE,"sine device 0");
SetAlarm("SINEQM",(short)devnr,io_error,"sine device 0");
```

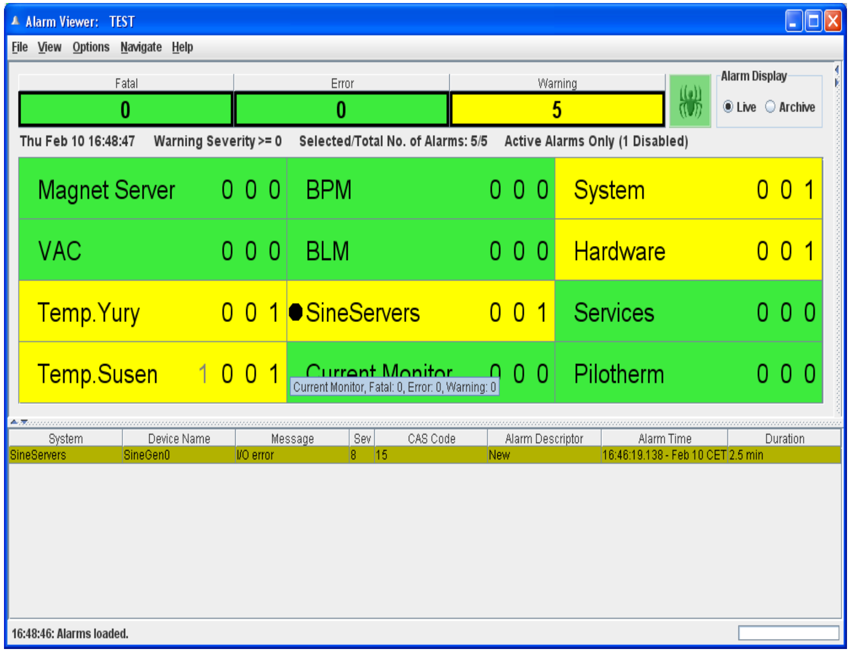
  - Java example:



```
int code = TAlarm.hideAlarm(513);
clearAlarm(code);

if (amplitude > 100) setAlarm(code,(float)amplitude);  // amplitude too high !
```

Release 4.2.0

- Showing hidden alarms:



Fatal	Error	Warning
0	0	5

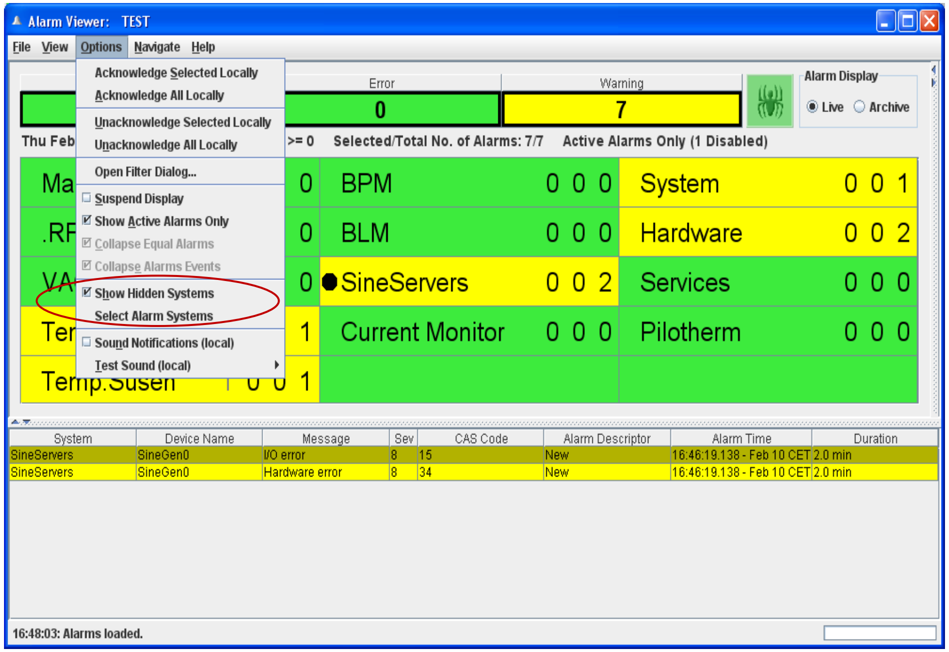
Thu Feb 10 16:48:47 Warning Severity >= 0 Selected/Total No. of Alarms: 5/5 Active Alarms Only (1 Disabled)

Magnet Server 0 0 0	BPM 0 0 0	System 0 0 1
VAC 0 0 0	BLM 0 0 0	Hardware 0 0 1
Temp.Yury 0 0 1	● SineServers 0 0 1	Services 0 0 0
Temp.Susen 1 0 0 1	Current Monitor 0 0 0	Pilotherm 0 0 0

Current Monitor, Fatal: 0, Error: 0, Warning: 0

System	Device Name	Message	Sev	CAS Code	Alarm Descriptor	Alarm Time	Duration
SineServers	SineGen0	I/O error	8	15	New	16:46:19.138 - Feb 10 CET	2.5 min

16:48:46: Alarms loaded.



Error	Warning
0	7

Thu Feb >= 0 Selected/Total No. of Alarms: 7/7 Active Alarms Only (1 Disabled)

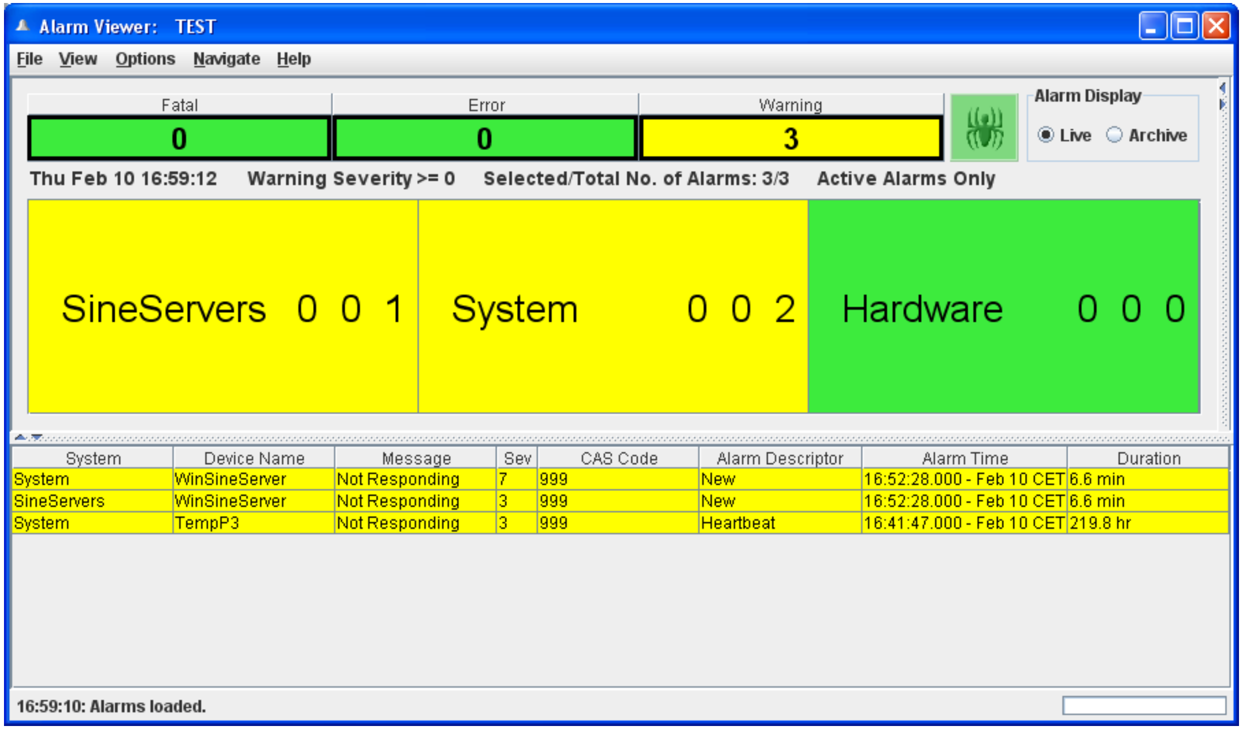
Ma 0	BPM 0 0 0	System 0 0 1
.RF 0	BLM 0 0 0	Hardware 0 0 2
VA 0	● SineServers 0 0 2	Services 0 0 0
Ter 1	Current Monitor 0 0 0	Pilotherm 0 0 0
Temp.Susen 1 0 0 1		

System	Device Name	Message	Sev	CAS Code	Alarm Descriptor	Alarm Time	Duration
SineServers	SineGen0	I/O error	8	15	New	16:46:19.138 - Feb 10 CET	2.0 min
SineServers	SineGen0	Hardware error	8	34	New	16:46:19.138 - Feb 10 CET	2.0 min

16:48:03: Alarms loaded.

Release 4.2.0

- Selecting subsystems:



Fatal	Error	Warning
0	0	3

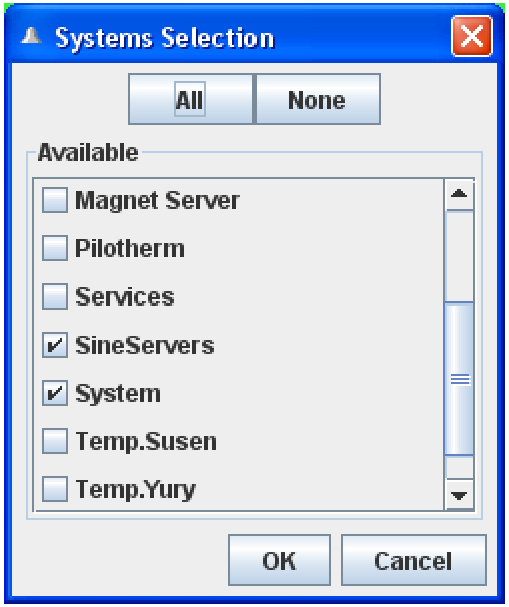
Alarm Display: ● Live ○ Archive

Thu Feb 10 16:59:12 Warning Severity >= 0 Selected/Total No. of Alarms: 3/3 Active Alarms Only

SineServers	0	0	1
System	0	0	2
Hardware	0	0	0

System	Device Name	Message	Sev	CAS Code	Alarm Descriptor	Alarm Time	Duration
System	WinSineServer	Not Responding	7	999	New	16:52:28.000 - Feb 10 CET	6.6 min
SineServers	WinSineServer	Not Responding	3	999	New	16:52:28.000 - Feb 10 CET	6.6 min
System	TempP3	Not Responding	3	999	Heartbeat	16:41:47.000 - Feb 10 CET	219.8 hr

16:59:10: Alarms loaded.



Available

- □ Magnet Server
- □ Pilotherm
- □ Services
- ☑ SineServers
- ☑ System
- □ Temp.Susen
- □ Temp.Yury

Release 4.2.0

- Local History Subsystem News
  - Much progress implementing the ‘standard’ minimally-fragmented history file set.
    - Important for NTFS
  - Save discussion and full report for next meeting!