



# LabView News

# [ Bounds Checking in the Buffered Server ]

- WRITE command automatically check incoming 'set' values against the range registered with the property.
- Return 'out\_of\_range' if not in range.

# [ LabView Wish List ]

- New Data Format NAME128I ?
  - Moratorium on new data system types
    - Remove out-dated or useless data types (e.g. DBLINT)
    - Establish common set with DOOCS ?
- Solution:
  - Allow user “tagged” structures in the buffered-server/listener interface!

# LabView Wish List

## ■ Practical Example (mhflib.dll):

```
#include "tine.h"
#include "listener.h"

#define STRING_SIZE 256
typedef struct
{
    char string[STRING_SIZE];
    int value;
} StringInt;

void registerStructs(void)
{
    short cc = 0;
    char errstr[256], tag[32];
    /* register the fields in a structure like this */
    strncpy(tag, "StringInt", TAG_NAME_SIZE);
    if ((cc=addFieldToStruct(tag, OFFSETIN(StringInt, string), STRING_SIZE, CF_TEXT, "string")) != 0) ccerr(cc);
    if ((cc=addFieldToStruct(tag, OFFSETIN(StringInt, value), 1, CF_LONG, "value")) != 0) ccerr(cc);
    /* close the registration when finished */
    if ((cc=sealTaggedStruct(tag, sizeof(StringInt), 100)) != 0) ccerr(cc);
    /* any others ? ...*/

err:
    if (cc != 0)
    {
        feclog("registerStructs : tag %s -> %s", tag, GetLastError(cc, errstr));
    }
}
```

Make your own NAME256I Data Type !

# LabView Wish List

## Client-Side 'get's and 'put's

```
/* Client-side : in labview VI call with tag = "StringInt" and pArray is a reference to a StringInt "thing" */
__declspec(dllexport) int mhfGetStructArray(char *fullDeviceNameAndProperty, char *tag, BYTE *pArray, int arrayLen, double *dTimeStamp)
{
    DTYPE dout;
    dout.dArrayLength = arrayLen;
    dout.dFormat = CF_STRUCT;
    strncpy(dout.dTag, tag, TAG_NAME_SIZE);
    return alsnCall(fullDeviceNameAndProperty, &dout, NULL, CA_READ, dTimeStamp);
}
__declspec(dllexport) int mhfPutStructArray(char *fullDeviceNameAndProperty, char *tag, BYTE *pArray, int arrayLen, double *dTimeStamp)
{
    DTYPE din;
    din.dArrayLength = arrayLen;
    din.dFormat = CF_STRUCT;
    strncpy(din.dTag, tag, TAG_NAME_SIZE);
    return alsnCall(fullDeviceNameAndProperty, NULL, &din, CA_WRITE, dTimeStamp);
}
/* Server-side : in labview VI call
pushBufferedData(char *prpName, char *devName, BYTE *prpData, long prpSiz, int prpSchedule)
pullBufferedData(char *prpName, char *devName, BYTE *prpData, long prpSiz)
with prpData pointing to a StringInt "thing" and prpName registered with format = struct.StringInt
*/
```

## Server-Side : 'push'es and 'pull's

# [ LabView Wish List ]

---

- LabView 'Events' vs 'Occurrences'
  - LabView 8.5