TINE Release 4.0 News
(July 6, 2012: That was the month that was !)

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

Release 4.3.0

- Embellishments and bug fixes (C Lib)
  - ACL problem with ‘*quick identification*’ fixed.
    - noticed by MDI !
    - if ‘*this*’ client is the same as the ‘*last* client granted WRITE access’ then he is automatically granted access !
      - substantial savings in CPU cycles !
      - **BUT**: need to pay attention to property ACL vs. device ACL vs. server ACL.
  - SetCycleMicroDelay(0)
    - now ‘*resets*’ the micro-delay.

Release 4.3.0

- Embellishments and bug fixes (java)
  - TAccess.CA_CONNECT now influences the Link '*mode*'.
    - If *access* carries this bit, then any .execute() or .attach() method will automatically apply TMode.CM_CONNECT.
    - So why is TAccess.CA_CONNECT there?
    - And what is the difference between TAccess and TMode?

Access vs. Mode

- Access is seen by the equipment module, i.e. property dispatch handler.
  - can set and send :
    - 'CA_READ, CA_WRITE' *from the client side*
  - also: (seen in the dispatch)
    - CA_FIRST, CA_LAST, CA_HIST, CA_ALARM, CA_LOCKED, etc.
  - also: (client-side instructions)
    - CA_MUTABLE, CA_SYNCNOTIFY, etc.
  - also: (server-side property registration)
    - CA_STATIC, CA_NETWORK, etc.

Access vs. Mode

- Mode gives the desired method of data transfer:
  - base:
    - CM_SINGLE, CM_DATACHANGE, CM_TIMER, CM_EVENT
  - modifier bits:
    - CM_CONNECT, CM_NETWORK, CM_GROUPED, etc.
  - EQM dispatch does **NOT** see this !

Access vs. Mode

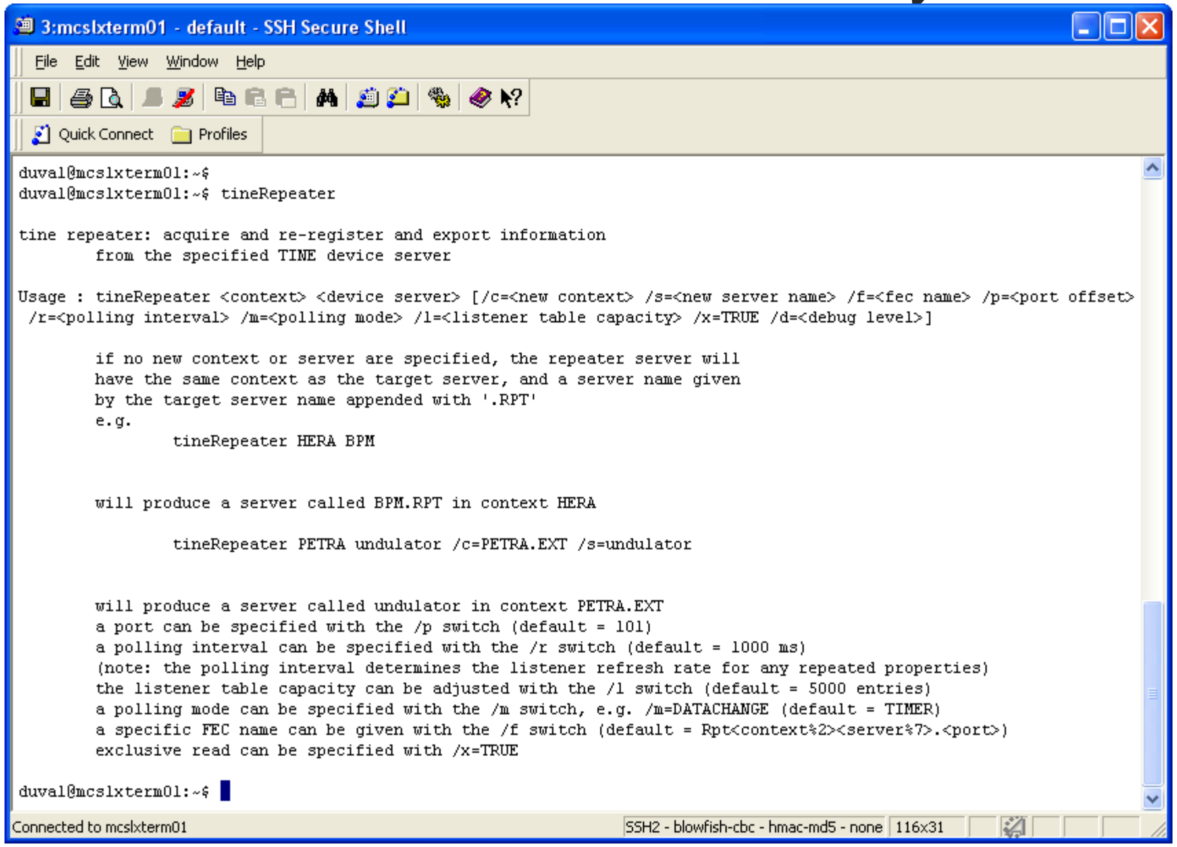
- Some APIs don’t allow a distinction
  - ACOP !
    - That’s why: “READ”, “READ_CONNECT”, etc.
  - ExecLink() from the C API
    - Only carries the access
      - After all : the base Mode is ‘CM_SINGLE’
      - So: allow CA_CONNECT to supply this extra information : use a connected socket for the synchronous call !
  - Until now: CA_CONNECT had no effect in java!

TINE Repeater News

- Primary function:
  - ‘*repeat*’ another server in the control system.

Many repeaters in use !

PETRA3 <-> DMZ
DESY <-> EMBL


3:mcslxterm01 - default - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Connected to mcslxterm01
SSH2 - blowfish-cbc - hmac-md5 - none
116x31

```
duval@mcslxterm01:~$
duval@mcslxterm01:~$ tineRepeater

tine repeater: acquire and re-register and export information
        from the specified TINE device server

Usage : tineRepeater <context> <device server> [/c=<new context> /s=<new server name> /f=<fec name> /p=<port offset>
 /r=<polling interval> /m=<polling mode> /l=<listener table capacity> /x=TRUE /d=<debug level>]

        if no new context or server are specified, the repeater server will
        have the same context as the target server, and a server name given
        by the target server name appended with '.RPT'
        e.g.
                tineRepeater HERA BPM


        will produce a server called BPM.RPT in context HERA

                tineRepeater PETRA undulator /c=PETRA.EXT /s=undulator


        will produce a server called undulator in context PETRA.EXT
        a port can be specified with the /p switch (default = 101)
        a polling interval can be specified with the /r switch (default = 1000 ms)
        (note: the polling interval determines the listener refresh rate for any repeated properties)
        the listener table capacity can be adjusted with the /l switch (default = 5000 entries)
        a polling mode can be specified with the /m switch, e.g. /m=DATACHANGE (default = TIMER)
        a specific FEC name can be given with the /f switch (default = Rpt<context%2><server%7>.<port>)
        exclusive read can be specified with /x=TRUE

duval@mcslxterm01:~$
```

TINE Repeater News

- tineRepeater queries target server for
  - properties and devices
  - structure information
  - dies if no information available !
- Problem: target must be active upon start of tineRepeater !
  - e.g. service day when ‘everything’ is restarted (or not) could lead to ‘dead’ repeaters.
- Now: upon initial success:
  - Keep a local database of all information obtained !
  - If target does not respond *use cached info*!

TINE Repeater News

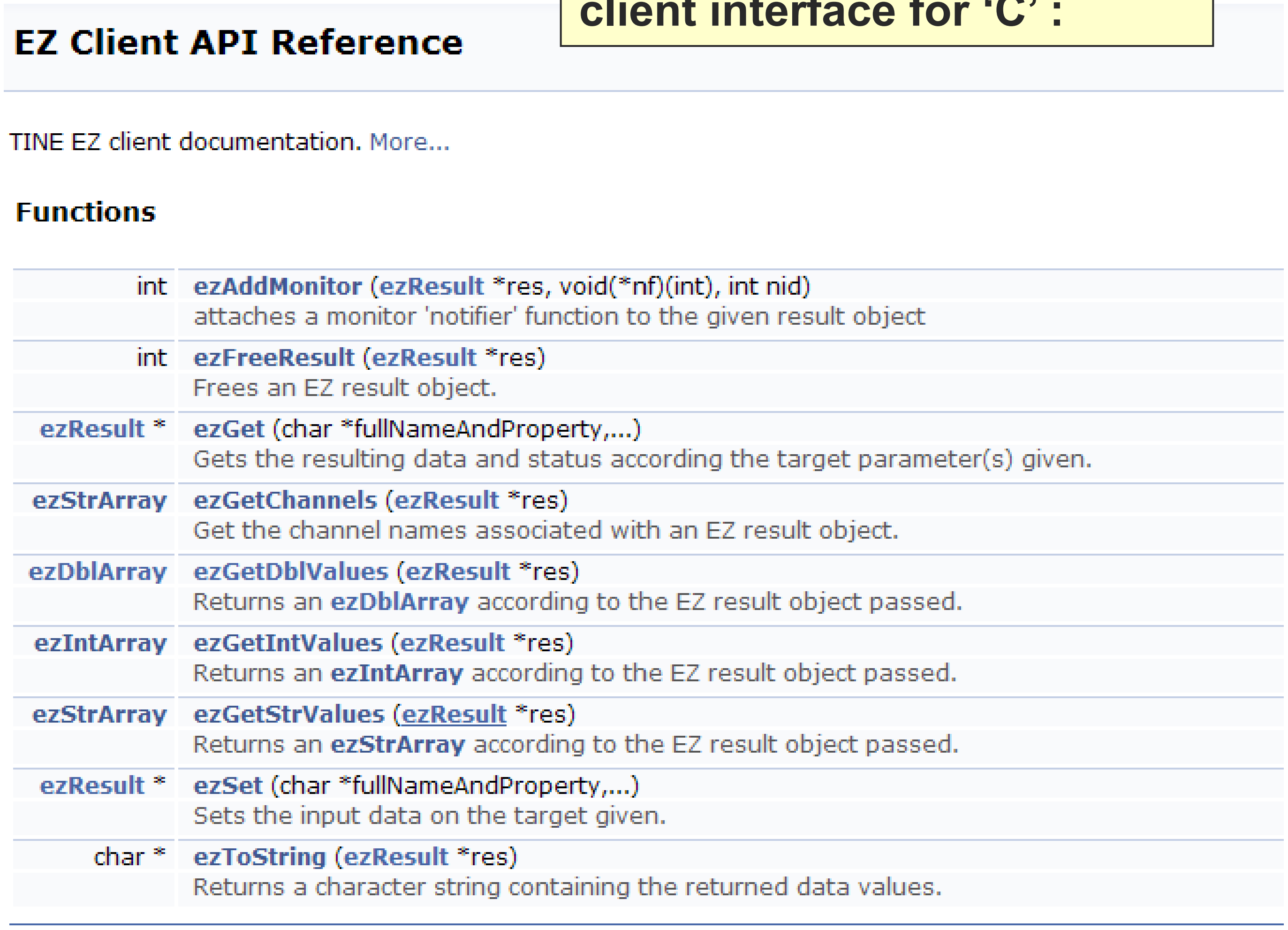
- Resurrect old functionality:
  - tineRepeater LOCALHOST
    - runs as a ‘*client-side*’ repeater service !
      - adds itself to local address cache
      - does **NOT** add itself as a server to the ENS !
  - Console command line tools
    - e.g. ‘tget’
    - ask for result from the local repeater
    - not there? -> then start it in the background !

EZ TINE

- Client Applications: 2 kinds
  - “no-coding” style (simple clients)
    - Use a ‘*panel builder*’
      - jddd, css, coma, etc.
  - code + API (rich clients)
    - TINE API
      - Rich, powerful, but complicated?
    - ACOP
      - Originally designed to be ‘easy’
      - ACOP ActiveX was/is easier than acopbeans?
    - EZ TINE

EZ TINE

Initial attempt at an 'easy' client interface for 'C' :



EZ Client API Reference

TINE EZ client documentation. More...

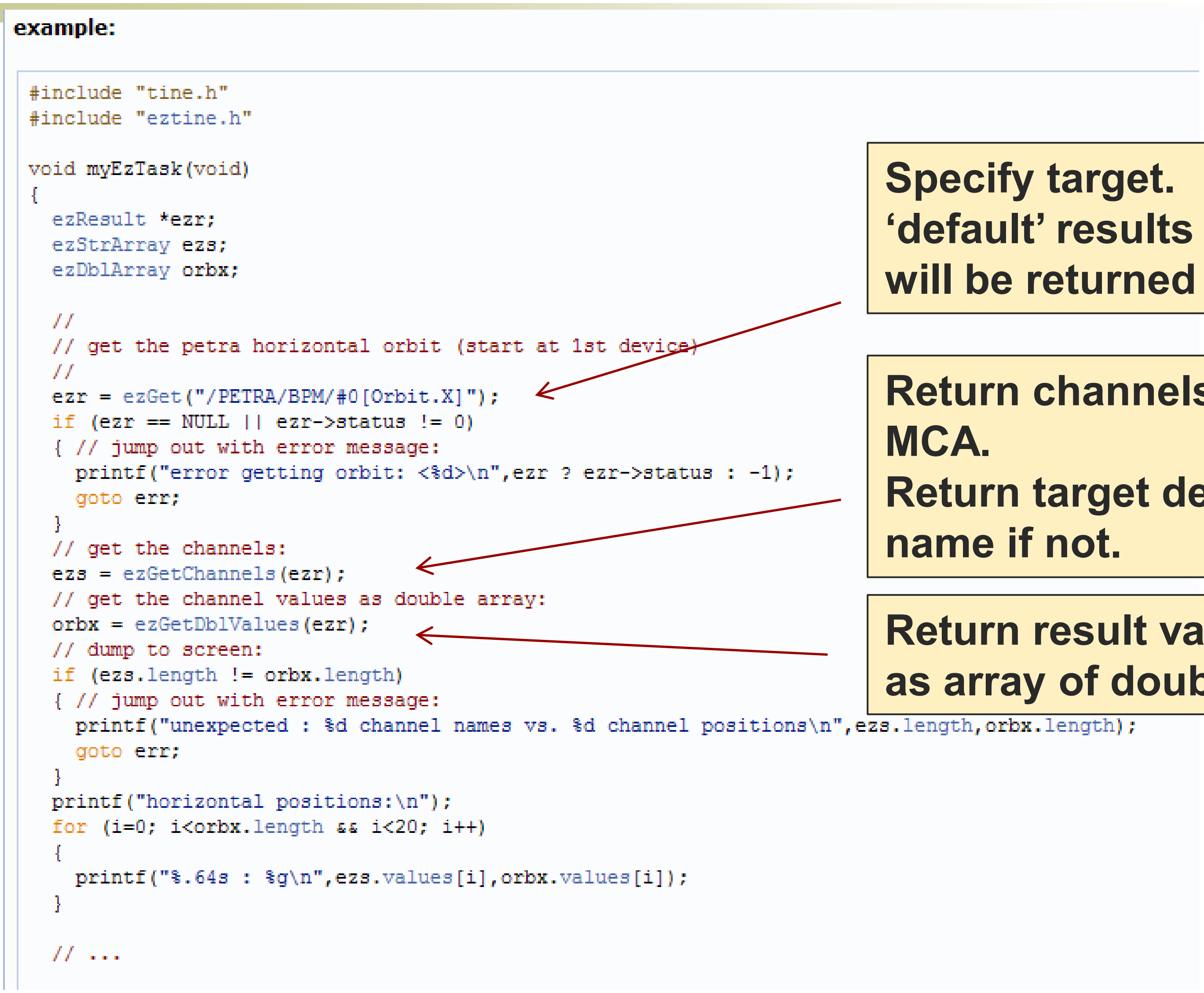
Functions

int	**ezAddMonitor** (**ezResult** *res, void(*nf)(int), int nid) attaches a monitor 'notifier' function to the given result object
int	**ezFreeResult** (**ezResult** *res) Frees an EZ result object.
ezResult *	**ezGet** (char *fullNameAndProperty,...) Gets the resulting data and status according the target parameter(s) given.
ezStrArray	**ezGetChannels** (**ezResult** *res) Get the channel names associated with an EZ result object.
ezDblArray	**ezGetDblValues** (**ezResult** *res) Returns an **ezDblArray** according to the EZ result object passed.
ezIntArray	**ezGetIntValues** (**ezResult** *res) Returns an **ezIntArray** according to the EZ result object passed.
ezStrArray	**ezGetStrValues** (**ezResult** *res) Returns an **ezStrArray** according to the EZ result object passed.
ezResult *	**ezSet** (char *fullNameAndProperty,...) Sets the input data on the target given.
char *	**ezToString** (**ezResult** *res) Returns a character string containing the returned data values.

EZ TINE

- Primarily:
  - ezGet() and ezSet()
    - use asynchronous listeners when possible !
    - use 'printf()' style arguments for input
    - return an ezResult object

EZ TINE



example:

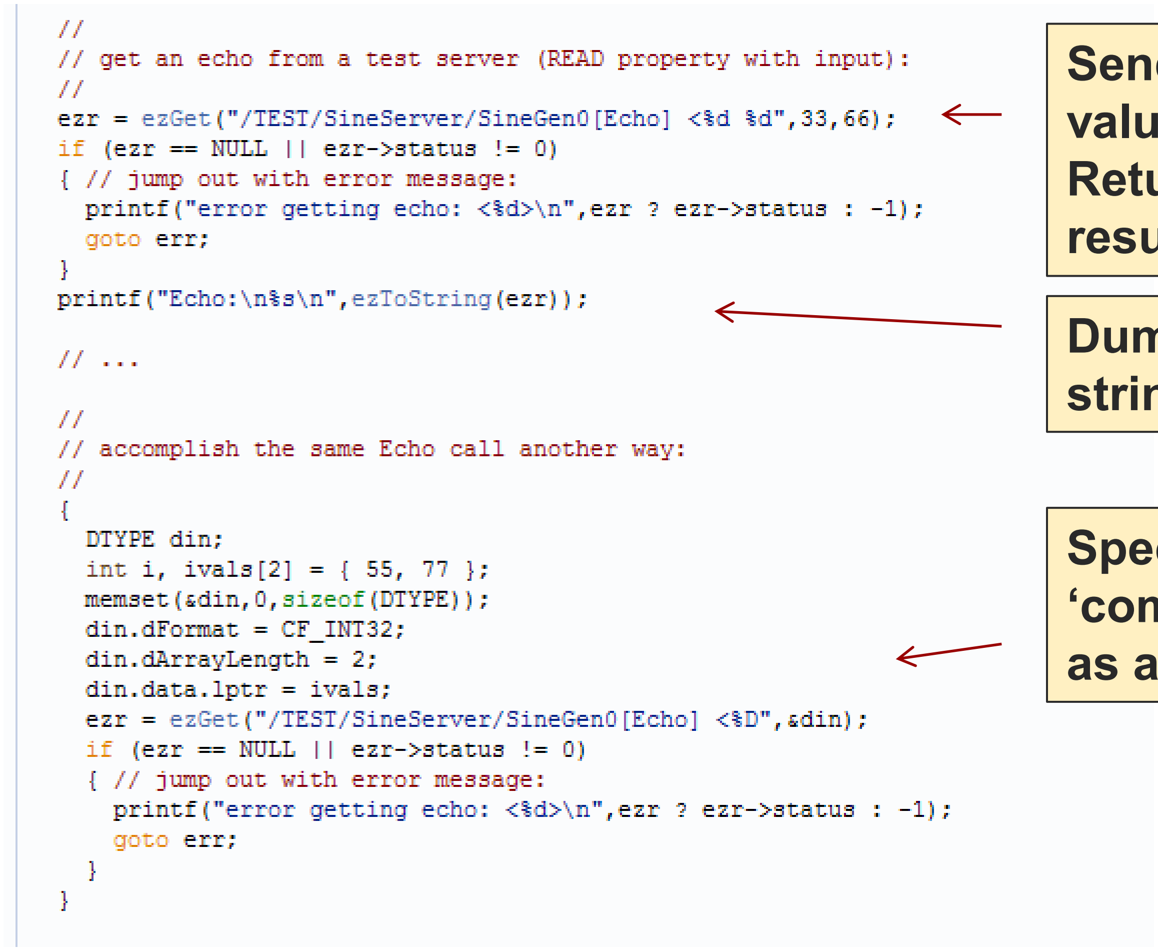
```
#include "tine.h"
#include "eztine.h"

void myEzTask(void)
{
  ezResult *ezr;
  ezStrArray ezs;
  ezDblArray orbx;

  //
  // get the petra horizontal orbit (start at 1st device)
  //
  ezr = ezGet("/PETRA/BPM/#0[Orbit.X]");
  if (ezr == NULL || ezr->status != 0)
  { // jump out with error message:
    printf("error getting orbit: <%d>\n",ezr ? ezr->status : -1);
    goto err;
  }
  // get the channels:
  ezs = ezGetChannels(ezr);
  // get the channel values as double array:
  orbx = ezGetDblValues(ezr);
  // dump to screen:
  if (ezs.length != orbx.length)
  { // jump out with error message:
    printf("unexpected : %d channel names vs. %d channel positions\n",ezs.length,orbx.length);
    goto err;
  }
  printf("horizontal positions:\n");
  for (i=0; i<orbx.length && i<20; i++)
  {
    printf("%.64s : %g\n",ezs.values[i],orbx.values[i]);
  }

  // ...
```

Specify target. 'default' results will be returned !

Return channels if an MCA. Return target device name if not.

Return result values as array of doubles

EZ TINE



```
//
// get an echo from a test server (READ property with input):
//
ezr = ezGet("/TEST/SineServer/SineGen0[Echo] <%d %d",33,66);
if (ezr == NULL || ezr->status != 0)
{ // jump out with error message:
  printf("error getting echo: <%d>\n",ezr ? ezr->status : -1);
  goto err;
}
printf("Echo:\n%s\n",ezToString(ezr));

// ...

//
// accomplish the same Echo call another way:
//
{
  DTYPE din;
  int i, ivals[2] = { 55, 77 };
  memset(&din,0,sizeof(DTYPE));
  din.dFormat = CF_INT32;
  din.dArrayLength = 2;
  din.data.lptr = ivals;
  ezr = ezGet("/TEST/SineServer/SineGen0[Echo] <%D",&din);
  if (ezr == NULL || ezr->status != 0)
  { // jump out with error message:
    printf("error getting echo: <%d>\n",ezr ? ezr->status : -1);
    goto err;
  }
}
```

Send 2 integer values to target. Return default results.

Dump results as string.

Specify ‘complicated’ input as a DTYPE.

EZ TINE

- ezGet() will start a background 'listener'
  - result object is cached and updated with the next ezGet() with the same argument.
  - listener is halted if 5-minute deadtime (no further ezGet()) elapses.

EZ TINE

- Note:
  - Target string can be of the form:
    - /context/server/device[property]

    Or
    - /context/server/device/property
    - i.e. distinguish between ‘which’ and ‘what’ if you prefer (or not)
- EZ TINE for
  - Java ?
  - C++
  - C#, VB.NET