IIF Error Handling

Chris Biddick, LBTO

Florence, Oct 1, 2007

Introduction

- Section 8 of “Instrument Interface for the LBT TCS C++ ICD” (CAN 481s011, Jose Borelli)
- Examples are in CVS
  - LBTO/TCS/IIF/Instrument/Examples/test.cpp
  - LBTO/TCS/IIF/Instrument/Examples/ctest.c
- C++ interface

IIF error levels

- IIF commands return two levels of errors
  - Error on Instrument side
    - Invalid IIF object instantiations
    - Parameter errors
    - Command is not sent to TCS
  - Error on TCS side
    - Invalid handle
    - Subsystem not active
    - Subsystem errors

Instrument side error

- IIF object instantiation can fail
  - Throws IIF_INIT_FAILED if
    - Cannot find CSQ subsystem
    - Instrument name/focal station invalid
    - Internal CSQ command creation failure
  - Sends messages to SysLog and stderr
- Command parameter errors
  - IIF commands return pointer to Result object (except Authorize)
  - Check Result object for error and reason

Result object

- Methods
  - GetresultCode()
  - GetresultString()
  - GetcommandHandle()

Result object (2)

- GetresultCode returns int
  - RESCODE_OK
  - RESCODE_FAIL
- GetresultString returns string
  - RESSTRING_OK
  - RESSTRING_FAIL
  - RESSTRING_UNAUTHORIZED
- GetcommandHandle returns pointer to CSQHandle (from the TCS)

Example part one

```
IIF * iif;
try {
  iif = new IIF(“bentGregorianFront left”, “LUCIFER1”);
} catch (int e) {
  cout << “The IIF failed to initialize: fix problem and retry” << endl;
  exit(-1);
}
…
Result * result;
result = iif->IIF_Command(…);
if(result->GetresultCode() == RESCODE_FAIL) { //failure
  cout << “Command failed: ” << result->GetresultString() << endl;
  …
} else { //OK
  …
}
delete result;
```

TCS side error

- If no instrument side error, populate StatusInfo object
- Wait for command done using Block(), or poll using GetCommandStatus()
- [optional] When done, check GetCommandStatus() for failure

TCS side error (2)

- When TCS commands are done they return a serialized TCS CommandReturn object (except Authorize and Deauthorize)
- Deserialize the CommandReturn object
  - Check for deserialize error
- Check for CommandReturn error
- Get error messages

StatusInfo object

- Keeps pointer to TCS command handle
- Methods
  - Block()
  - GetCommandStatus()
  - GetCommandResult()
  - IsCurrent()
  - GetEstimateTTC()

StatusInfo object (2)

- Block() waits for TCS done
- GetCommandStatus() returns string

STATE_RUNNING	not done
STATE_FAILURE	done
STATE_SUCCESS	done
STATE_CANCELED	done
STATE_WRONGHANDLE	no handle

StatusInfo object (3)

- GetCommandResult() returns string
  - STATE_NORESULT no handle
  - Serialized CommandReturn object
- IsCurrent() returns bool
  - true if handle exists, otherwise false
- GetEstimateTTC() returns time (double) for command to complete
  - Not supported for most TCS commands (get -1.0)

CommandReturn object

- Contains command status and result strings
- Methods
  - deserialize(string)
  - isError()
  - getResultCount()
  - getResultDescription(int)

CommandReturn object (2)

- deserialize(string) returns int
  - Loads CommandReturn object from XML string
  - Returns -1 if error deserializing, otherwise +1
    - On error has one result with error description
- isError() returns true if command error
- getResultCount() returns number (int) of results

CommandReturn object (3)

- getResultDescription(int n) returns n^{th} result string
  - n >= 1
  - If n omitted, default is 1

Example part two

```
StatusInfo * statusInfo = iif->GetCommandStatus(result->GetcommandHandle());
statusInfo->Block(); /* wait for TCS */
cout << statusInfo->GetCommandStatus() << endl; //optional

CommandReturn cmdRet;
if(cmdRet.deserialize(statusInfo->GetCommandResult()) < 0) { //deserialize error
  cout << “Error in deserialize: ” << cmdRet.getResultDescription() << endl;
  …
} else { //deserialize OK
  if(cmdRet.isError()) { //command execution error
    for(int i=1; i<=cmdRet.getResultCount(); i++)
      cout << “Error in execution: ” << cmdRet.getResultDescription(i) << endl;
    …
  } else { //command OK
    …
  }
}
delete statusInfo;
```

CommandReturn errors

- All subsystems should load failure reason into the CommandReturn result
- Examples
  - “Chase failed to make tolerance”
  - “Offset not found”
  - “Subsystem not running”

CommandReturn errors (2)

- There can be multiple failure results
  - Example: MoveFocus failure

left PSF primary mirror adjustMirrorCollimation for 0 seconds in the future failed because PMC subsystem not running

left PSF primary mirror setLocalOffsets failed because adjustMirrorCollimation failed

left PSF SetInstrumentOffsets failed

CommandReturn errors (3)

- SetMultiParameter returns a result for each invalid parameter name
  - Valid parameters are written to the DD
- GetMultiParameter returns a result for each invalid parameter name and does not return values for valid names

Special cases

- Authorize returns bool: false if it fails
- Deauthorize returns IIF Result object, but no CommandReturn object
  - For side “both” returns

DEREGISTERED	all done
DEREGISTERED + “ left”	right still authorized
DEREGISTERED + “ right”	left still authorized
AUTHORIZED	both still authorized

  - For side “left” or “right” returns
    - AUTHORIZED or DEREGISTERED

Side effects and recovery

- After an error, telescope subsystems may be in intermediate states
  - Example: PresetTelescope fails to acquire guide star
    - Telescope is on target and tracking
    - Guide probe stage has moved
    - Guiding is not active
- There is no “undo”
  - Fix the problem and repeat the command

CommandReturn success

- isError() is false
  - Commands that do not return data will have no result string
  - Commands that return data will have the data in the result strings
    - GetParameter, GetMultiParameter, GetRotatorTrajectory