Contents

• Introduction & Architectural overview
• Command request process
• Command result evaluation
• Commands at the IIF
• Status & Future improvements.
The IIF provides to the LBT instruments a set of commands by which they must communicate efficiently and transparently with LBT.

It is distributed to the instrument software teams as a set of libraries (code in c and c++).
**IIF Elements**

- **Deserialization library (TCS Side):**
  - CVS Path: IIF/TCS/
  - Files:
    - IIF/TCS/Position.cpp / .h
    - IIF/TCS/<Command>IIF.cpp / .h

- **Instrument side:**
  - Path: IIF/Instrument/
  - Files:
    - IIF/Instrument/IIF.cpp / .h
    - IIF/Instrument/IIFCommand.cpp / .h
    - IIF/Instrument/Result.cpp / .h and Status.cpp / .h

- **XML and XSD:**
  - IIF/Instrument/XMLSchemas/CommandParams.xsd
  - IIF/Instrument/XMLSchemas/sampleCommandParams.xml

- **Examples:**
  - IIF/Instrument/Examples (ansi-c and c++).
Creating the IIF

- Instruments identify themselves with the TCS by two pieces of information:
  - Their unique name (i.e. LBCRED, LINC, MODS1, etc)
  - Their focal station: location + telescope side (i.e. “prime left” or “bentGregorian left”, etc)

```cpp
IIF * anIIF;
try {
anIIF = new IIF("prime left", "LBCBLUE");
...```

Process Flow
Authorizing the instrument

- Before changes in the telescope state, the instruments must request authorization to the TCS.

- `bool IIF::authorize()`

- In case of success the telescope reserves the requested telescope side for exclusive use with the instrument.

- The telescope confirms success or failure of the operation.
Process Flow

Command request process

- Each IIF command returns a Result entity that contains a result code:
  - RESCODE_FAIL: indicates that there was a problem.
  - RESCODE_OK: successful case => Another component of the result becomes important, the Command Handle.

- Using a (non-null) command handle, the StatusInfo::GetCommandStatus() method inquires about the status of the command processing in the TCS.

- Status code (StatusInfo instance)
  - STATE_RUNNING: command is still being executed by the TCS
  - STATE_SUCCESS: TCS has successfully finished processing the command
  - STATE_CANCELED: TCS command processing has been previously canceled
  - STATE_FAILURE: Error occurred during the execution of the command
  - STATE_WRONGHANDLE: No valid TCS–side command associated with StatusInfo object.
Command result evaluation

主要集中

- Once command processing has finished on the TCS side, the StatusInfo object associated exhibits a status code that differs from STATE_RUNNING
- The StatusInfo instance provides method GetCommandResult which returns XML-serialized result from CSQ
- This XML string needs deserialization and handling via CommandReturn class (TCS Common SW)
- Retrieving the result data:
  - GetResultCount()
  - GetResultDescription(int n)
Process Flow

1. Init IIF()
2. Authorize()
3. Request IIF command
4. Get result code
5. Get Command Handle
6. Block() or Get command status
7. State_runninf
8. Deauthorize()
PresetTelescope

- Initiates observation sequence
- Slews telescope into target position

Arguments

- Target (incl. proper motion, color, magnitude)
- Guide star(s) requested
- Mode of operation (active, passive, guided, etc.)
- Initial rotator angle for IRC
- Mode of reference for IRC: 'position', 'vertical', or 'idle'
- Optionally: pointing offset to target, plus a “hotspot” where the target should appear in the focal plane
**Commands at the IIF**

**OffsetPointing**

- Sends the telescope to a new target position changing pointing coordinates

**Arguments**

- Instrument rotator offset (radians)
- Offset to the current target position (units)
- Side, 'left', 'right' or 'both'
 Commands at the IIF

OffsetGuiding

- Lets the instrument influence the guiding by sending target offsets
- Smaller & more frequent offsets than with Offset-Pointing
- Offset coordinates will be time-domain filtered by PCS??
- Keeps the pointing coordinates unchanged

Arguments

- Offset to the current target position.
- Side, 'left', 'right' or 'both'
- Instrument rotator offset (radians)
Commands at the IIF

MoveXY

- requests motion of M1 in its x–y plane
- motion request is meant relative to M1's current position

Arguments

- the desired motion along M1's x–axis. (range: −4.000 mm to 4.000 mm)
- the desired motion along M1's y–axis. (range: −4.000 mm to 4.000 mm)
- Side, 'left', 'right' or 'both'
**Commands at the IIF**

**MoveFocus**

- Request to move the respective primary mirror to a new focus position
- Absolute positioning (vs. relative move: StepFocus)

**Arguments**

- new position of M1's z-axis. (range: $-4.000$ mm to $4.000$ mm)
- Side, 'left', 'right' or 'both'
**Commands at the IIF**

**StepFocus**

- same as MoveFocus, but moves are relative to its current position

**Arguments**

- increment / decrement to position of M1's z-axis. (range: −4.000 mm to 4.000 mm)
- Side, 'left', 'right' or 'both'
Commands at the IIF

Standby

- Tells the telescope that the instrument is not using the specified side, but this will not be given over to another instrument for control.

Arguments

- Side: 'left', 'right' or 'both'
- Level: TBD.
Commands at the IIF

**Deauthorize**

- Tells the telescope that the instrument is not using the specified side.
- This side transits into a “idle” state and yield the control to the TCS.

**Arguments**

- Side: 'left', 'right' or 'both'
Commands at the IIF

LogEvent

- Enable the instrument to log a string into TCS log files with the format:
  "CSQ.<INSTRUMENTID>.<DESCRIPTION>

Arguments

- Event value (VALUE)
- Descriptive text (DESCRIPTION)
GetParameter

- Request various status information from TCS data dictionary
- Actual size & frequency of status information supplied are yet TBD.

Arguments

- Key name(s) for requested DD entry
- *not* a side-specific command
**SetParameter**

- Save information in the TCS data dictionary

**Arguments**

- Name of the parameter to be set by the command
- New value for the parameter.
- *not* a side-specific command
Commands at the IIF

GetCommandStatus

- All other command return only a 'command handle' (unique ID)
- Lets you inquire the status of command processing (queued, running, finished)

Arguments

- Command handle
GetRotatorTrajectory

- Request forecast of instrument rotator trajectory for limited time.
- Result is an array of \( \{t, \theta\} \) pairs, representing time (in MJD in days) & rotation angle (absolute, in radians) to be applied to rotator during pointing.

Arguments

- Number of seconds of desired look-ahead
- Number of seconds for the desired time interval b/w trajectory points.
- Start time for desired rotator prediction (in MJD, double precision, unit days)
Commands at the IIF

SendWavefront

- requests a compensation for a wavefront sensed by the instrument
- specific to instruments like LBC (in prime focus)

Arguments

- array of 28 Zernike coefficients to be compensated by M1 actuators. (ranges: TBD, precision is 0.01nm)
Commands at the IIF

**RotatePrimary**

- requests a rotation of M1 around specified rotation-al reference point on optical axis
- moves are relative to the mirror's current position

**Arguments**

- reference point's distance on axis above M1
- magnitude of the change in rotational angle relative to the reference point. (range: 0.000 to 999.999, specific limits depend on value of previous parameter)
- direction of rotation with respect to M1 plane. (range: 0.000000 to 6.283185307 radians, where 0 indicates positive direction along x axis, and \( \pi / 2 \) positive direction along y axis)
Commands at the IIF

TipTilt

• requests a tipping and/or tilting of M1
• moves are relative to the mirror's current position

Arguments

• desired change in rotation around M1's x−axis. (range: −999.999 to 999.999 micro radians)
• desired change in rotation around M1's y−axis. (range: −999.999 to 999.999 micro radians)
Future improvements

- Discussion and analysis of the use cases.
- New commands for all the instruments.
  - Move hexapod
  - Move tertiary (z, tip/tilt)
  - AOS, etc, ....
- Fill out template for new IIF commands !!!
- Updates and improvements:
  - Request a list of the reflective memory values (data dictionary items) in a single shot, saving some system overhead (serializing and deserializing of each individual request)
  - Review the definition of data structures, ranges, etc.
  - Remove SetIdle (deprecated)
  - Use the result structure to send remaining ranges ?
- Improve the IIF documentation (481g010d, 481s261a)