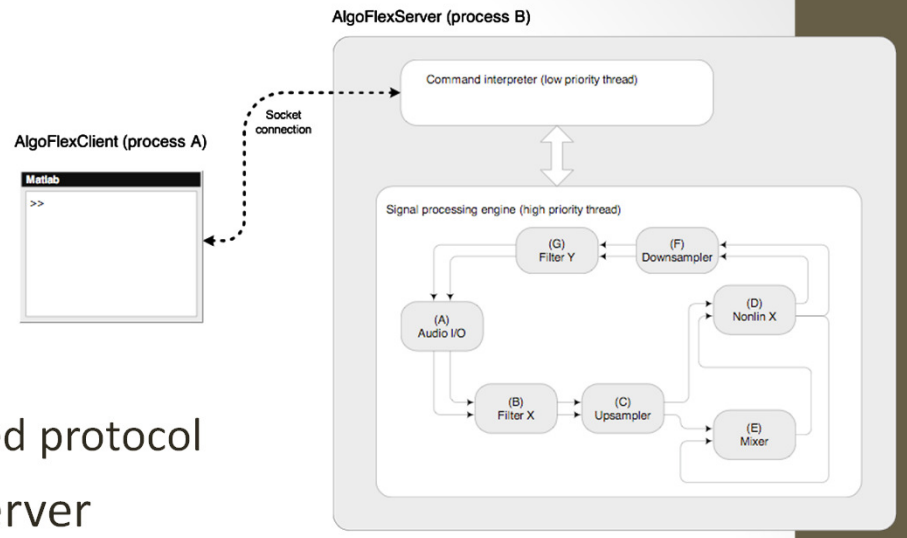
Advanced AlgoFlex

Best practices
and
Tips & tricks

Esben Skovenborg
MUSIC, December 2016



- You already know the **basics**:
  - AlgoFlex client and server
    - Commands and data via TCP-based protocol
  - Block instances on the AlgoFlex server
    - ”Audio” connections for synchroneous data
    - ”Parameter” connections for event-based data
  - Special blocks handle audio- and file-I/O
  - Running real-time vs ”batch” simulations
  - Installation, SVN
  - etc.!

- *… so now let’s dive into some advanced features*

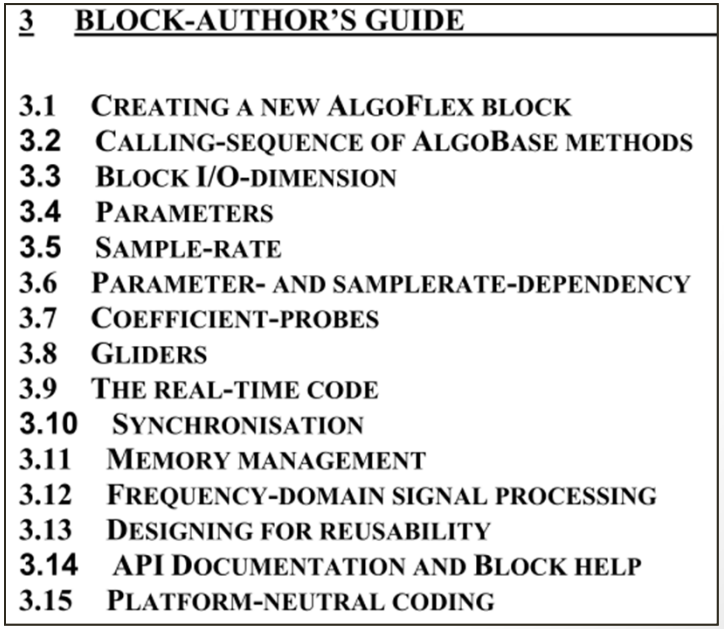
The AlgoFlex Manual

TC Group | Research

Last updated: 2009-05-05

- Writing a new block
  - When (not) to declare individual **parameter callbacks**
    - *RegisterDataParm("MyParm", &parmPtr, &parmUpdate)*
  - Parameter **dependency**
    - *RegisterCallbackDependency(&A, &B)*

  - AlgoFlex Manual, Ch. 3: **Block Author's Guide**

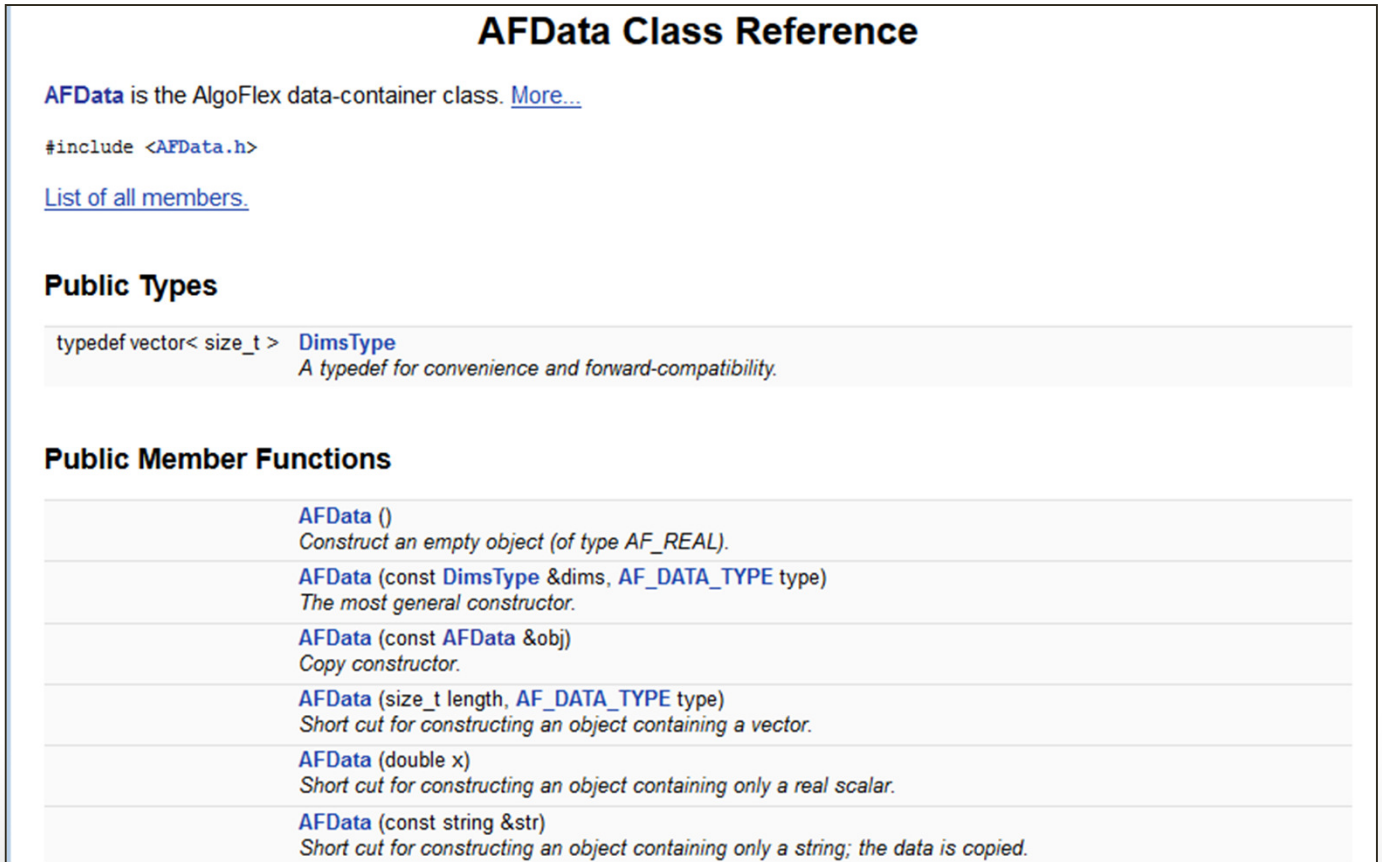


3 BLOCK-AUTHOR'S GUIDE

- Writing a new block (cont'd)
  - **AFData** container objects
    - Every AFData object has a *type* and a *dimension*
    - Similar to MATLAB
      - vector, matrix, string, and cell array
    - An AFData matrix can be indexed (without conversion to C array)
      - *myMatrix->GetAt<double>(x,y)*
    - Ref: Manual ch. 7.4, 7.5

- Writing a new block (cont'd)
  - **AFGliders** and coefficients
    - Gain and GainMatrix blocks as examples
    - Just use *AFGlider* instead of *double*
    - A glider's properties can be adjusted - even on runtime:
      - *AlgoFlexClient(srv,'SetData','MyGain','GLD_GainFactor',{'Linear','3','1e-6'})*
    - Ref: Manual ch. 3.8

- Friendly documentation of the **main classes**:
  - AlgoBase, AFData, AFGlider, …
    - Ref: AlgoFlex/doc/doxygen/html/index.html



AFData Class Reference

AFData is the AlgoFlex data-container class. More...

`#include <AFData.h>`

List of all members.

Public Types

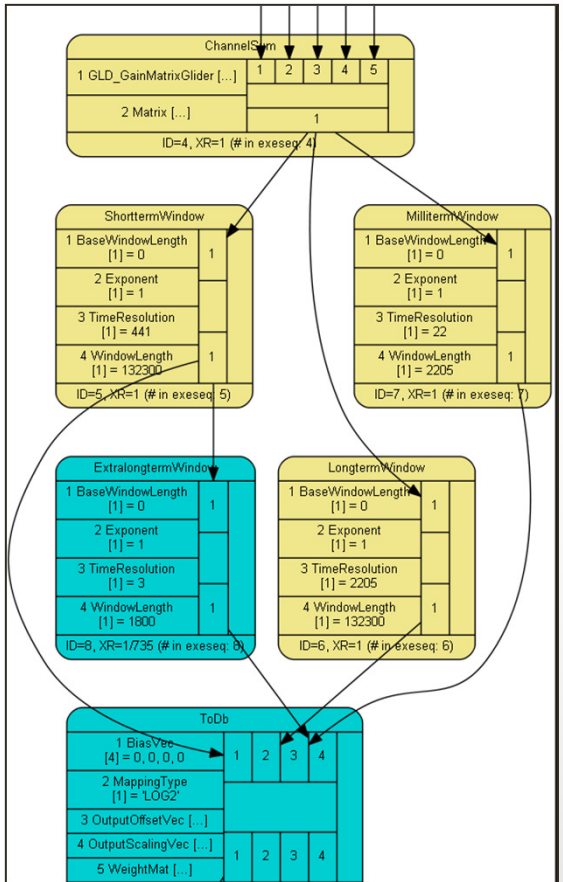
typedef vector< size_t >	**DimsType** *A typedef for convenience and forward-compatibility.*

Public Member Functions

	AFData () *Construct an empty object (of type AF_REAL).*
	AFData (const **DimsType** &dims, **AF_DATA_TYPE** type) *The most general constructor.*
	AFData (const **AFData** &obj) *Copy constructor.*
	AFData (size_t length, **AF_DATA_TYPE** type) *Short cut for constructing an object containing a vector.*
	AFData (double x) *Short cut for constructing an object containing only a real scalar.*
	AFData (const string &str) *Short cut for constructing an object containing only a string; the data is copied.*

- **Scheduling of blocks** in an algorithm
  - Execution-sequence
    - scheduling order of block instances
    - DetermineExeOrder server command
      - *automatic* optimal exe-sequence (experimental)
  - Up- and down-sampled block rates
    - relative to ”base rate” fs
  - *The blocks themselves don’t decide (but can ask on runtime)*
  - Ref: Manual ch. 2.6

- Generating a **block diagram**
  - Server command:
    - *GenerateGraph(shortcuts, hiddenblocks, what, size, format, filename)*
  - … so from Matlab:
    - *AlgoFlexClient(srvId,'GenerateGraph',{},{}, 'Audio+Parms',50,'svg','C:\tmp\MyDiagram')*
  - Ref: Manual ch. 7.11



- **XML diagrams**
  - Saving and loading an algorithm
    - not necessarily an entire effect
  - Block instances and routing
    - *Myalgo.diag.xml*
  - Partial presets
    - *Myalgo-1.parm.xml*
    - *Myalgo-2.parm.xml*
    - *…*
  - Use case:
    - a **complete** and **unambiguous** definition of an algorithm
    - suitable for revision control
      - but not for heavy editing
  - Ref: Manual ch. 6

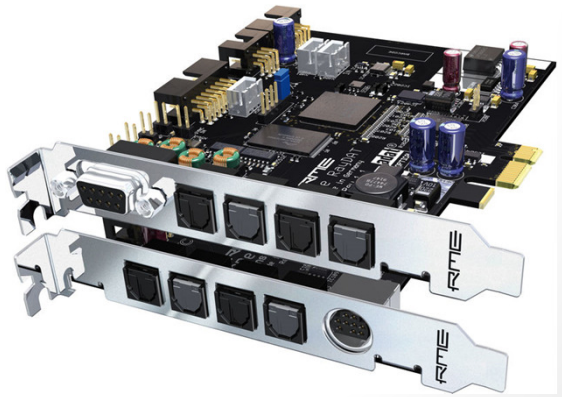
- **Vector and file processing**
  - *AlgoFlexEvals.m*
    - Simulating one AlgoFlex block as a MATLAB function
    - Ref: help AlgoFlexEvals
  - *AlgoFlexEval.m*
    - Simulating a **chain** of AlgoFlex blocks…
    - Ref: help AlgoFlexEval
  - **Audio file** processing
    - when your signal doesn't fit into RAM
      - *input file -> block chain -> output file*
    - Ref: help AFProcessFile

- Starting a **new server** (process)
  - *NewAFServer.m*
    - release/debug mode?
    - killing of already running servers?
    - minimised console window?
    - Ref: help NewAFServer
  - Multiple simultaneous servers - no problem
    - only the server ID is needed, after connecting
- For **distributed processing** -
  - start servers and connect manually
  - stream data between servers on LAN
    - TCPSource/TCPDestination blocks
  - Master/Slave server mode (*experimental*)

- AlgoFlex **clients**
  - Matlab (and Octave)
    - Java module handles network and protocol
    - Ref: AlgoFlex/Client/*
  - C++
    - Ref: MABUH
  - JavaScript
    - Ref: SILYN
  - AFGUI
    - (see other slide)
  - Server Command Files
    - **client-less** server scripts
    - Ref: Manuel ch. 7.14

- **Client-side callbacks**
  - Matlab's single-threaded event loop:
    - *while !quit*
      receive command
      interpret command
  - Callbacks from server to:
    - Matlab .m file
    - Java object
  - Great for meters and GUI updates
    - any AFData object
  - Ref: AlgoFlex\Client\DemoScripts\ClientCallback

- **Audio I/O** blocks
  - AsioStream vs AudioStream block
  - *AudioIOWiz.m*
    - making prototypes independent of *your* soundcard
  - DummySync block
  - TCPSource + TCPDestination = network streaming
    - Ref: Mabuh, Silyn
  - Linux and Jack
    - Ref: Andreas



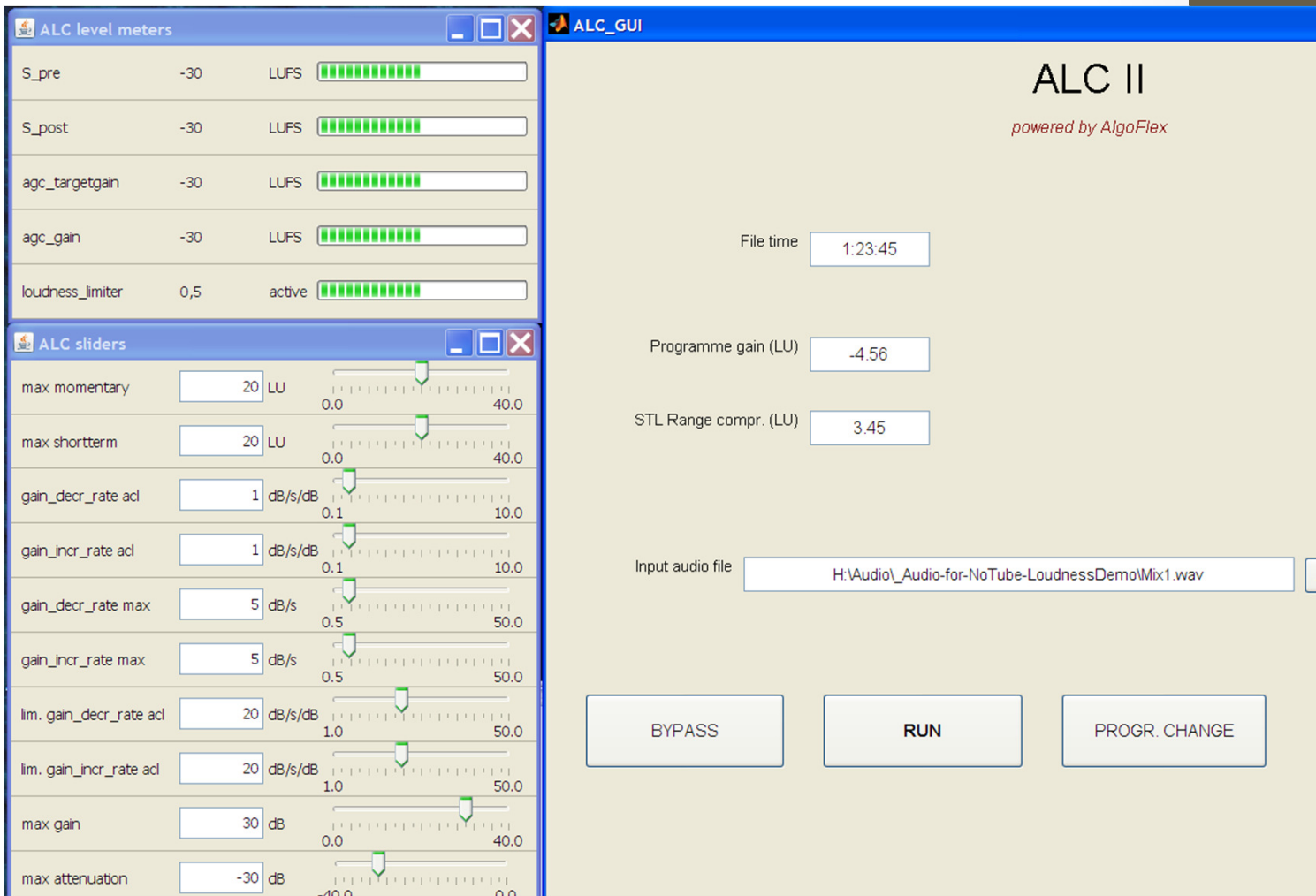
- **External controllers** and meters
  - MIDI
    - **MidiControl** block
      - … a wrapper around *RtMidi* library
    - Maps midi CC messages to/from AlgoFlex parameters in real-time
  - OSC (Open Sound Control)
    - ”a better midi” – network-based, typed, fast, hierarchical
    - **OpenSoundControl** block (*proof-of-concept*)
      - … a wrapper around *oscpack* library
    - iOS / Android apps

- AlgoFlex **Testing Framework**
  - Automatic regression tests
    - ”unit tests” per block
  - The magic starts with -
    - MyBlock\TESTCASES.m
    - *for example:* IirFilter\TESTCASES.m
  - Ref: help RunTests

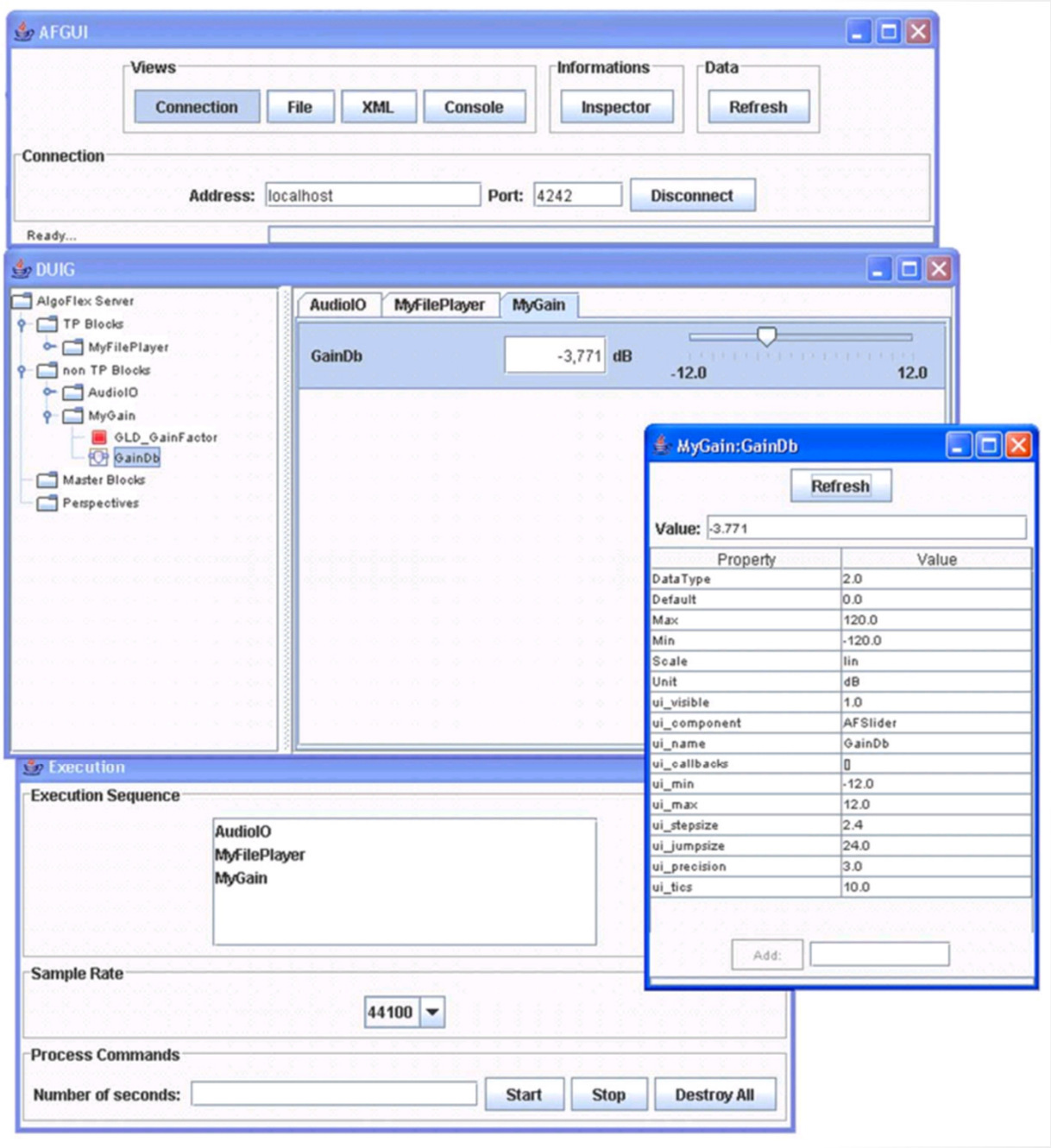
```
function TEST = TESTCASES()
% Define one or more test cases, using the AlgoFlex Testing Framework.


%% DEFINE TEST 1

[b,a] = butter(4, 0.1);  % a 4th order lowpass digital Butterworth filter
sos = tf2sos(b,a);  % convert to second-order sections

TEST(1).name = 'IIR filter simple';
TEST(1).parms = {sos,1,{}};
TEST(1).process = @(test_parms) process(test_parms{:});
TEST(1).reference = @(test_parms) reference(test_parms{:});
TEST(1).verify = @(y,y_ref,test_parms) verify(y,y_ref,test_parms{:});

%% DEFINE TEST 2

[b,a] = butter(6, 0.1); sos3(:,:,1) = tf2sos(b,a);
[b,a] = butter(6, 0.2); sos3(:,:,2) = tf2sos(b,a);
[b,a] = butter(6, 0.3); sos3(:,:,3) = tf2sos(b,a);
TEST(2).name = 'IIR filter multi-channel, individual coefficients';
TEST(2).parms = {sos3,3,{}};
% Reuse process() and verify() functions from TEST(1), above.
```

- **GUI** for rapid prototyping
  - AFComponents
    - Ref: Manual ch. 5
  - Java sliders and meters from Matlab
    - AlgoFlex/Client/
      - create_af_meter.m, create_af_slider.m
      - make_scalable_gui.m, make_simple_gui.m
    - Client\DemoScripts\AFComponents
  - **AFGUI** application (aka. Auto GUI, aka. DUIG)
    - Status: maintenance required
    - Ref: Manual ch. 4

- *AFComponents in action:*



- *AFGUI in action:*



- **Chunk**-based processing
  - in an AlgoFlex server
    - AlgoFlex\Client\DemoScripts\FilterTestScript_chunkproc.m
    - basically:
      - *AlgoFlexClient(srv,'SetSystemParameter','ProcessingMethod','Chunked')*
      - *AlgoFlexClient(srv,'SetSystemParameter','ChunkSize',64)*
      - before block instantiation
  - in an *AlgoLib* project
    - (other slide)
  - Algo feedback and latency considerations
  - Ref: Manuel ch. 7.10

- AlgoLib **code-generation**
  - a.k.a. Native Processing Framework
    - auto-generating C++ code for an algorithm by inling and glueing the blocks together
    - Result: a link library with a generic C++ interface
  - *Basically works, but some rough edges*
    - *used in several plug-in products*
  - AlgoFlex\NativeAlgorithmFramework*
    - class AFAlgorithmManager
    - class AFAlgorithmInterface
  - Ref: Manuel ch. 7.9

- **Metablock** feature
  - Handle a sub-algorithm as a separate block
  - … with routed internal connections
  - … with internal execution-sequence and -rates
  - … with selected external parameters
  - **Hierarchical** algorithm development
  - Use to separate *implementation* from *application*
    - in block diagrams
    - in XML or CreateAlgo.m scripts
  - More effective **reuse** of higher-level structures
  - Status: Not yet in SVN trunk (90% done)

- AlgoFlex is -
  - Rapid prototyping
  - Reference implementations
  - *A different tool for different users*

- ”In-house open source software”
  - community support
  - patches wellcome ☺
  - Collaboration tools:
    - SVN (Git?)
    - TestTrack (Jira?)
    - Twiki (Confluence??)
    - AlgoFlex-users@tcelectronic.com???
    - *coordination of future development…?*

TIME FOR QUESTIONS?