Eclipse
Identity
Recognition
Corporation

Frame Source Plugin Architecture

Methods of getting images into INDI² for processing

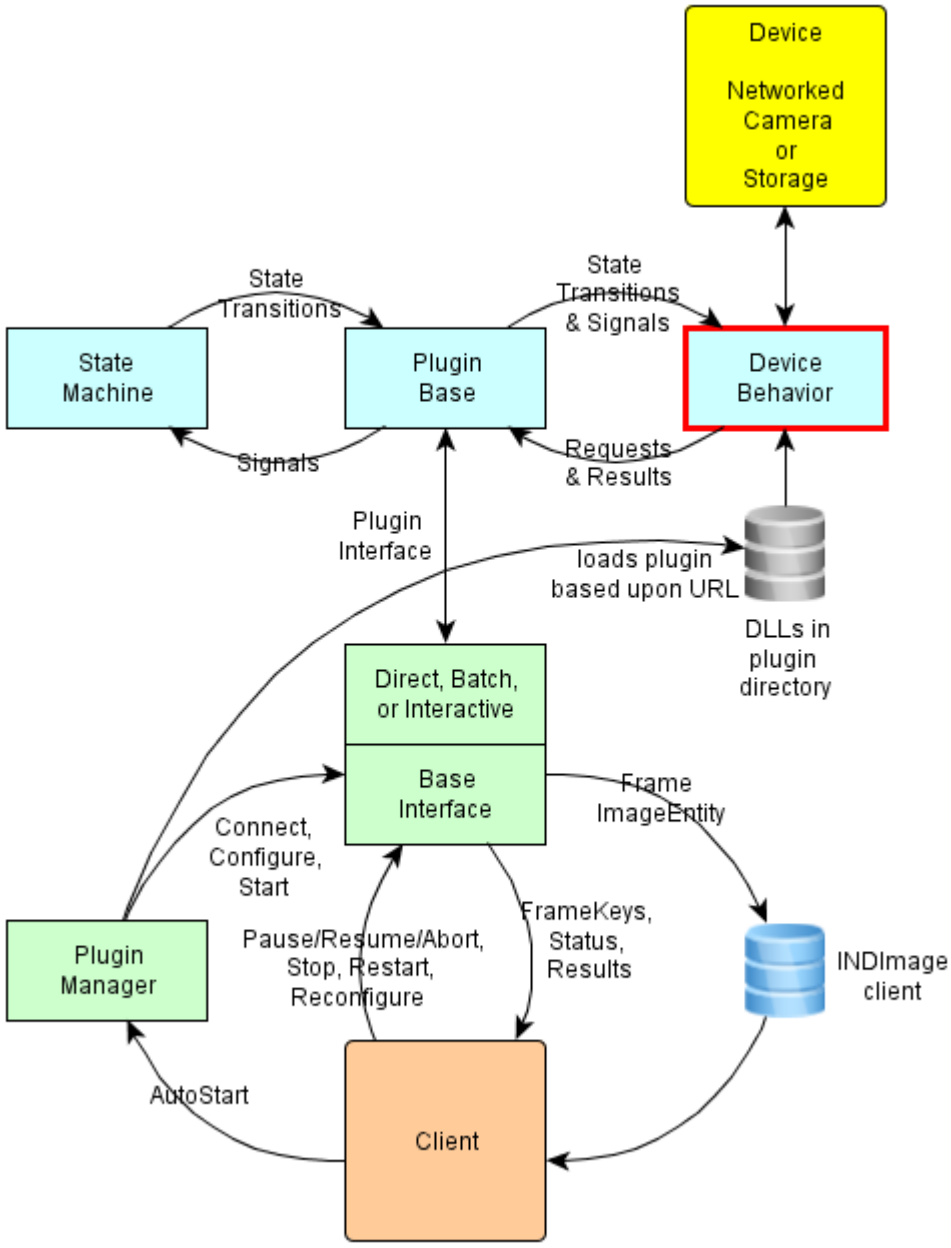
August 20, 2012 v2.00T

EclipseIR's second generation technology (INDIface²) will be presented to client applications in many ways. One of the ways that will contribute to that flexibility will be the **Frame Source Plugin Architecture** described within. It allows EclipseIR or its OEM customers to obtain image frames from diverse sources with a minimum of custom code.

Frame Sources

A Frame Source is a IP device containing one or more channels of live or stored video. You first **connect** to the device, **configure** its operation, and then **start** it on a channel (from specified begin and end times for stored frames). For interactive client interfaces, the input stream can be **pause**d, **resume**d, or **abort**ed.

Frame sources are defined based upon the network interface and protocol. They are defined by their URL's scheme portion (the stuff before the first colon). Only the device's behavior (indicated by the red box below) has to be defined for each new protocol. At this time three client interfaces are planned: Direct, Batch, and Interactive.



Builtin

Three frame sources will be built in to provide basic functions.

Scheme	Description
http	Support for a single URL that can retrieve the current frame from a source.
dirs	Support for a static directory and sub-directory structure. It scans the structure in a single pass. In batch processing the frames can start to be processed immediately.
ldir	Support for a live directory that starts empty and frames are processed as they arrive in the directory.

Plugin

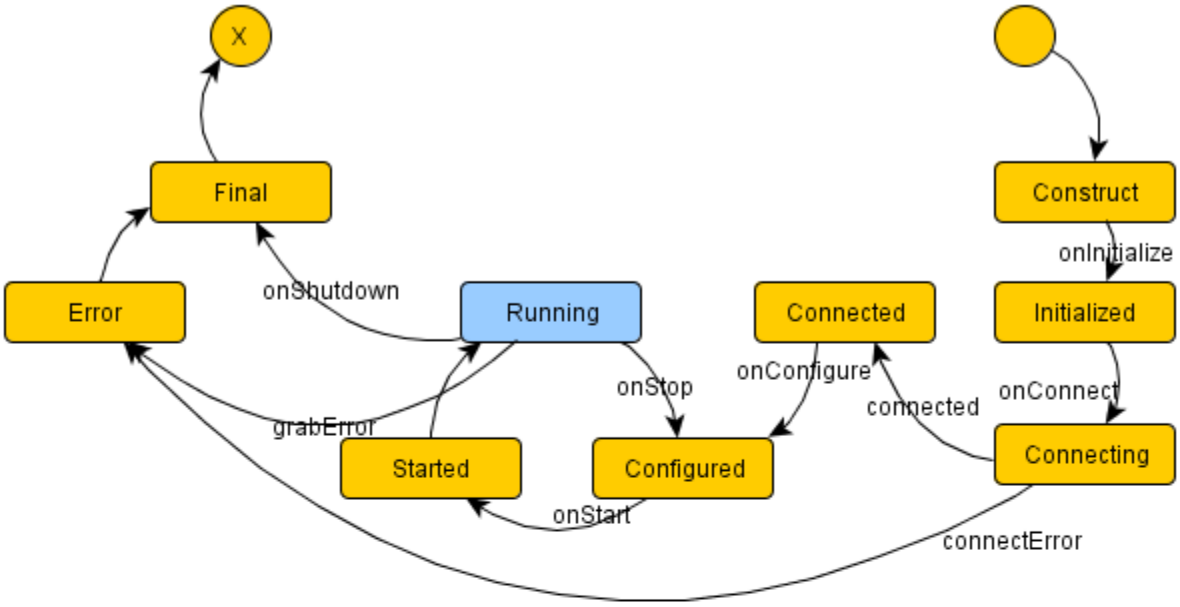
We will use the plugin facility in the Qt development framework to be able to quickly integrate new frame source protocols. The plugin will have to support any necessary enter and exit functions for each state. (See State Machine below.)

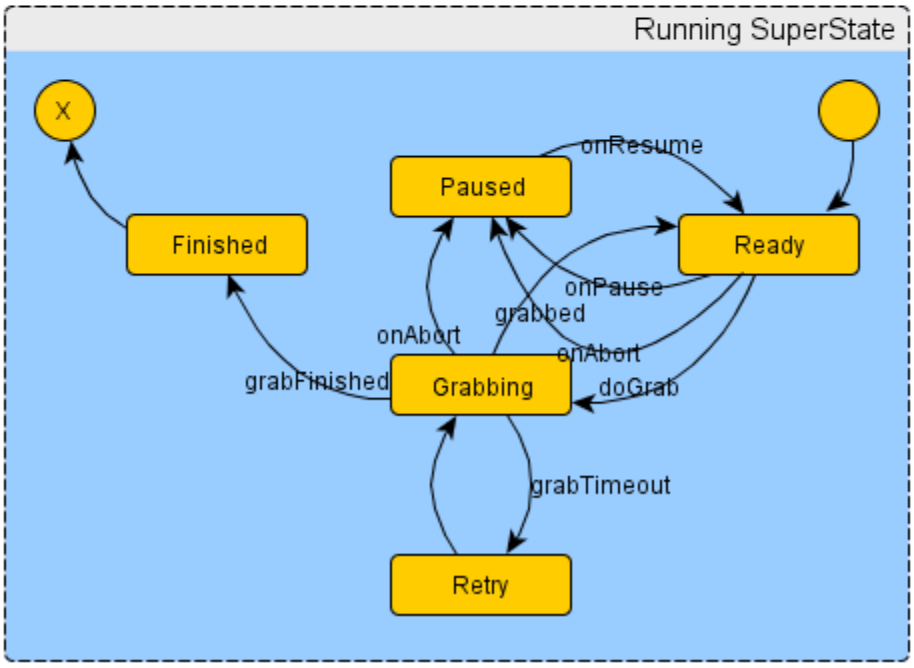
Scheme	Description
vigilLive	Support for retrieving the current frame from a camera attached to a 3xLogic DVR.
vigilPlay	Support for retrieving stored frames from any camera attached to a 3xLogic DVR for a specified begin and end time.
mjpFile	Support for reading frames stored in a MJP file by a 3xLogic DVR server or client.
wvhttp	Support for the session-oriented http protocol used by the Canon USA camera.
axis2	Support for Axis 200-series cameras and servers such as 210A and 240.
aviFile	Support for Microsoft AVI video files.
usbCam	Support for USB-based cameras.[1]
ftp	Support for directories obtained over FTP protocol.
onvif	Support from ONVIF compliant cameras and recorders.[2]

1 May need to be separate plugins for each camera's protocol.
2 In addition to INDIface's interface as an ONVIF analytic component.

State Machine

The plugin code will work in conjunction with a state machine that reacts to commands from the consumer of the frame images.





Interfaces

Three interfaces between the client and the Frame Source Architecture will be provided.

Commands

The consumer can issue the following commands to the Frame Source Architecture

Command	Description
Connect	Initiate a connection to the authority specified in the URL.
Configure	Send a set of configuration variables to the interface.
Start	Start collecting frame images using the given configuration.
Stop	Stop collecting frames.
Pause	Temporarily stop collecting frames.
Resume	Start collecting frames after a pause.
Abort	Abandon collection of the current frame and pause. May result in data loss.
Shutdown	Disconnect from the source.

Status & results

The interface will make available to the client results of commands and its current status.

Direct Interface

The direct interface provides frames to the client as they are received from the plugin.

Options

SampleMsec - The interval in milliseconds that frames are requested from the plugin. This should be zero for stored video so that frames are acquired as fast as possible.

Batch Interface

TBD

Interactive Interface

TBD

Builtin Source Reference

The usage information for the built-in frame sources is documented here.

HypterText Transfer Protocol [http]

TBD

Stored Directories [sdir]

TBD

Live Directory [ldir]

TBD

URL

TBD

Options

NoDeleteFirst – do not delete existing files before processing the directory
DeleteAfter – delete image files after they are ingested
DeleteAll – also delete non-image files as they are encountered
others TBD

Plugin Source Reference

Vigil DVR [vigilLive & vigilPlay]

This plugin supports 3xLogic's Vigil DVR Server HTTP protocol. The live frames from any of the cameras or frames stored in the server can be retrieved.

URL

Specify only the scheme (vigilLive or vigilPlay which will be converted to http by the plugin) and authority of the URL[3]. The plugin will supply paths and queries as needed by the Vigil DVR HTTP protocol.

Configuration

Channel - The camera number on the Vigil DVR (1 to 32).

Begin - The beginning time for playback. [vigilPlay only]

End - The ending time for playback. [vigilPlay only]

> Times can be specified in ISO standard date-time string, such as “2012-07-12T12:34:56.798” or numerically as the number of milliseconds since 1970-01-01 (so called epoch milliseconds). Note that due to limitations of the ISO date-time parser, the seconds and milliseconds need to be specified for the time to be correctly recognized. Note further due to the limitations of the Vigil HTTP protocol the seconds and milliseconds are ignored.

3 See Anatomy of a URL below.

Client Message Interface

TBD

Client Qt/C++ Interface

TBD

Pluging Programmer's Qt/C++ Interface

TBD

GrabberIR Application

A sample Grabber.exe application is provided to demonstrate the use of the Frame Source Plugin Architecture. In addition the Grabber application can be configured to output the frames captured from a frame source to a directory of individual frame files for further processing later.

Configuration

The configuration can be supplied in a .INI file or via the Windows registry. An .INI file can be specified on the command line as `Grabber @/path/to/Config.ini`. The default registry key is `HKCU/Software/EclipseIR/GrabberIR`. This can be overridden on the command line by specifying `Grabber %orgName/appName` where `orgName` would replace EclipseIR and `appName` would replace GrabberIR in the default key. in addition configuration values can be overridden on the command line by specifying /key/name=value. For example, /source/pluginDir=G:/lobal/plugindirectory would override the value from the registry or INI file. The following keys and names are recognized under the specified key or within the .INI file:

Source

The values below are interpreted by the grabber application and the architecture. Each plug-in may also have values specified in this key that have meaning to a particular plug-in.

PluginDir - Specify the directory where the plug-in files are located. The default is frame sources below the location of the Grabber.exe file.

Interface - Reserved for the future. At this time only the direct client interface is supported. See that section above for more information.

URL - Specify the URL appropriate to the plug-in needed. See the built-in and plug-in sections above for details on each one.

Channel - For multi-channel devices specify the channel number from one to the maximum number of available channels. Leave this value blank or zero for single channel devices.

Begin - Specify the begin time for retrieving stored video. See the built-in and plug-in sections above for details.

End - Specify the in time for retrieving stored video. See the built-in and plug-in sections above were details.

SampleMsec - For interactive client interfaces, specify the number of milliseconds between samples. For details, see each client interface section above. For batch client interfaces, typically specify zero for this value to collect frames as fast as they are available from the source.

ConnectMsec - Reserved for future use.

StartMsec - Specify the number of milliseconds to wait before the start of a stream times out.

GetMsec - Specify the number of milliseconds to wait on an individual frame retrieval before it times out. The architecture will try it again to retrieve that frame.

MaxRetry - Specify the number of frame retrieval timeouts that will be allowed before declaring an error on that stream. Each time a frame retrieval times out the retry value will be incremented and each time there is a successful frame retrieval this value will be decremented but not below zero.

Source/MessageImage

Controls the usage and appearance of message images in the stream. Implementation and complete documtation TBA.

Output

BaseDir - you can specify a base directory where the rest of the output directories will be created.

CaptureDir - This is the directory where captured frames from the source will be written. If relative, it is relative to the BaseDir specified above; if absolute, the absolute directory is used; if blank, no captured frames are written.

Options

AutoStart - If true, upon starting the grabber application it will automatically start a stream given the parameters in the source key above.

Command - The client can write commands to this value to control the operation of the grabber

application and through it the rest of the architecture. The client should only write to this value when the value is blank. The grabber application will clear this value upon acceptance of the command. See the command section below for the supported commands and their effects.

Status - [Read Only] status messages from the grabber application as well as the architecture will be written here. The client cannot write to this value.

UpdateMsec - Specify the number of milliseconds between scans of the registry for changes to the command value.

Log

Complete documentation TBA. To get a complete tracing log, specify the following in the log key:

Trace/Device=TxtFile

Trace/FileName=location/of/file.log [relative to Ouput/BaseDir]

Trace/Options=LevelChar TimeOfDay ShowFile ShowLine

Trace/Filter/999=~

Commands

The following commands are supported by the Frame Source Plugin Architecture.

AutoStart	Connect, configure, and start a stream according to the values in the Source key.
AutoStop	Shut down a running stream and disconnect its interface. This command must be issued between other autostart commands at this time.
Pause	Pause a running stream. (Only valid in interactive interfaces.)
Resume	Resume a paused stream. (Only valid in interactive interfaces.)
Connect	Reserved for future.
Configure	Reserved for future.
Start	Reserved for future.
Stop	Reserved for future.
Shutdown	Reserved for future.
Disconnect	Reserved for future.

Other Information

The anatomy of a URL

bhttp://admin:QS3afa3aa@server31.EclipseIR.com:8527/live?camera=1&quality=90
scheme = bhttp
user = admin
password = OS3afa3aa
user info = admin@OS3afa3aa
host = server32.EclipseIR.com
port = 8527
path = live
queries = ?camera=1&quality=90
query 1 = camera
value 1 = 1
query 2 = quality
value 2 = 90