Contents

V4.0 Notes

Highlights

This version supports the ability for "SlingerPlayer" to query Slinger so it can automatically populate a list of valid URL for the player to use. What is "SlingerPlayer". See below. The Windows .exe is now 32-bit by default, so there is no need for those who need a 32-bit executable to have to build their own version. To support the SlingerPlayer auto-populate feature, I've implemented a rudimentary API that users can call to locate the Sever on the LAN, get configured Slingbox information and send IR keycodes. See details below.

Updates

For all those people who don't like to type long URLs: In the [SERVER] section of the config file you can set URLbase to an empty string. i.e. URLbase=
With this option the software will now accept URLs like this for config files with the [SLINGBOXES] section
http://IP:Port/SlingBoxID
or just http://IP:Port if using a config file with the [SLINGBOX] section.
Warning: If you've opened up a port forwarding rule to remotely access Slinger, I would highly recommend not setting your URL base to null especially if you are using the default 8080 port.. This should help prevent hackers/port scanners from easily starting up an unauthorized video stream.

The code that returns the current streamer status in the Remote Control handler had been checking for a string that included an HTML header "<h3>Status:%s</h3>". This unfortunately forced some undesirable Remote Page formatting on the user. It now just looks for "Status:%s" and replaces the

whole string. The default remote no longer has the Status line as default, If you need it, you can add a line like this somewhere on your custom form.
:
Status:%s

The status returned by the Remote Control handler now includes the Slingbox name. This will help people with multiple Slingboxes using the same remote definitions file to know which box they are controlling.

When Slinger probes the LAN for available Slingboxes it will now print out the type of Slingbox detected along with the IP address port and finderid. It will use this type information if it is missing in the slingbox definition.

Slinger now prints out a warning if the “netifaces” module is missing instead of reporting an exception.

Slinger now gracefully handles the case where the Slingbox closes the control connection unexpectedly.

Slinger will retry three times to kick-off another user before giving up. Previously this turned into an infinite loop on a hosed up Slingbox. People have reported this issue on some Solo boxes. A reboot of the Slingbox is usually required to resolve the issue.

If no keycodes are detected when the stream starts up Slinger now generates a warning that the user might have selected the wrong video source.

The was some debug code still in place in the Solo/ProHD video stream processing. That has been removed and it should be faster now.

Slinger now supports selecting channels numbers above 255 on the ProHD tuner. THX Bob.

Slinger now supports Audio-Only channels on ProHD tuner. THX Bob.

There was a bug if you only had one keycode in your “StartChannel=”. It has been resolved.

Slinger now prints out a warning if the “Flask” module is missing instead of generating an exception.

Slinger now prints out Slinger IP address on startup, for the CMD window challenged.

Slinger no longer supports my web service for locating Slinger using the Slingbox FinderID. You will need to use a real Dynamic DNS service instead. If you configure a FinderId in your config file. It will be ignored.

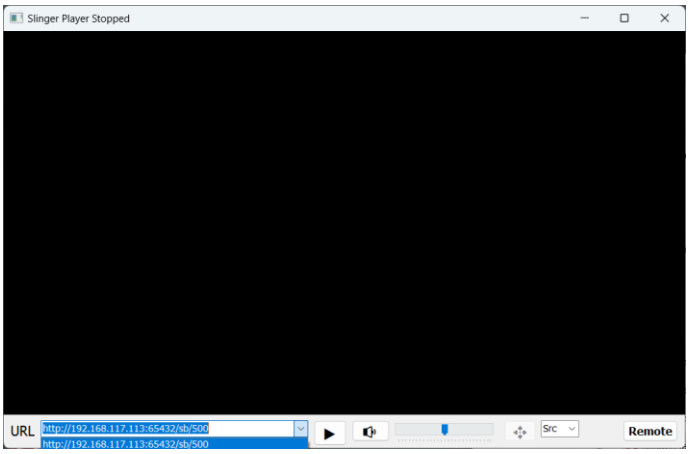
Recently some makers of HTML encoded remotes have wanted to add “Restart” buttons to their remotes. Which has highlighted an inconsistency in my implementation. To remedy the situation, I’ve defined a keycode for “Restart” of -1 which will cause Slinger to shutdown. It’s assumed that the user is running Slinger in a loop that will restart it when it stops. This will make the implementation for remotes consistent and not require any special handling.

Slinger will now replace “Slingbox” in the video stream header with the name of the slingbox being streamed. Note the length of the name is limited to 8 characters. If your slingbox name is longer than that it is truncated. THX mdehrlich.

There was an error path when the session failed to startup due to an issue with the slingbox and the number of remote connections wasn’t properly getting decremented.

Slinger Player

In an act of self-preservation to try to stem the constant stream of “I did something wrong on my stream Player (I’m talking about you VLC) and I can’t connect to Slinger” discussions. I’ve written a Player customized for accessing Slinger. With Slinger Player (SP) and Slinger V4.0, SP will search for Slinger on the LAN and auto populate a list of valid URLs to access Slinger with.



You can select one of the probed URL to start playing it or type a new one into the dialog box and then press the play button. Valid URLs are saved on disk so any sites that aren’t on your LAN will be saved.

There is a minimal user interface following the well know “KISS” principal. There are a Play, Mute, Volume, Full Screen, Aspect Ratio and Remote controls. The Remote button will pop up a minimal browser window and automatically load the remote control page for the selected URL. Yeah, no more “I typed the wrong URL into my browser and I can’t connect to the remote control page” support requests.

SP is configured with minimal buffering and knows in advance what kind of video stream it is connecting to, so it starts up faster than VLC and has minimal latency. The latency is ~ 1 second compared to 4-5+ seconds that VLC buffers out of the box. Yeah, no more “I clicked on my remote button but the result doesn’t show up for 4-5 seconds making the remote unusable for many scenarios. The one second-ish lag now makes it usable for most scenarios, FF and REW are still a little problematic but if you are aware of the lag you can do a much better job of getting it almost right.

In full screen mode there are some keyboard shortcuts you can use if your keyboard/PC doesn’t have Volume and Mute function keys. In fullscreen mode the keys +,-,m map to Vol+,Vol1 and Mute respectively. The “shifted” version of those keys also work for the keyboard challenged (me). You can press “Esc” to exit fullscreen mode.

SP will accept two different command line options. One for the URL you want to connect to automatically. I.E. SlingerPlayer http://192.168.117.102:65432/sb/500.

The second option is if you want to get your configuration from a Slinger not on your LAN. In this case put this on your command line. wan://IPAddressOfSlinger:PortNumber I.E. wan://192.169.232.22:65432 Note: You can use a FQDN here which might be useful if your using a Dynamic DNS service to connect to a remote Slinger. I.E. wan://MyDDNS.org:65432

The drop down URL list is saved in the Sites.dat file. You can clear out superfluous entries by editing that file.

You can easily make shortcuts to SP on the Desktop so different configurations are just a double click away.

Currently SP is a Windows only application. If it gets some traction, I might be convinced to generate an Android version.

Installing SlingerPlayer

A SlingerPlayerFull.zip file will be made available on the GitHub project. You can unzip it anywhere you like. It does not have to be on the same host as Slinger. It includes all the required DLLs. They must be in the same folder as SlingerPlayer.exe.

You do not have to use Slinger V4.0 along with SlingerPlayer. If you don't you will lose the auto-populate functionality but it will still work with manually entered URLs.

Slinger API

Locating Slinger on the LAN

Slinger now will respond to a UDP Broadcast message on port 50005. The request must contain the characters "GETSERVERCONFIG".

Slinger will respond with a '|' delimited list of [SERVER] information for all of the config files on Slinger's command line.
i.e. port=65432,maxremotestreams=2,urlbase=sb|port=8086,maxstreams=4,urlbase=sb

port=65432,maxremotestreams=2,urlbase=sb|port=8086,maxstreams=4,urlbase=sb
ProcessGetServerInfoport=65432,maxremotestreams=2,urlbase=sb|port=8086,maxstreams=4,urlbase=sb
Note: You may have to open UDP port 50005 on the firewall where Slinger is running to get this to work.

Slinger will also respond to GETSERVERCONFIG requests using TCP. In this case the user needs to know the TCP port(s) Slinger is listening on to get the URLbase information. This is needed if the API user is not on the LAN.

GETSLINGBOXNAMES

Slinger will respond to a TCP request with the characters "GETSLINGBOXNAMES".

Slinger will respond with a comma separated list of the name=value pairs need to generate streaming and Remote requests. i.e.
sb1=500,sb2=Solo,sb3=M1,sb4=ProHD,sb5=ProHDtuner,sb6=SoloAudio,sb7=M1Audio
In what are configured in the [SLINGBOXES] section. If there is a [SLINGBOX] section just "slingbox=/" will be returned.

GETVERSION

Slinger will return the version string.

Notes on returned TCP data

When Slinger returns data on the socket used in the GETFILE and GETSLINGBOXIDS requests, it prepends the length of the following data. This should make it easier for clients to manage retrieving the requested information. The length is a 32 bit unsigned integer using the default Network encoding (big endian).

SENDIRKEYCODES

Slinger will respond to a TCP request that contains the characters
"SENDIRKEYCODES{slingboxID}=keycodes"
Where {slingboxID} is one of the IDs returned by "GETSLINGBOXIDS"
keycodes is a "+" delimited list of keycodes to send to the Slingbox. It uses the same format as the documented "StartChannel" config file option. i.e. SENDIRKEYCODES/sb/500= 9+13+13 would change the channel to "155".
Note: There must be an active stream for this to work.