****

Richmond Sound Design Ltd.
Theatre Sound Design, Show Control & Virtual Sound System Software

SoundMan-Assistant Release Notes
****

Version 1.0.172.0
2016-12-29

****

1) Changed the copyright to 2017 in the About box and version info.

2) When using the AB64 network interface, you would get continuous warning messages in the log that the GPIO interface was not open, even if the GPIO interface was not licensed.

3) Corrected some minor coding errors in GPIO driver loading.

****

Version 1.0.105.0
2015-12-15

****

1) When running an ASIOGroup with multiple ASIO devices in it, the debug code that checked if the callback was on the same thread as previously would get very confused and make a log message for every callback, slowing things down to a crawl. This has been disabled.

2) Turned off the ASIOGroup debug that displayed a debug message on the first 40 callbacks or so.

3) Removed extraneous newline from "ASIO Driver started successfully" message.

4) Changed the log messages for failures setting thread priority to be more explicit about what the requested priority was.

5) Updated various copyright fields to include 2016.

6) When taking a dump, the date and time in the dump file name are the local date and time, not UCT as in the previous version.

7) There was a slight chance of a crash when loading a preset without first opening the ASIO interface. That has now been covered with an appropriate error message for restoring channel data without an interface. The original patch that covered 99.9% of this case was made in version 82.5 in 02/12/2014.

****

Version 1.0.104.0
2015-11-23

****

1) It looks like Radar is crashing in EnumerateProcessThreads. Wrap it in a try/catch and add some more debugging to tell where it is being called from when and if it crashes.

2) Examined many uses of EXCEPTION_EXECUTE_HANDLER and changed a number of them to try to take a dump when an exception occurs. Maybe this will stop the problem we have with crashing and not taking a dump.

3) Changed a number of TRACE commands to ODS to be sure we get the debug message out even in a release build. These are generally in places where something very bad has gone wrong.

****

Version 1.0.103.0
2015-11-19

****

1) Wrapped ASIO callback in try/catch to catch any possible processing errors. SMS will log a message and take a dump if something goes wrong during callback processing.

2) When the callback thread stops running the overtime messages are sent to the debug output as well as the log. This can make it easier to synchronize the debug log with the main log.

3) When a dump is taken for any reason the notice of the dump is sent to the debug output as well as the log. Again, this helps in seeing major events when only looking at the debug output.

4) Removed unnecessary use of GetThreadId function that is not available on Windows XP Embedded systems.

5) Corrected logging of the license information, which attempted to make two log lines with a single log call. The code now properly makes two single-line log calls.

6) When the ASIO callback thread dies things don't work very well, but if some external source continues to send play commands to SMS, we would get a whole lot of log messages about hung channels, and get a dump every 10 seconds to boot. The dump in this case is pointless, as is waiting 10 seconds before giving up on the play attempt. Shortened the wait interval to 2 seconds in this case, and eliminated the dump. If a playback selection gets hung for some reason while the callback thread is working we will still wait 10 seconds and take a dump.

7) When building an ASIOGroup, if one of the devices had a fixed buffer size, SMS could crash with a divide by zero error.

8) Also, if the device reported any of a number of possible insane values as a result of the ASIOGetBufferSizes call, the bad values would have been ignored and could have caused problems.

9) Some unlikely but possible optimizations were not done when combining devices into an ASIOGroup. These optimizations are now attempted, and if they happen to work they can make a better group blending.

****

Version 1.0.102.0
2015-10-05

****

1) asiosms was issuing a ResetRequest that should have been a Resync Request when it got a sample rate change request when running. This really didn't matter much, since it would only be issued for a sample rate change while running, and we don't allow sample rate changes while running, so the code should never be executed anyway.

2) Changed the ASIO interface to return 0 instead of 1 to a resync request. Maybe that will stop some drivers from issuing a continuous resync request.

3) Fixed a bad debug message that could crash the ASIO driverdurning the ASIOStart routine, without leaving any indications that this had happened.

4) Added a few more debugging messages.

****

Version 1.0.101.0
2015-09-17

****

1) Added additional debug messages to the AsioMultiple driver control path.

****

Version 1.0.100.0
2015-09-16

****

1) Moved project from VC6 to VS2010. Played lots of games with the build file to get things to build with minimal errors, warnings, and general gripes and inappropriate complaints.

2) Fixed some of the priority setting routine to return the correct error code when compiled in debug mode.

3) Added the thread id to several log and debug messages dealing with setting thread priority.

4) Fixed a bug in the priority handling code for Windows 7.

5) Fixed a problem where the main UI thread could inadvertently be set to realtime priority when starting the ASIO driver.

6) Made all debug and log messages mentioning threads consistently display the thread id in hex format to match the debugger display.

****

Version 0.170
2015-12-13

****

1) Updated copyrights to 2016.

2) Added support for new Advantech GPIO card driver interface. The new driver interface is nothing like the old driver interface, and is a great deal more complex to get the same simple results.

3) If the GPIO driver is loaded, SM-A will not output annoying popup messages about "Controlling application not running". The assumption is that if digital control is available that SM-D will not be the normal control method, and only used during show creation.

****

Version 0.169.1
2015-10-14

****

1) Added the DoADump routine from SM-S to get better dump data out of SM-A. The previous method of taking a dump made it hard to figure out what the actual cause of the problem was.

2) Fixed a problem with getting empty log messages in the dump file. The problem was caused by an oversight in the log change in the last version.

3) Eliminated a couple of useless global variables that led to problems in the way the logging code worked.

4) Updated the dump routine to put try/catch around the actual dump data creation. This should prevent empty memory dump files.

5) Documented the response to the QUERY_VERSION MIDI command. Fixed the generation of the version string, which was making an incorrectly formatted date value.

****

Version 0.169
2015-10-10

****

1) Updated copyrights to 2015.

2) Documented some of the logging procedures and methods, since I don't get this way much anymore, and I'm forgetting things since I don't work on it every day.

3) The logging changes made in the last version had a problem when running on a multi-processor system. Access to the log array was not thread safe, and could result in crashes. Added locking code.

****

Version 0.168
2014-05-19
Standard

****

1) The internal log is cleared every 65000 messages. If SoundMan-Monitor is running and messages are coming quickly, there was a chance that log messages just before the internal log was cleared could have been lost. There is now a short delay of a half second before clearing the internal log to give SM-M a chance to grab the current values.

2) When clearing the log, the last 100 or so messages of the old log are saved and put into the cleared log that is maintained in memory. This in-memory log is included in any crashdumps created by SM-A.

3) Previously if many messages to log occurred at once, and some of them were queued for logging while others were made directly, the messages would be logged out of order. Logging has been redone so that all messages are now logged in correct order, even when they occur in a quick burst.

4) Updated the copyright to 2014.

5) Improved the debug messages on GPIO triggers to better qualify when the cue number is a probably invalid value.

6) Changed the startup code that tries to start SM-S if it isn't running. There should be considerably less chance now of SM-A starting two copies of SM-S, and of the second SM-S copy failing to recognize that it should go away.

7) SM-A now waits up to one minute for SM-S to come ready, rather than waiting until SM-S started and then assumed that SM-S would be ready very soon.

8) The cue number in the cuelist display line was partially trimmed on the right. This has been fixed.

9) SM-A now logs a lot more information about the system and show configuration when it starts up.

10) If we try and fail to start SoundMan-Monitor, we then check to see if it is running. If it isn't, we complain that we started it but it isn't running. Of course we failed to start it, so it is perfectly reasonable that it wasn't running. Fixed the log messages so that if we can't start SM-M we don't complain that it isn't running.

11) When starting SoundMan-Monitor we did not wait long enough to see if it had successfully started.

12) The frequency of log messages about not being able to send to SM-S

when we can't start it and it isn't running has been greatly reduced.

****

Version 0.167
2013-03-07
Standard

****

1) Replaced the logging timer routine with a more accurate version.

****

Version 0.166.0
2011-12-31
Standard

****

1) The dongle serial number field in the configuration dialog can now be selected with the mouse and the text copied from the field.

2) The timestamp for log records could in some cases be off by several minutes from the correct value. This has been corrected.

3) Changed the copyright to include 2012.

****

Version 0.165.0
2011-12-06
Standard

****

1) Corrected a formatting problem in the textual description of the file transfer start response value (returned to SM-D) that is logged when a file download is attempted by SM-D.

2) Changed a number of direct calls to log a message in the file transfer path to queue log messages. This should eliminate some unexpected delays in the file transfer path that were resulting in messages appearing out of order.

3) Added logging as to whether the GPIO option is enabled in the dongle.

****

Version 0.164.1
2011-09-23
Standard

****

1) Added a textual description of the file transfer start response value (returned to SM-D) that is logged when a file download is attempted by SM-D.

2) Move the file transfer start request log line to an earlier point in the code so that the request will be logged even if the request is denied. This may help track down synchronism problems with SM-D.

3) If we get a Windows error trying to create a sound or show file we now log the Windows error number and text.

4) Changed a hack compatibility delay in SetupStoreEn from 300ms to 200ms. It seems that if there is no delay between receiving a SetupStoreEn command from ABSM that ABSM will fail on transfers of large sound files. But if the delay is too long SM-D will fail on all file transfers. There are indications that SM-D is failing on transfers that should work, so perhaps reducing this delay will make it more reliable, and perhaps won't disable the ability of ABSM to send large files.

****

Version 0.164.0
2011-09-08
Standard

****

1) Release the accumulation of patches since the last version.

****

Version 0.163.10
2011-08-01
USF Dervish

****

1) When using SM-A on a computer with multiple IP addresses, SM-A only checked the IP addresses of the host at the time that it was started. If a new IP address appeared at a later time it did not know that. This could lead to problems trying to get SM-D to talk over the new connection.

   The improved algorithm is not perfect, but if the two IP addresses assigned to the unit are reasonably far apart, and if the SM-D computer has an address that is reasonably close to the IP address on the SM-A box, SM-A should pick the right address. Unfortunately it does not seem to be possible to get your own IP address used for a received UDP message, so you have to guess what it might be.

****

Version 0.163.9
2011-07-20
USF Dervish

****

1) The GPIO handling has been moved to a separate thread so that it is not bogged down by any other processing, and will accurately show the real setup and hold times on the GPIO interface.

2) Eliminated a short delay that was happening inside a lock.
This speeds up playback handling and eliminates the "ripple" seen in SM-D when starting a large number of channels at once.

****

Version 0.163.8
2011-07-15
USF Dervish

****

1) Added logging code to log exactly when the GPIO lines change value.

2) Modified the log timestamp code to handle drift in the performance counter value.

****

Version 0.163.7
2011-07-11
USF Dervish

****

1) The GPIO playback and cue trigger lockout times were 1/10th of what they should have been. With the 1 second long playback trigger being sourced from the Dervish this is resulting in unexpected cue triggering.

2) The "Controlling application is not running" is now an information message rather than a warning message, and the popup balloon has been reworded to make this more obviously advisory.

3) When taking a dump, the popup messages said "SoundMan-Server" rather than the correct "SoundMan-Assistant".

4) Previously no attempt was made to start SoundMan-Server until the first command was sent to the server. Now one initial attempt will be made to start the server early in initialization. This will reduce the chance of later delays causing problems on slow systems.

****

Version 0.163
2011-04-18
USF Dervish

****

1) Disabled the Nagle algorithm and increased the size of the send and receive buffers on the link to SM-S.

2) When trying to send to SM-S and we get blocked (probably for no space in the sending buffer) we have to wait and retry the send. We now also continue to receive data from SM-S in this state, so that if SM-S is blocked trying to send to us, we can get out of the possible deadlock situation quickly.

3) Removed message boxes that appeared on taking a dump, as this could cause SM-A to lock up in an unattended install.

4) Changed the year from 2 to 4 digits in the log records.

5) Fixed a rounding error in the TimeToTimeCode routine that could result in loss of frame accuracy when times over 5 hours were used.

6) Fixed a bug in play track group that would prevent a group of tracks from playing if one track is the group did not have a loaded file.

7) The configuration setup has been modified to let the user control the "analog first" option on the ASIO connection. Analog first is set by default, and has been for a long time. When set, SM-S will move all of the channels with the word "analog" (in any combination of upper and lower case) to the front of the channel list on the interface, preserving the order of the analog channels. This is useful when you have a small license and the analog channels you want to access exceed the range of your license.

8) An "adat first" option has been added to the control panel. This is conceptually the same as the 'analog first" option, but it will move channels that have "ADAT" (in any case) to the front of the channel list on the interface. If both analog first and adat first options are set, both kinds of channels will be moved to the front of the list. The moved channels will remain in the same relative order, so the final order of the analog and adat channels will depend on the order they had on the interface before they were moved.

   The main use of this option is likely to be for users with a MOTU 828 Mk 3, which has 8 analog channels, then about 20 assorted other channels, and finally 16 ADAT channels.

9) Added a number of debug messages to the log relating to GPIO triggering of cues and playback.

10) SM-A was generating an invalid GET VU request to SM-S if the interface had no input channels.

11) Added a log message when a track loops.

12) The "Log file saved" log message was incorrectly formatted.

13) Added more GPIO debugging messages.

14) If an input is marked "playback only" SM-A won't send unmute messages to SM-S on the start of each playback as they are unnecessary.

15) Speeded up playback start slightly.

****

Version 0.162
2011-02-28

****

1) The first response message from SM-S showed erroneously as being overtime by many seconds. This has been corrected.

2) The start, stop, and resume points for a track are now sent to SM-S in samples rather than in seconds. This should eliminate rounding errors that could result in hiccups in positioning a track to an unusual start position.

   This REQUIRES using this version of SM-A with a compatible version of SM-S, which would be version 1.0.67 or later.

3) The code is smarter about asking for VU responses. This should cut down the chance for TCP/IP data overruns on slow links.

****

Version 1.0.161.0
2011-01-30

****

1) If the main window started minimized the toolbar icon menu still defaulted to "hide window" even though the window was already hidden.

2) Re-ordered the main window startup code to make is look better while it is searching for the dongle.

****

Version 1.0.160.1
2011-01-13

****

1) Added the machine serial number and the dongle ID (if the dongle is present) to the main config dialog on the top left. It should now be easier to find the machine serial number to get a temporary license file made.

2) Removed the "six channels in every eight" hack that had been added for one specific Cobranet installation, as the installation did not need the option after all, and it was an ugly hack.

3) Fixed a problem where the dongle serial number could display with incorrect leading FFFFs.

4) Change copyright to 2011.

5) Add extra debugging messages if the Advantec GPIO driver cannot be loaded.

6) Fixed some problems if a shutdown is done in the middle of starting up.

7) Added more error decoding to the GPIO setup path when attempting to open the digital IO card.

8) Added logging to show when playback or a cue is triggered from GPIO.

9) Fixed a problem with not closing the GPIO driver on exit.

10) Added additional log messages when attempting to start SoundMan-Server and SoundMan-Monitor.

****

Version 1.0.159.0
2010-07-08

****

1) Changing the root path or other configuration parameter will now completely reinitialize SM-A so that it picks up the new value. Previously it was necessary to restart SM-A after changing the disk path or the ASIO device before the change would be recognized.

2) Previously SM-A would only connect to SM-S if SM-S had been started first. SM-A will now detect that SM-S isn't present, and try every 30 seconds to connect to SM-S. This will usually succeed once SM-S becomes available. Note though that SM-S will not know the current show state when it connects in the middle of the show, and if a lot of cues depend on existing gain, eq, or other settings, they probably will not produce sound until the control values can be updated.

3) The UDP file transfer code incorrectly had show and audio file types reversed when it was checking if the current show had been replaced. The only result of this was that the completion message incorrectly identified the file type in the log.

4) An examination of the AB spec doesn't mention if delay enables are cleared to ON or OFF by default or by RESET or by CLEAR ALL DATA. There is no enable for crosspoint delays in a real AB, so it obviously defaults to ON all the time. There are indications that the input and output delays also default to ON, and probably aren't ever cleared to OFF in the real AB.

   The SM-D code was defaulting inputs and outputs to ON at startup time, but had crosspoints OFF and was clearing all of them to OFF in a RESET or CLEAR ALL DATA. There was also hack code to enable the delays whenever they were set non-zero, which defeated the input and output delay enable commands if they attempted to turn the delay enables off.

   Things have now been changed to set all delay enables to ON at startup time, and also at RESET and CLEAR ALL DATA. The hacks to turn the delay enabled on when the delays have been set non-zero have been removed. This should result in operation more consistent with a real AB, and perhaps make operation with ABShowMaker a little smoother. (ABSM does not appear to have any commands to enable or disable delays.)

5) Added code to keep the UI from hanging completely while polling for the dongle at startup. This should keep SM-M from wanting to take a dump because it believes (correctly) that the interface is hung.

6) When shutting down from the old SM-D that shut down SM-A and SM-S too, SM-A would erroneously restart SM-S just before exiting. This is now fixed.

7) Looking for the dongle on startup takes 20 seconds if there is no dongle (demo mode startup). The interface used to be hung during this time, resulting in nasty messages in the log in SoundMan-Monitor, and possibly even requests from SM-M to take a program dump to analyze the hang. The interface will no longer be hung while looking for the dongle, so SM-M will not log nasty messages and try to take dumps. In addition to the 20 seconds for SM-A startup waiting for the dongle to appear, there was also an additional 20 second delay while SM-S looked for the dongle.

****

Version 1.0.158.0
2010-06-22

****

1) Canceling a file transfer wasn't being handled correctly in all cases.

2) The AB spec is unclear on what is supposed to happen if a file transfer times out. Previously NAK responses would be sent every 10 seconds until the file transfer completed or a new transfer was started. Now only one NAK will be sent (which matches the AB spec) and if the transfer has not resumed in another 10 seconds it will be canceled.

3) The AB spec indicates that a file transfer NAK response should be sent if a message is received that doesn't match the expected message number. But ABSM seems to be sending the same message multiple times, resulting in file transfer timeouts as it seems to expect that the duplicate messages will be ignored.

   Changed the file transfer receiving code to ignore received packet numbers that are lower than the current expected number. This should perhaps keep ABSM from timing out on large file transfers, and hopefully won't make the file transfer path notably less reliable.

4) A program change on the main MIDI input port will now trigger cues in the show. There should probably be an option to enable this feature (or more to the point disable it) but there doesn't seem to be one in the AB itself. A program change of 0 on channel 1 (the first MIDI channel) is "go next cue". The cue number that will be triggered is a combination of the channel number and the program number. The formula is:

   (channel number - 1) * 128 + program number

5) When a show or sound file is added to or removed from disk, the log message shows both the file number and the highest file number after the load or remove has completed.

6) The total number of disk blocks and disk free blocks is now logged.

7) Changed a number of routines that will hopefully eliminate the problem where SM-D and ABSM will sometimes not see any files on disk when first starting.

8) When the disk path is initialized we now log the disk path name and the total number of show and sound files found.

9) When the controlling application request status message update changes is now logged.

10) Made some changes that may reduce the chance of timing out the

interface to ABSM that can happen now and then.

11) The free space on disk was not being recomputed after a new file had been loaded to disk.

12) The log record for the beginning of a file transfer will now show the size of the file in bytes.

13) Added a delay in sending the file transfer start response. ABSM seems to need this when processing a large file, or it loses the response and then times out the file transfer.

14) Fixed a bug in GO command parsing that incorrectly handled cue numbers in a GO command when they had leading zeros.

15) When SM-A is started minimized the tray menu default will now correctly be the show the window, not to hide it.

16) The Shutdown Server command from the old SM-D was failing to shut down SM-S as it should have due to a command ordering problem.

17) It was possible to see a RESET twice in a row and end up loading the show twice, possibly overlapping. This could occasionally result in failure to load the show correctly. The extra reset path has been eliminated, and the show loader now checks to make sure that it is impossible to try to load two shows at the same time.

18) Under unusual conditions a track would stop playing soon after it was started, rather than at the end of the track as was desired. This should now be fixed.

19) Removing all files on disk with ABSM could result in a communications timeout and ABSM would stop talking. This seems to be fixed now.

****

Version 1.0.157.1
2010-05-19

****

1) The tray popup message could be nearly continuous if not connected to SM-S. (10 seconds out of every 12 displayed). Changed this to only display 10 out of every 30 seconds.

2) Through strange and unknown circumstances the flag that allowed SM-A to respond to the "all call" device id of 127 could somehow get reset. This would result in SM-A ignoring many of the messages in the setup cue. The typical symptom was that submasters would not work until after an MSC RESET command was issued, and not always then.

3) If the submaster zero was other than the default 64, every time you refreshed the show the value of the submaster settings would shift to a new position. This is fixed, and the submasters will now be be zero after refreshing the show.

****

Version 1.0.156.0
2010-05-17

****

1) The submaster assignment maps returned by GET SUBNMASTERS EN was incorrect.

2) Added a configuration option to use only the first 6 out of every 8 channels. This is for Cobranet connections that only use the first 6 channels out of each bundle.

   This requires SM-S version 1.0.59 or later to work correctly.

3) Submaster assignments would not always reload correctly when SM-D sent a show refresh command. This turned out to be a symptom of a bigger problem, which could have resulted in lost messages to SM-S as well as lost internal settings. Show loads are now done off of a dedicated load thread to avoid problems like this.

4) Changed the configuration settings entry method for the root directory and SM-S hostname. The root directory should always be on the same machine as SM-S, as SM-S must share this directory.

   Therefore there is now a directory requestor that will let you pick an existing directory (or create a new directory) on the current machine or any other accessible machine in the network. This replaces the separate text entry fields where you had to manually type in the path to the root directory and the SM-S hostname fields. The "SET" button brings up the directory requestor, and when you click OK, the new path will be shown in the text field.

   This should eliminate some lockup conditions that could happen if the name of a non-existent directory was entered, or if the hostname was entered incorrectly or the SM-S host was not accessible.

5) Corrected the startup ordering to insure that we detect that SM-S is on a remote host at startup and to get the ASIO configuration from that remote host, and also make sure we connect to the remote host even if we have no ASIO devices on the local host.

****

Version 1.0.155.0
2010-04-14

****

1) Now retry the connection between SM-A and SM-S for up to 60 seconds if we get "Connection Refused" responses back. This can happen when starting from a bootup situation and Windows networking isn't ready yet.

2) It can take up to 30 seconds to create the tray icon when starting from a bootup situation. Now retry the tray icon creation for up to 60 seconds. The "Initializing" message will be displayed on the main window while the tray icon is being created.

3) Previously if the tray icon creation failed we bailed out of setup without completing setup, so while things looked like they should be working, nothing worked. Now ignore tray icon creation failure, since SM-A will still work without a tray icon.

4) If a timeout dump occurred during startup the time for the dump was incorrect, resulting in a dump file name of SM-A_V155_0_Dump_Wed_Dec_31_19_00_01_1969.dmp for all dumps. This has been corrected.

5) Interface timeout dumps were being put in the root of the C: drive incorrectly. They should have gone with other dumps in the My Documents\SoundMan Dumps folder.

6) The recent change to reduce the number of commands sent to SM-S to clear the submaster assignments had an error in it. This is fixed.

7) The routine to clear all crosspoint delays was failing to clear the playback crosspoint delays.

8) The routine to clear all crosspoint phase reversals was failing to clear the playback phase reverse.

****

Version 1.0.154.0
2010-04-05

****

1) The "Dongle found" log message was being corrupted. This is fixed.

2) SM-A will now shut down considerably faster when attached to SM-S. Previously on Windows Embedded systems it was slow enough that it could leave the last dialog image on the desktop after it exited.

3) Reduced the number of messages sent to SM-A when a show is loaded. This should eliminate the occasional data overrun that previously could occur.

****

Version 1.0.153.0
2010-03-25

****

1) Improved (slightly) on the blank gray rectangle while SM-A is looking for the dongle during startup.

2) If SM-A was started without the dongle plugged in and then the dongle was plugged in, the dongle would be seen but not read correctly. It is now correctly read after appearing and being recognized

3) Fixed some more bugs with early dongle detection, and was able to change the maximum wait time for a dongle to appear back to 20 seconds. Demo runs without a dongle will start faster.

4) The STOP CHANNEL SET EXT and REALLY STOP CHANNEL SET EXT had a syntax error in the commands they set to SM-S. This is fixed.

****

Version 1.0.152.0
2010-03-23

****

1) A RESUME (Set Resume Point) command with a track number of zero is now interpreted as a CLEAR RESUME command, and a RESUME with a time of 0 but a valid track number is treated as a repeat starting at offset 0 in the specified track. While it is not at all clear from the AB spec that this is how things are supposed to work it appears to match how a real AB actually works.

2) There was a problem with the UDP delay change reports that could have resulted in losing the reports for 3/4 of the output channels. The delay change would have occurred, but it might not have been reported back for output channels 4-15, 20-31, 36-47, and 52-63. Whether or not reports would have been lost would be timing dependent, depending on what other changes were made at nearly the same time.

3) Two new sysex commands have been added to the command set:

STOP CHANNEL SET EXT	SYSEX: 62
REALLY STOP CHANNEL SET EXT	SYSEX: 63

These commands are the complement of GO CHANNEL SET EXT, and can be used to stop multiple playback channels simultaneously.

STOP CHANNEL SET EXT works as a normal STOP command for all of the channels specified in the mask. If the channel DOES NOT have a resume point set, it will stop playing. If a channel HAS a resume point set, the channel will jump immediately to the resume point, just as will happen with a normal STOP command.

REALLY STOP CHANNEL SET EXT is a STOP command that will always stop a playing channel at the current location, even if it has a resume point set. The resume point is not cleared, it is merely ignored. If the channel is later started from the current point with a GO command the resume and stop points previously set will still be in force, and the channel will loop from the stop point to the resume point in the normal manner.

STOP CHANNEL SET EXT   Stop playback on a specified set of channels (SM-A only)

type: sysex
[cmd/data field]: [62H f0 f1 f2 f3 f4 f5 f6 f7 f8 f9]
f0 - f9: channel flags (set to 1 to start playback):
f0 contains flags for playback channels 0 - 6:
f0: bit 0: playback channel 0
f0: bit 1: playback channel 1
f0: bit 2: playback channel 2
f0: bit 3: playback channel 3
f0: bit 4: playback channel 4
f0: bit 5: playback channel 5

f0: bit 6: playback channel 6
f1 contains flags for playback channels 7 - 13
f2 contains flags for playback channels 14 - 20
f3 contains flags for playback channels 21 - 27
f4 contains flags for playback channels 28 - 34
f5 contains flags for playback channels 35 - 41
f6 contains flags for playback channels 42 - 48
f7 contains flags for playback channels 49 - 55
f8 contains flags for playback channels 56 - 62
f9 contains flag for playback channel 63
message length: 19

STOP CHANNEL SET EXT stops synchronous multiple channel playback of single-channel files on SM-A. It is equivalent to a null STOP, except that it can act on an arbitrary subset of playback channels rather than all playing playback channels.

If any of the specified channels have RESUME points set, those channels will not stop. Instead, they will immediately resume playing from the specified resume point. The resume point will not be cleared.

If any channels have stop notification messages set, the stop notifications will be sent, even if the channel immediately resumes at the resume location. The notification messages will not be cleared. This is consistent with the operation of a normal STOP.

REALLY STOP CHANNEL SET EXT Always stop playback on a specified set of channels (SM-A only)

type: sysex
[cmd/data field]: [63H f0 f1 f2 f3 f4 f5 f6 f7 f8 f9]
f0 - f9: channel flags (set to 1 to start playback):
f0 contains flags for playback channels 0 - 6:
f0: bit 0: playback channel 0
f0: bit 1: playback channel 1
f0: bit 2: playback channel 2
f0: bit 3: playback channel 3
f0: bit 4: playback channel 4
f0: bit 5: playback channel 5
f0: bit 6: playback channel 6
f1 contains flags for playback channels 7 - 13
f2 contains flags for playback channels 14 - 20
f3 contains flags for playback channels 21 - 27
f4 contains flags for playback channels 28 - 34
f5 contains flags for playback channels 35 - 41
f6 contains flags for playback channels 42 - 48
f7 contains flags for playback channels 49 - 55
f8 contains flags for playback channels 56 - 62
f9 contains flag for playback channel 63
message length: 19

REALLY STOP CHANNEL SET EXT stops synchronous multiple channel playback of single-channel files on SM-A. The channels will stop playing even if they have resume points set. This command is similar to a null STOP, except that it can act on an arbitrary subset of playback channels rather than all playing playback channels.

If a channel has a resume point set, REALLY STOP CHANNEL SET EXT will ignore the resume point and cause the channel to stop. The resume point is preserved, and if the channel is restarted later, it will jump to the resume point when the stop point is reached or a regular STOP command occurs.

If a channel has a stop notification message set, the stop notification message will be sent when the channel stops. If this is not desirable (such as when stopping all tracks from a control program) the control program must clear the stop notification messages before issuing the REALLY STOP CHANNEL SET EXT command and then restore the stop notification messages afterward.

4) Crosspoint delays were being set in the wrong units in SM-S, and often would be ignored for being too big.

5) Found a problem that could cause some commands to not be sent to SM-S. This was affecting crosspoint delay, but could also affect other settings.

6) If SM-D isn't running, the annoying balloon popup mentioning this fact will only be displayed for 5 minutes, than it will quietly forget all about the problem, on the assumption that the machine is running in a standalone environment.

7) SM-D will wait at startup for up to 40 seconds for a dongle to appear. Without this check if SM-D is auto-started from the Startup folder at machine power up it may come up and run in demo mode before Windows detects that the dongle is present.

****

Version 1.0.151.0
2010-03-18

****

1) There was a timing window in the dongle code that could result in a deadlock on a multiprocessor machine.

2) Trying to copy in a new audio file over one that was set up to play or was playing could fail. The track will now be stopped and the new file copied over it.

3) Disabling a playback channel will now mark the channel as playback state changed so that the new disabled status is reported to UDP.

4) Fixed various places where playback state changed and the change might not have been flagged, or not flagged at exactly the right time.

5) Fixed various places where the stop/resume points for a playback channel changed but the playback_points_en message may not have been flagged to run.

6) The LOAD SETTINGS command now loads the submaster settings. Previously this was disabled for some long-forgotten reason.

7) Loading a setup cue from a show now clears the submaster settings just before the setup cue runs. Previously this clear was missing, possibly resulting in the removal of the submaster loading described in the previous item.

8) With the arrival of test hardware, the input portion of the GPIO functions implemented in v 148 have been verified.

9) Appear to have fixed the problem with SM-D not getting crosspoint gain updates in a timely manner.

10) Several input and output mute changes weren't getting flagged as changes, so might not have triggered automatic updates.

11) The UDP report of delay values for crosspoint channels was wrong, it would only report the values for the first 4 output channels no matter which channel was requested.

12) Bit 2 of the MIDItestResult field in the GET_DISK_STATE_EN response will now show if the GPIO functions are enabled.

13) Completely rewrote the change management functions for the UDP status change notification messages. Hopefully status changes will no longer be getting lost on slow network connections or when multiple clients are in use at once.

14) Deleting an audio selection that was playing (or even loaded on a channel ready to play) would fail. Now it correctly removes.

15) Individual Play Channel commands at nearly the same time could fail if a play command contained a channel number above 15.

16) Play Channel Set Ext failed with channel numbers above 31.

17) Fixed a bug in the V5 show loader that prevented it from loading multiple cuelists correctly.

18) Fixed another bug in the V5 loader that was setting the initial path number incorrectly for all lists but the first.

19) Play Channel Set Ext would fail to play assorted channels above channel 32.

20) The MTC generator was not working correctly and now is.

21) Reduced the amount of time it takes to produce UDP updates. This might reduce the number of "timeout" type messages from SM-D and ABSM.

22) A show control RESET was not clearing the currently running cues, so they would continue to run (and possibly trigger new events) after the reset.

23) A playback ALL-OFF was disabling the playback channels. It seems that the AB spec currently only calls for the channels being stopped, and not disabled. This was probably a spec change at some point that got missed.

24) Realtime cues were not firing when they should. They should be now.

****

Version 1.0.150.1
2010-03-02

****

1) The following documentation applies to features added in version 0.146. The documentation appeared in that version, but with several critical errors in it. It is reproduced below correctly.

   The major errors are that GET_IO_MODE_LEVELS_EN is command 0x7900, not 0x7A00 as previously documented. And the notify bit to trigger this response is bit 21, not bit 20.

2) A new GET_IO_MODE_LEVELS_EN command has been added. This is similar to the GET_IO_LEVELS_EN command, but it returns the input channel mode rather than the analog, digital, and playback source mix levels. As source mix is not implements in SM-A these values were 'fake", and attempted to represent the input channel mode. The new command has the mode directly.

```
typedef struct
{
        mushort code;                                   // 7900H
        mushort length;                                 // 196
        muchar  inputMode           [ 64 ]
        muchar  inputLev            [ 64 ];
        muchar  outputLev           [ 64 ];
} Response;
```

3) The new GET_IO_MODE_LEVELS_EN command can be accessed from the SET AUTO UPDATE EN command. The bit number for the new command is is 20. The list of defined update bit numbers is now:

```
//      The flags longword is defined as follows:
//        bit data returned
//              0  INQUIRY               (INQUIRY_EN)
//              1  STATS                 (GET_STATS_EN)
//              2  DISK STATE            (GET_DISK_STATE_EN)
//              3  VU                    (GET_VU_EN)
//              4  IO GAIN LEVEL         (GET_IO_LEVELS_EN)
//              5  IO DELAY              (GET_IO_DELAY_EN)
//              6  PLAYBACK              (GET_PLAYBACK_EN)
//              7  SHOW                  (GET_SHOW_EN)
//              8  MUTE/SOLO             (GET_MUTE_SOLO_EN)
//              9  SUBMASTER             (GET_SUBMASTER_EN)
//        10  REAL TIME                  (GET_TIME_EN)
//        11  CROSSPOINT LEVEL           (GET_XPT_LEVELS_EN)
//        12  CROSSPOINT DELAY           (GET_XPT_DELAY_EN)
//        13  EQ                         (GET_EQ_EN)
//        14  DIRECTORY                  (GET_DIRECTORIES_EN,
                                         GET_SELECTION_CHANNELS_EN)
//        15  FILE DATA                  (GET_FILEDATA_EN)
                                         (data stops at highestSelNum, highestShNum)
//        16  PLAYBACK POINTS            (GET_PLAYBACK_POINTS_EN)
```

```
//          17  LCD DATA                (GET_LCD_EN)
//          18  COBRANET                (GET_COBRANET_EN)
//          19  MIDI DATA               (GET_MIDI_EN)
//          20  MOTORMIX                (GET_MOTORMIX_EN)
//          21  IO LEVEL & MODE         (GET_IO_MODE_LEVELS_EN)
//          22-30 reserved, set to zero
//          31  SEND ALL DATA
```

Notes:

1 The GET_IO_LEVELS_EN returns analogSrcLev. digitalSrvLev, and playbackSrcLev, which are the AB64 "source mix" levels. SM-A does not do source mixing and does not have command support for digital (Cobranet) IO. Therefore the digitalSrcLev value will always be 00, and the analogSrcLev and playbackSrcLev values will either be 00 or 7F, depending on the current input mode:

mode	analog	playback	notes
disabled	00	00	
input only	7F	00	
playback only	00	7F	
automatic	7F	00	playback stopped
automatic	00	7F	playing

Since it is difficult to tell the automatic mode from the dedicated input and playback modes without looking at the current channel playback state and making some guesses, it is recommended to use the GET_IO_MODE_LEVELS_EN command instead, as it returns the channel mode directly.

2 Currently SM-A does not know sunrise and sunset times as it does not know the machine latitude and longitude. Therefore the 'sunrise' value in the response will always be 0600, and the 'sunset' value in the response will always be 1800. The 'daynight' flag will be 1 (day) between 0600 and 1800, and will be 0 (night) at all other times.

3 The GET_MOTORMIX_EN command has been added for SM-A, it is not valid for an AB64. Normally a MotorMix controller will be connected to the local computer on which SM-A is running. In some cases such as rehearsal conditions the designer's computer may be remote and connected over the network. This periodic update command will contain all of the MIDI packets to be sent to a MotorMix connected to the designer's computer. MIDI data from the MotorMix can be sent to SM-A using the SEND_MOTORMIX_EN command, which is also SM-A-only and not valid for the AB64.

4 The GET_IO_MODE_LEVELS_EN is a replacement for the GET_IO_LEVELS_EN AB64 command. It returns the input channel mode directly rather than the source mix levels.

When asking for automatic updates, only one of GET_IO_MODE_LEVELS_EN and GET_IO_LEVELS_EN should be used. They both use the same change update flags, and only the first will be updated on changes as the changes will be cleared before the second command is processed.

4) To attempt to get around some current limitations in ABShowMaker there is now a new configuration dialog that will let the user manually view and set configuration options that are normally set in the Setup Cue in a show. This new dialog is available from a button in the main configuration dialog.

   The most important settings here are probably the Device Id field and the Save Settings button. All of these settings are nonvolatile or "sticky" state that once set will always be available unless overridden by settings that could be loaded from a show. ABEdit and SM-D tend to load most of these settings in the Setup Cue for every show. ABSM doesn't seem to do this, and doesn't appear to ever issue a Save Settings command, so a box can remain with its virgin device id of 7F, which confuses things greatly.

   ABSM users are recommended to access this new dialog and set the Primary Device Id field to 0 and then Save Settings before attempting to run ABSM. It will probably only be necessary to do this once for any installation of SM-A.

5) It appears that ABSM cannot enable or disable delays, so all delays appear to be zero since delay is disabled by default in SM-A. To get around this a hack has been added to automatically enable the delay on a channel as soon as the delay value is set non-zero.

6) Not all requested status messages were being sent to the controlling application. This was causing problems for the new SM-D.

7) If a show file for some reason fails to open, the log message will now display the full path to the file and the reason for the file open failure.

8) A virgin SM-A will now come up as MSC ID 0 rather than only responding to id 7F as the spec requires. This is more compatible with ABSM, which seems to expect any AB64 that it talks to have a box id of 0 before it makes a connection to it, and will not set and properly save an id value.

9) Show files would sometimes fail to load when they were there and available. This has been fixed.

****

Version 1.0.149.0
2010-02-25

****

1) Eliminated the firmware selection in the config dialog.
Firmware is now fixed at version A451.

2) The delay time reported get_xpt_delay_en was wrong.

3) A crash could happen on a multiprocessor system with a looping GO
cue that just called itself repeatedly.

****

Version 1.0.148.0
2010-02-24

****

1) Changed the configuration dialog to not show the firmware version numbers, only the program names (ABEdit and ABShowMaker) that go with the versions.

2) Changed the timeout NAK for lost messages during a file transfer to 5 seconds from the previous 10 seconds. This may help with some file transfer problems that seem to happen with ABSM.

3) Added logging of timeouts and errors that occur during a file transfer. This may help isolate whatever the problems are with failed ABSM file transfers.

4) The big splash screen for "Controlling app not connected" will no longer be displayed. However the tray popup will still show up in this case.

5) The SM-A "device id" is now shown in the main window to help debug problems with show commands that do not appear to be executing.

6) All delay values are limited to 5.2 seconds individually. There is no longer any limitation on total delay for SM-A other than the total memory available on the system.

7) Should no longer be getting "Controlling app not connected" messages during file transfers and the like. Previously they would happen if the controlling app stopped status polling, even if it was otherwise active talking to SM-A.

8) The SM-A UI would not update correctly if the controlling app was not connected. This wasn't previously visible because the UI was covered with the splash screen message about the controlling app having gone missing.

9) The SET GPO command has been implemented. The SM-A implementation is a superset of the AB64 implementation. The documentation for this command is:

   SET GPO Set General Purpose Output states
     type: MSC AB64-only
     [cmd format byte]: [10H] (sound)
     [cmd/data field]: [06H 00H 1FH oo xx]
     oo: xxH: 00 to 0F for 16 outputs
     xx: 00H: turn output off, 01H: turn output on
   message length: 11

   SET GPO turns on or off one of the sixteen opto-isolated general purpose outputs. The state of the GPO outputs can be retrieved using a GET DISK STATE EN command.

The AB64 only allows two GPO outputs, 0 and 1. SM-A, if GPIO is enabled in the license key and if an Advantech PCI-1730 card and drivers are installed in the PC, has 16 output bits, numbered 0 to 15. Each bit is individually controllable. The PCI-1730 card dip switches must be set to card number 0.

The outputs on the 1730 card are opto-isolated pulldown outputs. Therefore they will pull a signal level to ground when ON and let the signal level float high (usually to 12 or 24V with a user-supplied pullup resistor). Therefore if an output is OFF the voltage on the output will be HIGH, and if an output is ON the voltage on the output will be LOW.

10) The GET DISK STATE EN command has been modified to show the current GPIO input and output state. For the AB64 the ab64_disk_state structure only has one byte for the output state and no input state. SM-A has allocated 4 bytes each for input and output state, allowing a possibility of 32 input bits and 32 output bits. Currently only 16 input and 16 output bits are implemented, and 10 of the input bits have fixed definitions. The remaining bits are currently available for use.

The changed part of the ab64_disk_state structure begins with the byte that previously contained the two GPO output bits and ends with the 24 byte filler at the end of that group of bytes. That section now looks like:

```
muchar  GPOState0;              //0 state of GPO outputs 0 to 7
muchar  autoConfIP;             //X using auto-configured IP address
muchar  slaveDisk;              //0 0: using master/primary disk,
                                //        1: using slave/secondary disk
muchar  MIDItestResult;         //0  bit 0: MIDI port 1 test OK,
                                //         bit 1: MIDI port 2 test OK
muchar  GPIState [  4 ];        //0 state of GPI inputs  0 to 31
muchar  GPOState [  4 ];        //0 state of GPO outputs 0 to 31
muchar  unused4  [ 16 ];        //0 reserved
```

The first byte is unchanged, except that it will show output bits 0 to 7 and not just 0 and 1. The 24 bytes of filler have been reduced to 16 bytes, and 8 bytes of data added where the filler previously started. The first 4 bytes show the status of 32 possible input bits, then 32 possible output bits. If GPIO is not configured in the license key, or the GPIO hardware is not installed all of these bits will read as zero.

If a PCI-1730 card is installed and configured, the first two GPI state bytes will show the current input state, and the first two GPO bytes will show the output state. In addition for backward compatibility, GPOState[0] is duplicated into the GPOState0 byte.

11) The GPI Lockout Time and GPI Trigger Mode values are now reported in the SCSI GET DISK STATE and UDP GET DISK STATE EN responses. Previously the response fields were not set as GPIO was not implemented.

12) The SET GPI command has been implemented for some time, however

since GPIO was not implemented it had no effect other than setting internal values.

The SET GPI command parameters have been changed slightly. One existing value is not useful and has been removed. A new bit has been added to adjust for differences in earlier AudioBoxes.
The current definition for SM-A is:

SET GPI Set General Purpose Input parameters
  type: MSC
  [cmd format byte]: [10H] (sound)
  [cmd/data field]: [06H 09H op dd dd]
   op: trigger options

bit	meaning
0	1 = start playback on playback trigger
1	1 = add cue number offset on playback trigger
2	<unused>
3	<unused>
4	1 = cue number offset is 128 instead of 256

  dd dd: trigger lockout in milliseconds
    (lsb first, 14 bit unsigned value, 7 bits per byte)
message length: 11

Example pp codes:
  01H: start playback on playback trigger
    do not start any cue on playback trigger
  02H: do not start playback on playback trigger
    trigger cue number + 256 on playback trigger
  03H: start playback on playback trigger
    trigger cue number + 256 on playback trigger
  13H: start playback on playback trigger
    trigger cue number + 128 on playback trigger

The default for the op field is 03H.
The default for the trigger lockout time is 1000, which is 1 second.

SET GPI sets parameters for the General Purpose Input GPI contact-closure interface.

The GPI has isolated two trigger inputs, the 'cue' trigger and the 'playback' trigger, and 8 isolated address inputs which may be driven by any contact closure device.

Note that the GPI interface is an option that is usually not present in an SM-A install. On a non-GPI system, GPI parameters may be set and read back without generating an error, but the parameters are not used.

After receiving a valid cue or playback GPI trigger, subsequent triggers of the same type are locked out until the trigger has been inactive for a length of time. This lockout prevents false retriggering from contact bounce. A typical switch or relay contact bounces for approximately 10 milliseconds when it closes. The default trigger lockout is 1 second. With the SET GPI command, the trigger lockout may be set to any value from 0 to 16,383 milliseconds. Note that there are separate lockout timers for each trigger, and both use the same trigger lockout value.

The cue trigger always operates the same way. The cue trigger performs a show control GO on the cue number equal to the binary-coded address lines at the time of the trigger. To insure that the correct cue is triggered, it is a very good idea to set up the cue number lines before setting the cue trigger line. This small delay will allow the cue number lines to debounce and settle before the cue trigger is seen and the cue number is latched internally.

The default operation of the playback trigger is to perform a null playback GO, starting all enabled playback channels, and also perform a show control GO on the cue number equal to the binary-coded address lines at the time of the trigger plus 256 (or 128, depending on the state of bit 4 of the op byte in the SET GPI command).

Using the SET GPI command, the operation of the playback trigger line can be programmed. For the playback trigger, the null playback GO and the cue number offset can be independently enabled or disabled by setting the appropriate playback trigger mode bits. The cue number offset can also be selected between 128 and 256 with the trigger mode bits. (When an offset of 128 is selected only the low 7 bits of the trigger cue number value are used.)

The current values for trigger lockout time and playback trigger mode may be retrieved from SM-A using a GET DISK STATE or GET DISK STATE EN command.

This command updates only the GPI parameter settings in RAM. To save the new GPI parameter settings to disk so that they are restored every time the AudioBox is powered on, follow this command with a SAVE SETTINGS command. When a show is opened containing a setup cue, all GPI parameter settings are reset to their default values.

13) GPIO interface info:

The GPIO interface must be enabled in the license key before it can be used.

Currently ONLY the Advantech PCI-1730 card is supported for digital input and output. More cards may be added in the future.

Advantech IO cards have a dip switch that selects the "card number". The card number MUST be set to the default of 0 for the card to be recognized by SM-A.

The Advantech IO driver software MUST be installed and the card MUST be configured with the Advantech driver configurator before it will work

The PCI-1730 card has 16 inputs and 16 outputs. Two of the low 8 inputs (those that will generate an interrupt) are used as the Cue Trigger and

Playback Trigger lines. The remaining six low input bits are currently unused. The 8 high input bits are used as the Cue Number field that is used with a Cue Trigger or Playback Trigger is sensed.

All 16 of the digital output lines can be controlled by the SET GPO command, which will set any of the bits on or off, one at a time. Currently there is no software interface to set a group of bits to some value all at once.

The current value of both the digital input and output bits can be read with the GET DISK STATE EN command.

Overall our input bit mapping is:

BIT	FUNCTION	PIN			
0	playback trigger	1			
1	cue trigger	20			
2	<unused>	2			
3	<unused>	21			
4	<unused>	3			
5	<unused>	22			
6	<unused>	4			
7	<unused>	23			
8	binary 0	5			
9	binary 1	24			
10	binary 2	6			
11	binary 3	25			
12	binary 4	7			
13	binary 5	26			
14	binary 6	8			
15	binary 7	27			
	GROUND	9	10	28	29

The digital outputs have no assigned function, but it is possible to program a bit on or off using a SET GPO command.

NOTE that the outputs are INVERTING. This means that when the output is OFF the voltage across it will be HIGH, and when the output is ON the voltage across it will be LOW. All outputs are OFF by default when SM-A is first started.

BIT	FUNCTION	
0	out 0	11
1	out 1	30
2	out 2	12
3	out 3	31
4	out 4	13
5	out 5	32
6	out 6	14
7	out 7	33
8	out 8	15
9	out 9	34
10	out 10	16
11	out 11	35

12	out 12	17
13	out 13	36
14	out 14	18
15	out 15	37

14) The input and output delay times reported by GET IO DELAY TIMES EN was incorrect. This has been fixed.

15) Changed the ABShowMaker firmware version from A450 to A451 to match the current AB64 firmware version.

16) Several values have been added to GET DISK STATE EN to report values previously only available with the SCSI GET DISK STATE command. These changes add to the changes in the response described above.

The complete GET DISK STATE EN response now looks like:

```
typedef struct
{
        muchar  diskEnable;          //X playback system is enabled; ready for playback
        muchar  playbackActive;      //X one or more channels are playing back
        muchar  dirError;            //0 error occurred on last directory verify;
                                     //  one or more files were erased
        muchar  transferActive;      //X audio or show file transfer is in progress
        muchar  eraseActive;         //0 audio or show file erasure is in progress
        muchar  defragActive;        //0 disk defragment is in progress
        muchar  formatActive;        //0 disk format operation is in progress
        muchar  dirClrActive;        //? clear directory is in progress (CLEAR DATA)
        muchar  MotorMixActive;      //X MotorMix interface has been activated
        muchar  AllOffLock;          //? ALLOFF lock
        muchar  unused1;             //0 reserved
        muchar  audioMode;           //0 see MULTICHANNEL RECORD&PLAYBACK

        mulong  diskCapacity;        //X total file capacity of the disk in blocks
        mulong  nowfreeBlks;         //X number of blocks now free on the disk
        mulong  freeBlocks;          //X number of blocks free after defragmentation
        mulong  highestSelNum;       //X highest selection number in use
        mulong  defragMarker;        //0 defragment marker (for progress indicator)
        mulong  defragEnd;           //0 defragment endpoint (for progress indicator)
        mulong  highestShNum;        //X highest show number in use

        muchar  GPOState0;           //X state of GPO outputs 0 to 7
        muchar  autoConfIP;          //X using auto-configured IP address
        muchar  slaveDisk;           //0 0: using master/primary disk,
                                     //      1: using slave/secondary disk
        muchar  MIDItestResult;      //0 bit 0: MIDI port 1 test OK,
                                     //      bit 1: MIDI port 2 test OK
        muchar  GPIState [ 4 ];      //X state of GPI inputs  0 to 31
        muchar  GPOState [ 4 ];      //X state of GPO outputs 0 to 31

        mulong  dongle_serial;       //X dongle serial number
        mulong  machine_serial;      //X machine serial number
        mchar   inputs;              //X live input channels
        mchar   outputs;             //X actual output channels
        mchar   playbacks;           //X max playback channels
        mchar   demo;                //X bit 0 = demo mode, bit 1 = dongle present
```

```
        muchar  unused4  [ 4 ];           //0 reserved

        muchar  MIDIID;                   //X MIDI device ID
        muchar  MIDIchannel;              //X assigned MIDI channel (FFH: all channels)
        mushort MIDIgroupIDs;             //? MIDI group IDs assigned (0: assigned)

        muchar  defaultShow;              //X default show number
        muchar  allOffActions;            //X MSC ALL_OFF action flags
        muchar  resetActions1;            //X MSC RESET action flags, byte 1
        muchar  resetActions2;            //X MSC RESET action flags, byte 2
        muchar  submasterGain;            //X current submaster gain setting
        muchar  submasterZero;            //X current submaster zero setting
        muchar  fixedIP;                  //0 set if fixed IP is in use
        muchar  GPIplbkTrigMode;          //X GPI playback trigger mode
        mulong  GPItrigLockout;           //X GPI trigger lockout time in milliseconds

        muchar  unused2[ 32 ];            //0 reserved

}       ab64_disk_state;
```

For each line, a //X indicates that the value is filled in, usually with a value identical to or very similar in meaning to what the AB64 placed in that field.

A //0 indicates that the field is NOT filled in, and the value will be zero. In most cases these fields are meaningless for SM-A.

A //? indicates that the purpose of this field has not been determined and therefore it is not being set.

The following fields have been added near the middle of the structure:

```
        mulong  dongle_serial;            //X dongle serial number
        mulong  machine_serial;           //X machine serial number
        mchar   inputs;                   //X live input channels
        mchar   outputs;                  //X actual output channels
        mchar   playbacks;                //X max playback channels
        mchar   demo;                     //X bit 0 = demo mode, bit 1 = dongle present
```

The dongle_serial is the current running serial number. If a dongle is present this will be the dongle serial number. If the machine does not have a dongle this will be the machine serial number, identical to the next field.

The machine_serial is the serial number of the machine. If it is the same as the dongle_serial, the machine is either unlicensed and running in demo mode (see "demo" below) or is running on a temporary dongle file. If this is different than dongle_serial the machine is running on a dongle, and this is still the serial number the machine would have without the dongle.

Inputs is the number of physical live inputs on the ASIO hardware interface. Outputs is the number of physical outputs on the ASIO interface. Both values may be limited by the license limits in the dongle, or by the demo license limits if there is no dongle or temp dongle file.

Playbacks is the number of licensed playbacks in the dongle, limited to 255 to fit into a byte in case it is larger than that. SM-A can currently not take advantage of more than 64 playbacks.

The Demo byte is bit encoded. Bit 0 is true if the machine is in demo mode. This would indicate that there is no dongle present OR that the dongle data is invalid, AND there is no valid temp dongle file. Bit 1 indicates if the dongle is present. A dongle may be present but not have a license for SM-A/SM-S. In this case the dongle present bit will be set, but the demo bit will be also.

****

Version 1.0.147.0
2010-02-15

****

1) When erasing an audio file the file length was not being set to zero correctly; the file 128 numbers lower would be clobbered instead.

2) There is no longer a "beep" sound when the tray icons for various status messages and warnings are displayed. This can prevent bad sounds coming out of speakers if Windows is sharing the show audio hardware.

3) Made another attempt at preventing timeout dumps while shutting down.

4) EQ changes were not being flagged properly, so ABSM would fail to show any change as it only seems to report changes in state.

5) Not all paths that set a delay value were flagging a delay change. This could have caused ABSM to not report changes in delay settings.

****

Version 1.0.146.0
2010-02-07

****

1) A new GET_IO_MODE_LEVELS_EN command has been added. This is similar to the GET_IO_LEVELS_EN command, but it returns the input channel mode rather than the analog, digital, and playback source mix levels. As source mix is not implements in SM-A these values were 'fake", and attempted to represent the input channle mode. The new command has the mode directly.

```
typedef struct
{
        mushort code;                                           // 7A00H
        mushort length;                                         // 196
        muchar  inputMode               [ 64 ]
        muchar  inputLev                [ 64 ];
        muchar  outputLev               [ 64 ];
} Response;
```

2) The new GET_IO_MODE_LEVELS_EN command can be accessed from the SET AUTO UPDATE EN command. The bit number for the new command is is 20. The list of defined update bit numbers is now:

```
//      bit data returned       command name                    valid for SM-A
//      0  INQUIRY              (INQUIRY_EN)                    yes
//      1  STATS                (GET_STATS_EN)                  yes
//      2  DISK STATE           (GET_DISK_STATE_EN)             yes
//      3  VU                   (GET_VU_EN)                     yes
//      4  IO GAIN LEVEL        (GET_IO_LEVELS_EN)              partially
//      5  IO DELAY             (GET_IO_DELAY_EN)               yes
//      6  PLAYBACK             (GET_PLAYBACK_EN)               yes
//      7  SHOW                 (GET_SHOW_EN)                   yes
//      8  MUTE/SOLO            (GET_MUTE_SOLO_EN)              yes
//      9  SUBMASTER            (GET_SUBMASTER_EN)              yes
//      10  REAL TIME           (GET_TIME_EN)                   partially
//      11  CROSSPOINT LEVEL  (GET_XPT_LEVELS_EN)               yes
//      12  CROSSPOINT DELAY  (GET_XPT_DELAY_EN)                yes
//      13  EQ                  (GET_EQ_EN)                     yes
//      14  DIRECTORY           (GET_DIRECTORIES_EN,            yes
//                               GET_SELECTION_CHANNELS_EN)
//      15  FILE DATA           (GET_FILEDATA_EN)               yes
//      16  PLAYBACK POINTS   (GET_PLAYBACK_POINTS_EN)          yes
//      17  LCD DATA            (GET_LCD_EN)                    no
//      18  COBRANET            (GET_COBRANET_EN)               no
//      19  MOTORMIX            (GET_MOTORMIX_EN)               yes *
//      20  IO MODE & LEVELS  (GET_IO_MODE_LEVELS_EN)           yes *
//      21-30 reserved, set to zero
//      31  SEND ALL DATA
```

Notes:

1 GET_IO_LEVELS_EN returns analogSrcLev. digitalSrvLev, and

playbackSrcLev, which are the AB64 "source mix" levels. SM-A does not do source mixing and does not have command support for digital (Cobranet) IO. Therefore the digitalSrcLev value will always be 00, and the analogSrcLev and playbackSrcLev values will either be 00 or 7F, depending on the current input mode:

mode	analog	playback	notes
disabled	00	00	
input only	7F	00	
playback only	00	7F	
automatic	7F	00	playback stopped
automatic	00	7F	playing

Since it is difficult to tell the automatic mode from the dedicated input and playback modes without looking at the current channel playback state and making some guesses, it is recommended to use the GET_IO_MODE_LEVELS_EN command instead, as it returns the channel mode directly.

2 Currently SM-A does not know sunrise and sunset times as it does not know the machine latitude and longitude. Therefore the 'sunrise' value in the response will always be 0600, and the 'sunset' value in the response will always be 1800. The 'daynight' flag will be 1 (day) between 0600 and 1800, and will be 0 (night) at all other times.

3 The GET_MOTORMIX_EN command has been added for SM-A, it is not valid for an AB64. Normally a MotorMix controller will be connected to the local computer on which SM-A is running. In some cases such as rehearsal conditions the designer's computer may be remote and connected over the network. This periodic update command will contain all of the MIDI packets to be sent to a MotorMix connected to the designer's computer. MIDI data from the MotorMix can be sent to SM-A using the SEND_MOTORMIX_EN command, which is also SM-A-only and not valid for the AB64.

4 The GET_IO_MODE_LEVELS_EN is a replacement for the GET_IO_LEVELS_EN AB64 command. It returns the input channel mode directly rather than the source mix levels.

When asking for automatic updates, only one of GET_IO_MODE_LEVELS_EN and GET_IO_LEVELS_EN should be used. They both use the same change update flags, and only the first will be updated on changes as the changes will be cleared before the second command is processed.

3) Disabled the "TCP received XXX bytes" debug message that was putting unneeded noise into the log file.

4) The input channel input mode has been added to the unused byte at the front of the PlaybackData structure returned by the GET_PLAYBACK_EN command.

5) SM-A can now load Version 5 show files, like those produced by ABShowMaker on the Mac.

6) Made some small improvements to the appearance of the sliders on the debug window.

7) Added a slider to the cuelists display in the debug window. Only eight cuelists are shown, but if a V5 show file is loaded there can be up to 32 open cue lists, requiring scrolling to see them all.

****

Version 1.0.145.0
2010-02-05

****

1) Change the copyright to 2010.

2) Changed things so that all 64 channels are available in SM-S, not just the first 16 as before.

3) The mute/solo state reported to SM-D was incorrect, the output mute and solo values overwrote the input channel values. This is fixed.

4) Setting the unit name over UDP from SM-D was not saving the name, so it would be lost if the user didn't manually update the configuration using the control panel. Also, the command buffer for the SET UNIT NAME EN command was incorrectly defined, losing the first 8 characters of the new unit name.

5) When loading the current show over Ethernet SM-A wasn't checking that it needed to reload the show.

6) When erasing a sound file the highest sound number would often be wrong. This is now fixed.

7) Rewrote the EraseDisk function so that it no longer brings up a DOS window. In addition, it should now be successful in removing the files from the "virtual AudioBox disk" directory.

8) When SM-A was shut down from SM-D, SM-A could end up taking a dump just as it exited. This should now be fixed.

****

Version 1.0.144.0
2009-11-11

****

1) The "SoundMan Dumps" directory will be compressed if it isn't already.

2) SM-A will respond to an external request to take a memory dump. This can be triggered from the "Dumps" menu in SoundMan-Monitor.

3) Fixed a number of minor memory leaks that could occur when loading shows and suchlike.

4) Changed the MIDI data path to only stop the data flowing on the MIDI ports when SM-A is finally shut down. Shutting down data flow when loading a show was proper, but the Edirol UM-1Ex has a problem in the driver where it corrupts memory on a stop, and causes SM-A to crash.

5) The display on the MotorMix will be cleared with SM-A shuts down.

6) The faders on the MM are now updated twice as often, so will be less jerky when following a slow fade.

7) The cuelist info in the debug window was being updated continuously which could cause a crash during show loading. The updating is now suspended during a show load when the cuelists are changing.

8) If a remote host is specified there is now a check during the path setting to be sure the host exists. If not, an error message will be displayed and the path will be set back to 'localhost'.

9) When setting the root path, a check is made if the specified directory exists. It it doesn't a check is made to see if the specified drive exists. If not, and error message will request setting the path to an existing drive.

   If the drive exists but the directory doesn't, the user will be asked if he wants to create the directory. If he answers yes and the directory can be created it will be used.

10) The UI hang checking logic is now turned off when the user is in the configuration dialog. It is often possible to get long responses in the configuration dialog when setting a path to a remote host and the hostname or the pathname are misspelled.

11) When setting the SM-S hostname in the config dialog there is now a check that the host exists on the network.

12) On startup a check is made to be sure the SM-S host is accessible. If it is not the hostname will be set back to localhost, and a warning message will appear.

13) The "dongle serial number" shown in the splash screen is either the

serial number of the dongle (if the dongle is present) or it is the machine serial number if no dongle is present.

14) Holding down the shift key when clicking the Close ("X") gadget will close the program rather than minimizing it to the tray.

15) Minimizing the window the first time would minimize it to the tray rather than the taskbar. This is now fixed, minimize always goes to the taskbar as it should.

16) Now create log messages if we don't receive messages from SM-S in a timely manner.

17) Fixed a problem in waiting for a multiline response that arrives in multiple messages from SM-S.

18) Added code to check periodically for any received data that is present from SM-S but we haven't been notified that the data is there.

****

Version 1.0.143.0
2009-10-18

****

1) Can't use 'localhost' in a network path successfully. It causes all sorts of problems. Also can't use 127.0.0.1 in a network path because that doesn't seem to have permission to write to local files! So go back to the normal path form when we are on the local host.

2) It looks like midiInReset is causing hangs on closing a MIDI port now that the corresponding midiOutReset (which causes memory corruption) has been removed. Try removing the midiInReset also to see if it eliminates the hang problems.

3) Fixed a day one problem in the MIDI data path that could result in losing the fade time off of various commands.

4) Now log the actual cue number(s) fired when a GO is received.

5) Backed out previous erroneous removal of midiInReset. Now I get MIDI crashes again, but it seems to work for everyone else.

6) If the hostname is set to something other than 'localhost' and the remote host cannot be accessed, it will be set back to 'localhost' and a warning message will pop up and be logged.

****

Version 1.0.142.0
2009-10-13

****

1) Removed some old unused dialog templates to simplify code.

2) Get File for audio files now seems to work correctly in ABEdit.

3) Removed the complaint about SM-D being already connected if it is ABEdit connecting with UDP.

4) Added some debug code to print the contents of fade messages to try to track down where fade times are disappearing.

5) Added a hostname to the configuration dialog. The hostname defaults to "localhost", but the user can enter some other PC hostname. This is the name of the computer where SoundMan-Server is expected to be running. SM-S must be started manually on the specified PC if it is not the same PC as SM-A. SM-A cannot start SM-S automatically on a remote PC.

6) Reduced the delay for Get Filedata packets from 10ms to 5 ms. ABEdit still seems to be able to keep up and not loose packets.

7) A memory corruption problem when switching shows has been fixed.

8) If SM-S isn't on the same machine as SM-A, SM-A will attempt to use a network path to access the files for SM-S. This will require that the drive containing the files be shared.

   Also the "format disk" command will not work when a network path is used.

9) A problem with logging "new controller appeared" for IP addresses too frequently has been fixed.

10) Fixed some major problems with flow control in the periodic status message loop. ABEdit seems to be much happier now when getting directory information.

****

Version 1.0.141.0
2009-10-01

****

1) File data changes weren't being marked in all cases, especially in the scsi io paths. This would result in ABEdit not seeing show and sound files uploaded by SM-D.

2) The routines to calculate the machine serial number were not the same in ShowMan, SM-A, SM-S, and the E-Show tools. This release has consistent routines that will produce the same serial number.

****

Version 1.0.140.0
2009-09-23

****

1) Now keep better track of when things on the disk change so that the auto-update stuff for file information returns changes when they actually happen. Before it was possible to not see a new show or selection after loading it to disk.

2) Now stop sending selection updates to auto-update 'get all' at the highest selection number on the disk. Previously sent all updates regardless of whether they contained data. This seems to be closer to how the AB might work, based on one cryptic marginal note in the AB spec. It also seems to make ABEdit a little happier.

3) Now log when LOAD and GO commands for cues are received. This makes it easier to relate log events to a part of the show.

4) Now log the start and end of show and sound file downloads to SM-A.

5) There was an error in the calculation of the machine serial number that could keep temp dongle files from working.

****

Version 1.0.139.0
2009-09-19

****

1) The SCSI-MIDI Passthru flag also controls Ethernet-MIDI passthru. This does not occur in the AB64, or at least it is not documented.

2) The MIDI-SCSI Passthru flag also controls MIDI-Ethernet passthru. This does not occur in the AB64.

3) A crash occurred when clicking OK on the main configuration dialog after setting up MIDI ports. This is fixed.

4) Fixed a problem where the config dialog might not have detected a change in the MIDI port settings correctly.

5) Interface changes from the config dialog are now logged.

6) The success or failure of opening the MIDI ports is now logged.

7) The periodic update abilities of the MIDI QUERY_VU command are not implemented. Only one RESP[ONSE_VU message will be sent back for each QUERY_VU received. Clients using MIDI to get VU values will have to poll. (And ShowMan, the only known MIDI client, does.) VU is also available through the SCSI and UDP interfaces. SCSI clients must poll as there is no automatic response of any sort on the SCSI interface. UDP clients can set up for automatic updates.

8) The Ethernet GET SHOW EN response was limited to once a second. It will now occur quickly if at least one list clock is changing.

9) The unit name field in the configuration dialog is now limited to 31 characters max, as it should have been all along.

10) Implemented the GO_CHANNEL_SET_EXT MIDI command to be able to play more than 16 channels in a group of tracks at once.

****

Version 1.0.138.0
2009-09-16

****

1) The current SM-A implementation is approximately similar to AB firmware version A439, with assorted extensions, limitations, and enhancements. The following notes will cover the vast majority of the differences.

2) Several new Ethernet commands have been implemented: GET MIDI EN, SEND MOTORMIX EN, and GET MOTORMIX EN.

   MIDI commands can be sent to SM-A over the Ethernet port using the SEND MIDI EN command. Each MIDI command is sent in a separate SEND MIDI EN packet. (This is unchanged from the AB64 implementation, save that the command to do this has been explicitly named.

3) SEND MIDI EN — Send a MIDI message to the AB64 or SM-A

short	command		8000H
short	length	n+4	
char	midi_data	[n]	

   Each MIDI message requires its own header, but multiple MIDI messages with their headers can be sent in a single Ethernet packet. If the MIDI message has an odd length (as is often the case) the length field will be odd. However packets must always start on an even boundary, so there must be a single filler byte following the message. This filler byte is NOT included in the length field.

   On the AB64, any response message to the MIDI command will be sent out the first MIDI output port. On SM-A, any response will be sent out the first MIDI output port, and also returned to the Ethernet port where it can be retrieved with a GET MIDI EN command.

4) GET MIDI EN — Get MIDI response data

short	command		8E00H
short	length	n+4	
char	midi_data	[n]	

   GET MIDI EN will retrieve MIDI response messages. Multiple MIDI responses (each in its own message) can be sent in response to a single GET MIDI EN command. MIDI responses can be included in Ethernet packets with other types of messages. If the MIDI response message has an odd length, there will be a single padding byte of 00H following the message and before the next message header. This padding byte is NOT included in the message length.

5) Remote MotorMix implementation. ABEdit can be used at a great distance from the SM-A computer since it communicates over the network. However, to have a remote MotorMix at the same location as the remote ABEdit would traditionally have required a pair of long MIDI cables in addition to the Ethernet cable. To get around

that, SM-A implements SEND MOTORMIX EN and GET MOTORMIX EN commands.

A controlling app can have a MIDI port open to the MotorMix. Any data sent from the MotorMix can be sent to SM-A using SEND MOTORMIX EN. SM-A will generate commands back to the MotorMix. These can be retrieved with GET MOTORMIX EN, and then forwarded to the MotorMix MIDI port. The GET MOTORMIX EN command can be set to automatically forward available data by using SET AUTOUPDATE EN.

The SEND MOTORMIX EN command works identically to SEND MIDI EN. The only differences are the command code and the internal destination of the MIDI data.

SEND MOTORMIX EN Send MotorMix messages to SM-A

short	command		9000H
short	length	n+4	
char	midi_data	[n]	

GET MOTORMIX EN Get MotorMix response data from SM-A

short	command		8F00H
short	length	n+4	
char	midi_data	[n]	

6) Auto-update for GET MIDI EN and GET MOTORMIX EN.
The SET AUTOUPDATE EN command has been extended by adding two previously reserved bits to control automatic periodic responses by GET MIDI EN and GET MOTORMIX EN.

If these bits are set, any time there is available data it will be sent to the controlling application. Any number of MIDI response messages may be sent on any 40ms update cycle.

The current list of automatically updateable commands is:

```
INQUIRY_EN,                 // 0
GET_STATS_EN,               // 1
GET_DISK_STATE_EN,          // 2
GET_VU_EN,                  // 3
GET_IO_LEVELS_EN,           // 4
GET_IO_DELAY_EN,            // 5
GET_PLAYBACK_EN,            // 6
GET_SHOW_EN,                // 7
GET_MUTE_SOLO_EN,           // 8
GET_SUBMASTER_EN,           // 9
GET_TIME_EN,                // 10
GET_XPT_LEVELS_EN,          // 11
GET_XPT_DELAY_EN,           // 12
GET_EQ_EN,                  // 13
GET_DIRECTORIES_EN,         // 14     // also GET_SELECTION_CHANNELS_EN
GET_FILEDATA_EN,            // 15
GET_PLAYBACK_POINTS_EN,     // 16
GET_LCD_EN,                 // 17
GET_COBRANET_EN,            // 18
GET_MIDI_EN,                // 19
GET_MOTORMIX_EN,            // 20
```

The last two entries above, 19 and 20, have been added to provide the MIDI response data.

7) The AB spec documents a flag bit in a MIDI TIMECODE value that is supposed to specify a linear or exponential ramp when the timecode is used as a fade time for a gain change. SM-A ignores this flag bit and picks an appropriate fade shape by itself.

8) SM-A does not implement a number of commands relating to hardware that it does not support. Most notably this includes the 'front panel", Cobranet, and the GPIO features. The following MIDI-formatted commands are not recognized and will be ignored:

    SET GPI
    SET GPO
    PUSH FRONT PANEL BUTTON
    SET COBRANET
    SET PAN
    ENABLE PAN
    DISABLE PANNING
    SET CROSSPOINT ROW EXCL PAN
    SET MTC SOURCE

The following SCSI-formatted commands are ignored:

The following Ethernet-formatted commands are ignored:

    SET COBRANET EN

9) The CLEAR ALL DATA command implements at least one option not available on hardware AudioBoxes. The list is:

    [cmd/data field]: [0CH kk kk kk]
    kk kk kk "key" bytes: the following key byte sequences are defined:
    51H 78H 33H: clear all submaster and controller assigns
    65H 5AH 04H: clear files on disk drive (empties all directories)
    75H 4AH 08H: clear all user data, i.e. reinitialize AudioBox
    to factory settings (see below)
    76H 49H 09H: clear all user data except default show and IP settings
    11H 63H 29H: zero all device statistics
    43H 1BH 68H: zero all input, output and crosspoint delays
    77H 69H 19H: shut down SoundMan-Server and then exit

10) Added a configuration option to select between A439 and A450 firmware. This doesn't change anything except the firmware version response returned through UDP.

11) The MUTE INTPUT CHANNEL command has been extended to take a channel number of 0x7F for "all input channels". If the channel number is 0x7F, then all input channels will either be muted or unmuted, depending on the "mute" parameter value. Adding this makes the MUTE INPUT CHANNEL command match the functionality of the MUTE OUTPUT CHANNEL command.

12) The AB Command Set document is unclear on whether the CLOSE CUE LIST command takes a Show Number as the OPEN CUE LIST command does,

or if it only takes a List number. In SM-A, the CLOSE CUE LIST command only accepts a list number, and any Show number that might be present will be ignored.

****

Version 1.0.137.0
2009-09-14

****

1) Split AB_Interface into the SCSI interface part, and moved all of the AB "guts" into a new class, AB_Container. This separates the SCSI interface from the audio, MIDI, and show engines.

2) Added a new skeleton class, AB64_Interface, which will become the AB64 UDP interface class. It is hooked in in most of the places that the AB_Interface class is, so it should be possible to connect using either scsi or UDP, eventually.

3) Changed the SCSI and UDP paths to make sure that if there is no client monitoring MIDI response messages, the response pipe will be flushed periodically. This will prevent the MIDI system from blocking on a full pipe and stalling processing.

4) A program can now send a message to SM-A to set the Unit Name for the AB emulation. This name is available through the UDP interface when emulating an AB64. To do this, do a SendMessage with the message number of 0x483 and a wParam of 0 and LParam pointing to the name string. The name string must be null terminated.

5) Added the ability to set the unit name in the SM-A configuration panel.

6) Partially implemented SET SOURCE MIX. SM-A will still not mix between live input and playback on a single channel. However it will recognize the command and set internal fields containing the analog and playback levels. The digital (Cobranet) levels are not stored and will always read as zero.

   Various combinations of analog and playback level in set source mix will have the same effect as setting input channel mode to various values.

7) Source mix analog and playback levels will change as playback starts and stops on a channel. In live mode or automatic mode when not playing, the analog input will read 7F and the playback input will read 0. When playing this will be reversed. if the channel is disabled both values will read 0.

8) Control over UDP seems to be basically functional.

****

Version 1.0.136.0
2009-08-31

****

1) Removed an extraneous newline from the "Loading default show" log message.

2) Added experimental code to make resume to another track start the other track rather than looping to the current track only. Since each track will do this independently, absolute sync cannot be guaranteed after the first track decides to loop.

3) Modified code internally to allow SM-A to be built for 64 channels. Modified the MIDI message processing to handle channel encoding as the AB64 claims to do it. (Note this is not sufficient to allow 64 channel playback; there aren't enough MIDI commands that can deal with more than 16 channels.)

4) Updated dongle reading code to not delete a temp dongle file that is valid but is for some other product.

****

Version 1.0.135.0
2009-05-31

****

1) Added extra logging messages when loading the default show to show the number of the show loaded, and whether the default show could be found.

2) Crosspoint phase reverse was not being cleared by a Reset command. The spec has no documentation on whether or not Reset should clear crosspoint phase reversal, but it seems like the right thing to do.

3) Added a hack so that SM-D can create random SM-S SET commands in cues.

4) When SM-Designer is first starting it uses a different window title than normal. This was confusing SM-A, who didn't think that SM-D was running when it really was.

5) Added code to retry a dongle read if it fails, since that seems to happen now and then.

****

Version 1.0.134.0
2009-03-22

****

1) In certain timeout conditions SM-A could end up in a loop burning processor.

2) Cleaned up a couple of shutdown conditions that could result in unnecessary memory dumps.

3) Changed the memory dump code to make larger dumps. The previous dumps often lacked information necessary to find the problems.

4) CCriticalSection doesn't seem to work as it is defined. Replaced all uses with my own critical section wrapper that does work.

5) A long-standing timing window during startup has been fixed. This could cause SM-D to claim that a show refresh was required when just loading the show, even though it would usually load the show correctly. Another symprom was SM-D claiming demo mode when there was a dongle installed.

****

Version 1.0.133.0
2008-12-18

****

1) Updated to new dongle files with some bug fixes.

2) Fixed a problem in list time management that could under strange conditions result in the list time "running away" and SM-A refusing to shut down.

3) Fixed a problem with the dongle configuration not always being read correctly, and occasionally coming up as demo mode when it wasn't.

4) Fixed a timing window on fast x64 systems where SM-D would sometimes claim there was no dongle on startup even though there was one.

****

Version 1.0.132.0
2008-11-05

****

1) If the window is minimized to the taskbar (rather than hidden in the tray) double-clicking the tray icon will now make the window visible. Previously double clicking in this case had no effect and was somewhat frustrating.

2) If the program is started with the window minimized (as would be the case from SM-D) the window will quietly be moved to the tray to clean up the unnecessary icons on the task bar.

3) Added some code from ShowMan to make sure that the cuelist clock time doesn't drift in relation to the wallclock time. Especially on laptops and newer systems with active power management the performance counter can drift relative to the wallclock. The new code will fix that problem.

****

Version 1.0.131.0
2008-10-14

****

1) Improved the accuracy of the time function that produces the log timestamps.

2) The initial log message now includes the version string.

3) Starting and ending of MIDI port linking is now logged. This may make some oddities in log records more obvious.

4) When the Close item is selected in the system menu, or the big X in the upper right corner is clicked, the program will now minimize to the tray. The normal Minimize bar (near the big X) will minimize to the normal taskbar location.

   To exit the program, use either the Exit item in the menu on the system tray icon, ir use the new Exit menu item in the system menu.

5) If SM-D goes away a balloon popup will now occur immediately rather than at some indeterminate future time.

****
Version 1.0.130.0
2008-10-10

****

1) Now log start and end of configuration and About dialog to help see what is happening with occasional problems talking to SM-A.

2) The thread hang checker now properly uses PostMessage to log the fact it took a dump. The SendMessage used previously could have caused it to hang.

3) If no ASIO port has been selected and we are sending commands to SM-S, a balloon popup will warn of the problem.

4) If for some reason a send to the SM-S socket blocks for 5 seconds, SM-A will display a balloon popup message about the problem (as well as logging the problem, as it did previously) and then attempt to close and reopen the socket connection to SM-S. There will be an additional log message (but not a balloon) if this succeeds.

5) SM-A has a registered window message named "SMAConfigRequest". If another program sends this message to SM-A, it will bring up the configuration dialog. There are no parameters required for this message. You should be able to send it by some technique such as

```
UINT msg = RegisterWindowMessage ("SMAConfigRequest");
if (msg != 0)
{
        DWORD recipients = BSM_APPLICATIONS;
        BroadcastSystemMessage (
                BSF_ALLOWSFW | BSF_IGNORECURRENTTASK | BSF_POSTMESSAGE,
                &recipients,
                msg, 0, 0);
}
```

6) An occasional hang could occur when exiting, resulting in a hung program dump. This has been fixed, or at least worked around.

7) Windows messages sent to SM-A from other programs were accidentally being ignored if the window was iconic, as it normally is when running with SM-D.

8) If we get a hang trying to talk to SM-S, an attempt will be made to try to get SM-S to take a dump for us. This may make it possible to figure out what is hanging up on some systems at random times.

9) SM-D can again display the configuration dialog from the "set SM-A config" button in one of its dialogs.

10) There are unsubstantiated reports that SM-D will sometimes go away but SM-A is unaware of this. Made a few code changes that attempt to possibly work around this problem. An actual log or dump showing the problem would be greatly appreciated.

****

Version 1.0.129.0
2008-09-12

****

1) Corrected some problems in the code that creates the dump files if something goes wrong.

2) Found a problem that merges multiple GO messages in a cue into a single GO CHANNEL SET command to insure that the tracks play in unison as intended. The code was not working if the GO commands were not the first sequential commands in the cue.

3) A RESET command would sometimes try to load a completely invalid show number. When this happened it was necessary to restart SM-D before the current show would play again. This is fixed.

4) The version number will now show up in the initial log message.

5) The program would sometimes hang when exiting. This should now be fixed.

6) Log messages now have timestamps accurate to about a millisecond, rather than 10-15mS as previously.

7) There is now a date stamp on messages as well as a time stamp. This helps when a log covers several days.

****
Version 1.0.128.0
2008-08-25

****

1) Added Stop and Start routines to AB_Interface class to make cleanup and restarting simpler.

2) Now stop the interface when shutting down much earlier than before. This may keep us from trying to restart SM-S if it happens to exit before we do when shutting everything down.

3) Log messages now have the time in milliseconds rather than just seconds.

4) SM-A log messages can now be captured by the SoundMan-Monitor logging monitor application.

5) Fixed a problem with correctly detecting if SM-D is running. The name of the main SM-D window changed at some point and SM-A didn't get the necessary corresponding change.

6) Many more internal actions are now logged during startup and shutdown.

7) Added code to automatically clear the debug message log if it gets more than about 65,000 messages in it.

8) Fixed a formatting problem in the About box where the program name was being truncated.

9) Added a top-level exception filter to try to catch any failures in the program and take a dump. This is in addition to the current routine that tries to catch application hangs and take a dump. If I receive dump files when something goes wrong it will be much easier to correct the problems.

10) Added a popup message to the hang detector to note when a dump has been taken and add indicate that it should be sent to me for analysis.

11) If a MotorMix is connected to a separate MIDI interface it is now possible to get the MM fader wiggles out of the main MIDI port. To make this work you must have the Show to MIDI filter turned OFF, and the MIDI-MIDI Passthru turned ON.

12) A long-standing problem with loading one show from inside another show has been fixed. Previously this would fail in most cases, now it should work in all cases.

13) The Clear all crosspoint delays function was not working correctly and now is.

14) It was possible for the RESET command to hang if there were no MIDI ports selected. This has been fixed.

15) Fixed a number of minor inconsistencies in the menus between SM-A and SM-S. Also insured that the version number will show up in the title bar of the main dialog at all times.

****
Version 1.0.127.0
2008-06-21

****

1) Corrected file creation parameters when creating a dump file.

2) Changed the socket code in an attempt to eliminate a hang problem being seen at one installation.

****
Version 1.0.126.0
2008-06-17

****

1) Added code to the SM-D and SM-S communication paths to try to log any unusual conditions that might result in communications hangs with SM-D. Receiving a copy of the SM-A log on an SM-D hang is VERY important!

2) Changed a compile option to put more debug data in the code file. This should help if it is necessary to look at crash dumps.

3) Added a routine to monitor the main display thread to see if it is still operating. If it appears to be hung for 5 seconds or more, a dump file will be written to the root directory of drive C. This dump file will have the general name "SM-A Dump <time>.dmp". The <time> field will be the time that SM-A appeared to hang.

   Please zip up any and all such dump files and send them to me!

****
Version 1.0.125.0

****

1) The running fades list could be manipulated by multiple threads and was not locked. This could eventually result in a crash if we were just starting to manipulate a fade when another thread decided to delete the fade we were working with. There is now a lock around all of the fade manipulation stuff, so this is now single threaded and should be safe.

****
Version 1.0.124.0

****

1) Fixed a problem with GetCrosspointDelay returning 0 for all delays.

2) Fixed a problem with not telling SM-S to clear the delays to zero when a reset occurred that cleared delays.

3) SM-A no longer sets xpoint delay enable based on the value of the xpoint delay. It is now necessary to call ENABLE_CROSSPOINT_DELAY with the appropriate values.

4) Added ANALOGFIRST to the CONFIG SET INTERFACE string sent to SM-S. For some ASIO devices this can result in getting the necessary 16 analog I/O channels up in the first 16 channels, where previously they would have been scattered around.

5) Changed code to directly access the dongle to get demo mode info rather than going through SM-S, but SM-D still starts up showing "demo" in the splash screen, even though it isn't demo mode.

6) Fixed a bug in crosspoint delay calculations in a routine that isn't normally used.

7) Eliminated a few debug messages from the release build.

8) Included new dongle files with more error recovery abilities.

****
Version 1.0.123.0

****

1) Added a tray icon that shows up in place of the normal taskbar icon when SM-A is minimized. The taskbar icon will also show important warning messages in the baloon help area.

2) Updated copyright date in the about box. That seems to have been missed on the prior copyright updates.

3) Removed a number of old unused string resources from the program.

4) If the server fails to connect (because the interface is powered off, for instance) we were telling SM-D that the server didn't start. SM-D didn't like this and would eventually fail to connect on startup. Now we just say it is demo mode and SM-D is happy.

5) Problems with getting "drive not ready" when deleting, replacing, or renumbering an audio file should now be fixed. They had been half fixed some time back, but one important case was mised.

6) The AudioBox has a "Set MTC Source" command that can be used to select whether MTC present on the MIDI input, or the output of the internal MTC generator, will be used as the MTC clock.
This can be different for each cuelist in the AudioBox.

This command is not implemented in SM-A, and will not be. There should only be one "live" source of MTC for all of SM-A, and this can be either an external MTC source on the MIDI input or it can be the output of the internal MTC generator looped back (internally or externally to the MIDI input). There can only be a single MTC source, and it must be the same for all cuelists.

This command has never been present in SM-A, but was previously scheduled for eventual implementation. This is notice that the command will never be implemented.

****
Version 1.0.122.0

****

1) Added timestamps to the log entries.

2) Increased the timeout when opening the interface. Some ASIO drivers can take a fairly long time to get the interface open. If we don't wait long enough then we end up getting the number of input and output channels wrong.

3) Log more of the configuration info when starting up.

****
Version 1.0.121.0

****

1) Corrected internal copyright date to 2008.

2) Removed the legacy options conversion that happened when upgrading from an older version to this one.

****
Version 1.0.120.0

****

1) Added MSC SET commands 0034 and 0035 to set the MotorMix input and MotorMix output ports, respectively. These have exactly the same format as commands 0031 and 0032 that set the MIDI input and output port numbers.

2) Changed the code that reads temporary dongle files to be more reliable.

****
Version 1.0.119.0

****

1) Fixed a crash that could happen on shutdown due to a timing window.

2) Building the masking for calender cues wasn't working completely correctly.

3) Fixed several minor bugs in calander cue logic. Most of these didn't affect how things worked, they just used extra
processor time they didn't need to use.

4) Fixed a missing line terminator when parsing MSC messages to the playback system. This could result in getting incorrect channel numbers on LOAD commands, often resulting in a track not playing.

5) The source and destination id fields were reversed in sysex
messages that we sent out as responses. SM-D ignored these fields, but this caused problems for ShowMan.

6) QUERY_SELECTION responses were not built correctly. This caused ShowMan to not see the responses to a Get Tracks request.

7) RESPONSE_PLAYBACK was missing the channel number field, resulting in
garbage for ShowMan track status.

8) RESPONSE_PLAYBACK messages were trashed when getting responses for all playback channels at once.

9) Stop notify messages work again.

****
Version 1.0.118.0

****

1) The logic to build Go Channel Set from multiple individual GO commands had a couple of bugs that could result in tracks not playing.

2) A show with many lists set to open at show open could open 9 lists rather than the correct number of 8. This is now fixed.

3) It could have been impossible to erase or rename some audio selections if they had been played or were currently playing. Any tracks playing the file to be erased will now be stopped and they will close the file so that it can be erased (or renamed).

4) SM-A will now read the dongle even if SM-S is not running. The title bar and splash screen will report the current allowed configuration however, the number if channels will not be limited by the license limitations.

5) Found there was old code around that would make SM-A die after 30 days. Disabled this, as we now have real licensing info.

6) Removed the configuration variables from the main dialog and moved them to a configuration dialog. There is now a "Configuration" button in the main dialog to access the settings.

7) Configuration variables have been removed from the win.ini file and moved to the appropriate section in the registry. There is code that will move the existing variables to the new locations.

8) There can now be a second pair of MIDI ports that are dedicated to the MotorMix interface. It is also possible still to use the main MIDI ports to talk to the MotorMix. And you can now have main MIDI ports without attempting to check for a MotorMix on the ports or send MotorMix data out the ports.

In the Configuration dialog, you can set the main MIDI input and output ports to None or to some selected port. Under these selections are selections for the MotorMix. You can set the MM to None, Same as Main MIDI Ports, or to individually selected MIDI ports.

While it is possible to have only an input port or only an output port for the main MIDI ports, the MotorMix requires both an input and an output port. If you only set one of the ports to some valid port and don't set the other port to a valid port too, the one that you set will be cleared back to None.

9) SM-A has a registered window message named "SM-AConfigRequest". If another program sends this message to SM-A, it will bring up the configuration dialog. There are no parameters required for this message. You should be able to send it by some technique such as

```
UINT msg = RegisterWindowMessage ("SMAConfigRequest");
if (msg != 0)
{
 BroadcastSystemMessage (
  BSF_IGNORECURRENTTASK | BSF_POSTMESSAGE,
  BSM_APPLICATIONS,
  msg, 0, 0);
}
```

10) Changed the logic so that any MSC or Sysex message received after an All-Off (except for another all-off or a Restore) will unmute the outputs if they were muted by the preceeding all-off. The AB spec gives no hint that this happens or should be done, but apparently it does.

11) Playback all-off was erroneously muting the outputs. It no longer does this.

Also, playback all-off was checking the main all-off actions to decide if it would stop tracks, etc. Now it does all of its actions unconditionally. This should fix reports of problems with the F6 key in SM-D not always stopping all tracks.

(But note that playback all-off does NOT stop list clocks, nor stop cues from firing. Those cues can start new playback immediately after you have used F6 to stop the current playback.)

12) The all-off option flags were not being propagated into the show engine, so an all-off that should have stopped clocks was not doing so.

13) Changed all-off to NOT remove the messages in the "running list" from cues that are currently in progress at the time of the all-off. This means that all-off is no longer ALL off, just MOSTLY off. This might cause trouble in the future, but at the moment it allows you to make a cue that has an all-off followed by a start clock and actually have the start clock message executed. If the running list was cleared (as it should be) the stop-clock message would just disappear into the night and never be executed.

14) Documentation:

Apparenntly on a real AudioBox the Playback Resume command would let you pick a resume point in another sound file, and when the current stop point was hit it would jump seamlessly to the middle of the other file specified in the resume point.

That doesn't happen on SM-A, because neither SM-A nor SM-S have a concept of a jump from the current file to some other file; especially a jump which is set up dynamically after the first track is loaded and playing, as you can do in a real AB.

In SM-A, when a track first starts, any start, stop, or resume point that is in a different track is automatically cleared. Start/stop/resume points that match the current track are retained.

After the track has started you can them set start/stop/resume points for other tracks. However, they will be ignored until such time as a matching track is loaded with a LOAD or GO command. (SM-D sets the stop and resume points for a track just before it loads the track.)

15) Changed the sequencing of the startup tests to see if SM-A is running and connected to eliminate a possible timing problem. There should be less chance of getting into a condition where SM-D says that SM-A isn't running, while SM-A is saying that SM-D is already running and needs to be restarted before it will connect.

16) Setting the submaster zero position on the MotorMix was not working correctly. The faders did not track the new zero position as they should have.

17) Fixed a timing window that could cause crashes on shutdown, at least in the debug build.

18) Fixed a day-one problem that caused incoming MSC messages from the MIDI port to be ignored.

****
Version 1.0.117.0

****

1) Set mute on was sending a blank command to the server and not setting the mute in all cases.

2) Various mute, solo, and similar commands could have been wrongly filtered out before being sent to the server.

3) When SM-A is faking playback because the playback channel is above the number of licensed playback channels in SM-S, the input and output delay settings are ignored as far as the VU display is concerned. The VU will show no delay regardless of the delay setting. Delay will work correctly for licensed channels.

4) When playing back from fake playback channels, crossmutes from other soloed input channels will now be taken into account.

5) When generating VU for fake output channels, crossmutes from other soloed output channels will now be taken into account.

6) Crossmute status for input and output solos is now maintained.

7) SCSI GET SUBMASTER was reporting incorrect masks for the submaster channel assignments, making it look like all channels were assigned to all submasters.

8) Submaster gain per step was computed incorrectly.

9) Checking for valid channels in submaster gain settings wasn't always happening correctly.

10) Submasters assigned to output channels will now work correctly. There may be residual assignments to incorrect input channels that were stored in the persistant data for SM-A. The simplest way to fix this (after loading the new version) is to delete the ConfigData file stored in the SM-A root directory for shows. After doing this and starting SM-D, SM-D will complain that the default show for SM-A does not match the default show set in SM-D. Let SM-D set the default show back to what it should be.

11) RESET was not properly clearing the submaster levels and assignments
in all cases.

12) Changed things so that the submaster gain and zero set by the setup cue are remembered and used on any subsequent RESET command to set the submasters to their 'default level", rather than setting them to the real default level of 64. While the spec doesn't say that this should happen, it seems to come closer to what users might maybe expect; although it is still lacking in a number of ways.

13) The MotorMix ALL-OFF, RESET, and RESTORE functions should now work correctly.

14) GET DISK STATE should now correctly return the MotorMix Active flag set to 1 if the motormix is active.

15) Problems with output VU meters flashing to the top inexpliciably should be fixed.

16) Input and output VU will only be computed from 'fake' values if there are any playback channels playing that don't actually exist on the Server. This generally means that it will only be done for a demo version, since the licensed versions all have 16 playback channels.

17) More problems with channels not always starting when requested have maybe been fixed.

18) Once you have gone into crosspoint mode and exited it, you can hit the crosspoint mode button from input or output mode with no input or output channel selected, and the MM will go back to the last crosspoint that was displayed.

19) When SM-D requested a shutdown, it was possibel to occasionally get a message stating that we can't shut down now because we were still connected to SM-D. This should no longer happen.

20) If a track has a Stop Notify message set, it is usually for the purpose of notifying that a loop has occurred, since there is generally a resume point set also. The AB spec is not clear on all of the cases that can cause a Stop Notify message to be sent. One possible case is if a track is stopped because it is playing when a LOAD of a new file is issued to the track. Since a new file is being loaded to the track, the old stop notify message probably makes little sense. SM-A does not send a Stop Notify message if the track is stopped with a LOAD command for a new file to play. In all other cases it probably behaves as a real AB does; although it is difficult to be sure in all cases due to the lack of rigor in the AB spec.

21) Added a hack to prevent more than 8 lists open at once. If you try to open a cue list with 8 lists already open, the new open request will simply be ignored.

22) A Restore after an all-off now correctly restores playing tracks.

23) Fixed a small bug in the last few test versions that would keep channels from playing if they were higher numbered than the max number of live input channels.

24) Banked mode is now enabled by default on the MM since this makes more sense than having it off. Of course you can still turn it off if you want to.

****
Version 1.0.116.0

****

1) The MM will only be polled every few seconds instead of several times per second. This reduces MIDI traffic.

2) If the MM goes away and comes back, it will now be reset to the state it had when it disappeared, modulo any changes that may have taken place due to running fades and cues.

3) The MM will now always display 16 input and output channels no matter how many channels there are in the physical interface, or if SM-A is running in demo mode.

4) Channel numbers between 1 and 9 on the MM are now displayed without a leading zero. This makes them easier to read.

5) Channel and submaster gains on the MM are now displayed in dB.

6) When you press the Shift key, the other keys that can respond with shifted functions will blink.

7) The MM 2 has some differences from the original MM. The most noticible difference is that the display formatting is off by one character. Hopefully I'll find a way to tell which version I'm talking to rather than requiring the user to enter the MM type. This is with MM 2 firmware version 1.05.

8) Crosspoint gain can now be set using the MM.

Setting a crosspoint is a three step process. First you display the input or output channels. Find the channel that you want to set crosspoints for. If you are displaying input channels, you will be able to set all crosspoints between that single input and all of the outputs. If you are displaying an output channel you will be able to set crosspoints between that output and all of its input channels.

Having located the correct input or output channel, press the Select button so that it lights. Now press the Crosspoint button (third from top in the left column). The display will switch from the input or output display to a display of the crosspoints. So that you know you are displaying crosspoints, the "I" or "O" channel designator will be in lower case.

The input or output channel you are setting crosspoints for will show up in the 7-segment LED display. If you are setting crosspoints for an output channel, the decimal point between the digits will be illuminated and the LCD will display input channels. If you are setting crosspoints for an input channel the decimal point will not be illuminated and you will be setting output channels.

If you select multiple input or output channels and enter crosspoint mode, the lowest numbered channel will be used, and the selects for the higher numbered channels will be cleared.

When in crosspoint mode you can select a new base channel with the main rotary knob rather than going back into input or output mode and selecting a new base channel with the select buttons.

****
Version 1.0.115.0

****

1) There is a new SCSI command GET CROSSPOINT DELAYS to get the current values of the crosspoint delays and crosspoint delay enables.
This is SCSI command number D8H, and has the format:

```
// byte
// num data description
// 0 D8H AudioBox-specific GET EQ operation code
// 1 00H LUN = 0
// 2   00H unused
// 3 00H unused
// 4 00H unused
// 5 00H flag and link fields = 0
//
// The Get Crosspoint Delays command returns a structure showing the
// current crosspoint delay values and crosspoint delay enables.
//
// The returned data structure has 256 crosspoint delay values in
// samples. and 256 bits that will be on if the delay is enabled at
// each crosspoint.
```

The returned structure has the form:

```
struct abxpdelay
{
 ushort inchan;    // first input channel
 ushort outchan;   // first output channel
 ushort enables[16];  // 16 bits of output enables for each input
 ulong delay[16][16];  // 16 output delays for each input [in][out]

};
```

This command returns exactly 1060 bytes of data.

The two channel numbers in the return will always be zero.

The 'enables' flags are 16 short words, one word for each of 16 input channels. The 16 bits in each word represent the delay enable for each of the 16 possible output channels for that input. The delay for an output channel is enabled if 1<<outchan is true.

The delay values are in samples. The maximum delay for any crosspoint is 5200ms, or 249600 samples. There is no cumulative limit on crosspoint delays other than available memory for delay buffers.

2) The GET REAL command has been modified to return the crosspoint delay values using the GET CROSSPOINT DELAYS command described above. The CDB for GET REAL now looks like:

```
// byte
// num    data        description
```

```
// 0     C9H     AudioBox-specific GET REAL operation code
// 1     00H     LUN = 0
// 2     ff1     data request flags
// 3     nl      number of lists: ignored by 1616HD and AB1616 running T228 and lower
// 4     ff2     data request flags
// 5     00H     flag and link fields = 0
 //
// ff1 request flags
// bit   data          size
// 0     VU            32 bytes    see GET VU
// 1     STATS         28 bytes    see GET STATS
// 2     PLAYBACK      128 bytes   see GET PLAYBACK
// 3     SHOW          76+(32*nl) bytes   see GET SHOW (0<=nl<=8)
// 4     LEVELS        288 bytes   see GET LEVELS
// 5     MUTE/SOLO     32 bytes    see GET MUTE/SOLO
// 6     DELAY         132 bytes   see GET DELAY
// 7     SUBMASTER     324 bytes   see GET SUBMASTER
 //
// ff2 request flags
// bit   data          size
// 0     DISK STATE    32 bytes    see GET DISK STATE
// 1     TIME          32 bytes    see GET TIME
// 2     XPDELAY       1060 bytes  see GET CROSSPOINT DELAY
// 3   unused, set to zero
// 4   unused, set to zero
// 5   unused, set to zero
// 6   unused, set to zero
// 7   unused, set to zero
```

Bit 2 of ff2 was previously unused. It will now return crosspoint delay values and delay enables using GET CROSSPOINT DELAYS. This will return an additional 1060 bytes of data on the GET REAL command. This will be the (currently) last data in the response buffer, so the previous data order remains unchanged.

3) The ENABLE CROSSPOINT DELAY command has been implemented. This is an MSC SOUND SET command that will enable or disable the delay for a crosspoint. The delay time must be set separately with the SET CROSSPOINT DELAY system exclusive command.

```
 ENABLE CROSSPOINT DELAY  Enable or disable the crosspoint delay.
   type: MSC
   [cmd format byte]: [10H] (sound)
   [cmd/data field]:  [06H  02H  ii  oo  en]
   ii: input channel number, 00H..0FH for inputs 1 to 16
   oo: output channel number, 00H..0FH for outputs 1 to 16
   en: enable; 00H = disabled, 01H = enabled
   message length: 11
```

4) The AB64 SET CROSSPOINT DELAY command was implemented in version 113. It only applies to channel numbers of 00 to 15 for input and output. Following is the correct documentation for SM-A:

```
SET CROSSPOINT DELAY       Set delay in samples on a crosspoint
  type: sysex
  [cmd/data field]:  [16H ii oo ss ss ss MTC]
```

ii: input channel 0..0FH for inputs 1..16
oo: output channel 0..0FH for outputs 1..16
ss: delay in samples (1/48000 second),
 (lsb first, 21 bit unsigned value, 7 bits per byte)
MTC: ramp time MTC (5 bytes)
message length: 19

SET CROSSPOINT DELAY sets the delay, in samples, on a crosspoint. One sample is one 48000th of a second, or 20.83 microseconds. Smooth delay changes are achieved by pitch shifting the current sample rate over the delay fade time if the pitch shift will be acceptably small. Otherwise the delay is stepped in small increments to the final delay value over the specified time. This can produce ticks in the audio, but they will usually be unnoticed in most theatrical material.

Note that if a SET CROSSPOINT DELAY command arrives for a crosspoint that currently has a delay fade running, the current delay fade will be stopped at the current point, and the new fade to the new target delay will be started. This corresponds to the action that occurs when a new gain fade arrives for a channel that is currently fading the gain.

Each crosspoint can be individually set for any delay value up to 5.2 seconds. There is no cumulative limit on delay values for all crosspoint channels, other than the memory available on the system for delay buffers. If all 256 crosspoints were set to the maximum delay of 5.2 seconds, approximately 130MB of system memory would be required to hold all of the delay buffers.

The AB64 ALLOCATE DELAY command is not needed and is not implemented.

A form of the CLEAR DATA command can be used to set all input, output, and crosspoint delays to zero. See CLEAR DATA.

5) The SCSI RENUMBER FILE command has been implemented. This will allow a show or selection to be renumbered to a new number. The current file must exist, and the new file number must be unused.

This command will NOT change the extended directory information associated with the renamed file. The calling program must use other commands to adjust the directory information to match the new location of the renamed file.

```
// byte
// num   data   description
// 0     C2H    AudioBox-specific SETUP STORE operation code
// 1     00H    LUN = 0
// 2     ff     file type: 00H: audio file, 01H: show file
// 3     oo     old file number MSB
// 4     oo     old file number LSB
// 5     nn     new file number MSB
// 6     nn     new file number LSB
// 7     00H    unused
// 8     00H    unused
// 9     00H    flag and link fields = 0
```

//
// This command will renumber an existing file fomr the current number
// to a new file number. The new file number must be unused when this
// command is issued, and the old file number must exist.
//
// Show numbers must be in the range of 0 to 127 and sound file numbers
// must be in the range of 0 to 8063.
//
// This command only renames the physical file. The control program
// should also change the extended directory information using other
// commands.
//
// The response code returned will be one of:
//
//   01H: renumber completed normally
//   02H: invalid selection/show number (greater than 511/127)
//   03H: source selection/show number does not exist
//   04H: target selection/show number already exists
//   05H disk error

6) The 5.2 second total delay value for inputs and outputs has been removed. In its place, each individual input or output is limited to 5.2 seconds maximum delay. This corresponds to the maximum of 5.2 seconds implemented for crosspoint delays in version 113.

7) The beginings of a MotorMix interface are now available. This MM interface does NOT duplicate that on an AB1616! It provides a number of useful functions currently, and will provide more as time goes on.

All of the important controls are the buttons on each side of the faders. The left buttons select the current operating mode. The right buttons are dedicated to show control functions and do not change with mode.

Most of the buttons have single functions. However, some buttons have a second function when SHIFT is held down. Some buttons will also have a third function when ESCAPE is held down; but these extra functions are not currently implemented.

The buttons on the left, from top to bottom, have the following function meanings:

Button Name	Unshifted Function	Shifted function
AUTO ENBL	Input Mode	none
SUSPEND	Output Mode	none
PLUG IN	Crosspoint Mode	none
WINDOW	none	none
ALL	none	none
DEFAULT	Submaster Mode	Zero all Submasters
UNDO	Show Mode	none

The buttons on the right, from top to bottom, have the following function meanings:

Button Name	Unshifted Function	Shifted Function

PLAY	Start Clock	none
STOP	Stop Clock	Zero Clock
F-FWD	All Off	Reset
REWIND	Restore	none
NEXT	Standby Next Cue	none
PREV	Standby Prev Cue	none
ENTER	Go	Jam-Clock Go

In Input Mode the faders and mute and solo buttons control the input channels directly. The left and right buttons can scroll the input channels one at a time, or if Bank is on, eight at a time. Currently the other buttons in the fader columns do not do anything. The upper row of the display shows the input channel numbers. The lower row of the display shows the MIDI level for the input channels. In the future this will show a level in dB.

In Output Mode the faders and mute and solo buttons control the output channels directly. The left and right buttons can scroll the output channels one at a time, or if Bank is on, eight at a time. Currently the other buttons in the fader columns do not do anything. The upper row of the display shows the output channel numbers. The lower row of the display shows the MIDI level for the output channels. In the future this will show a level in dB.

Crosspoint Mode is largely unimplemented at the moment.

In both Submaster Mode and Show Mode the faders control the submasters. The submaster numbers are shown in the upper row of the display. They can be scrolled left and right in the same way as the input and output channels using th eleft and right buttons and the Bank button.

In Submaster Mode, the lower row of the display shows the submaster levels. In Show mode, the lower line of the display shows the number of the standby cue in the show.

It is not possible to scroll the display to show non-existant channels. If there are more than 8 channels, scrolling to the right will stop when the last channel is displayed on the 8th fader. If there are fewer than 8 channels, the valid channels will appear on the left faders and the rightmost faders will be blank. It will not be possible to scroll in either direction in this case.

Currently the Group button, record function buttons, effects function buttons, and burn buttons, effects select buttons, channel rotary pots, channel select buttons, and the main rotary are unused. Many of these controls will be used in the future when additional capabilities such as setting crosspoint levels, delays, and eq values are enabled.

It is possible to set all submasters to the current zero level by holding down SHIFT while pressing the DEFAULT (Submaster) button. You can change the submaster zero level in either submaster or show mode by holding down the shift button and dragging any fader to the desired zero level. The other faders will track this move since their actual position relative to the zero level is not changing.

8) The AB Command Set documentation for SCSI ERASE FILE is incorrect. It states that the maximum audio selection number it can erase is 511. In fact it can erase any audio selection as well as any show.

9) Fixed a data overrun that could crash SM-A if more than 8 lists were open at once.

****
Version 1.0.114.0

****

1) Paths in shows are now limited to 1MB rather than 32KB. This should make it much rarer to need multiple paths just to get all of the necessary data into a cuelist.

2) Fixed a problem with GetExtendedFileData not reading past the first section of the audio tracks. This resulted in trash information in the selections display window.

****
Version 1.0.113.0

****

1) Changed open cue path to match the number order for the AB rather
than the number order for showman and concurrent.

2) Made similar changes for Close Cue Path.

3) Removed a popup error message that would happen occasionally when
shutting down.

4) Also added some logging that may help in cases where SM-D shuts down but SM-A keeps running

5) The beginnings of crosspoint delay setting is implemented.
Still need to provide a method to query crosspoint delay values,
and invent a new command to enable and disable crosspoint delays.

****
Version 1.0.112.0

****

1) Fixed a problem with a short track stopping as soon as it starts if
it steps on a longer track that was past the end length of the new
track when the new track was started.

2) If you send an MSC command with the command format of 7F (all-types)
it will now correctly select sound format 10 commands. Previously
the all-trpes format was ignored.

3) If SM-A can't talk to SM-S, GetDiskState will return 16/16/16 for the
number of inputs/outputs/playbacks.

Also, the demo byte will have bit 4 set to indicate that the server
isn't talking. This may be a temporary condition during startup,
so it should be retried for a couple seconds.

****
Version 1.0.111.0

****

1) Problems with output VU display at low input gain levels have been fixed.

2) Problems with displaying a low constant output gain level with no input, but with the matrix full on, have been fixed.

3) If you had an auto-fade running and moved the fader to a new position, it would jump back to where it was in the fade and continue fading to the original target level.

It now much more properly forgets about the original fade and moves to the new position.

4) Loading a new show with Open Cue List had a good chance of crashing before. It now works reliably.

5) Fixed a crash that would occur in a show with more than 8 cuelists.

6) Fixed the Close button back so that it works as all UI design manuals state that it will work -- it will close the program. If you want to minimize the window, USE THE MINIMIZE BUTTON, DESIGNED FOR THAT PURPOSE

****
Version 1.0.110.0

****

1) The signs of b0/b1 were wrong on GetAllEq. Fixed now.

2) The AB serial number returned is now the serial number of the dongle if a dongle is present. This is an 8-digit HEX value, not necessarily a pure number like it was on the AB1616.

****
Version 1.0.109.0

****

1) Put in an exceptionally ugly hack to allow SM-D to continue to talk to all 16 channels and think they are live regardless of the actual size of the matrix. This will probably lead to all sorts of confusion when actually working with a small interface.

2) Removed the check in SET INPUT CHANNEL MODE for the actual number of channels on the interface. Also corrected the code so that setting the channel to NONE properly mutes the channel.

3) When stopping playback suddenly VU levels could get 'stuck".

****

Version 1.0.108.0

****

1) Changed GetLevels to report crosspoint phase only by setting the top bit of the crosspoint gain for reversed, rather than making the gain negative as the spec incorrectly states that it should.

2) SM-A now talks to SM-S to find out how many input, output, and playback channels are actually present. It will report these values upward to SM-D in the GET DISK STATE response.

3) The SET CHANNEL MODE command to set a channel to automatic, live, or playback will now check if it is possible to set the channel as requested. If not, the channel will be forced to live, playback, or none, depending on what is available.

4) SM-A should in general not be making requests to SM-S for nonexistant channels now. Previously there were always 16 playback channels so it was not as much work trying to be sure to not talk to missing channels.

5) When running with SM-S, the actual VU readings are taken from SM-S rather than simulated for playback. This will result in live audio levels showing up in the VU meters.

6) Stop points are now detected when they are crossed, not simply when the current position is past the stop point. This will make a stop point at 10 seconds with a resume point at 20 seconds work correctly.

7) Bit 1 of the 'demo' byte in Get Disk State is now 'dongle present'.

8) I still don't know what was causing it, but the problems with EQ display in SM-D seem to be fixed now.

9) Several variants of CLEAR DATA that Tim added over the years that were not previously implemented now are:

51 78 33: Clear all submaster and controller assigns

Clear all submaster and controller assigns will clear all of the controller to submaster assignments, clear all of the submasters to centered, and reset the submaster zero and gain and controller channel values. I don't know if this is exactly what the AB does or not; the spec doesn't give any details of exactly what happens.

11 63 29: Zero all device statistics

Zeroing the device statistics clears the MIDI aand SCSI command and error counts. There are no Ethernet statistics as there were on the AB64.

43 1B 68: Clear all delays to zero

Clearing all delays sets the input and output delays to zero and clears the delay enable on all input and output channels. There are no crosspoint delays to clear a this point as there are on an AB64.

10) The DEASSIGN SUBMASTER and DEASSIGN SUBMASTER IO comands that were implemented along with the AB64 changes have been implemented. It appears that these rather redundant commands are also supposed to be applicable to the AB16.

11) The log window will now expand wider when dropped down, making it easier to read long log lines. If SM-A is too near the right edge of the window the dropdown will be trimmed at the right edge of the screen so that the scroll bar doesn't disappear.

****
Version 1.0.107.0

****

1) A show number of 0 in open cue list now opens a cuelist in the current show as it should.

2) Attempting to open a non-existant show with open cue list no longer deletes the current show before discovering that the target show doesn't exist.

3) Backed out the changes to playback stop made in the last version because they don't match the spec, and the thing just isn't usable with the changes in.

4) The MTC generator now runs. Whether you can lock a list to it is untested.

5) Setting a stop point of 0 will now correctly clear an existing stop point for the track.

6) The main dialog is somewhat narrower so that more data is visible in less space.

7) The position of the dialog is now remembered on close and restored the next time SM-A is opened.

8) The icon has been changed from a readable but inappropriate "AB" to a less readable but more appropriate "SM-A". It may get changed again in the future to something more readable.

9) GetLevels now returns negative gains for crosspoints with inverted phase. Note that negative zero looks just like plus zero, so you can't tell if a channel is inverted if the gain is zero!

****

Version 1.0.106.0

****

1) Moved the SET MIDI RECEIVE command from 1DH to 30H. Tim defined the 1DH command slot for the AB64, and using it for two different commands would not have worked well in the long run.

2) Added four new commands to let SM-D set up the persistent state for SM-A. These commands are:

   SET MIDI INPUT PORT
   SET MIDI OUTPUT PORT
   SET ASIO PORT
   SET DISK PATH

   The first three are sound format 10 SET commands. The last is a sysex command.

3) The close button in the window would shut down SM-A when it was clicked, even if SM-A was in use. It should have asked first. Now it will simply minimize the window without asking.

4) When a stop point is reached on a playing track the stop point is now cleared, or more specifically reset to be the end of the track.

5) When a track stops because it reaches a stop point the track becomes disabled rather than simply stopped.

****
Version 1.0.105.0

****

1) Implemented SET MIDI RECEIVE to set or clear the MIDI input to SCSI port passthru flag. This allows SM-D to capture from the MIDI port. In the process, removed this function from SET MIDI ECHO and from the show file header. The MIDI to SCSI passthru will be turned off when a show is loaded.

2) When the sequence clock is set to cuelist, it just means that the sequence clock is the same TYPE as the cuelist clock: stopwatch or MTC. It no longer means that if the cuelist clock is stopped the sequence clock is also stopped.

   This is different than ShowMan, but apparently compatible with how the AB1616 works. (The AB spec is moot on this topic.)

3) A GO with a track position on it did not correctly tell SM-S to position the track to that location. leading to things playing from places where they weren't supposed to.

****
Version 1.0.104.0

****

1) Changed to opening 127.0.0.1 explicitly instead of "localhost" since Vista doesn't seem capable of resolving the local host by name.

2) Added a Copy button to copy the log to the clipboard, making it easier to paste into a text file or mail message.

3) Added a number of progress messages in setting up the connection to SoundMan-Server. These messages will show up in the log, and will hopefully help indicate where problems might occur.

4) Some minor internal cleanup to make various forms of the same command to the server more consistent in form.

5) Starting tracks from a position other than the start of the file wouldn't always work, depending on the command sequence used to set up the file.

6) Changed RESET to use a SET MATRIX command to reset the gains, rather than sending individual commands to each possible channel.

7) Fixed a problem where Clear All Eq sent an invalid message to SM-S.

8) Changed RESET to send playback clears much more efficiently.

9) Cleaned up a track start command that was occasionally duplicating text in the command.

10) The SM-D Fast Forward command now works reasonably.

11) Changed the win.ini section name from SoundMan_AB to SoundMan_Assistant to make Charlie less stressed.

****
Version 1.0.103.0

****

1) The SEND MIDI command set the wrong bit on the front of the message to be sent, resulting in invalid MIDI messages. This is fixed.

   NOTE that SEND MIDI commands will only be recognized if sent from the show or from the matrix (a stop-notify message). They will NOT be recognized from the SCSI SEND MESSAGE command, nor will they be recognized if they appear on the MIDI In port.

   It is not clear from the AB spec under what conditions the SEND MIDI message will be recognized.

2) Implemented the Stop_Notify-Midi-Filter bit in the SET MIDI ECHO command and show file header.

3) Implemented a MIDI to SCSI ECHO bit in the SET MIDI ECHO command and show file header. This is bit 3 of the ss parameter in the SET MIDI ECHO command and bit 4 in the midi echo byte of the show file header.

4) Return the values of the stop-notify-midi-filter in the Response Show State command. These values are not defined to be returned in the AB command spec, but it appears that section is out of date.

5) Return the same flag information in the SCSI GET SHOW command response. Again, Tim's spec doesn't document these values.

6) Implemented the Stop_Notify->midi filter function. As implemented this simply stops all messages from the matrix to the midi out port. This should only be stop-notify messages, as far as I can remember.

7) If the new midi-to-scsi echo bit is set, the MIDI In data will be passed to the SCSI port where it can be received with GET RESPONSE SCSI messages.

   NOTE that there is no limit on the size of a MIDI message that can be received! The limit of 120 characters in a real AB does not apply here. If the receive buffer is not large enough, some of a large message may be lost.

   ALSO NOTE that if messages are being received on the MIDI port and GET RESPONSE messages are not issued in a timely manner, messages waiting for a GET RESPONSE command will be discarded. These may be messages from the MIDI In port, or may be messages from the show engine or other places.

8) There are several AB commands that are not implemented, and won't be as there is no need for them. This serves to document these commands:

SET GPI
SET GPO
PUSH FRONT PANEL BUTTON
SET COBRANET
SELECT DRIVE
COPY DRIVE
DISABLE PANNING
SET CROSSPOINT ROW EXCL PAN
SET PAN
SET SOURCE MIX
ALLOCATE DELAY
DEFRAGMENT DRIVE

If any of these commands are received they will be ignored.

9) The ClockBox is not recognized, and won't be. However, the capabilities for recognizing sunrise/sunset will be incorporated using the PC system clock in a future release.

10) If an assortment of GO commands to start tracks appear in a single cue near each other, they will be internally converted to LOAD commands and a GO CHANNEL SET command generated after the last such GO command. This should result in synchronous GO action. It seems the AB had strange code to cause this to happen so this is an attempt to get the same results.

****
Version 1.0.102.0

****

1) The SCSI inquiry string still identified the product as an AB with serial number 123.

2) There is a new hack version of CLEAR DATA that will shut down SM-A and SM-S.

3) Fixed a bug so that SM-S will shot down even if there is no ASIO port open currently.

4) Negated the last two eq param values to match how the AB does a biquad to how the server does a biquad.

5) Fixed problems with 'forgetting' the input mute state when starting or stopping track playback.

****
Version 1.0.101.0

****

1) Change "mode abemulation on" to "mode AudioBox' to match Server.

2) Change "SoundMan-ager" to "Soundman-Designer".

****
Version 1.0.100.0

****

1) Changed product name to SoundMan Assistant since AudioBoxes are now dead and gone and shouldn't be mentioned any more.

2) Changed references to ABEdit to SoundMan[ager], with the [ager] part in italics

3) Updated copyright from 2005 to 2007.

****
Version 1.0.97.0

****

1) Completely unknown, far too long ago. Probably never released.

****
Version 1.0.96.0

****

1) The VU meters would stop responding if a track was stopped and restarted.

2) Added code so that SoundMan-AB can continue to work in emulation mode even if the server is open, but the ASIO interface is closed.

3) Added code to not start the server if there are no ASIO devices on the system.

4) If not connected to ABEdit and not connected to the server, we will now exit when the Close button is clicked without asking if you really want to exit. If connected to the server, we still need a confirmation that you really wanted to exit.

5) Fixed several places where the program name was displayed as SoundMan_AB instead of SoundMan-AB.

6) Added a simple but real installer for SoundMan-AB.

****
Version 1.0.95.0

****

1) Yet Again fixed the problem of not starting a track from a GO command and requiring a LOAD command first. This time it works.

2) Fixed an annoying problem where a track would stop as soon as you tried to start it using a GO, or possibly a cue in ABEdit that had both a LOAD followed immediately by a GO.

3) Added a LARGE warning message when ABEdit isn't connected, so it will be fairly obvious why things aren't working.

4) Added a LARGE message if ABEdit is running when we are started, warning that it will fail to connect to us until it is restarted.

****
Version 1.0.94.0

****

1) Fixed a crash that happened in stand-alone mode without a server.

2) Fixed a problem where a GO to a single channel was being ignored and leaving the channel disabled.

****
Version 1.0.93.0

****

1) Fixed a crash that could occur if the server went away and came back.

****

Version 1.0.91.0

****

1) Fixed a problem with a Null GO command not correctly starting all ready channels.

2) Fixed a problem with track setup not properly clearing stop and resume points.

3) Changed code to set a stop point at the end of the track implicitly on a GO or LOAD if no 'official' stop point is set. This will cause a track with a resume point set but no stop point set to loop the entire track. This matches how the AB actually works, although the command set documentation seems to contradict this.

****
Version 1.0.90.0

****

1) Fixed a problem with LOAD on a track actually doing a GO.

2) Gain, delay, and other parameters were only getting set for playback channels and not real input channels or input crosspoints.

3) No longer reload track files on a GO after they were loaded with a LOAD.

4) When playing without real sound output and you stop in the middle of the track, the vu indicators will now go to zero instead of staying stuck at the vu level of that part of the file. This already worked when playing with sound playback.

5) Submasters are now implemented and will control the gain correctly when playing back with sound. There is no visible effect when not playing back with actual sound.

6) MIDI controller values will now correctly set submaster gains.

****
Version 1.0.89.0

****

1) No longer send MUTE or UNMUTE commands to nonexistent input channel numbers when starting or stopping tracks and the physical interface in use has fewer input channels than we have playback channels.

2) Will no longer erroneously set stop or resume points for another track number while a different track is currently playing and end up confusing the server about where to stop.

   Also will no longer carry over stop or resume points from a previous selection to the current playback track. This showed up as stopping a new track at the time the previous track had ended.

3) Cleaned up the functioning of LOAD, GO and GO CHANNEL SET to exactly match the manual on how these commands deal with channels that are disabled, stopped, or currently playing. There were some minor errors in the previous implementation.

4) Now validate on a Setup Store that the file number is in the right range for the type of file.

5) Go Channel Set would have started disabled channels if they were selected in the channel mask. Now it will correctly only start channels that are marked as 'stopped'.

****
Version 1.0.88.0

****

1) SoundMan-AB no longer opens ASIO devices, it asks the sound server for the device configuration information.

2) If the sound server is already running and has a device attached when SoundMan-AB starts, SoundMan-AB won't stop and reload the current device. This keeps it from clobbering any tracks that might currently be playing.

3) Fixed problem with crashing when reopening the socket to the server after the server had gone away and come back.

4) Fixed a problem with disabling a playing channel not stopping it.

5) Fixed a problem in initial startup if no ASIO device has been selected in the droplist.

****
Version 1.0.87.0

****

1) Corrected the way that good/bad/total scsi messages are counted to match how the real AB does it.

2) First version that supports SoundMan Server.

****
Version 1.0.86.0

****

1) Set Input Channel Mode no longer corrupts Playback State for the input channel.

2) Mute All Outputs (F5) now works correctly. Previously it was being ignored. (Another spec change I didn't know about.)

3) Likewise for Set Delay to all channels.

4) Likewise for Query Playback to all channels.

****
Version 1.0.85.0

****

1) A crash occurred when trigging a Stop Notify message. This is fixed.

2) Cleaned up some incorrect interactions between Stop Notify and Resume on a playback channel.

3) Fixed a slight delay in looping a track with Resume.

4) Fixed a problem with Reset not stopping a track that had a resume point set.

****
Version 1.0.84.0

****

1) The Resume command now works to make tracks loop. The resume point was not being checked on a Stop, so the track would just stop.

2) The splash screen has been prettied up a bit.

3) The About box has been prettied up, contains RSD contact info, and automatically loads the correct version string.

4) The version number in the main window title is now loaded automatically from the program version resource. No more cases of not having it match correctly.

5) The welcome message and MIDI port warning message now wait until the splash screen goes away before they display.

6) The playback channel positions no longer flash randomly when they aren't changing.

7) Fixed a problem where doing a LOAD on a track that wasn't on the disk would hang up ABEdit.

8) Fixed several small memory leaks that only occurred at program termination.

9) There is now a splash screen on startup.

****
Version 1.0.83.0

****

1) The actual version number matches the version number in the main window.

2) Removed the demo timeout logic that had disabled the last version 30 days after the compile.

3) Changed the default submaster level setting to 64 from 0.

4) Fixed the Get Submaster response to correctly invert the returned bit values.

5) Changed to GO MSC command to work normally and ignore a GO to a closed cue list. This should match the changes in the latest AB firmware.

****
Version 1.0.82.0

****

1) Calendar cues will now only fire once per possible time window, rather than continuously during the window. This now matches the was the real AB does it.

2) Fixed problem with playback GO command not loading the correct track position when starting playback.

****
Version 1.0.81.0

****

1) Added code to save the debug log to a file.

2) GetDelay returns correct delay in samples values.

3) SetDelayInSamples sets output delays correctly.

4) SetEqCoefficients sets the ABEdit symbolic info if present.

5) SetEqBands sets the individual band enables in the Shelly info.

6) GetAllEq produces the results in the right places.

7) The internal eq info array is initialized so it doesn't contain garbage.

8) If you stop a selection and then go the selection, it will resume from where it was stopped instead of starting over.

9) The cue/review/previous buttons in the ABEdit playback window seem to work correctly.

10) Test Unit Ready messages are no longer logged to the message log.

****
Version 1.0.80.0

****

1) No longer get interface timeouts in ABEdit when trying to display current playback state.

2) If you open a list and don't fire a cue it no longer shows the current path for the list as -1 (unless the list has no paths).

3) Opening a list could fail and leave the list disabled. Now fixed.

****
Version 1.0.77.0

****

1) Made the display of number of input and output channels for the ASIO hardware work correctly.

2) Added a check to complain if you are trying to run on Win98, since CreateNamedPipe doesn't exist on Win98!

3) Added detection to automatically reload the show if the current show gets overwritten on disk. This seems to make the Refresh button work the way Troy expects.

4) Implemented the use of reset-actions in Reset and all-off-actions in AllOff. This will result in reloading the show (or the default show) in Reset, rather than just resetting to the top of the current show.

5) Made all-off immediately after loading the show work correctly.

6) This version of SoundMan-AB expires after 30 days.

****
Version 1.0.76.0

****

1) Changed things to separate the path number from the path index and allow arbitrary (but ascending) path numbers. This seems to make ABEdit happier.

2) Eliminated the debug 'load show' window on the main window.

3) Fixed GO CHANNEL SET decoding

4) Made a restart after a stop work correctly

5) Implemented VU meters simply. They ignore mutes and solos.

6) Changed pipe handling to allow up to 16 simultaneous pipe opens.

7) Now stop playback when downloading files to the box.

8) GetDiskState now returns 'playback active' flag.

9) First attempt at an icon.

****
Version 1.0.75.0

****

1) No longer give the generic 'do setup first' message if the directory is already set up but midi isn't. Instead there is a second message suggesting that midi might need to be set up.

2) No longer open input and output midi port # 0 by accident if the midi ports haven't been set up.

3) GetShow now returns path numbers 1 relative rather than 0 relative. This might make ABEdit a bit happier that things are really working right. I don't have a handy show to test with though at the moment.

****
Version 1.0.74.0

****

1) Added support for continuous controllers. Untested, but it should work.

2) Will not refuse to exit if connected to ABEdit.
Also asks if you really want to exit otherwise to prevent mistakes.

3) Should no longer crash when loading a new show over a show with a running realtime list.

4) Fixed some bugs in GetFile that resulted in trashed file transfers.

5) Fixed decoding of abrealtime field to set RT cue times right.

6) Better formatted the display of realtime cue times in debug display.

7) Added an icon, of sorts. Need to do better.

8) Implemented timed fades. You will now see faders move at about the right speed.

****
Version 1.0.73.0

****

1) First reasonably full-featured release version.

****
Version 1.0.4.0

****

1) Finished all of the scsi commands.

2) Redid the midi parser layout so it thinks it is an AB rather than an AB driver. Still need to add many commands.