

Changes in DarwinStreamingServer/trunk [90:743]

Location: DarwinStreamingServer/trunk

Files: ■ 8 added ■ 9 modified

- streamingserver.xml (1 diff)
- WebAdmin/src/streamingadminserver.pl (3 diffs)
- Install.orig
- Install (6 diffs)
- Makefile.POSIX (3 diffs)
- Install.rick
- APIStubLib/QTSS.h (1 diff)
- APICommonCode/QTAccessFile.cpp (1 diff)
- APIModules/QTSSAccessModule/ModuSessionMysql.cpp
- APIModules/QTSSAccessModule/ShibMysql.h
- APIModules/QTSSAccessModule/QTSSAccessModule.cpp (5 diffs)
- APIModules/QTSSAccessModule/ModuSessionMysql.h
- APIModules/QTSSAccessModule/ShibMysql.cpp
- APIModules/QTSSRefMovieModule/QTSSRefMovieModule
- APIModules/QTSSHomeDirectoryModule/QTSSHomeDirectoryModule
- Server.tproj/QTSServerPrefs.cpp (2 diffs)
- Server.tproj/RTSPSession.cpp (1 diff)

☐ Unmodified ☒ Added ☐ Removed

DarwinStreamingServer/trunk/streamingserver.xml

Tabular | Unified

r90	r95	
528	528	<PREF NAME="modAccess_qtaccessfilename">qtaccess</PREF>
529	529	
	530	<!-- added for shib -->
	531	<PREF NAME="shibmysqlSocket" >/tmp/mysql4.sock</PREF>
	532	<PREF NAME="shibmysqlName" >shib_sessions</PREF>
	533	<PREF NAME="shibmysqlUser" >darwin</PREF>
	534	<PREF NAME="shibmysqlPassword" >shib</PREF>
	535	
	536	<!-- added for modu -->
	537	<PREF NAME="moduMysqlSocket" >/tmp/mysql4.sock</PREF>
	538	<PREF NAME="moduMysqlName" >shib_sessions</PREF>
	539	<PREF NAME="moduMysqlUser" >darwin</PREF>
	540	<PREF NAME="moduMysqlPassword" >shib</PREF>
	541	
530	542	<!-- The default path and file name for the AccessModule's groups list -->
531	543	<PREF
		NAME="modAccess_groupsfilepath">/Library/QuickTimeStreaming/Config/qtgroups</PREF>

DarwinStreamingServer/trunk/WebAdmin/src/streamingadminserver.pl

Tabular | Unified

r90	r612	
78	78	}
79	79	else {
80		\$defaultConfigPath = "/etc/streaming/streamingadminserver.conf";
	80	\$defaultConfigPath = "/usr/local/DarwinStreamingServer-5.5.5-
		dram2/etc/streaming/streamingadminserver.conf";
81	81	}
82	82	
...	...	

229	229	%vital = ("port", 1220,
230	230	"sslport", 1240,
231		"root", "/var/streaming/AdminHtml",
232		"plroot", "/var/streaming/playlists/",
	231	"root", "/usr/local/DarwinStreamingServer-5.5.5-dram2/var/WebAdminHtml",
	232	"plroot", "/usr/local/DarwinStreamingServer-5.5.5-dram2/var/playlists/",
233	233	"server", "DSS 5.5 Admin Server/1.0",
234	234	"index_docs", "index.html parse_xml.cgi index.htm index.cgi",
...	...	
245	245	"qtssIPAddress", "localhost",
246	246	"qtssPort", "554",
247		"qtssName", "/usr/local/sbin/DarwinStreamingServer",
	247	"qtssName", "/usr/local/DarwinStreamingServer-5.5.5-dram2/sbin/DarwinStreamingServer",
248	248	"qtssAutoStart", "1",
249		"logfile", "/var/streaming/logs/streamingadminserver.log",
	249	"logfile", "/var/log/darwin/streamingadminserver.log",
250	250	"log", "1",
251	251	"logclear", "0",
252	252	"logtime", "168",
253		"messagesfile", "messages",
	253	"messagesfile", "messages",
	254	"logfile", "/var/log/darwin/streamingadminserver-messages.log",
254	255	"gbrowse", "0",
255	256	"ssl", "0",
256		"crtfile", "/etc/streaming/streamingadminserver.pem",
257		"keyfile", "/etc/streaming/streamingadminserver.pem",
	257	"crtfile", "/usr/local/DarwinStreamingServer-5.5.5-dram2/etc/streamingadminserver.pem",
	258	"keyfile", "/usr/local/DarwinStreamingServer-5.5.5-dram2/etc/streamingadminserver.pem",
258	259	#"keypasswordfile", "",
259		"qtssQTPasswd", "/usr/local/bin/qtpasswd",
260		"qtssPlaylistBroadcaster", "/usr/local/bin/PlaylistBroadcaster",
261		"qtssMP3Broadcaster", "/usr/local/bin/MP3Broadcaster",
	260	"qtssQTPasswd", "/usr/local/DarwinStreamingServer-5.5.5-dram2/bin/qtpasswd",
	261	"qtssPlaylistBroadcaster", "/usr/local/DarwinStreamingServer-5.5.5-dram2/bin/PlaylistBroadcaster",
	262	"qtssMP3Broadcaster", "/usr/local/DarwinStreamingServer-5.5.5-dram2/bin/MP3Broadcaster",
262	263	"helpurl",
		"http://helpqt.apple.com/dssWebAdminHelpR3/dssWebAdmin.help/DSSHelp.htm",
263	264	"tempfileloc", "/tmp",

DarwinStreamingServer/trunk/Install

Tabular | Unified

r90	r170	
4	4	# source release
5	5	
	6	DSSROOT=/usr/local/DarwinStreamingServer-5.5.5-dram2
6	7	
7	8	echo;echo Installing Darwin Streaming Server;echo
...	...	
36	37	
37	38	echo Removing previous versions of Darwin Streaming Server
38		if [-f /usr/local/sbin/DarwinStreamingServer]; then
39		echo removing /usr/local/sbin/DarwinStreamingServer
40		rm -f /usr/local/sbin/DarwinStreamingServer

41	fi
42	
43	if [-f /usr/local/bin/PlaylistBroadcaster]; then
44	echo removing /usr/local/bin/PlaylistBroadcaster
45	rm -f /usr/local/bin/PlaylistBroadcaster
46	fi
47	
48	if [-f /usr/local/bin/MP3Broadcaster]; then
49	echo removing /usr/local/bin/MP3Broadcaster
50	rm -f /usr/local/bin/MP3Broadcaster
51	fi
52	
53	if [-f /usr/local/bin/qtpasswd]; then
54	echo removing /usr/local/bin/qtpasswd
55	rm -f /usr/local/bin/qtpasswd
56	fi
57	
58	if [-f /usr/local/sbin/streamingadminserver.pl]; then
59	echo removing /usr/local/sbin/streamingadminserver.pl
60	rm -f /usr/local/sbin/streamingadminserver.pl
61	fi
62	
63	if [-f /usr/local/bin/streamingadminserver.pl]; then
64	echo removing /usr/local/bin/streamingadminserver.pl
65	rm -f /usr/local/bin/streamingadminserver.pl
66	fi
67	
68	if [-f /usr/local/bin/SpamPro]; then
69	echo removing /usr/local/bin/SpamPro
70	rm -f /usr/local/bin/SpamPro
39	if [-f \$DSSROOT/sbin/DarwinStreamingServer]; then
40	echo removing \$DSSROOT/sbin/DarwinStreamingServer
41	rm -f \$DSSROOT/sbin/DarwinStreamingServer
42	fi
43	
44	if [-f \$DSSROOT/bin/PlaylistBroadcaster]; then
45	echo removing \$DSSROOT/bin/PlaylistBroadcaster
46	rm -f \$DSSROOT/bin/PlaylistBroadcaster
47	fi
48	
49	if [-f \$DSSROOT/bin/MP3Broadcaster]; then
50	echo removing \$DSSROOT/bin/MP3Broadcaster
51	rm -f \$DSSROOT/bin/MP3Broadcaster
52	fi
53	
54	if [-f \$DSSROOT/bin/qtpasswd]; then
55	echo removing \$DSSROOT/bin/qtpasswd
56	rm -f \$DSSROOT/bin/qtpasswd
57	fi
58	
59	if [-f \$DSSROOT/sbin/streamingadminserver.pl]; then
60	echo removing \$DSSROOT/sbin/streamingadminserver.pl
61	rm -f \$DSSROOT/sbin/streamingadminserver.pl
62	fi
63	
64	if [-f \$DSSROOT/bin/streamingadminserver.pl]; then
65	echo removing /usr/darwin-5.5/local/bin/streamingadminserver.pl
66	rm -f \$DSSROOT/bin/streamingadminserver.pl
67	fi
68	

69	if [-f \$DSSROOT/bin/SpamPro]; then
70	echo removing \$DSSROOT/bin/SpamPro
71	rm -f \$DSSROOT/bin/SpamPro
71	fi
72	
73	if [-e StreamingLoadTool]; then
74	if [-f /usr/local/bin/StreamingLoadTool]; then
75	echo removing /usr/local/bin/StreamingLoadTool
76	rm -f /usr/local/bin/StreamingLoadTool
75	if [-f \$DSSROOT/bin/StreamingLoadTool]; then
76	echo removing \$DSSROOT/bin/StreamingLoadTool
77	rm -f \$DSSROOT/bin/StreamingLoadTool
77	fi
78	fi
...	...
82	echo Backing up previous config files
83	
84	if [-f /etc/streaming/streamingserver.xml]; then
85	echo backing up /etc/streaming/streamingserver.xml to
	/etc/streaming/streamingserver.xml.backup
86	mv /etc/streaming/streamingserver.xml
	/etc/streaming/streamingserver.xml.backup
87	fi
88	
89	if [-f /etc/streaming/streamingadminserver.pem]; then
90	echo backing up /etc/streaming/streamingadminserver.pem to
	/etc/streaming/streamingadminserver.pem.backup
91	mv /etc/streaming/streamingadminserver.pem
	/etc/streaming/streamingadminserver.pem.backup
92	fi
93	
94	if [-f /etc/streaming/qtusers]; then
95	echo backing up /etc/streaming/qtusers to
	/etc/streaming/qtusers.backup
96	mv /etc/streaming/qtusers /etc/streaming/qtusers.backup
97	fi
98	
99	if [-f /etc/streaming/qtgroups]; then
100	echo backing up /etc/streaming/qtgroups to
	/etc/streaming/qtgroups.backup
101	mv /etc/streaming/qtgroups /etc/streaming/qtgroups.backup
85	if [-f \$DSSROOT/etc/streamingserver.xml]; then
86	echo backing up \$DSSROOT/etc/streamingserver.xml to
	\$DSSROOT/etc/streamingserver.xml.backup
87	mv \$DSSROOT/etc/streamingserver.xml
	\$DSSROOT/etc/streamingserver.xml.backup
88	fi
89	
90	if [-f \$DSSROOT/etc/streamingadminserver.pem]; then
91	echo backing up \$DSSROOT/etc/streamingadminserver.pem to
	\$DSSROOT/etc/streamingadminserver.pem.backup
92	mv \$DSSROOT/etc/streamingadminserver.pem
	\$DSSROOT/etc/streamingadminserver.pem.backup
93	fi
94	
95	if [-f \$DSSROOT/etc/qtusers]; then
96	echo backing up \$DSSROOT/etc/qtusers to \$DSSROOT/etc/qtusers.backup
97	mv \$DSSROOT/etc/qtusers \$DSSROOT/etc/qtusers.backup
98	fi
99	
100	if [-f \$DSSROOT/etc/qtgroups]; then

101	echo backing up \$DSSROOT/etc/qtgroups to \$DSSROOT/etc/qtgroups.backup
102	mv \$DSSROOT/etc/qtgroups \$DSSROOT/etc/qtgroups.backup
102	fi
103	
104	if [-e StreamingLoadTool]; then
105	if [-f /etc/streaming/streamingloadtool.conf]; then
106	echo backing up /etc/streaming/streamingloadtool.conf to /etc/streaming/streamingloadtool.conf.backup
107	mv /etc/streaming/streamingloadtool.conf /etc/streaming/streamingloadtool.conf.backup
108	fi
109	fi
110	if [-f /etc/streaming/relayconfig.xml]; then
111	echo backing up /etc/streaming/relayconfig.xml to /etc/streaming/relayconfig.xml.backup
112	mv /etc/streaming/relayconfig.xml /etc/streaming/relayconfig.xml.backup
106	if [-f \$DSSROOT/etc/streamingloadtool.conf]; then
107	echo backing up \$DSSROOT/etc/streamingloadtool.conf to \$DSSROOT/etc/streamingloadtool.conf.backup
108	mv \$DSSROOT/etc/streamingloadtool.conf \$DSSROOT/etc/streamingloadtool.conf.backup
109	fi
110	fi
111	if [-f \$DSSROOT/etc/relayconfig.xml]; then
112	echo backing up \$DSSROOT/etc/relayconfig.xml to \$DSSROOT/etc/relayconfig.xml.backup
113	mv \$DSSROOT/etc/relayconfig.xml \$DSSROOT/etc/relayconfig.xml.backup
113	fi
114	
...	...
183	fi
184	
186	pushd "WebAdmin"
185	if ["\$noperlpath" = ""]; then
186	echo "Inserting path to perl into scripts.."
187	\$perl perlpath.pl \$perl streamingadminserver.pl AdminHtml/parse_xml.cgi
189	\$perl perlpath.pl \$perl src/streamingadminserver.pl AdminHtml/parse_xml.cgi
188	echo ""
189	fi
192	popd
190	
191	# Add the unprivileged user qtss as the server's run user
...	...
198	201
199	202 ## INSTALL NEW VERSION ##
200	if [! -d /usr/local/sbin]; then
201	echo creating "/usr/local/sbin" directory
202	mkdir -p /usr/local/sbin
203	if [! -d \$DSSROOT/sbin]; then
204	echo creating "\$DSSROOT/sbin" directory
205	mkdir -p \$DSSROOT/sbin
203	fi
204	if [! -d /usr/local/bin]; then
205	echo creating "/usr/local/bin" directory
206	mkdir -p /usr/local/bin
207	if [! -d \$DSSROOT/bin]; then
208	echo creating "\$DSSROOT/bin" directory
209	mkdir -p \$DSSROOT/bin

207	210	fi
208	211	
209		echo copying "DarwinStreamingServer" to "/usr/local/sbin/DarwinStreamingServer"
210		cp -f DarwinStreamingServer /usr/local/sbin/
211		
212		
213		echo copying "PlaylistBroadcaster" to "/usr/local/bin/PlaylistBroadcaster"
214		cp -f PlaylistBroadcaster /usr/local/bin/
215		
216		echo copying "MP3Broadcaster" to "/usr/local/bin/MP3Broadcaster"
217		cp -f MP3Broadcaster /usr/local/bin/
218		
219		echo copying "qtpasswd" to "/usr/local/bin/qtpasswd"
220		cp -f qtpasswd /usr/local/bin/
	212	echo copying "DarwinStreamingServer" to "\$DSSROOT/sbin/DarwinStreamingServer"
	213	cp -f DarwinStreamingServer \$DSSROOT/sbin/
	214	
	215	pushd "PlaylistBroadcaster.tproj"
	216	echo copying "PlaylistBroadcaster" to "\$DSSROOT/bin/PlaylistBroadcaster"
	217	cp -f PlaylistBroadcaster \$DSSROOT/bin/
	218	popd
	219	
	220	pushd "MP3Broadcaster"
	221	echo copying "MP3Broadcaster" to "\$DSSROOT/bin/MP3Broadcaster"
	222	cp -f MP3Broadcaster \$DSSROOT/bin/
	223	popd
	224	
	225	pushd "qtpasswd.tproj"
	226	echo copying "qtpasswd" to "\$DSSROOT/bin/qtpasswd"
	227	cp -f qtpasswd \$DSSROOT/bin/
	228	popd
221	229	
222	230	# For now, do not copy modules as there are no supported dynamic modules
223		# echo copying modules to "/usr/local/sbin/StreamingServerModules"
224		if [! -d /usr/local/sbin/StreamingServerModules]; then
225		echo creating "/usr/local/sbin/StreamingServerModules" directory
226		mkdir /usr/local/sbin/StreamingServerModules
227		fi
228		cp -f StreamingServerModules/* /usr/local/sbin/StreamingServerModules/
229		
230		if [-e /usr/local/sbin/StreamingServerModules/QTSSHomeDirectoryModule];
		then
231		echo copying "createuserstreamingdir" to "/usr/local/bin/createuserstreamingdir"
232		cp -f createuserstreamingdir \$INSTALLROOT/usr/local/bin/
233		fi
234		
235		if [! -d /etc/streaming]; then
236		echo creating "/etc/streaming" directory
237		mkdir -p /etc/streaming
238		fi
239		
240		if [! -e /etc/streaming/streamingserver.xml]; then
241		/usr/local/sbin/DarwinStreamingServer -x
242		fi
243		
244		chmod 600 /etc/streaming/streamingserver.xml
245		chown qtss /etc/streaming/streamingserver.xml
246		

247	echo;echo copying "relayconfig.xml-Sample" to "/etc/streaming/relayconfig.xml-Sample"
248	cp -f relayconfig.xml-Sample /etc/streaming/relayconfig.xml-Sample
249	chmod 600 /etc/streaming/relayconfig.xml-Sample
250	if [-e /etc/streaming/relayconfig.xml]; then
251	chown qtss /etc/streaming/relayconfig.xml
252	fi
253	
254	echo;echo copying "qtusers" to "/etc/streaming/qtusers"
255	cp -f qtusers /etc/streaming/qtusers
256	chmod 600 /etc/streaming/qtusers
257	chown qtss /etc/streaming/qtusers
258	
259	echo;echo copying "qtgroups" to "/etc/streaming/qtgroups"
260	cp -f qtgroups /etc/streaming/qtgroups
261	chmod 600 /etc/streaming/qtgroups
262	chown qtss /etc/streaming/qtgroups
263	
264	#echo;echo copying "streamingadminserver.pem" to "/etc/streaming/streamingadminserver.pem"
265	#cp -f streamingadminserver.pem /etc/streaming/streamingadminserver.pem
266	#chmod 400 /etc/streaming/streamingadminserver.pem
267	
268	if [! -d /var/streaming]; then
269	echo creating "/var/streaming" directory
270	mkdir -p /var/streaming
271	fi
272	
273	echo copying "readme.txt" to "/var/streaming/readme.txt"
274	cp -f readme.txt /var/streaming/readme.txt
275	
276	echo copying "3rdPartyAcknowledgements.rtf" to "/var/streaming/3rdPartyAcknowledgements.rtf"
277	cp -f 3rdPartyAcknowledgements.rtf /var/streaming/3rdPartyAcknowledgements.rtf
278	
279	if [! -d /var/streaming/logs]; then
280	echo creating "/var/streaming/logs" directory
281	mkdir -p /var/streaming/logs
282	fi
283	
284	if [! -d /usr/local/movies]; then
285	echo creating "/usr/local/movies" directory
286	mkdir -p /usr/local/movies
287	fi
288	
289	if [! -d /var/streaming/playlists]; then
290	echo creating "/var/streaming/playlists" directory
291	mkdir -p /var/streaming/playlists
292	fi
293	chmod 770 /var/streaming/playlists
294	chmod 775 /usr/local/movies
295	
296	echo copying "sample_100kbit.mov" into "/usr/local/movies/sample_100kbit.mov"
297	cp -f sample_100kbit.mov /usr/local/movies/
298	
299	echo copying "sample_300kbit.mov" into "/usr/local/movies/sample_300kbit.mov"
300	cp -f sample_300kbit.mov /usr/local/movies/
301	
302	echo copying "sample_100kbit.mp4" into "/usr/local/movies/sample_100kbit.mp4"
303	cp -f sample_100kbit.mp4 /usr/local/movies/

304	
305	echo copying "sample_300kbit.mp4" into "/usr/local/movies/sample_300kbit.mp4"
306	cp -f sample_300kbit.mp4 /usr/local/movies/
307	
308	echo copying "sample.mp3" into "/usr/local/movies/sample.mp3"
309	cp -f sample.mp3 /usr/local/movies/
310	
311	echo copying "sample_50kbit.3gp" into "/usr/local/movies/sample_50kbit.3gp"
312	cp -f sample_50kbit.3gp /usr/local/movies/
313	
314	echo copying "sample_h264_100kbit.mp4" into "/usr/local/movies/sample_h264_100kbit.mp4"
315	cp -f sample_h264_100kbit.mp4 /usr/local/movies/
316	
317	echo copying "sample_h264_300kbit.mp4" into "/usr/local/movies/sample_h264_300kbit.mp4"
318	cp -f sample_h264_300kbit.mp4 /usr/local/movies/
319	
320	echo copying "sample_h264_1mbit.mp4" into "/usr/local/movies/sample_h264_1mbit.mp4"
321	cp -f sample_h264_1mbit.mp4 /usr/local/movies/
231	# echo copying modules to "\$DSSROOT/sbin/StreamingServerModules"
232	if [! -d \$DSSROOT/sbin/StreamingServerModules]; then
233	echo creating "\$DSSROOT/sbin/StreamingServerModules" directory
234	mkdir \$DSSROOT/sbin/StreamingServerModules
235	fi
236	if [-d StreamingServerModules]; then
237	cp -f StreamingServerModules/* \$DSSROOT/sbin/StreamingServerModules/
238	fi
239	
240	if [-e \$DSSROOT/sbin/StreamingServerModules/QTSSHomeDirectoryModule]; then
241	echo copying "createuserstreamingdir" to "\$DSSROOT/bin/createuserstreamingdir"
242	cp -f createuserstreamingdir \$INSTALLROOT\$DSSROOT/bin/
243	fi
244	
245	if [! -d \$DSSROOT/etc]; then
246	echo creating "\$DSSROOT/etc" directory
247	mkdir -p \$DSSROOT/etc
248	fi
249	
250	if [! -e \$DSSROOT/etc/streamingserver.xml]; then
251	\$DSSROOT/sbin/DarwinStreamingServer -x
252	fi
253	
254	chmod 600 \$DSSROOT/etc/streamingserver.xml
255	chown qtss \$DSSROOT/etc/streamingserver.xml
256	
257	echo;echo copying "relayconfig.xml-Sample" to "\$DSSROOT/etc/relayconfig.xml-Sample"
258	cp -f relayconfig.xml-Sample \$DSSROOT/etc/relayconfig.xml-Sample
259	chmod 600 \$DSSROOT/etc/relayconfig.xml-Sample
260	if [-e \$DSSROOT/etc/relayconfig.xml]; then
261	chown qtss \$DSSROOT/etc/relayconfig.xml
262	fi
263	
264	echo;echo copying "qtusers" to "\$DSSROOT/etc/qtusers"
265	cp -f qtusers \$DSSROOT/etc/qtusers
266	chmod 600 \$DSSROOT/etc/qtusers
267	chown qtss \$DSSROOT/etc/qtusers
268	

269	echo;echo copying "qtgroups" to "\$DSSROOT/etc/qtgroups"
270	cp -f qtgroups \$DSSROOT/etc/qtgroups
271	chmod 600 \$DSSROOT/etc/qtgroups
272	chown qtss \$DSSROOT/etc/qtgroups
273	
274	echo;echo copying "streamingadminserver.pem" to "\$DSSROOT/etc/streamingadminserver.pem"
275	pushd "WebAdmin"
276	cp -f streamingadminserver.pem \$DSSROOT/etc/streamingadminserver.pem
277	chmod 400 \$DSSROOT/etc/streamingadminserver.pem
278	popd
279	
280	if [! -d /usr/local/logs/darwin]; then
281	echo creating "/usr/local/logs/darwin" directory
282	mkdir -p /usr/local/logs/darwin
283	fi
284	
285	pushd "Documentation"
286	echo copying "readme.txt" to "\$DSSROOT/var/readme.txt"
287	cp -f readme.txt \$DSSROOT/var/readme.txt
288	
289	echo copying "3rdPartyAcknowledgements.rtf" to "\$DSSROOT/var/3rdPartyAcknowledgements.rtf"
290	cp -f 3rdPartyAcknowledgements.rtf \$DSSROOT/var/3rdPartyAcknowledgements.rtf
291	popd
292	
293	if [! -d /usr/local/logs/darwin]; then
294	echo creating "/usr/local/logs/darwin" directory
295	mkdir -p /usr/local/logs/darwin
296	fi
297	
298	if [! -d \$DSSROOT/movies]; then
299	echo creating "\$DSSROOT/movies" directory
300	mkdir -p \$DSSROOT/movies
301	fi
302	
303	if [! -d \$DSSROOT/var/playlists]; then
304	echo creating "\$DSSROOT/var/playlists" directory
305	mkdir -p \$DSSROOT/var/playlists
306	fi
307	chmod 770 \$DSSROOT/var/playlists
308	chmod 775 \$DSSROOT/movies
309	
310	echo copying "sample_100kbit.mov" into "\$DSSROOT/movies/sample_100kbit.mov"
311	cp -f sample_100kbit.mov \$DSSROOT/movies/
312	
313	echo copying "sample_300kbit.mov" into "\$DSSROOT/movies/sample_300kbit.mov"
314	cp -f sample_300kbit.mov \$DSSROOT/movies/
315	
316	echo copying "sample_100kbit.mp4" into "\$DSSROOT/movies/sample_100kbit.mp4"
317	cp -f sample_100kbit.mp4 \$DSSROOT/movies/
318	
319	echo copying "sample_300kbit.mp4" into "\$DSSROOT/movies/sample_300kbit.mp4"
320	cp -f sample_300kbit.mp4 \$DSSROOT/movies/
321	
322	echo copying "sample.mp3" into "\$DSSROOT/movies/sample.mp3"
323	cp -f sample.mp3 \$DSSROOT/movies/
324	
325	echo copying "sample_50kbit.3gp" into "\$DSSROOT/movies/sample_50kbit.3gp"
326	cp -f sample_50kbit.3gp \$DSSROOT/movies/
327	

328	echo copying "sample_h264_100kbit.mp4" into "\$DSSROOT/movies/sample_h264_100kbit.mp4"
329	cp -f sample_h264_100kbit.mp4 \$DSSROOT/movies/
330	
331	echo copying "sample_h264_300kbit.mp4" into "\$DSSROOT/movies/sample_h264_300kbit.mp4"
332	cp -f sample_h264_300kbit.mp4 \$DSSROOT/movies/
333	
334	echo copying "sample_h264_1mbit.mp4" into "\$DSSROOT/movies/sample_h264_1mbit.mp4"
335	cp -f sample_h264_1mbit.mp4 \$DSSROOT/movies/
322 336	
323 337	if [-e StreamingLoadTool]; then
324	echo copying "StreamingLoadTool" to "/usr/local/bin/StreamingLoadTool"
325	cp -f StreamingLoadTool /usr/local/bin/
326	
327	echo copying "streamingloadtool.conf" to "/etc/streaming/streamingloadtool.conf"
328	cp -f streamingloadtool.conf /etc/streaming/
338	echo copying "StreamingLoadTool" to "\$DSSROOT/bin/StreamingLoadTool"
339	cp -f StreamingLoadTool \$DSSROOT/bin/
340	
341	echo copying "streamingloadtool.conf" to "\$DSSROOT/etc/streamingloadtool.conf"
342	cp -f streamingloadtool.conf \$DSSROOT/etc/
329 343	fi
330 344	
331 345	#WebAdmin install
332	echo copying "streamingadminserver.pl" into "/usr/local/sbin/streamingadminserver.pl"
333	cp -f streamingadminserver.pl /usr/local/sbin/streamingadminserver.pl
334	
335	if [-d /var/streaming/AdminHtml/]; then
336	echo removing old version of html from "/var/streaming/AdminHtml"
337	rm -r -f /var/streaming/AdminHtml/*
346	pushd "WebAdmin/src"
347	echo copying "streamingadminserver.pl" into "\$DSSROOT/sbin/streamingadminserver.pl"
348	cp -f streamingadminserver.pl \$DSSROOT/sbin/streamingadminserver.pl
349	popd
350	
351	pushd "WebAdmin"
352	if [-d \$DSSROOT/var/AdminHtml/]; then
353	echo removing old version of html from "\$DSSROOT/var/AdminHtml"
354	rm -r -f \$DSSROOT/var/AdminHtml/*
338 355	fi
339 356	
340	if [! -d /var/streaming/]; then
341	echo creating "/var/streaming/AdminHtml" directory
342	mkdir -p /var/streaming/AdminHtml
343	fi
344	echo copying Admin HTML to "/var/streaming/AdminHtml" directory
345	cp -f -r AdminHtml /var/streaming/
346	
347	
348	chown -R -f qtss /var/streaming/
349	chown -R -f qtss /etc/streaming/
350	chown -R -f qtss /usr/local/movies/
357	if [! -d \$DSSROOT/var/AdminHtml]; then
358	echo creating "\$DSSROOT/var/AdminHtml" directory
359	mkdir -p \$DSSROOT/var/AdminHtml

360	fi
361	echo copying Admin HTML to "\$DSSROOT/var/AdminHtml" directory
362	cp -f -r WebAdminHtml \$DSSROOT/var
363	popd
364	
365	chown -R -f qtss /usr/local/logs/
366	chown -R -f qtss \$DSSROOT/etc/
367	chown -R -f qtss \$DSSROOT/movies/
351 368	
352 369	echo;echo "Launching streamingadminserver.pl"
353	/usr/local/sbin/streamingadminserver.pl
370	\$DSSROOT/sbin/streamingadminserver.pl
354 371	
355 372	echo;echo Installation Complete
...	...
403 420	echo ""
404 421	
405	# Add the new admin username to /etc/streaming/qtusers
406	/usr/local/bin/qlpasswd -p \$password \$username
407	
408	# Add the new admin username to /etc/streaming/qtgroups
422	# Add the new admin username to \$DSSROOT/etc/qtusers
423	\$DSSROOT/bin/qlpasswd -p \$password \$username
424	
425	# Add the new admin username to \$DSSROOT/etc/qtgroups
409 426	# and delete the default admin username
410	echo admin: \$username > /etc/streaming/qtgroups.tmp
411	mv /etc/streaming/qtgroups.tmp /etc/streaming/qtgroups
412	
413	# Remove the default admin username to /etc/streaming/qtusers
414	/usr/local/bin/qlpasswd -F -d 'aGFja2ll' > /dev/null
427	echo admin: \$username > \$DSSROOT/etc/qtgroups.tmp
428	mv \$DSSROOT/etc/qtgroups.tmp \$DSSROOT/etc/qtgroups
429	
430	# Remove the default admin username to \$DSSROOT/etc/qtusers
431	\$DSSROOT/bin/qlpasswd -F -d 'aGFja2ll' > /dev/null
415 432	
416	chown -R -f qtss /etc/streaming/
433	chown -R -f qtss \$DSSROOT/etc/
417 434	
418 435	echo Setup Complete!

DarwinStreamingServer/trunk/Makefile.POSIX

Tabular | Unified

r90	r694	
43	43	CCFLAGS += -IPrefsSourceLib
44	44	CCFLAGS += -IServer.tproj
	45	CCFLAGS += -I/usr/include/mysql
45	46	
46	47	# EACH DIRECTORY WITH A STATIC LIBRARY MUST BE APPENDED IN THIS MANNER TO THE LINKOPTS
...	...	
48	49	LINKOPTS = -LCommonUtilitiesLib
49	50	LINKOPTS += -LQtFileLib
	51	LINKOPTS += -L/usr/lib/mysql -lmysqlclient
50	52	
51	53	C++FLAGS = \$(CCFLAGS)
...	...	

129	131	APIModules/QTSSAccessModule/QTSSAccessModule.cpp \
130	132	APIModules/QTSSHttpFileModule/QTSSHttpFileModule.cpp \
131		APIModules/QTSSAccessModule/AccessChecker.cpp
	133	APIModules/QTSSAccessModule/AccessChecker.cpp \
	134	APIModules/QTSSAccessModule/ShibMysql.cpp \
	135	APIModules/QTSSAccessModule/ModuSessionMysql.cpp
132	136	
133	137	# CCFLAGS += \$(foreach dir,\$(HDRS),-I\$(dir))

DarwinStreamingServer/trunk/APIStubLib/QTSS.h

Tabular | Unified

r90	r95	
177	177	qtssAuthNone = 0,
178	178	qtssAuthBasic = 1,
179		qtssAuthDigest = 2
	179	qtssAuthDigest = 2,
	180	qtssAuthShib = 3,
	181	qtssAuthModu = 4
180	182	};
181	183	typedef UInt32 QTSS_AuthScheme;

DarwinStreamingServer/trunk/APICommonCode/QTAccessFile.cpp

Tabular | Unified

r90	r95	
398	398	if (word.Equal("basic"))
399	399	authScheme = qtssAuthBasic;
	400	else if (word.Equal("shib"))
	401	authScheme = qtssAuthShib;
	402	else if (word.Equal("modu"))
	403	authScheme = qtssAuthModu;
400	404	else if (word.Equal("digest"))
401	405	authScheme = qtssAuthDigest;

DarwinStreamingServer/trunk/APIModules/QTSSAccessModule/QTSSAccessModule.cpp

Tabular | Unified

r90	r488	
50	50	#include "QTAccessFile.h"
51	51	#include "QTSSModuleUtils.h"
52		
	52	#include "ShibMysql.h"
	53	#include "ModuSessionMysql.h"
53	54	#ifndef __Win32__
54	55	#include <unistd.h>
...	...	
58	59	#include <errno.h>
59	60	
60		
	61	#include "SafeStdLib.h"
	62	
	63	#include <string.h>
	64	#include <strings.h>
61	65	
62	66	// ATTRIBUTES
...	...	
72	76	//static Bool16 sDefaultAuthenticationEnabled = true;
73	77	//static Bool16 sAuthenticationEnabled = true;

	78	
	79	
	80	static char* shibmysqlDefaultHost = NULL;
	81	static char* shibmysqlDefaultName = NULL;
	82	static char* shibmysqlDefaultUser = NULL;
	83	static char* shibmysqlDefaultPassword = NULL;
	84	static char* shibmysqlDefaultSocket = NULL;
	85	static char* shibmysqlDefaultPort = NULL;
	86	
	87	static char* shibmysqlHost = NULL;
	88	static char* shibmysqlName = NULL;
	89	static char* shibmysqlUser = NULL;
	90	static char* shibmysqlPassword = NULL;
	91	static char* shibmysqlSocket = NULL;
	92	static char* shibmysqlPort = NULL;
	93	
	94	static char* moduMysqlDefaultHost = NULL;
	95	static char* moduMysqlDefaultName = NULL;
	96	static char* moduMysqlDefaultUser = NULL;
	97	static char* moduMysqlDefaultPassword = NULL;
	98	static char* moduMysqlDefaultSocket = NULL;
	99	static char* moduMysqlDefaultPort = NULL;
	100	
	101	static char* moduMysqlHost = NULL;
	102	static char* moduMysqlName = NULL;
	103	static char* moduMysqlUser = NULL;
	104	static char* moduMysqlPassword = NULL;
	105	static char* moduMysqlSocket = NULL;
	106	static char* moduMysqlPort = NULL;
74	107	
75	108	static char* sDefaultUsersFilePath = DEFAULTPATHS_ETC_DIR "qtusers";
...	...	
203	236	QTAcessFile::Initialize();
204	237	
	238	shibmysqlHost = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlHost", shibmysqlDefaultHost);
	239	shibmysqlName = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlName", shibmysqlDefaultName);
	240	shibmysqlUser = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlUser", shibmysqlDefaultUser);
	241	shibmysqlPassword = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlPassword", shibmysqlDefaultPassword);
	242	shibmysqlSocket = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlSocket", shibmysqlDefaultSocket);
	243	shibmysqlPort = QTSSModuleUtils::GetStringAttribute(sPrefs, "shibmysqlPort", shibmysqlDefaultPort);
	244	
	245	moduMysqlHost = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlHost", moduMysqlDefaultHost);
	246	moduMysqlName = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlName", moduMysqlDefaultName);
	247	moduMysqlUser = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlUser", moduMysqlDefaultUser);
	248	moduMysqlPassword = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlPassword", moduMysqlDefaultPassword);
	249	moduMysqlSocket = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlSocket", moduMysqlDefaultSocket);
	250	moduMysqlPort = QTSSModuleUtils::GetStringAttribute(sPrefs, "moduMysqlPort", moduMysqlDefaultPort);
	251	
	252	// This is kind of pointless, since we *want* the stats in this database,
	253	// and this code only gets executed on server startup, which rarely happens.

254	/*
255	ShibMysql
	s(shibmysqlHost,shibmysqlName,shibmysqlUser,shibmysqlPassword,shibmysqlPort,shibmysqlSocket);
256	
257	if (s.clearDarwinSessions()) {
258	return QTSS_NoErr;
259	} else {
260	qtss_printf("failed to clear shibboleth session table!\n");
261	return QTSS_RequestFailed;
262	}
263	*/
264	
205	return QTSS_NoErr;
206	}
...	...
533	QTSS_ActionFlags authorizeAction = qtssActionFlagsRead;
534	
595	// NYU MODIFICATIONS
596	QTSS_RTSPRequestObject theRTSPRequest = inParams->inRTSPRequest;
597	
598	OSMutexLocker locker(sUserMutex);
599	
600	if ((NULL == inParams) (NULL == inParams->inRTSPRequest))
601	return QTSS_RequestFailed;
602	
603	// Get the user profile object from the request object
604	QTSS_UserProfileObject theUserProfile = NULL;
605	UInt32 len = sizeof(QTSS_UserProfileObject);
606	QTSS_Error theErr = QTSS_GetValue(theRTSPRequest, qtssRTSPReqUserProfile, 0,
	(void*)&theUserProfile, &len);
607	Assert(len == sizeof(QTSS_UserProfileObject));
608	if (theErr != QTSS_NoErr)
609	return theErr;
610	
611	// Check for a users and groups file in the access file
612	// For this, first get local file path and root movie directory
613	//get the local file path
614	char* pathBuffStr = QTSSModuleUtils::GetLocalPath_Copy(theRTSPRequest);
615	OSCharArrayDeleter pathBuffDeleter(pathBuffStr);
616	if (NULL == pathBuffStr)
617	return QTSS_RequestFailed;
618	//get the root movie directory
619	char* movieRootDirStr =
	QTSSModuleUtils::GetMoviesRootDir_Copy(theRTSPRequest);
620	OSCharArrayDeleter movieRootDeleter(movieRootDirStr);
621	if (NULL == movieRootDirStr)
622	return QTSS_RequestFailed;
623	// Now get the access file path
624	char* accessFilePath = QTAccessFile::GetAccessFile_Copy(movieRootDirStr,
	pathBuffStr);
625	OSCharArrayDeleter accessFilePathDeleter(accessFilePath);
626	// Parse the access file for the AuthUserFile and AuthGroupFile keywords
627	char* usersFilePath = NULL;
628	char* groupsFilePath = NULL;
629	
630	// Get the request action from the request object
631	QTSS_ActionFlags action = qtssActionFlagsNoFlags;
632	len = sizeof(action);
633	theErr = QTSS_GetValue(theRTSPRequest, qtssRTSPReqAction, 0, (void*)&action,
	&len);


```

634     Assert(len == sizeof(action));
635     if (theErr != QTSS_NoErr)
636         return theErr;
637
638     // Allocates memory for usersFilePath and groupsFilePath
639     QTSS_AuthScheme authScheme =
QTAccessFile::FindUsersAndGroupsFilesAndAuthScheme(accessFilePath, action,
&usersFilePath, &groupsFilePath);
640
641     if(authScheme == qtssAuthShib || authScheme == qtssAuthModu) {
642         QTSS_RTSPSessionObject theRTSPSession = inParams->inRTSPSession;
643
644         Bool16 allowed = false;
645         (void)QTSS_SetValue(theRTSPRequest, qtssRTSPReqUserAllowed, 0,
(void*)&allowed, sizeof(allowed)); // set allowed to default to deny
646
647         char* theSessionID;
648         char* theLocalAddrStr;
649         char* theBaseAddrStr;
650         char* theRequestedURL;
651
652         theErr = QTSS_GetValueAsString(theRTSPSession, qtssRTSPSesID, 0,
&theSessionID);
653         theErr = QTSS_GetValueAsString(theRTSPSession,
qtssRTSPSesRemoteAddrStr, 0, &theLocalAddrStr);
654         theErr = QTSS_GetValueAsString(theRTSPRequest,
qtssRTSPReqFullRequest, 0, &theRequestedURL);
655
656         theErr = QTSS_GetValueAsString(theRTSPRequest,
qtssRTSPReqFilePathTrunc, 0, &theBaseAddrStr);
657         if (theErr != QTSS_NoErr)
658             theBaseAddrStr = "/"; // in the case of / we want a / instead
of a NULL
659
660         /* will want to modify this for proper tokenization, but it works for
now */
661         char* sessionkey;
662         if(index(theRequestedURL, '?')){
663             char buf[512];
664             snprintf(buf, 512, theRequestedURL);
665
666             sessionkey = ::strtok(buf, "\n");
667             sessionkey = ::strtok(buf, "?");
668             sessionkey = ::strtok(NULL, "=");
669
670             if(authScheme == qtssAuthShib && ! ::strcmp(sessionkey,
"shib")) {
671                 sessionkey = ::strtok(NULL, " ");
672             }
673             else if(authScheme == qtssAuthModu && ! ::strcmp(sessionkey,
"sess")) {
674                 sessionkey = ::strtok(NULL, " ");
675             }
676             else {
677                 return QTSS_RequestFailed;
678             }
679         }
680
681         if(authScheme == qtssAuthShib){
682             ShibMysql s(shibmysqlHost, shibmysqlName, shibmysqlUser,
shibmysqlPassword, shibmysqlPort, shibmysqlSocket);
683

```



```

684         s.set(theBaseAddrStr, sessionkey, theLocalAddrStr,
theSessionID);
685
686         /*
687         if (! strcmp(theLocalAddrStr, "216.165.106.48")) {
688             qtss_printf("\n\n\nusing ip
authentication!\n\n\n\n");
689             allowed = true;
690             (void)QTSS_SetValue(theRTSPRequest,
qtssRTSPReqUserAllowed, 0, (void*)&allowed, sizeof(allowed));
691             return QTSS_NoErr;
692         }
693         else{
694             qtss_printf("not using ip authentication %s\n\n\n",
theLocalAddrStr);
695         }
696         */
697
698         if(s.check() || s.checkDarwinSession()){
699             allowed = true;
700             (void)QTSS_SetValue(theRTSPRequest,
qtssRTSPReqUserAllowed, 0, (void*)&allowed, sizeof(allowed));
701             return QTSS_NoErr;
702         }
703         else {
704             return QTSS_RequestFailed;
705         }
706     }
707     else{
708         ModuSessionMysql s(moduMysqlHost, moduMysqlName,
moduMysqlUser, moduMysqlPassword, moduMysqlPort, moduMysqlSocket);
709
710         s.set(theBaseAddrStr, sessionkey, theLocalAddrStr,
theSessionID);
711
712         /*
713         if (! strcmp(theLocalAddrStr, "216.165.106.48")) {
714             qtss_printf("\n\n\nusing ip
authentication!\n\n\n\n");
715             allowed = true;
716             (void)QTSS_SetValue(theRTSPRequest,
qtssRTSPReqUserAllowed, 0, (void*)&allowed, sizeof(allowed));
717             return QTSS_NoErr;
718         }
719         else{
720             qtss_printf("not using ip authentication %s\n\n\n",
theLocalAddrStr);
721         }
722         */
723
724         if(s.check() || s.checkDarwinSession()){
725             allowed = true;
726             (void)QTSS_SetValue(theRTSPRequest,
qtssRTSPReqUserAllowed, 0, (void*)&allowed, sizeof(allowed));
727             return QTSS_NoErr;
728         }
729         else {
730             return QTSS_RequestFailed;
731         }
732     }
733 }
734 // NYU MODIFICATIONS

```


	735	
535	736	return QTAcessFile::AuthorizeRequest(inParams,allowNoAccessFiles, noAction,
536	737	authorizeAction);
		}

DarwinStreamingServer/trunk/Server.tproj/QTSServerPrefs.cpp

Tabular | Unified

r90	r95	
519	519	static StrPtrLen sBasicAuthScheme("basic");
520	520	static StrPtrLen sDigestAuthScheme("digest");
	521	static StrPtrLen sShibAuthScheme("shib");
	522	static StrPtrLen sModuAuthScheme("modu");
521	523	
522	524	// Get the auth scheme attribute
...	...	
529	531	else if (theAuthScheme->Equal(sDigestAuthScheme))
530	532	fAuthScheme = qtssAuthDigest;
	533	else if (theAuthScheme->Equal(sShibAuthScheme))
	534	fAuthScheme = qtssAuthShib;
	535	else if (theAuthScheme->Equal(sModuAuthScheme))
	536	fAuthScheme = qtssAuthModu;
531	537	}
532	538	

DarwinStreamingServer/trunk/Server.tproj/RTSPSession.cpp

Tabular | Unified

r90	r95	
619	619	else if(scheme == qtssAuthDigest)
620	620	fRequest->SetAuthScheme(qtssAuthDigest);
	621	else if(scheme == qtssAuthShib)
	622	fRequest->SetAuthScheme(qtssAuthShib);
	623	else if(scheme == qtssAuthModu)
	624	fRequest->SetAuthScheme(qtssAuthModu);
621	625	}
622	626	