Player Registration				
App ID	provide value			
Client ID	provide value			
VCID	provide value			
Ad Measurement	provide value			
SDK Player Association				
SDK Version(s)	should be current SDK Version. If not, next version will be picked up on			
Prod App ID pointed to sfcode	eu-uat / eu			
based on: DCR Video - Browser Test Cases v.4.7				
updated: 21.04.2020				

Test Case	Test Index	Validation	Expected Results	Pass/Fail	Value	Notes
VIDEO-ON-DEMAND (VOD): PLAY FULL-LENGTH EPISODE						
1a.1 SDK Configuration	1a.1.1	SDK initialized (API call will not appear in logs if map local redirect is required)	**appid: [correct App ID]**		*provide value*	
			sfcode: [correct collection environment]		*provide value*	
			appname: [player name]		*provide value*	
			appversion: [correct build version]		*provide value*	
			nol_devDebug: 'DEBUG' - Debug logging should be enabled		*provide value*	
1a.2 Play Metadata	1a.2.1	play event is called for content	**Event: play, playerInfo**			
	1a.2.2	Required metadata is passed when calling play	**channelName: [channel name] (can be blank string or defaultChannelName)**		*provide value*	
1a.3 Content Metadata	1a.3.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**			
	1a.3.2	Required content metadata is passed in loadMetadata and is accurate No Special characters allowed (https://engineeringportal.nielsen.com/docs/Special_Characters)	**clientid: [assigned Nielsen client ID - if client is overriding]**		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			program: [program name]		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
			hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models		*provide value*	
			c-variables		*provide value*	
NO PLAYHEADS BEFORE PREROLL (*if applicable*)						
1a.4 Content Playheads	1a.4.1	No playheads were passed between content metadata and preroll metadata (if applicable)	No playheads were passed			
PLAY PREROLL AD (*if applicable*)						
1a.5 Preroll Ad	1a.5.1	loadMetadata event is called for ad	**Event: loadMetadata, metadata**			
	1a.5.2	Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'preroll' ['ad' is acceptable for SDK versions aa 5.1.1.26/ ai 5.1.1.29 +]**		*provide value*	
			assetid: [any value]		*provide value*	
	1a.5.3	playhead is passed every second for each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2 Event: playheadPosition, playhead: 3...[playhead for each second]			
		stop event called for each ad	**Event: stop**			
	1a.5.4	correct event sequence is called for ad pod or each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop			
1a.6 Content Metadata: Second Validation	1a.6.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**			
	1a.6.2	Content metadata is the **same as original metadata** passed during initial loadMetadata call	**clientid: [assigned Nielsen client ID - if client is overriding]**		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			program: [program name]		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
			hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models		*provide value*	
			c-variables		*provide value*	
VIEW PING AFTER CONTENT STARTS						
1a.8 View Ping	1a.8.1	View ping after preroll (if applicable)	You should only receive a view ping after the preroll, once content starts.			
CONTINUE PLAYING CONTENT UNINTERRUPTED UNTIL DURATION PING						
1a.9 Duration Ping	1a.9.1	Duration ping fires after 5 minutes	**c27=cln,[total number of seconds that content played]**		*provide value*	
			c51= adl, [total number of seconds that played for the ads within preroll ad pod (if there were preroll ads)]		*provide value*	
			c52= noad, [number of ads played within preroll ad pod (if there were preroll ads)]		*provide value*	
SCRUB BACKWARD						

Color Codes
testing steps
critical
initialization
content
ads
pings (checked on Nielsen's side)
optional metadata
required in certain scenarios/optional
required in dev sanity
always required (dev sanity, app update, post-prod)

	1a.10 Scrub Backward	1a.10.1	Provide a UNIX timestamp from any line and let content play around 30 seconds after scrubbing. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
		1a.10.2	stop event may be called in some ap	Event: stop		
		1a.10.3	new playhead position is accurate (*position decreases*)	Event: playheadPosition, playhead: [new position]		
		1a.10.4	playhead continues to be passed from new position	Event: playheadPosition, playhead: [new position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
	DURATION PING AFTER SCRUB					
	1a.11 Duration Ping	1a.11.1	Duration ping fires after scrubbing ***Note**: If the ping does not fire, make sure that it is still crediting correctly and note it for reference.*	**c27=[number of seconds that played. Make sure to factor in scrub time]** **c51= adl, [total number of seconds that played for the ads within preroll ad pod (if there were preroll ads)]** **c52= noad, [number of ads played within midroll ad pod (if there were midroll ads)]**		provide value provide value provide value
	SCRUB BACK TO BEGINNING OF CONTENT					
	1a.12 Scrub Back to Beginning	1a.12.1	Provide a UNIX timestamp from any line and let content play around 30 seconds after scrubbing. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
		1a.12.2	stop event may be called in some ap	Event: stop		
		1a.12.3	playhead is passed every second starting at 0 seconds	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2 Event: playheadPosition, playhead: 3...[playhead for each second]		
	NO VIEW PING AFTER SCRUBBING TO BEGINNING					
	1a.13 No View Ping	1a.13.1	Should not receive a new view ping after scrubbing to beginning	Pass = No view ping Fail = View ping		
	SCRUB FORWARD					
	1a.14 Scrub Forward	1a.14.1	Provide a UNIX timestamp from any line and let content play around 30 seconds after scrubbing. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
		1a.14.2	stop event may be called in some ap	Event: stop		
		1a.14.3	new playhead position is accurate (position increases)	Event: playheadPosition, playhead: [new position]		
		1a.14.4	playhead continues to be passed from new position	Event: playheadPosition, playhead: [new position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
	1a.15 Stop Call	1a.15.1	Stop (if midroll. If not, move on to pause)	Event: stop (optional between ad/content as of aa 5.1.1.5 and ai 5.1.1.6)		
	DURATION PING AFTER SCRUB					
	1a.16 Duration Ping	1a.16.1	Duration ping fires after scrubbing ***Note**: If the ping does not fire, make sure that it is still crediting correctly and note it for reference.*	**c27=[number of seconds that played. Make sure to factor in scrub time]** **c51= adl, [total number of seconds that played for the ads within preroll ad pod (if there were preroll ads)]** **c52= noad, [number of ads played within midroll ad pod (if there were midroll ads)]**		*provide value* *provide value* *provide value*
	PLAY MIDROLL AD (*if applicable*)					
	1a.17 Midroll Ad Break	1a.17.1	playhead for content stops being passed	Event: playheadPosition, playhead: [last position in seconds]		
		1a.17.2	loadMetadata event is called for each ad	**Event: loadMetadata, metadata**		
		1a.17.3	Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'midroll' ['ad' is acceptable for AppSDK aa 5.1.1.26/ ai 5.1.1.29 +]** **assetid: [unique Ad ID]**		*provide value* *provide value*
		1a.17.4	playhead is passed every second for each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2...[playhead for each second]		
		1a.17.5	stop event called for each ad	**Event: stop**		
		1a.17.6	correct event sequence is called for ad pod or each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop		
		1a.18.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]		*provide value (if passed)*
				subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*
				type: 'content'		*provide value*
				assetid: [unique asset ID]		*provide value*

			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**		*provide value*	
			hasAds : "0": [Non-ad-supported subscription models.2 **"1": Ad-supported subscription models.] - to be tested for all available subscription models**		*provide value*	
		1a.18.2	Content metadata is the **same as original metadata** passed during initial loadMetadata call	**c-variables**	*provide value*	
	1a.18 Content Resumes following Midroll	1a.18.3	playhead continues to be passed from resumed content position	Event: playheadPosition, playhead: [resumed content position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
				SCRUB BACKWARD ACROSS MIDROLL AD TO CHECK FOR PROPER CREDITING (if applicable)		
		1a.19.1	playhead for content stops being passed	Event: playheadPosition, playhead: [last position in seconds]		
		1a.19.2	loadMetadata event is called for each ad	**Event: loadMetadata, metadata**		
				assetid: [unique Ad ID]	*provide value*	
		1a.19.3	Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'midroll' ['ad' is acceptable for AppSDK aa 5.1.1.26/ ai 5.1.1.29 +]**	*provide value*	
		1a.19.4	playhead is passed every second for each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2...[playhead for each second]		
		1a.19.5	stop even called for each ad	**Event: stop**		
	1a.19 Midroll Ad Break	1a.19.6	correct event sequence is called for ad pod or each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop, (at ad end)		
		1a.20.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]	*provide value (if passed)*	
				subbrand: [assigned Nielsen subbrand - if client is overriding]	*provide value (if passed)*	
				type: 'content'	*provide value*	
				assetid: [unique asset ID]	*provide value*	
				program: [program name]	*provide value*	
				title: [episode title]	*provide value*	
				length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**	*provide value*	
				hasAds : "0": [Non-ad-supported subscription models.2 **"1": Ad-supported subscription models.] - to be tested for all available subscription models**	*provide value*	
		1a.20.2	Content metadata is the **same as original metadata** passed during initial loadMetadata call	**c-variables**	*provide value*	
	1a.20 Content Resumes following Midroll	1a.20.3	Playhead continues to be passed from resumed content position	Event: playheadPosition, playhead: [resumed content position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
				PAUSE CONTENT		
		1a.21.1	Provide a UNIX timestamp when pausing content. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H	*provide value*	
	1a.21 Content Paused	1a.21.2	playhead is not passed	**Event: playheadPosition, playhead: [last position of content in seconds]** ***playhead does not continue to be sent***		
	1a.22 Stop Call	1a.22.1	stop event after pause	**Event: stop**		
				DURATION PING AFTER PAUSE		
				c27=[number of seconds that played. Make sure to factor in scrub time]	provide value	
			Duration ping fires after pause.	**c51= adl, [total number of seconds that played for the ads within preroll ad pod (if there were preroll ads)]**	provide value	
	1a.23 Duration Ping	1a.23.1	Should contain proper c27 value after scrubbing	**c52= noad, [number of ads played within midroll ad pod (if there were midroll ads)]**	provide value	
				RESUME CONTENT PLAYBACK AFTER 30 SECONDS		
		1a.24.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]	*provide value (if passed)*	
				subbrand: [assigned Nielsen subbrand - if client is overriding]	*provide value (if passed)*	
				type: 'content'	*provide value*	
				assetid: [unique asset ID]	*provide value*	
				program: [program name]	*provide value*	
				title: [episode title]	*provide value*	
				length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**	*provide value*	

		Content metadata is the **same as original metadata** passed during initial loadMetadata call	**hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models**		*provide value*	
	1a.24.2		**c-variables**		*provide value (if passed)*	
1a.24 Content Resumed	1a.24.3	playhead continues to be passed from resumed content position	Event: playheadPosition, playhead: [resumed content position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
SCRUB FORWARD NEAR THE END OF CONTENT AND RESUME PLAYBACK						
	1a.25.1	Provide a UNIX timestamp from any line and let content play around 30 seconds after scrubbing. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*	
	1a.25.2	playhead continues to be passed from new position	Event: playheadPosition, playhead: [new position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
	1a.25.3	final playhead position is passed *Note: Make sure that the final playhead does not exceed the length of the video represented as SD in the pings.*	Event: playheadPosition, playhead: [final playhead position]			
1a.25 Content Completes	1a.25.4	end event called at completion of content playback	**Event: end, playhead: [final playhead position]**			
DURATION PING AT CONCLUSION OF CONTENT						
			c27=[number of seconds that played since last duration ping]		provide value	
		Duration ping fires at conclusion of Content	**c51= adl, [total number of seconds that played for the ads within any ad pods that played]**		provide value	
1a.26 Duration Ping	1a.26.1		**c52= noad, [number of ads played within ad pods]**		provide value	
PLAY POSTROLL AD (*if applicable*)						
	1a.27.1	loadMetadata event is called for each ad	**Event: loadMetadata, metadata**			
	1a.27.2	Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'postroll' ['ad' is acceptable for AppSDK aa 5.1.1.26/ ai 5.1.1.29 +]**		*provide value*	
			assetid: [unique Ad ID]		*provide value*	
	1a.27.3	playhead is passed every second for each ad pod or each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2...[playhead for each second]			
	1a.27.4	stop event called for each ad	**Event: stop**			
1a.27 Postroll Ad	1a.27.5	correct event sequence is called for each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop			
DURATION PING AFTER POSTROLL (*if applicable*)						
			c27=0 [Final duration ping must be 0]		provide value	
		Duration ping fires at conclusion of postroll. c27 should equal 0, while c51 and c52 have crediting for the postroll ad(s)	**c51= adl, [total number of seconds that played for postroll ad]**		provide value	
1a.28 Duration Ping	1a.28.1		**c52= noad, [number of ads played within postroll ad pod (if there were postroll ads)]**		provide value	
RESTART VIDEO (*if applicable*)						
	1a.29.1	play event is called for content	**Event: play, playerInfo**			
1a.29 Play Metadata	1a.29.2	Required metadata is passed when calling play	**channelName: [channel name] (can be blank string or defaultChannelName)**		*provide value*	
	1a.30.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**			
		"Required content metadata is passed in loadMetadata and is accurate No Special characters allowed (https://engineeringportal.nielsen.com/docs/Special_Characters)"	**clientid: [assigned Nielsen client ID - if client is overriding]**		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
			hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models		*provide value*	
1a.30 Content Metadata	1a.30.2		**c-variables**		*provide value*	
NO PLAYHEADS BEFORE PREROLL AFTER REPLAY (*if applicable*)						
1a.31 Content Playheads	1a.31.1	No playheads were passed between content metadata and preroll metadata (if applicable)	No playheads were passed			
PLAY PREROLL AD AFTER REPLAY (*if applicable*)						
	1a.32.1	loadMetadata event is called for ad	**Event: loadMetadata, metadata**			
		Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'preroll' ['ad' is acceptable for AppSDK aa 5.1.1.26/ ai 5.1.1.29 +]**		*provide value*	
	1a.32.2		**assetid: [any value]**		*provide value*	

		1a.32.3	playhead is passed every second for each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2 Event: playheadPosition, playhead: 3...[playhead for each second]		
			stop event called for each ad	**Event: stop**		
	1a.32 Preroll Ad	1a.32.4	correct event sequence is called for ad pod or each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop		
		1a.33.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]	*provide value (if passed)*	
				subbrand: [assigned Nielsen subbrand - if client is overriding]	*provide value (if passed)*	
				type: 'content'	*provide value*	
				program: [program name]	*provide value*	
				assetid: [unique asset ID]	*provide value*	
				program: [program name]	*provide value*	
				title: [episode title]	*provide value*	
				length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**	*provide value*	
	1a.33 Content Metadata: Second Validation	1a.33.2	Content metadata is the **same as original metadata** passed during initial loadMetadata call	**hasAds : "0": [Non-ad-supported subscription models.2** **"1": Ad-supported subscription models.] - to be tested for all available subscription models**	*provide value*	
				c-variables	*provide value*	
	1a.34 Content Playhead	1a.34.1	playhead is passed every second starting at 0 seconds	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2 Event: playheadPosition, playhead: 3...[playhead for each second]		
VIEW PING AFTER CONTENT REPLAY (*if applicable*)						
	1a.35 View Ping	1a.35.1	View ping after preroll (if applicable)	You should only receive a view ping AFTER the preroll, once content starts.		
PLAY NEW VIDEO						
VIDEO-ON-DEMAND (VOD): PLAY NEW FULL LENGTH EPISODE						
		1a.36.1	loadMetadata event is called for new content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]	*provide value (if passed)*	
				subbrand: [assigned Nielsen subbrand - if client is overriding]	*provide value (if passed)*	
				type: 'content'	*provide value*	
				program: [program name]	*provide value*	
				assetid: [unique asset ID]	*provide value*	
				program: [program name]	*provide value*	
				title: [episode title]	*provide value*	
				length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**	*provide value*	
		1a.36.2	Required content metadata is passed in loadMetadata and is accurate for new content	**hasAds : "0": [Non-ad-supported subscription models.2** **"1": Ad-supported subscription models.] - to be tested for all available subscription models**	*provide value*	
				c-variables	*provide value*	
	1a.36 Content Metadata - New Video Asset	1a.36.3	playhead is passed every second starting at 0 seconds	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2 Event: playheadPosition, playhead: 3...[playhead for each second]		
VIEW PING COMES AFTER CONTENT STARTS AND NOT BEFORE A PREROLL						
	1a.37 View Ping	1a.37.1	View ping after preroll	You should only receive a view ping after the preroll has concluded and once content starts.		
SCRUB FORWARD ACROSS MIDROLL AD *(if applicable)*						
	1a.38 Scrub forward past midroll	1a.38.1	playhead for content stops being passed	Event: playheadPosition, playhead: [last position in seconds]		
PLAY MIDROLL AD						
		1a.39.1	loadMetadata event is called for each ad	**Event: loadMetadata, metadata**		
			Required Ad Metadata is passed in loadMetadata and is accurate	**type: 'midroll' ['ad' is acceptable for AppSDK aa 5.1.1.26/ ai 5.1.1.29 +]**	*provide value*	
		1a.39.2		**assetid: [unique Ad ID]**	*provide value*	
		1a.39.3	playhead is passed every second for each ad	Event: playheadPosition, playhead: 0 Event: playheadPosition, playhead: 1 Event: playheadPosition, playhead: 2...[playhead for each second]		
		1a.39.4	stop event called for each ad	**Event: stop**		
	1a.39 Midroll Ad	1a.39.5	correct event sequence is called for ad pod or each ad	Event: loadMetadata (at ad start) Event: playheadPosition (every second for ad) Event: stop		
		1a.40.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		
				clientid: [assigned Nielsen client ID - if client is overriding]	*provide value (if passed)*	

				subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*
				type: 'content'		*provide value*
				program: [program name]		*provide value*
				assetid: [unique asset ID]		*provide value*
				program: [program name]		*provide value*
				title: [episode title]		*provide value*
				length: [length of content in seconds] **Value for livestream: 86400 (if the value is unknown, please provide approximate length)**		*provide value*
			Content metadata is the **same as original metadata** passed during	**hasAds : "0": [Non-ad-supported subscription models.2** **"1": Ad-supported subscription models.] - to be tested for all available subscription models**		*provide value*
		1a.40.2	initial loadMetadata call	**c-variables**		*provide value*
		1a.40.3	new playhead position is accurate (*position increases)*	Event: playheadPosition, playhead: [new position]		
	1a.40 Content Resumes after Midroll	1a.40.4	playhead continues to passed from new position	Event: playheadPosition, playhead: [new position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
				Interruption Scenarios		
				SCENARIO 1 - LEAVE VIDEO DURING PREROLL (*if applicable*)		
			Provide a UNIX timestamp at start of preroll. Timestamp Example: D/NielsenAPPSDK(24766):			
		2a.1.1	[1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
	2a.1 Leave during Preroll	2a.1.2	playhead is not passed after leaving	Event: loadMetdata Event: playheadPosition, playhead: [position of content in seconds] Event: playheadPosition, playhead: [position of content in seconds] *Playheads do not continue to be sent*		
		2a.1.3	stop event called	**Event: stop** (no LoadMetadata Event should be called once the ad playback is resumed /in contrary to content/)		
		2a.1.4	playhead continues to passed from new position	Event: playheadPosition, playhead: [new position] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
				NO VIEW PING SHOULD BE SENT		
	2a.2 View Ping	2a.2.1	View ping NOT be sent if you leave during preroll	Pass = No view ping Fail = View ping		
				DURATION PING AFTER LEAVING DURING PREROLL		
				c27=cln, 0 [Value should be blank since you did not reach content]		*provide value*
			Duration ping fires after leaving	**c51= adl, [total number of seconds that the preroll played before leaving]**		*provide value*
	2a.3 Duration Ping	2a.3.1	during a preroll.	**c52= noad, [number of ads played before leaving]**		*provide value*
				SCENARIO 2		
				ALARM / BACKGROUND INTERRUPT		
			Provide a UNIX timestamp at start of content. Timestamp Example: D/NielsenAPPSDK(24766):			
		2a.4.1	[1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
	2a.4 Play Content	2a.4.2	playhead passed every second	Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		
				MOVE APP TO BACKGROUND		
			Provide a UNIX timestamp when moving app to background. Timestamp Example: D/NielsenAPPSDK(24766):			
		2a.5.1	[1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*
		2a.5.2	playhead is not passed when app is in background	Event: playheadPosition, playhead: [last position of content in seconds] *playhead does not continue to be sent*		
		2a.5.3	stop event	**Event: stop**		
	2a.5 App in Background	2a.5.4	no End call	**No end call. This leads to overcrediting when returning to foreground.**		
				DURATION PING AFTER MOVING APP TO BACKGROUND		
				c27=[number of seconds of content that played]		*provide value*
				c51= adl, [total number of seconds that played for the ads within any ad pods (if there were ads)]		*provide value*
			Duration ping fires after moving app	**c52= noad, [number of ads played within any ad pods (if there were**		
	2a.6 Duration Ping	2a.6.1	to background	**ads)]**		*provide value*
				RETURN APP TO FOREGROUND - PLAY CALL REQUIRED ONLY FOR AA 5.1.1.5 and AI 5.1.1.6 ONLY		
	2a.7 Play Call	2a.7.1	play event is called for content	Event: play, playerInfo (only required for versions under aa 5.1.1.5 and ai 5.1.1.6)		
				RETURN APP TO FOREGROUND AND RESUME PLAYING AFTER SETTING ALARM		
		2a.8.1	loadMetadata event is called for content	**Event: loadMetadata, metadata**		

			clientid: [assigned Nielsen client ID - if client is overriding]		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
			hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models		*provide value*	
		2a.8.2 Content metadata is the **same as original metadata** passed during initial loadMetadata call	**c-variables**		*provide value (if passed)*	
	2a.8 Content Metadata after Returning to Foreground	2a.8.3 playhead continues to be passed when App is returned to foreground	Event: playheadPosition, playhead: [last position of content in seconds] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
ALARM SOUNDS						
		2a.9.1 Provide a UNIX timestamp when alarm sounds. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*	
	2a.9 Hard Interrupt: Alarm	2a.9.2 playhead is not passed when Alarm sounds and interrupts playback	Event: playheadPosition, playhead: [last position of content in seconds] *playhead does not continue to be sent*			
		2a.9.3 stop event	**Event: stop**			
SILENCE ALARM AND WAIT 30 SECONDS BEFORE RESUMING PLAYBACK						
		2a.10.1 loadMetadata event is called for content	**Event: loadMetadata, metadata**			
			clientid: [assigned Nielsen client ID - if client is overriding]		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
		2a.10.2 Content metadata is the **same as original metadata** passed during initial loadMetadata call	**hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models**		*provide value*	
			c-variables		*provide value*	
	2a.10 Metadata after Alarm Sounds	2a.10.3 playhead continues to be passed after silencing alarm	Event: playheadPosition, playhead: [last position of content in seconds] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
SCENARIO 3 - ENABLE AIRPLANE MODE, THROTTLING IN CHARLES						
PLAY BUFFERED VIDEO UNTIL COMPLETION						
		2a.11.1 Provide a UNIX timestamp when buffering completes and logs stop passing. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*	
	2a.11 Network Loss: Turn on Airplane Mode	2a.11.2 correct event sequence is called for buffered video	Event: playheadPosition, playhead: [increments every second until buffered video completes]			
		2a.11.3 Stop called	**Event: Stop**			If stop is not called, make sure that playheads do not increment at the end of buffered content.
DISABLE AIRPLANE MODE/THROTTLING IN CHARLES AND WAIT 30 SECONDS BEFORE RESUMING VIDEO						
		2a.12.1 loadMetadata event is called for content	**Event: loadMetadata, metadata**			If loadMetadata is not called, make sure that playheads resume properly from the last position of buffered content (4.13.3) and that content is credited correctly in the next CAT ping.
			clientid: [assigned Nielsen client ID - if client is overriding]		*provide value (if passed)*	
			subbrand: [assigned Nielsen subbrand - if client is overriding]		*provide value (if passed)*	
			type: 'content'		*provide value*	
			assetid: [unique asset ID]		*provide value*	
			program: [program name]		*provide value*	
			title: [episode title]		*provide value*	
			length: [length of content in seconds] Value for livestream: 86400 (if the value is unknown, please provide approximate length)		*provide value*	
		2a.12.2 Content metadata is the **same as original metadata** passed during initial loadMetadata call	**hasAds : "0": [Non-ad-supported subscription models.2 "1": Ad-supported subscription models.] - to be tested for all available subscription models**		*provide value*	
			c-variables		*provide value*	

2a.12 Content Metadata after Network Restored	2a.12.3	playhead continues to be passed from position when Airplane mode is disabled	Event: playheadPosition, playhead: [last position of content in seconds] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
CACHED DURATION PING AFTER NETWORK RESTORATION						
2a.13 Duration Ping	2a.13.1	Duration ping fires after network connectivity is restored	**c27=[number of seconds of content that played including buffered duration]** **c51= adl, [total number of seconds that played for the ads within any ad pods (if there were ads)]** **c52= noad, [number of ads played within any ad pods (if there were ads)]**		*provide value* *provide value* *provide value*	
SCENARIO 4 (Dev Sanity Testing Only)						
PLUG HEADPHONES INTO DEVICE						
2a.14 Audio Change: Plugin headphones	2a.14.1	Provide a UNIX timestamp when plugging headphones into device. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*	
	2a.14.2	playhead continues to passed every second	Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...		*The SDK should behave according to the player. If playback continues, then playheads should continue. If playback stops, playheads should no longer increment and a stop call is recommended.*	
2a.15 Stop Call	2a.15.1	stop event after pause	**Event: stop (if playback stops)**			
RESUME PLAYBACK (if stopped)						
2a.16 Content Resumes if Stopped	2a.16.1	loadMetadata event is called for content	**Event: loadMetadata, metadata (when playback resumes)**			
	2a.16.2	playhead increments correctly	Event: playheadPosition, playhead: [last position of content in seconds] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
REMOVE HEADPHONES FROM DEVICE						
2a.17 Audio Change: Remove headphones	2a.17.1	Provide a UNIX timestamp when removing headphones from device. Timestamp Example: D/NielsenAPPSDK(24766): [1501168015] [DEBUG]	Please provide a UNIX Timestamp in the column H		*provide value*	
		playhead follows video player behavior	Event: playheadPosition, playhead: playheads should increment or not depending on player behavior]		*The SDK should behave according to the player. If playback continues, then playheads should continue. If playback stops, playheads should no longer increment and a stop call is recommended.*	
2a.18 Stop Call	2a.18.1	stop event after pause	**Event: stop (if playback stops)**			
RESUME PLAYBACK (if stopped)						
2a.19 Content Resumes if Stopped	2a.19.1	loadMetadata event is called for content	**Event: loadMetadata, metadata (when playback resumes)**			
	2a.19.2	playhead increments correctly	Event: playheadPosition, playhead: [last position of content in seconds] Event: playheadPosition, playhead: [position increments 1 second] Event: playheadPosition, playhead: [position increments 1 second] ...			
SCENARIO 5 USE SETTINGS FEATURES IF APPLICABLE (SUBTITLES, VIDEO QUALITY CHANGE, CHANGE AUDIO SETTINGS, ETC)						
2.a 20 Change settings during content playback (add subtitles, change video quality, etc.)	2.22.1	provide timestamp at the point where buffering has completed and the player has stopped	Copy timestamp or paste the line into column H. In the Console tab of your browser, run this command to get a timestamp: **Date('now')**.		*provide value*	
	2.22.2	Playheads stop after buffered video	Event: playheadPosition, playhead: [increments every 1 second] *Playhead does not continue to be sent*			
	2.22.3	stop event should be sent	**Event: stop**			
SCENARIO 6 USE SETTINGS FEATURES IF APPLICABLE (SUBTITLES, VIDEO QUALITY CHANGE, CHANGE AUDIO SETTINGS, ETC)						
2a 21 Change settings during content playback (add subtitles, change video quality, etc.)	2.22.1	provide timestamp at the point where buffering has completed and the player has stopped	Copy timestamp or paste the line into column H. In the Console tab of your browser, run this command to get a timestamp: **Date('now')**.		*provide value*	
	2.22.2	Playheads stop after buffered video	Event: playheadPosition, playhead: [increments every 1 second] *Playhead does not continue to be sent*			
	2.22.3	stop event should be sent	**Event: stop**			
SCENARIO 7 OPEN VIDEO IN FULLSCREEN						
	2.23.1	provide timestamp at the point where buffering has completed and the player has stopped	Copy timestamp or paste the line into column H. In the Console tab of your browser, run this command to get a timestamp: **Date('now')**.		*provide value*	

			2.23.2	Playheads stop after buffered video	Event: playheadPosition, playhead: [increments every 1 second] *Playhead does not continue to be sent*												
		2a 22 open fullscreen	2.23.3	stop event should be sent if video pauses playheads should be passed if video continues	**Event: stop** **Playheads should be passed if video continues**												
				SCENARIO 8 GO TO PICTURE IN PICTURE MODE													
			2.24.1	provide timestamp at the point where buffering has completed and the player has stopped	Copy timestamp or paste the line into column H. In the Console tab of your browser, run this command to get a timestamp: **Date('now')**.		*provide value*										
			2.24.2	Playheads stop after buffered video	Event: playheadPosition, playhead: [increments every 1 second] *Playhead does not continue to be sent*												
		2a 23 open fullscreen	2.24.3	stop event should be sent if video pauses playheads should be passed if video continues	**Event: stop** **Playheads should be passed if video continues**												
				SCENARIO 9 CLOSE PIP PER X													
			2.25.1	provide timestamp at the point where buffering has completed and the player has stopped	Copy timestamp or paste the line into column H. In the Console tab of your browser, run this command to get a timestamp: **Date('now')**.		*provide value*										
			2.25.2	Playheads stop after buffered video	Event: playheadPosition, playhead: [increments every 1 second] *Playhead does not continue to be sent*												
		2a 24 close PiP per X	2.25.3	end should be called if video closes stop should be called if videos pauses	Event: end Note: call end if video is closed after hitting X. Event: stop Note: call stop if video continues												