

OUTPUT LOCKING: AWS ELEMENTAL LIVE 2.11 AND LATER





CONTENTS

How Output Locking Works.....	3
Output Failure Redundancy	3
Distributed Encoding.....	3
Input Requirements	4
Output Requirements.....	4
Setting up the Events.....	4
Input Fields	4
Global Processors Fields for Multicast.....	5
Global Processors Fields for Unicast.....	5
Archive Output Fields	5
HLS Output Fields.....	5
MSS Output Fields	5
Stream > Video Fields	6



The output locking feature has been redesigned for AWS Elemental Live 2.11 and later.

This feature is a way to guarantee that events on multiple nodes (appliances) can produce video outputs that are frame-accurate with each other. It can be used to support output failure redundancy or to support distributed encoding.

Output locking is supported with Archive, HLS, Microsoft Smooth and UDP/TS output groups. Note that it is not supported with the Reliable TS output group that is introduced with Elemental Live 2.14.

HOW OUTPUT LOCKING WORKS

The event on one of the locked encoders always starts encoding earlier than the other event or events (even if it is only a couple of frames earlier). This event is the “leader”. When subsequent events start, each will drop encoded frames until it gets to a segment boundary. It will obtain the master event’s embedded timecode and then lock to that timecode to remain synchronized through the life of the event.

The events communicate with each other (over multicast or unicast) in order to share the data required to ensure that their outputs are locked. Multicast is supported in version 2.11 and later. Unicast is supported in version 2.13 and later.

OUTPUT FAILURE REDUNDANCY

Two events on two different nodes are locked to each other so that they communicate as describe above. These two events are set up to ingest the same input, to perform the same processing (they have identical video encoding parameters). The outputs are locked to each other, which means the same frame has the same timecode in both events.

The events are set up to publish to the same downstream packager. The packager is receiving both outputs but it uses only one of them. If that output becomes unavailable or unusable, the packager switches to the second output. The packager can use the timecode from one output to synchronize the frames from that output to the corresponding frames in the output of the other event.

You can use multicast addresses or a unicast addresses to set up output locking for output failure redundancy.

You can lock any number of pairs of events. For example, event A (on node 1) and event B (on node 2) as a redundant pair, events C (on node 1) and D (on node 2) as another redundant pair, and so on.

DISTRIBUTED ENCODING

One example of distributed encoding is in building an ABR stack. Two or more events (each event on a different node) have the same input but each event encodes a different part of the ABR stack: each event creates one or more outputs.

The events do not necessarily specify identical encoding. But the events are locked together so that they communicate as described above. The locking means that the same frame from the output or outputs in one event has the same timecode as the output or outputs in all the other events.

The events publish to the same downstream packager, where the ABR stack is put together. The packager can use the timecode from one output to synchronize the frame from the outputs from one event to the corresponding frame in the outputs from the other event.



You can use multicast addresses or a unicast addresses to set up output locking for distributed encoding:

- If you use multicast addresses, you can lock together any number of events. For example, event A, event B, and event C (probably each on a separate node) are all locked together and are communicating over multicast.
- If you use unicast addresses, you can lock together only two events.

INPUT REQUIREMENTS

- Input type: SDI or transport stream or HLS (network only) source. File sources do not support output locking.
- Resolution: The inputs within one event must have the same resolution. The input for one event can have a different resolution to the input for another event.
- GOP: The inputs within one event and between two events must have the same GOP structure and framerate.
- Input timecode: Required. An SDI source must have an embedded or LTC timecode. A transport stream source or HLS source must have an embedded timecode.

OUTPUT REQUIREMENTS

- Output types:
 - Microsoft Smooth output (version 2.11 and later)
 - UDP/TS output (version 2.11 and later)
 - MPEG-2 TS container in an Archive output (version 2.11 and later)
 - HLS/TS and HLS/fMP4 (version 2.14 and later)
- All video encodes must use H.264 or H.265.
- Framerate: Within one event, the framerate in and out must be the same. Between two events, the framerate can be different but one must be a whole-number multiple of the other.

SETTING UP THE EVENTS

INPUT FIELDS

Input > Video Selector	Input name		
	Timecode Source	For SDI inputs, choose Embedded or LTC. For other inputs, choose Embedded.	AWS Elemental Live uses this timecode to lock the events.
Timecode Configuration	Source	Choose any value, but choose the same value across all events.	This timecode is the timecode that is inserted in the output. The downstream packager can use this timecode to determine that a frame from one event corresponds to the frame from another event.
	Sync Threshold	Uncheck this field.	
	Require Initial Timecode: Checked	Check this field to ensure that the input does in fact have a timecode.	



GLOBAL PROCESSORS FIELDS FOR MULTICAST

Global Processors > Output Locking	Multicast	Check this field.	
Global Processors > Output Locking	Address Port (optional) Interface (optional)	Enter the multicast address of any server. Enter identical address, port, interface across all events you want to lock together. See the tooltips for more information.	The events communicate with each other via this address.

GLOBAL PROCESSORS FIELDS FOR UNICAST

Global Processors > Output Locking	Multicast	Uncheck this field.	
Global Processors > Output Locking	Send Address Send Port (required) Send Interface (optional)	Enter the address where the other event will be listening. See the tooltips for more information. To lock together more than one pair of events, make sure each pair has a different port from the other pair.	This event sends message to the other event via this address.
Global Processors > Output Locking	Receive Port (required) Receive Interface (optional)	Enter the address where this event will be listening. See the tooltips for more information. To lock together more than one pair of events, make sure each pair has a different port from the other pair.	The other event sends message to this event via this address.

ARCHIVE OUTPUT FIELDS

Output Groups > Archive	Destination		
Output Groups > Archive > Output	Container	Choose MPEG-2 Transport Stream	

HLS OUTPUT FIELDS

Output Groups > HLS	Custom Group Name	Enter the same name across all events	
Output Groups > HLS > Output	Segment Type	Choose TS or fMP4	

MSS OUTPUT FIELDS

Output Groups >	Custom Group Name	Enter the same name across all events	
	Use Event ID	Unchecked	
	Send EOS	Unchecked	
	Send Delay	Complete as desired. See the tooltip for details.	
	Fragment length	This field can be different in each event.	



STREAM > VIDEO FIELDS

Stream > Video	Codec	H.264 or H.265	Must be the same codec in all events.
	Framerate	Ensure that the values you set in the events are whole-number multiples of each other.	
	Segmentation Markers	None or EBP CableLabs.	