

[Accelerate](#) / BNNSGraphContextMakeStreaming(\_:\_:\_:)

## Function

# BNNSGraphContextMakeStreaming(\_:\_:\_:\_:)

Returns an allocated and initialized graph context with streaming support from the specified graph.

iOS 18.0+ | iPadOS 18.0+ | Mac Catalyst 18.0+ | macOS 15.0+ | tvOS 18.0+ | visionOS 2.0+ | watchOS 11.0+

```
func BNNSGraphContextMakeStreaming(
    _ graph: bnns_graph_t,
    _ function: UnsafePointer<CChar>?,
    _ initial_states_count: Int,
    _ initial_states: UnsafePointer<BNNSTensor>?
) -> bnns_graph_context_t
```

## Parameters

### graph

The compiled graph object.

### function

The function that the new context initializes the state for. Specify as `nil` if the graph only contains one function.

### initial\_states\_count

The number of elements in the `initial_states_count` array.

### initial\_states

An array of BNNSTensor structures that describe the data that the context uses to initialize each state. The context uses `initial_states[i]` to initialize the state with the name

```
initial_state[i]->name.
```

## Return Value

A compiled graph context object. If the operation fails, the graph object's data property is `nil`.

## Discussion

In addition to the work that `BNNSTensorContextMake( : )` performs, this call allocates ring-buffer backed memory for all Core ML state arguments of the given function.

If your model runs on a stream of data, such as processing audio data, it may benefit from working on one frame at a time. You can express this functionality through a model that uses Core ML's concept of states. In this case, BNNS stores information that the model needs from a previous frame in the state that dilations on convolution layers may require.

If your model uses the pattern below, use this function to specify that BNNS uses an optimized mode that allocates and manages ring buffers for states. Doing so eliminates memory copies associated with the approach.

```
output_state = concat(input_state, tensorX)
```

BNNS advances the ring buffer by the size of `tensorX` in the concatenation dimension after the function executes.

To use a context with streaming support, your function must include an attribute dictionary named `BNNSOptions` that includes the entry { `'StateMode': 'Streaming'` }. This specifies that the `BNNSTensorCompileFromFile( : : : )` function encodes additional metadata into the graph that describes the streaming rate.

Calls to `BNNSTensorContextExecute( : : : : : )` that use contexts this function creates ignore any user-provided pointers for input and output arguments. Instead, context execution uses the internal ring buffer and updates any nonnull input and output arguments to point to the same ring-buffer backed memory. On return, an analysis of the compiled program for use in the next frame determines the distance by which the operation advances the ring buffer.

To prevent memory leaks, call `BNNSTensorContextDestroy( : )` when you're finished using the graph context.

---

## See Also

## Creating and destroying a context

```
struct bnns_graph_context_t
```

An object that wraps a compiled graph object.

```
func BNNSGraphContextMake(bnns_graph_t) -> bnns_graph_context_t
```

Returns an allocated and initialized graph context from the specified graph.

```
func BNNSGraphContextDestroy(bnns_graph_context_t)
```

Destroys the specified graph context.