

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

## Function

# BNNSGraphContextSetStreamingAdvanceCount(\_:\_:)

Sets the streaming advancement amount for cases with dynamically shaped inputs.

iOS 18.4+ | iPadOS 18.4+ | Mac Catalyst 18.4+ | macOS 15.4+ | tvOS 18.4+ | visionOS 2.4+ | watchOS 11.4+

```
func BNNSGraphContextSetStreamingAdvanceCount(
    _ context: bnns_graph_context_t,
    _ advance_count: Int
) -> Int32
```

## Parameters

### context

The graph context.

### advance\_count

An integer value that specifies the number of elements by which the function advances the internal state pointer.

## Discussion

BNNS can't unambiguously determine the streaming advancement size for models you compile with the BNNSOption attribute `StateMode=Streaming` in an enabled state, where `slice_update` operations use an update parameter of dynamic shape. Call this function before calling `BNNSGraphContextExecute( : : : : : )` to set the advancement size for each frame.

This function advances the internal state pointer by `advance_count` elements in the streaming dimension before returning from `BNNSGraphContextExecute( : : : : : )`.

#### Note

The BNNS streaming APIs do not support models that require different advancement amounts for different states.

## See Also

### Specifying and querying a context's properties

```
func BNNSGraphContextSetArgumentType(bnns_graph_context_t, BNNSGraphArgumentType) -> Int32
```

Specifies the argument type for a graph context.

```
struct BNNSGraphArgumentType
```

Constants that specify the argument type for a graph context.

```
func BNNSGraphContextSetDynamicShapes(bnns_graph_context_t, UnsafePointer<CChar>?, Int, UnsafeMutablePointer<bnns_graph_shape_t>) -> Int32
```

Specifies the dynamic shapes for a graph and, if possible, infers, the output shapes.

```
struct bnns_graph_shape_t
```

The specification of the shape of an argument.

```
func BNNSGraphContextSetBatchSize(bnns_graph_context_t, UnsafePointer<CChar>?, UInt64) -> Int32
```

Sets the batch size for a graph.

```
func BNNSGraphContextEnableNanAndInfChecks(bnns_graph_context_t, Bool)
```

Specifies that the context checks intermediate tensors for NaNs and infinities.

```
func BNNSGraphContextGetWorkspaceSize(bnns_graph_context_t, UnsafePointer<CChar>?) -> Int
```

Returns the minimum size, in bytes, of the workspace that graph context execution requires.