

L2ToL1MessageNitro

Classes

L2ToL1MessageNitro

Base functionality for nitro L2->L1 messages

Extended by

- [L2ToL1MessageReaderNitro](#)

Methods

fromEvent()

static

fromEvent < T

(I1SignerOrProvider :

T , event : object , I1Provider ? : Provider) : L2ToL1MessageReaderOrWriterNitro < T

Instantiates a newL2ToL1MessageWriterNitro orL2ToL1MessageReaderNitro object.

Type parameters

Type parameter T extends SignerOrProvider

Parameters

Parameter Type Description I1SignerOrProvider T Signer or provider to be used for executing or reading the L2-to-L1 message. event object The event containing the data of the L2-to-L1 message. event.arbBlockNum BigNumber - event.caller ? string - event.callvalue ? BigNumber - event.data ? string - event.destination ? string - event.ethBlockNum ? BigNumber - event.hash ? BigNumber - event.position ? BigNumber - event.timestamp ? BigNumber - I1Provider ? Provider Optional. Used to override the Provider which is attached toI1SignerOrProvider in case you need more control. This will be a required parameter in a future major version update.

Returns

[L2ToL1MessageReaderOrWriterNitro](#) <T

Source

[message/L2ToL1MessageNitro.ts:136](#)

L2ToL1MessageReaderNitro

Provides read-only access nitro for I2-to-I1-messages

Extends

- [L2ToL1MessageNitro](#)

Extended by

- [L2ToL1MessageWriterNitro](#)

Methods

getFirstExecutableBlock()

getFirstExecutableBlock (I2Provider : Provider) :

Promise < null

| BigNumber

Estimates the L1 block number in which this L2 to L1 tx will be available for execution. If the message can or already has been executed, this returns null

Parameters

Parameter Type Description I2Provider Provider

Returns

Promise <null |BigNumber

expected L1 block number where the L2 to L1 message will be executable. Returns null if the message can be or already has been executed

Source

[message/L2ToL1MessageNitro.ts:419](#)

hasExecuted()

protected

hasExecuted (I2Provider : Provider) :

Promise < boolean

Check if this message has already been executed in the Outbox

Parameters

Parameter Type I2Provider Provider

Returns

Promise <boolean

Source

[message/L2ToL1MessageNitro.ts:207](#)

status()

status (I2Provider : Provider) :

Promise < L2ToL1MessageStatus

Get the status of this message In order to check if the message has been executed proof info must be provided.

Parameters

Parameter Type I2Provider Provider

Returns

Promise <L2ToL1MessageStatus

Source

[message/L2ToL1MessageNitro.ts:222](#)

waitUntilReadyToExecute()

waitUntilReadyToExecute (I2Provider : Provider , retryDelay :

number) :

Promise < CONFIRMED

|

EXECUTED

Waits until the outbox entry has been created, and will not return until it has been. WARNING: Outbox entries are only created when the corresponding node is confirmed. Which can take 1 week+, so waiting here could be a very long operation.

Parameters

Parameter	Type	Default value	Description
I2Provider	Provider	undefined	- retryDelay number 500

Returns

Promise <CONFIRMED |EXECUTED

outbox entry status (either executed or confirmed but not pending)

Source

[message/L2ToL1MessageNitro.ts:397](#)

fromEvent()

static

fromEvent < T

(I1SignerOrProvider :

T , event : object , I1Provider ? : Provider) : L2ToL1MessageReaderOrWriterNitro < T

Instantiates a newL2ToL1MessageWriterNitro orL2ToL1MessageReaderNitro object.

Type parameters

Type parameter T extends SignerOrProvider

Parameters

Parameter	Type	Description
I1SignerOrProvider	T	Signer or provider to be used for executing or reading the L2-to-L1 message.
event	object	The event containing the data of the L2-to-L1 message.
event.arbBlockNum	BigNumber	-
event.caller	? string	- event.callvalue ? BigNumber - event.data ? string - event.destination ? string - event.ethBlockNum ? BigNumber - event.hash ? BigNumber - event.position ? BigNumber - event.timestamp ? BigNumber - I1Provider ? Provider
I1Provider	? Provider	Optional. Used to override the Provider which is attached toI1SignerOrProvider in case you need more control. This will be a required parameter in a future major version update.

Returns

[L2ToL1MessageReaderOrWriterNitro](#) <T

Inherited from

[L2ToL1MessageNitro](#) .fromEvent

Source

[message/L2ToL1MessageNitro.ts:136](#)

L2ToL1MessageWriterNitro

Provides read and write access for nitro I2-to-I1-messages

Extends

- [L2ToL1MessageReaderNitro](#)

Constructors

new L2ToL1MessageWriterNitro()

new

L2ToL1MessageWriterNitro (I1Signer : Signer , event : object , I1Provider ? : Provider) : L2ToL1MessageWriterNitro
Instantiates a newL2ToL1MessageWriterNitro object.

Parameters

Parameter Type Description I1Signer Signer The signer to be used for executing the L2-to-L1 message. event object The event containing the data of the L2-to-L1 message. event.arbBlockNum BigNumber - event.caller ? string - event.callvalue ? BigNumber - event.data ? string - event.destination ? string - event.ethBlockNum ? BigNumber - event.hash ? BigNumber - event.position ? BigNumber - event.timestamp ? BigNumber - I1Provider ? Provider Optional. Used to override the Provider which is attached toI1Signer in case you need more control. This will be a required parameter in a future major version update.

Returns

[L2ToL1MessageWriterNitro](#)

Overrides

L2ToL1MessageReaderNitro.constructor

Source

[message/L2ToL1MessageNitro.ts:514](#)

Properties

Property Modifier Type Description I1Signer private Signer The signer to be used for executing the L2-to-L1 message.

Methods

execute()

execute (I2Provider : Provider , overrides ? : Overrides) :

Promise < ContractTransaction

Executes the L2ToL1Message on L1. Will throw an error if the outbox entry has not been created, which happens when the corresponding assertion is confirmed.

Parameters

Parameter Type I2Provider Provider overrides ? Overrides

Returns

Promise <ContractTransaction

Source

[message/L2ToL1MessageNitro.ts:528](#)

getFirstExecutableBlock()

getFirstExecutableBlock (I2Provider : Provider) :

Promise < null

| BigNumber

Estimates the L1 block number in which this L2 to L1 tx will be available for execution. If the message can or already has been executed, this returns null

Parameters

Parameter Type Description I2Provider Provider

Returns

Promise <null | BigNumber

expected L1 block number where the L2 to L1 message will be executable. Returns null if the message can be or already has been executed

Inherited from

[L2ToL1MessageReaderNitro .getFirstExecutableBlock](#)

Source

[message/L2ToL1MessageNitro.ts:419](#)

hasExecuted()

protected

hasExecuted (I2Provider : Provider) :

Promise < boolean

Check if this message has already been executed in the Outbox

Parameters

Parameter Type I2Provider Provider

Returns

Promise <boolean

Inherited from

[L2ToL1MessageReaderNitro .hasExecuted](#)

Source

[message/L2ToL1MessageNitro.ts:207](#)

status()

status (I2Provider : Provider) :

Promise < L2ToL1MessageStatus

Get the status of this message In order to check if the message has been executed proof info must be provided.

Parameters

Parameter Type I2Provider Provider

Returns

Promise <L2ToL1MessageStatus

Inherited from

[L2ToL1MessageReaderNitro .status](#)

Source

[message/L2ToL1MessageNitro.ts:222](#)

waitUntilReadyToExecute()

waitUntilReadyToExecute (I2Provider : Provider , retryDelay :
number) :

Promise < CONFIRMED

|

EXECUTED

Waits until the outbox entry has been created, and will not return until it has been. WARNING: Outbox entries are only created when the corresponding node is confirmed. Which can take 1 week+, so waiting here could be a very long operation.

Parameters

Parameter	Type	Default value	Description
I2Provider	Provider	undefined	retryDelay number 500

Returns

Promise <CONFIRMED |EXECUTED

outbox entry status (either executed or confirmed but not pending)

Inherited from

[L2ToL1MessageReaderNitro](#) .[waitUntilReadyToExecute](#)

Source

[message/L2ToL1MessageNitro.ts:397](#)

fromEvent()

static

fromEvent < T

(I1SignerOrProvider :

T , event : object , I1Provider ? : Provider) : L2ToL1MessageReaderOrWriterNitro < T

Instantiates a newL2ToL1MessageWriterNitro orL2ToL1MessageReaderNitro object.

Type parameters

Type parameter T extends SignerOrProvider

Parameters

Parameter	Type	Description
I1SignerOrProvider	T	Signer or provider to be used for executing or reading the L2-to-L1 message.
event	object	The event containing the data of the L2-to-L1 message.
event.arbBlockNum	BigNumber	-
event.caller	? string	- event.callvalue ? BigNumber - event.data ? string - event.destination ? string - event.ethBlockNum ? BigNumber - event.hash ? BigNumber - event.position ? BigNumber - event.timestamp ? BigNumber - I1Provider ? Provider
I1Provider	? Provider	Optional. Used to override the Provider which is attached toI1SignerOrProvider in case you need more control. This will be a required parameter in a future major version update.

Returns

[L2ToL1MessageReaderOrWriterNitro](#) <T

Inherited from

[L2ToL1MessageReaderNitro.fromEvent](#)

Source

[message/L2ToL1MessageNitro.ts:136](#)

Type Aliases

L2ToL1MessageReaderOrWriterNitro

type

L2ToL1MessageReaderOrWriterNitro < T

:

T

extends

Provider

? L2ToL1MessageReaderNitro : L2ToL1MessageWriterNitro ; Conditional type for Signer or Provider. If T is of type Provider then L2ToL1MessageReaderOrWriter will be of type L2ToL1MessageReader. If T is of type Signer then L2ToL1MessageReaderOrWriter will be of type L2ToL1MessageWriter.

Type parameters

Type parameter T extends SignerOrProvider

Source

[message/L2ToL1MessageNitro.ts:57](#) [Edit this page](#) [Previous L2ToL1MessageClassic](#) [Next L2Transaction](#)