

# L2ToL1MessageClassic

## Classes

### L2ToL1MessageReaderClassic

Provides read-only access for classic l2-to-l1-messages

#### Extends

- L2ToL1MessageClassic

#### Extended by

- [L2ToL1MessageWriterClassic](#)

#### Properties

Property	Modifier	Type	Default value	Description
Inherited from batchNumber	readonly	BigNumber	undefined	The number of the batch this message is part of
L2ToL1MessageClassic.batchNumber	readonly	BigNumber	undefined	The index of this message in the batch
L2ToL1MessageClassic.indexInBatch	protected	null   string	null	Contains the classic outbox address, or set to zero address if this network did not have a classic outbox deployed -

#### 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

##### Parameters

Parameter	Type	Description
I2Provider	Provider	

##### Returns

Promise < null | BigNumber

Always returns null for classic l2toL1 messages since they can be executed in any block now.

##### Source

[message/L2ToL1MessageClassic.ts:374](#)

##### getOutboxAddress()

protected

getOutboxAddress ( I2Provider : Provider , batchNumber : number ) :

Promise < string

Classic had 2 outboxes, we need to find the correct one for the provided batch number

##### Parameters

Parameter	Type	Description
I2Provider	Provider	
batchNumber	number	

##### Returns

Promise <string

**Source**

[message/L2ToL1MessageClassic.ts:206](#)

**hasExecuted()**

hasExecuted ( I2Provider : Provider ) :

Promise < boolean

Check if given outbox message has already been executed

**Parameters**

Parameter Type I2Provider Provider

**Returns**

Promise <boolean

**Source**

[message/L2ToL1MessageClassic.ts:293](#)

**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/L2ToL1MessageClassic.ts:331](#)

**tryGetProof()**

tryGetProof ( I2Provider : Provider ) :

Promise < null

| MessageBatchProofInfo

Get the execution proof for this message. Returns null if the batch does not exist yet.

**Parameters**

Parameter Type Description I2Provider Provider

**Returns**

Promise <null |MessageBatchProofInfo

**Source**

[message/L2ToL1MessageClassic.ts:277](#)

**waitUntilOutboxEntryCreated()**

waitUntilOutboxEntryCreated ( 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/L2ToL1MessageClassic.ts:354](#)

**fromBatchNumber()**

static

fromBatchNumber < T

( I1SignerOrProvider :

T , batchSize : BigNumber , indexInBatch : BigNumber , I1Provider ? : Provider ) :

L2ToL1MessageReaderOrWriterClassic < T

Instantiates a newL2ToL1MessageWriterClassic orL2ToL1MessageReaderClassic 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. batchSize BigNumber The number of the batch containing the L2-to-L1 message. indexInBatch BigNumber The index of the L2-to-L1 message within the batch. 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

[L2ToL1MessageReaderOrWriterClassic](#) <T

#### Inherited from

L2ToL1MessageClassic.fromBatchNumber

#### Source

[message/L2ToL1MessageClassic.ts:125](#)

**L2ToL1MessageWriterClassic**

Provides read and write access for classic L2-to-L1-messages

## Extends

- [L2ToL1MessageReaderClassic](#)

## Constructors

**new L2ToL1MessageWriterClassic()**

new

L2ToL1MessageWriterClassic ( I1Signer : Signer , batchSize : BigNumber , indexInBatch : BigNumber , I1Provider ? : Provider ) : L2ToL1MessageWriterClassic Instantiates a newL2ToL1MessageWriterClassic object.

### Parameters

Parameter Type Description I1Signer Signer The signer to be used for executing the L2-to-L1 message. batchSize BigNumber The number of the batch containing the L2-to-L1 message. indexInBatch BigNumber The index of the L2-to-L1 message within the batch. 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

[L2ToL1MessageWriterClassic](#)

### Overrides

L2ToL1MessageReaderClassic.constructor

### Source

[message/L2ToL1MessageClassic.ts:394](#)

## Properties

Property Modifier Type Default value Description Inherited from batchSize readonly BigNumber undefined The number of the batch this message is part of [L2ToL1MessageReaderClassic](#) .batchNumber indexInBatch readonly BigNumber undefined The index of this message in the batch [L2ToL1MessageReaderClassic](#) .indexInBatch I1Signer private Signer undefined The signer to be used for executing the L2-to-L1 message. - outboxAddress protected null |string null Contains the classic outbox address, or set to zero address if this network did not have a classic outbox deployed [L2ToL1MessageReaderClassic](#) .outboxAddress

## 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/L2ToL1MessageClassic.ts:409](#)

**getFirstExecutableBlock()**

getFirstExecutableBlock ( I2Provider : Provider ) :

Promise < null

| BigNumber

Estimates the L1 block number in which this L2 to L1 tx will be available for execution

**Parameters**

Parameter Type Description I2Provider Provider

**Returns**

Promise <null |BigNumber

Always returns null for classic I2toL1 messages since they can be executed in any block now.

**Inherited from**

[L2ToL1MessageReaderClassic](#) .getFirstExecutableBlock

**Source**

[message/L2ToL1MessageClassic.ts:374](#)

**getOutboxAddress()**

protected

getOutboxAddress ( I2Provider : Provider , batchNumber :

number ) :

Promise < string

Classic had 2 outboxes, we need to find the correct one for the provided batch number

**Parameters**

Parameter Type Description I2Provider Provider batchNumber number

**Returns**

Promise <string

**Inherited from**

[L2ToL1MessageReaderClassic](#) .getOutboxAddress

**Source**

[message/L2ToL1MessageClassic.ts:206](#)

**hasExecuted()**

hasExecuted ( I2Provider : Provider ) :

Promise < boolean

Check if given outbox message has already been executed

**Parameters**

Parameter Type I2Provider Provider

#### Returns

Promise <boolean

#### Inherited from

[L2ToL1MessageReaderClassic](#) .hasExecuted

#### Source

[message/L2ToL1MessageClassic.ts:293](#)

#### 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

[L2ToL1MessageReaderClassic](#) .status

#### Source

[message/L2ToL1MessageClassic.ts:331](#)

#### tryGetProof()

tryGetProof ( I2Provider : Provider ) :

Promise < null

| MessageBatchProofInfo

Get the execution proof for this message. Returns null if the batch does not exist yet.

#### Parameters

Parameter Type Description I2Provider Provider

#### Returns

Promise <null |MessageBatchProofInfo

#### Inherited from

[L2ToL1MessageReaderClassic](#) .tryGetProof

#### Source

[message/L2ToL1MessageClassic.ts:277](#)

#### waitUntilOutboxEntryCreated()

waitUntilOutboxEntryCreated ( 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

[L2ToL1MessageReaderClassic](#) .[waitUntilOutboxEntryCreated](#)

#### Source

[message/L2ToL1MessageClassic.ts:354](#)

**fromBatchNumber()**

static

fromBatchNumber < T

( I1SignerOrProvider :

T , batchSize : BigNumber , indexInBatch : BigNumber , I1Provider ? : Provider ) :

L2ToL1MessageReaderOrWriterClassic < T

Instantiates a newL2ToL1MessageWriterClassic orL2ToL1MessageReaderClassic 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.
batchNumber	BigNumber	The number of the batch containing the L2-to-L1 message.
indexInBatch	BigNumber	The index of the L2-to-L1 message within the batch.
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

[L2ToL1MessageReaderOrWriterClassic](#) <T

#### Inherited from

[L2ToL1MessageReaderClassic](#) .[fromBatchNumber](#)

#### Source

[message/L2ToL1MessageClassic.ts:125](#)

## Type Aliases

## L2ToL1MessageReaderOrWriterClassic

type  
L2ToL1MessageReaderOrWriterClassic < T  
:  
T  
extends  
Provider  
? L2ToL1MessageReaderClassic : L2ToL1MessageWriterClassic ; 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/L2ToL1MessageClassic.ts:98](#) [Edit this page](#) [Previous L2ToL1Message](#) [Next L2ToL1MessageNitro](#)