# ChildToParentMessage

## Classes

### ChildToParentMessage

Base functionality for Child-to-Parent messages

**Extended by**

- [ChildToParentMessageReader](ChildToParentMessageReader)

**Methods**

**fromEvent()**

static

fromEvent < T

( parentSignerOrProvider :

T , event : ChildToParentTransactionEvent , parentProvider ? : Provider ) : ChildToParentMessageReaderOrWriter < T

Instantiates a newChildToParentMessageWriter orChildToParentMessageReader object.

**Type parameters**

Type parameter T extends SignerOrProvider

**Parameters**

Parameter Type Description parentSignerOrProvider T - event ChildToParentTransactionEvent The event containing the data of the Child-to-Parent message. parentProvider ? Provider -

**Returns**

[ChildToParentMessageReaderOrWriter](ChildToParentMessageReaderOrWriter) <T

**Source**

[message/ChildToParentMessage.ts:76](message/ChildToParentMessage.ts:76)

**getChildToParentEvents()**

static

getChildToParentEvents ( childProvider : Provider , filter : object , position ? : BigNumber , destination ? :

string , hash ? : BigNumber , indexInBatch ? : BigNumber ) :

Promise < ChildToParentTransactionEvent & object [ ]

Get event logs for ChildToParent transactions.

**Parameters**

Parameter Type Description childProvider Provider filter object Block range filter filter.fromBlock BlockTag - filter.toBlock ? BlockTag - position ? BigNumber The batchnumber indexed field was removed in nitro and a position indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same batchnumber. For post nitro events it will be used to find events with the same position. destination ? string The parent destination of the ChildToParent message hash ? BigNumber The uniqueId indexed field was removed in nitro and a hash indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same uniqueId. For post nitro events it will be used to find events with the same hash. indexInBatch ? BigNumber The index in the batch, only valid for pre-nitro events. This parameter is ignored post-nitro

Promise

Any classic and nitro events that match the provided filters.

**Source**

[message/ChildToParentMessage.ts:109](message/ChildToParentMessage.ts:109)

# ChildToParentMessageReader

Provides read-only access for Child-to-Parent messages

## Extends

* [ChildToParentMessage](ChildToParentMessage)

## Extended by

* [ChildToParentMessageWriter](ChildToParentMessageWriter)

## Methods

**getFirstExecutableBlock()**

getFirstExecutableBlock ( childProvider : Provider ) :

Promise < null

| BigNumber

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

**Parameters**

Parameter Type Description childProvider Provider

**Returns**

Promise <null |BigNumber

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

**Source**

[message/ChildToParentMessage.ts:273](message/ChildToParentMessage.ts:273)

**status()**

status ( childProvider : Provider ) :

Promise < ChildToParentMessageStatus

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

**Parameters**

Parameter Type childProvider Provider

**Returns**

Promise <ChildToParentMessageStatus

**Source**

**waitUntilReadyToExecute()**

waitUntilReadyToExecute ( childProvider : 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 childProvider Provider undefined - retryDelay number 500

**Returns**

Promise <CONFIRMED |EXECUTED

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

**Source**

**fromEvent()**

static

fromEvent < T

> ( parentSignerOrProvider :

T , event : ChildToParentTransactionEvent , parentProvider ? : Provider ) : ChildToParentMessageReaderOrWriter < T

> Instantiates a newChildToParentMessageWriter orChildToParentMessageReader object.

**Type parameters**

Type parameter T extends SignerOrProvider

**Parameters**

Parameter Type Description parentSignerOrProvider T - event ChildToParentTransactionEvent The event containing the data of the Child-to-Parent message. parentProvider ? Provider -

**Returns**

[ChildToParentMessageReaderOrWriter](#) <T

**Inherited from**

[ChildToParentMessage](#) [.fromEvent](#)

**Source**

**getChildToParentEvents()**

static

getChildToParentEvents ( childProvider : Provider , filter : object , position ? : BigNumber , destination ? :

string , hash ? : BigNumber , indexInBatch ? : BigNumber ) :

Promise < ChildToParentTransactionEvent & object [ ]

Get event logs for ChildToParent transactions.

**Parameters**

Parameter Type Description childProvider Provider filter object Block range filter filter.fromBlock BlockTag - filter.toBlock ? BlockTag - position ? BigNumber The batchnumber indexed field was removed in nitro and a position indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same batchnumber. For post nitro events it will be used to find events with the same position. destination ? string The parent destination of the ChildToParent message hash ? BigNumber The uniqueId indexed field was removed in nitro and a hash indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same uniqueId. For post nitro events it will be used to find events with the same hash. indexInBatch ? BigNumber The index in the batch, only valid for pre-nitro events. This parameter is ignored post-nitro

**Returns**

Promise

Any classic and nitro events that match the provided filters.

**Inherited from**

[ChildToParentMessage](#) [.getChildToParentEvents](#)

**Source**

[message/ChildToParentMessage.ts:109](#)

# ChildToParentMessageWriter

Provides read and write access for Child-to-Parent messages

## Extends

- [ChildToParentMessageReader](#)

## Constructors

**new ChildToParentMessageWriter()**

new

ChildToParentMessageWriter ( parentSigner : Signer , event : ChildToParentTransactionEvent , parentProvider ? : Provider ) : ChildToParentMessageWriter Instantiates a newChildToParentMessageWriter object.

**Parameters**

Parameter Type Description parentSigner Signer - event ChildToParentTransactionEvent The event containing the data of the Child-to-Parent message. parentProvider ? Provider -

**Returns**

[ChildToParentMessageWriter](#)

**Overrides**

ChildToParentMessageReader.constructor

**Source**

[message/ChildToParentMessage.ts:296](#)

**Methods**

**execute()**

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

Promise < ContractTransaction

> Executes the ChildToParentMessage on Parent chain. Will throw an error if the outbox entry has not been
> created, which happens when the corresponding assertion is confirmed.

**Parameters**

Parameter Type childProvider Provider overrides ? Overrides

**Returns**

Promise <ContractTransaction

**Source**

message/ChildToParentMessage.ts:325

**getFirstExecutableBlock()**

getFirstExecutableBlock ( childProvider : Provider ) :

Promise < null

| BigNumber

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

**Parameters**

Parameter Type Description childProvider Provider

**Returns**

Promise <null |BigNumber

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

**Inherited from**

ChildToParentMessageReader .getFirstExecutableBlock

**Source**

message/ChildToParentMessage.ts:273

**status()**

status ( childProvider : Provider ) :

Promise < ChildToParentMessageStatus

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

**Parameters**

Parameter Type childProvider Provider

**Returns**

Promise <ChildToParentMessageStatus

**Inherited from**

[ChildToParentMessageReader](#) [.status](#)

**Source**

[message/ChildToParentMessage.ts:237](#)

**waitUntilReadyToExecute()**

waitUntilReadyToExecute ( childProvider : 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 childProvider Provider undefined - retryDelay number 500

**Returns**

Promise <CONFIRMED |EXECUTED

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

**Inherited from**

[ChildToParentMessageReader](#) [.waitUntilReadyToExecute](#)

**Source**

[message/ChildToParentMessage.ts:252](#)

**fromEvent()**

static

fromEvent < T

> ( parentSignerOrProvider :

T , event : ChildToParentTransactionEvent , parentProvider ? : Provider ) : ChildToParentMessageReaderOrWriter < T

> Instantiates a newChildToParentMessageWriter orChildToParentMessageReader object.

**Type parameters**

Type parameter T extends SignerOrProvider

**Parameters**

Parameter Type Description parentSignerOrProvider T - event ChildToParentTransactionEvent The event containing the data of the Child-to-Parent message. parentProvider ? Provider -

**Returns**

[ChildToParentMessageReaderOrWriter](#) <T

[ChildToParentMessageReader](#) [.fromEvent](#)

[message/ChildToParentMessage.ts:76](#)

**getChildToParentEvents()**

static

getChildToParentEvents ( childProvider : Provider , filter : object , position ? : BigNumber , destination ? :

string , hash ? : BigNumber , indexInBatch ? : BigNumber ) :

Promise < ChildToParentTransactionEvent & object [ ]

Get event logs for ChildToParent transactions.

**Parameters**

Parameter Type Description childProvider Provider filter object Block range filter filter.fromBlock BlockTag - filter.toBlock ? BlockTag - position ? BigNumber The batchnumber indexed field was removed in nitro and a position indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same batchnumber. For post nitro events it will be used to find events with the same position. destination ? string The parent destination of the ChildToParent message hash ? BigNumber The uniqueId indexed field was removed in nitro and a hash indexed field was added. For pre-nitro events the value passed in here will be used to find events with the same uniqueId. For post nitro events it will be used to find events with the same hash. indexInBatch ? BigNumber The index in the batch, only valid for pre-nitro events. This parameter is ignored post-nitro

**Returns**

Promise

Any classic and nitro events that match the provided filters.

[ChildToParentMessageReader](#) [.getChildToParentEvents](#)

[message/ChildToParentMessage.ts:109](#)

# Type Aliases

## ChildToParentMessageReaderOrWriter

type

ChildToParentMessageReaderOrWriter < T

:

T

extends

Provider

? ChildToParentMessageReader : ChildToParentMessageWriter ; Conditional type for Signer or Provider. If T is of type Provider then ChildToParentMessageReaderOrWriter will be of type ChildToParentMessageReader. If T is of type Signer then ChildToParentMessageReaderOrWriter will be of type ChildToParentMessageWriter.

**Type parameters**

Type parameter T extends SignerOrProvider

**Source**