Calls

This section contains a full API reference of all public functions & events related to making and tracking xchain calls.

Events

XCalled

```
Copy event XCalled(bytes32 transferId, uint256 nonce, bytes32 messageHash, struct TransferInfo params, address asset, uint256 amount, address local)
```

Emitted whenxcall is called on the origin domain of a transfer.

Parameters

Name Type Description transferId bytes32 - The unique identifier of the crosschain transfer. nonce uint256 - The bridge nonce of the transfer on the origin domain. messageHash bytes32 - The hash of the message bytes (containing all transfer info) that were bridged. params struct TransferInfo - TheTransferInfo provided to the function. asset address - The asset sent in with xcall amount uint256 - The amount sent in with xcall local address - The local asset that is controlled by the bridge and can be burned/minted

ExternalCalldataExecuted

```
Copy eventExternalCalldataExecuted(bytes32transferId,boolsuccess,bytesreturnData)
```

Emitted when a transfer has its external data executed

Parameters

Name Type Description transferId bytes32 - The unique identifier of the crosschain transfer. success bool - Whether calldata succeeded returnData bytes - Return bytes from the IXReceiver

Executed

```
Copy event Executed(bytes32 transferId, address to, address asset, struct ExecuteArgs args, address local, uint256 amount, address caller)
```

Emitted whenexecute is called on the destination domain of a transfer.

execute may be called when providing fast liquidity or when processing a reconciled (slow) transfer.

Parameters

Name Type Description transferId bytes32 - The unique identifier of the crosschain transfer. to address - The recipientTransferInfo.to provided, created as indexed parameter. asset address - The asset the recipient is given or the external call is executed with. Should be the adopted asset on that chain. args struct ExecuteArgs - TheExecuteArgs provided to the function. local address - The local asset that was either supplied by the router for a fast-liquidity transfer or minted by the bridge in a reconciled (slow) transfer. Could be the same as the adoptedasset param. amount uint256 - The amount of transferring asset the recipient address receives or the external call is executed with. caller address - The account that called the function.

TransferRelayerFeesIncreased

```
Copy eventTransferRelayerFeesIncreased(bytes32transferId,uint256increase,addresscaller)
```

Emitted when_bumpTransfer is called by an user on the origin domain both inxcall andbumpTransfer

Parameters

Name Type Description transferId bytes32 - The unique identifier of the crosschain transaction increase uint256 - The additional amount fees increased by caller address - The account that called the function

SlippageUpdated

```
Copy eventSlippageUpdated(bytes32transferId,uint256slippage)
```

Emitted whenforceUpdateSlippage is called by an user on the destination domain

Parameters

Name Type Description transferId bytes32 - The unique identifier of the crosschain transaction slippage uint256 - The updated slippage boundary

Getters

routedTransfers

```
Copy functionroutedTransfers(bytes32_transferId)publicviewreturns(address[])
```

Gets a list of routers that routed a transfer bytransferId .

Parameters

Name Type Description _transferId bytes32 Unique transfer ID of a given transaction

Return Values

Name Type Description [0] address[] Array containing addresses of routers

transferStatus

```
Copy functiontransferStatus(bytes32_transferId)publicviewreturns(enumDestinationTransferStatus)
```

Gets a transfer's status bytransferId . Note - this function MUST be called on the destination chain.

Parameters

Name Type Description _transferId bytes32 Unique transfer ID of a given transaction

Return Values

Name Type Description [0] enum Status of the transfer

domain

```
Copy functiondomain()publicviewreturns(uint32)
```

Gets thedomain identifier of the chain.

Parameters

Return Values

Name Type Description [0] uint32 Domain identifier of the chain

Functions

xcall

```

Copy function xcall(uint32 _destination, address _to, address _asset, address _delegate, uint256 _amount, uint256 _slippage, bytes _callData, uint256 _relayerFee) external payable returns (bytes32)

```

Initiates a cross-chain transfer of funds, calldata, and/or various named properties.

For ERC20 transfers, this contract must have approval to transfer the input (transacting) assets. The adopted assets will be swapped for their local (connext-flavored) asset counterparts (i.e. bridgeable tokens) via the configured AMM if necessary. In the event that the adopted assetsare local assets, no swap is needed. The local tokens will then be sent via the bridge router. If the local assets are representational for an asset on another chain, we will burn the tokens here. If the local assets are canonical (meaning that the adopted to local asset pairing is native to this chain), we will custody the tokens here.

Parameters

Name Type Description _destination uint32 The destination chain's Domain ID (not equivalent to “Chain ID”). See [Domains] for details TODO _to address The target address on the destination chain. xcall will send funds to whatever address is specified here regardless of whether it is a contract or EOA. If calldata is provided, xcall will additionally attempt to callxReceive on this contract. _asset address The contract address of the asset to be bridged. If thexcall is calldata-only (e.g. doesn't bridge any funds), any registered asset can be used here as long asamount: 0 . _delegate address An address on destination domain that has rights to update slippage tolerance, retry transactions, or revert back to origin in the event that a transaction fails at the destination. _amount uint256 The amount of tokens to bridge specified in wei units (i.e. to send 1 USDC, a token with 10^6 decimals, you must specify the amount as1000000). _slippage uint256 The maximum slippage a user is willing to take, in BPS, due to the StableSwap Pool(s), if applicable. For example, to achieve 0.03% slippage tolerance this will be3 . _callData bytes In the case of bridging funds only, this should be empty bytes ("0x"). If calldata is sent, then the encoded calldata must be passed here. _relayerFee uint256 (Optional) This is available in an overloadedxcall . If provided, the relayer fee will be taken in_asset rather than the native asset.

Return Values

Name Type Description [0] bytes32 bytes32 - The transfer ID of the newly created crosschain transfer.

xcallIntoLocal

```

Copy function xcallIntoLocal(uint32 _destination, address _to, address _asset, address _delegate, uint256 _amount, uint256 _slippage, bytes _callData, uint256 _relayerFee) external payable returns (bytes32)

```

Helper function that xcalls as normal but forces the receipt of the local (Connext-flavored) asset at destination. This function is used typically to generate nextAssets that can be used to LP into the destination chain stableswap. Params and returned data function exactly the same way asxcall .

execute

```

Copy functionexecute(structExecuteArgs_args)externalreturns(bytes32)

```

Called on a destination domain to disburse correct assets to end recipient and execute any included calldata.

Can be called before or after handle [reconcile] is called (regarding the same transfer), depending on whether the fast liquidity route (i.e. funds provided by routers) is being used for this transfer. As a result, executed calldata (including properties like originSender) may or may not be verified depending on whether the reconcile has been completed (i.e. the optimistic confirmation period has elapsed).

Parameters

Name Type Description _args struct ExecuteArgs - ExecuteArgs arguments.

Return Values

Name Type Description [0] bytes32 bytes32 - The transfer ID of the crosschain transfer. Should match the xcall's transfer ID in order for reconciliation to occur.

bumpTransfer (native asset)

```
Copy functionbumpTransfer(bytes32_transferId)externalpayable
```

Anyone can call this function on the origin domain to increase the relayer fee for a transfer. MUST be called on the origin domain.

Parameters

Name Type Description _transferId bytes32 - The unique identifier of the crosschain transaction

bumpTransfer (transacting asset)

```
Copy functionbumpTransfer(bytes32_transferId,address_relayerFeeAsset,uint256_relayerFee)externalpayable
```

Anyone can call this function to increase the relayer fee for a transfer (using the _relayerFeeAsset specified). MUST be called on the origin domain.

Parameters

Name Type Description _transferId bytes32 - The unique identifier of the crosschain transaction _relayerfeeAsset address - The asset you are bumping fee with _relayerFee uint256 - The amount you want to bump transfer fee with

forceUpdateSlippage

```
Copy functionforceUpdateSlippage(structTransferInfo_params,uint256_slippage)external
```

Allows a user-specified account (delegate inxcall) to update the slippage they are willing to take on destination transfers. MUST be called on the destination chain.

Parameters

Name Type Description _params struct TransferInfo TransferInfo associated with the transfer _slippage uint256 The updated slippage

Interfaces

xReceive

```
Copy function xReceive(bytes32 _transferId, uint256 _amount, address _asset, address _originSender, uint32 _origin, bytes _callData) external returns (bytes)
```

Interface that the Connext contracts call into on the_to address specified duringxcall . Developers MUST implement this on the destination chain to receive incoming calldata.

Parameters

Name Type Description _transferId bytes32 Unique id of the xchain transaction _amount uint256 Amount of token, if any, passed into the contract in Wei units _asset address Address of token, if any, passed into the contract _originSender address Address of the contract or EOA that calledxcall on the origin chain. NOTE: this param willonly be populated if the transaction went through the slow path rather than being executed immediately by a Connext router (see TODO for details) _origin uint32 Domain ID of the chain that the transaction is coming from _calldata bytes Data, in bytes, that is passed intoxcall on the origin chain

Previous Contracts Next Routers Last updated8 months ago

Edit on GitHub