Error Codes & Handling

Invalid Argument

info This section shows the error that typically occurs when a function is called with parameters that do not match the expected type, range, or format. * InvalidArgument() * A general error code that implies the parameter passed is invalid. * InvalidAmount() * An invalid amount has been passed as input. For example, when setting Treasury Native Fee Cap, if the new value is larger than the old valid, this error would occur. * InvalidNonce() * The error occurs if the nonce value is not the expected nonce. Returned either by the Endpoint if the inbound nonce is not verifiable, or if the provided nonce value is not the next expected nonce (i.e., current nonce + 1). This ensures that nonces are processed in order and no nonce is missed or processed out of order. * InvalidSizeForAddress() * The error occurs when the input parameter is of the incorrect size. * InvalidAddress() * The error occurs when the input parameter is of the incorrect length. * InvalidMessageSize() * The error occurs when the actual message size exceeds the message size cap. * InvalidPath() * The error occurs when the path length doesn't match20 * + remoteAddressSize. * InvalidSender() * The error occurs when the sender doesn't match the source address in the path. * InvalidConfirmations() * The error occurs when a call is made before the required OApp block confirmations. * InvalidExpiry(uint256 expiry, uint256 minExpiry) * When setting expiry time for the default library, thrown if the expiry time is set before the block timestamp. * InvalidReceiveLibrary() * The error occurs when the receive library is not a valid library when verifying a message. * InvalidPacketVersion() * The error occurs when the version number of the packet header does not match the expected packet version defined in the ULN. * InvalidRequiredDVNCount() * The error occurs if the verifier list is not empty while the DVNCount is configured to NONE or DEFAULT. * InvalidPacketHeader() * The error occurs if the length of packetHeader is not 81. * InvalidDVNIdx() * The error occurs when the_DVNIdx * is 255 or greater. The max number of DVN is 255. * InvalidLegacyType1Option() * The error occurs if there's invalidtype1 * option settings (invalid adapterParams from v1). * InvalidLegacyType2Option() * The error occurs if there's invalidtype2 * option settings (invalid adapterParams from v1). * InvalidDVNOptions() * The error occurs if an invalid or unsupported DVN was set in the DVN config params. * InvalidRequiredDVNCount() * The error occurs if the actual amount of required DVN violates the config. * InvalidOptionalDVNCount() * The error occurs if the actual amount of optional DVN violates the config. * InvalidOptionalDVNThreshold() * The error occurs if the actual threshold of optional DVN violates the config. * InvalidExecutorOptions() * The error occurs if cursor is not equal to the length of options. * InvalidExecutor() * The error occurs if the executor address is zero in config params. * InvalidConfigType() * The error occurs if the config type is invalid. * InvalidPayloadHash() * The error occurs if the payloadHash passed in the _inbound argument is empty. * OnlySendLib() * The error occurs when the message library is ReceiveLibrary while it is supposed to beSendLibrary * . * OnlyReceiveLib() * The error occurs when the message library is SendLibrary while it is supposed to beReceiveLibrary * . * OnlyRegisteredLib() * Only registered libraries can be passed as a parameter. Unregistered libraries can't be included. * OnlyRegisteredOrDefaultLib() * Only non-default libraries can be passed as a parameter. Unregistered or non-default libraries can't be included. * OnlyNonDefaultLib() * The error occurs when the_newLib * is either the same as thedefaultLib * or theoldLib * . Pass a new library address (_newLib *) that's not the default or current library. * PathNotInitializable()

The error occurs if the path can not be initialized.

- PathNotVerifiable()

The error occurs if the path is not verifiable.

- ZeroMessageSize()
- The error occurs if max message size is zero in config params.
- ZeroLzTokenFee()
- IfpayInLzToken
- is true, the supplied fee must be greater than 0 to prevent a race condition in which an oapp sending a message with lz token and the lz token is set to a new token between the tx being sent and the tx being mined. If the required lz token fee is 0 and the old lz token would be locked in the contract instead of being refunded.
- AtLeastOneDVN()
- The error occurs if zero (0) is set for both requiredDVNCount and optionalDVNThreshold.
- InsufficientFee()
- The error occurs ifrequired.nativeFee
- is larger thansuppliedNativeFee
- orrequired.lzTokenFee
- is larger thansuppliedLzTokenFee
- , or when the msg.value is less than the returned fee amount.
- InsufficientMsgValue()
- The error occurs if msg.value is less than the total NativeFee.
- UnknownL2Eid()
- This error occurs if the L2 Eid is unknown when looking up the L1 EID for the particular L2 networks.
- Unsorted()
- The error occurs when there are duplicate addresses in the_dvns
- array.
- UnsupportedEid()
- The error occurs when the endpoint id of the packet header does not match the expected packet endpoint defined in the ULN.

- CannotWithdrawAltToken()
- The error occurs if native token is the same as lzToken.
- LzTokenPaymentAddressMustBeSender()
- The error occurs if lzToken payment address is not the sender.
- SameValue()
- The error occurs if the provided_newLib
- address is the same as the currently setdefaultSendLibrary
- for the given_eid
- , or if a user attempts to set thedefaultReceiveLibrary
- for a specific_eid
- to the same address it's currently set to.
- NoOptions()
- The error occurs if the length of options is zero.
- InvalidWorkerOptions()
- The error occurs if the worker options are invalid (less than 2 bytes).
- InvalidWorkerId()
- The error occurs if the worker ID is 0.
- NativeAmountExceedsCap()
- The error occurs if the native amount to be received on destination exceeds native airdrop cap.
- Verifying()
- The error occurs if the state of a packet with the passed arguments (_config
- ,_headerHash
- and_payloadHash
-) is not verfiable yet.

Invalid State

info This section shows the error that typically occurs if it does not meet certain expected conditions when a function is called or a transaction is executed. * TransferNativeFailed * The error occurs when sending less than the_required * amount of native token to the receiver. * SendReentrancy() * The error occurs when the_sendContext * has already been entered. TheMessagingContext * requires that_sendContext has not been entered, and acts as a non-reentrancy guard.

Permission Denied

info This section shows the errors that typically occur when a function or operation is attempted by an address that doesn't have the necessary permissions. * OnlyAltToken() * OnlyaltFeeToken * can be used for fees. * OnlyEndpoint() * SimpleMessageLib.sol * : requires endpoint == msg.sender * OnlyExecutor() * The error occurs when the msg.sender is not the executor when executing the message. * OnlyPriceUpdater() * The error occurs if an unauthorized address (not the contract owner and not in the priceUpdater list) tries to call the function. * OnlyWhitelistCaller() * SimpleMessageLib.sol * : requiresmsg.sender == whitelistCaller * to callvalidatePacket * ToIsAddressZero() * The error occurs if the _to address is zero when calling withdrawFee. * LzTokenIsAddressZero() * The error occurs if the lzToken address is zero to call withdrawLzTokenFee. * Unauthorized() * When the msg.send is not the OApp or the delegates of the OApp. * NotTreasury() * The error occurs if msg.sender is not Treasury when calling treasury only function.

Not Found

info This section shows the errors that typically occur when a requested resource does not exist. * PayloadHashNotFound

In MessagingChannel.sol, the error occurs when the actual payload hash doesn't match the expected payload hash.

- ComposedMessageNotFound

In MessagingComposer.sol, the error occurs when the actual hash doesn't match the expected hash of a composed message.

Already Exists

info This section shows the errors that typically error when adding something that conflicts with an existing resource in the contract. * AddressSizeAlreadySet()

The error occurs when an endpoint's address size has already been set.

- AlreadyRegistered()

The error occurs when the_lib has already been registered.

- ComposeExists()

The error occurs when message hash doesn't pass the identity check in the composeQueue. The message must have not

been sent before.

Not Implemented

info This section shows the error that typically occur when a certain function, method or feature that is not yet defined in the contract. * NotImplemented()

A general error code that implies undefined function.

- UnsupportedInterface()

The error occurs if the library does not implement ERC165 interface.

- UnsupportedOptionType()

The error occurs when the option type is not supported. For example, Endpoint V1 does not support type 3 options.

Unavailable

info This section shows the error that typically occur when a requested resouce is not currently available. * LzTokenUnavailable()

The error occurs if LzToken is not available for payments but users passed LzTokens in for payments. Simply set payInLzToken to false in this case.

- LzTokenNotEnabled()

The error occurs if the lzToken is not enabled when calling _getFee .

- DefaultSendLibUnavailable()

The error occurs if the send message library doesn't support the specific endpoint ID.

- DefaultReceiveLibUnavailable()

The error occurs if the receive message library doesn't support the specific endpoint ID.Edit this page

Previous Debugging Messages Next Integration Checklist