Class: ConditionalOrder

An abstract base class from which all conditional orders should inherit.

This class provides some basic functionality to help with handling conditional orders, such as:

- Validating the conditional order
- Creating a human-readable string representation of the conditional order
- Serializing the conditional order for use with theIConditionalOrder
- struct
- Getting any dependencies for the conditional order
- Getting the off-chain input for the conditional order

NOTE : Instances of conditional orders have anid property that is akeccak256 hash of the serialized conditional order.

Type parameters

Name D S

Hierarchy

- ConditionalOrder
- ↳Twap

Constructors

constructor

•new ConditionalOrder <D ,S

(params):ConditionalOrder <D ,S

A constructor that provides some basic validation for the conditional order.

This constructorMUST be called by any class that inherits fromConditionalOrder .

NOTE : The salt is optional and will be randomly generated if not provided.

Type parameters

Name D S

Parameters

Name Type params ConditionalOrderArguments <D

Returns

ConditionalOrder <D ,S

Throws

If the handler is not a valid ethereum address.

Throws

If the salt is not a valid 32-byte string.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:54

Properties

data

•Readonly data :D

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:37

handler

•Readonly handler :string

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:35

hasOffChainInput

•Readonly hasOffChainInput :boolean

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:39

salt

•Readonly salt :string

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:36

staticInput

•Readonly staticInput :S

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:38

Accessors

context

•get context ():undefined |ContextFactory

Get the context dependency for the conditional order.

This is used when callingcreateWithContext orsetRootWithContext on a ComposableCoW-enabled Safe.

Returns

undefined |ContextFactory

The context dependency.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:91

createCalldata

•get createCalldata ():string

Get the calldata for creating the conditional order.

This will automatically determine whether or not to usecreate orcreateWithContext based on the order type's context dependency.

NOTE : By default, this will cause the create to emit theConditionalOrderCreated event.

Returns

string

The calldata for creating the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:113

ctx

•get ctx ():string

The context key of the order (bytes32(0) if a merkle tree is used, otherwise H(params)) with which to lookup the cabinet

The context, relates to the 'ctx' in the contracthttps://github.com/cowprotocol/composable-cow/blob/c7fb85ab10c05e28a1632ba97a1749fb261fcdfb/src/interfaces/IConditionalOrder.sol#L38

Returns

string

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:166

id

•get id ():string

Calculate the id of the conditional order (which also happens to be the key used forctx in the ComposableCoW contract).

This is akeccak256 hash of the serialized conditional order.

Returns

string

The id of the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:157

isSingleOrder

•get isSingleOrder ():boolean

Returns

boolean

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:76

leaf

•get leaf ():ConditionalOrderParams

Get theleaf of the conditional order. This is the data that is used to create the merkle tree.

For the purposes of this library, theleaf is theConditionalOrderParams struct.

Returns

ConditionalOrderParams

Theleaf of the conditional order.

See

ConditionalOrderParams

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:177

offChainInput

•get offChainInput ():string

If the conditional order has off-chain input, return it!

NOTE : This should be overridden by any conditional order that has off-chain input.

Returns

string

The off-chain input.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:201

orderType

•get orderType ():string

Get a descriptive name for the type of the conditional order (i.e twap, dca, etc).

Returns

string

The concrete type of the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:83

removeCalldata

•get removeCalldata ():string

Get the calldata for removing a conditional order that was created as a single order.

Returns

string

The calldata for removing the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:145

Methods

assertIsValid

▸assertIsValid ():void

Returns

void

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:95

cabinet

▸cabinet (params):Promise <string

Checks the value in the cabinet for a given owner and chain

Parameters

Name Type Description params OwnerContext owner context, to be able to check the cabinet

Returns

Promise <string

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:337

encodeStaticInput

▸encodeStaticInput ():string

Encode thestaticInput for the conditional order.

Returns

string

The ABI-encodedstaticInput for the conditional order.

See

ConditionalOrderParams

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:225

encodeStaticInputHelper

▸encodeStaticInputHelper (orderDataTypes ,staticInput):string

A helper function for generically serializing a conditional order's static input.

Parameters

Name Type Description orderDataTypes string [] ABI types for the order's data struct. staticInput S -

Returns

string

An ABI-encoded representation of the order's data struct.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:234

handlePollFailedAlreadyPresent

▸handlePollFailedAlreadyPresent (orderUid ,order ,params):Promise <undefined |PollResultErrors

This method lets the concrete conditional order decide what to do if the order yielded in the polling is already present in the Orderbook API.

The concrete conditional order will have a chance to schedule the next poll. For example, a TWAP order that has the current part already in the orderbook, can signal that the next poll should be done at the start time of the next part.

Parameters

Name Type orderUid string order DataStruct params PollParams

Returns

Promise <undefined |PollResultErrors

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:364

isAuthorized

▸isAuthorized (params):Promise <boolean

Checks if the owner authorized the conditional order.

Parameters

Name Type Description params OwnerContext owner context, to be able to check if the order is authorized

Returns

Promise <boolean

true if the owner authorized the order, false otherwise.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:326

isValid

▸isValid ():IsValidResult

Returns

IsValidResult

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:102

poll

▸poll (params):Promise <PollResult

Poll a conditional order to see if it is tradeable.

Parameters

Name Type params PollParams

Returns

Promise <PollResult

The tradeableGPv2Order.Data struct and thesignature for the conditional order.

Throws

If the conditional order is not tradeable.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:249

pollValidate

▸pollValidate (params):Promise <undefined |PollResultErrors

Allow concrete conditional orders to perform additional validation for the poll method.

This will allow the concrete orders to decide when an order shouldn't be polled again. For example, if the orders is expired. It also allows to signal when should the next check be done. For example, an order could signal that the validations will fail until a certain time or block.

Parameters

Name Type Description params PollParams The poll parameters

Returns

Promise <undefined |PollResultErrors

undefined if the concrete order can't make a decision. Otherwise, it returns a PollResultErrors object.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:354

serialize

▸serialize ():string

Serializes the conditional order into it's ABI-encoded form.

Returns

string

The equivalent ofIConditionalOrder.Params for the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:217

toString

▸toString (tokenFormatter?):string

Create a human-readable string representation of the conditional order.

Parameters

Name Type Description tokenFormatter? (address :string ,amount :BigNumber) =>string An optional function that takes an address and an amount and returns a human-readable string.

Returns

string

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:210

transformDataToStruct

▸transformDataToStruct (params):S

Converts a friendly data object modelling the smart order into the struct that the contract expect as an encodedstaticInput .

NOTE : This should be overridden by any conditional order that requires transformations. This implementation is a no-op if you use the same type for both.

Parameters

Name Type Description params D {S} Parameters that are passed in to the constructor.

Returns

S

The static input for the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:390

transformStructToData

▸transformStructToData (params):D

Convert the struct that the contract expect as an encodedstaticInput into a friendly data object modelling the smart order.

NOTE : This should be overridden by any conditional order that requires transformations. This implementation is a no-op if you use the same type for both.

Parameters

Name Type Description params S {S} Parameters that are passed in to the constructor.

Returns

D

The static input for the conditional order.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:379

deserializeHelper

▸deserializeHelper <T

(s ,handler ,orderDataTypes ,callback):T

A helper function for generically deserializing a conditional order.

Type parameters

Name T

Parameters

Name Type Description s string The ABI-encodedIConditionalOrder.Params struct to deserialize. handler string Address of the handler for the conditional order. orderDataTypes string [] ABI types for the order's data struct. callback (d :any ,salt :string) =>T A callback function that takes the deserialized data struct and the salt and returns an instance of the class.

Returns

T

An instance of the conditional order class.

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:400

leafToId

▸leafToId (leaf):string

Calculate the id of the conditional order.

Parameters

Name Type Description leaf ConditionalOrderParams Theleaf representing the conditional order.

Returns

string

The id of the conditional order.

See

ConditionalOrderParams

Defined in

external/cow-sdk/src/composable/ConditionalOrder.ts:191 Previous errorType Next ConditionalOrderFactory