

IOrderMixin

Quick links

[Types](#)[Functions](#)[Events](#)[Errors](#)

Types

Order

```
struct Order {  
    uint256 salt;  
    Address maker;  
    Address receiver;  
    Address makerAsset;  
    Address takerAsset;  
    uint256 makingAmount;  
    uint256 takingAmount;  
    MakerTraits makerTraits;  
}
```

Functions

bitInvalidatorForOrder

```
function bitInvalidatorForOrder(address maker, uint256 slot) external view returns (uint256 result)
```

Returns bitmask for double-spend invalidators based on lowest byte of order.info and filled quotes

Parameters

Name	Type	Description
maker	address	Maker address
slot	uint256	Slot number to return bitmask for

Return Values

Name	Type	Description
result	uint256	Each bit represents whether corresponding was already invalidated

remainingInvalidatorForOrder

```
function remainingInvalidatorForOrder(address maker, bytes32 orderHash) external view returns (uint256 remaining)
```

Returns bitmask for double-spend invalidators based on lowest byte of order.info and filled quotes

Parameters

Name	Type	Description
maker	address	
orderHash	bytes32	Hash of the order

Return Values

Name	Type	Description
remaining	uint256	Remaining amount of the order

rawRemainingInvalidatorForOrder

```
function rawRemainingInvalidatorForOrder(address maker, bytes32 orderHash) external view returns (uint256 remainingRaw)
```

Returns bitmask for double-spend invalidators based on lowest byte of order.info and filled quotes

Parameters

Name	Type	Description
maker	address	
orderHash	bytes32	Hash of the order

Return Values

Name	Type	Description
remainingRaw	uint256	Inverse of the remaining amount of the order if order was filled at least once, otherwise 0

cancelOrder

```
function cancelOrder(MakerTraits makerTraits, bytes32 orderHash) external
```

Cancels order's quote

Parameters

Name	Type	Description
makerTraits	MakerTraits	Order makerTraits
orderHash	bytes32	Hash of the order to cancel

cancelOrders

```
function cancelOrders(MakerTraits[] makerTraits, bytes32[] orderHashes) external
```

Cancels orders' quotes

Parameters

Name	Type	Description
makerTraits	MakerTraits[]	Orders makerTraits
orderHashes	bytes32[]	Hashes of the orders to cancel

bitsInvalidateForOrder

```
function bitsInvalidateForOrder(MakerTraits makerTraits, uint256 additionalMask) external
```

Cancels all quotes of the maker (works for bit-invalidating orders only)

Parameters

Name	Type	Description
makerTraits	MakerTraits	Order makerTraits
additionalMask	uint256	Additional bitmask to invalidate orders

hashOrder

```
function hashOrder(struct IOrderMixin.Order order) external view returns (bytes32 orderHash)
```

Returns order hash, hashed with limit order protocol contract EIP712

Parameters

Name	Type	Description
order	struct IOrderMixin.Order	Order

Return Values

Name	Type	Description
orderHash	bytes32	Hash of the order

simulate

```
function simulate(address target, bytes data) external
```

Delegates execution to custom implementation. Could be used to validate if `transferFrom` works properly

The function always reverts and returns the simulation results in revert data.

Parameters

Name	Type	Description
target	address	Addresses that will be delegated
data	bytes	Data that will be passed to delegatee

fillOrder

```
function fillOrder(struct IOrderMixin.Order order, bytes32 r, bytes32 vs, uint256 amount, TakerTraits takerTraits) external payable returns (uint256 makingAmount, uint256 takingAmount, bytes32 orderHash)
```

Fills order's quote, fully or partially (whichever is possible).

Parameters

Name	Type	Description
order	struct IOrderMixin.Order	Order quote to fill
r	bytes32	R component of signature
vs	bytes32	VS component of signature
amount	uint256	Taker amount to fill
takerTraits	TakerTraits	Specifies threshold as maximum allowed takingAmount when takingAmount is zero, otherwise specifies minimum allowed makingAmount. The 2nd (0 based index) highest bit specifies whether taker wants to skip maker's permit.

Return Values

Name	Type	Description
makingAmount	uint256	Actual amount transferred from maker to taker
takingAmount	uint256	Actual amount transferred from taker to maker
orderHash	bytes32	Hash of the filled order

fillOrderArgs

`function fillOrderArgs(struct IOrderMixin.Order order, bytes32 r, bytes32 vs, uint256 amount, TakerTraits taker Traits, bytes args) external payable returns (uint256 makingAmount, uint256 takingAmount, bytes32 orderHash)`

Same as `fillOrder` but allows to specify arguments that are used by the taker.

Parameters

Name	Type	Description
order	struct IOrderMixin.Order	Order quote to fill
r	bytes32	R component of signature
vs	bytes32	VS component of signature
amount	uint256	Taker amount to fill
takerTraits	TakerTraits	Specifies threshold as maximum allowed takingAmount when takingAmount is zero, otherwise specifies minimum allowed makingAmount. The 2nd (0 based index) highest bit specifies whether taker wants to skip maker's permit.
args	bytes	Arguments that are used by the taker (target, extension, interaction, permit)

Return Values

Name	Type	Description
makingAmount	uint256	Actual amount transferred from maker to taker
takingAmount	uint256	Actual amount transferred from taker to maker
orderHash	bytes32	Hash of the filled order

fillContractOrder

```
function fillContractOrder(struct IOrderMixin.Order order, bytes signature, uint256 amount, TakerTraits takerTraits) external returns (uint256 makingAmount, uint256 takingAmount, bytes32 orderHash)
```

Same as `fillOrder` but uses contract-based signatures.

See tests for examples

Parameters

Name	Type	Description
order	struct IOrderMixin.Order	Order quote to fill
signature	bytes	Signature to confirm quote ownership
amount	uint256	Taker amount to fill
takerTraits	TakerTraits	Specifies threshold as maximum allowed takingAmount when takingAmount is zero, otherwise specifies minimum allowed makingAmount. The 2nd (0 based index) highest bit specifies whether taker wants to skip maker's permit.

Return Values

Name	Type	Description
makingAmount	uint256	Actual amount transferred from maker to taker
takingAmount	uint256	Actual amount transferred from taker to maker
orderHash	bytes32	Hash of the filled order

fillContractOrderArgs

```
function fillContractOrderArgs(struct IOrderMixin.Order order, bytes signature, uint256 amount, TakerTraits takerTraits, bytes args) external returns (uint256 makingAmount, uint256 takingAmount, bytes32 orderHash)
```

Same as `fillContractOrder` but allows to specify arguments that are used by the taker.

See tests for examples

Parameters

Name	Type	Description
order	struct IOrderMixin.Order	Order quote to fill
signature	bytes	Signature to confirm quote ownership
amount	uint256	Taker amount to fill
takerTraits	TakerTraits	Specifies threshold as maximum allowed takingAmount when takingAmount is zero, otherwise specifies minimum allowed makingAmount. The 2nd (0 based index) highest bit specifies whether taker wants to skip maker's permit.
args	bytes	Arguments that are used by the taker (target, extension, interaction, permit)

Return Values

Name	Type	Description
makingAmount	uint256	Actual amount transferred from maker to taker
takingAmount	uint256	Actual amount transferred from taker to maker
orderHash	bytes32	Hash of the filled order

Events

OrderFilled

`event OrderFilled(bytes32 orderHash, uint256 remainingAmount)`

Emitted when order gets filled

Parameters

Name	Type	Description
orderHash	bytes32	Hash of the order
remainingAmount	uint256	Amount of the maker asset that remains to be filled

OrderCancelled

`event OrderCancelled(bytes32 orderHash)`

Emitted when order without `useBitInvalidator` gets cancelled

Parameters

Name	Type	Description
orderHash	bytes32	Hash of the order

BitInvalidatorUpdated

```
event BitInvalidatorUpdated(address maker, uint256 slotIndex, uint256 slotValue)
```

Emitted when order with `useBitInvalidator` gets cancelled

Parameters

Name	Type	Description
maker	address	Maker address
slotIndex	uint256	Slot index that was updated
slotValue	uint256	New slot value

Errors

InvalidatedOrder

```
error InvalidatedOrder()
```

TakingAmountExceeded

```
error TakingAmountExceeded()
```

PrivateOrder

```
error PrivateOrder()
```

BadSignature

```
error BadSignature()
```


OrderExpired

error `OrderExpired()`

WrongSeriesNonce

error `WrongSeriesNonce()`

SwapWithZeroAmount

error `SwapWithZeroAmount()`

PartialFillNotAllowed

error `PartialFillNotAllowed()`

OrderIsNotSuitableForMassInvalidation

error `OrderIsNotSuitableForMassInvalidation()`

EpochManagerAndBitInvalidatorsAreIncompatible

error `EpochManagerAndBitInvalidatorsAreIncompatible()`

ReentrancyDetected

error `ReentrancyDetected()`

PredicatesNotTrue

error `PredicatesNotTrue()`

TakingAmountTooHigh

error `TakingAmountTooHigh()`

MakingAmountTooLow

error `MakingAmountTooLow()`

TransferFromMakerToTakerFailed

error `TransferFromMakerToTakerFailed()`

TransferFromTakerToMakerFailed

error `TransferFromTakerToMakerFailed()`

MismatchArraysLengths

error `MismatchArraysLengths()`

InvalidPermit2Transfer

error `InvalidPermit2Transfer()`

SimulationResults

error `SimulationResults(bool success, bytes res)`

Previous

[< ICreate3Deployer](#)

Next

[IPermit2WitnessTransferFrom >](#)