Router Calls

The baseuserCmd() treates the calling address (msg.sender) as the owner of the call for crediting/debiting the output/input of transactions and assigning ownership of LP positions and surplus collateral. However Ambient supports allowing for a third-party smart contract (router) to execute arbitrary commands on the user's behalf.

Invocation

In most cases whatever the user wants to do can be accomplished by directly calling theCrocSwapDex contract. But Ambient also allows users to authorize an intermediate smart contract to call Ambient on their behalf. Router calls on behalf of users will execute as if a user had directly calleduserCmd on the Ambient contract. The method to do this is:

```

Copy function userCmdRouter ( uint16 callpathIdx, bytes calldata cmd, address client, uint256 salt) returns (bytes memory)
```

callpathIdx andcmd are the same parameters that would be passed to the standarduserCmd and will behave identically.

client is the address of the end user the router is calling on behalf.

salt is an arbitrary value mapping to router approval. It allows for a multi-dimensional approval, but for cases not requiring this functionality we recommend this value be set to 0.

Approval

Before a router call is invoked, the usermust previously approve the specific smart contract address making the call. Otherwise the contract call will revert. The user must either directly call the approve command.

```

Copy userCmd(0, abi.encode( 72, // Fixed initPool subcode router, // address nCalls, // uint32 salt, // uint256 )
```

nCalls is the number of calls the router is approved to make on a user's behalf. Setting this value to the 32-bit maximum will approve the router for an unlimited number of calls (which reduces the gas overhead from decrementing a router's call count).

salt is an arbitrary value mapping to router approval. It allows for a multi-dimensional approval, but for cases not requiring this functionality we recommend this value be set to 0.