Protocol API

This page covers the Built-in actors Protocol API.

The protocol level built-in actors API is split into the following sections:

Account actor

The account actor is responsible for user account. If you want to call these methods in your smart contracts, you need to specify method number of that method you want to invoke. Please refer the each method for its method number.

AuthenticateMessage

```

Copy funcAuthenticateMessage(params AuthenticateMessage) EmptyValue ()

```

Authenticates whether the provided signature is valid for the provided message.

uint AuthenticateMessageMethodNum = 2643134072.

Parameters:

- struct
- AuthenticateMessageParams
-
    - bytes
-
    - AuthenticateMessageParamsSignature - it should be a raw byte of signature, NOT a serialized signature object with a signatureType.
-
    - bytes
-
    - Message - The message which is signed by the corresponding account address.
- *
-

Results:

- struct
- EmptyValue.
-

UniversalReceiverHook

```

Copy funcUniversalReceiverHook(params RawBytes) EmptyValue ()

```

Whenever the account receives transfers, this method will be invoked.

uint UniversalReceiverHookMethodNum = 3726118371.

Parameters:

- bytes[]
- RawBytes - passes the bytes through how it is received.

-

Results:

- struct
- EmptyValue - always success.
-

Datacap

DataCap Actor is responsible for DataCap token management. The ActorCode for DataCap actor ishex"0007" which will be used to call DataCap actor APIs. You also need to specify the method number of which method you want to invoke. Refer to each method for its method number.

Name

```

Copy funcName() String {}

```

Return the name of DataCap token which is ‘DataCap’.

Unit NameMethodNum : 48890204.

Parameters:

- null
-

Results:

- String
- : DataCap
-

Symbol

```

Copy funcSymbol() String {}

```

Return the symbol of DataCap token which is ‘DCAP’.

unit SymbolMethodNum: 2061153854.

Parameters:

- null
-

Results:

- String
- : DCAP
-

TotalSupply

```

Copy funcTotalSupply() TokenAmount {}

```

Return the total supply of the DataCap token.

uint TotalSupplyMethodNum: 114981429.

Parameters:

- null
-

Results:

- int256
- TokenAmount - Total DataCap token supply.
-

Balance

```

Copy funcBalance(params Address) TokenAmount {}

```

Return the DataCap token balance for the wallet address.

unit BalanceOfMethodNum: 3261979605.

Parameters:

- bytes
- Address - the wallet address.
-

Results:

- int256
- TokenAmount - the DataCap token balance for the specified wallet address.
-

Transfer

```

Copy funcTransfer(params TransferParams) TransferReturn {}

```

Transfers DataCap tokens from caller address to the to address.

uint TransferMethodNum = 80475954;

Parameters:

- struct
- TransferParams
-
    - bytes
-
    - To - the address to receive DataCap token.
-
    - int256
-
    - Amount - A non-negative amount to transfer.
-
    - bytes[]
-
    - OperatorData - Arbitrary data to pass on via the receiver hook.
- *
-

Results:

- struct
- TransferReturn
-
    - int256
-
    - FromBalance - the balance of from_address.

-
    - int256
-
    - ToBalance - the balance of to_address.
-
    - bytes
-
    - RecipientData: data returned from receive hook.
- *
-

TransferFrom

```

Copy funcTransferFrom(params TransferFromParams) TransferFromReturn {}

```

Transfers DataCap between the from_address to the to_address.

uint TransferFromMethodNum = 3621052141.

Params:

- bytes
- TransferFromParams
-
    - bytes
-
    - From - the address to send DataCap Token.
-
    - bytes
-
    - To - the address to receive DataCap Token.
-
    - int256
-
    - Amount - A non-negative amount to transfer.
-
    - bytes
-
    - OperatorData: Arbitrary data to pass on via the receiver hook.
- *
-

Results:

- struct
- TransferFromReturn
-
    - int256
-
    - FromBalance - the balance of from_address.
-
    - int256
-
    - ToBalance - the balance of to_address.
-
    - int256
-
    - Allowance - the remaining allowance of owner address.
-
    - bytes
-
    - RecipientData - data returned from receive hook.
- *
-

IncreaseAllowance

```

Copy funcIncreaseAllowance(params IncreaseAllowanceParams) TokenAmount {}

```

Increase the DataCap token allowance that an operator can control by the requested amount.

uint IncreaseAllowanceMethodNum = 1777121560.

Params:

- struct
- IncreaseAllowanceParams
-
    - bytes
-
    - Operator - the wallet address of the operator.
-
    - int256
-
    - increaseAmount - increase DataCap token allowance for the operator address.
- *
-

Results:

- int256
- TokenAmount - the new DataCap allowance of the operator address.
-

DecreaseAllowance

```

Copy funcDecreaseAllowance(params DecreaseAllowanceParams) TokenAmount {}

```

Decrease the DataCap token allowance that an operator controls of the owner’s balance by the requested amount.

uint DecreaseAllowanceMethodNum = 1529376545;

Params:

- struct
- DecreaseAllowanceParams
-
    - bytes
-
    - Operator - the wallet address of the operator.
-
    - int256
-
    - IncreaseAmount - the decreased DataCap token allowance of the operator address.
- *
-

Results:

- int256
- TokenAmount - the new DataCap allowance of the operator address.
-

RevokeAllowance

```

Copy funcRevokeAllowance(params RevokeAllowanceParams) TokenAmount {}

```

Revoke the DataCap token allowance from the operator and set the operator’s allowance in behave of owner/caller address to 0.

uint RevokeAllowanceMethodNum = 2765635761.

Params:

- struct
- RevokeAllowanceParams
-
    - bytes
-
    - Operator - the wallet address of the operator.
- *
-

Results:

- int256
- TokenAmount - the old Allowance amount of the operator address.
-

Burn

```

Copy funcBurn(params BurnParams) TokenAmount {}

```

Burn an amount of DataCap token from the owner/caller address, decreasing total token supply.

uint BurnMethodNum = 1434719642.

Params:

- struct
- BurnParams
-
    - int256
-
    - Amount - the amount the DataCap token to be burned.
- *
-

Results:

- int256
- TokenAmount - the updated DataCap token balance of the owner/caller address.
-

BurnFrom

```

Copy funcBurnFrom(params BurnFromParams) BurnFromReturn {}

```

Burn an amount of DataCap token from the specified address (owner address), decrease the allowance of operator/caller, and decrease total token supply.

uint BurnFromMethodNum = 2979674018.

Params:

- struct
- BurnFromParams
-
    - bytes
-
    - Owner - the wallet address of the owner.

-
    - int256
-
    - Amount - the amount of DataCap token to be burned.
- *
-

Results:

- struct
- BurnFromReturn
-
    - bytes
-
    - Owner - the wallet address of the owner.
-
    - int256
-
    - Amount - the new balance of owner wallet.
- *
-

Allowance

```

Copy funcAllowance(params GetAllowanceParams) TokenAmount {}

```

Return the allowance between owner and operator address.

uint AllowanceMethodNum = 4205072950;

Params:

- struct
- GetAllowanceParams
-
    - bytes
-
    - Owner : the wallet address of the owner.
-
    - bytes
-
    - Operator : the wallet address of the owner.
- *
-

Results:

- int256
- TokenAmount - the allowance that an operator can control of an owner’s allowance.
-

Miner

The miner built-in actor responsible to deal with storage mining operations and collect proof. To interact with a specific storage provider, you must use their miner address to invoke the methods in the built-in miner actor. You also need to specify the method number for the method you want to invoke. Please refer to each method for its method number.

GetPeerID

```

Copy funcGetPeerID() GetPeerIDReturn {}

```

Return the Peer ID for the caller/miner address.

uint GetPeerIDMethodNum = 2812875329.

Params:

- null
-

Results:

- struct
- GetPeerIDReturn
-
    - bytes
-
    - PeerID - the peer ID for the specified storage provider/miner.
- *
-

ChangePeerID

```

Copy funcChangePeerID(params ChangePeerIDParams) EmptyValue {}

```

Change the peer ID for the caller/miner address.

uint ChangePeerIDMethodNum = 1236548004.

Params:

- struct
- ChangePeerIDParams
-
    - bytes
-
    - NewID - the new peer ID.
- *
-

Results:

- struct
- EmptyValue
-

GetMultiaddrs

```

Copy funcGetMultiaddrs() GetMultiAddrsReturn {}

```

Returns the multi-signature address for this caller/miner address.

uint GetMultiaddrsMethodNum = 1332909407.

Params:

- null
-

Results:

- struct
- GetMultiAddrsReturn
-
    - byte[]
-
    - MultiAddrs - the multi-signature address.

- *
-

ChangeMultiaddrs

```

Copy funcChangeMultiaddrs(params ChangeMultiaddrsParams) EmptyValue {}

```

Change the multi-signature address for this caller/miner address.

uint ChangeMultiaddrsMethodNum = 1063480576.

Params:

- struct
- ChangeMultiaddrsParams
-
    - byte[]
-
    - NewMultiaddrs - the new multi-signature address.
- *
-

Results:

- struct
- EmptyValue
-

ChangeWorkerAddress

```

Copy funcChangeWorkerAddress(params ChangeWorkerAddressParams) EmptyValue {}

```

Change the worker address for the caller/miner address, and overwrite the existing addresses with the new control addresses passed in the params.

uint ChangeOwnerAddressMethodNum = 1010589339.

Params:

- struct
- ChangeWorkerAddressParams
-
    - byte
-
    - NewWorker - the new worker address.
-
    - byte[]
-
    - NewControlAddrs - the new controller addresses.
- *
-

Results:

- struct
- EmptyValue
-

ConfirmChangeWorkerAddress

```

Copy funcConfirmChangeWorkerAddress() EmptyValue {}
```

```

Confirm the worker address has been changed for the caller/miner address.

uint ConfirmChangeWorkerAddressMethodNum = 2354970453.

Params:

- null
-

Results:

- struct
- EmptyValue
-

RepayDebt

```

Copy funcRepayDebt() EmptyValue {}

```

Repay as much fee debt as possible for the caller/miner address.

uint RepayDebtMethodNum = 3665352697.

Params:

- null
-

Results:

- struct
- EmptyValue
-

GetOwner

```

Copy funcGetOwner() GetOwnerReturn {}

```

Return the owner address of the caller/miner address.

uint GetOwnerMethodNum = 3275365574.

Params:

- null
-

Results:

- struct
- GetOwnerReturn
-
    - byte
-
    - Owner - owner address.
- *
-

ChangeOwnerAddress

```

Copy funcChangeOwnerAddress(bytes address) {}

```

Proposes or confirms a change of owner address.

uint ChangeOwnerAddressMethodNum = 1010589339.

Params:

- bytes
- Address - the new owner address.
-

Results:

- struct
- EmptyValue
-

GetBeneficiary

```

Copy funcGetBeneficiary() GetBeneficiaryReturn {}

```

Return the currently active and proposed beneficiary information.

uint GetBeneficiaryMethodNum = 4158972569.

Params:

- null
-

Results:

- struct
- GetBeneficiaryReturn
-
    - struct
-
    - ActiveBeneficiary - current active beneficiary.
-
    -
        - byte
-
    -
        - Beneficiary - the address of the beneficiary.
-
    -
        - struct
-
    -
        - BeneficiaryTerm
-
    -
        -
            - int256
-
    -
        -
            - Quota - the quota token amount.
-
    -
        -
            - int256
-
    -
        -
            - UsedQuota - the used quota token amount.
```

-
  -
    -
      - uint64
-
  -
    -
      - Expiration - the epoch that the quota will be expired.

-
  - struct
-
  - PendingBeneficiaryChange - the proposed and pending beneficiary.
-
  -
    - bytes
-
  -
    - newBeneficiary - the new beneficiary address.
-
  -
    - int256
-
  -
    - NewQuota - the new quota token amount.
-
  -
    - uint64
-
  -
    - NewExpiration - the epoch that the new quota will be expired.
-
  -
    - bool
-
  -
    - ApprovedByBeneficiary - if this proposal is approved by the beneficiary or not.
-
  -
    - bool
-
  -
    - ApprovedByNominee - if this proposal is approved by the nominee or not.

- *
-

ChangeBeneficiary

```

Copy funcChangeBeneficiary(params ChangeBeneficiaryParams) EmptyValue {}

```

Propose or confirm a change of beneficiary information.

uint ChangeBeneficiaryMethodNum = 1570634796.

Params:

- struct
- ChangeBeneficiaryParams
-
  - bytes
-
  - newBeneficiary - the new beneficiary address.

-
    - int256
-
    - NewQuota - the new quota token amount.
-
    - uint64
-
    - NewExpiration - the epoch that the new quota will be expired.
- *
-

Results:

- struct
- EmptyValue
-

IsControllingAddress

```

Copy funcIsControllingAddress(params IsControllingAddressParams) IsControllingAddressReturn {}

```

Returns whether the provided address is the Owner, the Worker, or any of the control addresses.

uint IsControllingAddressMethodNum = 348244887.

Params:

- byte
- IsControllingAddressParams - the address to be verified.
-

Results:

- bool
- IsControllingAddressReturn - if the specified address is the control address.
-

GetSectorSize

```

Copy funcGetSectorSize() GetSectorSizeReturn {}

```

Returns the miner’s sector size.

uint GetSectorSizeMethodNum = 3858292296;

Params:

- null
-

Results:

- struct
- GetSectorSizeReturn
-
    - unit64
-
    - SectorSize - the sector size of this miner.
- *
-

GetAvailableBalance

```
```

Copy funcGetAvailableBalance() GetAvailableBalanceReturn {}

```

Returns the available balance of this miner.

uint GetAvailableBalanceMethodNum = 4026106874.

Params:

- null
-

Results:

- int256
- GetAvailableBalanceReturn - the available token balance amount.
-

WithdrawBalance

```

Copy funcWithdrawBalance(params WithdrawBalanceParams) WithdrawBalanceReturn {}

```

Withdraw the token balance for this miner.

Params:

- struct
- WithdrawBalanceParams
-
    - int256
-
    - AmountRequested - withdraw token amount.
- *
-

Results:

- int256
- WithdrawBalanceReturn - the token amount withdrawn.
-

GetVestingFunds

```

Copy funcGetVestingFunds() GetVestingFundsReturn {}

```

Return the funds vesting in this miner as a list of (vesting_epoch, vesting_amount) tuples.

uint GetVestingFundsMethodNum = 1726876304.

Params:

- null
-

Results:

- struct
- GetVestingFundsReturn
-
    - struct VestingFunds[]
-
    - Funds
-
```

    - int64
-
    - Epoch - the epoch of funds vested.
-
    - int256
-
    - Amount - the number of funds vested.

- *
-

Multisig

Multisig built-in actor is responsible for dealing with operations involving the Filecoin wallet. To interact with a specific multi-signature wallet address, you need to use this wallet address to invoke the methods in the built-in multisig actor. You also need to specify the method number of which method you want to invoke. Please refer to each method for its method number.

Propose

```

Copy funcPropose(params ProposeParams) ProposeReturn {...}

```

Propose a token transfer transaction for signers to approve. The proposer automatically approves this transaction.

uint ProposeMethodNum = 1696838335.

Params:

- struct
- ProposeParams
-
    - bytes
-
    - ToAddress - the address to receive the token.
-
    - int256
-
    - Value - the token amount to be transferred.
-
    - uint64
-
    - Method: ?
-
    - bytep[]
-
    - Params: ?
- *
-

Results:

- struct
- ProposeReturn
-
    - int64
-
    - TxnID - the ID of the proposed transaction.
-
    - bool
-
    - Applied - if the transaction was applied as proposed or not?

-
    - uint31
-
    - Code - the exit code of the transaction. IfApplied
-
    - isfalse
-
    - this field can be ignored.
-
    - bytes
-
    - Ret - the return value of the transaction. IfApplied
-
    - isfalse
-
    - this field can be ignored.
- *
-

Approve

```

Copy funcApprove(params TxnIDParams) ApproveReturn {}

```

Other signers of the multi-signature address can use this method to approve the proposed messages.

uint ApproveMethodNum = 1289044053.

Params:

- struct
- TxnIDParams
-
    - int64
-
    - ID - the signed message ID.
-
    - bytes
-
    - ProposalHash - Hash of proposal to ensure an operation can only apply to a specific proposal.
- *
-

Results:

- struct
- ApproveReturn
-
    - bool
-
    - Applied - if the transaction was applied as proposed or not?
-
    - uint31
-
    - Code - the exit code of the transaction. IfApplied
-
    - isfalse
-
    - this field can be ignored.
-
    - bytes
-
    - Ret - the return value of the transaction. IfApplied
-
    - isfalse
-
    - this field can be ignored.

- *
-

Cancel

```

Copy funcCancel(param TxnIDParams) EmptyValue {}

```

Multi-signature wallet signer to cancel a pending multi-signatures transaction.

uint CancelMethodNum = 3365893656.

Params:

- struct
- TxnIDParams
-
    - int64
-
    - ID - the signed message ID.
-
    - bytes
-
    - ProposalHash - Hash of proposal to ensure an operation can only apply to a specific proposal.
- *
-

Results:

- struct
- EmptyValue.
-

AddSigner

```

Copy funcAddSigner(params AddSignerParams) EmptyValue {}

```

Add a signer to the multi-signature wallet.

uint AddSignerMethodNum = 3028530033.

Params:

- struct
- AddSignerParams
-
    - bytes
-
    - Signer - the new signer address.
-
    - bool
-
    - Increase - increase threshold or not.
- *
-

Results:

- struct
- EmptyValue.
-

RemoveSigner

```
```

Copy funcRemoveSigner(params RemoveSignerParams) EmptyValue {}

```

Remove a signer from the multi-signature wallet.

uint RemoveSignerMethodNum = 21182899.

Params:

- struct
- RemoveSignerParams
-
    - bytes
-
    - Signer - the signer address to be removed.
-
    - bool
-
    - Decrease - decrease threshold or not. Only able to decrease when the threshold is larger than 2.
- *
-

Results:

- struct
- EmptyValue.
-

SwapSigner

```

Copy funcSwapSigner(params SwapSignerParams) EmptyValue {}

```

Swap signers for the multi-signature wallet.

uint SwapSignerMethodNum = 3968117037;

Params:

- struct
- SwapSignerParams
-
    - bytes
-
    - From - the signer address to be removed from the multi-signature wallet.
-
    - bytes
-
    - To - the signer address to be added to the multi-signature wallet.
- *
-

Results:

- struct
- EmptyValue.
-

ChangeNumApprovalsThreshold

```

Copy funcChangeNumApprovalsThreshold(params ChangeNumApprovalsThresholdParams) EmptyValue {}

```

Change the threshold number required for the approvals for the multi-signature wallet.
```

uint ChangeNumApprovalsThresholdMethodNum = 3375931653.

Params:

- struct
- ChangeNumApprovalsThresholdParams
-
    - unit64
-
    - NewThreshold - the new threshold number.
- *
-

Results:

- struct
- EmptyValue.
-

LockBalance

```

Copy funcLockBalance(params LockBalanceParams) EmptyValue {}

```

Lock a number of tokens in a multi-signature wallet from thestart epoch to theunlock epoch.

uint LockBalanceMethodNum = 1999470977.

Params:

- struct
- LockBalanceParams
-
    - int64
-
    - StartEpoch - the epoch to start locking the balance.
-
    - int64
-
    - UnlockDuration - the epoch to unlock the balance.
-
    - int256
-
    - Amount - the amount of token to be locked.
- *
-

Results:

- struct
- EmptyValue.
-

Storage market actor

Storage market actor is responsible for managing storage and retrieval deals. The ActorCode for storage market actor ishex"0005" which will be used to call this actor. You also need to specify the method number of the method you want to invoke. Please refer to each method for its method number.

AddBalance

```

Copy funcAddBalance(address Address) EmptyValue {}

```

Deposit the received FIL token, which is received along with this message, into the balance held in the escrow address of

the provider or client address.

uint AddBalanceMethodNum = 822473126.

Params:

- bytes
- Address - the address of the provider or client.
-

Results:

- struct
- EmptyValue.
-

GetBalance

```

Copy funcGetBalance(address Address) GetBalanceReturn {}

```

Return the escrow balance and locked amount for an address.

uint GetBalanceMethodNum = 726108461.

Params:

- bytes
- address - the wallet address to request balance.
-

Results:

- struct
- GetBalanceReturn
-
    - int256
-
    - Balance - the escrow balance for this address.
-
    - int256
-
    - Locked - the escrow-locked amount for this address.
- *
-

WithdrawBalance

```

Copy funcWithdrawBalance(params WithdrawBalanceParams) WithdrawBalanceReturn {}

```

Withdraw the specified amount from the balance held in escrow.

uint WithdrawBalanceMethodNum = 2280458852.

Params:

- struct
- WithdrawBalanceParams
-
    - bytes
-
    - ProviderOrClientAddress - the address of the provider or client.
-
    - int256
-

    - TokenAmount - the token amount to withdraw.
- *
-

Results:

- struct
- WithdrawBalanceReturn
-
    - int256
-
    - AmountWithdraw - the token amount withdrawn.
- *
-

PublishStorageDeals

```

Copy funcPublishStorageDeals(params PublishStorageDealsParams) PublishStorageDealsReturn {}

```

Publish a new set of storage deals that are not yet included in a sector.

uint PublishStorageDealsMethodNum = 2236929350.

Params:

- struct
- PublishStorageDealsParams
-
    - struct ClientDealProposal[]
-
    - Deals - list of deal proposals signed by a client
-
        - struct DealProposal
-
        - Proposal
-
        -
            - bytes
-
        -
            - PieceCID.
-
        -
            - uint64
-
        -
            - PieceSize - the size of the piece.
-
        -
            - bool
-
        -
            - VerifiedDeal - if the deal is verified or not.
-
        -
            - bytes
-

- Client - the address of the storage client.
- bytes
- Provider - the address of the storage provider.
- string
- Label - any label that the client chooses for the deal.
- int64
- StartEpoch - the chain epoch to start the deal.
- int64
- EndEpoch - the chain epoch to end the deal.
- int256
- StoragePricePerEpoch - the token amount to pay to the provider per epoch.
- int256
- ProviderCollateral - the token amount as collateral paid by the provider.
- int256
- ClientCollateral - the token amount as collateral paid by the client.

- bytes
- ClientSignature - the signature signed by the client.

- *
-

Results:

- struct
- PublishStorageDealsReturn
-
    - uint64[]
-
    - IDs - returned storage deal IDs.
-
    - bytes
-
    - ValidDeals - represent all the valid deals.
- *
-

GetDealDataCommitment

```

Copy funcGetDealDataCommitment(params GetDealDataCommitmentParams) GetDealDataCommitmentReturn {}

```

Return the data commitment and size of a deal proposal.

uint GetDealDataCommitmentMethodNum = 1157985802.

Params:

- uint64
- GetDealDataCommitmentParams - Deal ID.
-

Results:

- struct
- GetDealDataCommitmentReturn
-
    - bytes
-
    - Data - the data commitment of this deal.
-
    - uint64
-
    - Size - the size of this deal.
- *
-

GetDealClient

```

Copy funcGetDealClient(params GetDealClientParams) GetDealClientReturn {}

```

Return the client of the deal proposal.

uint GetDealClientMethodNum = 128053329.

Params:

- uint64
- GetDealClientParams - CID of the deal proposal.
-

Results:

- bytes

- GetDealClientReturn - the wallet address of the client.
-

GetDealProvider

```

Copy funcGetDealProvider(params GetDealProviderParams) GetDealProviderReturn {}

```

Return the provider of a deal proposal.

uint GetDealProviderMethodNum = 935081690.

Params:

- uint64
- GetDealProviderParams - CID of the deal proposal.
-

Results:

- bytes
- GetDealProviderReturn - the wallet address of the provider.
-

GetDealLabel

```

Copy funcGetDealLabel(params GetDealLabelParams) GetDealLabelReturn {}

```

Return the label of a deal proposal.

uint GetDealLabelMethodNum = 46363526.

Params:

- uint64
- GetDealLabelParams - CID of the deal proposal.
-

Results:

- string
- GetDealLabelReturn - the label of this deal.
-

GetDealTerm

```

Copy funcGetDealTerm(params GetDealTermParams) GetDealTermReturn {}

```

Return the start epoch and duration(in epochs) of a deal proposal.

uint GetDealTermMethodNum = 163777312.

Params:

- uint64
- GetDealTermParams - CID of the deal proposal.
-

Results:

- struct
- GetDealTermReturn

-
    - int64
-
    - Start - the chain epoch to start the deal.
-
    - int64
-
    - End - the chain epoch to end the deal.
- *
-

GetDealTotalPrice

```

Copy funcGetDealTotalPrice(params GetDealTotalPriceParams) GetDealTotalPriceReturn {}

```

Return the total price that will be paid from the client to the provider for this deal.

uint GetDealEpochPriceMethodNum = 4287162428.

Params:

- uint64
- GetDealTotalPriceParams - CID of the deal proposal.
-

Results:

- int256
- GetDealTotalPriceReturn - the token amount that will be paid by the client to the provider.
-

GetDealClientCollateral

```

Copy funcGetDealClientCollateral(params GetDealClientCollateralParams) GetDealClientCollateralReturn {}

```

Return the client collateral requirement for a deal proposal.

uint GetDealClientCollateralMethodNum = 200567895.

Params:

- uint64
- GetDealClientCollateralParams - CID of the deal proposal.
-

Results:

- int256
- GetDealClientCollateralReturn - the token amount as collateral paid by the client.
-

GetDealProviderCollateral

```

Copy funcGetDealProviderCollateral(params GetDealProviderCollateralParams) GetDealProviderCollateralReturn {}

```

Return the provided collateral requirement for a deal proposal.

uint GetDealProviderCollateralMethodNum = 2986712137.

Params:

- uint64
- GetDealProviderCollateralParams - CID of the deal proposal.
-

Results:

- int256
- GetDealProviderCollateralReturn - the token amount as collateral paid by the provider.
-

GetDealVerified

```

Copy funcGetDealVerified(params GetDealVerifiedParams) GetDealVerifiedReturn {}

```

Return the verified flag for a deal proposal.

uint GetDealVerifiedMethodNum = 2627389465.

Params:

- uint64
- GetDealVerifiedParams - CID of the deal proposal.
-

Results:

- bool
- GetDealVerifiedReturn - if the deal is verified or not.
-

GetDealActivation

```

Copy funcGetDealActivation(params GetDealActivationParams) GetDealActivationReturn {}

```

Return the activation state for a deal.

uint GetDealActivationParams = 2567238399.

Params:

- uint64
- GetDealVerifiedParams - CID of the deal proposal.
-

Results:

- struct
- GetDealActivationReturn
-
    - int64
-
    - Activated - Epoch at which the deal was activated, or -1.
-
    - int64
-
    - Terminated -Epoch at which the deal was terminated abnormally, or -1.
- *
-

Storage power actor

Storage power actor is responsible for keeping track of the storage power allocated at each storage miner. The ActorCode for the built-in storage power actor ishex"0004" which will be used to call methods in the storage power actor. You also need to specify the method number for the method you want to invoke. Please refer to each method for its method number.

CreateMiner

```

Copy funcCreateMiner(params CreateMinerParams) CreateMinerReturn {}

```

Create a new miner for the owner address and worker address.

uint CreateMinerMethodNum = 1173380165.

Params:

- struct
- CreateMinerParams
-
    - bytes
-
    - Owner - the address of the owner.
-
    - bytes
-
    - Worker - the address of the worker.
-
    - RegisteredPoStProof
-
    - WindowPoStProofType - the type of RegisteredPoStProof.
-
    - bytes
-
    - Peer - peerID.
-
    - bytes[]
-
    - Multiaddrs - the multi-address which is used to control the newly created miner.
- *
-

Results:

- CreateMinerReturn
-
    - bytes
-
    - IDAddress - The canonical ID-based address for the actor.
-
    - byte
-
    - : RobustAddress -A more expensive but re-org-safe address for the newly created actor.
- *
-

NetworkRawPower

```

Copy funcNetworkRawPower() NetworkRawPowerReturn {}

```

Return the total raw power of the network.

uint NetworkRawPowerMethodNum = 931722534.

Params:

- null
-

Results:

- int256
- NetworkRawPowerReturn - the raw storage power of the whole network.
-

MinerRawPower

```

Copy funcMinerRawPower(params MinerRawPowerParams) MinerRawPowerParams {}

```

Return the raw power claimed by the specified miner and whether the miner has more than the minimum amount of active storage.

uint MinerRawPowerMethodNum = 3753401894.

Params:

- MinerRawPowerParams
-
    - uint64
-
    - Miner - Miner ID
- *
-

Results:

- struct
- MinerRawPowerParams
- int256
- RawBytePower - the row power of the miner.
- bool
- MeetsConsensusMinimum - if the miner power meets the minimum for consensus.
-

MinerCount

```

Copy funcMinerCount() MinerCountReturn {}

```

Returns the total number of miners created, regardless of whether or not they have any pledged storage.

uint MinerRawPowerMethodNum = 3753401894.

Params:

- null
-

Results:

- uint64
- MinerCountReturn - the count of the miners that the caller address has.
-

MinerConsensusCount

```

Copy funcMinerConsensusCount() MinerConsensusCountReturn {}

```

Returns the total number of miners that have more than the minimum amount of active storage.

uint MinerConsensusCountMethodNum = 196739875.

Params:

- null
-

Results:

- uint64
- MinerConsensusCountReturn - the count of the miners that meet the consensus minimum that the caller address has.
-

Verified registry actor

Verified registry actor is responsible for managing verified clients. The ActorCode for the verified registry built-in actor ishex"0006" which will be used to call the exported methods in the verified registry built-in actor. You need to specify the method number for the method you want to invoke. Please referer to each method for its method number.

AddVerifiedClient

```

Copy funcAddVerifiedClient(params AddVerifiedClientParams) EmptyValue {}

```

To add a verified Client address to Filecoin Plus program.

uint constant AddVerifierClientMethodNum = 3916220144.

Params:

- struct
- AddVerifierClientParams
-
    - bytes
-
    - Address - the verified client address
-
    - int256
-
    - Allowance - approved DataCap for this verified client
- *
-

Results:

- struct
- EmptyValue.
-

RemoveExpiredAllocations

```

Copy funcRemoveExpiredAllocations(params RemoveExpiredAllocationsParams) RemoveExpiredAllocationsReturn {}

```

Remove the expired DataCap allocations and reclaim those DataCap tokens back to the client. If the allocation amount is not specified, all expired DataCap allocations will be removed.

uint RemoveExpiredAllocationsMethodNum = 2873373899.

Params:

- struct
- RemoveExpiredAllocationsParams
-
    - uint64
-
    - Client - the client address to remove the expired tokens from.
-

    - uint64[]
-
    - AllocationIDs - List of allocation IDs to attempt to remove. If empty, this method will remove all eligible expired tokens.
- *
-

Results:

- struct
- RemoveExpiredAllocationsReturn
-
    - unit64[]
-
    - Considered - Allocation IDs are either specified by the caller or discovered to be expired.
-
    - BatachReturn
-
    - Results - results for each processed allocation.
-
    - int256
-
    - DataCapRecoverd - The amount of DataCap token reclaimed for the client.
- *
-

GetClaims

```

Copy funcGetClaims(params GetClaimsParams) GetClaimsReturn {}

```

Return a list of claims corresponding to the requested claim ID for a specific provider.

uint GetClaimsMethodNum = 2199871187.

Params:

- struct
- GetClaimsParams
-
    - uint64
-
    - Provider - the provider address.
-
    - unit64[]
-
    - ClaimIDs - A list of Claim IDs for a specific provider.
- *
-

Results:

- struct
- GetClaimsReturn
-
    - struct
-
    - BatchReturn
-
        - uint32
-
        - SuccessCount - total successes in the batch.
-
        - struct

-
  -
    - FailCode[] {uint32
-
  -
    - idx,uint32
-
  -
    - code} - list of failure code and index for all failures in batch.

-
  - struct Claim[]
-
  - Claims - list of Claims returned.
-
  -
    - uint64
-
  -
    - Provider - The provider that is storing the data.
-
  -
    - uint64
-
  -
    - Client - The client that originally allocated the DataCap.
-
  -
    - bytes
-
  -
    - Data - Identifier for the data committed.
-
  -
    - uint64
-
  -
    - Size - The size of the data.
-
  -
    - int64
-
  -
    - TermMin - The minimum period after the term starts, during which the provider must commit to storing data.
-
  -
    - int64
-
  -
    - TermMax - The maximum period after the term starts for which the provider can earn Quality Adjusted power for the data.
-
  -
    - int64
-
  -
    - TermStart - the epoch at which the piece was committed.
-
  -
    - unit64
-
  -
    - Sector - ID of the provider’s sector in which the data is committed.

- *
-

ExtendClaimTerms

```

Copy funcExtendClaimTerms(params ExtendClaimTermsParams) ExtendClaimTermsReturn {}

```

Extends the maximum term of some claims up to the largest value they could have been originally allocated. This method can only be called by the claims' client.

uint ExtendClaimTermsMethodNum = 1752273514.

Params:

- struct
- ExtendClaimTermsParams
-
    - struct ClaimTerm[]
-
    - Terms
-
    -
        - uint64
-
    -
        - Provider - The provider address which stores the data.
-
    -
        - uint64
-
    -
        - CliamID - Claim ID.
-
    -
        - int64
-
    -
        - TermMax - The max chain epoch to extend.

- *
-

Results:

- struct
- ExtendClaimTermsReturn
-
    - struct
-
    - BatchReturn
-
    -
        - uint32
-
    -
        - SuccessCount - total successes in the batch.
-
    -
        - struct
-
    -
        - FailCodes[] {uint32
-
    -
        - idx,uint32
-
    -
        - code} - list of failure code and index for all failures in batch.

- *
-

RemoveExpiredClaims

```

Copy funcRemoveExpiredClaims(params: RemoveExpiredClaimsParams) RemoveExpiredClaimsReturn {}

```

To remove a claim with its maximum term has elapsed.

uint RemoveExpiredClaimsMethodNum = 2873373899.

Params:

- struct
- RemoveExpiredClaimsParams
-
    - uint64
-
    - Provider - the provider address.
-
    - unit64[]
-
    - ClaimIDs - A list of Claim IDs with an expired term. If no claims are specified, all eligible claims will be removed.
- *
-

Results:

- struct
- RemoveExpiredClaimsReturn
-
    - uint64[]
-
    - Considered - a list of IDs of the claims that were either specified by the caller or discovered to be expired.
-
    - struct
-
    - BatchReturn
-
    -
        - uint32
-
    -
        - SuccessCount - total successes in the batch
-
    -
        - struct
-
    -
        - FailCodes[] {uint32
-
    -
        - idx,uint32
-
    -
        - code} - list of failure code and index for all failures in batch.

- *
-

Previous Built-in actors Next Filecoin.sol

Last updated4 months ago