



Ownerable
+ origOwner: address + TransferOwnership: event
+ owner(): address + onlyOwner(): modifier + isOwner(address): bool + transferOwnership(address): void

Roles
+ Role: mapping(address => bool)
+ add(Role, address): void + remove(Role, address): void + has(Role, address): void

CustomerRole
+ CustomerAdded: event + CustomerRemoved: event + consumers: Role
+ isConsumer(address): bool + addConsumer(address): void + renounceConsumer(address): void + onlyConsumer: modifier

DistributorRole
+ DistributorAdded: event + DistributorRemoved: event + distributors: Role
+ isDistributor(address): bool + addDistributor(address): void + renounceDistributor(address): void + onlyDistributor: modifier

FarmerRole
+ FarmerAdded: event + FarmerRemoved: event + farmers: Role
+ isFarmer(address): bool + addFarmer(address): void + renounceFarmer(address): void + onlyFarmer: modifier

RetailerRole
+ RetailerAdded: event + RetailerRemoved: event + retailers: Role
+ isRetailer(address): bool + addRetailer(address): void + renounceRetailer(address): void + onlyRetailer: modifier

SupplyChain
+ owner: address + upc: uint + sku: uint + items: mapping(uint->item) + itemsHistory: mapping(uint->string[]) + state: enum + item: struct
+ kill(): void + harvestItem(uint, address, string, string, string, string, string): void + processItem(uint): void + packItem(uint): void + sellItem(uint, uint): void + buyItem(uint): void + shipItem(uint): void + receiveItem(uint): void + purchaseItem(uint): void + fetchItemBufferOne(uint): uint uint address address string string string string

```
+ fetchItemBufferOne(uint): uint, uint, address, address, string, string, string, string
+ fetchItemBufferTwo(uint): uint, uint, uint, string, uint, uint, address, address, address
+ onlyOwner(): modifier
+ verifyCaller(address): modifier
+ paidEnough(uint): modifier
+ checkValue(uint): modifier
+ harvested(uint): modifier
+ processed(uint): modifier
+ packed(uint): modifier
+ forSale(uint): modifier
+ sold(uint): modifier
+ shipped(uint): modifier
+ received(uint): modifier
+ purchased(uint): modifier
```



