



# Managing Accounts

---

This chapter contains the following sections:

- [Accounts, page 1](#)
- [Adding a Custom Account Type, page 1](#)

## Accounts

You can use Open Automation to add a new custom Account type to Cisco UCS Director.

A custom account type provides new Cisco UCS Director data infrastructure that enables you to work with accounts in new ways. For example, a new account type allows you to manage and report on Cisco UCS Director managed elements in new ways.

## Adding a Custom Account Type

To add an account type, perform these tasks:

- Extend the **com.cloupia.lib.connector.account.AbstractInfraAccount** class.
- Define values for AccountType and AccountLabel
- Assign mandatory parameters for the AccountTypeEntry. These include setPodTypes, setAccountClass, setAccountType and setAccountLabel
- Complete implementation for Connectivity Test

The code snippet below demonstrates how to provide new Account details.

```
//This class is used to register a new connector into the UCSD.  
AccountTypeEntry entry=new AccountTypeEntry();  
  
//Set implementation class for Account type  
entry.setCredentialClass(FooAccount.class);  
  
//Account Type  
entry.setAccountType(FooConstants.INFRA_ACCOUNT_TYPE);  
  
//Account Label which will be shown in UI  
entry.setAccountLabel(FooConstants.INFRA_ACCOUNT_LABEL);
```

```

//Account Category like Compute, Storage, Network or Multi-Domain.
entry.setCategory(InfraAccountTypes.CAT_NETWORK);

//This is mandatory , report generation on context Level for the new account type
entry.setContextType(ReportContextRegistry.getInstance().
getContextByName(FooConstants.INFRA_ACCOUNT_TYPE).getType());

//Account class like Physical-1, Virtual-2, Network-5, Multi-domain-3
//or Other-4
entry.setAccountClass(AccountTypeEntry.PHYSICAL_ACCOUNT);

// This will be used along with the account name to show
// in the System Tasks Report.
entry.setInventoryTaskPrefix("Open Automation Inventory Task");

//Set time frequency to collect account Inventory
entry.setInventoryFrequencyInMins(15);

//Supported POD types for this connector.
entry.setPodTypes(new String[]{"FooStack"});

//To add this account type entry
PhysicalAccountTypeManager.getInstance().addNewAccountType(entry);

// This is mandatory, to test the connectivity of the new account. The
// Handler should be of type PhysicalConnectivityTestHandler.
entry.setTestConnectionHandler(new FooTestConnectionHandler());
// This is mandatory, we can implement inventory listener according to
// the account Type , collect the inventory details.
entry.setInventoryListener(new FooInventoryListener());

//This is mandatory , to show in the converged stack view
entry.setConvergedStackComponentBuilder(new DummyConvergedStackBuilder());

//This is required to show up the details of the stack view in the GUI
entry.setStackViewItemProvider(new DummyStackViewItemProvider());

// This is required credential.If the Credential Policy support is
// required for this Account type then this is mandatory, can implement
// credential check against the policyname.
entry.setCredentialParser(new FooAccountCredentialParser());

```

**Important**


---

Refer to the SDK samples. See `com.cloupia.feature.foo.accounts.FooAccount` and `com.cloupia.feature.foo.FooModule` for implementations.

---