Use Adapter SDK

This section contains the following topics:

Prerequisites

To start using the CWM Adapter SDK, you need to install a **Golang** environment, Protocol buffers, dedicated **go** plugins and download the Adapter SDK contained in the CWM software package.

Install Protocol buffers

To define an adapter interface and generate the input and output parameters, you need the Protobufs compiler. Follow the installation instructions dedicated for your OS: https://grpc.io/docs/protoc-installation/. Note that you need at least version **3.15** (proto3).

Install Go and plugins

To develop and test an adapter, you need to install the **Golang** environment. Follow the installation instructions dedicated for your OS: https://grpc.io/docs/protoc-installation/.

Step 1 Install additional protocol compiler plugins for **go**:

```
go install google.golang.org/protobuf/cmd/protoc-gen-go@v1.28
go install google.golang.org/grpc/cmd/protoc-gen-go-grpc@v1.2
go install github.com/pseudomuto/protoc-gen-doc/cmd/protoc-gen-doc@latest
```

Step 2 Install protocol compiler plugin for **JSON schema**:

```
go install github.com/chrusty/protoc-gen-jsonschema/cmd/protoc-gen-jsonschema@latest
```

Step 3 Update your system PATH so that the `protoc` compiler can find the plugins:

```
export PATH="$PATH:$(go env GOPATH)/bin"
```

Get CWM Adapter SDK

Go to Cisco Software Download page to download the CWM Software Package, where the Adapter SDK binary resides.

Include the location of adapter developer binaries by setting the environment variable path:

```
export PATH=/path/to/adapter-dev-binaries:$PATH
```

Note Remember to replace the `/path/to/adapter-dev-binaries` with your actual path.

Overview of commands

The Adapter SDK application offers the following set of commands for managing an adapter:

- `cwm-sdk version` - display the version of cwm-sdk application.
- `cwm-sdk create-adapter` - create a go module with a package and the corresponding .proto files.
- `cwm-sdk extend-adapter` - add a new feature to an existing adapter (go package and .proto files).
- `cwm-sdk update-adapter` - update activities, input and output (go code).
- `cwm-sdk upgrade-adapter` - upgrade the adapter to match CWM.
- `cwm-sdk create-installable` - create an archive installable by CWM.

Create new adapter

To create an adapter, open a command-line terminal and run:

```
cwm-sdk create-adapter [options] -product <product-name>
```

Note While the `-product` parameter is required for adapter creation, other options can be skipped.

Options

These are the options you can add to the `create-adapter` command:

Option	Data type	Requirement	Description
`-exclude-resource`	string	optional	skip creation of the `.resource.proto` file from template.
`-feature`	string	optional	provide name for the go package assigned to activities (default: "<adapter-name>").
`-go-module`	string	optional	provide name for the module assigned to the go.mod file (default: "www.cisco.com/cwm/adapters/<vendor>/<adapter-name>").

Option	Data type	Requirement	Description
`-ignore-template`		optional	skip generation of example code in the go and proto files.
`-location`	string	optional	point to adapter location (default: current directory).
`-os-architecture`	string	optional	define architecture in which adapter is developed. Valid options are: 'linux','mac-intel','mac-arm' and 'windows' (default: "linux").
`-product`	string	required	provide name for the go module corresponding to the product name you create an adapter for.
`-vendor`	string	optional	provide unique name for the company creating the adapter (default "cisco").
`-verbose`	string	optional	output progress info. Options are: `off`, `on` and `very` (default "off").

Output

Once the command is executed, verify the generated output inside the new adapter directory:

- `<adapter-name>/adapter.properties`
- `<adapter-name>/go/go.mod`
- `<adapter-name>/proto/<vendor\>.<product\>.common.adapter.proto`
- `<adapter-name>/proto/<vendor\>.<product\>.<feature\>.adapter.proto` (if you defined the `-feature` option)
- `<adapter-name>/Makefile`

Extend adapter with features

To add a feature (a **go package**) for an adapter, open a terminal and from your main adapter directory, run:

```
cwm-sdk extend-adapter [options] -feature <feature_name>
```

Options

Option	Data type	Requirement	Description
`-activity`	string	optional	Provide name for a new activity to add.
`-feature`	string	required	Provide name for the feature to add (default: "<adapter-name>").
`-ignore-template`		optional	Skip generation of example code in the go and proto files.
`-location`	string	optional	Point to the location of adapter to which you add the feature (default: current directory).
`-verbose`	string	optional	Output progress info. Options are: `off`, `on` and `very` (default "off").

Output

Once the command is executed, verify the generated output inside the new adapter directory:

- `<adapter-name>/proto/<vendor>.<module>.<package>.adapter.proto`

Generate/update activity

Once you have an adapter with at least one feature added, you can proceed to creating activities. Activities are defined within the `.proto` file for a specific feature (**go package**). You can do this manually or use the OASX extension for OpenAPI-enabled services to automatically build of message logic in the `.proto` files.

Once the activities are defined, you can generate the input and output files for the adapter. Go to your main adapter directory and run:

```
cwm-sdk update-adapter
```

Options

- `-features` *string* - provide a comma-separated list of features to update.
- `-location` *string* - point to location of adapter to update (default: current directory).
- `-verbose` *string* - output progress info. Options are: `off`, `on` and `very` (default "off").

Output

Once the command is executed, verify the generated output inside the adapter directory:

- `go/<feature\>/<vendor>.<product>.<feature>.adapter.pb.go`
- `go/common/<vendor>.<product>.common.adapter.pb.go`

The `.pb.go` files contain **go** structs defining the input and output parameters of the adapter. They shouldn't be altered manually.

Once the command is executed, verify the generated output inside the adapter directory:

- `go/<feature\>/activities.go`

The `activities.go` file contains stubs for the gRPCs defined in the `.adapter.proto`. Once generated, you can add functionality to the activities by defining the message.

Upgrade adapter

To upgrade the **go module** to contain matching versions for **go** and required imports, go to the root directory of your adapter and run:

```
cwm-sdk upgrade-adapter [options]
```

Options

- `-cwm-version` *string* - provide the version of CWM to upgrade to (default is latest).
- `-location` *string* - point to location of adapter to upgrade (default: current directory).

- `-verbose` *string* - output progress info. Options are: `off`, `on` and `very` (default "off").

Output

- go/go.mod

The `go.mod` file module will be modifed allowing the adapter to be installed correctly.

Export library to local directory

The `cwm-sdk` uses the SDK go module for performing tasks. In certain cases you might want to have the SDK go module created in the adapter directory beforehand. For this purpose, use the `export-lib` command.

The `export-lib` command comes with the following options:

Option	Data type	Description	Status
-location	string	Provide location where SDK lib should be created. (default: current directory)	optional
-verbose	string	Show command progress info. Options are: off, on, or very.	optional

Create adapter installable version

To create a `tar.gz` archive for installing your adapter for different operating systems, go to the root directory of your adapter and run:

```
cwm-sdk create-installable [options]
```

Generates code based on the proto file.

Options

- `-cwmversion` *string* - provide a CWM version to match the created installable (default is latest).
- `-location` *string* - point to where the installable should be created (default: current directory).
- `-verbose` *string* - output progress info. Options are: `off`, `on` and `very` (default "off").

Output

- out/<vendor>-<product>-v<X.Y.Z>.tar.gz

!!! note The generated archive contains the adapter **go** module and proto files. The **go** module is modified using the **go** vendor command in order to not have any external dependencies.