API Field Deployer Instructions

Installation

You can install the package in your org using this link:
https://login.salesforce.com/packaging/installPackage.apexp?p0=04tgL00000012Nx
Alternatively, use the package id: 04tgL00000012Nx
This can be done in any org of your choosing, even a trailhead org.

Setup

There are a few things you will need to do before you can use the tool. Follow these steps to set up everything you need:

1. Create the default Custom Metadata Type: After installing the package, you should already have the API_Type Custom Metadata Type in your org. Find it by going to Setup -> Custom Metadata Types.
2. Click “Manage Records” next to API_Type, then click “New”: You will need to create a default record for the code to use so that it uses the correct API. In this case, we are using the Metadata API, so the record should have these values (Note that you need to type out the word “Default” - not leave the default value which is blank):
   - Label: Default
   - API Type Name: Default
   - API Type: Metadata
3. Click Save

Next you need to set up the CSVs that will be used. It is important that these files are set up correctly and that they are set up with data Salesforce would expect. Duplicate fields or missing information can cause errors.

1. For your new field CSV, make sure the name of the file is <objectName>.csv.
2. Ensure that the columns of the CSV represent these values in order (Note that the header is optional, but be sure to know if your file has a header): Field API Name, Field Label, Data Type, Help Text, Required, Unique, Case Sensitive, External Id, Picklist Options, Related Object.

    a. The last two columns are optional. Fill out information in the other fields to avoid errors.
    b. Required through External ID are all boolean values.
    c. Picklist options are separated by | with no spaces between them.

3. For the permissions CSV, the name can be whatever you want as long as you remember it.
4. Ensure that the columns represent these values in order (header also optional here): Name, Profile or Permission Set, Readable, Editable.
    a. Readable and Editable are Boolean.
    b. Write out “Profile” or “Permission Set” depending on what you need. Make sure that Profile or Permission Set exists in your org.
    c. Name needs to be the API name. API names for standard profiles are available online (i.e. System Administrator’s API Name is Admin).
5. In Salesforce, click the App Launcher and go to Files. Upload both CSVs separately and note the names of the files (without the .csv).

Run the Tool

In my testing, I used the Developer Console and Execute Anonymous to run the code. There may be multiple ways to run the code at this point, but I will walk through how to do it with execute Anonymous:

1. In Salesforce, click the gear icon and click Developer Console.
2. Click Debug -> Open Execute Anonymous Window.
3. Type this line of code:
    a. CreateCustomFields.createFieldsFromCsv(<'fieldFileName'>, <fieldCsvHasHeader>, < 'permissionsFileName'>, <permissionsCsvHasHeader>);
    b. The file names are Strings and the hasHeaders are booleans. So if you have a csv for your new fields called Account with a header row, and a Permissions csv called Account-Permissions without a header row, the code would look like this: CreateCustomFields.createFieldsFromCsv(‘Account’, true, ‘Account-Permissions’, false);
    c. Note that custom fields will need the __c as part of the file name.
4. That code will run both API callouts and the dml statement for the Permission Set. Once it completes, check your org for the new fields and updated permissions.