Bill Anderson

Bill.anderson@blusoftwave.com

Abstract

Simple steps using SFDX to Package multiple components

Sample DX Packing

Multiple Packages within a Project

# Multiple Packages

This is a simple document that outlines steps taken to create multiple 2GP unlocked packages with the same project file. The steps are outlined along with the final result. We package only two of the components (**Core** and Growth-1) and leave the final one, *Common*, to the user as an exercise.

# General Comment

Packaging process can be broken down in steps. As SFDX is evolving, some steps may change overall time. Thus, this document will need updating.

In addition, some of the larger questions need to be answered (and not covered here):

* Who owns the content? Or, is it a shared resource?
  + Should Governance, ARB, or some Committee decide ownership or shared resource?
* Does the content require sign-off and/or governance (i.e. Data Architect for compliance, etc.)?
* Naming Conventions (package names, version numbers, etc.)

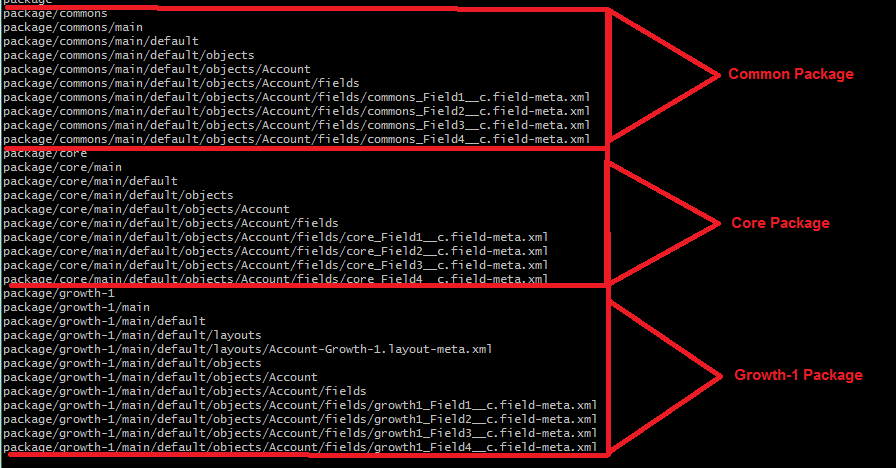
For example, should all custom and standard objects be owned and controlled by Central/Core group in order to ensure proper governance is followed as prescribed by the Data Architect and/or Enterprise Architect?

These are general comments that require a larger, and higher-level, discussion.

# Sample

Initially, we have three components which we will package 2 with steps in this document:

* **Core** (to be Packaged)
* *Common* (will not Package in this scenario, but can)
* **Growth-1 (to be Packaged)**



This is the current structure of source. This will evolve as we progress in the DX packaging. The sections which follow go through the process of creating two packages.

This document does not go into getting access to a Dev Hub nor the steps in creating a Scratch Org. The Scratch Org will be working with has an alias name of **mpd**.

Below is the basic flow in creating our packages:

# SFDX-Project.JSON

At the beginning we have a basic structure for our *sfdx-project.json*. This will evolve as we create packages.

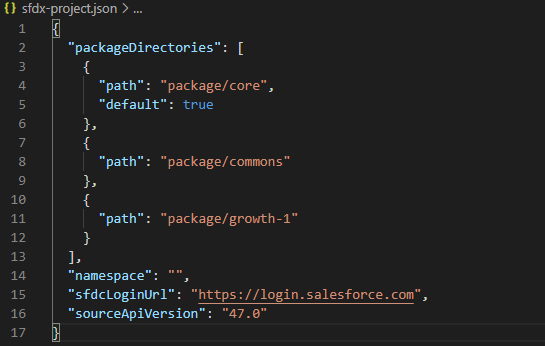
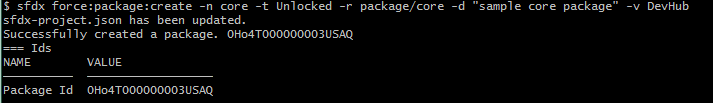


Figure Beginning Project File

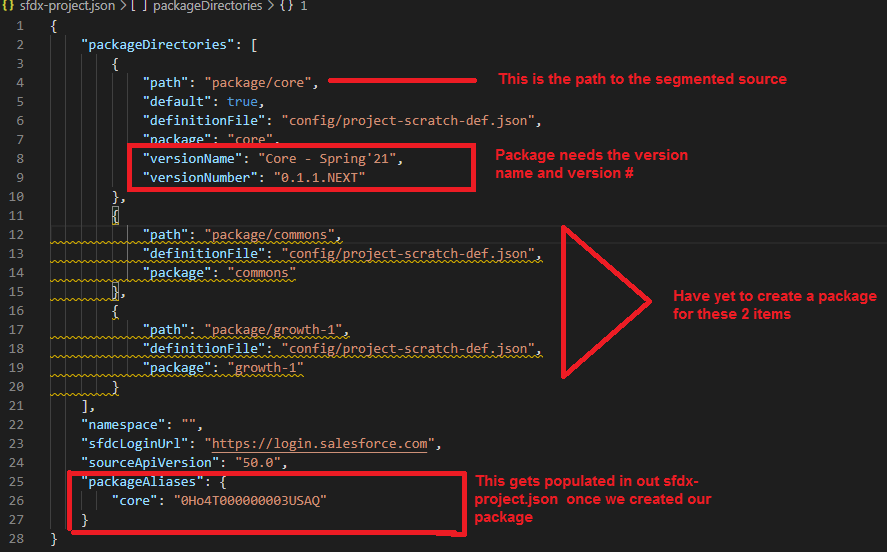
# Package CORE

## Create a Package

**Create a package named core.**



After we have created our package, ***sfdx*** command updates the *sfdx-project.json* file.



Please note, before we created the package, we added line 6-9:

* Line-6 – definition of the Org Shape
* Line-7 – package name
* Line-8 – package version name
* Line-9 – package version number

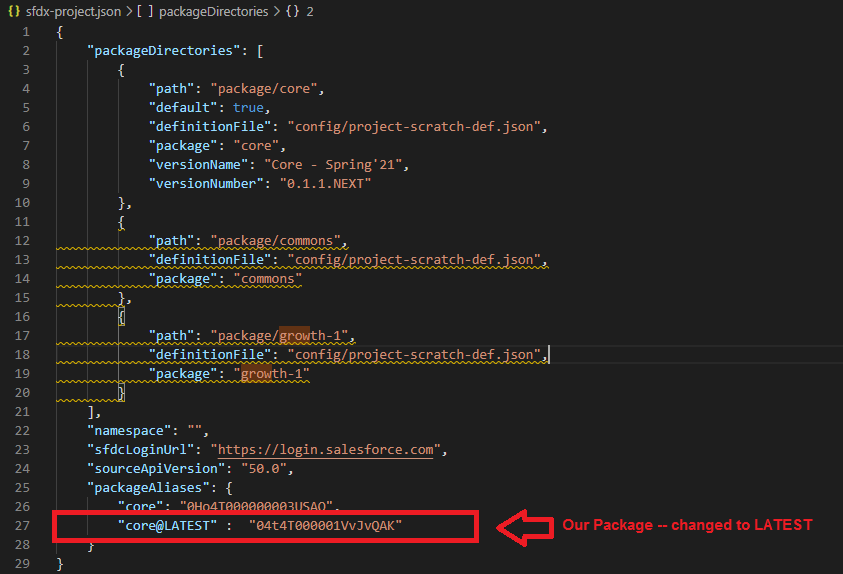
After the created package command was done, ***sfdx*** added lines 25-27.

## Create a Package Version

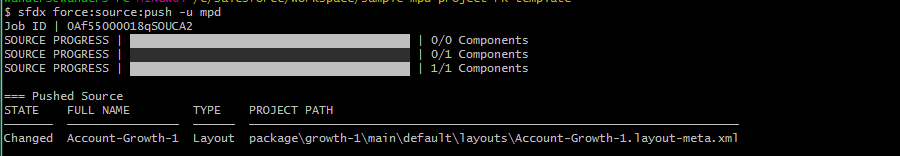
Next, we create the package version of **core**.



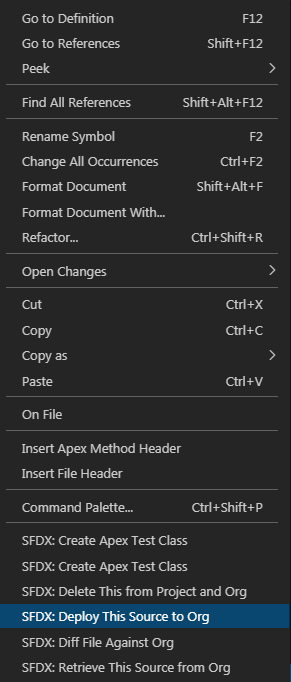
After we created the package version, ***sfdx*** updates the *sfdx-project.json* file as follows:



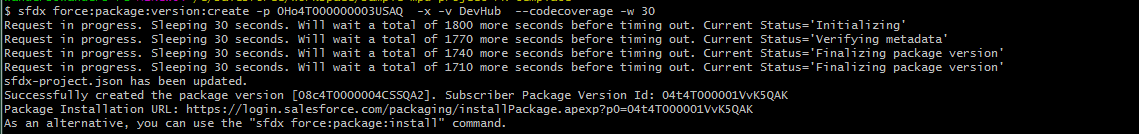
Next, I made some changes to the Account layout in my project and pushed to the Scratch Org (for Growth-1). This will not affect this package.



Developers will go back and forth pull/push scenario with a Scratch Org. If using a non-Scratch Org, you can use the Retrieve/Deploy options in Visual Studio Code.

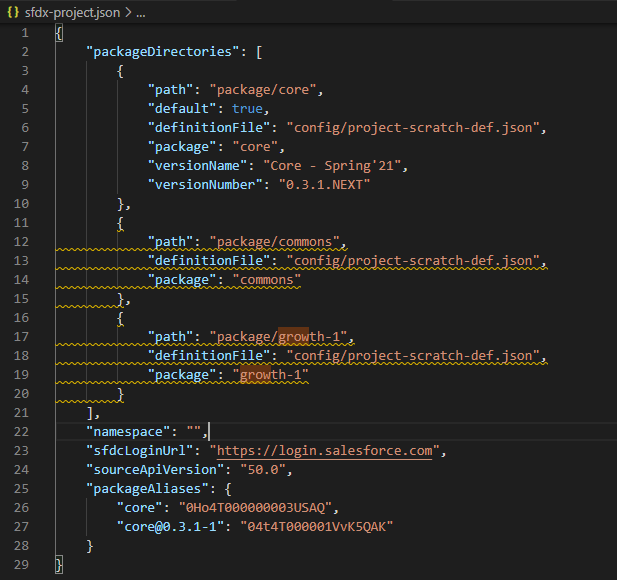


Once we are done, unit tested with code-coverage >= 75%, we can promote. Please note, initially when we created our package version, **we DID NOT** include code coverage. Why? We were in the process of updating our Beta version and I had not validated nor tested in my Scratch Org. Once we have tested and validated, we are ready to create a version to promote. But, first one **MUST** create a version with code-coverage option!





The *sfdx-project.json*, after promotion of core, looks like:



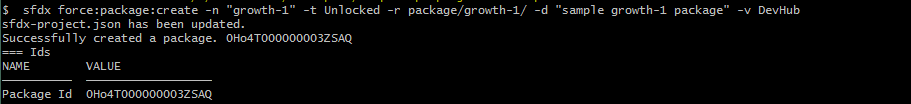
# Package Growth-1

We have packaged **Core**, now, let us package **Growth-1**. **Growth-1** adds new fields and a layout to the *Account*.

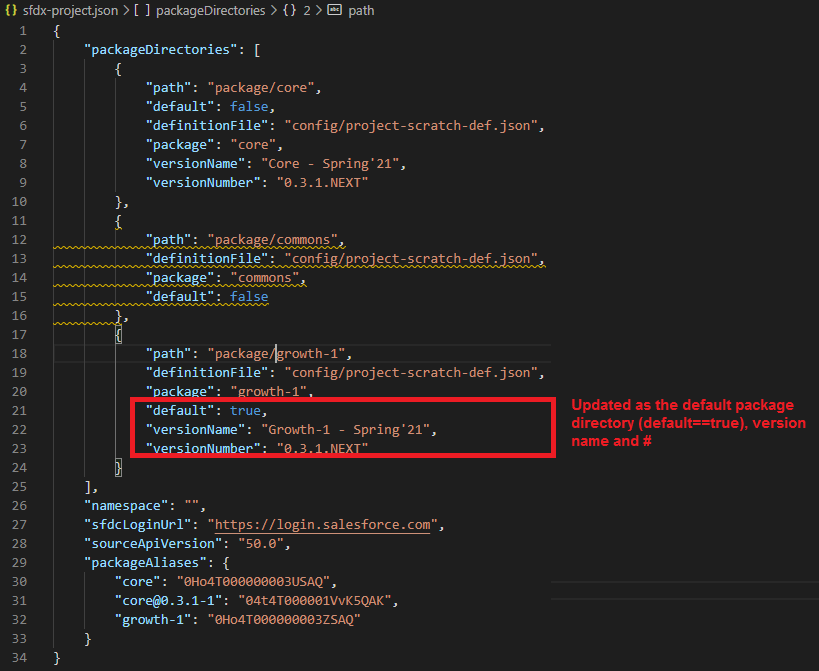
## Create a Package

Here we create a package named **growth-1**.

sfdx force:package:version:create -p 0Ho4T000000003ZSAQ -x -v DevHub --codecoverage -w 30

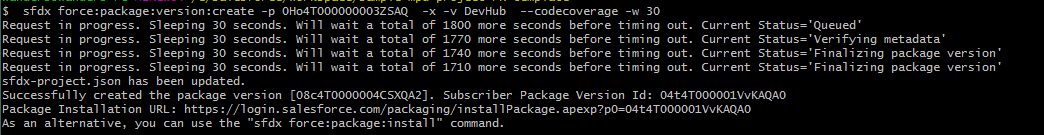


After we have created our package, ***sfdx*** command updates the *sfdx-project.json* file.



## Create a Package Version

Next, we create the package version of **growth-1**. Please note, for this version we are adding the ***codecoverage*** switch.

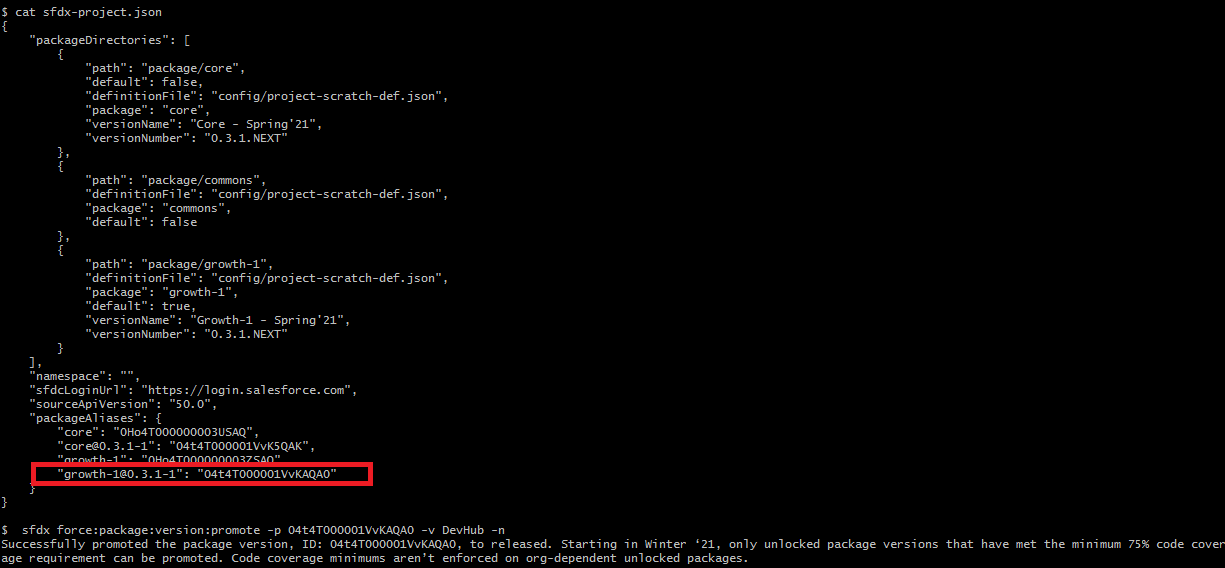


After we created the version, sfdx updates the *sfdx-project.json* file as follows:



Because we used the ***–codecoverage*** switch, we can now, promote our growth-1 package,

sfdx force:package:version:promote -p 04t4T000001VvKAQA0 -v DevHub -n



# Appendix: Zipped file of Project

Contents are the final result of the project.

