**Vlocity Build Tool Config Steps**

1. **Install Node.js**

* Download and Install Node at:
* <https://nodejs.org/>
* Note - This tool requires Node Version 10+.
* Use node -v to find out which version you are on.(Run this command on CMD)
* Install or Update Vlocity Build through NPM
* You can install or update this package like any other Node Package!
* Run the below commands from CMD

npm install --global vlocity

vlocity help

1. **Cloning Vlocity Build - Not Recommended**
   * Go to the below GitHub Link
   * <https://github.com/vlocityinc/vlocity_build#installation-and-update-instructions>
   * Then Click on Code
   * And then select Download Zip
   * And then Unzip the folder in any specific location where you will perform all Build Tool related activity
2. **Getting Started**

* Create your own property files for your Source and Target Salesforce Org With the following:

sf.username = < Salesforce Username >

sf.password = < Salesforce Password + Security Token >

sf.loginUrl = < https://login.salesforce.com or https://test.salesforce.com for Sandbox >

* It is a best practice to don’t rely on a single build.properties file instead use named properties for each org: like build\_source.properties and build\_target.properties

1. **Running the Process**

* Commands follow the syntax:

vlocity packExport -propertyfile <filepath> -job <filepath>

* Property File
  + The propertyfile is used to provide the credentials of the org you will connect to. It will default to build.properties if no file is specified.
* Job File
  + The Job File used to define the project location and the various settings for running a DataPacks Export / Deploy.

1. **Step by Step Guide**

* Once you have your build\_source.properties file setup, you can get started with migration with the following:

Simple Export

projectPath: ./example\_vlocity\_build

queries:

- VlocityDataPackType: DataRaptor

query: Select Id from %vlocity\_namespace%\_\_DRBundle\_\_c where Name = 'DataRaptor Migration' LIMIT 1

* Example.yaml shows the Simplest Job File that can be used to setup a project:
* After creating the Example.yaml file, run the following command to export with this Job File:
* vlocity -propertyfile build\_source.properties -job Example.yaml packExport

Note:- Indentation is very important in job (.yaml) file else you might encounter an error

Which will produce the following output:

Salesforce Org >> source\_org@vlocity.com

VlocityDataPackType >> DataRaptor

Query >> Select Id from vlocity\_cmt\_\_DRBundle\_\_c where Name = 'DataRaptor Migration' LIMIT 1

Records >> 1

Query Total >> 1

Initializing Project

Exporting >> DataRaptor a191N000012pxsYQAQ

Creating file >> example\_vlocity\_build/DataRaptor/DataRaptor-Migration/DataRaptor-Migration\_DataPack.json

Salesforce Org >> source\_org@vlocity.com

Current Status >> Export

Successful >> 1

Errors >> 0

Remaining >> 0

Elapsed Time >> 0m 5s

Salesforce Org >> source\_org@vlocity.com

Export success:

1 Completed

This has exported data from the org specified in your build\_source.properties file and written it to the folder example\_vlocity\_build specified in the Example.yaml file found at dataPacksJobs/Example.yaml

**Org to Org Migration**

All together the commands are

# Source Org

vlocity -propertyfile build\_source.properties -job EPC.yaml cleanOrgData

vlocity -propertyfile build\_target.properties -job Platform.yaml packUpdateSettings

vlocity -propertyfile build\_source.properties -job EPC.yaml packExport

vlocity -propertyfile build\_source.properties -job EPC.yaml packRetry # If any errors

vlocity -propertyfile build\_source.properties -job Platform.yaml validateLocalData

vlocity -propertyfile build\_source.properties -job Platform.yaml validateLocalData –fixLocalGlobalKey #if any error encounters

# Target Org

vlocity -propertyfile build\_target.properties -job EPC.yaml cleanOrgData

vlocity -propertyfile build\_target.properties -job Platform.yaml packUpdateSettings

vlocity -propertyfile build\_target.properties -job EPC.yaml packDeploy

vlocity -propertyfile build\_target.properties -job EPC.yaml packRetry # If any errors

**Explanation of the above-mentioned commands**

Create Property Files for each Org. build\_source.properties and build\_target.properties

Create a Job File which identifies your projectPath and the queries for the DataPack Types you would like to export. To export the Vlocity Product Catalog, use the following Platform.yaml file:

projectPath: vlocity

queries:

- OmniScript

- IntegrationProcedure

- DataRaptor

- VlocityUILayout

- VlocityUITemplate

1. Ensure High Data Quality in the Source Org by running:  
   vlocity -propertyfile build\_source.properties -job Platform.yaml cleanOrgData
2. Run the Export:  
   vlocity -propertyfile build\_source.properties -job Platform.yaml packExport

Settings will automatically be deployed if it is the first time you are deploying to this org. This will take 30 seconds and will only need to be rerun when the settings change.

1. If you encounter any Errors during Export, please evaluate their importance. Any error during export points to potential errors during deploy. Once errors are fixed, run the following to re-export any failed data:  
   vlocity -propertyfile build\_source.properties -job Platform.yaml packRetry  
   If your Export fails midway through due to connection issues, you can also use the following to pick the export back up where it left off:  
   vlocity -propertyfile build\_source.properties -job Platform.yaml packContinue
2. Check the Exported Data for Potential Issues:  
   vlocity -propertyfile build\_source.properties -job Platform.yaml validateLocalData  
   This will give a summary of Duplicate and Missing Global Keys. Use the argument --fixLocalGlobalKeys to automatically add missing and change duplicate keys. However, it is generally better to fix the data in the Org itself.
3. Ensure High Data Quality in the Target Org by running:  
   vlocity -propertyfile build\_target.properties -job Platform.yaml cleanOrgData
4. Update the DataPack Settings in the Target Org:  
   vlocity -propertyfile build\_target.properties -job Platform.yaml packUpdateSettings
5. Run the deploy:  
   vlocity -propertyfile build\_target.properties -job Platform.yaml packDeploy
6. If you encounter any Errors during deploy, they must be fixed. But first, evaluate whether the error has been mitigated by later uploads of missing data. Run:  
   vlocity -propertyfile build\_target.properties -job Platform.yaml packRetry  
   Which will retry the failed DataPacks, often fixing errors due to issues in the order of deploy or Salesforce Governor limits. packRetry should be run until the error count stops going down after each run.