# SuiteCloud Development Framework







#### AUTHORS

Bryan Willman

Luke Pirtle

RenHong Zhu

Nick Klug

### PUBLISHED DATE

8/12/2019

### LAST UPDATED

8/12/2019

### VERSION HISTORY

Version: V1 Date: 8/12/2019 Description: Initial publication



### **Table of Contents**

Techfino's SuiteCloud Development Framework	1
Audience and Purpose	4
Introduction	4
Guide to Account Customization with SDF	6
Understanding Customization XMLs	7
Customization Object XML	7
Manifest XML	8
SDF Development Environment Setup	8
IDE Integrations	8
Install Eclipse	9
Install SDF Plugin	10
Enable SDF and Create Custom Role for SDF in NetSuite	13
Configure Accounts for SDF Project in Eclipse	17
Change Project Settings	17
Add Customizations to SDF	20
Import by ScriptID	20
Import by Bundle ID	22
Manually Create Customizations	24
Add Dependency to SDF	25
Validation and Deploying SDF Project	26
Validate Project Against Account	26
Deploy SDF Project to Account	27
Sample: Deploy Mass Delete Script from One Account to Another	28
Background	28
Project and Git setup in Eclipse	31
Import Customizations and Validate before Deployment	41
Deploy to Target Account	48
Additional Notes for This Example	50
SDF Gotchas & Our Experience	51
Closing Remarks	53

### **Audience and Purpose**

This guide is written for NetSuite developers and technical resources who are looking for documentation and step-by-step process guides for the SuiteCloud Developer Framework (SDF). After reading this ebook a user should have a good understanding of the framework, its advantages and shortcomings as a DEVOPS tool. The reader is guided through the process of setting up and configuring an SDF project for ongoing development and support of NetSuite customizations. Techfino has also included a sample SDF project to walk the user through a real-world deployment for demonstration purposes.

### Introduction

Over the past few years NetSuite has been hard at work adding exciting features to the NetSuite development process with the introduction of the SuiteCloud Developer Framework or "SDF" as it's most commonly abbreviated. SDF is in essence a platform-agnostic big brother to the Eclipse SuiteCloud IDE many developers are already quite familiar with. However, SDF is a much more mature and established tool that can deploy an entire NetSuite customization project or suiteapp instead of just the source code. Adoption of SDF has been quite slow to the NetSuite developer community due to its added complexity and lack of guides and documentation. This document is meant as a guide to help a developer understand what SDF is, its advantages and how to start using SDF through the examples below.

The precursor to SDF, SuiteCloud IDE, allows a developer to directly upload their Suite-Script source code files into the NetSuite File Cabinet. This streamlines ongoing development by allowing a developer to quickly push code changes to NetSuite without context / window switching to another app. However, it is far from ideal as a deployment tool for NetSuite developers and pales in comparison to other platforms' deployment tools. NetSuite projects are more than just suitescript source code and require customization and configuration in the account to be setup and run. SuiteScripts are reliant on custom records, custom fields, saved searches, workflows etc. which are not supported by SuiteCloud IDE and require manual creation or another tool working in tandem.

Most NetSuite partners utilize SuiteBundler to fill this deployment gap. While it may bridge this gap, in practice SuiteBundler can be incredibly obtuse and difficult to use at times. It often leaves a heavy reliance on documentation and manual process. Suite-Bundler doesn't have version control support and components aren't linked to specific file versions which can lead to conflicting code and deployment issues. SDF aims to address the gaps and pain points of the alternatives by giving a holistic deployment tool that can be used to create and deploy projects seamlessly. To start off, SDF is available for a few supported IDE's (Eclipse, Webstorm) just like its predecessor but can also be used as a command line interface (CLI). This addition allows SDF to be adapted for community plugins and deployment pipelines such as Jenkins. This allows it to be used for small agile projects or full blown application development. SDF also supports account configuration and customization by serializing definitions into xml. In practice, this allows entire projects to be updated and installed to new accounts by a single user without having to leave their IDE. SDF centralizes the code, customization and configuration which eliminates the tedious documentation necessary to keep everything in sync. This also allows for entire projects to be version controlled as opposed to only its source code.

	SuiteCloud IDE	SuiteBundle	SDF
Supports SuiteScript	√	✓	√
Supports Configuration		✓	✓
Centralized Project		✓	√
Can be deployed from an IDE / External Tool	5		5
Can be paired with version control	✓		5
Typical time spent validating deployment	Instant (code out of sync is marked in IDE)	30 min - 5 hours	< 5 min
Typical time spent deploying	< 5 min	< 20 min	< 5 min

NetSuite Customization Deployment Tool Comparison

The biggest drawback to SDF is that it is a big upgrade. NetSuite has added a lot recently to a mostly barren deployment landscape. It's early stages have been buggy with lacking support and documentation. However, since SDF's release, a large number of these issues have been addressed. At Techfino, we are now using SDF for all new development projects and encourage other NetSuite Partners and Customers to do the same.

### **Guide to Account Customization with SDF**

As presented in the Techfino <u>Guide to SuiteBundling in NetSuite</u> by Nick Klug, we assist customers with NetSuite deployments involving the transfer of both configuration and customizations created in one NetSuite account to another. We have been shifting into utilizing the SuiteCloud Development Framework (SDF for short) to help with this process for a variety of reasons which are outlined below. In the following section, we will investigate SDF for managing customizations in multiple environments and deployment. We will also be discussing how SDF differs from SuiteBundling and some of the things we have learned over the past year.

For those not familiar with SDF, it can best be thought of as a new framework in which both configurations and customization objects are represented in XML format. Some of the benefits of this new process include:

- **Greater Control** the files have a much more precise state with SDF. The XML that is present in the SDF project is exactly what is imported into the target system.
- Version Control the XML files can also be stored in a version control system, such as Github or bitbucket. This allows for analysis of changes in the life cycle of a project and can allow for reverting to a previous customization.
- Build-Automation while this is not fully baked as of now, Techfino has discovered a technology stack leveraging SDF that allowed us to successfully automate the deployment of versioned changes from one account to another. SDF is an exciting, enabling technology that will allow NetSuite to advance into more of a mature development and automated build deployment platform allowing for Continuous Integration and Test Automation to follow.

When starting to utilize the SDF platform, an easy way to start is to simply create a few basic scripts and fields within NetSuite. Then, import them into SDF and examine the new entries in the SDF project. This gives a good starting point for your customization project. See the Sample Mass Delete Script Deployment section for a "Hello World" example of deploying a simple development package via SDF.

### **Understanding Customization XMLs**

### **Customization Object XML**

SDF works by representing a customization record in NetSuite with an XML file. This works both ways – extracting an existing customization from an account and creating an XML, and pushing the XML into a target account, creating the customization there. The below example shows the XML object for a User Event script. It contains references to all the fields on that customization record within NetSuite. This includes the scriptID, description, what functions it executes on, and whether or not it is active.

Node	Content
✓ e usereventscript	
③ scriptid	customscript1
e aftersubmitfunction	saveReview
e beforeloadfunction	
e beforesubmitfunction	
e description	
e isinactive	F
e name	Link Product Review to Inventory Item
e notifyadmins	F
e notifyemails	
e notifyowner	Т
e notifyuser	F
e scriptfile	[/SuiteScripts/linkReviewToltem_v3.js]
✓ e scriptdeployments	
> e scriptdeployment	

# Additionally, it also details script deployments at the parameters it uses and what their values are set to.

<ul> <li>e scriptdeployments</li> </ul>	
✓ e scriptdeployment	
③ scriptid	customdeploy1
e allemployees	F
e allpartners	F
e allroles	Т
e audsictrole	
e eventtype	
e executeinwebstore	Т
e isdeployed	Т
e loglevel	DEBUG
e recordtype	[scriptid=customrecord_product_review]
e runasrole	ADMINISTRATOR
e status	RELEASED

### Manifest XML

```
1 <= <manifest projecttype="ACCOUNTCUSTOMIZATION">
 2 <projectname>Test Project</projectname>
  3
     <frameworkversion>1.0</frameworkversion>
  4 <dependencies>
  50
       <features>
         <feature required="true">SERVERSIDESCRIPTING</feature>
  6
  7
         <feature required="false">CREATESUITEBUNDLES</feature>
  8
          <feature required="false">CRM</feature>
  9
       </features>
 100
       <files>
 11
          <file>/SuiteScripts/script core lib.js</file>
 12
        </files>
 13
     </dependencies>
14 </manifest>
```

The manifest for the project contains the requirements for the manifest to be deployed in an environment. For example, if you want to use SDF to deploy SuiteScripts to the target account, then the target account is required to have Server Side Scripting enabled.

One of the greatest strengths of SDF compared to SuiteBundling is allowing for the deployment of objects with dependencies without having to include those dependencies. This is assuming they may have already been deployed via another SuiteBundle. In practice, workflows are some of the most difficult objects to move with a SuiteBundle. This is because workflows often reference multiple customizations and with a SuiteBundle, each related customization will be included automatically. With each migration to an environment, there is a possibility of unintentionally changing the customizations in the target environment with the bundled customizations.

### **SDF Development Environment Setup**

### **IDE Integrations**

SDF officially integrates with Eclipse and WebStorm, with Eclipse being the more mature of the two options. In the following sections, we will be using Eclipse for demonstration purposes. Below you will find the detailed instructions to setup Eclipse with SDF. You can also follow NetSuite provided instructions (See SuiteAnswers ID #10315 for Setting Up SuiteCloud IDE Plug-in for Eclipse).

### Install Eclipse

To get started with SDF, you will have to download Eclipse first. NetSuite recommends using Eclipse Mars. (02/26/2016 package. Link: <u>https://www.eclipse.org/downloads/packages/release/mars/2/eclipse-ide-java-ee-developers</u>.) Although, we haven't experienced any issues with the latest version of Eclipse IDE. J2EE or Javascript/Web Developer version both works. Link: <u>https://www.eclipse.org/downloads/packages/</u>. Once installed, you can change the workspace to a different path. This will make access to the folder easier.

Eclipse IDE	Launcher		×
Select a dire	ctory as workspace		
Eclipse IDE u	ses the workspace directory to store its preferences and development artifacts.		
Workspace:	D:\SDF Demo	Browse	
Use this a	s the default and do not ask again		
Recent Wo	orkspaces		
	Launch	Cancel	

### **Install SDF Plugin**

#### 1. Run Eclipse, Under Window tab, find Preferences



2. In the opened window go to Install/Update > Available Software Site

Preferences		0	
type filter text	Available Software Sites		← → ⇒ → →
<ul> <li>General</li> <li>Help</li> <li>Install/Update <ul> <li>Automatic Updates</li> <li>Available Software Sites</li> </ul> </li> <li>Language Servers</li> <li>Mylyn</li> <li>Oomph</li> <li>Remote Systems</li> <li>Run/Debug</li> <li>Server</li> <li>Team</li> <li>Terminal</li> <li>TextMate</li> <li>YAML</li> </ul>	Available Software Sites         type filter text         Name         Image: Imag	Locatic http://o http://o	Add Edit Remove Reload Enable Import
	<	>	
? 🎽 🖉 🔘	Ap	ply and Close	Cancel

- 3. Click Add and enter the following information
  - Name: SuiteCloud IDE
  - Location: <u>https://system.NetSuite.com/download/suitecloud-sdk/ideplugin/</u> <u>eclipse/19.1</u>(This address changes with releases, so if you are using a builder later than 2019.1 you might need to lookup the address)

🖨 Add Si	te	×
Name:	SuiteCloud IDE	Local
Location:	te.com/download/suitecloud-sdk/ideplugin/eclipse/19.1	Archive
?	Add	Cancel

### 4. Apply and Close

### 5. Go to Help tab, click Install New Software

#### SDF - Eclipse IDE



6. In the opened window, select SuiteCloud under Work with field and verify the address. Check the SuiteCloud IDE checkbox in the list and follow the Eclipse wizard to complete installation.

🖨 Install	-	
Available Software		
Check the items that you wish to install.		()
Work with: SuiteCloud IDE - https://system.NetSuite.com/dow	nload/suitecloud-sdk/ideplugin/eclipse/19.1 v Add	Manage
type filter text		Select All
Name > M IOU SuiteCloud IDE	Version	Deselect All
1 item selected		
Details		
		A.
Show only the latest versions of available software	Hide items that are already installed	
Group items by category	What is <u>already installed</u> ?	
Show only software applicable to target environment		
Contact all update sites during install to find required software	e	
0	< Back Next > Finish	Cancel

- 7. When prompted, restart Eclipse
- 8. If installation is successful, user will see the following dialog after restart

SuiteCloud IDE Setup Wizard						$\times$
SuiteCloud IDE Setup Wizard						
Welcome to the SuiteCloud IDE setup wizard.						
This setup wizard walks you through the creation of y information.	our master passw	ord and addition of	your NetSuite	accoun	t	
Important: You need Internet access in order to comp	lete this setup.					
Click "Next" to continue or "Cancel" to defer setup. If menu.	you choose to de	fer, you can run the	e setup wizard I	ater fro	m the N	etSuite
(?)	< Back	Next >	Finish		Cancel	

 Setup master password for SuiteCloud IDE. Once the password is set up, user will have to use the master password to enable SuiteCloud plugin functions for all SuiteCloud projects

### Enable SDF and Create Custom Role for SDF in NetSuite

Because NetSuite doesn't come with the SDF feature enabled by default, an Administrator will have to manually enable the feature. A new custom role should be created for SDF since the Administrator role and Developer role will encounter an issue with 2FA when using SDF. The new SDF role should be created from the Developer role for SDF purposes.

 Login to NetSuite using Administrator role and go to Setup > Company > Enable Features



#### 2. In the new page, click on SuiteCloud tab



3. Scroll all the way to the bottom and check SDF option, agree to the terms and save the change



4. Once the change is saved, go to Setup > Users/Roles > Manage Roles

			(	2			<b>\$</b>	<b>?</b> H
cuments	Setup	Supp	ort	Suite Social	Fixed Assets	Dem	io Frame	work
	Setup N	/lanage	r					
	Compa	ny	Þ					
	Accoun	ting	Þ					
	Sales		Þ					
	Marketi	ng	×	M/STANDARD				CENTER
	Suppor	t		n				Classic
	ouppor			n				Executi
	Issues		Þ	n				Engine
	Cito Du	ildor		n				Sales C
	Sile Du	liuei		n				Sales C
	Import/I	Export	•	n				Shippin
				-			1	Shippin
	Users/F	Roles		USER MANA	GEMENT			Shippin
	Integrat	tion		Manage Us	ers			Shippin
	····- 3·					-		Shippin
			Custo	r Manage Ro	bles		New	Shippin
			Custo	Show Dolo Diff	lo Difforoncos		Search	Cost Ac
			Custo	or Show Role	Dillefences		Search	Accoun
			Custo	View Login	Audit Trail			Accoun
			Custo	or				Accoun
			Stand	Access Tok	ens	•		Accoun
			Stand	TWO-FACTO	RAUTHENTICATI	ON		Accoun
			Stand	le				Accoun
			Stand	a Two-Factor	Authentication F	Roles		Accoun
			Custo	Two Eactor	Deset Tool			Advanc
			Stand	a iwo-racioi	Reset 1001			Advanc
			Ctone	land				Accourt

5. User should now see the Developer role. (This role won't be available until user enables SDF feature.) Click Customize

Edt	1039	Customer Service Rep	Custom	Support Center
Edt	1028	Customer Service Rep - Sales	Custom	Sales Center
Customize	55	Developer	Standard	Classic Center
Edt	1047	Ecommerce Manager	Custom	E-Commerce Management Center
6.0	1005	Freemanne Manager Ald	Australia (Contralia)	F Commence Management Comme

6. In the role creation page, update the name to SDF Role. Set Access Subsidiaries to All.

Role
Save Cancel Reset Change ID
General
NAME * SDF Role
ID
CENTER TYPE Classic Center
EMPLOYEE RESTRICTIONS
none - no default   ALLOW VIEWING
ISSUE ROLE
Subsidiary Restrictions
ACCESSIBLE SUBSIDIARIES     ALL  ACTIVE  USER SUBSIDIARY  SELECTED  ALLOW CROSS-SUBSIDIARY RECORD VIEWING

### 7. Add Find Transaction under Permissions > Transactions

Permissions	Restrictions	<u>F</u> orms	<u>S</u> earc	hes	<u>U</u> sers	Pr <u>e</u> ferences	<u>D</u> ashboard	<u>T</u> ranslation	<u>H</u> istory
Transactions •	Reports •	Lists •	Setup •	<u>C</u> ust	tom Record				
PERMISSION*									LEVEL
Find Transaction									Full
		•							
✓ Add	× Cancel	+ Insert	Re	move					

8. Add Publish Search under Permissions > Lists (If there is a custom search needs to be set as public for custom objects to use, this permission is required.)

Permissions	<u>R</u> estrictions	<u>F</u> orms	Searches	Pr <u>e</u> ference	s <u>D</u> ashboard	<u>T</u> ranslation	
Transactions	Reports •	Lists • Se	t <u>u</u> p• <u>C</u> us	tom Record			
PERMISSION*							LEVEL
Custom Record	Entries						Full
Documents and	Files						Full
Email Template							Full
Marketing Temp	late						Full
PDF Template							Full
Perform Searc	h	•					Full
✓ ОК	× Cancel	+ Insert	TRemove	e			
Publish Search							Full
Website (Externa	al) publisher						Full
+ Add Row							

9. Remove Integration Applications from Permissions > Lists (Not needed for 2019.2)

Permissions Bestrictions Forms Searches Preferences Dashboard Translation	
Transactions Reports Lists Setup Qustom Record	
PERMISSION *	LEVEL
Custom Record Entries	Full
Documents and Files	Full
Email Template	Full
Entegration Applications 👻	Full
✓ OK × Cancel + Insert TRemove	
Marketing Template	Full
PDF Template	Full
Perform Search	Full
Website (External) publisher	Full
+ Add Row	

 Remove Access Token Management from Permissions > Setup (This option is only available when the Token-Based Authentication feature is enabled) (Not needed for 2019.2)



11. Remove Integration Applications from Permissions > Setup (Not needed for 2019.2)



12. Save the Customized role.

### **Configure Accounts for SDF Project in Eclipse**

### **Change Project Settings**

The 'Change Project Settings' menu option shown in the figure below is accessible from the SDF Project in Eclipse and allows the user to select which account and role combinations SDF will point to by default.

✓ 2 T > 2 > 2 > 2 > 2 > 2	Fest Proj Java → .setti → Accc → FileC → Obje		New Go Into Open in New Window Show In	> Alt+Shift+W >	Node Vermanifest (a) projecttype Projectname Projectname Projectname
	X .proj X deple		Copy Copy Qualified Name	Ctrl+C	
	A man	ß	Paste	Ctrl+V	
		×	Delete	Delete	
		<u>&amp;</u>	Remove from Context Refactor	Ctrl+Alt+Shift+Down	
		è	Import		
		2	Export		
		G.	Refresh Close Project Close Unrelated Projects	F5	
			Assign Working Sets		
		П.	NetSuite	>	Change Project Settings

Right-click the parent project folder and navigate to NetSuite > Change Project Settings.

Note: The Ample Project Settings				>
Account:			~	
Role:			$\sim$	
SuiteScript Version:	2.0		~	
Action on Account-Specific Values:	ERROR		~	
		ОК	Cancel	

### To add environments at roles to a project, click on the ellipsis in the top-right.

Nanage Accounts				- 🗆 X
Account ID	Company Name	Domain	Email	Add
TSTDRV1680326	CleanSweep Demo 3	Production	nick.klug@techfino.com	Remove
				Remove All
				Manage Authentication
			Select and Close	Close

### Select "Add" at the top right.

Add Acc	ount(s)				<u></u>		×
Add Acco	unt(s)						
Email must	be specified.						
Domain:	Production						~
Email:							
Password:							
?		< Back	Next >	Finish		Cance	el 🛛

Now, select which environments you wish to add account logins for. With the reconfiguration of NetSuite's account specific logins, these will all be under "Production" or "Release Preview." Enter your credentials and click "Next."

Account ID	Company Name	^
TSTDRV1610419	CleanSweep Archive Demo	
TSTDRV1680326	CleanSweep Demo 3	
TSTDRV1697099	CleanSweep QA 4 (Leading)	~
Select All	Deselect All	

You will then see a 'Manage Accounts' window allowing you to check all the accounts you wish to use your SDF project with.

Account ID	Company Name	Domain	Email	Add
ISTDRV1610419	CleanSweep Archive Demo	Production	nick.klug@techfino.com	
STDRV1680326	CleanSweep Demo 3	Production	nick.klug@techfino.com	Remove
				Remove All
				Manage Authenticatio

Press "Select and Close" to select one of the environments as a default or "Close" to leave the dialog window.

N Change Project Settings			$\times$
Account:	TSTDRV1680326 CleanSweep Demo 3	~	
Role:	Administrator (User Credentials)	~	
SuiteScript Version:	2.0	~	
Action on Account-Specific Values:	ERROR	~	
	OK	Cancel	

The 'Change Project Settings' window shown in the figure above allows the user to select which account and role combination SDF will default to for its operations. This process can be repeated at any point and it is recommended to start with it set to your Source environment as you develop, then switching to your target environment as you validate and deploy.

### **Add Customizations to SDF**

### Import by ScriptID

In this section, the flexibility to pull pre-existing customizations into your SDF project from a target NetSuite Account is on full display.



# Right-click on the Objects folder in Eclipse and select "Import Custom Objects from Account."

🔘 Import C	ustom Objects from Account			_		×
Import Cu	stom Objects from Account					
Select the a	ccount that contains the custom objects	s you want to imp	oort.			
Account:	TSTDRV1610419 CleanSweep Archive D	emo				~
Role:	Administrator (User Credentials)					~
Search a	nd Select Objects					
?		< Back	Next >	Finish	Can	cel
-						

### Select the account and role needed to access the object, then click "Next"

bject Types:	^ _	Add/Remov
cript Id Contains: customscript		
	Search	Reset
arch Results		
Select All Deselect All	Total Custom Ob	iects Found: 22
		,
Object Type	Script Id	
bundleinstallationscript	customscript_2663_bundle_in	
bundleinstallationscript	customscript_bi_ss	
bundleinstallationscript	customscript_ep_license_upd	
bundleinstallationscript	customscript_ncfar_bundlein	
	customscript netsuite rm bu	
bundleinstallationscript	customsenpt_nebuite_nn_built	
<ul> <li>bundleinstallationscript</li> <li>bundleinstallationscript</li> </ul>	customscript_suitesocial_bun	
bundleinstallationscript         bundleinstallationscript         clientscript	customscript_interante_int_built customscript_suitesocial_bun customscript579	
bundleinstallationscript         bundleinstallationscript         clientscript         clientscript	customscript_interante_int_oum customscript579 customscript_2663_bank_deta	
bundleinstallationscript         bundleinstallationscript         clientscript         clientscript         clientscript	customscript_interante_int_built customscript579 customscript_2663_bank_deta customscript_2663_entity_ba	
bundleinstallationscript         bundleinstallationscript         clientscript         clientscript         clientscript         clientscript         clientscript         clientscript	customscript_interante_int_oum customscriptS79 customscript_2663_bank_deta customscript_2663_entity_ba customscript_2663_field_valid	

Enter the script ID in the above pane and then click "search." After the search finds all matches for the Script ID, check the checkboxes to select the objects you want to import into the project. Take careful note of the "Select All" and "Deselect All" buttons. Click "Finish" to begin importing the selected objects.



The objects will begin the import process. If there are any errors during the import process, the console will display error messages detailing why the import failed. The majority of the issues that occur during the import process are due to permissions. Ensure the role you are using to import has the proper permissions for the objects being imported.

### Import by Bundle ID

Another nice feature is the ability to import objects into your project by SuiteBundle ID. This can be accomplished by right-clicking on the parent project name in Eclipse and select "Import Bundle Components."

≥	Go Into		<ul> <li>✓ e manifest</li> <li>(a) projecttype</li> </ul>
≥ ≥ ≥	Open in New Window Show In	Alt+Shift+W >	e projectname e frameworkversion
	Copy Copy Qualified Name	Ctrl+C	
	Paste	Ctrl+V	
×	Delete	Delete	<u></u>
<u>.</u>	Remove from Context Refactor	Ctrl+Alt+Shift+Down	
	Import Export		
S.	Refresh Close Project Close Unrelated Projects Assign Working Sets	F5	
7.	NetSuite	>	Change Project Settings
	Validate Show in Remote Systems view		Deploy Deploy to Account
	Run As	>	Validate Project against Account
	Debug As	>	Add Dependency References to Manifest
	Profile As	>	Import Bundle Components

Another nice feature is the ability to import objects into your project by SuiteBundle ID. This can be accomplished by right-clicking on the parent project name in Eclipse and select "Import Bundle Components."

Import B	undle Components	
Account:	TSTDRV1680326 CleanSweep Demo 3	\ \
Role	Administrator (User Credentials)	
nore.		
Bundles:		V List Bundles
Bundles:	233457:Test Harness	∼ List Bundle
Bundles:	233457:Test Harness 241868:Techfino Mass Deletion Script	✓ List Bundle

Next, select "List Bundles", select the bundle you wish to import into SDF, then click OK. SDF will automatically import the objects in the bundle into your SDF project.

### Manually Create Customizations

Copy Coalified Name   Copy Coalified Name   Copy Coalified Name   Prest   Copy Coalified Name   Copy Coalified Name   Copy Coalified Name   Copy Coalified Name   Copy Coalifie	v 🖶 Ob	iects		e trameworkversion			1.0
Go line   Open in New Window   Show in   Alt-Shift-W >   Copy   Ctri-C   Copy   Ctri-C   Copy Cualified Name   Part and the shift-Down   Refractor   Show in Remote Systems view   Show in Remote Systems view   Refractor   Show in Remote Systems view   Show in Remote Systems view   Societta As   Overwrife Cutaton Objects wi	. 0.2	Mere	>				
Open in New Window   Show in   Atta-Shift+W>   Copy   Copy Cutified Name   Pate   Pate   Cutified Name   Pate   Cutified Name   Pate   P	X	Go Into					
Show in       Alt-Shift-W3         Copy       Ctri-C         Copy Cualified Name       Ctri-C         Pate       Ctri-V         Pate       Delete         Pate       Delete         Pate       Ctri-Alt-Shift-Down         Refered       Delete         Pate       Ctri-Alt-Shift-Down         Refered       Delete         Validate       Deploy         Show in Remote Systems view       Refered         Num As       Deploy to Account         Validate       Deploy to Account         Source	X	Open in New Window					
Cepy Ctrl-C   Cepy Coalfied Name   Paste Ctrl-V   Delete    Delete   Delete		Show In	Alt+Shift+W >				
Copy Qualified Name   Paste   Copy Qualified Name   Remove from Context   Citi-Alt-Shift-Down   Refector   Import   Import   Refector   Refector   National Spatems view   Robitster   Show in Remote Systems view   Robitster   Profile As   Profile As   Compare With   Source   Properties   Alt-Enter     Debug As   Profile As   Compare With   Source   Properties     Alt-Enter     Debug As   Profile As   Compare With   Source   Properties     Alt-Enter     Debug As   Profile As   Compare With   Compare With   Compare With   Properties     Alt-Enter     The following object (a) were import   Context Age Searches   <	100	Com	04.0				
Copy Case   Pathe   Curley   Renove from Contest   Ctrl+Alt-Shift-Down   Refeach   Delate   Delate   Refeach   Export   Report   Report   National Server         Validate   Delay ds   Delay ds   Delay ds   Delay ds   Compare With   Retore from Cocal History   Compare With   Retore from Cocal History   Derige tiss   Alt-Enter         Device Alt-Enter           Device Alt-Enter        Device Custom Object (s) were importance of the Matter Project as Zip File   Log in to Project Account.   Network Custom Object (s) were importance of the Matter Project as Zip File   Log in to Project Account   The following object (s) were importance of the Matter Project as Zip File   Log in to Project Account   Record Type   Import/Eport   Compare With   Restore from Local History   Delay ds   Delay in the Field   Compare With   Restore from Local History   Delay in the Field   Restore from Local History   Delay in the Field   Contern Hidd   The following object (s) were importance   Delay in the Field   Delay in the Field	100	Copy Qualified Name	cure				
Veloce       Delete         Veloce       Delete         Remove from Contest       Ctrl+Alt-Shift-Down         Refactor       Similar         Refactor       Similar         Refactor       Similar         Validate       Deloy         Validate       Deloy         Validate       Deloy         Validate       Deloy to Account         Validate       Deloy to Account         Validate       Deloy to Account         Validate       Deloy to Account         Validate       New Custom Objects from Account         Add Dependency References to Manifest       Impact Custom Objects from Account         New Custom Objects from Account.       Scripting         New Custom Objects from Account.       Scripting         New Custom Objects in Account.       Scripting         New Custom Objects in Account.       Centers and Tabs         New Custom Object (s)       Nexe import Export         Device Master Passoord       Conserver Lipst Account         Record Type       Impact/Export         The following object (s)       West Properties         Alt-Enter       The following object (s)       West Properties         The following object (s)       Wes		Parte	CirleV				
Remove from Context       Ctrl+Alt+Shift+Down         Refactor		Delete	Delete				
Remove from Context       CthPAR+Shift+Down         Import       Import         Import       Export         Validate       Deploy         Show in Remote Systems view       Run As         Projetile As       Deploy to Account         Add Dependency References to Manfest       Import Custom Objects from Account         New Custom Objects from Account       Exits, Records, & Fields         New Custom Objects in Account       Scripting         New Custom Objects with Account Version       Scripting         Source       Alt-Enter         The following object (s) were import       CMS Content Type         The following object (s) were import       CMS Content Type         Chyperties       Alt-Enter			0.1.15.01% D				
Referent   Show in Remote Systems view   Run As   Debug As   Profile As   Team   Overwrite Custom Objects from Account   New Custom Objects from Account   New Custom Objects with Account   Source   Properties   Alt-Enter     New Kaster Passward   Destor Systems of Projecties     Alt-Enter     Destor Spread     Destor Spread     Destor Spread     Market Projecties     Alt-Enter     Destor Spread     Destor Spread     Properties     Alt-Enter     Destor Spread     Destor Spread <t< td=""><td>0</td><td>Remove from Context</td><td>Ctrl+Alt+Shift+Down</td><td></td><td></td><td></td><td></td></t<>	0	Remove from Context	Ctrl+Alt+Shift+Down				
Imparf       Export		Refactor	,				
Image: Sport       FS         Image: Sport       FS         Assign Working Sets       Deploy         Image: Sport       Deploy         Validate       Deploy to Account         Validate Project against Account       Add Dependency References to Manifest         Import Custom Objects from Account       New Custom Objects from Account         New Custom Objects from Account       Scripting       Record Type         Compare With       Overwrite Custom Object account.       Scripting       Instantion Type         Source       Overwrite Custom Object Account       Source       Entity Field         New Custom Object (s) were import       User/Roles       CRM Field       Transaction Type         Content Type       Deploy to Account       Deploy to Account       The following object (s) were import       CRM Field       Transaction Odyrin Field         New Custom Soript1       Decuments       The following object (s) were import       Transaction Column Field       Transaction Column Field         New Custom Soript1       Decuments       The following object (s) wer	<u>ile</u>	Import					
Properties       Att-Enter         Properties       Att-Enter	2	Export					
Assign Working Sets       Deploy         Validate       Deploy to Account         Show in Remote Systems view       Validate Project against Account         Run As       Add Dependency References to Manifest         Debug As       Import Custom Objects from Account         Profile As       New Castom Objects from Account         Compare With       Overwrife Custom Objects with Account         Source       Source         Properties       Alt-Enter         The following object (s) were import       CMS Content Type         Conspare Vith       CMS Content Type         Revice Master Password       Saved Searches         The following object (s) were import       CMS Content Type         Users/Roles       Transaction Rype         Transaction Rype       Transaction Rype         Content Type       CMS Content Type         Outcoments       Transaction Column Field         Revice Master Password       CMS Content Type         The following object (s) were import       CMS Column Field         Transaction Rype       Transaction Rype         Goustomsoript1       CMS Content Type         Outcomsoript1       CMS Content Type         Custom Field       Transaction Rype	- 30	Refresh	F5				
NetSuite       Deploy         Validate       Deploy to Account         Show in Remote Systems view       Add Dependency References to Manifest         Debug As       Add Dependency References to Manifest         Import Custom Objects from Account       Add Dependency References to Manifest         Team       Overwrite Custom Objects from Account         New Custom Objects an Account       Scripting         Source       Overwrite Custom Object account.         Brook Matter Password       Entdy Field         Properties       Alt-Enter         Team continue of progeness of the following object (s) were import       Centers and Tabs         Properties       Alt-Enter		Assign Working Sets					
Validate     Deploy to Account     Validate Project against Account     Validate Project against Account       Show in Remote Systems view     Add Dependency References to Manifest     Insect Custom Objects from Account     Validate Project against Account       Profile As     Import Custom Objects from Account     Validate Project against Account     Validate Project against Account       New Custom Objects from Account     New Custom Objects with Account Version     Scripting     Lists, Records, & Fields       Compare With     Source     Source     Entity Field       Properties     Allt-Enter     Revoke Mater Password     Forms     Intern Field       The following object (s) were import     CMS Content Type     CMS Content Type       Userom Scient     Yeard Custom Field     Transaction Rody Field       Transaction Rody Field     Transaction Rody Field     Transaction Rody Field	12	NetSuite	>	Deploy			
Show in Remote Systems view       Validate Project against Account       Add Dependency References to Manifest         Run As       Impart Custom Objects from Account       Impart Custom Objects from Account       Itists, Records, & Fields       Ist         Profile As       Impart Custom Objects with Account Version       Scripting       Ist       Record Type         Compare With       Saurce       Saurce       Item Field       Item Field         Properties       Alt-Enter       The following object (s) were import       Saurce import (Edition Custom Cu		Validate		Deploy to Account			
Run As       Add Dependency References to Manifest         Debug As       Import Custom Objects from Account       >         Profile As       New Custom Objects from Account       >         Team       >       Overwrite Custom Objects with Account Version       >       Cscripting       >       Record Type         Compare With       >       Saurce       >       Entity Field       >       Entity Field         Source       >       New Custom Object (s) were important of the following object (s) were imp		Show in Remote Systems view		Validate Project against Account			
Debug As       Import Custom Objects from Account       Issts, Records, & Frields       Issts, Records, & Frields       Issts, Record Type         Team       Overwrite Custom Objects with Account Version Save Project as Zip File       Scripting       Record Type         Compare With       Save Project as Zip File       Debug As       Entity Frield       Transaction Type         Source       Compare With       Revice Master Password       Entity Frield       Entity Frield         Properties       Alt=Enter       The following object (s) were import Outpents       CAM Field       Transaction Type         Transaction Review       The following object (s) were import Outpents       Transaction Column Field       Transaction Review         Becomments       Transaction Review       Transaction Column Field       Transaction Review       Transaction Review		Run As	>	Add Dependency References to Manifest			
Profile As       New Custom Object in Account       Itsts, Records, & Fields       List         Team       Overwrite Custom Objects with Account Version       Scripting       Record Type         Compare With       Save Project as Zip File       Dog into Project Account       Properties       Entity Field       Entity Field         Source       Revoke Matter Password       Revoke Matter Password       Save Searches       CMS Content Type         The following object(s) were impo       User/Roles       CMS Content Type         User/Roles       Transaction Rody Field       Transaction Rody Field         Documents       Transaction Rody Field       Transaction Rody Field		Debug As	>	Import Custom Objects from Account		1	
Team       Overwrite Custom Objects with Account Version       Scripting       >       Record Type         Compare With       Save Project as Zip File       Duestion       Plug-ins       >       Transaction Type         Source       Compare With       Revoke Master Password       Centers and Tabs       >       Entity Field         Properties       Alt=Enter       The following object (s) were import custom object (s) were import for the Number of Users/Roles       >       CMM Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Users/Roles       >       Transaction Rody Field       >       Transaction Rody Field         Custom Segment       Other Custom Field       Custom Segment       Other Custom Field		Profile As	>	New Custom Object in Account >	Lists, Records, & Fields	> List	
Compare With       Save Project as Zip File       Plug-ins       Transaction Type         Restore from Local History       Log in to Project Account       Centers and Tabs       Entity Field         Source       Revoke Master Password       Saved Searches       CRM Field       CRM Field         Properties       Alt-Enter       The following object (s) were impo- customsoript1       Import/Zuport       CRM Field       CRM Field         Documents       Transaction Type       Import/Zuport       CRM Field       Transaction Body Field         Documents       Transaction Type       Import/Zuport       Transaction Type         Users/Roles       Transaction Type       Transaction Revolution Field         Octorent Type       Entity Field       Transaction Revolution Field         Documents       Transaction Revolution Field       Transaction Revolution Field		Team	>	Overwrite Custom Objects with Account Version	Scripting	Record Typ	e
Restore from Local History       Log in to Project Account       Centers and Tabs       Entity Field         Source       Revoke Master Pastword       Forms       Item Field         Properties       All+Enter       The following object (s) were impo- customeoript1       Source       CRM Field         The following object (s) were impo- customeoript1       CRM Field       Transaction Body Field       Transaction Column Field         Transaction Item Option Item Number Field       Column Field       Column Field       Column Field		Compare With	>	Save Project as Zip File	Plug-ins	> Transaction	Туре
Source     Forms     Item Field       Properties     Alt-Enter     The following object(s) were import oustomsoripti     Saved Searches     CMS Content Type       Users/Roles     CMField     Transaction Rody Field     Transaction Rody Field       Documents     Transaction Red Option Item Number Field     Transaction Red Option Item Number Field		Restore from Local History		Log in to Project Account	Centers and Tabs	> Entity Field	
Properties     Alt+Enter     Incomposition     Saved Searches     CMS Content Type       The following object(s) were impo- customscript1     Import/Export     CRM Field     CRM Field       Documents     Transaction Body Field     Documents     Transaction Column Field       Custom Segment     Other Custom Field     Custom Segment		Source	>	Revoke Master Password	Forms	> Item Field	
The following object(s) were import import/Export CRM Field Gustomscript1 CRM Field Documents Transaction Sody Field Transaction Column Field Transaction Item Option Item Number Field Custom Segment Other Custom Field		Properties	Alt+Enter	itariojet	Saved Searches	CMS Conte	ent Type
Guiscumscripti     Users/Roles     >     Transaction Body Field       Documents     >     Transaction Column Field       Transaction Item Option Item Number Field     Custom Segment       Other Custom Field     Other Custom Field	_			The following object(s) were impo:	Import/Export	CRM Field	
Documents Transaction Column Field Transaction tern Option Item Number Field Custom Segment Other Custom Field				customscripti	Users/Roles	> Transaction	Body Field
Transaction Item Option Item Number Field Custom Segment Other Custom Field					Documents	> Transaction	Column Field
Item Number Field Custom Segment Other Custom Field						Transaction	Item Option
Custom Segment Other Custom Field						Item Numi	ver Field
Other Custom Field						Custom Se	gment
						Other Cust	om Field

Right-click on the Objects folder in the SDF project. Navigate to NetSuite > New Object In Account, then select the customization object you want to add.

### **Adding Dependencies in SDF**

With SuiteBundler, you often times run into deployment issues using SuiteBundles to deploy Customizations especially when they automatically include referenced objects from other SuiteBundles. This leads to some SuiteBundles including fields or records from other SuiteApps or SuiteBundles unintentionally because they were referenced in a Workflow for example. But this is one of the areas where SDF provides greater control and flexibility to either automatically include referenced objects.



Updating the manifest can be done two ways. The simplest method is to right-click anywhere in the project, navigate to NetSuite > Add Dependency References to Manifest. This will add all required dependencies for the objects currently in the project and add them to the project.

Alternatively, you can manually add dependencies to the project for the objects you know you will be working with as you go.

For the first few SDF projects, we highly recommend using the automatic addition of dependencies and paring back as you discover objects that you need to be included for the project to deploy.

### Validation and Deploying SDF Project

### Validate Project Against Account

Before deploying your SDF Project to a target account you can first validate the Project has all the prerequisites for installation. To do so, right-click anywhere in the project, navigate to NetSuite > Validate Project Against Account.

<u>- 1 (0) E</u> (1 – 1	* • 0 • • • 🖉 🖉 🖉 😫 😭	[월 • 祠 • ♥ � • ↔ •
🗄 NS Explorer 🔀 🚼 Outline		fest.xml 🔀 customscript_mr_mass_delete.xml 🔀
<ul> <li>SDF-Demo [sdf-demo mastella JavaScript Resources</li> <li>JavaScript Resources</li> <li>AccountConfiguration</li> <li>PileCabinet</li> <li>Pil</li></ul>	New     Go Into       Open in New Window       Show In       Copy       Copy Qualified Name       Paste       Pete       Remove from Context     Ctrl+Alt+ Refactor       Import       Export       Refresh       Close Project       Assign Working Sets	<pre>anneducescript scriptid="customscript_mr_mass_delete")</pre>
	NetSuite	Change Project Settings
	<ul> <li>Run As</li> <li>Debug As</li> <li>Profile As</li> <li>Restore from Local History</li> <li>Team</li> <li>Compare With</li> <li>Replace With</li> <li>Configure</li> <li>Source</li> <li>Validate</li> </ul>	Deploy     Deploy     Deploy to Account     Validate Project against Account     Add Dependency References to Manifest     Import Bundle Components     Overwrite Custom Objects with Account Version     Save Project as Zip File     Log in to Project Account     Revoke Master Password
	Properties	Alt+Enter

Nalidate Project Against Account			×
Account:	TSTDRV1610419 CleanSweep Archive Demo		~
Role:	Administrator (User Credentials)		~
Action on Account-Specific Values:	ERROR		~
	Valid	late	Cancel

Next, select the account and role that you wish to use in order to validate the SDF project. Click validate and the process will begin.

Test Project Validate account settings Failed

Validation of account settings failed.

An error occurred during account settings validation. Details: The manifest contains a dependency on customrecord\_product\_review object, but it is not in the account. Details: The manifest contains a dependency on /SuiteScripts/linkReviewToItem v3.js file, but it is not in the account.

The validation process will check against the account to see if the project contains the settings and objects necessary to support the deployment - for example, if your SDF project requires Server Side Scripting in the manifest, and the target account does not have it, it will throw the related error. If the SDF project contains a reference in the manifest to customrecord\_product\_review, then customrecord\_product\_review needs to already be present in the target account. If it is not present, you will receive a warning similar to the above.

### **Deploy SDF Project to Account**

After successfully validating your project against the target account, you are then set to perform the deployment. To initiate the deployment start by right-clicking anywhere in the project and navigate to NetSuite > Deploy to Account.



Next, select the account and role for project deployment. Click Deploy, and the process will begin. If the target Account is a Production environment, then there will be an additional confirmation window as shown below:

N Deploy SuiteCloud Project		×
Account:	TSTDRV1610419 CleanSweep Archive Demo	~
Role:	Administrator (User Credentials)	~
Action on Account-Specific Values:	ERROR	~
	Deploy	Cancel

If the project deployment is not successful, you will see error messages similar to what was contained in the validation section. Sometimes, there are additional issues in an environment that cannot be caught during the validation stage. This typically relates to data conflicts in the target Account.or example, duplicate script IDs, values used in things such as a script parameter or workflows that are not contained in the target Account.

### Sample: Mass Delete Script Deployment

### Background

Previously,we posted **NetSuite Mass Delete Tool**, developed and documented by Luke. Today, we are going to use this solution for a quick example for using SDF as a deployment tool. Below are the steps for what's involved in the customization.

- Mass Delete Search A saved search created to identify the records to be deleted.
  - In this case, we don't actually want to delete any transactions. As a result, we added Date Created after today as a criteria for this search
  - Search columns doesn't really matter. We kept as is and didn't change anything
  - The search must be set to public to be allowed to be selected under script deployment

Saved Transaction Search	List Search Copy to Account (Beta) More Exp	ort
Mass Delete Search		
Save & Run V Reset Cancel Preview New Templat	te Pivot Report Change ID Actions	•
SEARCH TITLE * Mass Delete Search	AVAILABLE A3 DA3HBOARD VIEW	
ID customsearch_mr_delete_search OWNER	AVAILABLE FOR REMINDERS	
<u>Criteria</u> Results Highlighting Available Filters Audience Roles	<u>E</u> mail Au <u>d</u> it Trail Execution Log <u>S</u> earch Title	Tra
Use this tab to specify criteria that narrow down your search.		
Standard • Summary		
FILTER*	DESCRIPTION* FORMULA	
Main Line	is true	
Date Created	is on today	
Туре	is Journal	
Add Cancel + Insert Remove		
Save & Run v Reset Cancel Preview New Templat	te Pivot Report Change ID Actions	
<u>Criteria</u> <b>Results</b> Highlighting <u>A</u> vailable Filters Audience Roles <u>Em</u>	nail Audit Irail Execution L	
Use this tab to indicate columns to be included in the search results as well as sort		

Order Type		Normal	-
citter type	DESCENDING	NUT THE	•
THEN BY		CONSOLIDATED EXCHANGE RATE	
		Per-Account	-
THEN BY		SHOW TOTALS	
	DESCENDING	MAX RESU	LTS
		RUN UNRESTRICTED	
		MY PREFERRED SEARCH RESULT	rs

FIELD*	SUMMARY TYPE	FUNCTION	FORMULA	WHEN ORDERED BY FIELD	CUSTOM LABEL	CUSTOM LABE
Order Type						
ż						
Date						
As Of Date						
Period						
lax Period						
Туре						
Document Numbe	r					
Name						
Account						
Memo						
Amount						

• Utility | Mass Delete (MR) - The Map/Reduce script record

Script				
Edit Back	Deploy Script	Actions -		
TYPE Map/Reduce NAME Utility   Mass Delete (Mi ID customscript_mr_mass_ API VERSION 2.0	R) _delete			
<u>Scripts</u> Parame	ters <u>U</u> nhandled Errors	Execution Log	<u>D</u> eployments	System Notes
SCRIPT FILE preview TF_MR_Mas GET INPUT DATA MAP REDUCE SUMMARIZE	ters <u>Unhandled Errors</u> sDeletionScript2.0.js <u>downloa</u>	Execution Log	<u>D</u> eployments	System Notes
SCRIPT FILE preview TF_MR_Mas GET INPUT DATA MAP REDUCE SUMMARIZE	ters <u>Unhandled Errors</u> sDeletionScript2.0.js <u>downloa</u>	Execution Log	<u>D</u> eployments	System Notes
Scripts Parameter SCRIPT FILE preview TF_MR_Mas GET INPUT DATA MAP REDUCE SUMMARIZE	ters <u>Unhandled Errors</u> sDeletionScript2.0.js <u>downloa</u>	Execution Log	Deployments	System Notes

• Script parameter: Saved Search for Delete - this parameter is passed to the script to identify what is going to be deleted

<u>S</u> cripts <u>P</u> arameter	s Unhandled Errors	Execution Log	<u>D</u> eployments	System Notes
New Parameter				
DESCRIPTION	ID		TYPE	LIST/RECORD
Saved Search for Delet	e custscript_tf_	saved_search	List/Rec	ord Saved Search
Edit Back	Deploy Script	Actions •		

### Script Deployment

• Set Parameter value to Mass Delete Search

Script Deployment	
Edit Back Actions -	
SCRIPT Utility   Mass Delete (MR)	STATUS Not Scheduled
TITLE Utility   Mass Delete (MR)	SEE INSTANCES Status Page
ID customdoploy_mr_mass_doleto	LOG LEVEL Debug
✓ DEPLOYED	EXECUTE AS ROLE Administrator
	PRIORITY Standard
	CONCURRENCY LIMIT 2 SUBMIT ALL STAGES AT ONCE
	YIELD AFTER MINUTES
	BUFFER SIZE
Schedule • Parameters • Execution Log System Notes	
SAVED SEARCH FOR DELETE Mass Delete Search	

We are going to use Eclipse for this example to pull above customization from our CleanSweep Archives & Purge Dev and deploy to QA account. We will show you each step, hope this will be useful to you. We are assuming you already have Eclipse and NetSuite setup according to our SDF Development Environment Setup section which includes custom role setup, assigned to user and SDF plugin for Eclipse. This example is using Eclipse IDE 2019-03 R for Javascript and Web Developers edition.

### Project and Git setup in Eclipse

For demonstration purposes, we are using Eclipse git plugin for all repository interaction because it doesn't require any other installations. You can use your choice of repository and tool (SVN, GitKraken, etc.). An empty git repository sdf-demo has also been created via bitbucket for this example.

### Add Git Perspective and Add Repository

1. Click on Open Perspective located at the top right corner



2. Select Git from the list and Click Open. Now the Git Perspective is opened, you should be able to see Git Repository section (Shown below)



3. Click on Clone a Git repository, enter the URI and required information and click next

nter the locatio	on of the s	ource reposite	ory.		1	5
			1993 N			
Location						
URI:	https:/	// @	bitbucket.org/1	echfino/sdf-demo.	gi Local Fi	ile
Host:	bitbuc	ket.org				
Repository pa	th: /techf	ino/sdf-demo	o.git			
Port: Authenticatio User:	n					
Password:	cure Store					

4. Setup path and click Finish. Please be aware sdf-demo here is the name of the repository instead of SDF-Demo which is the name of the project.

🖨 Clone Git Re	pository					×
Local Destinat	ion				GI	T
Configure the I	ocal storage location for s	df-demo.			-	-4
Destination						
Directory:	D:\git\sdf-demo				Brow	se
Initial branch:	master					~
Clone subn	odules					
Configuration	3					
Remote name	sdf-demo					
Projects						
Working sets	xisting Eclipse projects aft	er clone finish	es			
Add proj	ect to working sets			N	ew	11
Working set	51		~	Se	lect	
?	< Back	Next >	Finish		Cance	I

5. Now the bitbucket repository is added to your repository list



### Create New SuiteCloud Customization Project

1. Click on SuiteCloud perspective at the top right corner



2. Go to File > New > SuiteCloud Project...



3. Select Account Customization and click Next

### 4. Enter Project Name: SDF-Demo and Click Finish

New SuiteCloud Pr	oject			×
reate an Account C	ustomization Project			
Create an Account Cu ocation.	stomization project in the workspace or in an external			
Project Name:	SDF-Demo			
Framework Version:	1.0			
SuiteScript Version:	2.0			~
Location, Drabt De			Drowse.	
?	< Back Next > Finis	n	Cance	el

#### 5. You will see your new SuiteCloud project in NS Explorer



### Link SuiteCloud Project to Repository

#### 1. Right Click on SDF-Demo project and find Team > Share Project



### 2. Select sdf-demo under Repository field and click Finish

<ul> <li>Share Project</li> <li>Configure Git Re</li> </ul>	pository		GIT
Select an existing	repository or create a new one		
Use or create rep	ository in parent folder of project		
Repository:	sdf-demo - D:\git\sdf-demo\.g	jit	✓ Create
Working tree:	D:\git\sdf-demo		
Path within reposit	ory:		Browse
Project	Current Location	Target Location	
🗹 🗁 SDF-De	D:/SDF Demo/SDF-Demo	D:/git/sdf-demo/SDF-Demo	
?		Fin	ish Cancel

3. Now the Project is linked to our sdf-demo repository. If you go to Git perspective and expand sdf-memo repository. You will be able to find all the folders and files there.



### **Commit Empty SuiteCloud Project**

At this point, we can commit and push the empty project to Git. It would help us to identify what are changed in later stage.

 Right click on the SDF-Demo project and go to Team > Commit, the move all files to Staged Changes



- 2. Enter Commit Message and click Commit and Push
- 3. Now the empty SuiteCloud project has been committed and pushed to our bitbucket hosted git repository

#### Setup SuiteCloud Project Access

Now we need to set up the project access.

 Right click SDF-Demo project and go to NetSuite > Authenticate Master Password and enter your master password. This password is the one you entered when you installed the SDF plugin for the first time.



2. Once entered, all NetSuite related functions are enabled.

### 3. Right Click on the project and find NetSuite > Change Project Settings



#### 4. Click Add, enter your NetSuite username and password, then click Next

Nanage Accounts	Add	Add Acco	ount(s)	-		×	— 🗆 X
Account ID	C Imp and	ort your entering	r account information by connecting to your desired NetSuite server g your NetSuite login credentials.				Add
	Dor	main:	Production			~	Remove
	Ema	ail:	renhong.zhu@techfino.com				Remove All
	Pas	sword:	*********				Manage Authentication
	_						-
							-
							-
	?	)	< Back Next > Finish		Canc	el	Close

5. If the credential you entered is correct, you will see a list of account your have access to in the next page. In this case, we are moving customization from Clean-Sweep Archives & Purge Dev to QA. We are selecting those 2 accounts and Click finish.

TSTDRV1	CleanSweep Archive & Purge DEV
TSTDRV1	CleanSweep QA 4 (Leading)

6. Now we need to setup Token access using SDF Role. Otherwise, NetSuite will kick you out of your UI session every time you import a customization object. Click on the first account in the list and click Manage Authentication on the right side. Then click Issue Token. You should see Token Issued messaged after a couple seconds. Do the same for the other account.

Account ID	Company N	Vame		Domain	E	mail		Add
TSTDRV1	CleanSweep	p Archive & Purge DEV		Production	n	enhong.	zhu@techfino	Develop
TSTDRV1	CleanSweep	p QA 4 (Leading)		Production	, P	enhong.	zhu@techfino	Kemove
		Manage Authenti	cation			×		Remove All
		Account ID: TSTDRV Company Name: Cle Domain: Production Email: renhong.zhu(	anSweep Archive	Purge DEV				Manage Authentication
		Role	Authenticatio	n Type	Issue Toke	n		
		Administrator	User Credenti	als	Revoke Tol	ken		
		SDF Role	User Credenti	als	Save Toke	n		
							ind Close	Close

7. Next, select Dev account and Click Select and Close. In the future, when we need to import data, the Dev account will be the default account.

#### 8. On the next page, change role to SDF Role

Change Project Settings			×
Account:	TSTDRV1		·
Role:	SDF Role (Token Based Authentication)		•
SuiteScript Version:	2.0		
Action on Account-Specific Values:	ERROR		-
		OK Canc	el

At this Point, access has been captured in Eclipse. For the next project, if you are using the same accounts, you don't have to do this again.

#### Import Customizations and Validate before Deployment

Finally it's time to import customization from Dev to Eclipse. Keep in mind, the script contains two parts. The script record under Objects folder and the actual script file under File Cabinet > SuiteScripts folder alone with all the libraries.

#### 1. Right click the AccountConfiguration and find NetSuite > Import Configuration



### 2. Click Get Configuration List, select Enable Features then OK

N Import C	onfiguration	$\times$
Account:	TSTDRV1632486 CleanSweep Archive & Purge DEV	~
Role:	SDF Role (Token Based Authentication)	~
		Get Configuration List
Configura	ation	
E	nable Features	
	OK	Cancel

3. Right click the FileCabinet and find NetSuite > Import Files from Account



4. Click Get File List and select the target folder. In this case, we only need files from the Utility folder. Click OK. Then NetSuite will start importing the script files from File Cabinet. Once done, you should see the script files under FileCabinet folder. Also, it's important to pay attention to the logs in the console. Should an error occur, you will see details highlighted in red. You will want to pay attention to logs whenever you import or deploy anything.

Role:	SDF Role (Token Based Authentication)		~
		[	Get File List
File(s) to	download		
V 🗆 🖌	> File Cabinet		^
× [	🔁 SuiteScripts		
>	CreateRecordRestlet		
>	🗌 🔁 Honeycomb Mfg		
>	🗌 🔁 HoneycombMfg		
	🗌 📄 SystemStatus.js		
	TF_MR_MassDeletionScript2.0.js		
>	C > TestHarness		
~	🖂 🗁 Utility		
	🗹 📄 TF_LIB_MassDeletionScript2.0.js		
	🗹 📄 TF_MR_MassDeletionScript2.0.js		
	🔲 📄 WishList_Ramsey.js		
	🔲 📄 linkReviewToltem.js		
	🔲 📄 linkReviewToltem_v3.js		
	🔲 📄 massdelete.js		
	🗌 📄 testPortlet.js		
	🔲 📄 testSS_Processor.js		
>	🔁 Templates		~
Exclude	File and Folder Properties		
	L	ОК	Cancel

Import the Saved Search object by right clicking Objects folder and go to NetSuite
 Import Custom Objects from Account

声 Objects	۱.			e trameworkversion
X .projec		New	>	
X deploy		Open in New Window Show In	Alt+Shift+W >	
		Copy	Ctrl+C	
		Paste Delete	Ctrl+V	
	<b>8</b> _	Remove from Context Refactor	Ctrl+Alt+Shift+Down	
E E		Import Export		
•	Ş	Refresh Assign Working Sets	F5	
	L.	NetSuite	>	Deploy
		Validate Show in Remote Systems view Run As	>	Deploy to Account Validate Project against Account Add Dependency References to Manifest
		Debug As	>	Import Custom Objects from Account

6. Click Next, put the NetSuite id for saved search in Script ID Contains field and click search. Find the check the correct search in the list and click Finish. Once imported, you should see the search object under Objects folder

bject Types:				Add/Remove
cript Id Contai	ns: customsearch	_mr		
			Search	Reset

7. Now we need to import the script object by right clicking Objects folder and go to NetSuite > Import Custom Objects from Account again

#### 8. Enter and Select the correct object in the window and click finish.

Import Custom Object	ts from Account			87 <b>-</b> 3		Х
Search and Import Cus	tom Objects from Acco	ount				
Select or enter search crit	teria to find the custom ob	jects you want to im	iport.			
Search Criteria						
Object Types:					Add/Remo	ve
Script Id Contains: cu	istomscript_mr					
				Search	Reset	
Select All Dr	eselect All Script Id t customs	cript_mr_mass_delet	e	fotal Custom Obje	ects Found	:1
?		< Back	Next >	Finish	Cance	

9. Now everything is imported. Your NS Explorer should look like the below image.



10. Last step for import, we need to update the dependency list. To do this, right click on the manifest.xml file and go to NetSuite > Add Dependency References to Manifest. This will automatically update your manifest.xml to eliminate some of the dependency issue. (In this example, there are not really any dependencies. In a different scenario, you might have to manually update manifest.xml.)



 Before we deploy the customization to targeted QA account, we want to make sure the deployment would be successful and is not going to cause any errors. To do this, right click on the SDF-Demo project and go to NetSuite > Validate Project against Account

	New Go Into	>	
	Open in New Window Show In	Alt+Shift+W >	
	Copy Copy Qualified Name	Ctrl+C	
1	Paste	Ctrl+V	
×	Delete	Delete	
<u>_</u>	Remove from Context Refactor	Ctrl+Alt+Shift+Down	
2	Import Export		
G.	Refresh Assign Working Sets	F5	
	NetSuite	>	Deploy
	Validate Show in Remote Systems view Run As	, [	Deploy to Account Validate Project against Account Add Dependency References to Manifest

12. In the pop up dialog, select QA account and SDF Role. Since we will deploy this customization to QA, we need to ensure it passes validation. Again, you should check the validation message in console.

Nalidate Project Against Account		×
Account:	TSTDRV1697099 CleanSweep QA 4 (Leading)	$\sim$
Role:	SDF Role (Token Based Authentication)	$\sim$
Action on Account-Specific Values:	ERROR	$\sim$
	Validate Cancel	

😰 Problems @ Documentation 🔫 Progress 📮 Console 🔀 🏰 Git Sta	aging					
SDF-Demo						
Validating using account (TSTDRV1697099). Validate manifest Success						
Validate configuration Success						
Validate objects Success						
Validate files Success						
Validate folders Success						
Validate preferences Success						
Validate flags Success						
Validate account settings Success						
Validate Custom Objects against the Account Success						
Validate file cabinet items against the account Success						
Validation COMPLETE						

#### **Deploy to Target Account**

We are finally at the point to deploy this change to our QA account now that the validation is complete.

1. Right click on SDF-Demo project and go to NetSuite > Deploy to Account



2. Change Account to QA and Role to SDF Role and Click Deploy.



3. Validate customization has been successfully deployed to target account from console logs.



4. Log into NetSuite and verify all changes has been deployed

	UITE 🕂 🌗	s	Search		٩		N- @ Help 🗖 Feed	rack . RenHong	Zhu QA 4 (Leading) - Administrator
۵ 🖈 🙆 ۸۰	tivities Box Files Pa	yments Transactions Lists	Reports Analytics Custo	mization Doc	uments Setup Support Demo Fram	ework Fixed Asset	SuiteSocial Sales Knowledge Base		
Scripts New Script									View Deployn
FILTERS									
Map/Reduce 🗑	- All - VERSION	TF_MR_MassDeletionScript2.0.js	TROM	TEUNDLE					
SHOW INACTIVES									TOTA
EDIT   VIEW NAM	E.A.	FROM BUNDLE	10		API VERSION	DEPLOYMENTS	SCRIPT		LIBRARY SCRIPT
Script De	ploymer ack Ad	It otions <del>-</del>							
Utility   Mass Del	ete (MR)			Ì	Not Scheduled				
TITLE Utility   Mass Del	ete (MR)			0	SEE INSTANCES Status Page				
ID customdeploy_m	ır_mass_delete			l	LOG LEVEL Debug				
✓ DEPLOYED				E	EXECUTE AS ROLE Administrator				
				F	PRIORITY Standard				
					CONCURRENCY LIMIT 2 SUBMIT ALL STAGES #	AT ONCE			
				1	YIELD AFTER MINUTES				
				ł	RUFFFR SI7F 1				
Schedule •	Parameters •	Execution Log	System Notes						
SAVED SEARCH Mass Delete Se	H FOR DELETE earch								
THE		EDOM R				TYPE		OWNED	MODERS

customsearch\_mr\_deleto\_search Transaction RenHong Zhu Public

Mass Delete Search

5. Don't forget Commit and push all the changes to Git

That concludes this example for deploying Mass Delete Customization from our Dev account to QA account. We hope that you found this example helpful.

### Additional Notes for This Example

- Some of the steps only apply to Account Customization. The process for SuiteApp will be slightly different for the initial project setup, and you will need to supply your Publisher ID.
- Import AccountConfiguration may not be necessary if the settings are the same between accounts. Using SDF role may cause error due to permission issues. If that's the case, switch to Administrator role for AccountConfiguration for import and export.
- 3. Validate against Account action is more for dependency check. It does not guarantee deployment will be successful.
- 4. Always log into NetSuite and check deployment result after SDF deployment.
- 5. This demo does not follow git best practices and uses only the master branch for demo simplicity
- 6. There are other ways to import/export project to Git. This is just the writer's way of setting things up.
- 7. We recommend committing the *.project* since the recent release of Eclipse requires this file for a project. If this particular file is missing, you will run into errors while initializing the SuiteCloud project in Eclipse.
- 8. To create a new project out of existing Git repository, you can first clone the repository from Git perspective. Then go to File > Open Projects from File System and select the correct root. If the correct root is selected, you will see a SuiteCloud project added to your Explorer. (You must have the .project file in the root folder.)

### **SDF Gotchas & Our Experience**

NetSuite has worked to address many of the issues that we discovered on different projects. They continue to improve upon SDF and we are thankful that many of these challenges have been addressed. Below are some of the remaining issues that you may experience :

#### Workflows

- Permissions Importing Workflows from an environment or deploying to a target environment requires permissions for your deployment role for all transaction types that are involved in the workflow.
- Workflow Action Script return types Issue deploying workflows with workflow actions with "List/Record" return types when the value is a List. Netsuite expects a record type and rejects the list. The workaround for this is to remove any return type, deploy, then manually reconfigure the return type.
- Account Specific Values Be aware that Account Specific Values could result in unintended behavior for workflows. For example, in an Expense Approval workflow, if the chart of accounts does not match between environments, it may use a seemingly random account for approved expenses.
- **Roles** All roles centers are not supported and not all sublists are supported. Most notably, employee and custom center are not supported and the following sublists are not supported:
  - Subsidiaries field
  - Forms sublist
  - Searches sublist
  - Dashboard sublist

**NOT RECOMMENDED** - The workaround for role center types that are incompatible is to create the role in the target environment with the same ID and make sure the role center type is the intended type. Next, use the console to set the role center type to "BASIC", then save the role. Edit the XML of the imported role and set the role type in the XML to "BASIC". After deployment, set the role center type back to the intended value. We still encountered difficulties with this process, but it was the most consistent solution available.

- Deployment Roles When deploying with SDF the user deploying will need a role with permissions to various records and objects and their dependencies as well as the ability to authenticate and make web service calls to NetSuite preferably using Token Based Authentication. When deploying this creates a Chicken and Egg paradox as you can't deploy the role needed to deploy to the account without deploying. A native or pre-existing role must be created for an SDF deployment to solve this issue. Previously the Administrator role could be used for convenience or as a last resort but with Two Factor Authentication mandatory for any highly privileged roles now this disallows the role from being used. There are a few options and none are very streamlined
  - 1. Manually create a role in the target account. While the most tedious option it is also the most straightforward
  - 2. Install an SDF developer role from a suitebundle. There are many pre-existing bundles for an SDF friendly role. Installing one of these or creating your own SDF developer role bundle can save you a lot of time and headache in the future
  - 3. Create a custom role based on the Native Developer role. The native developer role contains highly privileged permissions and has the same issues as the Administrator role. However you can strip out a few permissions and create a custom role pretty easily that doesn't have mandatory two-factor authentication
    - Access Token Management
    - Integration Application

This role will likely need to be altered with additional permissions but works for simple projects and as a good starting point

- **Error Messages** When an SDF operation fails, it can and will be vague, confusing, unintuitive, or a combination of all three. Focus on resolving any and all small issues with the project being deployed and work through each issue.
- Ensure Uniqueness of script IDs One issue encountered was a nebulous error message while trying to deploy a script "Invalid record type reference key CUSTOMER". However, we found that the underlying issue was that there was a script deployment with the ID "custom deploy1" that would not be unique in the target environment, resolving the issue. In general, validate that the IDs used for your customizations are not default IDs, such as "custbody1".
- Forms The difficulty of migrating forms scales with the level of customization in the source account. As they are currently implemented, an SDF Form XML contains a reference to every customization available to be associated with the form in the account, regardless of whether the form uses the field or not. In more complex accounts, this can lead to an extremely large and dependency heavy XML.

• **Saved Searches** - Saved Searches deployed without a "Publish Search" permission defined in the deployment role will be automatically switched from "Public" to "Private" during deployment. This can break a lot of functionality so it is highly recommended that this permission always be included for a deployment role.

**Validate Against Account:** As we mentioned, you should always use Validate against Account before deployment. However, this feature does not guarantee successful deployment even if it returned a positive result since it only checks against dependencies within project. For example, if there is an issue with permissions, the feature will be unlikely to return an error. We hope that NetSuite could improve the Validate against Account feature so that it can give users warnings as if the change/customization were deployed - especially considering this is the last line of defense before actual deployment.

### **Closing Remarks**

Be sure to check back from time-to-time as we continue to update this guide with new content and information. NetSuite is continuing to build upon this SuiteCloud Development Framework and adding new exciting features you will want to take advantage of. As of this writing, NetSuite is continuing to add more capabilities to round out some of the gaps we have discovered and pain-points we have raised. We at Techfino eagerly look forward to the day where all changes in an account can be managed in version control, deployed via SDF, and automatically validated unit tests and end-to-end integration and functional test cases ensuring quality from end-to-end.

If you are considering using SDF and would like our assistance, we would be more than happy to assist you. Or, if you have any questions, corrections or comments, please send us a note at <u>contact@techfino.com</u> We love hearing from NetSuite customers, practitioners and partners alike!



## Who is Techfino?

Techfino LLC is a passionate group of techno-functional NetSuite experts with an innate desire to help companies of all sizes scale and grow. Headquartered in Philadelphia, this national consultancy specializes in designing, implementing, and managing best-inclass cloud solutions. From NetSuite licensing, to training, to support, to integrations and optimizations, Techfino is hyperfocused on solving the demanding challenges of ever-changing business models, bringing specialized expertise in retail, wholesale, manufacturing, professional services, non-profit, commerce and beyond. Techfino is also proud to have developed several proprietary data management products for NetSuite including Cleansweep File Manager and Archive tools.

For more information about Techfino, their products and service offerings, visit <u>www.techfino.com</u>



"Each of our employees has an innate curiosity to dissect challenges and discover their solutions. That's what keeps our team up at night and keeps us innovating every single day."



Bryan Willman **Managing Partner** 

www.techfino.com contact@techfino.com

### (877) 563-1405



in www.linkedin.com/company/techfino

© 2019 Techfino LLC, All Rights Reserved