

# NetSuite Certified SuiteCloud Developer Exam

---

*Study Guide: April 2018*

## Contents

About the NetSuite Developer Exam .....	2
Description of a Qualified Candidate: .....	3
Maintaining Your Certification .....	3
Recommended Training and Background: .....	4
Subject Areas Covered by the Test:.....	4
Sample Questions.....	7

## About the NetSuite Developer Exam

This is the second exam required for NetSuite Certified SuiteCloud Developer title, to be taken after the SuiteFoundation Exam has been passed.

Passing this SuiteCloud Developer Exam certifies that you have the knowledge and skills necessary to be a **NetSuite Certified SuiteCloud Developer**. See Description of a Qualified Candidate below.

Fee: \$250

Conditions:

- This will be a live proctored examination. Online proctored testing is also available.
- No written or online reference materials may be used during the exam.
- 120 minutes allotted to complete approximately 80 multiple-choice and matching questions

## Description of a Qualified Candidate:

The candidate has ideally 1-2 years of experience working with a range of SuiteCloud Technologies. Additionally, this person has 2-3 years of relevant software development experience.

This person has the knowledge and skills necessary to design, develop, test and deploy secure, scalable, reliable performance solutions to customize, extend and/or interact with NetSuite.

The candidate can explain the implications and benefits of NetSuite development and platform options.

Knowledge of JavaScript, SQL, SOAP, XML, JSON, and REST technologies will also be helpful.

## Maintaining Your Certification

There will be 2 ongoing requirements for maintaining your NetSuite Certified Developer title:

### Annual Release Maintenance

NetSuite will publish a once per year unproctored “Upgrade Test” to validate awareness of new NetSuite features that have been released in the prior year. Information on availability of this Upgrade Test will be communicated in the following ways:

- Posted to the Announcements on the Certification webpage: [www.netsuite.com/certification](http://www.netsuite.com/certification)
- Posted to the closed LinkedIn Certified Developer Group
- Emailed out to the email of record in the Webassessor testing system

### Refresh Developer Exam

NetSuite will rewrite the Developer Exam every 3 years. NetSuite Certified SuiteCloud Developers will be required to retake and pass this Exam every 3 years to maintain their Certification.

## Recommended Training and Background:

### Recommended Training Courses:

- [NetSuite Essentials](#) (5 days)
- [SuiteAnalytics: Reports and Searches](#) (2 days)
- [SuiteAnalytics: Advanced Searches](#) (2 days)
- [SuiteCloud: Exploring the NetSuite Platform](#) (2 days)
- [SuiteFlow: Workflows for Developers](#) (3 days)
- [SuiteScript 1.0: Extend NetSuite with JavaScript](#) (5 days)
- [SuiteTalk: Integrate Your Applications](#) (5 days)

### Recommended Skill Level:

1-2 years of experience working with a range of SuiteCloud Technologies

2-3 years of relevant software development experience

Knowledge of JavaScript, SQL, SOAP, XML, JSON, and REST technologies helpful

## Subject Areas Covered by the Test:

These are the broad subject areas covered on the Exam:

- SuiteScript (pre-2.0)
- Analytics
- SuiteFlow
- SuiteTalk
- SuiteBundler
- SuiteBuilder
- Design Fundamentals

Below is each subject area broken out into more detail:

Topic	Objective
SuiteScript (pre-2.0)	Select the SuiteScript (pre-2.0) code snippet that implements a described business process.
	Recognize the purpose of Plug-Ins in SuiteScript (pre-2.0).
	Identify the functionality and capabilities of SuiteScript debugger
	Given a scenario, identify the proper pattern to invoke a scheduled script to support a long-running process.
	Calculate the governance of a script

	Identify the implications of deploying multiple User Event scripts against a single record, order of execution, and how user events interact with other records in SuiteCloud technologies.
	Identify the risks of implementing only client-side validations and strategies to address them.
	Identify how to dynamically scale a scheduled script utilization to match available queues in account.
	Recognize script deployment configuration across script types.
	Determine which variables would be good candidates to use as a company preference, user preference, or Script Deployment Parameter.
	Identify the impact of execution context on user event scripts.
	Determine how to interact with custom child record sublists in SuiteScript (pre-2.0).
	Determine how to dynamically customize UI field attributes in SuiteScripts (pre-2.0).
	Identify the time zone implications of various ways of setting date and/or time values and the ways in which these values are interpreted in SuiteScript (pre-2.0).
	Identify the capabilities of UI Objects.
SuiteFlow	Identify how to configure standard actions and define custom actions.
	Identify how to control the User Interface throughout the workflow lifecycle.
	Describe the ways in which conditions and triggers can be configured to execute actions and transitions, and Identify capabilities and controls available with states and branching.
	Identify the functionality of record and workflow fields.
	Compare capabilities of SuiteFlow and SuiteScript (pre-2.0) and determine when to use which technology.
	Identify the usage of scheduling in workflow.
	Determine how to use formulas in SuiteFlow.
SuiteTalk	Identify the SuiteTalk (Web Services) support, versioning, deprecation and General Availability policy of endpoints and the process for upgrading them.
	Identify SuiteTalk (Web Services) authentication methods and benefits.
	Determine session policy for Web Services vs UI; and how sessions are managed for the same user across multiple integrated applications, including managing sessions with SuiteCloud Plus.
	Identify how to develop Data Center-agnostic integrations.
	Determine the appropriate search technique to use in SuiteTalk (Web Services).
	Identify how asynchronous and synchronous Web Services APIs impact integration design and implementation.
	Identify strategies for accessing and managing account-specific customizations.
	Identify how to build efficient data synchronization.
Identify the impact of scripts and workflows on integrations.	

SuiteBundler	Identify the role of installation scripts in SuiteBundler.
	Determine how SuiteBundler handles collisions or conflicts during installation or update.
	Identify the impact of a sandbox account refresh on bundles.
SuiteBuilder	Determine the impact of form customization on records.
	Identify the performance implications of adding custom fields and the strategies to mitigate performance impact.
	Given a scenario, select the sourcing and filtering criteria, or the defaulting and validation options for custom fields.
	Identify record-locking behavior and options.
	Identify the various options available under the permissions model for custom records.
	Identify implications of deleting or inactivating custom record types.
SuiteAnalytics	Identify implications of using saved searches and coded searches.
	Identify implications of managing a large volume of search results.
	Identify implications of search techniques in SuiteTalk and SuiteScript (pre-2.0).
Design Fundamentals	Determine how to set roles and permissions for a given situation in various environments.
	Identify strategies for and implications of role management and authentication when integrating with external systems.
	Identify NetSuite functionality and recommended practices related to restricted data fields (PII, PCI, e.g.).
	Determine how to optimize performance, scalability, and reliability in SuiteScript (pre-2.0), SuiteFlow and SuiteTalk.
	Determine how to trouble-shoot and debug in SuiteScript (pre-2.0), SuiteFlow and SuiteTalk.
	Identify considerations when working with different environments (e.g. differences in behavior, testing, customizations, configurations, settings, and preferences, etc.) (e.g. production vs. nonproduction, and different accounts)
	Given a scenario, identify applicable technologies for inbound and outbound integrations, along with their implications.
	Identify how to detect and prevent duplicate records.

## Sample Questions

### Sample Question 1:


Which three conditions will require parts of an integration to be re-coded when the NetSuite WSDL is updated to the newest version? (Choose three.)

- A. The data type for a field has changed
- B. A fix to a defect was released in the latest version
- C. A field that is referenced in an integration has been removed from NetSuite
- D. A field that was required in the previous version is optional in the latest version
- E. A field that was optional in previous versions of NetSuite becomes required in the latest version

Correct Answer: A, C, E

### Sample Question 2:

Given the following screenshot:

Workflow Log	
Workflow Process Vendor	Start Time 2/24/2014 - 14:35:24.119
State State 1	
<input type="checkbox"/> Show Rejected Actions/Transitions	
Entry	Result
Workflow initiated	
Running ONENTRY trigger under BEFORESUBMIT (Event: EDIT; Context: USERINTERFACE)	
SENDEMAIL action skipped. Not executable with trigger BEFORESUBMIT	 Skipped

What two selections can be configured for the Send Email action's Trigger On to fix the error depicted in this snippet of the workflow execution log? (Choose two.)

- A. Entry
- B. Scheduled
- C. Before User Submit
- D. Before Record Load
- E. After Record Submit

Correct answer: B, E