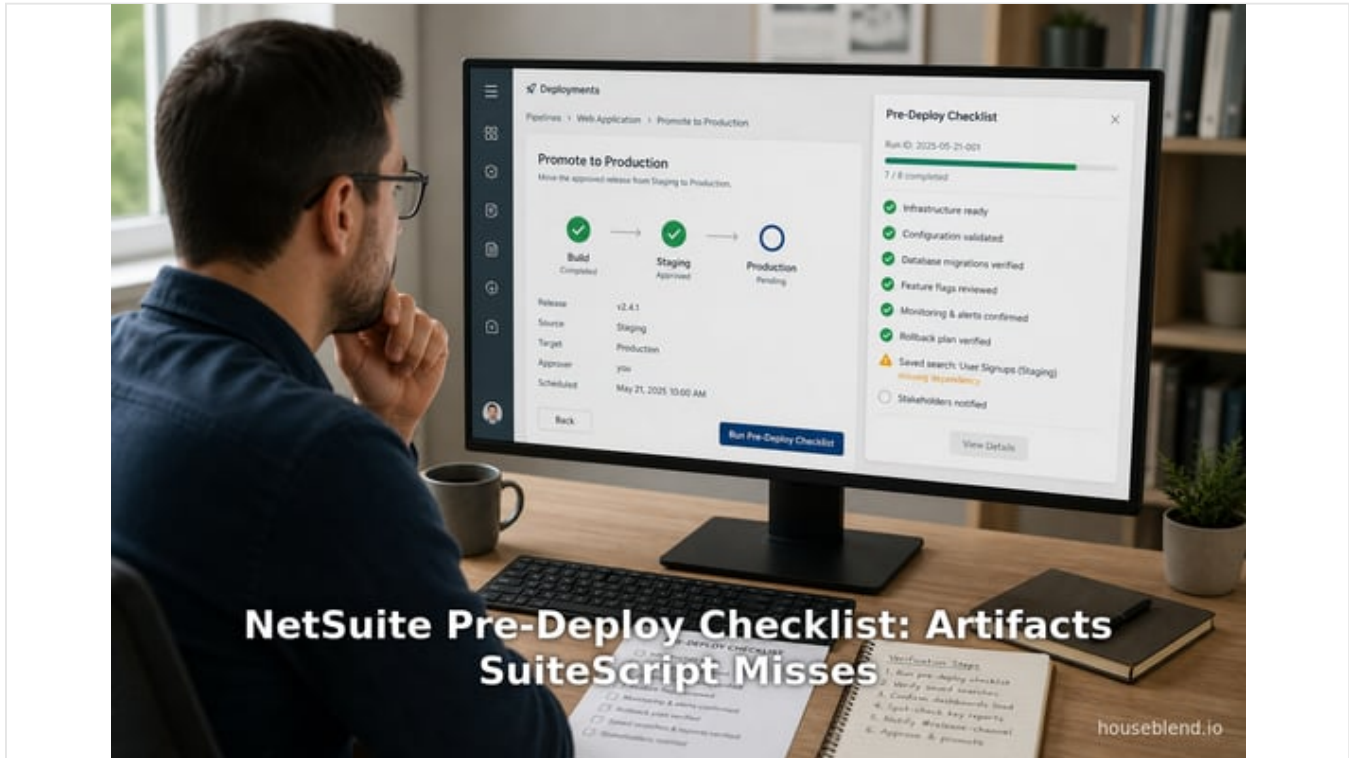


NetSuite Pre-Deploy Checklist: Artifacts SuiteScript Misses

Published August 11, 2026 3 min read



NetSuite Pre-Deploy Checklist: Artifacts SuiteScript Misses

A clean SuiteScript deployment does not by itself prove that the target account has every record, configuration value, permission, and file the script needs. The practical control is to test the complete customization in a production-like target account before go-live.

[SuiteCloud Development Framework \(SDF\)](#) can deploy many custom objects, but only when the applicable objects and files are included in the project. For example, [saved searches](#) can be imported into a project and deployed with SDF, and their dependencies must be imported for the project to deploy successfully. Likewise, SDF deployment packages only the files and folders specified in `deploy.xml`. Treat every external reference in a script as a pre-deployment verification item—not as an assumption that it will be created automatically.

Pre-deployment checks

Complete these checks in the target sandbox or other production-like account, then repeat the critical checks immediately before production deployment.

1. Review the deployment package

- Confirm that `deploy.xml` includes every required object and File Cabinet path. SDF deploys only the files and folders listed there, subject to limited documented exceptions.
- Import and version-control supported objects rather than relying on a source-account configuration that is absent from the project.
- Review [SDF validation output](#) and resolve missing-dependency and account-specific-value errors before promotion.
- If the project references an object supplied by another bundle or SuiteApp, confirm that the required bundle or SuiteApp version is already installed in the target account.

2. Verify saved searches

- Inventory every saved-search reference in code and script-parameter values. Record the intended search name, type, script ID where available, and target purpose.
- In the target account, open each required search and verify that it exists, is active, has the expected criteria and result columns, and is available to the execution context that needs it.
- Do not assume an internal ID from one account identifies the same search in another. Prefer durable, documented references where the implementation supports them, and test the actual lookup in the target account.
- If the search is managed through SDF, verify that its object is in the project. [Oracle documents saved searches as SDF custom objects](#) that can be imported and deployed; a saved search with dependencies requires those dependent objects to be imported as well.

3. Verify script records, deployments, and parameters

- Confirm that the target script deployment is present, deployed, released or scheduled as intended, and points to the intended script file.
- Compare execution settings, audience, event type, schedule, queue or concurrency settings, logging, and run-as role with the approved deployment plan.
- Verify every parameter value on the target deployment—not just the parameter definition on the script record. [When a parameter has no preference setting](#), NetSuite treats it as a deployment script parameter and its value is defined on the script deployment record.
- For parameters that point to a search, custom record, list value, template, folder, employee, or other account object, open the referenced target record and confirm it is the intended one.

4. Verify custom records, custom lists, and custom fields

- Confirm that every required custom record type, custom list, custom field, and list value exists in the target account and has the expected script ID, field type, access settings, and inactive status.
- Check that code which loads a record or reads a field resolves to the expected target object. Do not rely on a sandbox internal ID unless the implementation and deployment method explicitly preserve that reference.
- If the customization depends on existing data rather than only the record definition, verify that required instances and list values have been migrated or created through an approved process.

5. Verify roles and permissions

- Test with the role and execution context the script will actually use in production. Confirm the role exists, is assigned where required, and has the least privileges necessary to read, create, edit, or delete the relevant records and files.
- Verify access to searches, File Cabinet locations, email templates, workflows, and feature-dependent records—not only access to the main transaction or custom record.
- Record any required NetSuite features and confirm they are enabled in the target account. [SDF can declare feature dependencies](#) so a deployment fails when a required feature is unavailable; use that mechanism where it fits the project.

6. Verify workflows and email templates

- Confirm that each workflow definition is present, released or inactive as intended, attached to the correct record type, and has the expected states, actions, conditions, and transitions.
- Confirm that workflow actions which call scripts, send email, or update fields reference target-account objects that exist and are accessible.
- Open each email template used by a script or workflow and verify its subject, content, sender and recipient logic, restrictions, and referenced files. [Custom email templates can be managed as SDF objects](#), but the project must include the template object and, where applicable, its HTML, FreeMarker, or text definition file.

Run a production-like smoke test

After deployment, run the smallest safe end-to-end transaction or record scenario that exercises the entry point, each required search, parameter lookup, workflow action, permission boundary, and notification. Review the script execution log and resulting records before enabling the customization for normal processing.

Keep the checklist with the release record. The goal is not to prove that code deployed; it is to prove that the code, configuration, dependencies, permissions, and target-account data work together.

Tags: netsuite, suitescript, saved searches, sandbox to production, script parameters, sdf deployment, go-live checklist

DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will [Houseblend](#) or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.

© 2026 [Houseblend](#). All rights reserved.