should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Master Agreement, Oracle License and Services Agreement, Oracle PartnerNetwork Agreement, Oracle distribution agreement, or other license agreement which has been executed by you and Oracle and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Sample Code

Oracle may provide sample code in SuiteAnswers, the Help Center, User Guides, or elsewhere through help links. All such sample code is provided “as is” and “as available”, for use only with an authorized NetSuite Service account, and is made available as a SuiteCloud Technology subject to the SuiteCloud Terms of Service at www.netsuite.com/tos.

Oracle may modify or remove sample code at any time without notice.

No Excessive Use of the Service

As the Service is a multi-tenant service offering on shared databases, Customer may not use the Service in excess of limits or thresholds that Oracle considers commercially reasonable for the Service. If Oracle reasonably concludes that a Customer's use is excessive and/or will cause immediate or ongoing performance issues for one or more of Oracle's other customers, Oracle may slow down or throttle Customer's excess use until such time that Customer's use stays within reasonable limits. If Customer's particular usage pattern requires a higher limit or threshold, then the Customer should procure a subscription to the Service that accommodates a higher limit and/or threshold that more effectively aligns with the Customer's actual usage pattern.

Beta Features

Oracle may make available to Customer certain features that are labeled “beta” that are not yet generally available. To use such features, Customer acknowledges and agrees that such beta features are subject to the terms and conditions accepted by Customer upon activation of the feature, or in the absence of such terms, subject to the limitations for the feature described in the User Guide and as follows: The beta feature is a prototype or beta version only and is not error or bug free and Customer agrees that it will use the beta feature carefully and will not use it in any way which might result in any loss, corruption or unauthorized access of or to its or any third party's property or information. Customer must promptly report to Oracle any defects, errors or other problems in beta features to support@netsuite.com or other designated contact for the specific beta feature. Oracle cannot guarantee the continued availability of such beta features and may substantially modify or cease providing such beta features without entitling Customer to any refund, credit, or other compensation. Oracle makes no representations or warranties regarding functionality or use of beta features and Oracle shall have no liability for any lost data, incomplete data, re-run time, inaccurate input, work delay, lost profits or adverse effect on the performance of the Service resulting from the use of beta features. Oracle's standard service levels, warranties and related commitments regarding the Service shall not apply to beta features and they may not be fully supported by Oracle's customer support. These limitations and exclusions shall apply until the date that Oracle at its sole option makes a beta feature generally available to its customers and partners as part of the Service without a “beta” label.
Send Us Your Feedback

We'd like to hear your feedback on this document. Answering the following questions will help us improve our help content:

- Did you find the information you needed? If not, what was missing?
- Did you find any errors?
- Is the information clear?
- Are the examples correct?
- Do you need more examples?
- What did you like most about this document?

Click [here](#) to send us your comments. If possible, please provide a page number or section title to identify the content you're describing.

To report software issues, contact NetSuite Customer Support.
## Table of Contents

SuiteCloud IDE Plug-in for WebStorm Overview ................................................................. 1
UI Interface of SuiteCloud IDE Plug-in for WebStorm ....................................................... 1
Installing and Setting Up SuiteCloud IDE Plug-in for WebStorm .................................... 4
Installing SuiteCloud IDE Plug-in for WebStorm ............................................................... 5
  - Installation Requirements for SuiteCloud IDE Plug-in for WebStorm ...................... 5
  - Downloading WebStorm ......................................................................................... 6
  - Adding the SuiteCloud IDE Plug-in to WebStorm .............................................. 6
Setting Up SuiteCloud IDE Plug-in for WebStorm ......................................................... 8
  - SuiteCloud IDE Plug-in for WebStorm Master Password Setup .......................... 8
  - SuiteCloud IDE Plug-in for WebStorm Domain Setup ....................................... 10
  - SuiteCloud IDE Plug-in for WebStorm Account Setup ....................................... 11
  - Managing Token-based Authentication in an Account using SuiteCloud IDE Plug-in for WebStorm .............................................. 12
SuiteCloud IDE Plug-in for WebStorm Usage ................................................................. 15
  - Working with SDF Projects in SuiteCloud IDE Plug-in for WebStorm ............... 15
  - Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm ....................................................... 15
  - Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm ................ 16
  - Customizing SuiteCloud IDE Plug-in for WebStorm ........................................ 16
  - Importing Account Components to your SDF Project with SuiteCloud IDE Plug-in for WebStorm .............................................. 17
Working with Custom NetSuite Objects in SuiteCloud IDE Plug-in for WebStorm .......... 19
  - Creating an XML Definition for a Custom Object in SuiteCloud IDE Plug-in for WebStorm ....................................................... 19
  - Overwriting Custom SuiteCloud Objects with Custom NetSuite Objects in SuiteCloud IDE Plug-in for WebStorm .............................................. 20
  - Comparing a Custom SuiteCloud Object with a Custom NetSuite Object in SuiteCloud IDE Plug-in for WebStorm .............................................. 21
  - Comparing an SDF Project File with a File in SuiteCloud IDE Plug-in for WebStorm ....................................................... 21
Working with SuiteScript Files in SuiteCloud IDE Plug-in for WebStorm .................... 22
  - Creating a SuiteScript File in SuiteCloud IDE Plug-in for WebStorm ............... 22
  - Uploading Files or Folders to Your Account with SuiteCloud IDE Plug-in for WebStorm ....................................................... 24
Adding Dependencies to the Manifest in SuiteCloud IDE Plug-in for WebStorm .......... 24
  - Defining Dependencies on SuiteApp and Bundle Objects with SuiteCloud IDE Plug-in for WebStorm ....................................................... 24
  - Defining Account Component Dependencies from an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm ....................................................... 26
  - Defining Feature Dependencies in SuiteCloud IDE Plug-in for WebStorm .......... 27
Changing Project Settings in SuiteCloud IDE Plug-in for WebStorm ......................... 28
  - Setting Installation Preferences in SuiteCloud IDE Plug-in for WebStorm .......... 28
Using SuiteScript Code Completion in SuiteCloud IDE Plug-in for WebStorm .......... 31
Validating an SDF Project with SuiteCloud IDE Plug-in for WebStorm .................... 33
Deploying an SDF Project to Your NetSuite Account With SuiteCloud IDE Plug-in for WebStorm ....................................................... 34
Shortcut Keys in SuiteCloud IDE Plug-in for WebStorm ................................................ 35
SuiteCloud IDE Plug-in for WebStorm Overview

SuiteCloud IDE plug-in for WebStorm is an integrated development environment (IDE) that is packaged for NetSuite platform development. It provides a user interface for SuiteCloud Development Framework (SDF). SDF is a development framework and deployment mechanism.

You can download SuiteCloud IDE plug-in for WebStorm as a plug-in installation, develop SDF projects, and deploy them to your account.

From an SDF project in the WebStorm plug-in, you can do the following:

- Manage multiple NetSuite domains and accounts.
- Import custom objects from account.
- Upload files and folders to the NetSuite File Cabinet.
- Compare files or objects with the NetSuite version.
- Use SuiteScript 2.0 API code completion.
- Set a master password for each project.
- Access NetSuite menus at the file folder and project level to select context menu actions. For example, you can add dependencies to the manifest, validate against account, deploy to account, and change project settings.
- View SDF logs in the IDE plug-in Tool Window.

**Note:** An SDF project can be either an account customization project type or a SuiteApp project type.

To learn more and start using the WebStorm plug-in, see the following topics:

- Installing and Setting Up SuiteCloud IDE Plug-in for WebStorm
- SuiteCloud IDE Plug-in for WebStorm Usage

**UI Interface of SuiteCloud IDE Plug-in for WebStorm**

SuiteCloud IDE plug-in for WebStorm is a custom plug-in added to WebStorm. After you install it, the menus and toolbars you need for managing and working with SuiteCloud projects and resources are accessible within the IDE plug-in.

When you open an SDF project, the UI consists of an editor area, tabs, and tool windows that you use as you work with your SuiteCloud projects.
The following table lists the elements of the WebStorm plug-in interface:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
</table>
| **File Menu**    | From the File menu, you can do the following and more:  
  - Create new SDF projects  
  - Open, rename, or save projects.  
  - Check for plug-in updates by opening Settings  
  For more information, see the WebStorm product help. |
| **NetSuite Menu**| Use the NetSuite menu to manage your domains, accounts, and project settings. You can also change, revoke, or reset the master password.  
  The Outline view displays the organizational structure of your source file that is active in the editor area. In general, it lists the structural elements of your source file by function.  
  For more information, see [SuiteCloud IDE Plug-in for WebStorm Usage](#). |
| **SuiteCloud IDE Tool Window**| The IDE plug-in tool window displays the hierarchical view of your SDF projects (their folders and files) and resources.  
  You can hover over the project folder to see the path where the project is located.  
  From the top-left of the screen, click Project to minimize or maximize the window.  
  For more information, see the [Project tool window](#) topic in the WebStorm product help. |
| **Editor Area**  | The Editor area displays the source file editors you can use for your code.  
  For more information, see the Editors topic in the WebStorm product help. |
| **SuiteScript Libraries** | SuiteScript 1.0 and SuiteScript 2.0 API libraries are automatically included in an SDF project to support code completion for both SuiteScript versions. If you do not need code completion for a particular SuiteScript version, you can delete the file from the project.  
  For more information, see [Using SuiteScript Code Completion in SuiteCloud IDE Plug-in for WebStorm](#). |
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDF Logs</td>
<td>SDF Logs display the output, including any errors and warnings found, during SDF project validation or deployment. From the bottom-left of the screen, click the SuiteCloud IDE tab to show or hide the window.</td>
</tr>
<tr>
<td>Event Log</td>
<td>The Event Log displays a summary of requests and actions. For example, if you upload a file from a project, when SDF has finished attempting the operation, the log displays the following: Uploading files done. From the bottom-right of the screen, click Event Log to minimize or maximize the window.</td>
</tr>
<tr>
<td>Progress View</td>
<td>The Progress view displays the progress of all your upload or download actions.</td>
</tr>
</tbody>
</table>
Installing and Setting Up SuiteCloud IDE Plug-in for WebStorm

SuiteCloud IDE plug-in for WebStorm is installed from within the WebStorm client. After you download and install WebStorm, you can add the IDE plug-in update site to the list of software updates.

The following help videos demonstrate how to install and set up the SuiteCloud IDE plug-in for WebStorm:

- To see how to install WebStorm and IDE plug-in for WebStorm, see SuiteCloud Development Framework: Installing the WebStorm IDE Plug-in for SDF.
- To see how to set up your domains and accounts in IDE plug-in for WebStorm, see SuiteCloud Development Framework: NetSuite Domains and Accounts in the IDE Plug-in for WebStorm.

Installation of SuiteCloud IDE Plug-in for WebStorm

Follow these steps to install the WebStorm plug-in:

2. **(Optional) Configure your account.**
   - Configuring NetSuite for SuiteScript if you are using SuiteScript.
   - Enabling SDF in Your NetSuite Account (Admin Only) if you are using SuiteCloud Development Framework.
3. **Download WebStorm.** See Downloading WebStorm.
4. **Add the SuiteCloud IDE Update Site.** See Adding the SuiteCloud IDE Plug-in to WebStorm.

**Important:** The version of the WebStorm plug-in must match the version of your account to avoid the risk of compatibility errors.

For information about the WebStorm plug-in features, see SuiteCloud IDE Plug-in for WebStorm Overview.

Setup for SuiteCloud IDE Plug-in for WebStorm

After installing the WebStorm plug-in, follow these steps to complete your setup:

1. **Create or open an SDF project.** See Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm or Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm.
2. **Set a master password.** See SuiteCloud IDE Plug-in for WebStorm Master Password Setup.
3. **Add your production and sandbox URLs.** See Adding a Domain in SuiteCloud IDE Plug-in for WebStorm.
4. **Connect to your accounts and roles.** See Adding an Account in SuiteCloud IDE Plug-in for WebStorm.
5. **(Optional) Manage authentication.** If you want to use an administrator role, set up login for token-based authentication (TBA). See Managing Token-based Authentication in an Account using SuiteCloud IDE Plug-in for WebStorm.
Installing SuiteCloud IDE Plug-in for WebStorm

The following topics guide you through WebStorm and the IDE plug-in installation:

- Installation Requirements for SuiteCloud IDE Plug-in for WebStorm
- Supported Operating Systems for SuiteCloud IDE Plug-in for WebStorm
- Downloading WebStorm
- Adding the SuiteCloud IDE Plug-in to WebStorm

Installation Requirements for SuiteCloud IDE Plug-in for WebStorm

Verify the following prerequisites before you install and use WebStorm with the IDE plug-in:

- **Operating system and browser** — Verify that the operating system version that you want to use with the WebStorm plug-in is supported.
  See [Supported Operating Systems for SuiteCloud IDE Plug-in for WebStorm](#).
- **Java Runtime Environment (JRE)** — The IDE plug-in requires Java SE Development Kit version 11.
  Note: JRE 11 is bundled with the WebStorm 2019 distribution. You do not need to install Java on your computer to run WebStorm.
- **64-bit requirements** — To properly run the 64-bit version of the WebStorm plug-in, make sure your operating system and Java versions are both 64-bit versions.
- **Two-factor authentication setup** — If you want to use an administrator role and any other role that requires 2FA for SDF development, the SuiteCloud Development Integration bundle must be installed to the account.
  For more information about using a role requiring 2FA for SDF development, see [Managing Token-based Authentication in an Account using SuiteCloud IDE Plug-in for WebStorm](#).
- **Web services enabled** — Make sure SuiteTalk (Web Services) is enabled in the NetSuite account.
  See the help topic [Enabling SDF in Your NetSuite Account (Admin Only)](#).
- **Account** — Your WebStorm plug-in version should match the version of your account. You may experience compatibility issues by using an older version of the WebStorm plug-in with the latest version of NetSuite.
- **WebStorm version** — Depending on the WebStorm plug-in version you use, use one of the following WebStorm versions:
  □ For 19.2 WebStorm IDE plug-in version, the IDE plug-in requires WebStorm version 2018.3.5.
  □ For 20.1 WebStorm IDE plug-in version, the IDE plug-in requires WebStorm version 2019.3.1.

---

**Important:** If you want to use WebStorm 2019.2, downgrade to JRE version 1.8 by following the instructions in the IntelliJ website. However, the IDE plug-in for WebStorm is not fully functional with WebStorm 2019.2. You should use WebStorm 2018.2.5, or WebStorm 2018.3.5, depending on your account's version.

---

**Note:** To set up a different SuiteScript development environment instead of the IDE plug-in, see the help topic [Working with IDEs Other Than SuiteCloud IDE Plug-ins](#).
Installing SuiteCloud IDE Plug-in for WebStorm

Note: The 20.1 version of the WebStorm plug-in does not support working with multiple projects in one window.

Supported Operating Systems for SuiteCloud IDE Plug-in for WebStorm

The IDE plug-in can be used on the following operating systems:

- Windows 10
- Ubuntu 18.04.01
- Mac OS High Sierra 10.13.6

Downloading WebStorm

For supported versions of WebStorm for the IDE plug-in, see Installation Requirements for SuiteCloud IDE Plug-in for WebStorm.

1. Download WebStorm for your operating system at https://www.jetbrains.com/webstorm/.
2. Run the file you downloaded and follow the prompts to complete installation.

Next you must add the SuiteCloud IDE Update site. For more information, see Adding the SuiteCloud IDE Plug-in to WebStorm.

Adding the SuiteCloud IDE Plug-in to WebStorm

The WebStorm plug-in is installed to WebStorm as a custom plug-in. You need to download the IDE plug-in package from the SuiteCloud IDE update site. Complete the following tasks:

1. Add the SuiteCloud IDE Plug-in for WebStorm Update Site.

Important: When your account is upgraded, your instance of the IDE plug-in does not get updated automatically. After your account upgrade, you must manually update the WebStorm plug-in to match the version of your account. You may experience compatibility issues by using an older version of the WebStorm plug-in with the latest version of NetSuite.

Add the SuiteCloud IDE Plug-in for WebStorm Update Site

When you add the SuiteCloud IDE update site to WebStorm, you can download the latest version of the IDE plug-in that is compatible with your NetSuite account as a custom plug-in. Installing the WebStorm plug-in enables you to access the IDE plug-in and SDF options on your WebStorm client.

To add the update site in SuiteCloud IDE Plug-in for WebStorm:

1. Launch WebStorm.
2. Depending on your operation system, do one of the following:
   - If you are using a Windows OS, select File > Settings.
   - If you are using a Mac OS, select WebStorm > Preferences.
Installing SuiteCloud IDE Plug-in for WebStorm

3. Select Plugins, and click.
4. Select Manage Plugin Repositories....
5. In the Custom Plugin Repositories window, click and enter the following:
   - If you want to get the latest changes automatically every time the plug-in is updated, use
     https://system.netsuite.com/download/suitecloud-sdk/ideplugin/webstorm/latest/updatePlugins.xml
   - If your account is on version 20.1, use:
   - If your account is on version 19.2, use:
     https://system.netsuite.com/download/suitecloud-sdk/ideplugin/webstorm/19.2/updatePlugins.xml
6. Select OK.
7. Click on the NetSuite's SuiteCloud IDE Plug-in plugin, and click Install.
8. Restart WebStorm.

Update SuiteCloud IDE Plug-in for WebStorm

To update SuiteCloud IDE Plug-in for WebStorm:
1. Launch WebStorm.
2. Depending on your operation system, do one of the following:
   - If you are using a Windows OS, select File > Settings.
   - If you are using a Mac OS, select WebStorm > Preferences.
3. Select Plugins.
4. Set the Show field to Custom.
5. In the list, click SuiteCloud IDE for WebStorm.
   If an update is available, the Update button appears.
6. Click Update.
7. Restart WebStorm.

Verify the Installation of SuiteCloud IDE Plug-in for WebStorm

When you install SuiteCloud IDE plug-in for WebStorm, information about the NetSuite SuiteCloud IDE version is added. You can verify this installation information at any time.

To verify the installation of SuiteCloud IDE plug-in for WebStorm:
1. Launch WebStorm.
2. Depending on your operation system, do one of the following:
   - If you are using a Windows OS, select File > Settings.
   - If you are using a Mac OS, select WebStorm > Preferences.
3. Select Plugins.
4. In the list, click SuiteCloud IDE for WebStorm.
   The WebStorm plug-in version information is displayed.
Setting Up SuiteCloud IDE Plug-in for WebStorm

See the following topics to set up SuiteCloud IDE plug-in for WebStorm for creating SDF projects:

- SuiteCloud IDE Plug-in for WebStorm Master Password Setup
- SuiteCloud IDE Plug-in for WebStorm Domain Setup
- SuiteCloud IDE Plug-in for WebStorm Account Setup
- Managing Token-based Authentication in an Account using SuiteCloud IDE Plug-in for WebStorm

SuiteCloud IDE Plug-in for WebStorm Master Password Setup

SuiteCloud IDE plug-in for WebStorm stores all of your account login information every time you add a project. To prevent unauthorized account changes, you must set up a master password.

**Note:** Your IDE plug-in master password is not your NetSuite password. It is used to ensure that the data in your IDE plug-in remains secure.

To upload a project, download a project, or interact with your NetSuite account, you must authenticate your master password once per session.

Your master password not only protects all of your account login information, but it also saves you from entering different passwords for different accounts when you have multiple NetSuite accounts.

See the following procedures for working with the master password:
Setting a SuiteCloud IDE Plug-in for WebStorm Master Password

Password length must be 7 to 16 characters.

1. From WebStorm, select **NetSuite > Master Password > Set Master Password**.
2. In the Set Master Password window, enter the following information: **New Master Password** and **Re-enter New Master Password**.
3. Click **OK**.

Authenticating a SuiteCloud IDE Plug-in for WebStorm Master Password

To upload a project, download a project, or interact with your NetSuite account, you must authenticate your master password once per session.

1. In WebStorm, select **NetSuite > Master Password > Authenticate Master Password**.
2. Enter the master password.
3. Click **OK**.

The keyboard shortcut to open the Authenticate Master Password window is **Ctrl+Alt+A**. The shortcut works only when the password has not been authenticated in the session.

Revoking a SuiteCloud IDE Plug-in for WebStorm Master Password

You can revoke your master password to avoid accidental or unauthorized interaction with your NetSuite account. For example, if someone else needs to look at your IDE plug-in editor to do a peer review of your code, you can revoke your master password to protect your NetSuite accounts.

Revoking a master password only logs you out. It does not delete the master password.

1. In WebStorm, select **NetSuite > Master Password > Revoke Master Password**.
2. After the Revoke Master Password window appears, click **OK**.

Changing a SuiteCloud IDE Plug-in for WebStorm Master Password

You can change a master password if you know the current password. If you do not remember your master password, see **Resetting a SuiteCloud IDE Plug-in for WebStorm Master Password**.

1. In WebStorm, select **NetSuite > Master Password > Change Password**.
2. Enter the following information: **Old Master Password**, **New Master Password**, and **Re-enter New Master Password**.
3. Click **OK**.

### Resetting a SuiteCloud IDE Plug-in for WebStorm Master Password

If you forgot your master password, you must reset your master password and account information. For security reasons, there is no way for you to retrieve your master password.

**Note:** A password reset removes connection to any associated NetSuite accounts. You must set up your IDE plug-in domains and accounts again.

1. In WebStorm, select **NetSuite > Troubleshoot > Reset Master Password and Account Info**.
   
   A message appears warning you that the WebStorm plug-in will be restarted and that your master password and all of your account information will be deleted.

2. Click **OK**.
   
   The WebStorm plug-in restarts with your project still intact but your master password and account info deleted.

3. Re-add your master password and account info. For more information, see **Setting a SuiteCloud IDE Plug-in for WebStorm Master Password and SuiteCloud IDE Plug-in for WebStorm Account Setup**.

### SuiteCloud IDE Plug-in for WebStorm Domain Setup

After setting up your master password, you can set up a NetSuite domain in which to run your customizations. The predefined NetSuite domains in the WebStorm plug-in are:

- Production

To set up a domain, see the following procedures:

- **Adding a Domain in SuiteCloud IDE Plug-in for WebStorm**
- **Modifying a Domain in SuiteCloud IDE Plug-in for WebStorm**
- **Removing a Domain in SuiteCloud IDE Plug-in for WebStorm**

### Adding a Domain in SuiteCloud IDE Plug-in for WebStorm

You can set up the domains that include the accounts you want to add. You can choose NetSuite domains with accounts associated with your **Production** or **Sandbox** accounts.

1. In WebStorm, select **NetSuite > Manage Domains**.
2. From the Manage Domains window, click **Add (+)**.
3. From the New Domain window, enter information for the **URL** field.

**Note:** If you do not know the correct URL to use for a particular URL, see the help topic **URLs for Account-Specific Domains**.

4. Click **OK** and then **Close**.

### Modifying a Domain in SuiteCloud IDE Plug-in for WebStorm

You can change the domain name or point to its updated URL.
1. In WebStorm, select **NetSuite > Manage Domains.**
2. In the **Domains** list, select the domain that you want to edit.
3. Click **Edit (ISTRIBUTION).**
4. From the Edit Domain window, modify the information in the **URL** field.

   **Note:** If you do not know the correct URL to use for a particular environment, see the help topic **URLs for Account-Specific Domains.**

5. Click **OK** and then **Close.**

### Removing a Domain in SuiteCloud IDE Plug-in for WebStorm

You can remove domains from the IDE plug-in, such as unused domains you no longer need.

1. In WebStorm, select **NetSuite > Manage Domains.**
2. In the **Domains** list, select the domain that you want to remove.
3. From the Manage Domains window, click **Remove (minus).**
4. From the Remove Domain window, click **OK.**
5. Click **Close.**

### SuiteCloud IDE Plug-in for WebStorm Account Setup

After setting up your master password, you need to set up the NetSuite accounts you will be customizing. If you have set up domains in addition to the default NetSuite domains, you need to configure account-specific details for these additional domains you added.

To set up an account, see the following procedures:

- Adding an Account in SuiteCloud IDE Plug-in for WebStorm
- Removing a Domain in SuiteCloud IDE Plug-in for WebStorm

### Adding an Account in SuiteCloud IDE Plug-in for WebStorm

Use the following steps to add an account. The WebStorm plug-in passes your account information to NetSuite during the IDE plug-in authentication process.

**Important:** Ensure that the NetSuite accounts you add comply with NetSuite password requirements regarding the special characters you can use. For more information, see the help topic **NetSuite Password Requirements.**

1. In WebStorm, select **NetSuite > Manage Accounts.**
2. Click **Add (plus).**
3. In the **URL** field, select the domain.
4. In the **Email** field, enter the **Email** associated with the NetSuite role.
5. In the **Password** field, enter the **NetSuite password** associated with the email address.
6. Click **Next**.

7. To select the account(s) you want to add, click **Select All** ( ), or click **Deselect All** ( ) and then select what you need.

8. Click **Finish**. The Manage Accounts window opens with your added accounts. Existing accounts are maintained. Newly-found accounts are added when you select and add them.

9. Click **Close**.

When you change your NetSuite password, you must add your accounts again using your latest NetSuite login credentials. However, the IDE plug-in automatically prompts you to enter your latest login credentials if you have updated your login credentials but have not re-added your accounts in the IDE plug-in.

### Removing an Account in SuiteCloud IDE Plug-in for WebStorm

You can remove inactive or unused accounts that you no longer need.

**Note:** If you want to modify a domain in the IDE plug-in, you must first remove the associated accounts associated with a domain.

1. In WebStorm, select **NetSuite > Manage Accounts**.
2. In the **Accounts** list, select the account that you want to remove.
3. Click **Remove** ( ).
4. Click **OK** and then **Close**.

### Managing Token-based Authentication in an Account using SuiteCloud IDE Plug-in for WebStorm

You can use token-based authentication (TBA) to log into custom roles in NetSuite Production account from the IDE plug-in. By using TBA with the IDE plug-in, you experience fewer session terminations when using a single login to develop SDF projects and navigate NetSuite. Additionally, TBA provides increased development environment security. For more information about setting up TBA with your Production account, see the help topic **Getting Started with Token-based Authentication**.

Depending on whether your role requires 2FA, do one of the following:

- For accounts that do not require 2FA, you can issue and revoke a TBA token in IDE plug-in for WebStorm. See **To issue or revoke a TBA token in SuiteCloud IDE Plug-in for WebStorm**:
- For accounts that require 2FA, you need to issue the token in NetSuite and save it in IDE plug-in for WebStorm. See **To save a TBA token in SuiteCloud IDE Plug-in for WebStorm**:

**Important:** To work with TBA, your 2FA role needs to be properly set up. For more information, see the help topic **Setting Up Your Role for SDF Development**.

### Issue or revoke a TBA token in SuiteCloud IDE Plug-in for WebStorm

To use TBA, issue a TBA token for a custom role in an account and log in to that account with that role assigned. To use account credentials instead of TBA, revoke the active TBA token on the account.
To issue or revoke a TBA token in SuiteCloud IDE Plug-in for WebStorm:

1. Ensure that your account administrator has set up a custom role for you to use with SDF.
   For information about granting permissions to a custom SDF developer role, see the help topic Assigning the Developer Role (Admin Only).
2. In WebStorm, select NetSuite > Manage Accounts.
3. In the Accounts list, select the account that you want to log in to using token-based authentication.
4. Click Manage Authentication ().
   A window appears, showing all the roles that are assigned to the user.
5. In the Roles list, select the role that you want to log in to using token-based authentication.
6. To issue or revoke the TBA token on the role, click Issue () or Revoke ().
   Issue indicates that TBA is available for the role but not in use. Users with this role are logged in to NetSuite from the WebStorm plug-in using account credentials.
   Revoke indicates that the role uses TBA to log in to NetSuite from the WebStorm plug-in.
7. Click Close on both windows.

To save a TBA token in SuiteCloud IDE Plug-in for WebStorm:

1. In WebStorm, select NetSuite > Manage Accounts.
2. In the Accounts list, select the account that you want to log in to using token-based authentication.
3. Click Manage Authentication ().
   A window appears, showing all the roles that are assigned to the user. If Authentication Type is set to Token Based Authentication, you can save a TBA token for that role.
4. In the Roles list, select the role that you want to log in to using token-based authentication.
5. Click Save Token ().
6. Enter or edit the token ID and token secret using the credentials issued from NetSuite for that role.
   For information on how to issue a token in NetSuite, see the help topic Viewing, Editing, Creating, and Revoking TBA Tokens.
7. Click **Save**.
8. Click **Close** on both windows.

**Note:** After successfully saving the token, SDF stores the information and you can use the role for SDF features that depend on account login such as validation, deployment.
SuiteCloud IDE Plug-in for WebStorm Usage

Before you start working with the IDE plug-in, you should familiarize yourself with the following:

- SuiteCloud Development Framework
- SuiteCloud IDE Plug-in for WebStorm Overview
- Installing and Setting Up SuiteCloud IDE Plug-in for WebStorm

Working with the IDE plug-in is divided into the following task areas:

- Working with SDF Projects in SuiteCloud IDE Plug-in for WebStorm
- Working with Custom NetSuite Objects in SuiteCloud IDE Plug-in for WebStorm
- Working with SuiteScript Files in SuiteCloud IDE Plug-in for WebStorm

Working with SDF Projects in SuiteCloud IDE Plug-in for WebStorm

- Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm
- Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm
- Customizing SuiteCloud IDE Plug-in for WebStorm

Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm

To customize your account, you can use SDF to create account customization projects.

After you create a project, you can set it up to connect the accounts you want to customize.

**Note:** By default, all account customization projects created in SuiteCloud IDE plug-in for WebStorm work with SuiteScript 2.0.

For more information about account customization projects, see:

- Account Customization Projects
- Differences Between SDF Projects

**To create an account customization project in SuiteCloud IDE Plug-in for WebStorm**

1. In WebStorm, select **File > New > Project....**
2. From the New Project dialog, select the project type: **Account Customization**.
3. Specify a location where you want to save your project or accept the default.
4. Set the remaining project properties. For more information, see the help topic **Properties of an Account Customization Project**.
5. Click **Create**.
   WebStorm opens a new window for each project.
Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm

You can use SDF to create SuiteApp projects, which are intended for SuiteCloud Developer Network (SDN) members creating SDF SuiteApps.

**Note:** By default, all SuiteApp projects created in SuiteCloud IDE plug-in for WebStorm work with SuiteScript 2.0.

For more information about SuiteApp projects, see:
- SuiteApp Projects
- Differences Between SDF Projects

**To create a SuiteApp project in SuiteCloud IDE Plug-in for WebStorm**

1. In WebStorm, select **File > New > Project**.
2. From the New Project dialog, select the project type: **SuiteApp**.
3. Specify a location where you want to save your project or accept the default.
4. Set the remaining project properties. For more information, see the help topic Properties of a SuiteApp Project.
5. Click **Create**.

WebStorm opens a new window for each project.

**Note:** If you do not see the project, look for another WebStorm window. From the window, if necessary, click **Project** and expand the project to see the project structure. For more information about the IDE plug-in UI, see SuiteCloud IDE Plug-in for WebStorm Overview and Customizing SuiteCloud IDE Plug-in for WebStorm.

6. Add the domains and accounts you want to customize. See Adding a Domain in SuiteCloud IDE Plug-in for WebStorm and Adding an Account in SuiteCloud IDE Plug-in for WebStorm.

Customizing SuiteCloud IDE Plug-in for WebStorm

You can customize SuiteCloud IDE plug-in for WebStorm according to your preferences:

- Changing Where Projects Open in SuiteCloud IDE Plug-in for WebStorm
- Turning Off Code Completion in SuiteCloud IDE Plug-in for WebStorm

For more information, visit WebStorm Help.
Changing Where Projects Open in SuiteCloud IDE Plug-in for WebStorm

By default, the WebStorm plug-in prompts you to choose opening the project in the same window or in a new window. You can change this setting.

**To always open projects in a new window:**

1. In WebStorm, select **File > Settings...**
2. Select **Appearance & Behavior > System Settings**.
3. Under **Project Opening**, choose **Open project in new window** to open projects in a separate window.

Turning Off Code Completion in SuiteCloud IDE Plug-in for WebStorm

**Note:** If you disable code completion, you can still get suggestions when pressing Ctrl + Spacebar. For information about SuiteScript code completion, see Using SuiteScript Code Completion in SuiteCloud IDE Plug-in for WebStorm.

**To turn off code completion in SuiteCloud IDE Plug-in for WebStorm:**

1. In WebStorm, select **File > Settings...**
2. Select **Editor > General > Code Completion**.
3. Clear all the boxes, click **Apply** and then, **OK**.

Importing Account Components to your SDF Project with SuiteCloud IDE Plug-in for WebStorm

You can import custom NetSuite objects to account customization projects or SuiteApp projects after you have authenticated the master password. You can also use SDF to import custom NetSuite objects, files, and scripts from your account into an account customization project.

See the following topics:

- Importing Custom NetSuite Objects into an SDF Project with SuiteCloud IDE Plug-in for WebStorm
- Importing Files and Scripts into an Account Customization Project with SuiteCloud IDE Plug-in for WebStorm
- Account Component Imports

Importing Custom NetSuite Objects into an SDF Project with SuiteCloud IDE Plug-in for WebStorm

1. From an SDF project in the WebStorm plug-in, right-click the **Objects** folder.
2. Select **NetSuite > Import Custom Objects from Account**.
3. Choose the account and role associated with your account, and click **Next**.
4. Select or enter the appropriate search criteria to find the custom objects that you want to import, and click **Search**.

SDF generates a list of custom objects that match your search criteria.

5. Select the custom objects that you want to import.

Optionally, under **Import Options**, uncheck the **Include the SuiteScript files referenced in the custom objects** box not to import the SuiteScript files referenced in the custom objects.

   **Note:** This option is only available for account customization projects (ACP).

6. Click **Finish**.

SDF imports the custom objects from your account, and places them in the **Objects** folder of the SDF project. If applicable, the referenced SuiteScript files that have been imported are placed in the **SuiteScripts** folder.

### Importing Files and Scripts into an Account Customization Project with SuiteCloud IDE Plug-in for WebStorm

If your account has the Web Site Hosting Files folder, you need to fulfill the following prerequisites to see it in the IDE plug-in:
- The **Host HTML Files** feature enabled.
- The **Website (External) publisher** permission.

1. In an account customization project, right-click the **FileCabinet** folder.
2. Select **NetSuite > Import Files from Account**.
3. Choose the account and role associated with your account, and then click **Get Files**.
4. Select the files or folders that you want to import, and click **OK**.

**Note:** You may be prompted to overwrite any files that already exist in the project.

SDF imports the files from the account and places them in subdirectories of the FileCabinet folder of the account customization project. For the list of subdirectories, see the File Cabinet Files section in **SDF Project Components**.

### Working with Custom NetSuite Objects in SuiteCloud IDE Plug-in for WebStorm

Custom NetSuite objects are created when XML representations in an SDF project are deployed to your NetSuite account.

You can customize NetSuite objects by downloading XML definitions of those objects from your account, importing them, or creating new representations. The XML files are converted into custom NetSuite objects when you deploy them to your NetSuite account.

See the following topics to learn more about working with custom objects using SDF:

- [Downloading the XML Definition of an Object from Your NetSuite Account](#)
- [Creating an XML Definition for a Custom Object in SuiteCloud IDE Plug-in for WebStorm](#)
- [Comparing a Custom SuiteCloud Object with a Custom NetSuite Object in SuiteCloud IDE Plug-in for WebStorm](#)
- [Comparing an SDF Project File with a File in SuiteCloud IDE Plug-in for WebStorm](#)

### Creating an XML Definition for a Custom Object in SuiteCloud IDE Plug-in for WebStorm

You create custom NetSuite objects by creating XML definitions of those objects. The XML files are converted into custom NetSuite objects when you deploy them to your NetSuite account.

Custom NetSuite object types must be stored in the Objects folder. By default, your deploy file is configured to deploy all custom NetSuite objects in the Objects folder. For more information, see the help topic **SDF Project Deployment Preparation**.

To understand how an XML definition in an SDF project is structured and which XML tags are available within nested elements, see the following topics:

- [Object File Structure](#)
- [Syntax Guidelines and Reference Formats](#)
- [Customizations Supported by SuiteCloud Development Framework](#)
- [SuiteCloud Development Framework XML Reference](#)
To create a custom object in SuiteCloud IDE plug-in for WebStorm

1. In WebStorm, open an SDF project.
2. Right-click the Objects folder and select New > Custom Object.
3. In the New Custom Object dialog, do the following:
   - From the Object Type dropdown list, select the type of object you want to create.
   - In the Script ID field, enter an ID for the object.
4. Click OK.

The custom NetSuite object is created based on the template for that object.

Overwriting Custom SuiteCloud Objects with Custom NetSuite Objects in SuiteCloud IDE Plug-in for WebStorm

You can overwrite your project objects with the objects from your file cabinet. Overwriting is only supported for a SuiteCloud object that shares the same script ID and is the same object type as the corresponding object in your NetSuite account.

Tip: For custom records, you can also overwrite the object together with its instances.

To overwrite custom SuiteCloud objects with custom NetSuite objects:

1. Open WebStorm, and create or select the SDF project.
2. Expand your project in the NS Explorer subtab.
3. Select the most appropriate option to overwrite your custom objects.
   - To overwrite specific custom SuiteCloud objects, expand the **Objects** folder, right-click each object and click **NetSuite > Update with Account Object**.

   **Tip:** For custom records, in the **Do you want to include all custom record instances?** dialog, click **Yes** to overwrite the object and all its instances.

   - To overwrite all custom SuiteCloud objects, right-click the **Objects** folder and click **NetSuite > Update All Objects with Account Objects**.

4. Choose the account and role associated with your account, and click **OK**.

5. Click **OK** to confirm that you want to perform the update.

   SDF connects to your NetSuite account and imports custom objects that share the same script IDs as the custom objects selected in your SDF project. The selected custom SuiteCloud objects from your project are overwritten by the custom objects from your file cabinet.

Comparing a Custom SuiteCloud Object with a Custom NetSuite Object in SuiteCloud IDE Plug-in for WebStorm

You can compare the XML definition of a custom SuiteCloud object with the XML equivalent of a custom NetSuite object. Objects can only be compared when they share the same script IDs and are the same object type.

**To compare a custom SuiteCloud object with a custom NetSuite object in SuiteCloud IDE Plug-in for WebStorm:**

1. In WebStorm, open an SDF project and expand the **Objects** folder.
2. Right-click on the object file and click **NetSuite > Compare Custom Object with Account Version**.
   
The side-by-side comparison window appears if an object with the same script ID exists in your NetSuite account. The differences between the two objects are highlighted.

Comparing an SDF Project File with a File in SuiteCloud IDE Plug-in for WebStorm

You can compare a file in the File Cabinet folder of your project with the equivalent account version. Files can only be compared when they share the same script IDs and are the same file type.

**To compare a SDF Project File with a File in SuiteCloud IDE Plug-in for WebStorm:**

1. From an SDF project in the WebStorm plug-in, expand the **FileCabinet** folder or one of its subfolders.
2. Right-click a file, and click **NetSuite > Compare File with Account Version**.
   
The side-by-side comparison window appears if a file with the same script ID exists in your NetSuite account. The differences between the two files are highlighted.

**Example**

The following shows the side-by-side comparison window of a custom client script file that exists in the SDF project and the NetSuite account:
Working with SuiteScript Files in SuiteCloud IDE Plug-in for WebStorm

See the following topics for information about working with SuiteScript files:

■ Creating a SuiteScript File in SuiteCloud IDE Plug-in for WebStorm
■ Uploading Files or Folders to Your Account with SuiteCloud IDE Plug-in for WebStorm

Creating a SuiteScript File in SuiteCloud IDE Plug-in for WebStorm

You can create SuiteScript files in IDE plug-in for WebStorm and the appropriate SuiteScript file type template is automatically added. Additionally, if you add SuiteScript modules when creating the file, you can use code completion for those modules.

To create a SuiteScript file in SuiteCloud IDE Plug-in for WebStorm

1. In WebStorm, open an SDF project.
2. Right-click the SuiteScripts or Web Site Hosting Files folder, and select New > SuiteScript File. If you are in a SuiteApp project, right-click the SuiteApps/<Application ID> or Web Site Hosting Files folder.

   **Note:** For more information about the Application ID, see the help topic SuiteApp Application ID.

3. In the New SuiteScript File dialog, do the following:
   ▪ From the Type dropdown list, select the type of SuiteScript file you want to create.
   ▪ In the Name field, enter a name for your SuiteScript file.
   ▪ From the Dependencies area, check the SuiteScript modules you want to include in your SuiteScript file.
4. Click OK.

The SuiteScript file is created based on the template for that file.
Uploading Files or Folders to Your Account with SuiteCloud IDE Plug-in for WebStorm

You can upload files or folders to the File Cabinet in your account, without requiring deployment of the entire project. The files or folders must be within a File Cabinet folder supported by SDF. The following folders are supported, depending on SDF project type:

- SuiteScripts
- SuiteApps/<Application ID>

**Note:** For more information about the Application ID, see the help topic SuiteApp Application ID.

- Templates
- Web Site Hosting Files

**To upload a file or folder from your SDF project to your account with SuiteCloud IDE plug-in for WebStorm:**

1. From the File Cabinet folder of your SDF project, do one of the following:
   - For multiple files, select NetSuite > Upload Files to Account.
   - Right-click a folder. Select NetSuite > Upload Folder to Account.
   - For multiple folders, select NetSuite > Upload Folders to Account.

If a file or folder with the same file cabinet location and name already exists in your account, it is overwritten by the SDF file upload.

If the folder containing the file did not exist in the account, SDF creates the folder in your account during the upload process.

Adding Dependencies to the Manifest in SuiteCloud IDE Plug-in for WebStorm

- Defining Dependencies on SuiteApp and Bundle Objects with SuiteCloud IDE Plug-in for WebStorm
- Defining Account Component Dependencies from an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm
- Defining Feature Dependencies in SuiteCloud IDE Plug-in for WebStorm

Defining Dependencies on SuiteApp and Bundle Objects with SuiteCloud IDE Plug-in for WebStorm

**SuiteCloud Development Framework: Referencing a SuiteBundle from Your SuiteApp Project**

You can define dependencies on SuiteApp objects that are external to your SDF project if they are custom NetSuite objects supported by SDF. For more information about custom object dependencies, see:

- SDF SuiteApp Object Dependencies
- Specifying an SDF SuiteApp Object Reference
- Specifying a Bundle Object Reference
If your project does not contain any references to the object dependency, you must define the dependency manually. See the help topic Manually Defining SDF SuiteApp Object Dependencies from the SDF Project Manifest.

To add SuiteApp object dependencies to the project manifest in SuiteCloud IDE Plug-in for WebStorm:

To complete this task, your project must already reference the SuiteApp object from an object definition in the Object folder. If your project does not contain references to the object, you can define the reference manually instead. See the help topics Manually Defining SDF SuiteApp Object Dependencies from the SDF Project Manifest and Manually Defining SDF SuiteApp Object Dependencies from the SDF Project Manifest.

For examples of a project manifest with SuiteApp object dependencies, see Example 1 and Example 2.

1. From the Project window in WebStorm, right-click the SDF project.
2. Select NetSuite > Add Dependency References to Manifest.

SDF evaluates the custom NetSuite objects that are referenced in the SDF project and adds any new dependencies to the manifest.xml file.

Example 1

The following example shows a SuiteApp project manifest that references an object called customrecord_sample that is not part of the project. During a server-side validation against a NetSuite account, SDF logs any referenced object that is missing from the account.

```xml
<manifest projecttype="SUITEAPP">
  <publisherid>com.example</publisherid>
  <projectid>mysuiteapp</projectid>
  <projectname>SDFSample</projectname>
  <projectversion>1.0.0</projectversion>
  <frameworkversion>2.0</frameworkversion>
  <dependencies>
    <applications>
      <application id="com.example.mysuiteapp">
        <objects>
          <object>customrecord_sample</object>
        </objects>
      </application>
    </applications>
  </dependencies>
</manifest>
```

Example 2

The following example shows a SuiteApp project manifest that uses references to depend on the following objects:

- customrecord_sample object from com.samples.mysuiteapp
- customrecord_sample2 object from bundle 351
- customrecord_sample3 object which may be found in a few bundles due to a bundle copying or deprecation

```xml
<manifest projecttype="SUITEAPP">
  <publisherid>com.example</publisherid>
  <projectid>mysuiteapp</projectid>
  <projectname>SDFSample</projectname>
  <projectversion>1.0.0</projectversion>
  <frameworkversion>2.0</frameworkversion>
  <dependencies>
    <applications>
      <application id="com.example.mysuiteapp">
        <objects>
          <object>customrecord_sample</object>
          <object>customrecord_sample2</object>
          <object>customrecord_sample3</object>
        </objects>
      </application>
    </applications>
  </dependencies>
</manifest>
```
Adding Dependencies to the Manifest in SuiteCloud IDE Plug-in for WebStorm

Defining Account Component Dependencies from an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm

You can define dependencies on existing custom NetSuite objects, files, and scripts from your NetSuite accounts in an account customization project. For more information, see:

- Account Component Dependencies
- Specifying a NetSuite Custom Object Reference

If your project does not contain any references to the dependency, you must define it manually. See the help topic Manually Defining Account Component Dependencies in the Account Customization Project Manifest.

To automatically define account component dependencies in SuiteCloud IDE Plug-in for WebStorm:

1. From the Project window in WebStorm, right-click the SDF project.
2. Select NetSuite > Add Dependency References to Manifest.

SDF evaluates the custom NetSuite objects, files, and scripts that are referenced in an account outside of the SDF project and adds any new dependencies to the manifest.xml file.

**Note:** This option does not remove dependencies that are not referenced by any objects.

Example

The following example shows an account customization project manifest that references objects and files that are not part of the project, such as customrecord_sample, customworkflow_sample, customlist, and
ext_UserScript.js. During a server-side validation against a NetSuite account, SDF logs any referenced object or file that is missing from the account.

```xml
<manifest projecttype="ACCOUNTCUSTOMIZATION">
  <projectname>SDFTutorial</projectname>
  <frameworkversion>1.0</frameworkversion>
  <dependencies>
    <features>
      <feature required="true">CUSTOMRECORDS</feature>
      <feature required="true">SERVERSIDESCRIPTING</feature>
    </features>
    <objects>
      <object>customrecord_sample</object>
      <object>customworkflow_sample</object>
      <object>customworkflow_sample.workflowstate_a</object>
      <object>customlist_sample.red</object>
      <object>customlist_sample.green</object>
      <object>customlist_sample.blue</object>
    </objects>
    <files>
      <file>/SuiteScript/reference/ext_UserScript.js</file>
    </files>
  </dependencies>
</manifest>
```

Defining Feature Dependencies in SuiteCloud IDE Plug-in for WebStorm

When you use SDF with NetSuite features, such as SuiteScript or custom records, you must define the dependencies in the SDF project.

For more information, see the help topic Feature Dependencies.

To automatically reference dependencies in the SDF project manifest in SuiteCloud IDE Plug-in for WebStorm:

1. From the Project window in WebStorm, right-click the SDF project.
2. Select NetSuite > Add Dependency References to Manifest.
   SDF evaluates the NetSuite feature requirements in the project and adds any new dependency references to the manifest.xml file.

   **Note:** This option does not remove dependencies that are not referenced by any objects.

3. Review the new dependency references in the manifest. If desired, change the required attribute.

Example

The following example shows a project manifest that references dependencies on the server-side SuiteScript and custom record features. The references enable you to use these features in the project.

```xml
<manifest projecttype="ACCOUNTCUSTOMIZATION">
  <projectname>SDFTutorial</projectname>
  <dependencies>
    <features>
      <feature required="true">CUSTOMRECORDS</feature>
      <feature required="true">SERVERSIDESCRIPTING</feature>
    </features>
    <objects>
      <object>customrecord_sample</object>
      <object>customworkflow_sample</object>
      <object>customworkflow_sample.workflowstate_a</object>
      <object>customlist_sample.red</object>
      <object>customlist_sample.green</object>
      <object>customlist_sample.blue</object>
    </objects>
  </dependencies>
</manifest>
```
Adding Dependencies to the Manifest in SuiteCloud IDE Plug-in for WebStorm

```
<frameworkversion>1.0</frameworkversion>
<dependencies>
  <features>
    <feature required="true">CUSTOMRECORDS</feature>
    <feature required="true">SERVERSIDEGRAPHIC</feature>
  </features>
</dependencies>
</manifest>
```

Changing Project Settings in SuiteCloud IDE Plug-in for WebStorm

You can set an account and role as default for a project so you use them for actions such as deploy and validate. By doing so, you do not need to specify an account and role every time you want to validate or deploy your project.

**To change project settings in SuiteCloud IDE plug-in for WebStorm**

1. In WebStorm, right-click a project, and select **NetSuite > Project Settings** to open the **Project Settings** window.
2. From the **Account** dropdown list, select the account you want to use as default.
3. From the **Role** dropdown list, select the role you want to use as default.
4. Click **OK**.

Setting Installation Preferences in SuiteCloud IDE Plug-in for WebStorm

To define installation preferences, you can create new preference files in the InstallationPreferences folder of your SuiteApp project. There are two types of preferences files: locking preferences and hiding preferences.

For more information, see the following topics:

- Installation Preferences in a SuiteApp Project
- Creating an Installation Preference File with SuiteCloud IDE Plug-in for WebStorm
Creating an Installation Preference File with SuiteCloud IDE Plug-in for WebStorm

To create an installation preference file with SuiteCloud IDE plug-in for WebStorm

1. Open a SuiteApp project in WebStorm.
2. Right-click on the InstallationPreferences folder and select either New > Locking File or New > Hiding File.
   
   The file is created inside of the InstallationPreferences folder.

For more information, see the following topics:

- Installation Preference File Structure
- Locking Custom Objects in an SDF SuiteApp Project
- Locking Files in an SDF SuiteApp Project

Locking Custom Objects in a SuiteApp Project

To use the locking preference, you must first create a locking.xml file in the InstallationPreferences Folder of a SuiteApp project. See Creating an Installation Preference File with SuiteCloud IDE Plug-in for WebStorm.

From the locking.xml file, you must include a valid preference type, default action, and the object. Supported values for an action are “LOCK” or “UNLOCK”.

For more information about installation preferences, see the help topics Locking Custom Objects in an SDF SuiteApp Project and Installation Preferences in a SuiteApp Project.

To lock a custom object:

1. Specify a value for defaultAction.
2. Specify a value for action.
3. Specify the script ID of the object. For the list of lockable object types, see the help topic Lockable Custom Objects Supported by SDF.
4. When you deploy the project, select the Apply Content Protection box to enable the specifications in the locking.xml file. See the help topic Using the Content Protection Option to Apply Installation Preferences.

Note that setting a value for default action can reduce the steps in applying a preference to many objects at a time. The default action applies to custom objects in your project that are not in the apply action list of your locking.xml file. In the following example, the installation locking preference for lockable custom objects in the SuiteApp project is UNLOCK, so all SuiteApp objects other than myobject12 are unlocked.
Example: Locking a Custom Object using SDF

For example, to lock the myobject12 object, in the locking.xml file, set the preference type to LOCKING, set the default action to UNLOCK, set the apply action to LOCK, and add the object:

```xml
<preference type="LOCKING" defaultAction="UNLOCK">
  <apply action="LOCK">
    <object>myobject12</object> <!-- lock an object -->
  </apply>
</preference>
```

Hiding Files in a SuiteApp Project

To use the hiding preference, you must first create a hiding preferences file in the InstallationPreferences Folder of a SuiteApp project. See Creating an Installation Preference File with SuiteCloud IDE Plug-in for WebStorm.

Supported values for an action or default action are "HIDE" or "UNHIDE".

For more information, see the help topics Hiding Files in an SDF SuiteApp Project and Installation Preferences in a SuiteApp Project.

To hide files:

Note: A path element with a value that contains a wildcard requires an appliesTo attribute.

1. From the hiding.xml file, specify the preference type.
2. Specify the default action.
3. Specify the apply action.
4. Specify the file path.
5. (Optional) To use a wildcard path, specify a file path for the appliesTo attribute. The appliesTo attribute value must be "FILES".
6. When you deploy the project, select the Apply Content Protection box to enable the specifications in the hiding.xml file. See the help topic Using the Content Protection Option to Apply Installation Preferences.

Example: Hiding a file using SDF

For example, to hide the a.js file, set the preference type to HIDING, set the apply action to HIDE, and add the file paths to the hiding.xml file.

The first file path sets a preference to hide a single file. The second file path applies the preferences to files within the folder specified in the path.

The default action applies to content in your project that is not in the apply list of your hiding.xml file. In the example above, the default preference for files in the SuiteApp project is UNHIDE, so all files except files in the a.js file and files in the myFile folder are not hidden.

```xml
<preference type="HIDING" defaultAction="UNHIDE">
  <apply action="HIDE">
    <path>~/FileCabinet/SuiteApps/com.netsuite.myFile/a.js</path> <!-- hide a file -->
    <path appliesTo="FILES">~/FileCabinet/SuiteApps/com.netsuite.myFiles/*</path> <!-- hide all the files in my application -->
  </apply>
</preference>
```
Using SuiteScript Code Completion in SuiteCloud IDE Plug-in for WebStorm

Code completion works by examining the source code around the cursor in the editor. Depending on where the caret is positioned, a popup displays with SuiteScript (and also JavaScript) text options to insert at the cursor.

When you create a new SDF project, both 1.0 and 2.0 versions of SuiteScript APIs are added as external libraries to the SDF project. The IDE plug-in can read the SuiteScript 1.0 and 2.0 External Libraries to help check SuiteScript syntax and assist with entering functions, methods, objects, and references.

For SuiteScript 1.0, code completion works natively. For SuiteScript 2.0, when you create a SuiteScript file and add SuiteScript modules, code completion is available for those modules through JSDoc annotation tags. If you do not add any SuiteScript module upon the file creation, code completion is not available automatically. Similarly, if you add more SuiteScript modules after the file has been created, code completion is not available.

Note: Code completion of parameters, such as record types, is not supported.

If you want to have code completion for modules you add after creating a SuiteScript file, you need to add JSDoc annotation tags manually. Then, you need to add the module names in the SuiteScript file.

To enable code completion for SuiteScript 2.0 in SuiteCloud IDE plug-in for WebStorm

1. In the define function of your SuiteScript file, add a dependency such as `['N/config']`. 
Important: To ensure you use the right name of the modules, see the help topic SuiteScript 2.0 Modules.

2. In the `define` function, add the module name as a JSDoc annotation tag. Use the same module name as in the previous step, but without the `N/` prefix.

3. In the `function` argument, add the module name without the `N/` prefix.
Using SuiteScript Code Completion in SuiteCloud IDE Plug-in for WebStorm

For more information, see the following topics:

- SuiteScript 2.0 JSDoc Validation
- define Object
- If you do not want to use code completion, see Turning Off Code Completion in SuiteCloud IDE Plug-in for WebStorm.

Validating an SDF Project with SuiteCloud IDE Plug-in for WebStorm

SDF can run server-side validation to catch errors that may occur during the project deployment process.

For more information, see the help topic SDF Project Validation.

To validate an SDF project against your account with SuiteCloud IDE plug-in for WebStorm:

1. In WebStorm, create or select the SDF project.
   For more information, see Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm or Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm.
2. Right-click on an SDF project in the Project window.
4. Choose the account and role associated with your account.
5. Select Validate.
   SDF validates your project against the chosen account. The validation progress log is displayed in the SuiteCloud IDE console.
Errors in the validation log are displayed in red. If you encounter errors during server-side validation, review the error messages to determine possible causes and troubleshoot the issues.

Example

The following example shows a validation log that indicates a successful validation:

```
Info -- Account [ACCOUNTNAME]
Info -- Account Customization Project [PROJECTNAME]
Info -- Framework Version [1.0]
Validate manifest -- Success
Validate deploy file -- Success
Validate configuration -- Success
Validate script file -- Success
Validate objects -- Success
Validate files -- Success
Validate folders -- Success
Validate account settings -- Success
Validate Custom Objects against the Account -- Success
Validation COMPLETE
Validate preferences - Success
Validate flags - Success
```

Deploying an SDF Project to Your NetSuite Account With SuiteCloud IDE Plug-in for WebStorm

When you create an SDF project, a deploy file is automatically generated. By default, the file is configured to deploy all supported files in the FileCabinet folder and all custom NetSuite objects in the Objects folder.

For more information, see the help topic SDF Project Deployment Preparation.

To deploy an SDF project to your NetSuite account With SuiteCloud IDE Plug-in for WebStorm:

1. In WebStorm, create or select the SDF project. For more information, see Creating an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm or Creating a SuiteApp Project in SuiteCloud IDE Plug-in for WebStorm.

   **Important:** Account customization projects cannot be uninstalled from your account after they have been deployed. Undesired account components need to be manually deleted.

2. Ensure that your project references all the required dependencies. For more information, see Defining Feature Dependencies in SuiteCloud IDE Plug-in for WebStorm, Defining Dependencies on SuiteApp and Bundle Objects with SuiteCloud IDE Plug-in for WebStorm, and Defining Account Component Dependencies from an Account Customization Project in SuiteCloud IDE Plug-in for WebStorm.

3. Ensure that your deploy file references all the NetSuite components that you want to deploy. For more information, see the help topic SDF Project Deployment Preparation.

4. Right-click on an SDF project in the Project window.
5. Select one of the following:
   - NetSuite > Deploy
   - NetSuite > Deploy to Account

6. If you chose to deploy to an account, select the appropriate account and role, and then click Deploy.

7. If you are deploying a SuiteApp project, to enable or disable the installation preferences, select or clear the Apply Content Protection box.
   For more information, see the help topic Installation Preferences in a SuiteApp Project.

8. If you are deploying to a NetSuite production account, a notification window may appear. Confirm that you want to deploy to the production account to continue the deployment process.

   SDF validates and deploys the SDF project to your NetSuite account. The installation log is displayed in the SuiteCloud IDE tool window.

Example

The following example shows the installation log of a successful deployment to a NetSuite production account for a company named Wolfe Electronics:

```
2018-08-09 06:05:10 (PST) Installation started
Info -- Account [(PRODUCTION) Wolfe Electronics (ACCOUNTNAME)]
Info -- Account Customization Project [SDFtutorial]
Info -- Framework Version [1.0]
Validate manifest -- Success
Validate deploy file -- Success
Validate configuration -- Success
Validate script file -- Success
Validate objects -- Success
Validate files -- Success
Validate folders -- Success
Validate account settings -- Success
Validate Custom Objects against the Account -- Success
Begin deployment
Upload file -- ~/FileCabinet/SuiteScripts/UserEventScript.js
Create object -- customrecord_tutorial (customrecordtype)
Create object -- customscript_userevent (usereventscript)
Create object -- customrecord_tutorial.custrecord_tut_memo (customrecordcustomfield)
Create object -- customscript_userevent.customdeploy_ue_tut (scriptdeployment)
2018-08-09 06:05:37 (PST) Installation COMPLETE (0 minutes 27 seconds)
```

Shortcut Keys in SuiteCloud IDE Plug-in for WebStorm

You can use keymaps to trigger the following SDF functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Key Combination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Dependency References to Manifest</td>
<td>Ctrl+Alt+Shift+A</td>
</tr>
<tr>
<td>Function</td>
<td>Windows</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Authenticate Master Password</td>
<td>Ctrl+Alt+A</td>
</tr>
<tr>
<td>Compare with Account File</td>
<td>Alt+Shift+E</td>
</tr>
<tr>
<td>Deploy</td>
<td>Ctrl+Alt+D</td>
</tr>
<tr>
<td>Deploy to Account...</td>
<td>Ctrl+Alt+Shift+D</td>
</tr>
<tr>
<td>Import Files from Account...</td>
<td>Ctrl+Alt+F</td>
</tr>
<tr>
<td>Import Objects from Account...</td>
<td>Ctrl+Alt+O</td>
</tr>
<tr>
<td>Open Project Settings</td>
<td>Ctrl+Alt+P</td>
</tr>
<tr>
<td>Revoke Master Password</td>
<td>Ctrl+Alt+R</td>
</tr>
<tr>
<td>Upload File(s) to Account</td>
<td>Alt+Shift+U</td>
</tr>
<tr>
<td>Update with Account Object(s)</td>
<td>Ctrl+U</td>
</tr>
<tr>
<td>Validate</td>
<td>Ctrl+Alt+V</td>
</tr>
<tr>
<td>Validate to Account...</td>
<td>Ctrl+Alt+Shift+V</td>
</tr>
</tbody>
</table>