

Create and manage content in IntelliJ

Docs-as-Code

Exported on 08/31/2021

Table of Contents

1	Project end-to-end workflow	4
2	Create content	6
2.1	Step 1: Pull files from the repository	7
2.2	Step 2. Create a feature branch	10
2.3	Step 3: Create a new project.....	12
2.3.1	A. Create a directory	12
2.3.2	B. Create an images folder and add images	21
2.3.3	C. Copy and paste index, topic templates	24
2.3.4	D. Set attributes in _index.adoc file	32
2.3.5	E. Rename topic file	53
2.3.6	F. Copy or type text in topic files	65
2.3.7	G. Create the table of contents	79
2.4	(Optional) Step 4: Import topics into IntelliJ	98
2.5	Step 5: Commit and push files to the repository.....	100
2.6	Step 6: Create a pull request	108
2.7	(Optional) Step 7: Make requested changes to a document	111
2.7.1	Scenarios:.....	116
3	Update content	118
3.1	Step 1: Pull files from repository	119
3.2	Step 2: Create a feature branch.....	122
3.3	(Optional) Step 3: Import topics into IntelliJ	124
3.4	(Optional) Step 4: Delete a topic	126
3.5	(Optional) Step 5: Add a topic.....	129
3.6	(Optional) Step 6: Edit a topic.....	136
3.7	Step 7: Commit and push files to the repository.....	144
3.8	Step 8: Create a pull request	152
3.9	(Optional) Step 9: Make requested changes to a document.....	155
3.9.1	Scenarios:.....	160
4	Review content	162
4.1	Step 1: Go to the pull request assigned to you	163

4.2	Step 2: Review content.....	164
4.2.1	1. View a topic's changes	164
4.2.1.1	View a topic in WYSIWYG.....	166
4.2.2	2. Add a comment.....	166
4.2.2.1	Comment on a topic file.....	166
4.2.2.2	Comment on a single line of content/code.....	167
4.2.3	3. (Optional) Suggest a change to the content.....	168
4.2.3.1	Apply a suggested change to the content.....	169
4.2.4	Nice to Know: Filter or search for a topic.....	170
4.3	Step 3: Complete the review	172
4.3.1	Needs work	172
4.3.2	Approve	172
4.4	Manage multiple reviewers.....	174
5	Manage multiple authors or guides	176
5.1	Manage multiple authors	177
5.2	Manage multiple guides	178
6	Manage IntelliJ branches.....	184
6.1	Remove local branches	185
6.2	Work on an existing feature branch	186
6.3	Work on the same local branch.....	189
6.3.1	Scenarios:.....	189

Project instructions

The following sections provide step-by-step instructions for creating content, updating content, or reviewing content created by others.

Additional resources

See the following resources for details about authoring, reviewing and formatting content; AsciiDoc language; and Git tutorials.

- AsciiDoc formatting
- AsciiDoc cheatsheet
- AsciiDoctor User Manual
- Atlassian git tutorials

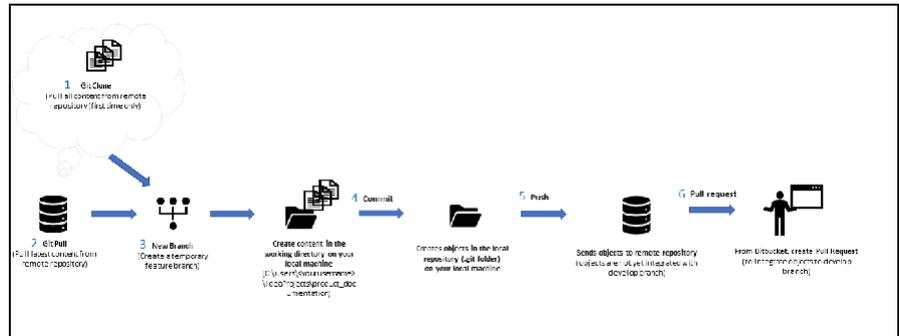
Get help

Join the Docs-as-Code platform Workplace group to share your experience with the tools and methods and seek guidance. You can also review the following pages for help:

- Quick reference
- FAQ
- Troubleshooting

1 Project end-to-end workflow

Project files for documents are stored in the Product Documentation repository, which is hosted by Atlassian's Bitbucket. The following diagram explains the end-to-end workflow and terminology used for creating and manage documentation projects. See Docs-as-Code software platform for additional details.



1. **Git Clone:** The first time you work with files, you must *clone* the Product Documentation repository. All the content from the remote repository is copied to your local machine.

☐ You only have to clone once. You do not have to do a Git Pull immediately following a clone.

2. **Git Pull:** Every time you start work, you must *pull* (retrieve) the latest files from the Product Documentation repository. This ensures you get all the latest changes and the files on your local machine are in sync with the remote repository.
3. **New Branch:** Before creating or updating files, you *must* create a *feature branch*. A feature branch is a temporary container that keeps your work separate from everyone else's.
4. **Commit:** When you are done making changes, you must commit those changes. This moves your content from your working directory to your local repository.

☐ If you are familiar with Git, this next step is unorthodox. However, when using IntelliJ, we have found that after committing, it is best to pull changes from develop to the feature branch. If there are no updates, you can continue to push. If there are changes, you must amend your commit message. You can find more details about this in Step F on this page: [Step 5: Commit and push files to the repository](#) (see page 100).

5. **Push:** After committing your content to your local repository, you must then send it to the remote repository. To do this, you *push* your changes to the Product Documentation repository.

6. **Pull (review) request:** After your changes have been pushed, you must request that your changes are added to the main branch where all the integrated work is stored (called develop). Go to the Bitbucket URL in your web browser to create a *pull request* to ask that your changes be reviewed. You will specify the feature branch to let the reviewer know what they must review. After the reviewer approves the pull request, a moderator merges the feature branch to develop and the feature branch is deleted from the remote repository.

2 Create content

Use the following steps to create content such as a new guide, topic, site help, or online help.



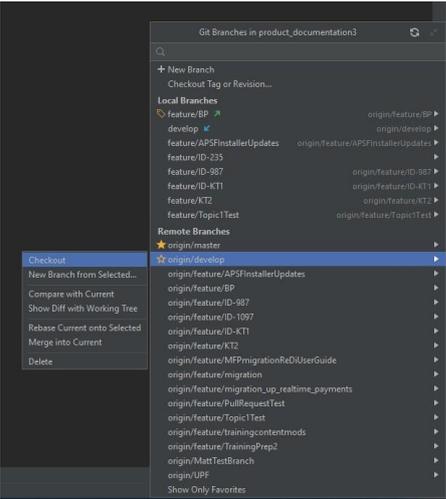
- For steps about editing or adding content to an existing project, see [Update content](#)(see page 118).
- For steps about reviewing content written by someone else, see [Review content](#)(see page 162).
- For guidelines about managing a deliverable written by multiple authors, see [Manage multiple authors](#)(see page 177).

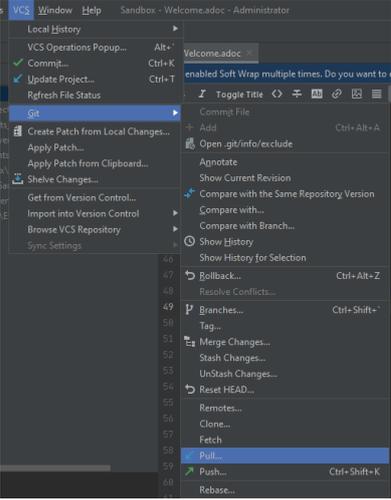
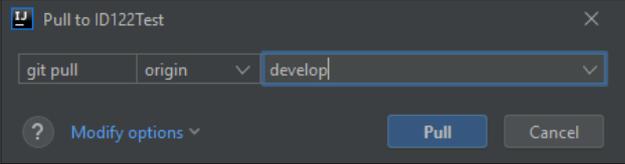
Search

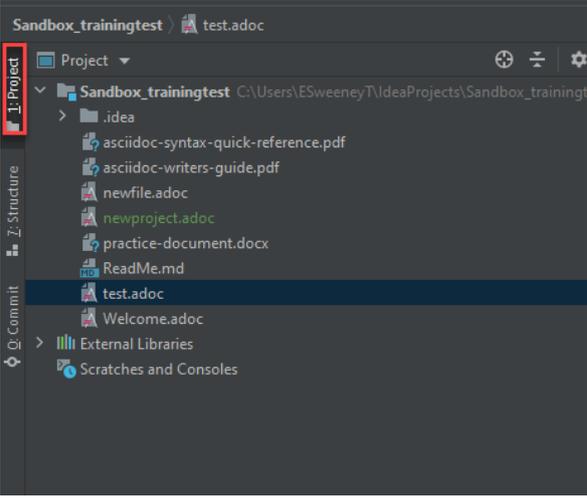
2.1 Step 1: Pull files from the repository

Before editing content, you must retrieve (or "pull") the latest files from the remote repository. This ensures that you have the latest version of the remote repository, which includes all content changes made by others.

☐ *If you recently cloned the repository, you don't need to pull the latest files. Skip to **Step 2: Create feature branch**.*

Steps	Additional Information
<p>A. Open IntelliJ.</p> <p>Click Start > JetBrains > IntelliJ IDEA Community Edition.</p>	
<p>B. If you are not already on the develop branch, change your branch to develop.</p> <p>Click on the branch in the bottom right corner. When it expands, under Remote Branches, select origin/develop > Checkout.</p>	<p>Example</p> 

Steps	Additional Information
<p>C. Go to Git > Pull.</p>	<p>Example</p>  <p>The screenshot shows the 'VCS' menu in IntelliJ IDEA. The 'Git' sub-menu is open, and the 'Pull...' option is highlighted. Other visible options include 'Commit...', 'Update Project...', 'Refresh File Status', 'Create Patch from Local Changes...', 'Apply Patch...', 'Apply Patch from Clipboard...', 'Shelve Changes...', 'Get from Version Control...', 'Import into Version Control', 'Browse VCS Repository', 'Sync Settings', 'Commit File', 'Add', 'Open .git/info/exclude', 'Annotate', 'Show Current Revision', 'Compare with the Same Repository Version', 'Compare with...', 'Compare with Branch...', 'Show History', 'Show History for Selection', 'Rollback...', 'Resolve Conflicts...', 'Branches...', 'Tag...', 'Merge Changes...', 'Stash Changes...', 'UnStash Changes...', 'Reset HEAD...', 'Remotes...', 'Clone...', 'Fetch', 'Push...', and 'Rebase...'.</p>
<p>D. Set the repository to the develop branch so you pull all the content from the remote repository.</p> <div data-bbox="172 1189 536 1296" style="border: 1px solid red; padding: 5px; margin: 10px 0;"> <p><input type="checkbox"/> Do not pull from another branch.</p> </div> <div data-bbox="172 1364 536 1538" style="border: 1px solid yellow; padding: 5px; margin: 10px 0;"> <p><input type="checkbox"/> You cannot pull specific content from the repository. It must be the entire repository.</p> </div>	<p>Example</p>  <p>The screenshot shows the 'Pull to ID122Test' dialog box. The 'git pull' command is entered in the text field. The 'origin' dropdown is selected, and the 'develop' branch is chosen from the dropdown menu. There are 'Modify options', 'Pull', and 'Cancel' buttons at the bottom.</p>

Steps	Additional Information
<p>E. Click Pull. The latest files are shown on the Project tab.</p>	<p>Example</p>  <p>The screenshot shows the IntelliJ IDEA interface with the Project tool window open. The window title is 'Sandbox_trainingtest' and it shows a file tree for the project 'Sandbox_trainingtest'. The files listed are: .idea, asciidoc-syntax-quick-reference.pdf, asciidoc-writers-guide.pdf, newfile.adoc, newproject.adoc, practice-document.docx, ReadMe.md, test.adoc (highlighted), and Welcome.adoc. The 'Project' tab is selected in the top-left corner of the tool window.</p>

2.2 Step 2. Create a feature branch

A feature branch is a temporary workspace where you create or update content without updating any content in the ID repository. The feature branch is created on your local machine and stores your changes until they are merged into the develop branch on the ID (remote) repository.

Required prerequisite

Before creating a feature branch, ensure you have a Jira for your work. You will use the Jira number when naming your feature branch.
The Jira does not have to be an ID Jira. It can be any Jira, like XXX-12345 or ABCD-12345.

Steps

Additional Information

A. Create a Jira that is assigned to your work. If you already have a Jira, skip to **Step B**.

Best practice

You **must** use a Jira number when naming your feature branch.

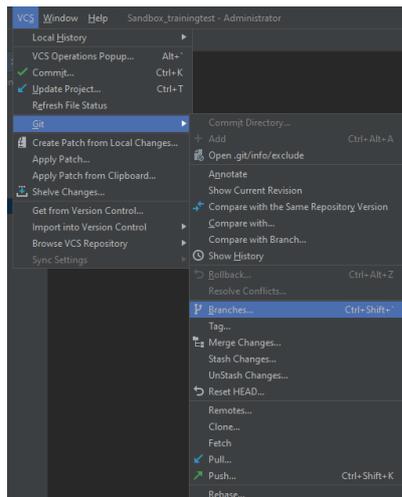
- If you do not have a Jira, you must create one.
- If your updates are the result of multiple Jiras, you will name the branch using one Jira number, and include all Jira numbers in the commit message.

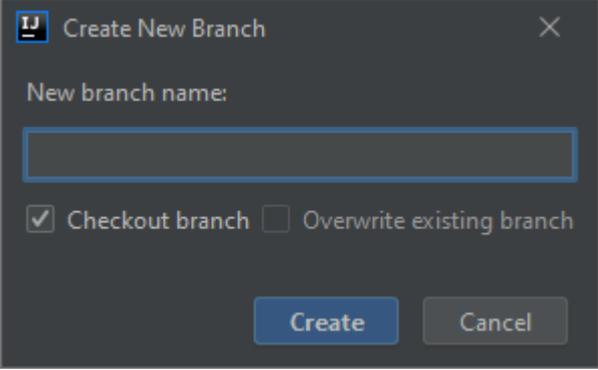
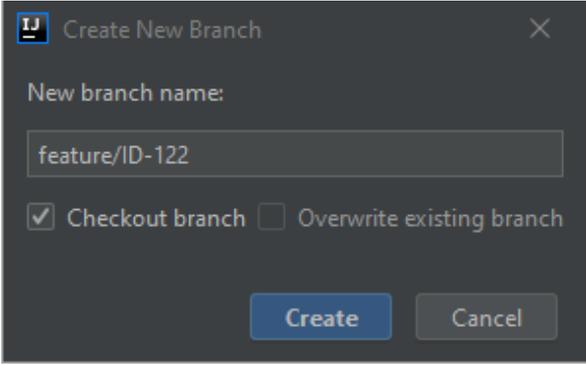
You **must** include the Jira number in the commit message and it must be in square brackets like [XXX-12345].

B. In IntelliJ, go to the **Git** menu at the top of the window and select **Branches**.

Example

If you have an older version of IntelliJ, the menu might be **VCS > Git > Branches**. See "Current recommended versions" on Install and configure Docs-as-Code platform and upgrade IntelliJ IDEA and the AsciiDoc Plugin.

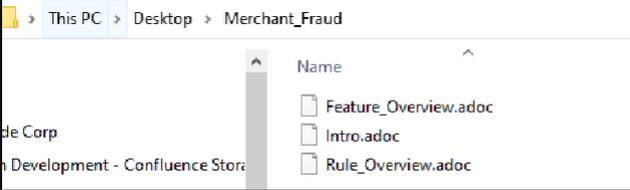
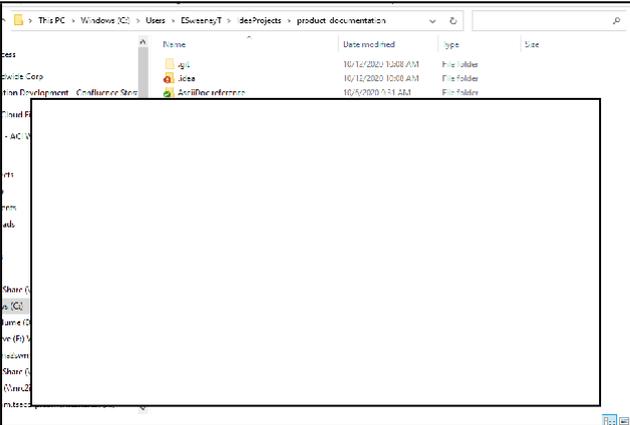


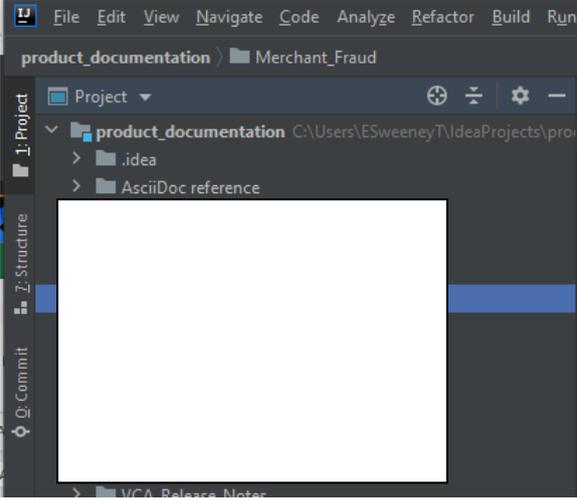
Steps	Additional Information
<p>C. Click + New Branch.</p>	<p>Example</p> 
<p>D. Type a name for the branch.</p> <div data-bbox="172 965 536 1361" style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Best practice Use "feature/" and the Jira number such as:</p> <p>feature/ID-122 If you don't use this naming convention, you will get an error message when pushing your changes to the remote repository.</p> </div> <div data-bbox="172 1429 536 1727" style="border: 1px solid #ccc; padding: 5px; margin: 5px 0; background-color: #fff9c4;"> <p>When you type feature/, you may see an error message that the branch name is invalid. This error will disappear when you add the Jira number.</p> </div>	<p>Example</p> <div data-bbox="616 958 1238 1205" style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Best practice You must use a Jira number when naming your feature branch, for example, feature/ID-123.</p> <ul style="list-style-type: none"> If you do not have a Jira, you must create one. </div> <p>Example</p> 
<p>E. Click Create. You can see the branch that is checked out at the bottom of the page and in the bottom right corner.</p>	<p>Example</p> 

<<REDACTED INFORMATION>>

2.4 (Optional) Step 4: Import topics into IntelliJ

If you have already created the .adoc files in a text editor, you can import these files into IntelliJ.

Steps	Additional Information
<p>A. In Windows File Explorer, go to the location where the IntelliJ files are stored.</p>	<p>Example</p> 
<p>B. Copy the folder from another location in Windows File Explorer to the location where the IntelliJ files are stored. In this example, the / MF folder is copied to the /product_documentation folder.</p> <div data-bbox="173 1350 536 1525" style="border: 1px solid red; padding: 5px; margin-top: 10px;"> <p><input type="checkbox"/> Make sure there is no .idea or .git folder in the originating folder.</p> </div>	<p>Example</p>  <p>Example</p> 

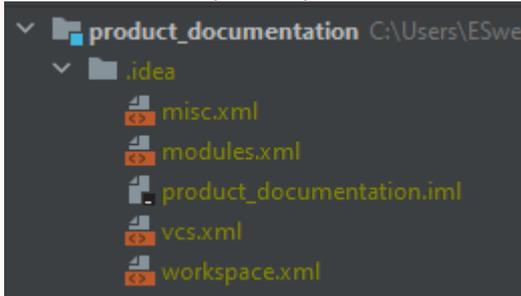
Steps	Additional Information
<p>C. Open IntelliJ and you will see the folder that you added. The files are in red, indicating that they have not yet been checked into Git.</p>	<p>Example</p>  <p>The screenshot shows the IntelliJ IDEA interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, and Run. The breadcrumb path is 'product_documentation > Merchant_Fraud'. The Project tool window on the left shows the project structure: 'product_documentation' (C:\Users\ESweeneyT\IdeaProjects\pro...), '.idea', and 'AsciiDoc reference'. The files under 'product_documentation' are highlighted in red, indicating they are not yet checked into Git. The Structure tool window is also visible, showing a large empty white area.</p>

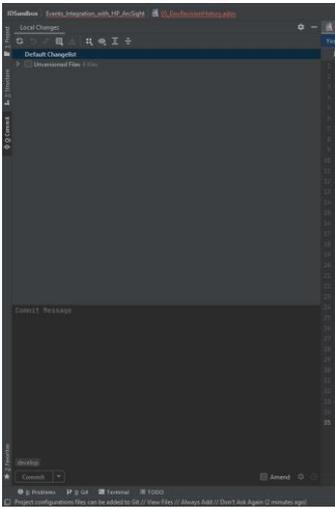
2.5 Step 5: Commit and push files to the repository

After creating your content, you must commit and push it to the ID repository. By committing and pushing, you're transferring the branch you created to the ID repository. After it's pushed, your content will be reviewed and then merged into the ID (remote) repository.

Use the following steps to commit and push your content.

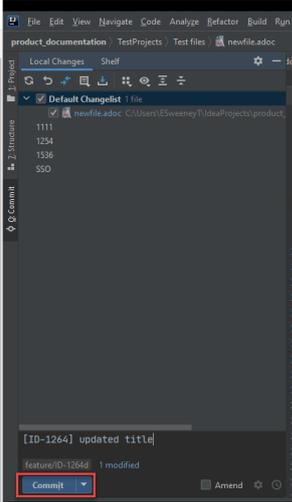
- ❑ Never commit and push any files that are in the `.idea` folder to the repository. These are user-specific files.

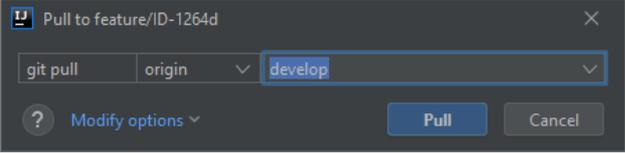


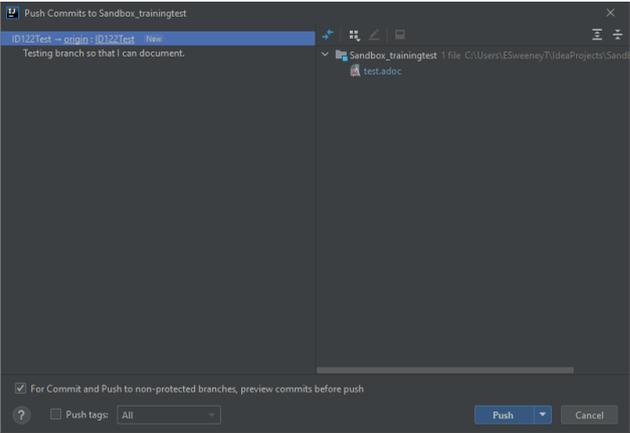
Steps	Additional Information
<p>A. Go to <code>Git > Commit</code>. This left panel opens.</p>	<p>Example</p>  <p>The screenshot shows the 'Default Commit' dialog box in IntelliJ. The dialog has a title bar 'Default Commit' and a main area with a 'Commit Message' field. Below the message field, there are buttons for 'Commit', 'Amend', and 'Cancel'. The background shows the IDE interface with a dark theme.</p>
<p>B. Expand <code>Unversioned documents</code>.</p>	

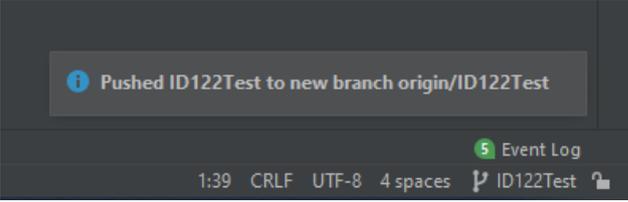
Steps	Additional Information
<p>C. Select the documents that you modified and drag them to the Default Changelist. Then, select the files to commit.</p> <p>Alternative: You can also right-click the files and select Move to Another Changelist. Then, in the window, ensure the Default Changelist is selected in the Name list and click OK. Then, select the files to commit.</p> <div data-bbox="172 813 536 1081" style="border: 1px solid #f0e68c; padding: 10px;"><p><input type="checkbox"/> Do NOT include any documents in the <code>.idea</code> folder in the changelist. These folders should not be checked into Git as they are user-specific files.</p></div>	

Steps	Additional Information				
<p data-bbox="169 421 504 479">D. Type a commit message in the bottom panel.</p> <div data-bbox="169 555 536 1798" style="border: 1px solid #ccc; padding: 10px;"><p data-bbox="201 577 405 609">Best practice</p><p data-bbox="245 613 520 824">This message must include only the Jira number in square brackets. If this is not correct, the push to the remote repository will fail.</p><p data-bbox="245 833 520 1527">If the feature branch is named with the Jira number and has a letter or word appended, do NOT include the appended text in the commit message. Note: As you learn more about using IntelliJ and Git, you might encounter scenarios where you need a "clean" branch to commit your changes. This is a use case for when you would add a letter to the Jira number. Do not use letters in your feature branches unless you understand the basics.</p><table border="1" data-bbox="245 1536 510 1798"><tr><td style="width: 50px; height: 40px;"></td><td style="width: 50px; height: 40px;"></td></tr><tr><td style="width: 50px; height: 40px;"></td><td style="width: 50px; height: 40px;"></td></tr></table></div>					<p data-bbox="568 421 676 452">Example</p> <pre data-bbox="616 465 1241 564">[ID-122] update column sizes in Facility codes table</pre>

Steps	Additional Information				
<table border="1" data-bbox="248 405 512 663"> <thead> <tr> <th data-bbox="248 405 371 501">Feature Name</th> <th data-bbox="371 405 512 501">Commit message</th> </tr> </thead> <tbody> <tr> <td data-bbox="248 501 371 663">feature / ID-1234</td> <td data-bbox="371 501 512 663">[ID-1234] Your commit message</td> </tr> </tbody> </table> <p data-bbox="248 663 512 815">The message should be meaningful as it will be used to describe the changes to reviewers.</p>	Feature Name	Commit message	feature / ID-1234	[ID-1234] Your commit message	
Feature Name	Commit message				
feature / ID-1234	[ID-1234] Your commit message				
<p data-bbox="169 860 504 920">E. At the bottom of the page, click Commit.</p> <p data-bbox="169 943 539 1032">The files that you changed are moved from a working directory to your local repository.</p>	<p data-bbox="571 860 676 889">Example</p> 				

Steps	Additional Information
<p>F. Go to Git > Pull and click Pull.</p> <p>If there are any conflicts, the Conflicts window opens. Click Merge. See Merge issues (conflicts) for details.</p> <div style="border: 1px solid gray; padding: 10px; margin-top: 10px;"> <p>❑ Avoiding push issues:</p> <ol style="list-style-type: none"> 1. After your pull, if you see a message that says "X files were updated in X commits" in the bottom right corner, this means that commits were merged from the develop branch. 2. Go to the Terminal window (click the tab at the bottom of the page), and type: <code>git commit --amend</code> 3. Notepad++ opens. 4. At the beginning of the file, add the Jira number in square brackets, as follows: Example  5. Close Notepad++. 6. Continue with Step G. </div>	<p>Example</p> 

Steps	Additional Information
<p>G. Go to Git > Push. If you see additional information on the left, it is log information.</p>	<p>Example</p>  <p>The screenshot shows a dialog box titled "Push Commits to Sandbox_trainingtest". On the left, there is a commit message "Testing branch so that I can document." and a branch indicator "ID122Test -- origin : ID122Test New". On the right, there is a file tree showing "Sandbox_trainingtest" with a sub-entry "test.doc". At the bottom, there is a checkbox "For Commit and Push to non-protected branches, preview commits before push" which is checked. There is also a "Push tags:" dropdown menu set to "All" and "Push" and "Cancel" buttons.</p>

Steps	Additional Information
<p>H. Confirm that the feature branch is the one that you created (for example, ID-122) and click Push.</p> <p>A message is shown in the bottom left to indicate if your push was successful.</p> <p>If it was not, you can click the Event Log to see the error. See Troubleshooting for any issues.</p> <div data-bbox="172 824 536 1865" style="border: 1px solid #ccc; padding: 10px;"><p>In some cases, your push might fail if Git merged files. You will see a message like the following EXTRACT:</p><pre>remote: To see the commit message: remote: remote: git show - s f6fa197483c87b22 remote: remote: To edit the commit message: remote: remote: git commit --amend</pre><p>You can fix this quickly by copying the line: <code>git commit--amend</code> and pasting it into the Terminal window (click the tab at the bottom of the page). Then, press Enter. When Notepad++ opens, on the first line of the file, add the Jira number in square brackets like: Example</p></div>	<p>Example</p> 

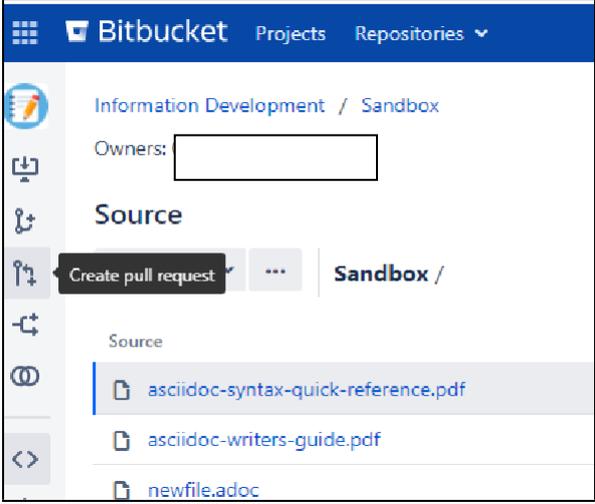
Steps	Additional Information
 <p>Close Notepad++ and push again. See Push issues if you need more details.</p>	
<p>I. Go to File > Close Project.</p> <p>Best practice Always close your project when you are done working.</p>	

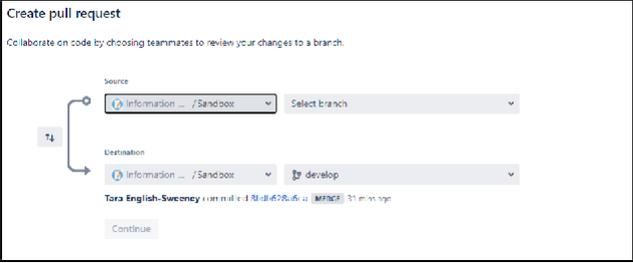
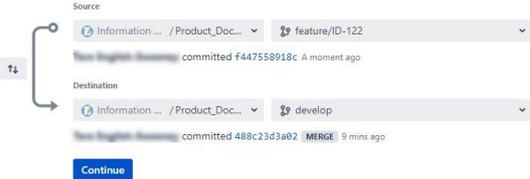
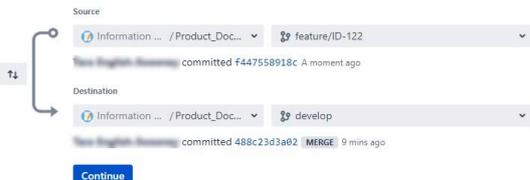
2.6 Step 6: Create a pull request

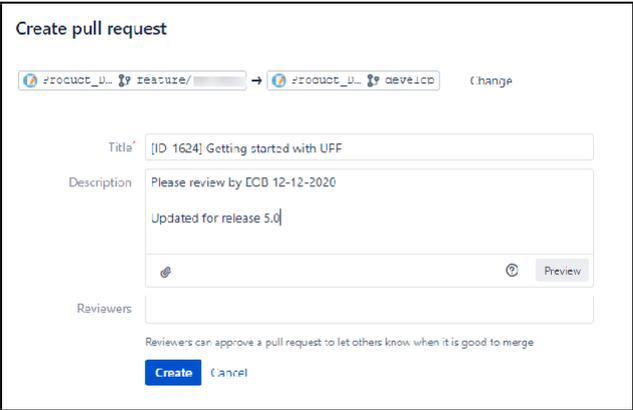
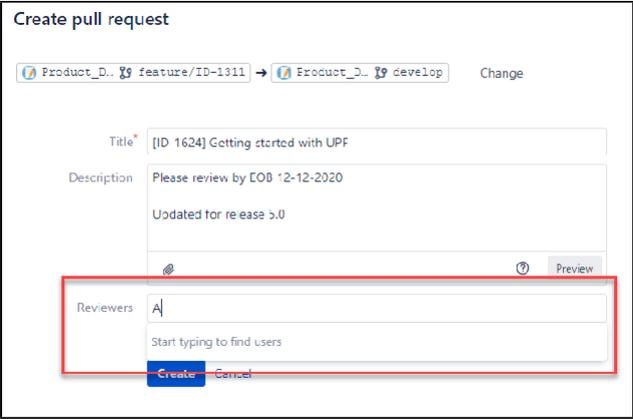
Use a Pull Request (PR) to submit your contributions to a project.

After you submit the request, reviewers (such as IDPMs or SMEs) will review and approve the changes or request clarifications or modifications. After all reviewers approve the Pull Request, a moderator merges it into the develop branch and the changes will be available to everyone who pulls the content from the repository.

For guidelines on managing your content review through multiple reviewers, see [Manage multiple reviewers](#)(see page 174).

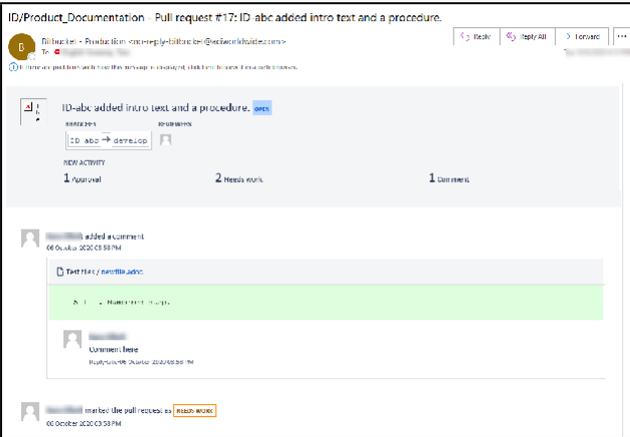
Steps	Additional Information
<p>A. Go to the repository, such as <code>https://</code></p>	
<p>B. Click Create pull request.</p>	<p>Example</p> 

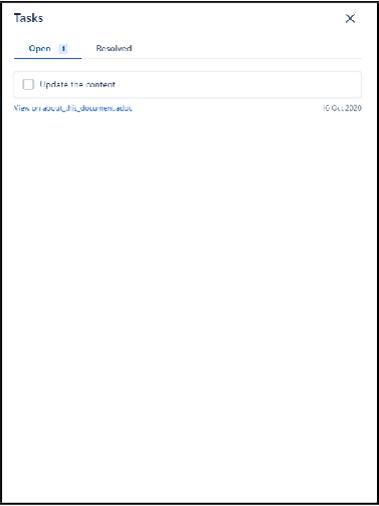
Steps	Additional Information
<p>C. After clicking Create pull request, the page in the example opens.</p>	<p>Example</p>  <p>The screenshot shows the 'Create pull request' dialog. The 'Source' dropdown is set to 'Information ... / Sandbox' and the 'Select branch' dropdown is empty. The 'Destination' dropdown is set to 'develop'. Below the dropdowns, a commit by 'Tara English-Sweeney' is shown with a commit hash 'f447558918c' and a 'MERGE' icon. A 'Continue' button is at the bottom.</p>
<p>D. From the Source Select branch list, select the branch that you created previously such as ID-122.</p>	<p>Example</p>  <p>The screenshot shows the 'Create pull request' dialog. The 'Source' dropdown is set to 'Information ... / Product_Doc...' and the 'Select branch' dropdown is set to 'feature/ID-122'. The 'Destination' dropdown is set to 'develop'. Below the dropdowns, a commit by 'Tara English-Sweeney' is shown with a commit hash 'f447558918c' and the text 'A moment ago'. A 'Continue' button is at the bottom.</p>
<p>E. From the Destination list, select develop.</p>	<p>Example</p>  <p>The screenshot shows the 'Create pull request' dialog. The 'Source' dropdown is set to 'Information ... / Product_Doc...' and the 'Select branch' dropdown is set to 'feature/ID-122'. The 'Destination' dropdown is set to 'develop'. Below the dropdowns, a commit by 'Tara English-Sweeney' is shown with a commit hash '488c23d3a02' and the text 'MERGE 9 mins ago'. A 'Continue' button is at the bottom.</p>

Steps	Additional Information
<p>F. Click Continue.</p> <p>In the Title, include a description that will be helpful to your reviewers such as the following:</p> <pre>[<Jira Number>] <Doc Title></pre> <p>For example:</p> <pre>[ID-1624] Getting started with UPF</pre> <p>In the Description, include a review by date. You can also include a description of the changes as needed and a due date for the reviewers.</p> <p>See Manage reviewers(see page 108) for additional guidelines.</p>	<p>Example</p> 
<p>G. In the Reviewers box, type the names of all reviewers. This can be the IDPM for the project and/or SMEs.</p> <div style="border: 1px solid red; padding: 5px; margin: 10px 0;"> <p>Do not leave the Reviewers box blank.</p> </div>	<p>Example</p> 
<p>H. Click Create.</p> <p>The reviewers will review the Pull Request and approve it or request additional changes.</p> <p>After the request is approved, a moderator will merge the request.</p>	

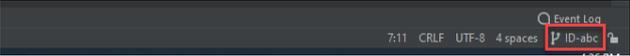
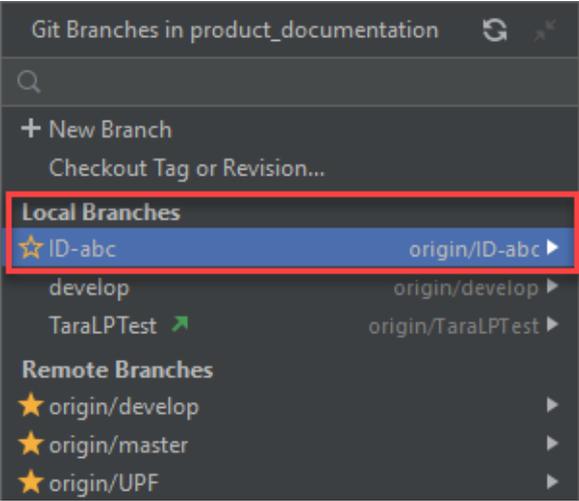
2.7 (Optional) Step 7: Make requested changes to a document

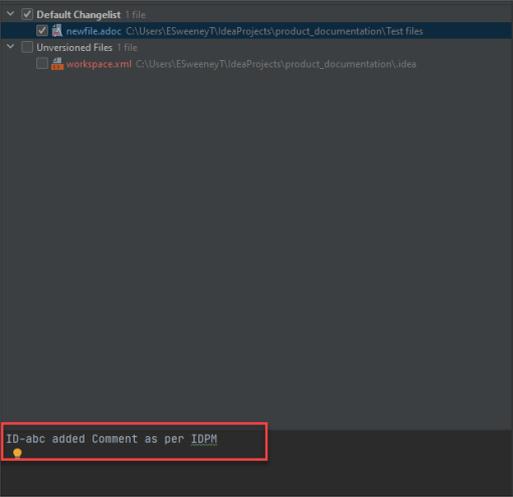
If a reviewer wants additional changes, they can mark the pull request as "Needs Work". Use the following steps to review their comments and make any updates.

Steps	Additional Information
<p>A. If your IDPM or SME is recommending changes to a file that you created or edited, you will receive an email similar to the one in the example.</p>	<p>Example</p> 

Steps	Additional Information
<p>B. Click View pull request to open Bitbucket and view details about the requested changes.</p> <p>You can do any of the following (or none):</p> <ul style="list-style-type: none"> • Type a comment in the Activity window to reply to the reviewer. • Click Reply to respond to a specific comment from the reviewer. This allows you to have a tracked discussion. • If a task exists, after you complete the work, select the check box to indicate that the work is done. See the examples on the right for more information about tasks and task lists. • Click ... and you will see one or more of the following options: <ul style="list-style-type: none"> • Click Convert to Task to convert a comment to a task which is shown in the list at the top of the pull request. This means that you don't have to search through comments for your to-do items. 	<p>Example</p>  <p>At the top of the window, you can click the task list button:</p> <p>Example</p>  <p>A task list opens. Click the link to go directly to the document that must be updated.</p> <p>Example</p> 

Steps	Additional Information
<ul style="list-style-type: none">• Click Create Jira issue to create a new issue in Jira. This is useful if a change is so large that it must be tracked separately and cannot be completed with the other changes that you are currently working on.• Click Convert to Comment to convert a task to a comment. <p>Next, you must fix the content.</p>	

Steps	Additional Information
<p>C. In IntelliJ, make sure that you are on the branch that you used to submit the file.</p>	<p>Check this on the bottom right corner of the window.</p> <p>Example</p>  <p>If the branch is not correct, click the branch above and select the correct branch from the list of Local Branches.</p> <p>Example</p>  <p>Then, go to the file that you must update.</p>
<p>D. Open the file that needs work and make the necessary changes.</p>	

Steps	Additional Information
<p>E. Make sure the file is listed (and selected) in the Default Changelist.</p>	<p>Example</p>  <p>Then, update the Commit message.</p> <p>Example</p> 
<p>F. Click Commit and Push. In the Push Commits box, click Push.</p>	<p>Your pull request already exists and is automatically updated.</p> <p>Example</p> 
<p>G. The reviewer can review the request again and can approve it. See Review content(see page 162).</p>	

<<REDACTED INFORMATION>>