PI Tasks For a Product Release

A Ready Reckoner

Version 1.0

PI BLR Team

opentext™

1 ➡ Procure Project Charter

Purpose: Project Charter lists all new features, enhancements, certifications, and so on agreed between Product Management and Project team (Development and Quality Assurance).

Action: The draft or final version of the Project Charter will be shared by PI Manager/Project team. If not received, send an email to PI Manager/Project team.

Source: Project Charter and resourcing

2 ➡ Obtain SAP ID

Purpose: Record the time for the product release documentation work.

Action: Send an email to the Program Manager/Project team and ask for the SAP ID for the product release. Share the SAP ID with the PI Manager. Add the SAP ID to your SAP Timesheet. For example, if your SAP ID is RD-03104. The SAP ID for documentation will be **RD-03104.04 (End User Documentation)**.

3 ➡ Create product-specific documentation Epics

Purpose: The documentation Epic Jira IDs are used to link to all Doc Tasks.

Action: To create a documentation Epic, perform the following steps:

1. In Jira, click **Create**.
2. Specify the details for the following required fields:
   - **Project**: Select the project Jira ID from the list.
   - **Issue Type**: Select **Epic** from the list.
   - **Epic name**: Type the name in the following format:
     PI: <release number> <product name> <description>
     For example: **PI: 23.4 Documentum Search documentation Epic**
   - **Summary**: Type the summary in the following format:
     Tracking ticket for all Doc Tasks for release <version>
     For example: **Tracking ticket for all Doc Tasks for release 23.4**
   - **Priority:** Select **Major** from the list.
   - **Fix Version/s**: Type the version and select the version from the list.
     For example: **23.4**
   - **Assignee**: Click **Assign to me**.
   - **Acceptance Criteria**: Type the following text:
     Release documentation must be complete.
3. Click **Create**.

When you are ready to start the documentation work for a documentation Epic, open the documentation Epic, click **Mark as To Do** and in the subsequent window, click **Mark as To Do**. Then, click **Start Progress**.

Source: Documentation Epics

opentext™

4 ➡ Create guide-specific subtasks for deliverables

Purpose: Creating guide-specific subtask for a product helps to know the list of documentation deliverables to be produced and published on My Support for that product release.

Action: To create guide-specific subtask for a product, perform the following steps:

1. Open the product-specific documentation Epic in Jira.
2. Click **More** > **Create sub-task**.
3. Specify the details for the following required fields:
    - **Issue Type**: Retain the default value **Sub-task** from the list.
    - **Summary**: Type the summary in the following format:
      PI: <short product name> <version> <guide name>
      For example: **PI: Documentum Search 23.2 User Guide**
    - **Priority**: Select **Major** from the list.
    - **Fix Version/s**: Type the version and select the correct version from the list.
    - **Assignee**: Click **Assign to me**.
    - **Labels**: Type a label in the following format:
      pi_<version>_deliverable
      For example: **pi_23.4_deliverable**
    - **Acceptance Criteria**: Type the following text:
      Release documentation must be complete.
4. Click **Create**.

When you are ready to start the documentation work for this Doc Task, open the Doc Task, click **Mark as To Do** and in the subsequent window, click **Mark as To Do**. Then, click **Start Progress**.

Source: Documentation Epics

5 ➡ Send PI milestones to team

Purpose: Inform the project team about the PI deliverables milestone dates such as finalizing core guides for technical review, releasing guides for publishing, and so on.

Action: Share the PI deliverables milestone information received from PI Manager with the project team.

6 ➡ Create Doc Tasks and Sub-tasks

Purpose: Creating Doc Task is mandatory and important to understand the reason for all the documentation updates made to the guide and to understand the processes followed for the guide for a product release.

Actions:

A. The project team must create Doc Tasks for PI. Educate the following procedure to the team on how to create a Doc Task for PI:

   1. In Jira, open the Feature Request or User Story or Bug that needs documentation updates.
   2. Click the **Documentation Required** button or **Workflow** > **Documentation Required**.
   3. From the **Documentation Required** list, select **Required**.
   4. Click **Documentation Required**.

   After the Doc Task is created and assigned to you, open the Doc Task and specify the details for the following required fields:

   - **Priority**: Select **Major** from the list.
   - **Fix Version/s**: Type the version and select the version from the list.
   - **Acceptance Criteria**: Type the following text:
     Release documentation must be complete.
   - **Assignee**: Click **Assign to me**.
   - **Story Point**: Assign the story point based on the complexity of the work. The story point number follows the Fibonacci series.
     For example: Type **1** or **2** or **3**.
     Note: Assigning a story point is not mandatory and it depends on the project team's requirement. Check with your project team if you need to assign story points for Doc Tasks.
   - **Epic Link**: Type **PI:** and select the product-specific documentation Epic from the auto-populated list.

B. If a Feature Request or User Story or Bug does not exist to create Doc Tasks for PI, you or the project team can manually create a Doc Task using the following steps:

   1. In Jira, click **Create**.
   2. Specify the details for the following required fields:
      - **Project**: Select the project Jira ID from the list.
      - **Issue Type**: Select **Doc Task** from the list.
      - **Summary**: Type the summary in the following format:
        PI: <name of the activity>
        For example: **PI: Connect with team and get inputs for the indexing feature**
      - **Priority**: Select **Major** from the list.
      - **Fix Version/s**: Type the version and select the version from the list.
      - **Acceptance Criteria**: Type the following text:
        Release documentation must be complete.
      - **Assignee**: Click **Assign to me**.
      - **Story Point**: Assign the story point based on the complexity of the work. The story point number follows the Fibonacci series.
        For example: Type **1** or **2** or **3**.
        Note: Assigning a story point is not mandatory and it depends on the project team's requirement. Check with your project team if you need to assign story points for Doc Tasks.

- **Epic Link**: Type **PI:** and select the product-specific documentation Epic from the auto-populated list.

3. Click **Create**.

When you are ready to start the documentation work for this Doc Task, open the Doc Task, click **Mark as To Do** and in the subsequent window, click **Mark as To Do**. Then, click **Start Progress**.

C. Create a subtask to perform the additional activity that belongs to the parent Doc Task. To create a subtask, perform the following steps:

1. Open the Doc Task in Jira.
2. Click **More** > **Create sub-task**.
3. Specify the details for the following required fields:
    - **Issue Type**: Retain **Sub-task**, the default value.
    - **Summary**: Type the summary in the following format:
      PI: <name of the activity>
      For example: **PI: Schedule a meeting with SME**
    - **Priority**: Select **Major** from the list.
    - **Fix Version/s**: Type the version and select the version from the list.
    - **Assignee**: Click **Assign to me**.
    - **Story Point**: Assign the story point based on the complexity of the work. The story point number follows the Fibonacci series.
      For example: Type **1** or **2** or **3**.
      Note: Assigning a story point is not mandatory and it depends on the project team's requirement. Check with your project team if you need to assign story points for Doc Tasks.
    - **Acceptance Criteria**: Type the following text:
      Release documentation must be complete.
4. Click **Create**.

Source: Doc Tasks

7 ➡ Upversion all guides

Purpose: Upversioning of the guide ensures that the guide is ready for documentation update as per the Project Charter items.

Action: To upversion the guide, perform the following steps:

1. In the XML source, change the **Revision number**.
   For example: For release 23.4, change to EDCSRCH**230400**-IGD-EN-01.
2. Search for all instances of previous release version number (for example, 23.2) and replace with the current version (for example, 23.4) or retain the release version number depending on the context (for example, if a statement is applicable for *23.2 or later,* the statement need not be changed to *23.4 or later* as it will change the context of the documentation).

Tip: Alternatively, you can highlight all the release version number in the guide and send it to the project team for confirmation to change the release version number.

Source: Changing the revision number

8 ➡ Process Documentation List (DocList)

Purpose: Processing the DocList Spec file is mandatory. It is important for the documentation release process and to obtain the processed DocList URL to be shared with the Release Management (RM) team for publishing on My Support along with your product guides. The processed DocList URL is either shared in the Release Workflow Form to the RM team during the GA or is shared in an email to the RM team after the product release announcement to publish Documentation List on My Support.

Action: To update an existing DocList Spec file (XML format) from the Ollie location, perform the following steps:

1. Go to the DocList Spec sources page in Ollie.
2. Open the folder that contains the DocList Spec file for your product.
3. Select the DocList Spec file you want to update and click **Reserve**.
4. Click **Download**.
5. Open the DocList Spec file from the **Downloads** folder using the Microsoft Excel application.
6. Select the last release version column, copy, and paste as next column.
7. Change the **Variants** value in Row 14 to new revision number.
   For example: **230400**
8. Update the **Release Notes** value to **KCO-KB0000000**.
9. Update the value for all the other guides to the release version number.
   For example: **23.4.0**
10. Save the file and when prompted in the **Do you want to keep using that format** window, click **Yes**.
11. Go to the DocList Spec sources page in Ollie.
12. Make sure that the DocList Spec file you edited is selected.
13. Click **Add version**.
14. Go to the **Downloads** folder, select the updated DocList Spec file, and click **Open**.
15. Go to the DocList Spec sources page in Ollie.
16. Make sure that the DocList Spec file you edited is selected.
17. Click **Unreserve**.
18. Send an email to the PI BLR DocList processing team (Vikas and Naga) to process the updated DocList Spec file for your product.
19. After the confirmation email from the PI BLR DocList processing team (Vikas and Naga), verify if the processing of the DocList was successful the next day using the following steps:
    a. Go to the Available DocLists documentation library page.
    b. In the **Filter** check box, type your product ManID. For example, the ManID for Documentum Search is **EDCSRCH**.
    c. If you find a blue flag ( ) or a gray flag ( ) against the release version for the DocList, it confirms that DocList has been processed successfully.

d. For confirmation, click the **KC** link and provide the OpenText My Support credentials (same as your laptop login credentials). The documentation list page opens and should have the following message for all the guides:
The documentation for this node is under preparation
If you do not see this message, send an email to the PI BLR DocList processing team (Vikas and Naga).
The link to the guides does not appear and the message about the node under preparation appears because none of your guides are in the PI Production Server (PI Root).

Note: If you want to create a new DocList Spec XML file, please reach out to any member of the PI BLR writing team for help.

Source: Updating and posting a DocList

9 ➡ Connect with project team, identify POCs, and advocate PI processes

Purpose: Understand which feature or enhancement or certification has an impact to documentation, procure the list of Dev and QA point of contacts to work with, and to create appropriate Doc Tasks either along with them or to educate them to create Doc Tasks in Jira to update the impacted guide.

Action: Connect with the project team (Engineering Manager, Development Engineer, and Testing Engineer) using the email or meet in person and get yourself introduced. Discuss about the Project Charter items with the project team and ask if any of the new features, enhancements, certifications, and so on has any impact to documentation and procure the list of engineers (point of contacts) to work with. Attend the daily standups, PMT (Project Management Team) calls and other required scrum ceremonies as and when possible, to understand the scope and progress of the product release, and associated documentation requirements and updates.

Educate the engineering team about creating Doc Tasks for the impacted Epic, Feature Request, User Story, Bug, and Sub-tasks using the **Documentation Required** option in Jira.

Sources: Meetings and Determining documentation requirements

10 ➡ Update impacted guides for release

Purpose: Document all the required documentation updates across guides (also known as Documentation Set) as defined in the Project Charter and for any other ad hoc documentation updates outside of the Project Charter.

Action: Collaborate with the project team to get inputs and update the guides accordingly. Make sure that you have a Doc Task for each update you make to the guides for a release.

Tip: For all documentation updates you make in a guide, make sure that you add a comment with the Doc Task Jira ID and/or mention the name of the engineer who provided the documentation inputs or comments in the impacted chapters and/or sections. This will be helpful to find all the documentation updates made in a guide for a particular Doc Task Jira ID that may belong to a new feature, enhancements, certifications, ad hoc inputs or comments, and so on during the release cycle or any time in the future too.

Important: PI does not own or update the product Release Notes. Release Notes are created, updated, and owned by Product Managers. If the Product Manager sends the Release Notes to you for editorial review and Word-related formatting fixes, you can help.

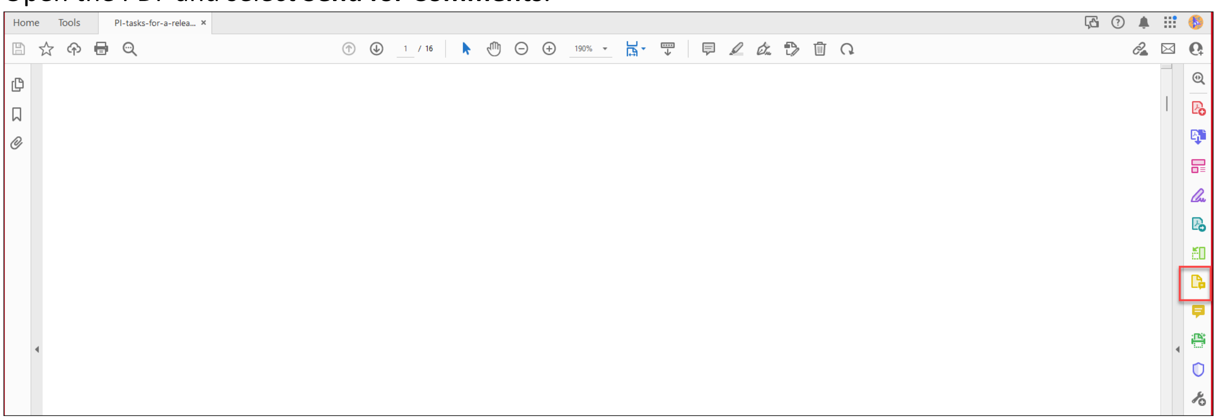
Source: [Authoring and producing]

11 ➡ Set up and send draft content for technical review using shared PDF review

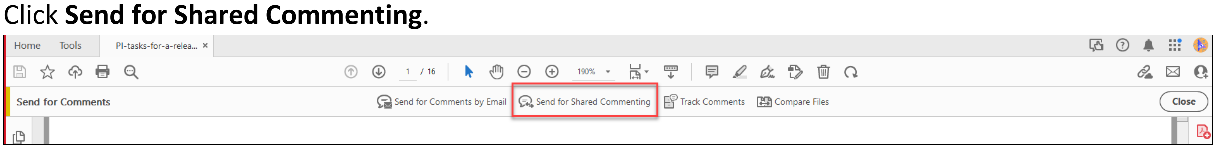
Purpose: Collect new inputs and review comments from multiple reviewers. It helps reviewers to look at other reviewers' comments to add to the existing comment or to avoid duplication of comments.

Action: To set up and send the updated guide for shared PDF review, perform the following steps:

1. Open the PDF and select **Send for Comments**.



2. Click **Send for Shared Commenting**.



3. In the **Send for Shared Commenting** dialog box, select the location and click **Next**.
   Note: Location for collecting the review comments is a one-time activity and must be set using the following steps:
   a. In the **Send for Shared Commenting** dialog box, select **Automatically collect comments on my own internal server** and click **Next**.
   b. Select the **Network folder** and browse the path. The default value is [\\wlpifs01.opentext.net\pdf_reviews\ecd_<Name>]. Provide a value for **<Name>**. Do not change the path value.
   c. Click **Next**.
   d. Select **Save a local copy and manually send it later** and type the location.
   e. Click **Next**.
   f. In the **Provide a name for this server profile** field, type the name of the folder, and click **Next**.
4. Click **Review Deadline** and modify the date, as required. The recommended timeline is three working days.

opentext™

5. Click **Finish**.
6. Copy the PDF that is set up for shared review from your local file system and send to the reviewers through email.

Note: You can send the draft content for technical review through email too without using shared PDF review.

Source: Posting documents for review

12 ➡ Incorporate comments and send for final technical review with timeline

Purpose: Get a closure for the writer and the project team on the technical accuracy of the documentation.

Action: Update the document with new inputs and review comments received from the project team and send the next/final updated document for final technical review with the timeline in your email or shared PDF review. The timeline can be approximately three to five working days depending on the volume of content.

13 ➡ Perform self-review of technical reviewed content

Purpose: Self-review the technically reviewed content to make it style guide compliant.

Action: To self-review the technically reviewed content before sending it for peer review or PDF check, perform the following steps:

1. Go to the Document self-review checklist page.
2. Follow and apply the guidelines as described in the **Style Guide compliance** and **PDF check** sections.

Source: Document self-review checklist

14 ➡ Send technical reviewed content for peer review and incorporate comments

Purpose: Collect review comments from the PI BLR peer review team (Renny, Mary, Preethi, and Priscilla) to ensure that the updated content is compliant with OpenText Corporate Style Guide.

Action: Send the final technically reviewed content to the PI BLR peer review team highlighting the content that needs to be peer reviewed through email or shared PDF review with timeline. Update the document with the peer review comments received from the PI BLR peer review team.

Source: Peer review best practices

15 ➡ Send final guide for PDF check and incorporate comments

Purpose: Receive review comments for the PDF output format with respect to typos, formatting changes, and presentability of the finalized guide.

Action: After completing the technical and peer reviews of the entire guide, send the finalized PDF output format of the guide with the timeline for PDF check through email or shared PDF review to the PI BLR PDF check team (Mary and Naga). Update the document with the PDF Check comments received from the PI BLR PDF Check team.

Sources: [Document self-review checklist]() and [Final review checklist]()

16 ➡ Share final guide copies with all stakeholders (Optional)

Purpose: Inform the stakeholders about the completion of all documentation updates to the guide for the product release.

Action: Send the finalized guide copy as a reference to all stakeholders through email mentioning about the readiness of the guide for publishing on My Support.

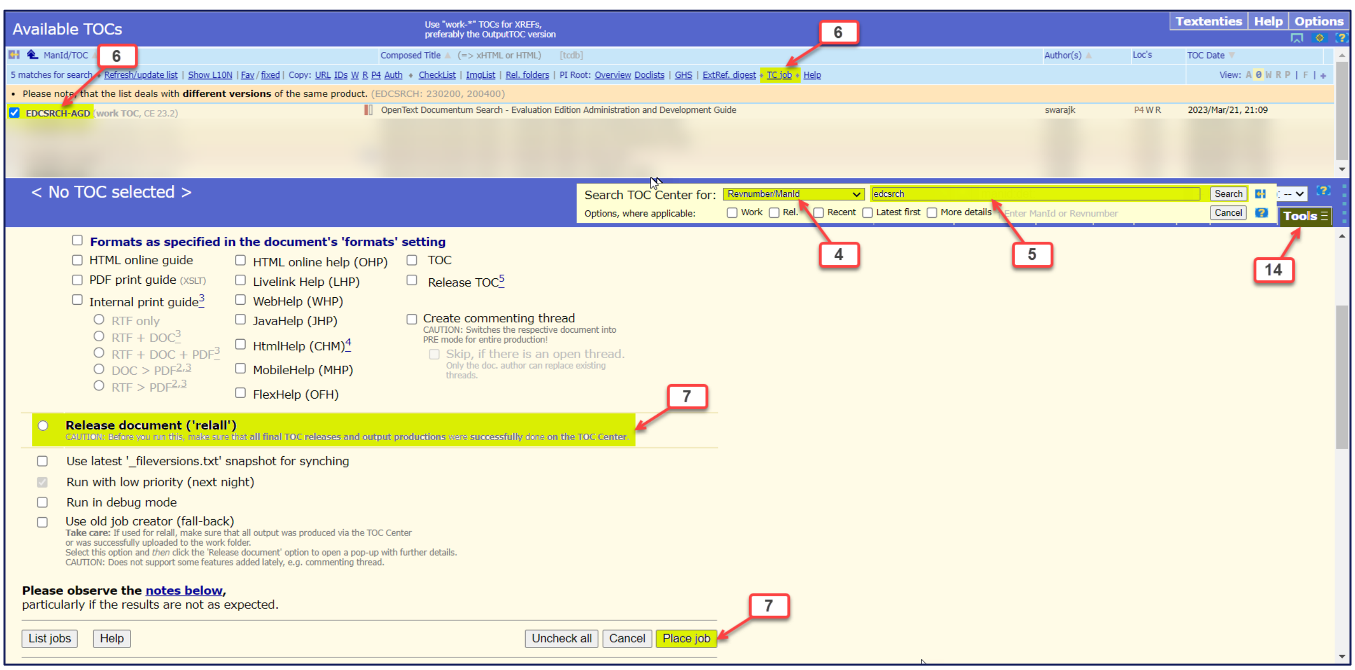
17 ➡ Obtain signoff for final guides from project team

Purpose: Get confirmation from the project team that the release-specific documentation updates are completed, and the guide is ready for publishing on My Support.

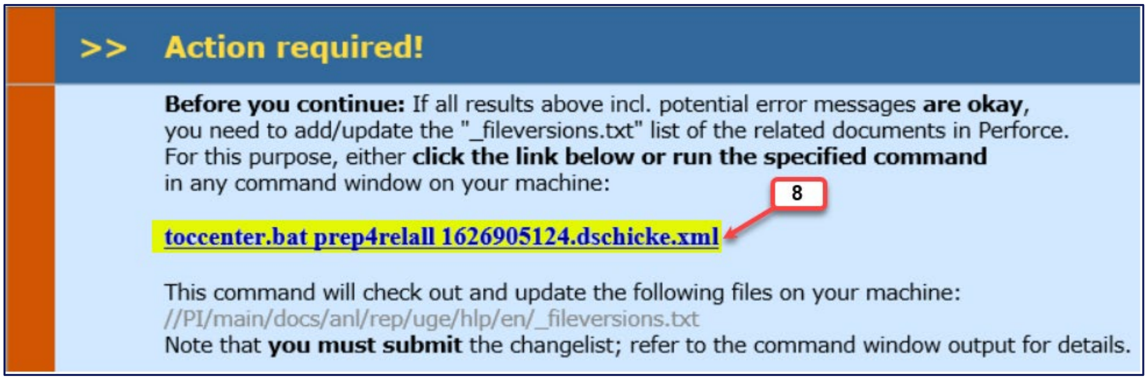
Action: In [Jira](), attach the PDF format of finalized guides to the guide-specific Sub-tasks and assign them to the Engineering Manager/project team to verify and close them.

18 ➡ Release final guides using TOC center

Purpose: Release the finalized guides (preferably one guide at a time) and send to the PI Production Server (PI Root).



opentext™



Action: In the [PI TOC Center](), select the guides and release them using the **Relall** option. To release a guide, perform the following steps:

1. Open Perforce.
2. Open [PI TOC Center](), in Google Chrome or Mozilla Firefox, using one of the following ways:
    - Go to [PI TOC Center]().
    - In XML source, click **PI-Tools** > **Open TOC Center**.
3. In the **<No TOC selected>** ribbon, click the **Search** icon.
4. In the **Search for TOC Center for** list, select **Revnumber/ManId**.
5. In the **Enter the search string here** field, type the ManID of your product and click **Search**.
   For example: **edcsrch**
6. Select the guide you want to release and click **TC job**.
7. In the **TOC Tools** window, select **Release Document ('relall')** and click **Place job**.
   Note: Make sure that your email address is available in the **Send result to this e-mail address:** field.
8. In the **Action required** window, click the link with the filename in the .xml format that downloads a batch file.
9. Click the downloaded batch file. A command prompt window opens mentioning the Perforce changelist followed by the **Press any key to continue** message.
   WARNING: Do not press any key in the command prompt window until you submit the changelist in Perforce.

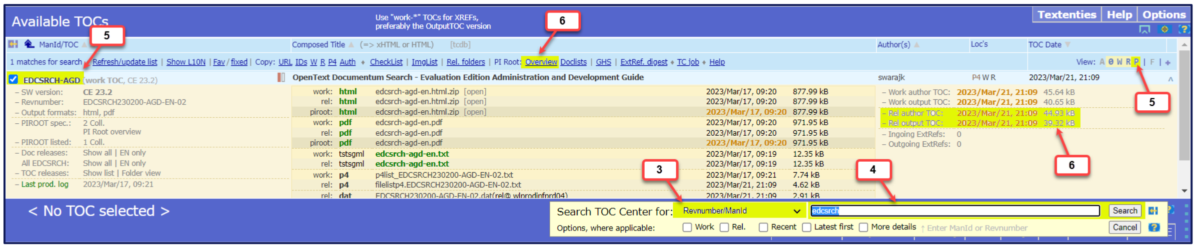
10. Go to **Perforce** and click the **Refresh** button.
11. Open the **Pending** tab, select the changelist for the relall job, right-click and click **Submit**.
12. Retain the default value in the changelist description and click **Submit**.
13. Go back to the command prompt window and press any key to continue (preferably spacebar).
    The job for releasing the guide is placed in the **PI TOC Center**.
14. Check the list of jobs by clicking the three horizontal lines beside **Tools** in the **<No TOC Selected>** ribbon.

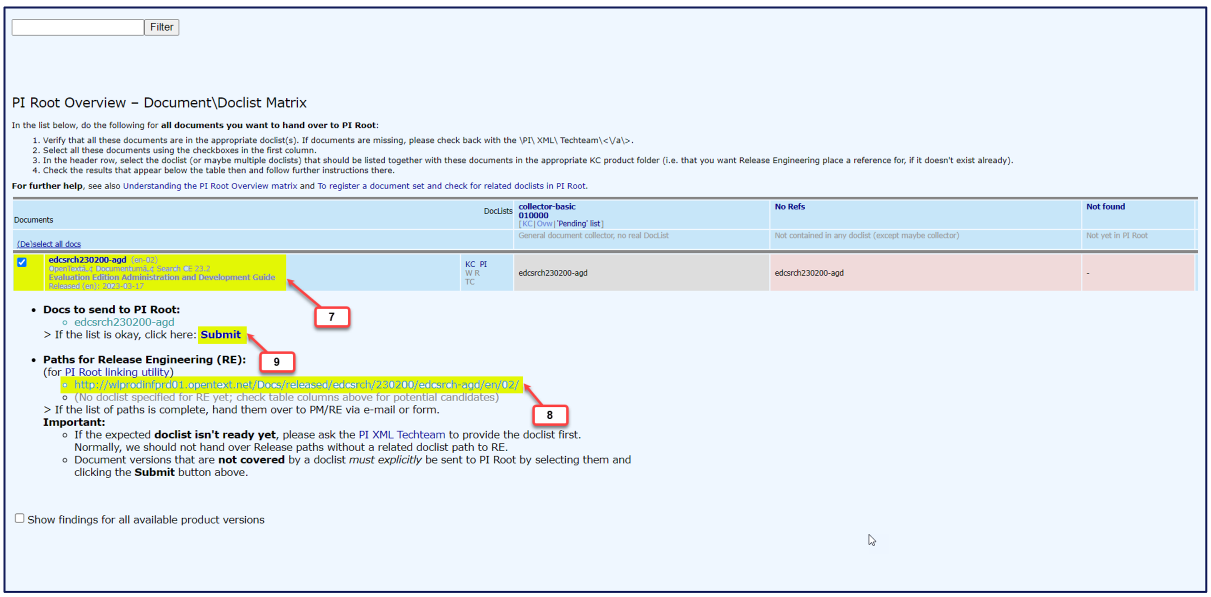
By default, the **relall** job is placed in the night queue. To move the job to the day queue, select the job and click **Run checked job 'asap'**.
After the guide is released, an automated email is generated. Click the released URL link in the email to verify if the link to the output formats such as PDF, HTML, and HTML ZIP are listed in the released folder path.

Source: [Releasing a document]()

19 Send released guides to PI Root using TOC center and verify

Purpose: Send the released guide to the PI Production Server (PI Root) and verify if the guide has reached PI Root successfully to confirm that the guide is ready for the RM team to pick up for publishing on My Support for the product release.





Actions:

A. To send the guide to PI Root, perform the following steps:

1. Open PI TOC Center, in Google Chrome or Mozilla Firefox, using one of the following ways:
   - Go to PI TOC Center.
   - In XML source, click **PI-Tools** > **Open TOC Center**.
2. In the **<No TOC selected>** ribbon, click the **Search** icon.
3. In the **Search for TOC Center for** list, select **Revnumber/ManId**.
4. In the **Enter the search string here** field, type the ManID of your product and click **Search**.
   For example: **edcsrch**
5. Select the guide you want to send to PI Root and click **View:** > **P**.
6. Verify if the date provided for **Rel author TOC** and **Rel output TOC** has the same date on which you released the guide and then click **PI Root:** > **Overview**.
7. In the **PI Root Overview – Document\Doclist Matrix** page, select the guide.

8. Copy the URL in the **Paths for Release Engineering (RE)** section and save in a Notepad file. You will need this in one of the upcoming activities.
   For example: http://wlprodinfprd01.opentext.net/Docs/released/edcsrch/230400/edcsrch-igd/en/01/
   (This is a dummy link. Do not click this link. 😊)
9. In the **Docs to send to PI Root** section, click **Submit.**
   Note: The time taken to send the guide to PI Root is approximately 24 to 48 hours.

The process for a guide to reach PI Root and/or My Support area involves the following steps:

i. The PI Root script runs automatically at 6.45 AM IST every morning and processes the following items:
   a. DocLists.
   b. Guides that are marked to send to PI Root.

ii. The PI Root script runs based on the ManID and in the alphabetical order.

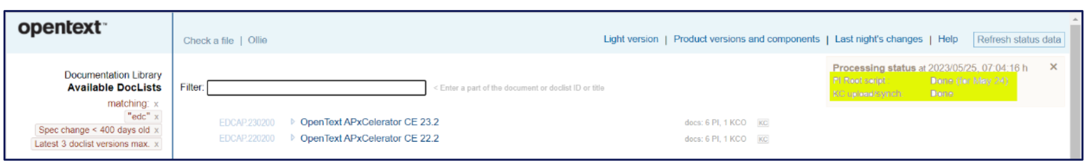
iii. After the PI Root script completes all the processes, the KC upload/synch script is run manually by one of our Information Architects (Ekbert from Germany or Paul from Columbus, USA or Rob from Carlsbad, USA).

iv. After the KC upload/synch script completes all the processes, the processing of DocLists and the task of sending the guides to PI Root and/or My Support area are completed.

Source: Releasing a document

B. To verify if the guide has reached PI Root successfully, perform the following steps:

1. Go to the Available DocLists documentation library page.
2. In the **Processing status** section, verify if the status of the PI Root and KC upload/synch scripts is shown as **Done**.



3. Type the following URL format in any browser and press **Enter**:
   https://webapp.opentext.com/piroot/edcsrch/v230400/edcsrch-ugd/docovw.xml
   where,
   edcsrch is your ManID, 230400 is your revision number, and ugd is the guide type.
   For example:
   To verify if Documentum Search Installation Guide has reached PI Root, use the following URL:
   https://webapp.opentext.com/piroot/edcsrch/v230400/edcsrch-igd/docovw.xml (This is a dummy link. Do not click this link. 😊)
4. Provide the OpenText My Support login credentials, when prompted.
5. If the guide has reached PI Root successfully, you can view the output formats (HTML guide, PDF, and HTML ZIP) links of the guide.

opentext™

20 ➡ Prepare, update, and save release staging spreadsheet in Ollie

Purpose: Send the Product Release Staging spreadsheet updated with all PI information about the finalized guides to the Product Manager or Program Manager or Engineering Team. The spreadsheet will be uploaded in the Release Workflow Form that will be used by the RM team for publishing the guides on My Support for the product release.

Action: To prepare and update the product release staging metadata spreadsheet, perform the following steps:

1. Go to the [Release Tracker] page in Ollie.
2. Open the previous version release folder. For example, 23.2.
3. Select the product-specific Release Tracker spreadsheet and click **Reserve**.
4. Click **Download**.
5. Open the downloaded Release Tracker spreadsheet and update the following information:
    - **Filename**: Update the guides with current release version.
    - **Product Name**: Retain the existing value.
    - **Version**: Type the current release version.
    - **Content Kind**: Retain the existing value.
    - **Content Type**: Retain the existing value.
    - **Platform**: Retain the existing value.
    - **Filename Description**: Update all version numbers with current release version.
    - **KB Article in ServiceNow**: Do not provide any value.
    - **File Location**: Type the released URL you saved in a notepad file as described in step 19.
      Note: If any guide is not published for a release, remove the entry from the spreadsheet.
    - **Search Criteria**: Type the ManID along with the revision number.
      For example: **EDCSRCH230200**
6. Retain the Release Notes entry as is.
   Note: The **File Location** for Release Notes will be filled or provided by the Product Manager.
7. Save the file.
8. Go to the [Release Tracker] page in Ollie.
9. Select the Release Tracker spreadsheet and click **Add version**.
10. In the browse window, select the updated Release Tracker spreadsheet in your local file system and click **Open**.
11. After the Release Tracker spreadsheet is uploaded, click **Unreserve**.

21 ➡ Send the product release staging spreadsheet for Release Workflow

Purpose: Share the updated *Product Release Staging spreadsheet* (metadata sheet) which will be attached in the Release Workflow Form for the product release.

Action: Send an email to the Product Manager/Program Manager/Engineering team/PI Manager attaching the *Product Release Staging spreadsheet* updated with PI information as described in step 20.

opentext™

22 ➡ Resolve and/or close Doc Tasks, Sub-tasks, and documentation Epics

This task can be performed either before or after sending the product release staging spreadsheet for Release Workflow. This task varies from one project team from another.

Purpose: All the Doc Tasks, Sub-tasks, and documentation Epics created by you and the project team for the product release must be resolved and/or closed either by you or the project team before or after the submission of the *Product Release Staging spreadsheet* to Product Manager/Program Manager/Engineering Team/PI Manager.

Action: To resolve the Doc Task, Sub-tasks, and documentation Epics in Jira, perform the following steps:

1. Open the Doc Task or Sub-task or documentation Epic in Jira.
2. Attach the PDF snippet related to the Doc Task or Sub-task.
   Note: You need not attach any PDF for documentation Epic.
3. Click **Resolve Issue**.
4. Select **Fixed** from the **Resolution** list.
5. Select the engineer name from the **Assignee** list.
   Note: For documentation Epic, click **Assign to me**.
6. Type relevant comments in the **Comments** box.
7. Click **Resolve Issue**.

   Note: If the project team marks the resolved Doc Tasks or Sub-tasks as *verified*, then close those Doc Tasks or Sub-tasks using the **Close Issue** button in Jira. However, if the project team wants to ***take the responsibility*** to verify and close the resolved Doc Tasks or Sub-tasks, do not close them. In this case, follow up with the project team to get the resolved Doc Tasks or Sub-tasks closed. After all the Doc Tasks and/or Sub-tasks are verified/closed either by you or the project team, then close the product-specific documentation Epic.

Source: Final authoring tasks

23 ➡ Verify the published documents on My Support

Purpose: Verify if all the release documents are published on My Support correctly after the product release announcement email.

Action: To verify the list of documents on OpenText My Support, perform the following steps**:**

1. On the Home page, click **Knowledge** > **Product Documentation**.
2. In the **Search the Knowledge Base** box, type the meta tag in the following format:
   #<Product name><release version>Documentation and then click **Search**.
   For example: **#DocumentumSearch23.2Documentation**

   Make sure all the guides that you worked for the product release are listed and the links to the guides are working correctly. Make sure the Release Notes worked by your Product Manager is also listed.

24 ➡ Update KB article number for Release Notes in DocList, process, and verify

Purpose: Update the KB article number for the Release Notes in the DocList Spec file and get it processed so that the processed DocList URL can be shared with the RM team through email for publishing the Documentation List on My Support.

Action: To update the KB article number for Release Notes in the DocList Spec file, send for processing, and verify the successful update, perform the following steps:

1. On the My Support Home page, click **Knowledge** > **Product Documentation**.
2. In the **Search the Knowledge Base** box, type the meta tag in the following format:
   #<Product name><release version>Documentation and then click **Search**.
   For example: **#DocumentumSearch23.2Documentation**
3. Click and open the Release Notes document and copy the KB article number.
4. Go to the [DocList Spec sources] page in Ollie.
5. Open the folder that contains the DocList Spec file for your product.
6. Select the DocList Spec file you want to update and click **Reserve**.
7. Click **Download**.
8. Open the DocList Spec file from the **Downloads** folder using the Microsoft Excel application.
9. Paste the KB article number you obtained from step 3 for **Release Notes** in Row 25.
   For example, change the **Release Notes** value from **KCO-KB0000000** to **KCO-KB1234567**.
10. Save the file and when prompted in the **Do you want to keep using that format** window, click **Yes**.
11. Go to the [DocList Spec sources] page in Ollie.
12. Make sure that the DocList Spec file you edited is selected.
13. Click **Add version**.
14. Go to the **Downloads** folder, select the updated DocList Spec file, and click **Open**.
15. Click **Unreserve**.
16. Send an email to the PI BLR DocList processing team (Vikas and Naga) to process the updated DocList Spec file for your product.
17. After the confirmation email from the PI BLR DocList processing team, verify if the processing of DocList was successful the next day using the following steps:
    a. Go to the [Available DocLists documentation library] page.
    b. In the **Filter** check box, type your product ManID. For example, the ManID for Documentum Search is **EDCSRCH**.
    c. If you find a blue flag () or a gray flag () against the release version for the DocList, it confirms that DocList has been processed successfully.
    d. For confirmation, click the **KC** link and provide the OpenText My Support credentials (same as your laptop login credentials). The documentation list page opens and displays the list of all the guides including the Release Notes for your product.

If you are not able to see the list of all the guides and if you see **The documentation for this node is under preparation**, then send an email to the PI BLR DocList processing team (Vikas and Naga) and ask for a fix.

Source: [Updating and posting a DocList]

opentext™

25 ➡ Publish Documentation List on My Support and verify

Purpose: Documentation List contains the list and the link to all the guides including Release Notes for a given product in one view for customers' easy reference to product documentation.

Actions:

A. To publish the Documentation List on My Support, perform the following steps:

1. Go to the Available DocLists documentation library page.
2. In the **Filter** check box, type your product ManID. For example, the ManID for Documentum Search is **EDCSRCH**.
3. Click the **KC** link and provide the OpenText My Support credentials (same as your laptop login credentials). The documentation list page opens and lists all the guides.
4. Copy the Documentation List URL from the address bar, prepare and send an email to the RM team (send to: releasehelp@opentext.com) copying the PI Manager with the following information requesting them to publish the Documentation List on My Support:
   a. Documentation List URL
   b. Meta tag that must be associated with the Documentation List URL

B. To verify the published Documentation List on OpenText My Support, perform the following steps**:**

1. On the Home page, click **Knowledge** > **Product Documentation**.
2. In the **Search the Knowledge Base** box, type the meta tag in the following format:
   #<Product name><release version>Documentation and then click **Search**.
   For example: **#DocumentumSearch23.2Documentation**

   Make sure the Documentation List is listed and the link is working correctly.

opentext™

About OpenText

OpenText enables the digital world, creating a better way for organizations to work with information, on premises or in the cloud. For more information about OpenText (NASDAQ: OTEX, TSX: OTC) visit opentext.com.

Connect with us:

OpenText CEO Mark Barrenechea's blog

Twitter | LinkedIn