

OFBiz Documentation Team

Initial wiki page to help co-ordinate efforts of OFBiz Documentation team.

Documentation Roles

We are looking for volunteers in the following roles:

Writers / Authors

- to review sources of information for the topic
- select information to be incorporated
- to write content
- move documentation sources to document attic

Reviewers / Proofreaders

- to review documents written by Authors
- check for grammar and spelling mistakes
- confirm that content is clear to understand
- liaise with authors

Editors

- organise documents into the correct locations
- manage structure of overall documents
- test documents and create patch files
- update patch files into the trunk

NOTE: If you author an document, then you cannot be the proofreader

Mentors

- help documentation contributors
- provide general background OFBiz help
- provide examples documents
- help contributors test their documentation

Team Members

The list below is the list of people who are taking part in the OFBiz documentation effort. Please add your details below if you would like to volunteer to help

Some people have volunteered to be mentors to other team members. If you would like a mentor, then please feel to approach any of the mentor volunteers.

Name	Confluence Id	Location / Timezone	In Skype Group	Willing to be a Mentor	Documentation Role	Mentor
Sharan Foga	Sharan Foga	Prague, UTC+1	Yes	Yes	Author, Proofreader, Editor	
Olivier Heintz	Olivier Heintz	France, UTC+1	Yes		Author	
Deepak Nigam	Deepak Nigam					
Tim Boyden	Timothy Boyden	Boston, UTC-4	Yes		Author, Proofreader	
Craig Parker	Craig Bachelor	Maine, UTC-5	Yes			Sharan Foga
Arthur Marquez						
Swapnil M Mane	Swapnil Mane	India, UTC+5.5	Yes	Yes	Author, Proofreader, Editor	
Michael Brohl	Michael Brohl	Germany, UTC+1	Yes	Yes	Author, Proofreader, Editor	
Pranay Pandey	Pranay Pandey	India, UTC+5.5	Yes			
Aditya Sharma	Aditya Sharma					
Dennis Balkir	Dennis Balkir	Germany, UTC+1				
Akash Jain	Akash Jain	India, UTC+5.5	Yes			
Tarun Thakur	Tarun Singh Thakur		Yes			
Piotr Walesiak						
Giulio Speri	Giulio Speri	Italy, UTC+1				

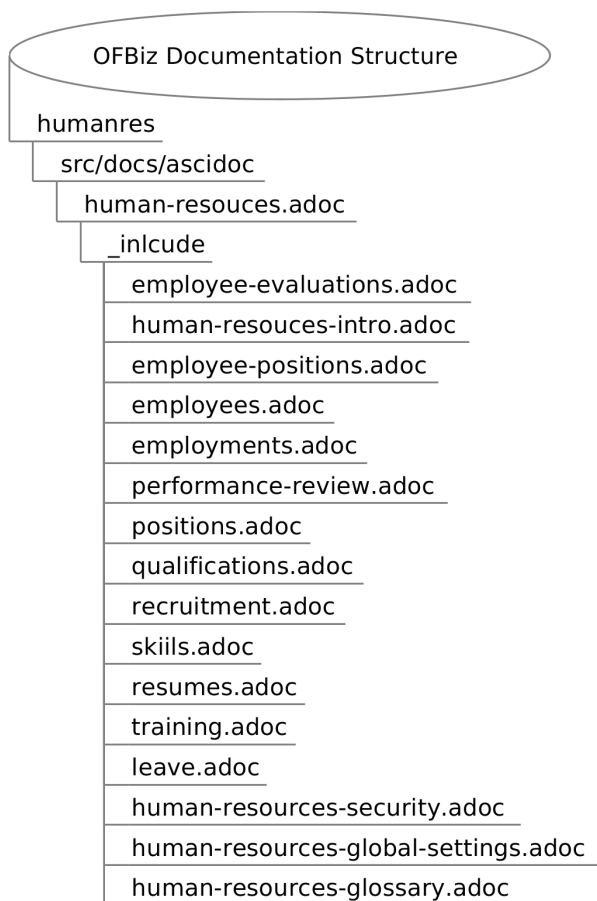
Taher Alkhateeb	Taher Alkhateeb	Kuwait, UTC +3	Yes	Technical Advice		
Vikram Gupta	Vikram Gupta	Durban, SA, UTC+2				
Mauricio Tavares		UTC+1				
Badar Ali	Badar Ali		Yes			Swapnil Mane
Allan Zarsuela	Allan Zarsuela	UAE, UTC+4	Yes		Author, Proofreader	Sharan Foga
Rebecca Johnson						
Benjamin Jugl	Benjamin Jugl	Germany, UTC+1				
Daniel Mejia					Spanish Translation, Author	Sharan Foga
Sanjay Yadav	Sanjay Yadav	India, UTC+5.5		QA Advice	Author, Proofreader, Editor	
Wolfgang Rauchholz	wp.rauchholz	Barcelona, Spain, UTC +2	No	Need a mentor		

Documentation Example : Writing Our First Guide Together

To get started we will be collaborating on writing the Human Resources guide together.

[Human Resources Guide](#)

Example structure for adoc files



Suggested Processes

Creating Documentation Jiras

1. Create one main umbrella JIRA per module (eg Human Resources Guide JIRA Task List



2. Create a JIRA for each of the individual documents that need to be written (e.g one for resumes.adoc, and another for human-resources-intro.adoc)
3. JIRAs for individual documents will include the name of the document and either an template (or a link to a template) to use for the document
4. Link the individual JIRAs as sub tasks to the main umbrella JIRA

Assigning Yourself to Work on a JIRA

1. A JIRA is available to be picked up and worked on if it does not have anyone assigned to it
2. To pick up and start working on a JIRA, assign yourself to a JIRA that that do not have anyone assigned
3. Click the "Start Progress" button and keep it like that as long as you are working on the task. This to let know others that you are actively working on the issue. Possibly click the "Stop Progress" button if you are pausing for this task. You may even unassign yourself if it's for a long period.

Writing the Documentation

1. Each JIRA will have a template or a link to template for the write to use
2. Writers will attach a text file to the JIRA ticket (NOTE: This will not be a patch file: Reasoning is that we do not want to lose contributors because they don't understand how to use subversion etc and create patches. The patch creation can be done at a later stage by the editors!)
3. Writers will add a comment to the ticket that the file is ready for review (Can we use any existing status to help?)

Reviewing Documentation

1. Reviewers will check for any JIRA tickets ready for review (Not sure if we can use an existing status or rely on notification comments. Perhaps when a document is started we already assign a reviewer who could also be a mentor....)
2. Reviewers will read to ensure that the documentation is clear and readable (if they have any queries they can contact the writer)
3. and check the written documentation and correct any minor grammatical errors
4. Reviewers will then approve the text file as ready to be tested

Creating and Testing the Documentation

1. Editors will then take the text file copy it into a working version of the Trunk
2. They will copy the text file into the correct location in the documentation tree
3. They will build the documentation and check that it the document appears correctly in the generated guide (NOTE Maybe include a separate sub section here on the commands to use for building and the location of the generated documents: Use command **`./gradlew generateOfbizDocumentation`** to generate the PDF and HTML files)
4. They will create a patch file for the documentation and attach it to the individual JIRA (NOTE: This means that the individual JIRA will contain the original text file received from the writer and also a patch file with the intergrated written text)
5. They will change the status that the ticket has been tested and is ready to be committed to the Trunk

Updating Documentation into the Trunk

1. Editors that are committers will look for tickets that are ready to be committted and commit them into the trunk
2. They will send notifications to the writers and reviewers that the document has been uploaded
3. Editors will close the individual issue

Document Guidelines

Please refer to <https://asciidocor.org/docs/asciidoc-recommended-practices/>

Formatting

- Each .adoc file must contain the Apache license header (put between "//// license... ////")
- Recommendation say one sentence per line but are currently doing limiting lines to approx 80 characters... Not sure if are at the one sentence per line stage yet?
- Filenames will be in lower case and extension will be .adoc
- Section titles will use asymmetric atx style (e.g == This is an example of an Asymetric Section Title)
- When including another file using the 'include' directive, please ensure that there is a blank line between each include line (NOTE: From Taher - if we stack directives on top of each other asciidoctor might get confused especially if the headers in the different documents are a different levels)

Naming

We will be implementing a consistent naming standard for the documentation content files.

- Each guide will be named after the component / module name (e.g. humanres.adoc, accounting.adoc, manufacturing.adoc, party.adoc etc).
- Lower level files that are in the include directory will include a prefix/shortname indicating the component name, separated by dashes (e.g hr-intro.adoc, hr-glossary.adoc...etc)
- Similar pages will have consistent naming. We will have several intro, glossary, FAQ, settings, security, so the naming format will be ([shortname]-intro, [shortname]-glossary, [shortname]-faq, [shortname]-settings, [shortname]-security etc.)

Example for Human Resources this will be as follows:

1. humanres.adoc
 - a. hr-intro.adoc
 - b. hr-employee-evaluations.adoc
 - c. hr-glossary.adoc
 - d. hr-employee-positions.adoc
 - e. hr-employees.adoc
 - f. hr-employments.adoc
 - g. hr-performance-review.adoc
 - h. hr-positions.adoc
 - i. hr-qualifications.adoc
 - j. hr-recruitment.adoc
 - k. hr-skills.adoc
 - l. hr-resumes.adoc
 - m. hr-training.adoc
 - n. hr-leave.adoc
 - o. hr-security.adoc
 - p. hr-global-settings.adoc


So for the party manager this could be :

1. party.adoc
 - a. party-intro.adoc (NOTE: Do we look at making the short code pty or something shorter than party????)
 - b. party-glossary.adoc
 - c. party-faq.adoc
 - d. party-settings.adoc
 - e. party-security.adoc
 - f. etc.

This could give people a guideline for the base structure and you can immediately recognize what the file contains.

JIRA Issue Task List

Human Resource Guide

 Unable to render Jira issues macro, execution error.

Assigning Yourself A Jira Issue

The first step in the process is looking at the list of open documentation issues / sub tasks and choosing one to work on. Once you have decided to work on an issue, please assign it to yourself. You can do this by:

1. Log into our OFBiz Jira issue tracker
2. Locate the Jira issue you want to work on (Note: Unassigned means that no one is working on it)
3. On the upper right hand side under the 'People' section you will see a link that says 'Assign to me'
4. Click the 'Assign to me' link and the issue will be assigned to yourself

You have now assigned yourself to work on an issue.

=====

Initial Team Setup Tasks

- ☐ Ensure all team members have ICLAs filed (Started 13/02/18 In progress)
- ☐ Collect all Confluence and JIRA ids and ensure contributor permissions are setup (Started 13/02/18 and in progress)
- ☒ Skype call to talk about how to get started : (Details of Skype Call OFBiz Documentation Skype Call: February 2018)

Next Steps:

Based on the discussions the proposed high level roadmap of next steps looks like this

- ☒ Get the Proof of Concept (PoC) documentation framework written by Taher committed into the trunk (Done by Taher 8th March 2018)
- ☒ Identify mentors who will be available to help less experienced documentation contributors (In progress. Michael, Sharan and Olivier have indicated they are available so far)
- ☒ Use a wiki page to act as reference. (Done, this page is being used as that)
- ☐ High level plan to show what is being done
- ☐ a reference or FAQ for how to get started,
- ☒ Details of the process that we want to follow and also a list of available mentors etc)
- ☐ Define a Table of contents structure for each application (In Progress: Initial one is Human Resources)
- ☐ Mentors will create the document structure within OFBiz (some files with data, some empty)
- ☐ Create Jira tasks for the outstanding documentation work
- ☐ Create some documentation and rules for 'how to write the documentation' (how to use asciidoc syntax for specifics situation, how to use asciidoc syntax for specifics situation)
- ☐

TASKS IN WAITING

This area is for work that the team will need to do but not yet!

Remove markdown files added to Birt in the following commit and incorporate it into the documentation framework: <https://s.apache.org/eTqQ>

BRAINSTORMING AREA

This area is used for adding ideas and suggestions for brainstorming. If the idea or suggestion is move into the task list then it can be deleted from here. Also remove anything that is not relevant.

We can start, in the same time as other documents (Comment from [Sharan Foga](#) in response to this. Once we have done an example together so that everyone knows the process and how to work, then we can split off and do parallel work. Trying to do parallel work at the start, I think will cause problems until people are confident enough to work alone)

Documentation Reference for Contributors

What tools will they need to install and use

What process will they need to follow?

What templates will they need to use?

Where are examples of what the documentation should look like?

At the beginning this page will be more a draft than a documentation but it can be help us to see if our documentation is usable by us 😊

Doing a small modification on showHelp view to have a header with a link a the main manual (which have links to all the other files 😊)

[Michael Brohl](#):

- We should set up a small documentation quality team which is responsible that the documentation send in by contributors is of good quality, consistent and (mostly) error free. This team should also propose the structure of the documentation and maybe add the empty .adoc files and includes to lead the way for contributors.
- It could also be a good idea to track who is working on a piece of documentation to avoid double work and encourage collaboration between contributors interested in the same topic(s). I think this might be Jira with a main task containing the main "rules" of documentation and several subtasks for each topic.
- I suggest to have also a process to move documentation from the Wiki to the documentation in the repository. We should make sure that we do not have different documentation in both places. If someone is working on a topic he also could search the Wiki for it and try to merge/move it where applicable. Moved/merged documentation from the Wiki should then be moved to an adoc Attic until everthing is cleared and the community agrees to remove it from the Wiki.
- We should also define which contents should be in the repository and what the Wiki should contain or be linked to.

Sharan Foga

- To get started we need to focus on co-ordination, structure and organisation (we have a lot of people wanting to contribute but their efforts need to be co-ordinated)
- We need a visual high level plan (see example roadmap above) so that everyone can quickly see progress and main milestones
- Need to define the scope of the effort (maybe do this by saying what we are not going to do??, eg. not an FAQ, not a tutorial, not a use case, not a cookbook) so will be feature documentation (i.e. describing what we have available)
- Look at using JIRA as main co-ordination and tracking tool. We could look at usin a new label eg 'documentation' in conjunction with the existing components. Maybe update workflow to have a new QA status
- Assumption is that the work will be done in the trunk (since 17.12 branch already created) and will be part of the 18.xx branch. Do we look at backporting to 17.12 ???
- Ways to get started working together : Let's all work together on one component (suggest HR since a complete HR manual already exists. It will need review and maybe rewrite but the main content is already there. Each person can be allocated an area and be responsible for submitting a patch

Questions

Which collaborations tools : jira, branch, github and with which rules

How the reader can search on multiple help files ?

How to be able to manage multi-language ? it's not a priority for ofbiz trunk but should be available on customer site (currently it's manage by content multi-language capabilities)

Automatic translation for one file from docbook to asciidoc