



## Publishing WebHelp

### What's covered?

How to publish webhelp produced by RoboHelp and how to keep the server free of unused files.

June 2019 - This page covers the steps for use with RoboHelp Classic versions up to and including RoboHelp 2019 Classic. It is mostly still relevant in RoboHelp New UI but the RoboHelp Tour has specific information about the process and the dialogs. Note that output also offers SFTP as a publishing option.

### Generating and Publishing - What's the difference?

**Generating** is about RoboHelp converting the content of your source files to HTML code that can be interpreted by different browsers.

**Publishing** is about taking those output folders and files and putting them onto a server.

Whenever you generate, RoboHelp empties the target folder and creates a completely new copy of the output. In many companies, you will simply zip up the generated folders and files and the developers will publish to the server. This topic is about what you need to do to if it is your job to publish the help.

You could copy all the generated files to the server outside RoboHelp but that would mean uploading all the files rather than just those that have changed. What a lot of people fail to take into account is that the files to be uploaded are not limited to just the topics you have changed. A lot of "internal" files change too - for example, the search files will be completely rewritten even if only one topic has changed. Identifying all the changed files is difficult without using the publishing function or some similar software.

It is much easier if you use the publishing function built into RoboHelp as various checks will be made and, if you select not to republish all, RoboHelp will just update what has changed.

The key points with publishing if you use the RoboHelp Publish function are:

- Nothing is deleted from the server.  
Any files that are no longer part of the build remain on the server until you delete them manually.
- Unless the Republish All option is ticked, only updated files are published to the server.  
This is by design to minimise publishing time.

### Using RoboHelp

Publishing is about taking your generated folders and files and putting them onto a server. If you use the publishing function built into RoboHelp, various checks will be made and, if you select not to Republish All, RoboHelp will just update what has changed.

I normally use the wizard to publish my help as well as to generate it. The main points to note are some settings that can otherwise cause confusion.

Setting	About
Optimise Speed	<ul style="list-style-type: none"> <li>• If you are publishing to a web site that can be accessed via the internet, then in theory you should use the web site option.</li> <li>• If you are publishing anywhere else, you use the intranet option.</li> </ul>

	In practice it has been found that using the intranet option has resolved more than a few poor performance issues with help accessible via the intranet.
Check for deleted files	<p>This will cause the publishing wizard to check for files that have been deleted from the server. It can only be selected if the Republish All option is not ticked.</p> <p>When Republish All is not ticked, normally only updated files will be published. If you tick this box, the wizard will also check to see if any files have been deleted from the server and if it finds any, it will then also publish those as well.</p> <p>It does not look to see what files are on the server that are no longer required.</p>
Republish all	<p>This causes all your project files to be replaced on the server but I repeat it does not delete any files that are no longer part of the project.</p> <p>Deleting needs to be managed separately. You can either periodically delete everything from the server or use SyncBack just to compare folders. Care is needed if you have files on the server that are not part of the RoboHelp output.</p>

One of the common complaints about the publishing wizard is the time it takes to upload to the server. You have to be fair to RoboHelp as it is either doing a partial audit of the whole project and uploading what is required, or it is uploading the whole project. Both will take some time. Upload speeds are slower than download speeds and often authors do not realise just how slow a process uploading can be.

Using RoboHelp is the simplest method for publishing but if you want to try others, read on.

## Using SyncBack

At one time I found RoboHelp was taking an inordinate amount of time to publish so I needed to find another solution until I could locate the cause. I realised that the SyncBack program I use for backing up my work had the answer. The name SyncBack comes from the two parts of the program, **S**ynchronisation and **B**ack Up.

The first pass may take a bit of time if your server housekeeping has not been too clever but after that updating is very simple and, importantly, the process will check for files that should be deleted from the server thereby ensuring the site is tidy at all times. You don't need to be a programmer. Any author should be able to manage this.

You will need download SyncBack from [www.2brightsparks.com](http://www.2brightsparks.com) (I think that is an award winning site name, still makes me smile).

## The Process

### First Time

Identify files that should not be on the server.

Create a SyncBack profile to synchronise the generated output with the published output on the server. Until you have run the profile a few times, run it in Simulation mode so that no changes are made. There are a number of options in SyncBack to deal with the differences. Initially set them all to Prompt.

When you run the profile (simulated) you will get an on screen report. Where the file exists in both locations you will most likely see File Collision reported. At this stage, ignore that. What you are looking for is where SyncBack reports that the file is only on one side of the synchronisation. In most cases it will be files on the server that are not in the generated output. Unless they are there for valid reasons, delete them and keep running the profile until only File Collisions are reported or you can account for the differences.

At this point, the server has all the right files although the process has not checked they are the right versions.

Get a copy of the published files onto your hard disk.

This is needed for a second profile you need to create in SyncBack. The second profile is used to compare this local copy of the published output with the server copy later in the process.

Why do you need a local copy? Generating always creates new files so every file date and time will be new. If the profile compared the local Generate folder with what is on the server, all the files would be reported as changed. In reality only some of them are different so what you need is to identify the changed files. If you generate and then publish locally with Republish All deselected, what you have in the local publish folder will be some files with old dates and some with the the new publishing date. The latter are what you need to get onto the server.

Publishing locally is a quicker process than publishing to a server so overall this method works well.

Run this second profile in simulated mode. It should not find any differences at this stage as it is a copy of what is on the server.

## Normal Use

Generate and Publish the WebHelp

Generate to the usual location but publish to the folder you created on your hard disk, not to the server. Make sure that the Republish All option is not ticked as the whole process relies on identifying just the changed files.

Compare the generated and locally published help

This step is to ensure that you are alerted to any differences between what is in the generated folder and the local published folder.

Remember that the generated help folder gets cleaned out every time you generate whereas the published folder does not. Comparing these folders allows you to check differences are valid and to remove any files that no longer need to be published.

Amend the SyncBack profile that you used earlier to compare the content of the generated output with what was on the server so that it now compares the generated output with what is in the **local** published folder. Run this profile in simulated mode only as you did at the beginning. We are just identifying files that only exist on one side of this synchronisation. There should never be files that are only in the generate folder but there may be files identified in the locally published help that are not in the generate folder. Use Windows Explorer to delete any files that should not be in the locally published help folder.

Compare the locally published help with the server

It is this step that updates the server with new files and, importantly, can remove files that are no longer required on the server.

Run the SyncBack profile that compares the source (the publish folder on your hard disk) and target (the server). Do this in simulated mode at first if you prefer. However you may find it more efficient to run it in normal mode; SyncBack reports the changes it is going to make before it makes them and gives you the chance to change the action for each file or to abort.

## Other file comparison tools

You can use other file comparison tools if you prefer, just follow the general principles outlined here.

### FTP

You can of course update your server by using FTP to transfer the topics to the server. That may be the quickest option and is fine if you have the expertise to know what needs to be updated.

What you don't get if you FTP is an audit. Also it is just too easy to not update some of the internal files that hold search data and so on.

For most authors the best advice is not to use an FTP client other than for minor topic changes.

## Donations

If you find the information and tutorials on my site save you time figuring it out for yourself and help improve what you produce, please consider making a [small donation](#).

## Topic Revisions

Revisions