

Configuring

[Configuring](#) / [Preparing for job submission](#) / Preparing for FTP or SCP clients

Preparing for FTP or SCP clients

To get a job into a workflow, an application can copy or send the input file into a hot folder using a variety of methods, including File Transfer Protocol (FTP), Secure Copy Protocol (SCP), and RICOH ProcessDirector web services.

For security reasons, many administrators disable FTP on their Linux systems. For file transfers, they use SCP instead. Some makers of FTP software are aware of this practice and have added functions to their FTP clients to let them try to use FTP first. If FTP fails, the software tries other methods, including SCP.

For RICOH ProcessDirector, it does not matter which protocol you use to transfer the data files to the input device directory. The most important thing, regardless of the mechanism you use, is to make sure that the job is submitted using a user ID and password that have the correct permissions to let RICOH ProcessDirector access the file.

During installation, RICOH ProcessDirector creates a RICOH ProcessDirector group (**aiwgrp1** is the default) and gives the RICOH ProcessDirector system user (**aiw1** is the default) authority to this group. You can assign the group and system user names or use the defaults. Any user who is in the RICOH ProcessDirector group can access files that RICOH ProcessDirector creates. If you have users with user IDs on the primary computer who need to work directly with RICOH ProcessDirector files (such as users who submit jobs using FTP or SCP), you must add their user IDs to the RICOH ProcessDirector group. Be sure to use the RICOH ProcessDirector group name as an additional group for your users, not as their default group.

When you explain how to submit jobs using FTP (if you have FTP enabled on your system) or an FTP client that uses SCP, tell your job submitters that they must be logged in with a user ID that is a member of the RICOH ProcessDirector group. If they use any other user ID, RICOH ProcessDirector does not have the correct access permissions set on the input file and cannot process the job. The job stays in **Error** state in the **Receive** phase until it is deleted and resubmitted correctly.

Parent topic: [Preparing for job submission](#)