



«File Converter» Project

Defect Reports Sample **SAMPLE**

This is **NOT** a part of project documentation! This is just a way to demonstrate some defect reports.

Defect report sample 1:

ID	Summary	Descriptions	Steps to reproduce
19	Infinite loop on input file with read-only attribute	<p>If an input file has the “read-only” attribute, the app can not move the processed file into the destination directory: so the app processes the file again and again and thus falls into the infinite loop.</p> <p>Exp: the processed file is moved from the input directory to the destination directory.</p> <p>Act: the processed data (new file) appears inside the destination directory, but the original file is not deleted from the input directory.</p> <p>Req: DS-2.1.</p>	<ol style="list-style-type: none"> 1. Place a valid file (size, type) into the input directory. 2. Set the “read-only” attribute on this file.. 3. Start the app. <p>Bug: the processing result appears inside the destination dir (and the file is repeatedly updated according to the last write time), but the original file stays inside the input directory.</p>

Reproducibility	Severity	Priority	Symptom	Workaround	Comments	Attachments
Always	Medium	Normal	Incorrect operation	No	If the customer has no special plans for using “read-only” attribute on files in the input directory, the Easiest solution is to remove the attribute once it is detected.	-

Defect report sample 2:

ID	Summary	Descriptions	Steps to reproduce			
27	Non-empty subdirectory in SOURCE_DIR is deleted w/o any warning	<p>The app doesn't produce any warning and deletes any non-empty (empty as well) subdirectory in SOURCE_DIR directory.</p> <p>Exp: if the app detects a non-empty subdirectory inside SOURCE_DIR, it produces the error message and stop. The message should be: "Non-empty subfolder [name] in SOURCE_DIR folder detected. Remove it manually or restart application with --force_file_operations key to remove automatically."</p> <p>Act: any non-empty subdirectory (along with all its contents) in SOURCE_DIR is deleted without any warning.</p> <p>Req: DS-8.5.12.</p>	<ol style="list-style-type: none"> 1. Create a directory with any valid name (e.g. TEST). Create a subdirectory (e.g. BUG) inside it and place some files inside this subdirectory. 2. Start the app with the SOURCE_DIR parameter pointing to the TEST directory (see the step 1), e.g.: "php file_converter.phar c:\TEST c:\OUT". <p>Bug: the subdirectory BUG is deleted with all its contents.</p>			
Reproducibility	Severity	Priority	Symptom	Workaround	Comments	Attachments
Always	Major	Normal	Data loss	No		-

Defect report sample 3:

ID	Summary	Descriptions	Steps to reproduce
41	File processing stops after 0.5 hour of work	<p>After something about 30 minutes of work the app stops reacting to the new valid files inside the SOURCE_DIR.</p> <p>Exp: the app continues to process the new valid files in the SOURCE_DIR until it is stopped by an appropriate command.</p> <p>Act: the app still utilizes some system resources (CPU, RAM, drives) but does not react to any new valid file inside the SOURCE_DIR.</p> <p>Req: UR-1.</p>	<ol style="list-style-type: none"> 1. Start the app and let it work for at least 30 minutes. 2. Put a valid file inside the SOURCE_DIR. <p>Bug: the app does not process the file.</p>

Reproducibility	Severity	Priority	Symptom	Workaround	Comments	Attachments
Always	Critical	Normal	System crash	No	<p>It looks like some descriptors of file system objects are closed automatically due to some timeout.</p> <p>This behavior may have the same cause as in bug 38 (once started, the app does not see that SOURCE_DIR and/or DESTINATION_DIR are deleted).</p>	-

Defect report sample 4:

ID	Summary	Descriptions	Steps to reproduce
2352	Multiple instances crash on competition for the SOURCE_DIR	<p>With 2+ instances of the application running with the same input directory configured, they compete for files in this directory that leads to multiple file operation failures and (finally) to crash of some instances.</p> <p>Exp: all instances work (not crashes); if a file operation fails, the affected instance makes a log record and retries the operation.</p> <p>Act: the affected instance crashes.</p> <p>Req: UR-12.</p>	<ol style="list-style-type: none"> 1. Start several (3-5-7+) instances of the app with the same valid parameters. 2. Start putting valid files into the SOURCE_DIR. <p>Bug: one by one most instances crash.</p>

Reproducibility	Severity	Priority	Symptom	Workaround	Comments	Attachments
Always	Critical	Normal	System crash	No	It looks like some semaphore or synchronization mechanism is either broken or do not take the SOURCE_DIR into account. Or some file operation failure handler does not work as it has to.	-