



**White Paper**

## **The “Watch Folder” pattern**

A Workflow Design Pattern

By Howard Treisman

Senior Technical Director

October 2006

Copyright © 2006 – Avoka Technologies <sup>(1)</sup>

[www.avoka.com](http://www.avoka.com)

## Overview

It is sometimes extremely useful to be able to initiate a workflow when a file arrives in a particular folder. This pattern is sometimes known as a "Watch Folder".

This paper describes how to achieve this using a series of QPACs designed and developed by Avoka Technologies.

These QPACs can be downloaded and used for free for development and testing purposes from Avoka's web site at: [http://www.avoka.com/avoka/qpac\\_library.shtml](http://www.avoka.com/avoka/qpac_library.shtml)

## In a nutshell

The fundamental idea is to separate out the "Watcher" process from the "Real" process that does the actual work. The Watcher process uses the Folder Watch QPAC to wait for the arrival of a file matching some pattern. It then extracts some data from the file (which in this case is a simple properties file), and invokes the Real process, passing data from the properties files as parameters. It then moves the processed file into a sub-folder, and loops around to wait for the next file.

## The Problem

We have a simple pre-existing application that allows members of the public to interactively ask questions during an on-line event. When someone asks a question, they are prompted for their name, email address, and the question. When they click OK, the application generates a simple file which it places into the "c:\incoming" directory.

Our application will monitor the "c:\incoming" directory, and whenever a question arrives, allocate it to a user to answer. Once the question has been answered, the workflow will send an email directly to the person asking.

## The Watcher Process

The Watcher process is shown below.

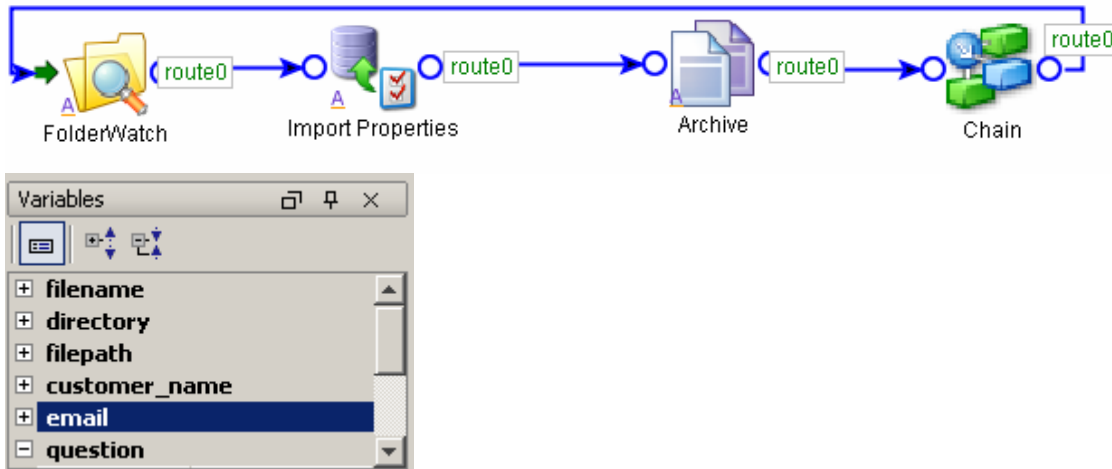


Fig 1: The Watcher Process

The steps are:

FolderWatch	The FolderWatch QPAC monitors a designated directory for the arrival of a file. The QPAC will check every minute (this is configurable). If nothing is found, it will wait for another minute. If a matching file is found, the FolderWatch QPAC stores the filename, directory and full pathname into process variables, and then continues to the next step.
Import Properties	The Import Properties QPAC opens the file (which in this case is a simple properties file), and reads the contents into process variables.
Archive	Once the properties have been read, we no longer need the original file. To avoid re-processing it, we can either delete the file, or archive it. Archiving is preferable, so that we retain a copy of the original file for historical purposes.
Chain	We now use the Chain QPAC to instantiate a new process to do the actual work. (See below.) The chained process will execute asynchronously (i.e. in parallel o the WatchFolder process), so control is returned immediately.
Repeat	The workflow loops around back to wait for another file.

```

1 customer=Avoka Technologies
2 email=info@avoka.com
3 question=How do I implement a watch folder process

```

Fig 2: Simple properties file

## The “Real” process

The Real process really does anything that it needs to do. It will usually have data passed into it from the Watch Process. In this case, we will populate a form with the passed data, assign the form to a user to answer the question, and then email the results back to the end-user.

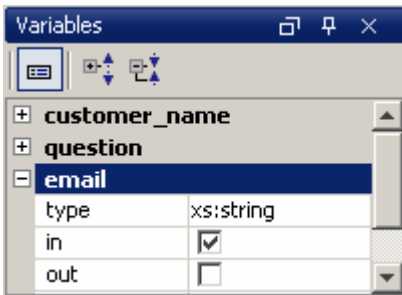
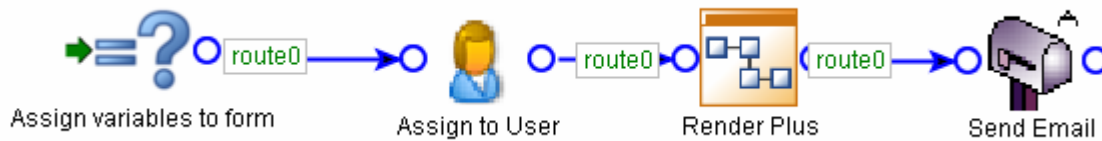


Fig 3: The “Real” process

The steps are:

Assign variables to form	Using the standard SetValue QPAC, we assign the input variables customer_name, email and question to fields in a form variable.
Assign to User	Using the standard User QPAC, we assign the form to a user to answer the question.
Render Plus	Using Avoka’s RenderPlus QPAC, we Render the form as a PDF, and store the result in a document variable.
Send Email	We then send the rendered form to a user, using the Email with Attachments QPAC.

## The FolderWatch QPAC

Using the FolderWatch QPAC is very straight-forward. This is shown below.

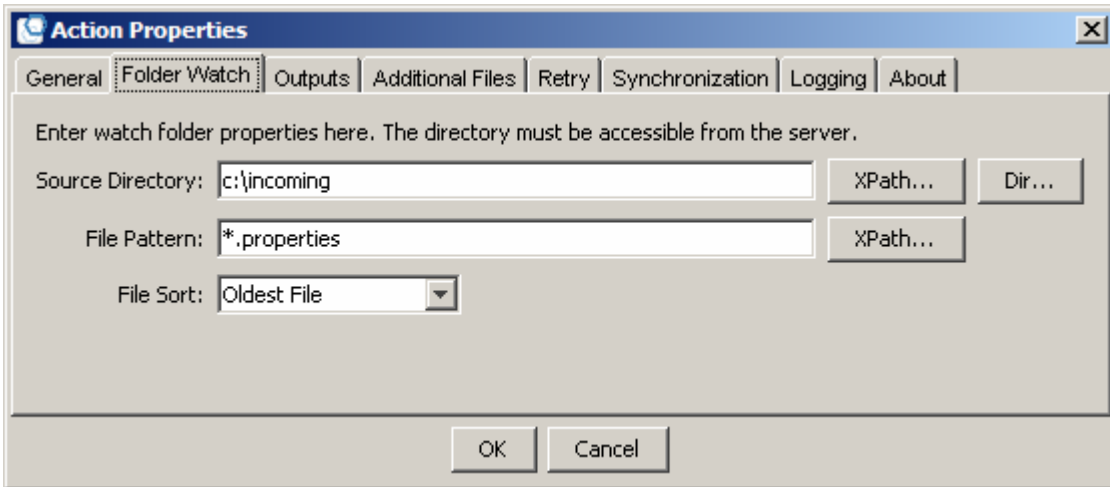


Fig 4: Folder Watch Tab – Specify the directory and file pattern

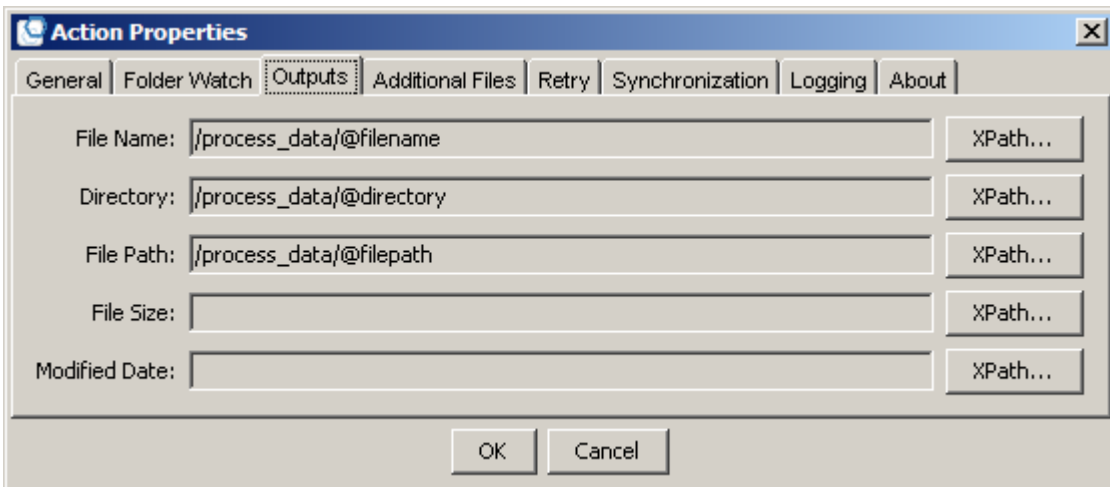


Fig 5: Outputs tab – Store the filename, directory and file path in process variables

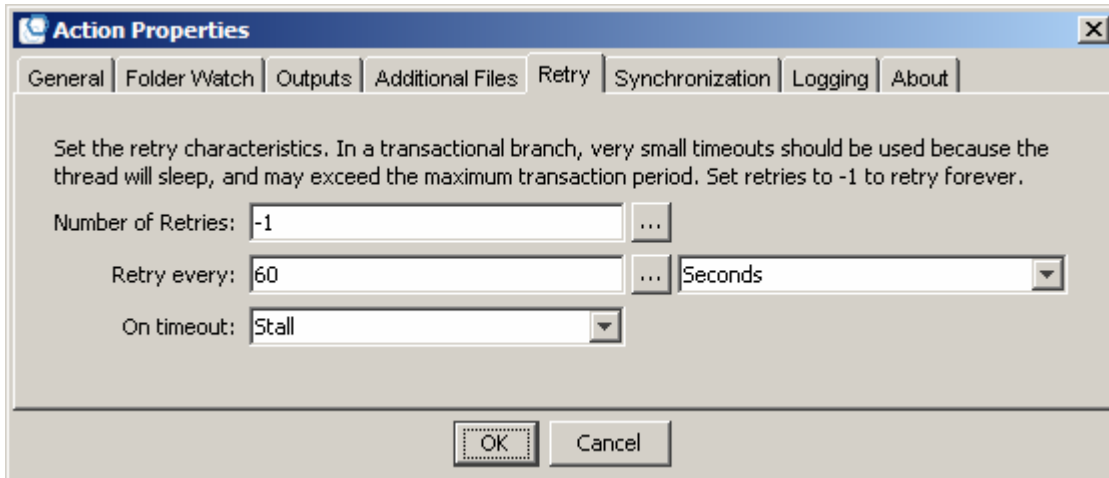


Fig 6: Retry tab –Retry every 60 seconds, and keep trying forever

There are some additional advanced features that allow you to specify “companion files”, and control synchronization and logging. We won’t worry about these for our simple process.

## Import Properties QPAC

The Import Properties QPAC is very simple to use – it simply imports the data from the properties file (see Fig 2) into process variables.

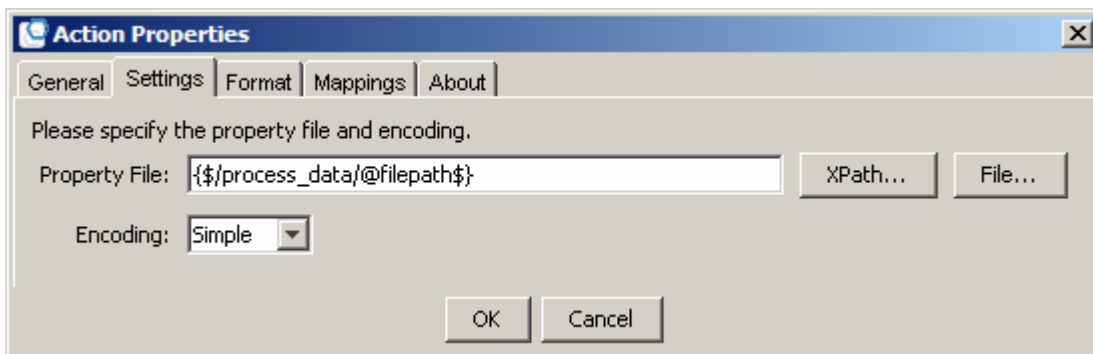


Fig 7 – Specify the properties file to import

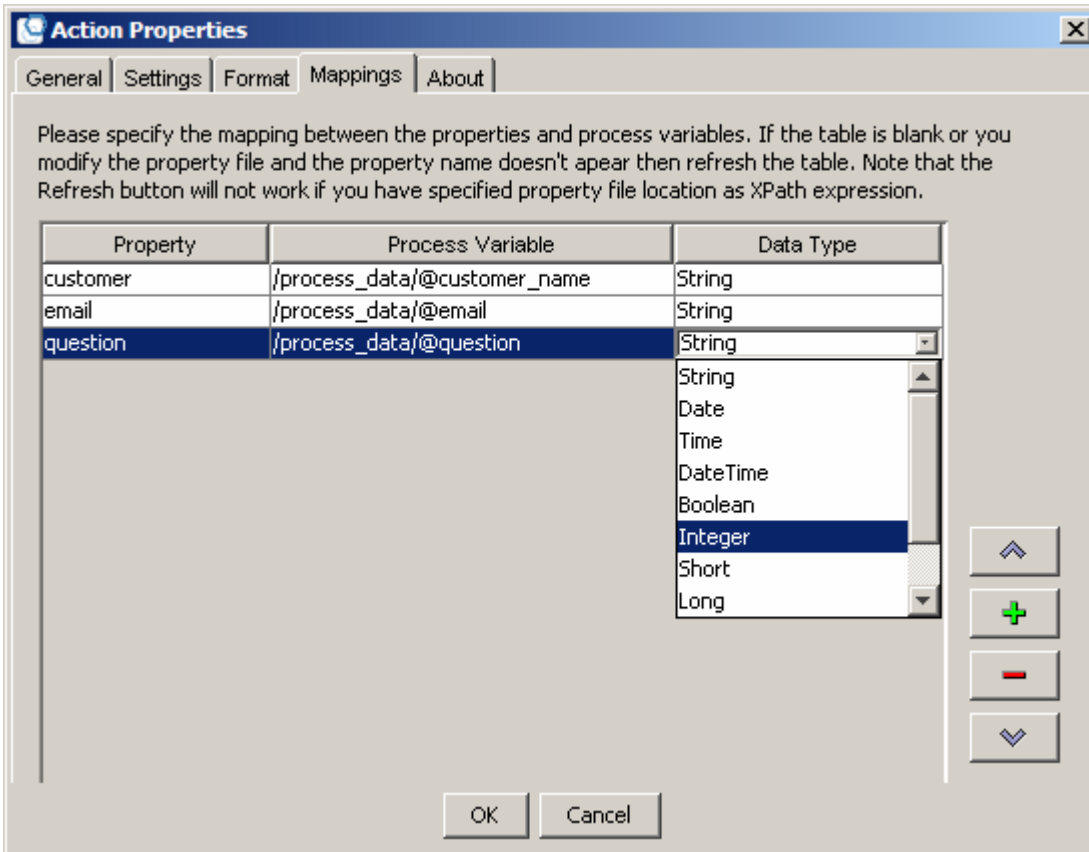


Fig 8 – Map property names to process variables. Property names do not have to necessarily match process variable names.

## The FileMove QPAC

Moving the file to an archive directory is very simple, as shown below. Note the use of the "Append to sub-directory with Date as directory name" option. This will ensure that each day's files go into a separate, uniquely named directory, thus making it easy to locate files historically. There are some advanced logging features which we will ignore for now.

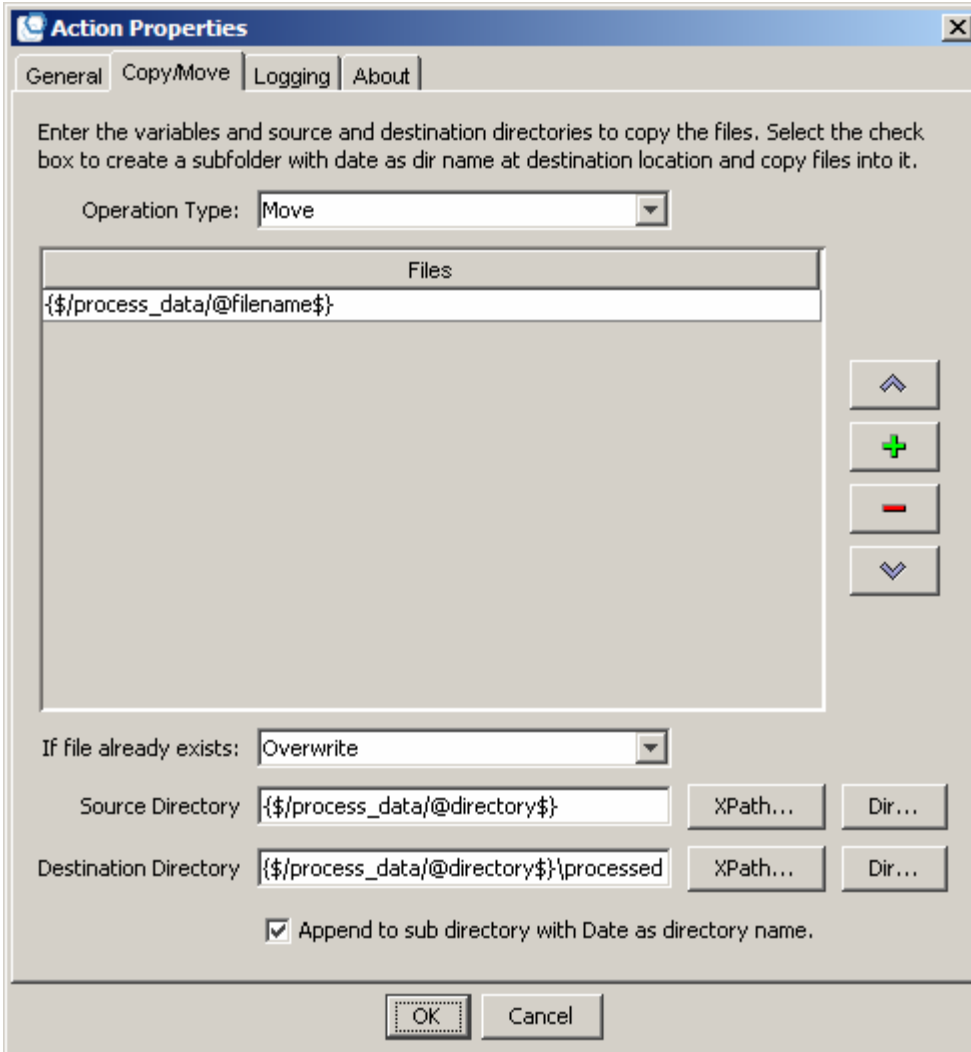


Fig 9 – FileMove QPAC

## The Chain QPAC

The chain QPAC simply invokes the “Real” process, passing the values obtained from the properties file to the instantiated process.

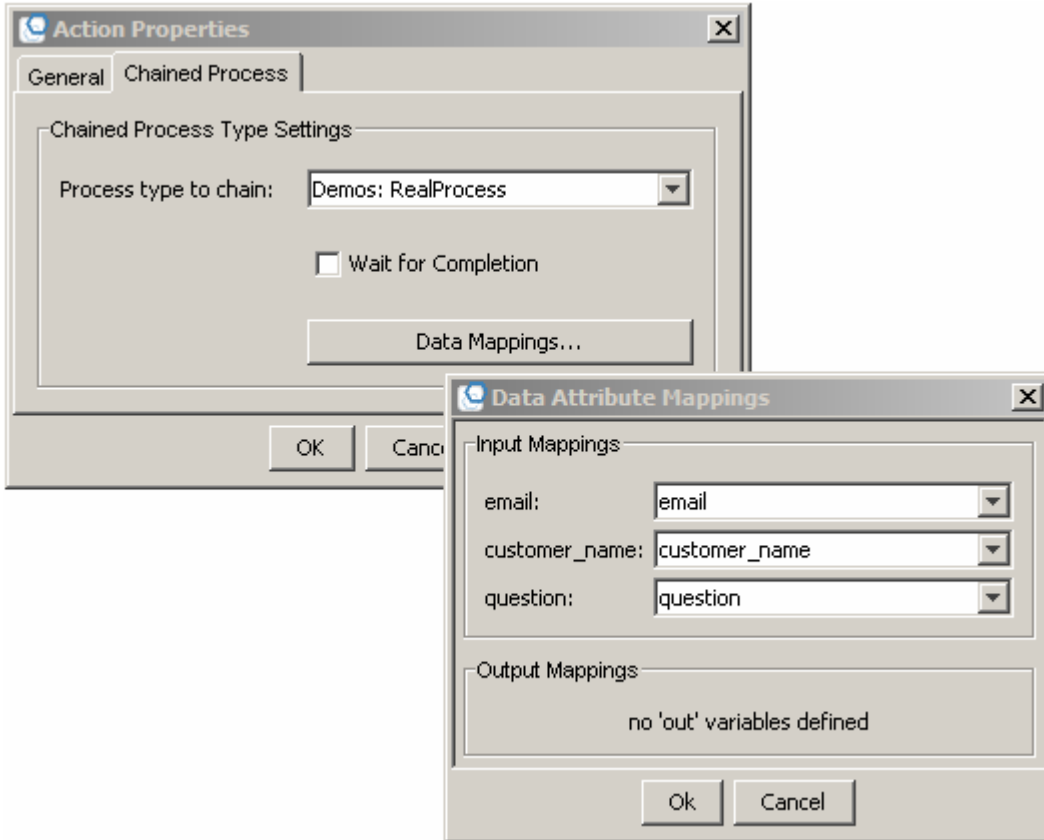


Fig 10 – Chain QPAC

In this case, we have used the same variable names in the WatchFolder process and the Real process. This helps to keep things clear, but is not required.

We do not check the “Wait for Completion” checkbox – we want the chained process to continue asynchronously, and for the Watch Process to immediately loop around and watch for another file.

## Render Plus QPAC

The Real Process uses mostly off-the-shelf components, and we won't go into those in detail. The only one requiring some additional explanation is the RenderPlus QPAC. This QPAC is similar to Adobe's RenderForm QPAC, but is easier to set up and configure. One of its tabs is shown below.

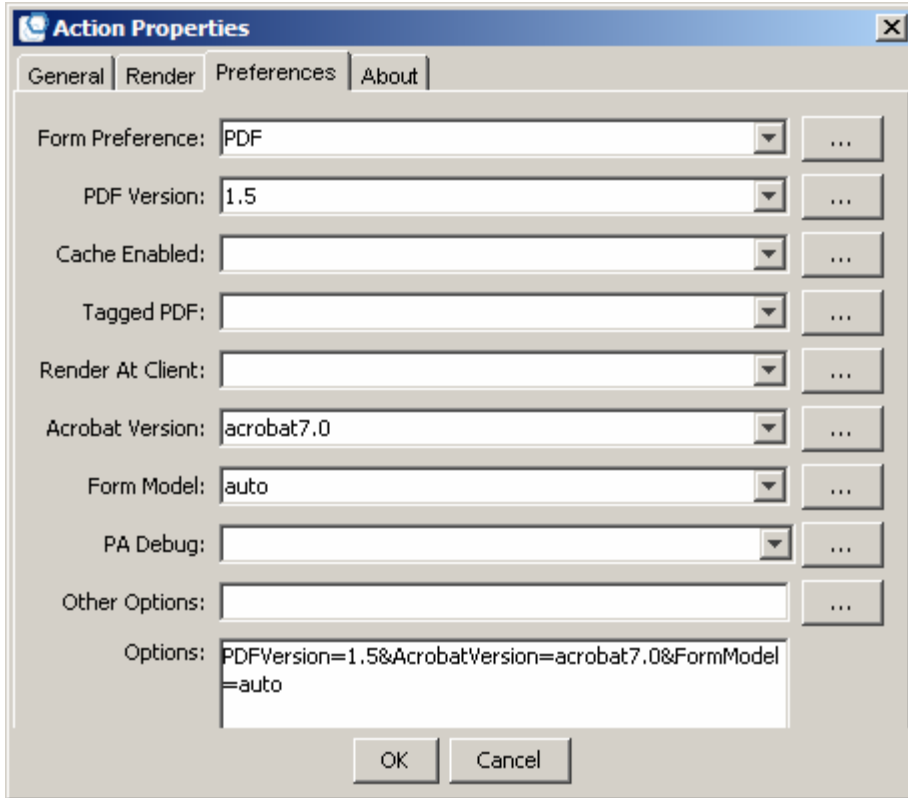


Fig 11 – RenderPlus QPAC

## Advanced features

Several of the QPACs in this document have advanced features that have not been described here. Please refer to the QPAC documentation for more details.

## Summary

The technique described above is a very simple yet effective way of invoking a workflow based on the arrival of a file in a folder. This approach can be modified to perform a variety of useful functions. Many of the QPACs described can also be used for other purposes.

(1) – this document may be freely copied and redistributed, as long as it is reproduced in its entirety, including the above copyright notice.

## About the Author



Howard Treisman is the Technical Director of Avoka Technologies, an Adobe Enterprise Partner based in Sydney Australia.

Howard has been working with the Adobe LiveCycle Workflow product (and its predecessor) for four years, and the LiveCycle suite for the last two years. He has been instrumental in delivering several enterprise-class LiveCycle applications to customers in Australia and internationally.

In addition, Howard spear-headed Avoka's development of over 50 QPACs (re-usable workflow components), many of which have been developed in direct response to customer needs, as well as the Workflow QPAC development environment for Adobe.

Howard's hands-on experience has provided a thorough understanding of the intricacies of the LiveCycle technologies, and how to apply these technologies to solve real-world problems.