

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
 HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
 Gestor de contenidos (Alfresco)
 Aplicaciones híbridas

Tareas programadas (Quartz)
 Gestor documental (Alfresco)
 Inversión de control (Spring)

Control de autenticación y
 acceso (Spring Security)
 UDDI
 Web Services
 Rest Services
 Social SSO
 SSO (Cas)

JPA-Hibernate, MyBatis
 Motor de búsqueda empresarial (Solr)
 ETL (Talend)

Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

BPM (jBPM o Bonita)
 Generación de informes (JasperReport)
 ESB (Open ESB)



AdictosAlTrabajo.com

"¡La primavera ha venido,
nadie sabe cómo ha sido!"



Soporte a desarrollo informático

Entra en Adictos a través de



E-mail

Contraseña

Entrar

Regístrate
Olvidé mi contraseña

[Inicio](#) [Quiénes somos](#) [Formación](#) [Comparador de salarios](#) [Nuestros libros](#) [Más](#)

» Estás en: [Inicio](#) [Tutoriales](#) Imprimiendo documentos Office y PDF existentes con Java en entorno Windows....



Alberto Moratilla Ocaña

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software. DEA en doctorado en Información, Documentación y Conocimiento. MBA en empresas de Nuevas Tecnologías. CQF.

Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)

Fecha de publicación del tutorial: 2015-04-07

Tutorial visitado 2 veces [Descargar en PDF](#)

Imprimiendo documentos Office y PDF existentes en una red Windows desde Java con Batch & Print

0. Índice de contenidos.

- 1. Introducción
- 2. Entorno
- 3. Impresión de documentos existentes desde Java en una red Windows
- 4. Otras soluciones. Pensamiento lateral
- 5. Instalación y Configuración Básica de Batch & Print
- 6. Conclusiones

1. Introducción

En este tutorial vamos a ver cómo solucionamos un problema aparentemente sencillo, imprimir documentos existentes en formato Office y PDF en impresoras conectadas a una red Windows, empleando enfoque de "pensamiento lateral" ante la imposibilidad de lograr una solución satisfactoria con las tecnologías que estábamos empleando en un proyecto.

Ya hace tiempo que la tecnología Web le está ganando la batalla a las aplicaciones de escritorio en diferentes campos, incluido el campo de la gestión. Los típicos ERP desarrollados en entornos nativos, se están viendo beneficiados por las ventajas de los sistemas Web. Es raro que una empresa que decida cambiar sus programas de gestión, se incline por una solución en la que no tenga que usar un navegador.

Pero no todo son ventajas. Algunas de las operaciones típicas que se pueden realizar desde una aplicación nativa en Windows (por ser el más popular en estos entornos), se tornan complicadas a la hora de portarlas a su versión Web. Y no solo complicadas, en ocasiones, podemos decir que son prácticamente imposibles.

2. Entorno

Para realizar este tutorial se ha empleado el siguiente entorno de desarrollo:

- Hardware: Mac Book Pro 15" Intel Core i7 2,8 GHz, 16 GB RAM.
- Sistema Operativo: Mac OS X Yosemite.
- Netbeans 8.0.1.
- VMware Fusion 7.
- Máquina virtual con Windows XP.
- Acrobat Reader X.
- Microsoft Office 2007.
- Impresoras conectadas en Red Windows.
- Diversos ficheros en formato .doc, .docx, .xls, .xlsx, .pdf

3. Impresión de documentos existentes desde Java en una red Windows

Imaginemos un caso en el que un cliente nos hace una petición bastante lógica sobre un desarrollo existente. Su intención era que se añadiese, dado un programa de gestión para su empresa realizado en Spring MVC sobre un servidor Tomcat 7, un usuario pudiese imprimir [documentos Office y PDF ya existentes](#) en un directorio en la red Windows de la oficina por la impresora que quisiera de las configuradas en su red (también Windows).

Todo ello sin intervención desde el ordenador del cliente puesto que tenía que ser una operación automática y no se podían permitir la pérdida de tiempo que supone descargar los documentos e imprimirlos en local. Si pensamos que esta aplicación Web de gestión ha sustituido a un programa de escritorio realizado en .NET, seguramente la petición cobre más sentido.

El problema no es sencillo. Si lo analizamos, tenemos los siguientes problemas destacados:

- La orden de impresión debe ser ejecutada desde el servidor, que es un Tomcat 7. Afortunadamente el sistema operativo del servidor es Windows Server.
- Los ficheros a imprimir deberían salir con total calidad. Esto es, un Word debe salir tal cual fue escrito con Microsoft Word; lo mismo para Excel o los ficheros PDF. Por tanto, usar una librería de un tercero para interpretar los ficheros es inviable (*Ojo: queremos imprimir ficheros existentes, no generarlos!*).
- Las impresoras no disponen de un protocolo que podamos emplear fuera de Windows: no disponen de un FTP interno

Catálogo de servicios Autentia



Síguenos a través de:



Últimas Noticias

» 2015: ¡Volvemos a la oficina!

» Curso JBoss de Red Hat

» Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris

» Portales, gestores de contenidos documentales y desarrollos a medida

» Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer

[Histórico de noticias](#)

Últimos Tutoriales

» Enfréntate con éxito a la crisis de la hoja en blanco

» Patrones de diseño en Hadoop: Patrón Partitioner

» Movimientos de la cámara

» Introducción a Apache Hive, el datawarehouse de Hadoop

» Mixins en Java y Java8 ¡Sí, es posible!

en el que volcar los ficheros y ser ellas las que interpreten los .docx o los .xlsx.

¿Cómo podemos hacer esto desde Java?

Una tarea aparentemente fácil y de tecnología ya superada (ya hace unas cuantas décadas de la llegada del hombre a la luna), se torna muy complicada si no somos un poco creativos. Básicamente debería hacer lo siguiente:

- *Localizar el fichero o los ficheros a imprimir*. Es sencillo, un par de búsquedas respecto a unos criterios y sin problema, por eso no perderemos tiempo en este tutorial en tratar este tema en profundidad.
- *Abrir y leer el fichero*. Esto es muy complicado o imposible hacerlo tan bien como Microsoft Word. Podemos usar Apache POI u otra librería, pero seguro que hay algunos casos en los que no va a abrir el fichero correctamente.
- *Imprimir por la impresora indicada por parámetro*. Todas ellas conectadas a una red Windows.

Vamos a ver algunas alternativas sobre cómo hacer esta tarea con Java, intentando ceñirnos a esta tecnología, tal y como haríamos para implementar cualquier otra historia de usuario. Se me ocurren las siguientes alternativas:

Alternativa 1: Impresión desde Java usando java.awt.Desktop

Dentro del paquete *AWT* de Java (herramienta de Java para hacer aplicaciones de escritorio), desde Java 6, existe una clase que nos permite acceder a diferentes funcionalidades del sistema operativo en el que se está ejecutando el programa, aunque no permite ningún tipo de configuración.

- Referencia: <http://docs.oracle.com/javase/7/docs/api/java/awt/Desktop.html>

El código para imprimir un fichero de Microsoft Word ubicado en `c:\Documentos\fichero.docx` podría ser el siguiente:

```
package printjava;

import java.awt.Desktop;
import java.io.File;
import java.io.IOException;

public class PrintJava {
    public static void main(String[] args) throws IOException {
        File fileToPrint = new File("c:\\Documentos\\fichero.docx");
        Desktop.getDesktop().print(fileToPrint);
    }
}
```

Lo que hace este código es invocar a la aplicación por defecto que haya instalada en el sistema (Microsoft Word para .docx por ejemplo) y llamar directamente al comando imprimir. Es la responsabilidad del sistema operativo elegir el programa con el que lo va a abrir y elegir la impresora y los settings seleccionados por defecto.

El problema radica en que no se puede indicar a qué impresora queremos imprimir el documento, y tampoco algunos settings que podríamos utilizar. Parece una opción demasiado básica. Además tampoco hay mucho control sobre la cola de impresión.

Alternativa 2: Impresión desde Java utilizando javax.print de Java 2D

Si buscamos algo más sin salirnos de Java, podemos encontrar la API de Java Print Service de Java 2D. Concretamente `javax.print`:

- Referencia: <http://docs.oracle.com/javase/tutorial/2d/printing/services.html>.
- Tutorial: <https://docs.oracle.com/javase/tutorial/2d/printing/>.

Esta librería nos permite indicar la impresora sobre la que queremos hacer la impresión y algunas de sus características, como tamaño de páginas, páginas a imprimir o buscar las impresoras de la red, así como el tipo de documento (`DocFlavor`).

La parte negativa para nuestros intereses es que se comunica directamente con la impresora, sin pasar por un programa que interprete el fichero, y sólo soporta algunos tipos comunes de datos como por ejemplo imágenes (JPEG, GIF, PNG...) o documentos en un formato abierto para impresoras como PCL, PDF o PS. Por tanto al final se trata de un envío de datos directamente a la impresora desde nuestro programa Java, utilizando protocolos de comunicación con la impresora (como PostScript). Esto es algo demasiado primitivo para lo que necesitamos, además de no permitirnos interpretar los documentos que no estén en estos formatos estándar y que eran requisito indispensable.

Alternativa 3: Impresión desde Java utilizando Apache POI u otras librerías que interpretan el contenido

O también otras librerías similares para abrir documentos de Office como la API de Open Office (<http://www.openoffice.org/api/>) o archivos PDF (con la librería `PDFBox` sin salirnos de Apache). Como se ha venido diciendo a lo largo de estos tutorial, no es la solución adecuada porque al tratarse los ficheros de Microsoft Office de tecnología propietaria, solamente con sus productos podremos interpretar de forma adecuada el contenido de los ficheros.

Por tanto, descartamos esta opción en nuestro caso ya que los documentos no son generados por la aplicación, sino que son existentes, y además no podemos realizar conjeturas sobre su complejidad para ser tratados por bibliotecas alternativas a Office (como las indicadas arriba).

4. Otras soluciones. Pensamiento lateral

Con las alternativas descritas anteriormente parece que hemos llegado al límite de la programación en Java con las librerías disponibles tanto nativas de Java como desarrolladas por terceros. Efectivamente, no todo lo podemos hacer con nuestros super poderes de programación en Java. Hemos encontrado el borde de "Matrix". Llega entonces el momento de utilizar otra de las características que tiene que tener un buen desarrollador: el llamado "pensamiento lateral"

El Pensamiento lateral como ayuda para los programadores

(Nota del autor: Aquí expongo una reflexión nada técnica sobre cómo, a menudo, los programadores caemos en un agujero tecnológico a la hora de buscar soluciones, actuando como martillos buscando clavos, cuando en realidad la solución pasa por poner una tira de velcro).

Tenemos una profesión compleja, en la que nuestro trabajo consiste en resolver problemas con un alto grado de abstracción. No quiero decir que seamos más o menos inteligentes, simplemente quiero decir que para entender los problemas a los que nos enfrentamos como programadores no podemos encontrar fácilmente una metáfora con la que comparemos en el mundo real que nos rodea... ¿cómo explicarías a alguien que no sabe nada de informática qué es JPA y por qué es útil, utilizando metáforas o comparaciones que cualquiera pueda entender?

El alto grado de abstracción provoca que nos concentremos demasiado en intentar resolver los problemas de un modo "informático", esto es, con la tecnología del ámbito cerrado que estamos usando en ese momento, obcecándonos en implementar soluciones que no nos llevan a nada en algunas ocasiones. Lo que no puede suceder es que el cliente se quede sin esa funcionalidad porque nosotros estamos demasiado obcecados en una forma de resolver el problema cuando pueden existir otras soluciones si somos algo más creativos.

Esta es la base del llamado "pensamiento lateral", que básicamente consiste en intentar tener ideas que se salgan del ámbito

» Experimenta con tu código en Eclipse utilizando Scrapbooks

» Introducción a Polymer

en el que estamos pensando en ese momento. En una frase: "intentar ser creativos" para resolver un problema en el que nos quedamos atrapados con una idea completamente diferente a lo que estamos pensando en ese momento.

Para ello es importante hacernos una serie de preguntas acerca de lo que queremos resolver: ¿cuál es la necesidad que hay que satisfacer? ¿Qué me ha pedido el cliente realmente? ¿Estoy siendo condicionado excesivamente por la tecnología que estoy usando? ¿Soy yo el que impone restricciones a la solución sin que me lo haya pedido nadie porque no me he parado a pensar en ello?

En el ejemplo que propongo en este tutorial tenemos un problema que resolver: *imprimir desde un servidor Tomcat corriendo Spring MVC una serie de documentos ya existentes en formato Office y PDF en impresoras conectada a su red Windows*. Nos preguntaremos:

- ¿Forma parte de la necesidad del cliente que tengamos que usar algo programado en Java para imprimir?
- ¿Me ha pedido el cliente que imprima documentos usando Java, o que simplemente los imprima, sea cual sea la alternativa siempre y cuando funcione y tenga un coste y un mantenimiento asumibles?
- ¿Soy yo el que con mi visión obcecada en buscar la librería ideal de Java impone que la única solución pase por desarrollar un complejo sistema desde Java?

Parece evidente, tal y como hemos demostrado en secciones anteriores de este tutorial, que como programadores estamos centrándonos en un dominio del problema que parece no tener solución en nuestro pequeño mundo de Java. Y para colmo somos nosotros mismos los que nos autocastigamos poniéndonos las *antojeras* cual caballo para no distraernos con otras alternativas más sencillas. *Sin duda es tiempo de reflexionar y buscar la solución por otros lugares usando la creatividad, el pensamiento lateral.*

Vamos a ver dos alternativas que se nos ocurrieron para cumplir con lo que el cliente nos había solicitado de un modo algo más creativo.

Alternativa lateral 1: Realizar un programa en .NET que actúe como interfaz de impresión

La alternativa más evidente como programadores que necesitan cubrir una necesidad en un entorno Windows, es implementar un programa en .NET con el que poder acceder a funcionalidades nativas de Windows que nos permitirán elegir y configurar impresora y emplear los programas adecuados para cada tipo de archivos (Microsoft Word para los .docx, Microsoft Excel para los .xls, etcétera).

Este programa realizado en .NET podría comunicarse con la aplicación a través de REST o una tabla compartida en base de datos de manera que nuestros problemas con la impresión desde Java se convertirían en desarrollar una comunicación escribiendo en base de datos o invocando servicios REST del programa .NET (por indicar dos alternativas de comunicación). Se trata de tareas más que probadas y con librerías suficientes en Java (es posible que ya las hayamos utilizado dentro de nuestro desarrollo).

La parte negativa de esta alternativa consiste en que debemos hacer el desarrollo del programa en .NET. Por buenos que seamos, si somos especialistas en Java, y por mucho que se parezca C# a Java, al final necesitamos de un período de adaptación y formación que quizá el cliente no quiera asumir. Además trataría de manejar librerías específicas, lo que incrementa la dificultad. Si queremos cambiar algo sustancial o soportar nuevos tipos de archivos, deberemos modificar este programa en .NET.

Si te ves con confianza usando .NET o tienes tiempo suficiente o simplemente quieres aprender cosas nuevas, puede ser una buena opción. O también puedes seguir buscando una solución mejor.

Alternativa lateral 2: Invocar comandos nativos del sistema operativo desde Java

Otra de las alternativas es desistir de usar librerías de Java para imprimir, y emplear el esfuerzo en aprender a hacer llamadas a los programas específicos que manejan cada tipo de archivos a base de comandos del sistema operativo (en este caso Windows).

Hay diferentes formas para hacer esto. Básicamente habría que, usando la shell de windows (cmd.exe), ser capaz de obtener los comandos necesarios para imprimir por cada tipo de archivo, configurando como mínimo la impresora de destino entre las instaladas en el ordenador en el que se ejecutará el comando (el servidor en este caso). Microsoft en su site de soporte dispone de algunos manuales sobre cómo abrir Word en línea de comando y qué parámetros emplear para poder abrir un fichero concreto e imprimirlo siempre que tengamos una macro creada con los pasos adecuados para imprimir a nuestro gusto. (<http://support.microsoft.com/en-us/kb/210565>)

Hacer una llamada al sistema operativo desde Java no es algo complicado, aunque puede haber problemas de concurrencia, con el número de instancias de Word o simplemente puede que no se cierre el programa abierto (Word):

```
Runtime.getRuntime().exec("cmd /c start impresionWord.bat");
```

Esta solución transforma la problemática de imprimir los documentos de Office y PDF en Java en un entorno Windows, a una simple invocación a un comando del RunTime tal y como podríamos hacer con un fichero .bat. Por tanto ya tenemos otra alternativa factible.

La contrapartida a esta solución puede ser que para añadir un nuevo tipo de fichero o cambiar la configuración, habrá que cambiar el programa y alterar los ficheros .bat. Adicionalmente hay que programar una macro en cada programa asociado al archivo, lo que puede ser algo complicado (son fáciles al principio, pero todo se puede complicar). Tampoco parece algo excesivamente estable, aunque como digo puede ser una solución.

Alternativa lateral 3: Usar un programa externo adquirido para realizar la impresión: Batch & Print Enterprise

Finalmente vamos a proponer una tercera solución alternativa que trataremos más en profundidad que las anteriores. *Se trata de una solución comercial de pago, por tanto no es open source* :_(, pero que por un coste relativamente bajo nos va a solucionar el problema de una forma más satisfactoria que las alternativas anteriores.

Se trata de la herramienta *Batch & Print versión Enterprise o versión Pro* (el funcionamiento es el mismo en ambas versiones, sólo es un tema de licencias) que permite mapear un directorio a una impresora, de modo que los archivos que son copiados a ese directorio son impresos de forma automática en la impresora. Por tanto para solucionar nuestro problema simplemente habría que hacer los siguientes pasos:

1. Crear el o los directorios para asignarlos a las impresoras de destino.
2. Configurar Batch & Print para mapear cada directorio a la impresora deseada. Podría darse al directorio el nombre de la impresora destino. Además se especifican otras características como el tiempo de espera para la comprobación de nuevos ficheros o propiedades de la impresora.
3. Copiar los ficheros que deseamos imprimir al directorio que corresponda con la impresora deseada.

Por tanto con este programa, el problema de impresión de ficheros en diferentes formatos de Office o PDF en un ordenador Windows se traduce en configurar la herramienta (que veremos a continuación) y en copiar los ficheros deseados a la carpeta correspondiente a la impresora seleccionada desde nuestro código Java de Tomcat. Algo que podemos hacer fácilmente a través de la clase de utilidades *FileUtils* de Apache Commons. Por tanto estamos transformando un problema prácticamente irresoluble en Java (el proceso de impresión) en algo más sencillo para este lenguaje de programación.

Tiene los siguientes puntos fuertes frente a las otras alternativas:

- Coste: es posible que nuestro coste como desarrolladores que realizan un programa para esta labor (como en la

alternativa de .NET) sea inferior al coste de la licencia. Esto es algo que atañe al cliente, pero que es de suma importancia.

- Reutilización: no tiene por qué ser utilizada solo por nuestro desarrollo. Se pueden configurar más carpetas para ser utilizado por otras aplicaciones o por los propios usuarios directamente.
- Cuenta con otras operaciones que nos pueden ser muy útiles como por ejemplo un log de las operaciones, comunicación con FTP o POP3, o mover los ficheros impresos a una carpeta y los fallidos a otras.

Puede consultarse el listado de features en el siguiente enlace: <http://www.traction-software.co.uk/batchprint/FEATURES.HTM>.

A continuación se mostrará un pequeño manual de configuración del software Batch & Print Enterprise, para mapear un directorio a una impresora conectada a la red de modo que los ficheros que volquemos en esos directorios se impriman por las impresoras adecuadas.

5. Instalación y Configuración Básica de Batch & Print

En este apartado vamos a ver cómo instalar Batch & Print en un ordenador Windows y cómo realizar la configuración básica para poder imprimir los ficheros que copiemos en un directorio, que al fin y al cabo es la necesidad que tenemos que cubrir.

Prerequisitos

- Windows XP o superior.
- 50 Mb de espacio en disco.
- Memoria de 16 Mb de RAM, más la que consuman las aplicaciones usadas para imprimir.
- Programas instalados para interpretar los ficheros: para ficheros de Office hará falta Microsoft Office; para PDF por ejemplo Acrobat Reader, para txt el Block de Notas, etcétera. La interpretación de los ficheros no las realiza Batch & Print sino que las realizan estos programas instalados en el ordenador que lo ejecuta.

Descarga e instalación

1. Nos dirigimos a la página del producto y pulsamos sobre el enlace "Download Now". En el momento de desarrollar este tutorial la versión era la v8.03:

Traction Software
—PDF & Print Tools Specialist

Download v8.03 42mb

Download Now

v8.03 Download (free trial)

64bit version Download
Old Versions

Menu Links
News
Buy / Register **BUY**
Upgrade from
v1/v2/v3/v4/v5/v6/v7.xx
What's New (History)
Trial Restrictions
Full Features List
How to Video Examples
Knowledge Base
Add-on's
Active X (Run from web)
Print Associations
End User License Agreement
How to uninstall
Forum
Contact
batch-print.com
printtopdfpro.com
Batch & Print Enterprise

Home Products Support Order Corporate

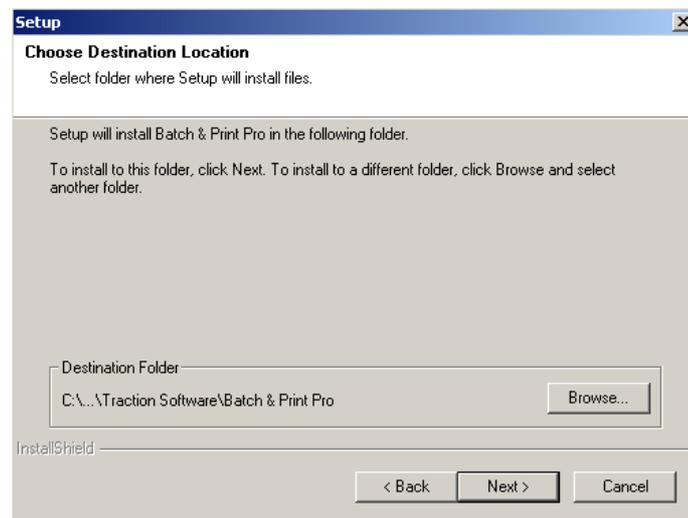
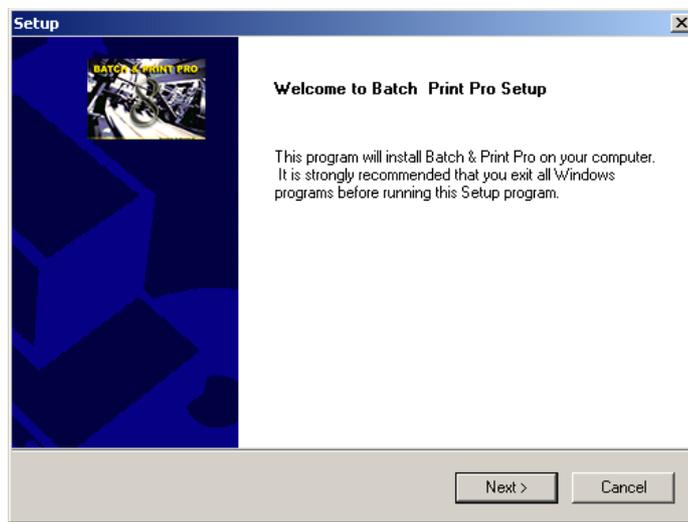
Traction Software's Batch & Print Pro 8.03 is now available for **download**, when Batch & Print Pro 8.03 is installed, click help menu --> what's new for full list of new features in the latest version, check our **comparison chart** if you're thinking of upgrading from v1, v2, v3, v4, v5, v6, v7.xx - see TOP Reasons why you should upgrade today! **here**
Batch & Print Pro Addon's for extra page control and other options specific to image/document types, **click here** for more details: HTML Page & Header/Footer control, PDF Page Control, Word Doc / RTF control, TIFF Page Control + Stamping Header/Footer Text and scaling options, IMG Addon supports BMP, ICON, JPEG, GIF, Exif, PNG, WMF and EMF formats internally, Outlook MSG files Addon, Outlook PST files Addon, Office Doc, Excel, EML, XML, Solid Edge (DFT, DWG, ASM, PAR, PSM, PWD, PCF, DXF), eDrawings (DXF, DWG & SLDDRW), DWF (Autodesk DWF) Addon's, PST Addon for outlook .pst/.ost mail archives, Development Custom SDK available! **click here**
NEW!!! Active X component for running Batch & Print Pro from the Web, click here
NEW!!! Print logging over the web and multiple machines central database logging, click here
Print to PDF with our free PDF Printer Driver **here**
Looking for large scale multi-threaded printing? **click here** for Enterprise version.

What is Batch & Print Pro ? **BUY Trial Download**

Selected	Filename	Pages	Status
Yes	C:\Vest files\WIN00		Done
Yes	C:\Vest files\Sp. Res	13	Done
Yes	C:\Vest files\Vest pd		Error - No print assor
Yes	C:\Vest files\Min08		Done
Yes	C:\Vest files\Vest pd		Done
Yes	C:\Vest files\Invalid		Error - No print assor

Batch & Print Pro Scheduler
Schedule prints to print automatically to any printer at a specified time of day or month or year.

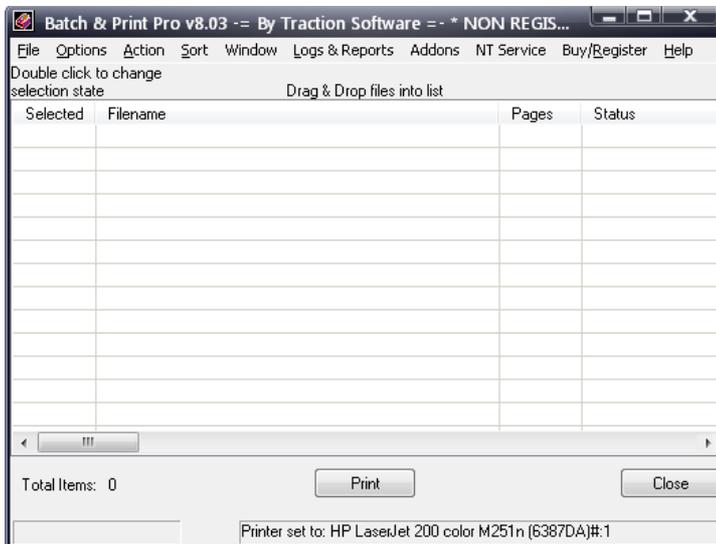
2. Una vez descargado el fichero "BatchPrintProInstall.exe" (el equivalente para la versión Enterprise), hacemos doble click para realizar la instalación. Aparecerá el clásico asistente de instalación para indicar el directorio en el cual vamos a dejar el ejecutable y algunos detalles de la licencia.



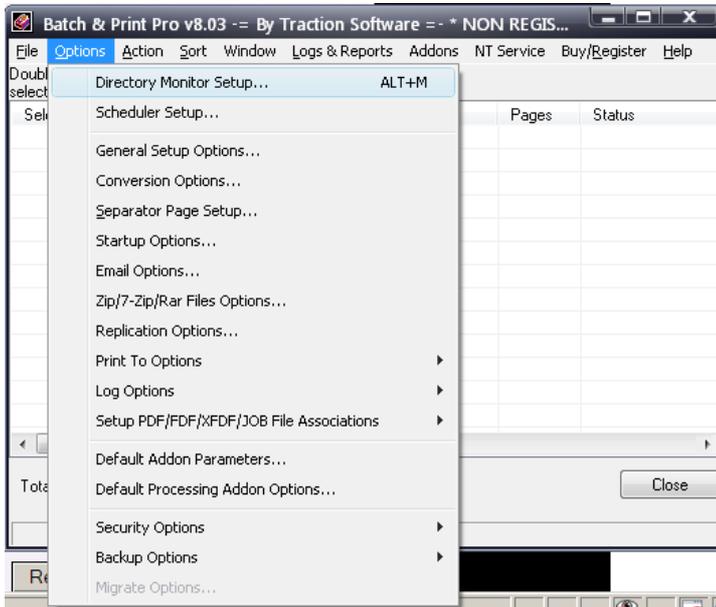
3. Nada más arrancar el programa aparecerá una ventana para la introducción de la licencia ya que estamos usando el trial. Recuerda que no es un programa de código abierto, por lo que si no usamos el tipo de licencia adecuado estaremos infringiendo la ley.



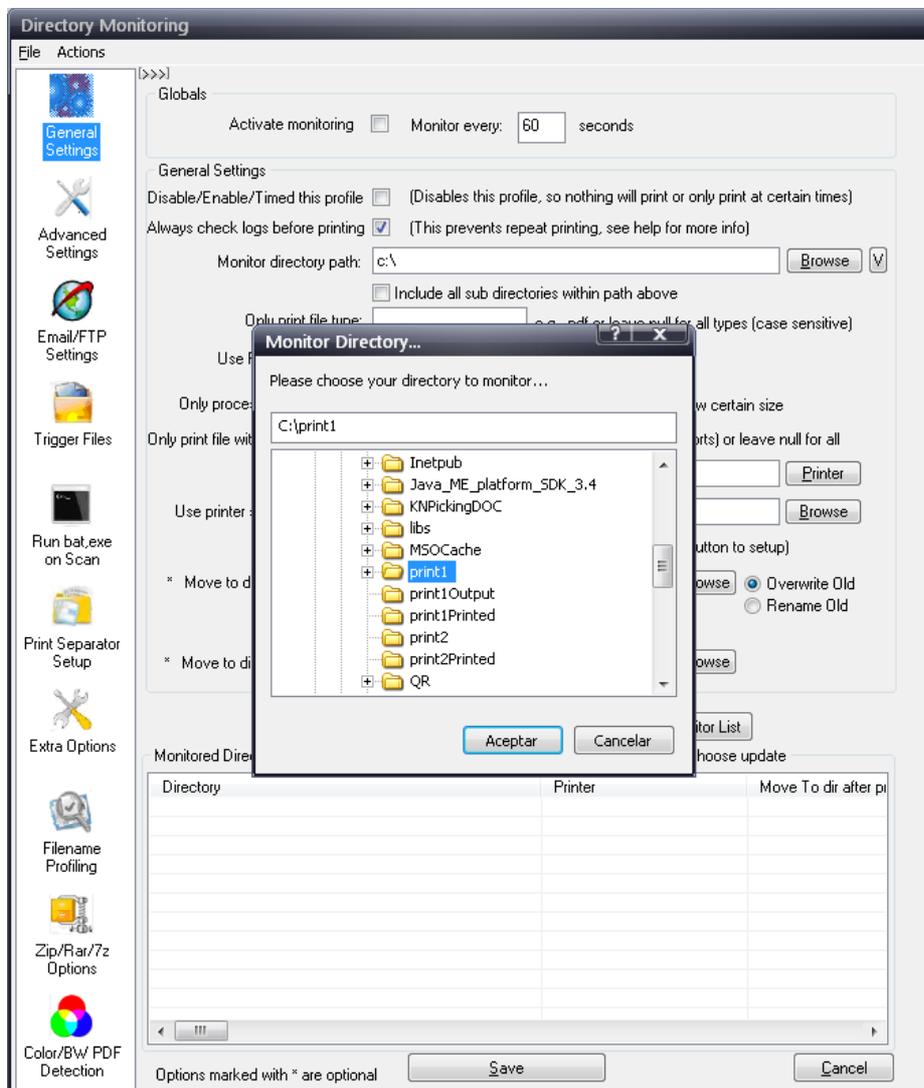
4. La apariencia principal de la aplicación es la que aparece en la imagen a continuación: un listado de documentos a imprimir o que se están imprimiendo, con el número de páginas y su estado. Se añaden arrastrando sobre el programa, o lo que más nos interesa, configurando una serie de directorios sobre lo que mapea una impresora:



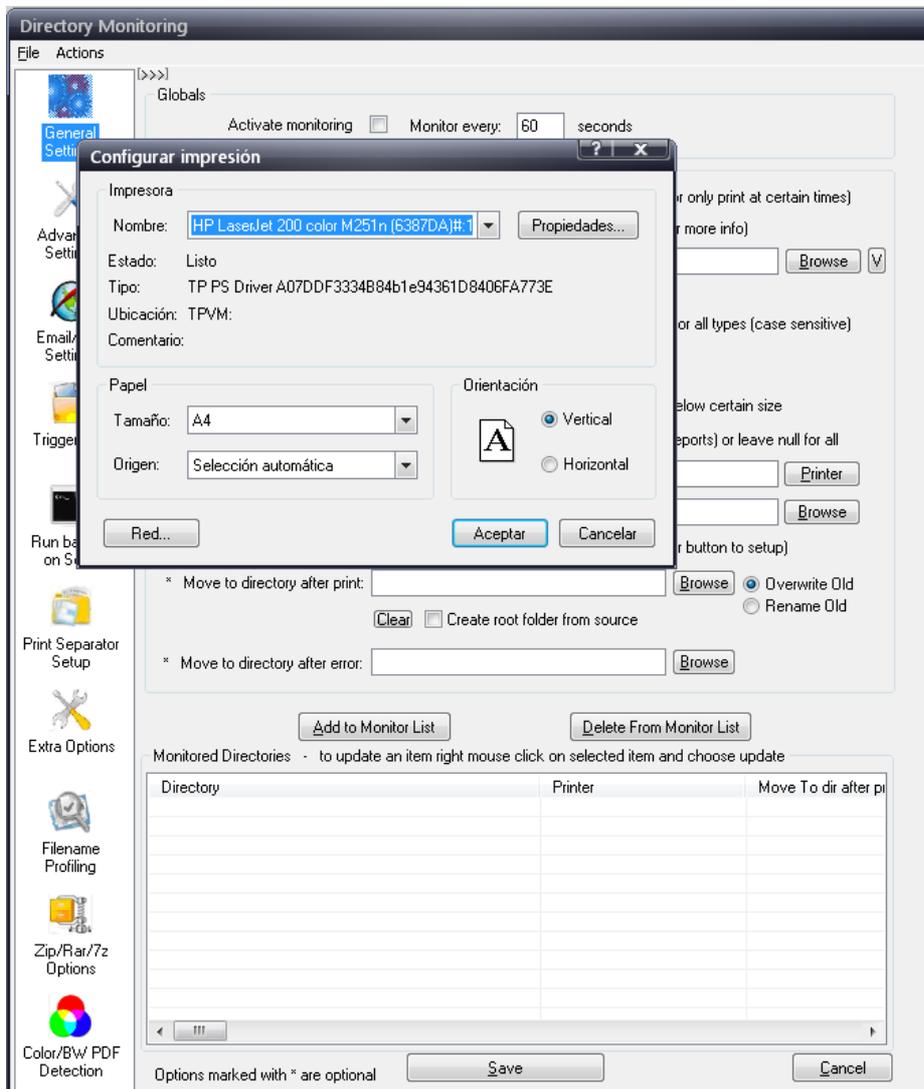
5. Para configurar los directorios y las impresoras en las que se van a imprimir los ficheros que contienen estos directorios es muy sencillo: se accede al menú *Options -> Directory Monitor Setup*.



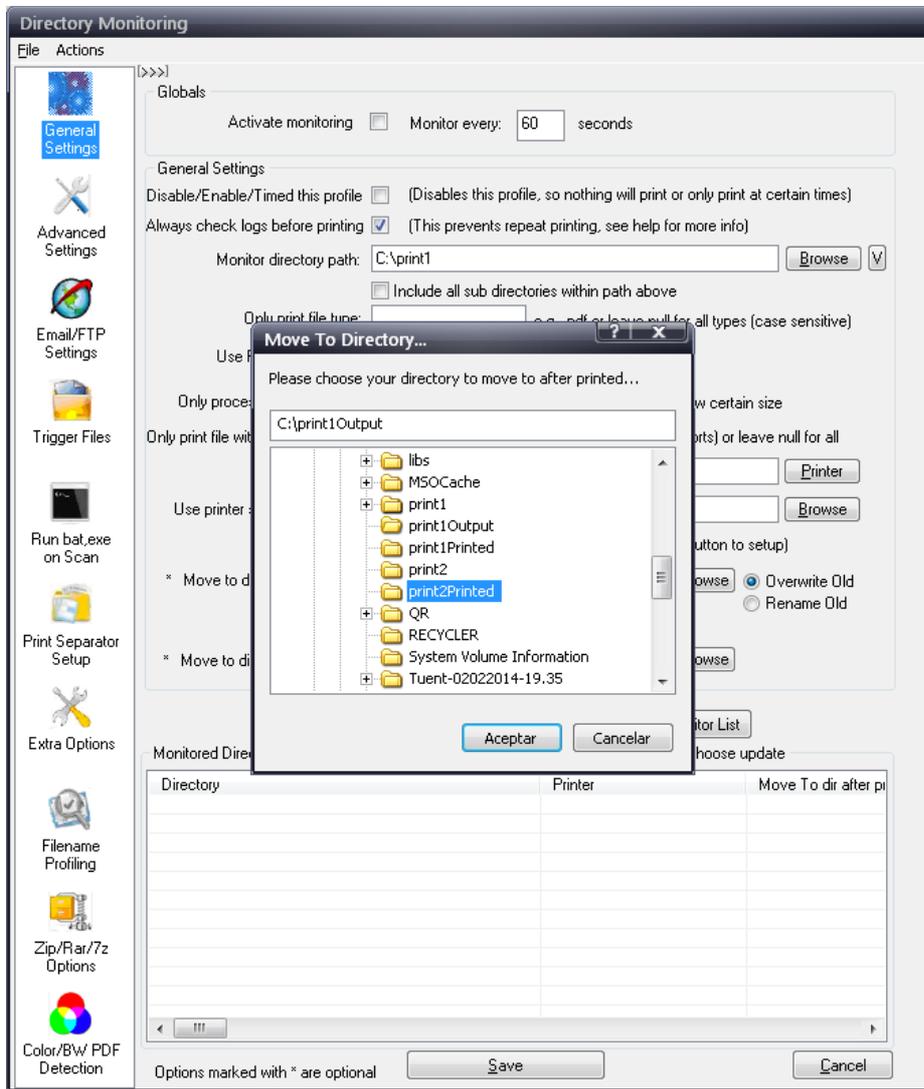
6. En la siguiente pantalla podemos ir configurando el primer directorio. Para ello pulsamos en la opción *Monitor directory path* sobre el botón *browse*, o introducimos el directorio escribiéndolo (o copiando la ruta). En nuestro caso será *c:\print1*. Este directorio será monitorizado cada 60 segundos, tal y como indica en la primera opción de la pantalla. Además activaremos la casilla *Activate monitoring*. Podemos usar un valor menor que 60.



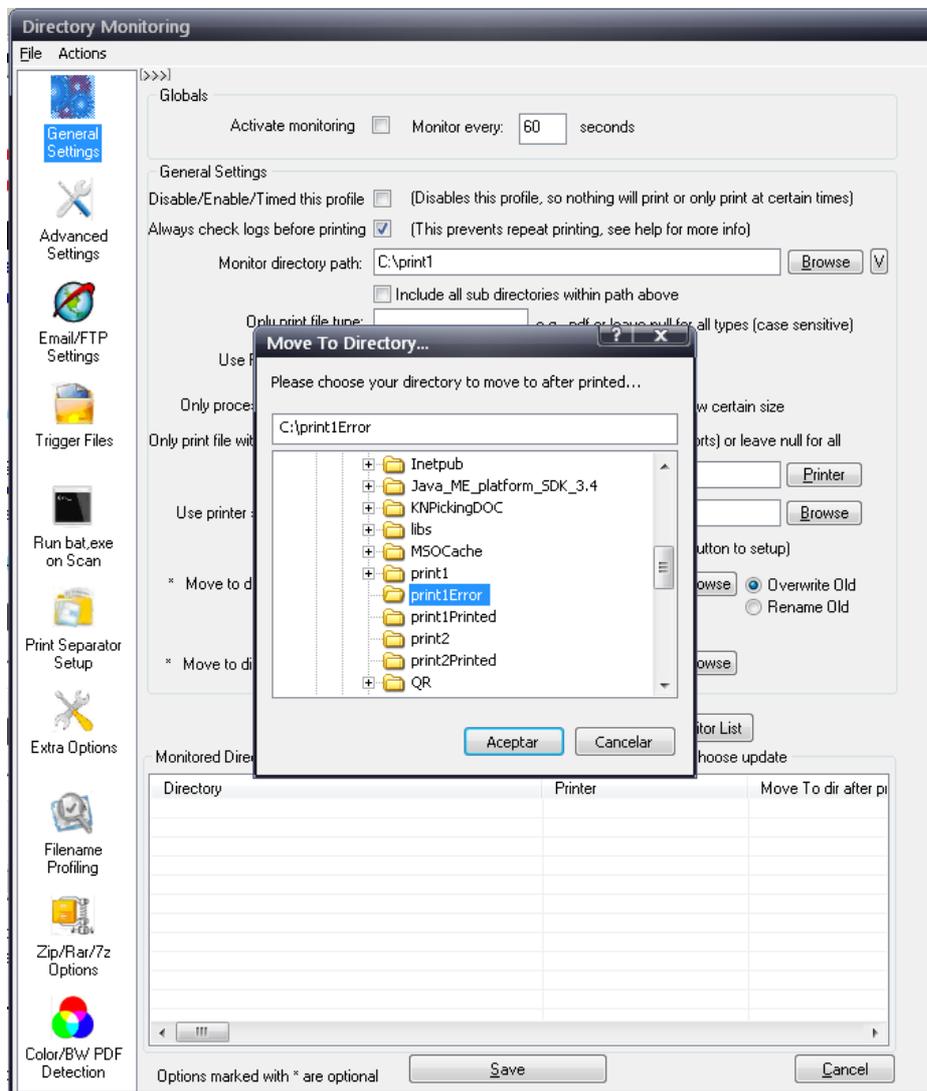
7. El siguiente paso consiste en la selección de la impresora sobre la que se va a imprimir los documentos. Se pulsa sobre el botón *Printer* de la opción *Use printer*. Configuraremos la impresora destino con los parámetros necesarios. Incluso podemos incluir un fichero .prs de configuración en el apartado *Use printer settings file (.prs)* para facilitar esta labor.



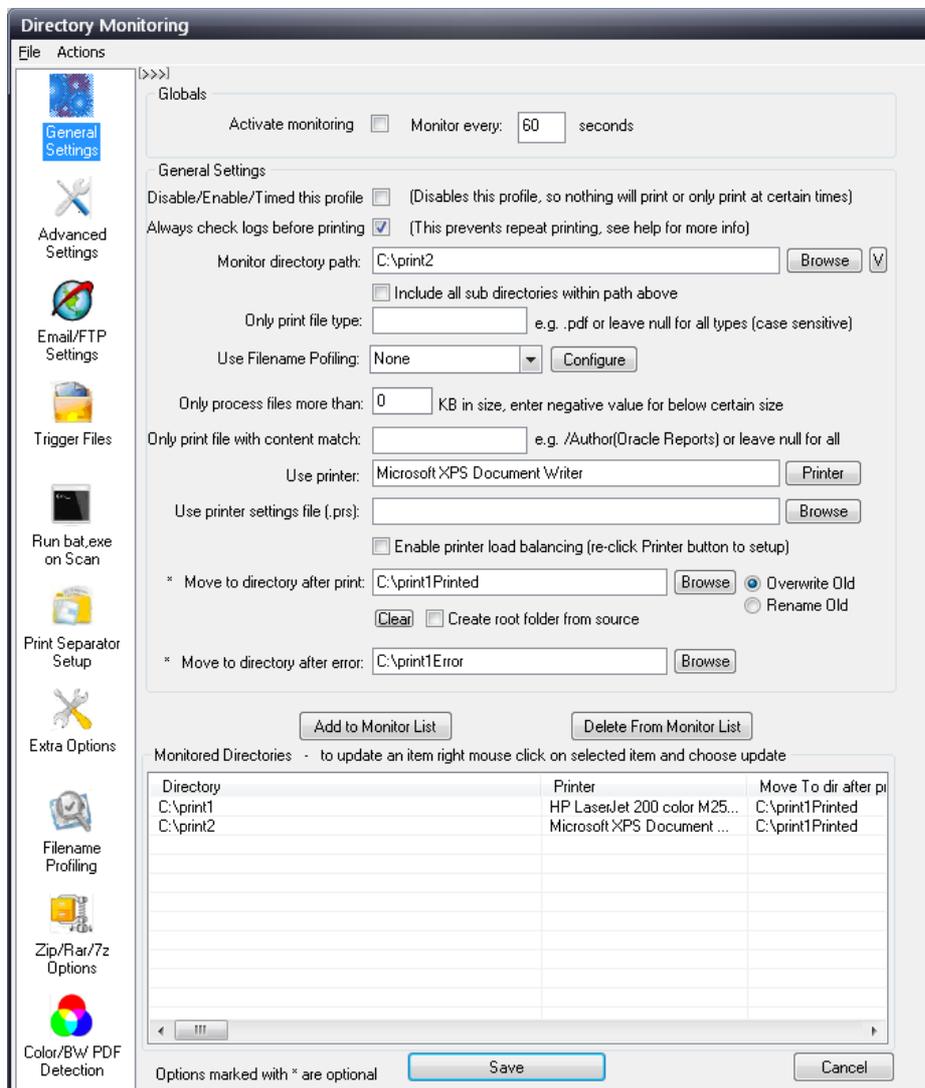
8. El siguiente paso básico (pero opcional) puede ser el directorio al que se mueven los ficheros una vez impreso o cuando no se han podido imprimir por que presentan un error. Se hace indicando el directorio en sendas opciones *Move to directory after print* y *Move to directory after error*. Como antes, se pulsa sobre el botón *Browse* o se escribe o pega la ruta del directorii, en nuestro caso será *c:\print1\Printed* y *c:\print1>Error*.



Para el directorio de error:



9. Finalmente añadimos esta configuración al conjunto de directorios monitorizados pulsando sobre el botón *Add to Monitor List*. La nueva configuración queda añadida a la lista, y ya podemos activarlo para que realice su trabajo. **IMPORTANTE: debe estar activada la opción *Activate monitoring* para que funcione correctamente.**

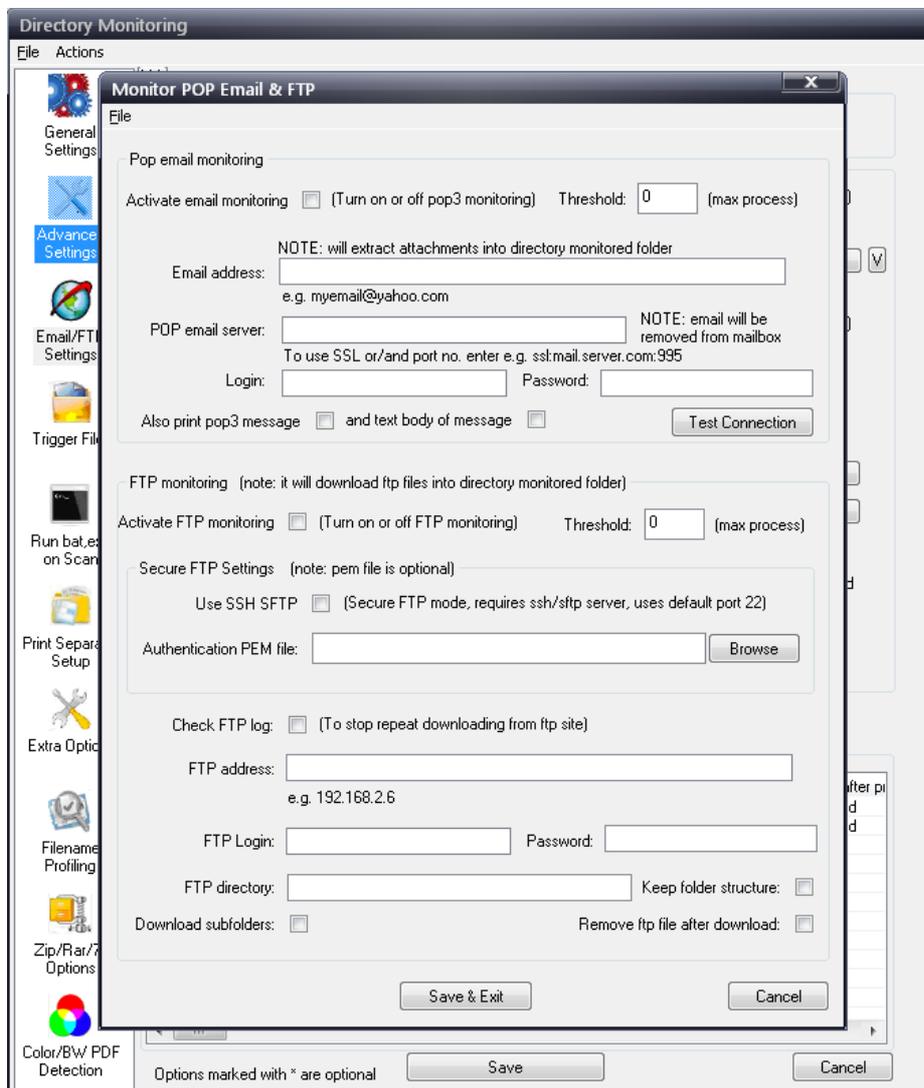


Con esta mínima configuración ya tenemos a nuestro programa Batch & Print monitorizando los directorios que hemos configurado, que en nuestro caso son *c:\print1* y *c:\print2*. Solamente queda probarlo copiando algún fichero con formatos de Office, PDF, txt o cualquiera que soporte nuestro ordenador (gracias al software - Office, Acrobat... - que tiene instalado), al directorio que hayamos mapeado a la impresora seleccionada y esperar a que salga por la impresora.

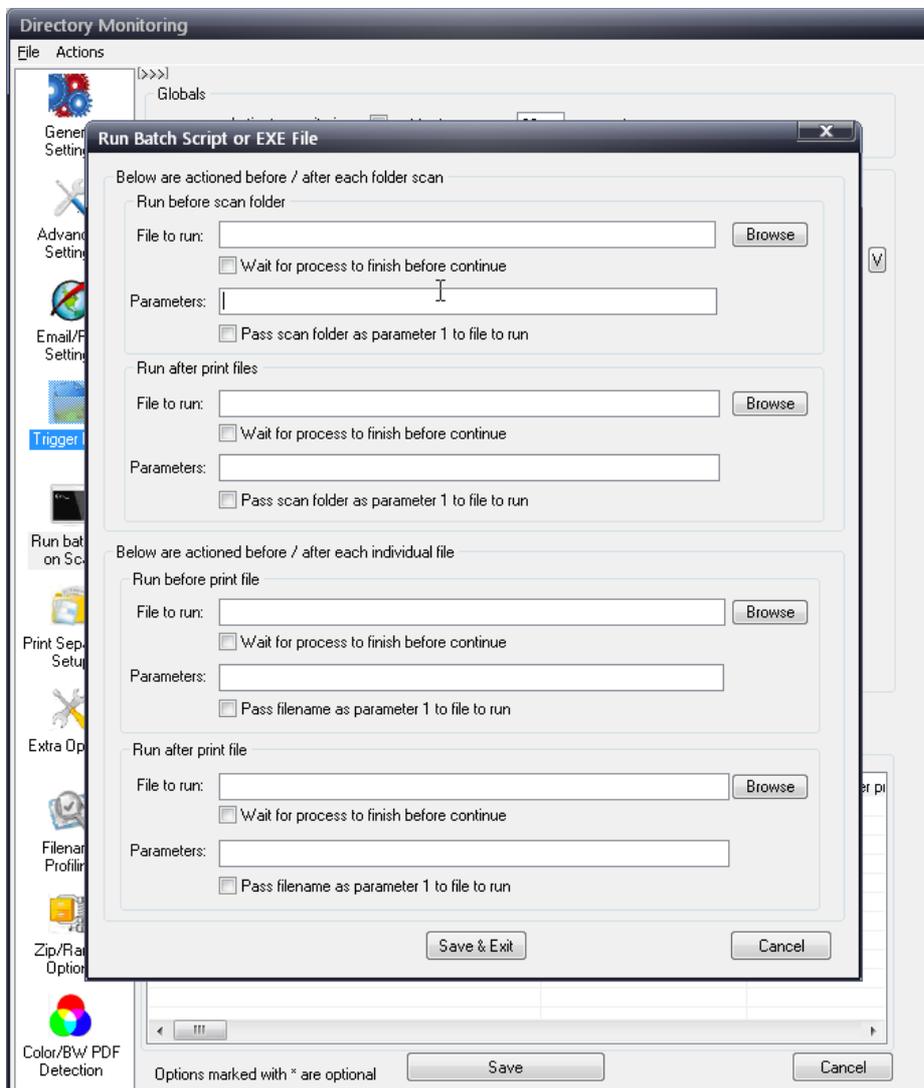
Configuración extra

Además de la labor básica de impresión de documentos en un directorio mapeado a una impresora, podemos ampliar la configuración y utilizar más capacidades de la aplicación (consultables en este [enlace](#)). Por citar un par de ellas que pueden ser de interés:

1. Por ejemplo podemos hacer que monitorice cuentas de correo o FTP, de modo que podríamos enviar ficheros por correo electrónico o dejarlos en un FTP y el programa sería capaz de imprimirlos. Es interesante para aquellas ocasiones en las que el programa no tiene acceso a la red local: simplemente configurando una cuenta de correo de la organización y un ordenador que tenga instalado Batch & Print con acceso a esa cuenta, podemos imprimir los documentos. Igual para el FTP



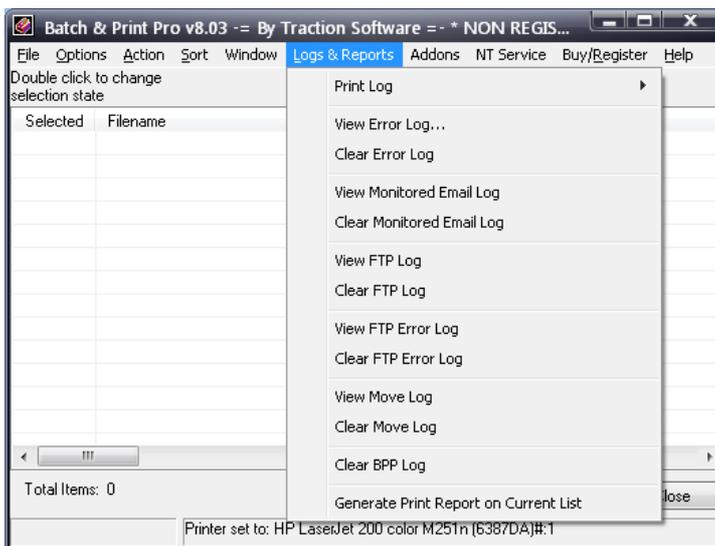
2. También se puede asociar al evento de impresión la ejecución de un fichero .exe o .bat que realiza otras operaciones antes o después de la impresión. Se puede alimentar un log, enviar un correo, etcétera



Más configuraciones de interés

Log de operaciones

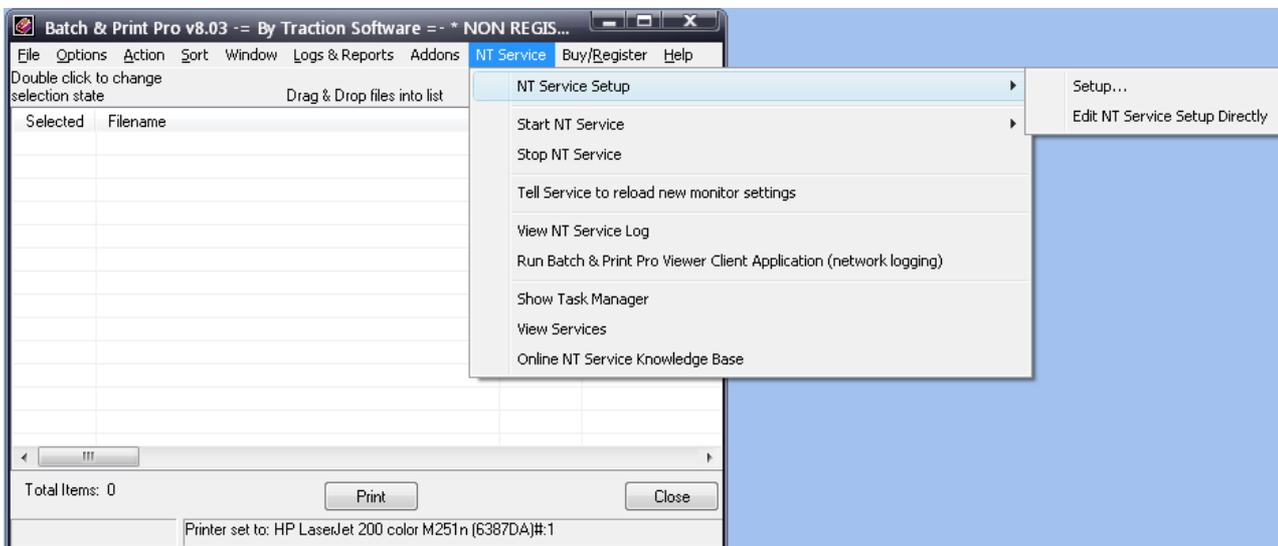
A través del menú *Logs & Reports* podremos tener acceso al log de operaciones realizadas por la aplicación, tanto de impresión realizada como de errores en la monitorización, ficheros arrastrados o acceso a la cuenta de correo o de FTP. Es sumamente interesante a la hora de obtener estadísticas o detectar errores.



Configuración como Servicio de NT

Como se ha podido ver en este tutorial, Batch & Print es un programa que está claramente destinado a residir como servicio dentro del servidor o de un ordenador que tiene acceso a los mismos directorios compartidos que el servidor (donde se va a dejar los ficheros en los directorios mapeados) y acceso a las impresoras.

Para configurarlos disponemos un menú completo en *NT Service*. Desde ahí podemos configurar diferentes parámetros.



Como por ejemplo el setup del servicio en Windows, indicando el usuario y password que corren el servicio o la capacidad para arrancarlo o pararlo.



6. Conclusiones

En este tutorial hemos visto las diferentes alternativas que pueden existir a la hora de realizar una tarea tan sencilla pero a la vez compleja de realizar en Java como es la impresión de documentos Office y PDF en una red Windows. Java es un excelente lenguaje de programación para la realización de las más exigentes aplicaciones de gestión, pero no tiene las ventajas de otros lenguajes como .NET cuando se trata de desenvolverse dentro de un entorno cerrado como es Windows.

Se han tratado como alternativas las funcionalidades de impresión básica desde Java, y se ha visto que no eran suficientes para satisfacer las necesidades planteadas, ni en forma ni en tiempo de desarrollo. Ante esta situación límite de uso de una tecnología hemos visto cómo tenemos que hacer uso de una estrategia de "pensamiento lateral", intentando abstraernos de nuestro papel de especialistas y planteándonos si realmente pueden existir otras alternativas fuera del entorno que instintivamente habíamos descrito para el problema.

Gracias a plantearnos una solución fuera de Java, hemos podido ver otras tres alternativas factibles de verdad para hacer la impresión desde Java, delegando la responsabilidad de la impresión en otro tipo de desarrollo, y reduciendo la dificultad de la parte a desarrollar en Java.

Dentro de estas soluciones hemos analizado la solución comercial Batch & Print que nos permite mapear directorios accesibles a disco con impresoras, de modo que a través de un monitoreo constante, permite detectar los nuevos documentos existentes en un directorio e imprimirlos automáticamente en la impresora indicada, permitiendo además un gran número de configuraciones y funcionalidades.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)

Por favor, vota +1 o compártelo si te pareció interesante

Share |



Anímate y coméntanos lo que pienses sobre este **TUTORIAL**:

» **Regístrate** y accede a esta y otras ventajas «



Esta obra está licenciada bajo licencia [Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

PUSH THIS

Page Pushers

Community

Help?

no clicks

0 people brought clicks to this page



powered by [karmacracy](#)

Copyright 2003-2015 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

