Autobahn DX Reference Guide

Version 1.10

July 2006

http://www.aquaforest.com/

CONTENTS

1 PRODUCT OVERVIEW AND CONCEPTS

1.1 Overview

Autobahn DX is a Document Processing product designed to fit into an organization's document workflow. It is able to process and convert a variety of different types of document including TIFF images, Microsoft Office documents and HTML pages, convert those documents to searchable PDFs and perform a variety of other customizable activities.

Autobahn includes PDF Junction (Professional Edition) and TIFF Junction (either Standard or Professional if the OCR Option of Autobahn DX is chosen)

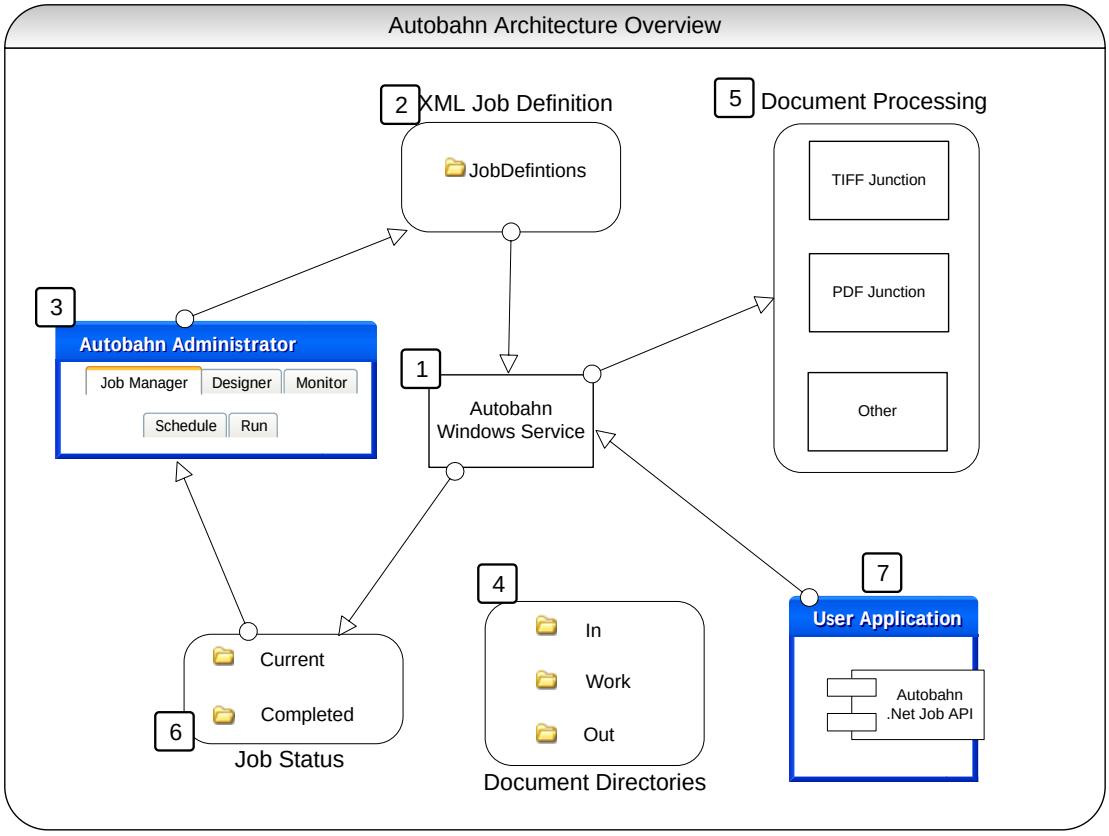
For reference, this matrix highlights the functionality of Aquaforest's document conversion and processing products :

Function	TIFF Junction Standard	TIFF Junction Professional	PDF Junction Standard	PDF Junction Professional	Autobahn DX (Standard)	Autobahn DX (OCR)
Convert TIFF to PDF	X				X	X
Split TIFFs	X				X	X
Merge TIFFs	X				X	X
OCR & Convert TIFF to Searchable PDF		X				X
OCR & Convert Image-Only PDF to Searchable PDF		X				X
Split PDFs			X	X	X	X
Merge PDFs			X	X	X	X
Set Security on PDFs			X	X	X	X
Convert Microsoft Office Documents to PDF*				X	X	X
Convert HTML, Text and Other Documents to PDF*				X	X	X
Process Files, Folders or Entire Trees	X	X	X	X	X	X
Command Line and GUI Interface	X	X	X	X	X	X
Support for Windows 2000, 2003 & XP Professional	X	X	X	X	X	X
Graphically Define Document Processing					X	X
Windows Service					X	X
Scheduled & Ad-hoc Job Support via XML Job Tickets					X	X
Watched Folders					X	X
.Net API Interface					X	X
Customizable Script Support					X	X
Email Alerts					X	X

*Requires native application present on server.

2 AUTOBAHN DX ARCHITECTURE AND CONCEPTS

2.1 Autobahn DX Product Architecture



2.2 Autobahn Windows Service

This is the heart of the product and controls the execution of both scheduled jobs and ad-hoc jobs whether submitted via the Autobahn Manager or via the Autobahn Job API. The service analyses the XML Job Definition files on start-up and when new files are created in the Job Definition directory by the Autobahn Manager, or via the Job API. The XML Job Definition Files describe the steps to be carried out to complete the job and the Autobahn Service will spawn sub-jobs (such as TIFF Junction or PDF Junction) where required. Job Status records and logs are maintained and can be reviewed in the Job Monitor and Job Manager.

2.3 XML Job Definition Files

Each Job has one or more XML Job Ticket files that define a number of aspects of the job including :

- Job ID
- Job Name
- Scheduling Information
- Input Directory
- Work Directory
- Output Directory
- etc

Section 7 of this document describes the XML Job Definition Files in full detail.

2.4 Autobahn Administration

This is a windows application that provides the primary administration interface and comprises the following main sections :

2.4.1 Job Manager

This provides a method of managing the scheduled jobs previously defined in the Process Designer they can be selected and then deleted, edited (in the process designer) or scheduled.

2.4.2 Designer

This provides a graphical interface to allow creation of a series of steps that make up a job. The process specification can then be saved (as an XML Job File), and run or scheduled via the Job Manager section of the application.

2.4.3 Monitor

This shows the current service status and provides a monitor interface to currently executing jobs. It also allows the service to be stopped or started.

2.5 Document Directories

Each job will have a set of directories containing the source documents, and directories for work, errors and output.

2.6 Document Processing

Each step in a Job will involve a separate Job Element process being spawned by the Autobahn Service. This may involve its own Sub-Job XML File and always will in the case where the step will execute TIFF Junction or PDF Junction.

2.7 Job Status

To support the Job Monitor, a mechanism is required to indicate the progress of current jobs, in particular start time and current log file output.

2.8 Junction Server .Net Job API

A .Net API is provided to allow user application to create and execute ad-hoc jobs. See section 9 for further details.

3 INSTALLATION AND LICENSING

3.1 System Requirements

Supported Operating Systems	Windows 2003, Windows XP
Disk Space	120Mb
.Net Framework	1.1 or 2.0

3.2 License Key

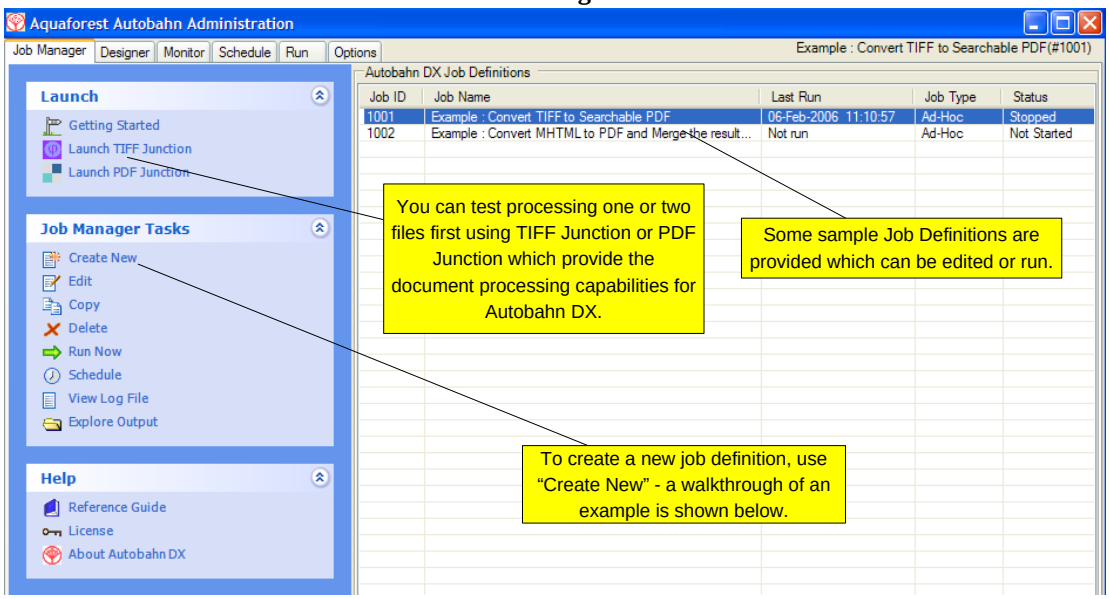
Autobahn DX may be licensed as a “Standard Edition” or an “OCR Edition” which allows creation of searchable PDF files from TIFFs or Image-Only PDFs. By default the trial edition is configured as the “OCR Edition” with a license key of TRIALOCR. Standard-Edition only can be tested by using a license key of TRIAL.

License keys may be entered by using the “License Menu” item in the Help menu of the Autobahn DX Administration Tool.

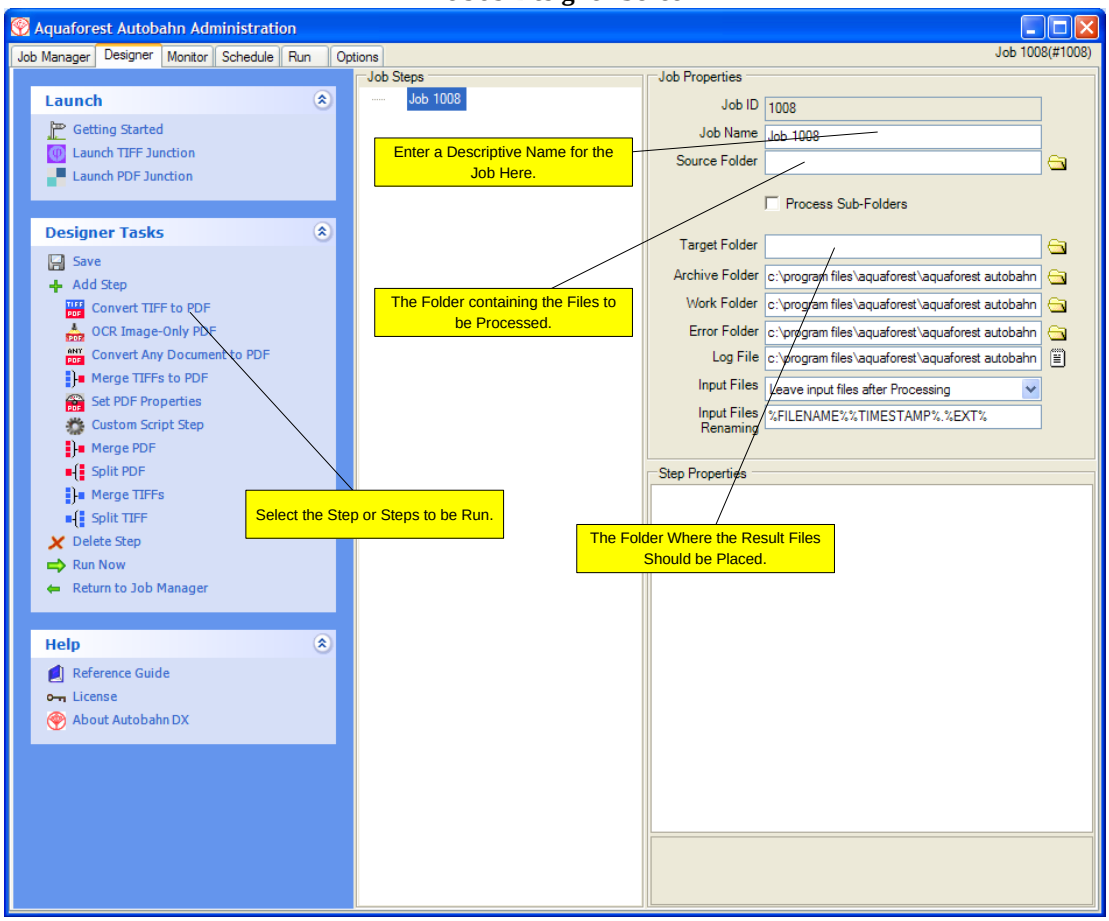
4 AUTOBAHN DX QUICK START

Autobahn DX is designed to allow definition of document processing jobs ("Job Definitions") and then scheduling them to run continuously on watched folders or on a daily scheduled basis. The annotated screen shots in this section are designed to enable rapid familiarization with the product.

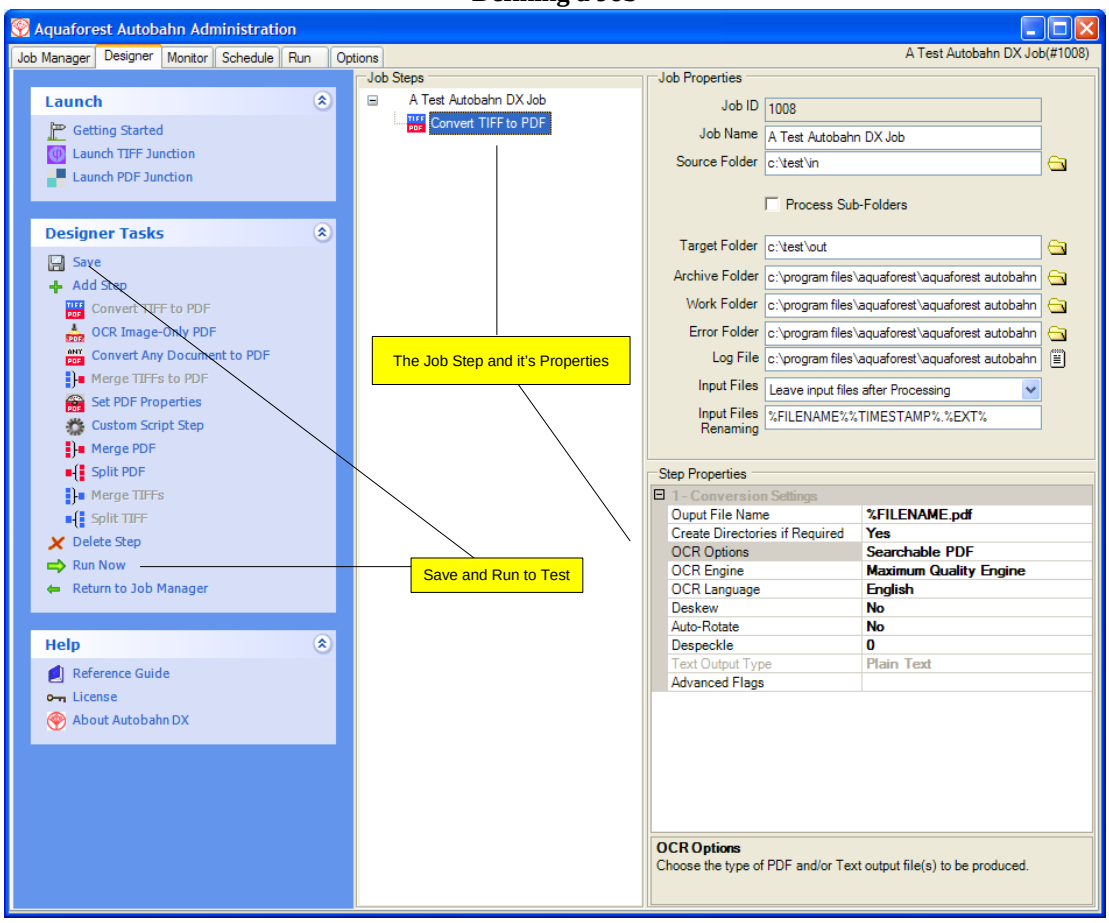
The Job Manager Screen



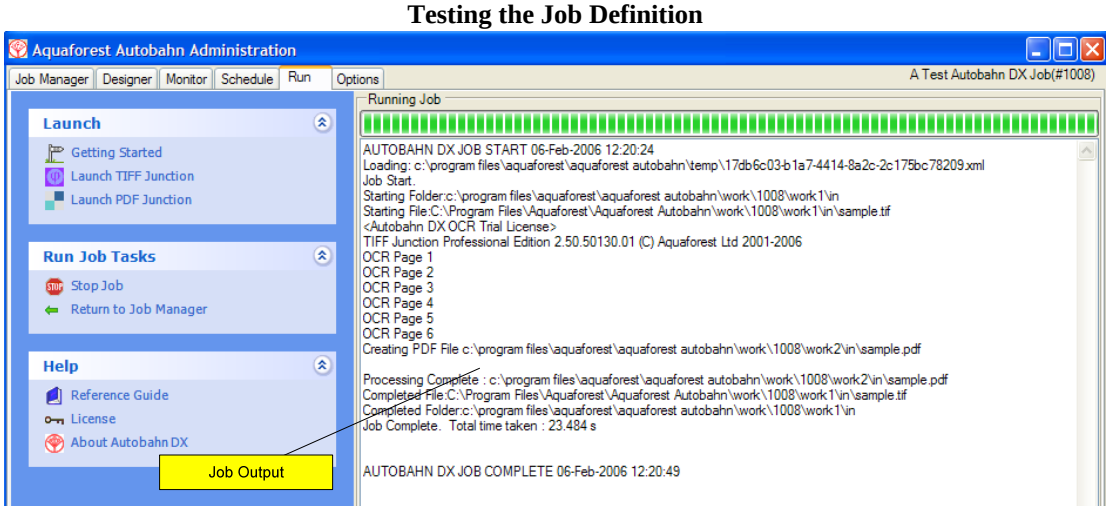
The Job Designer Screen



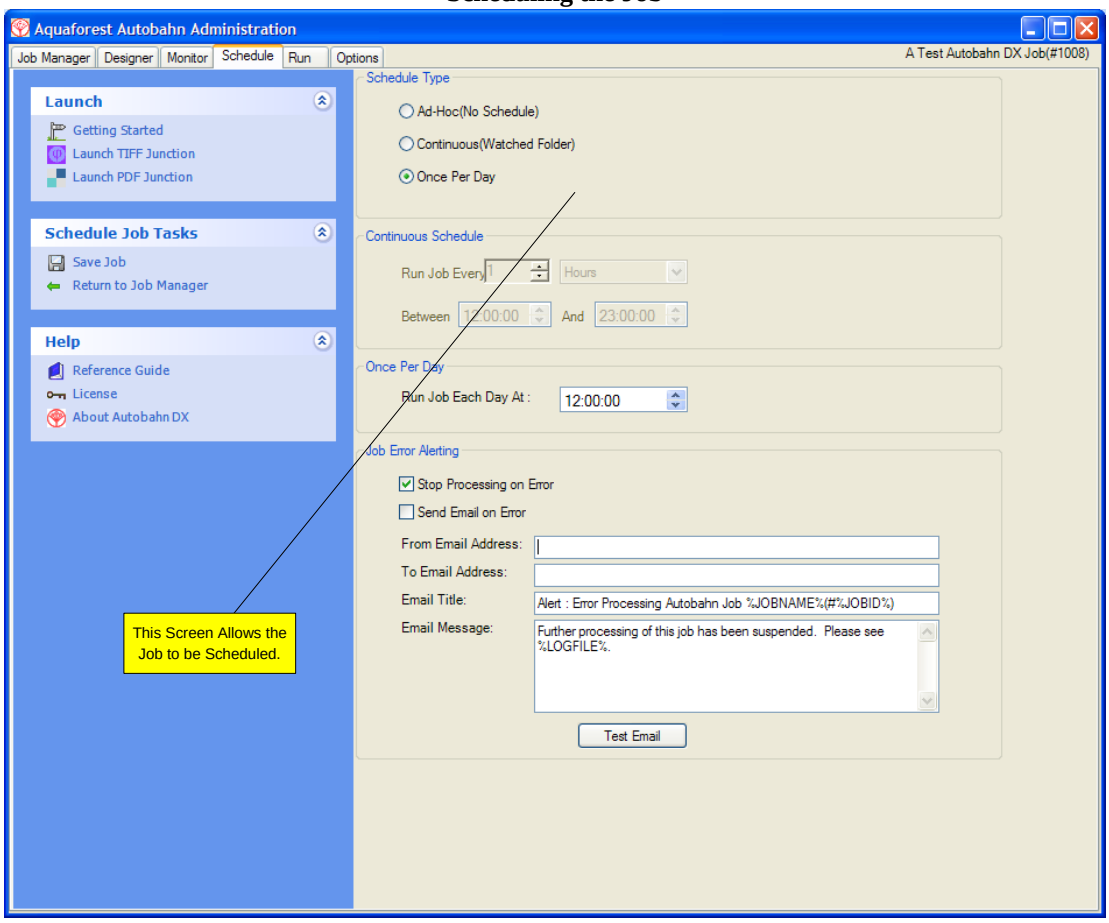
Defining a Job



Testing the Job Definition



Scheduling the Job

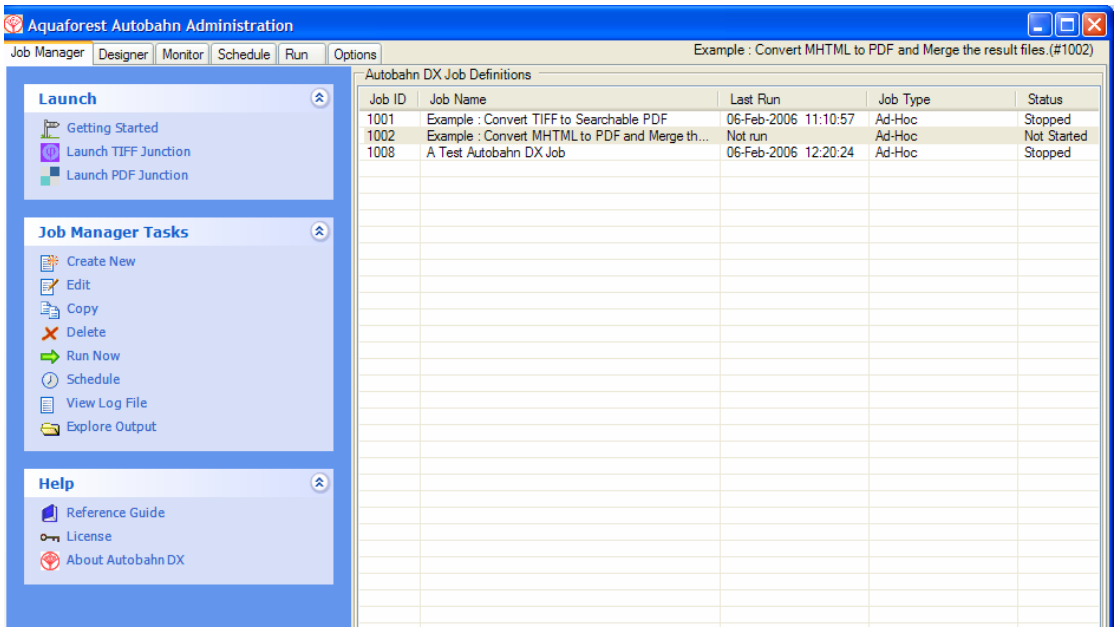


5 THE AUTOBAHN DX ADMINISTRATION TOOL

The Autobahn administration tool is a Windows Forms application designed to enable system administrators to be able to create and configure Job Definitions that control document processing. A number of example Job Definitions are included with the product which can be tailored to meet a user's precise requirements.

5.1 Job Manager

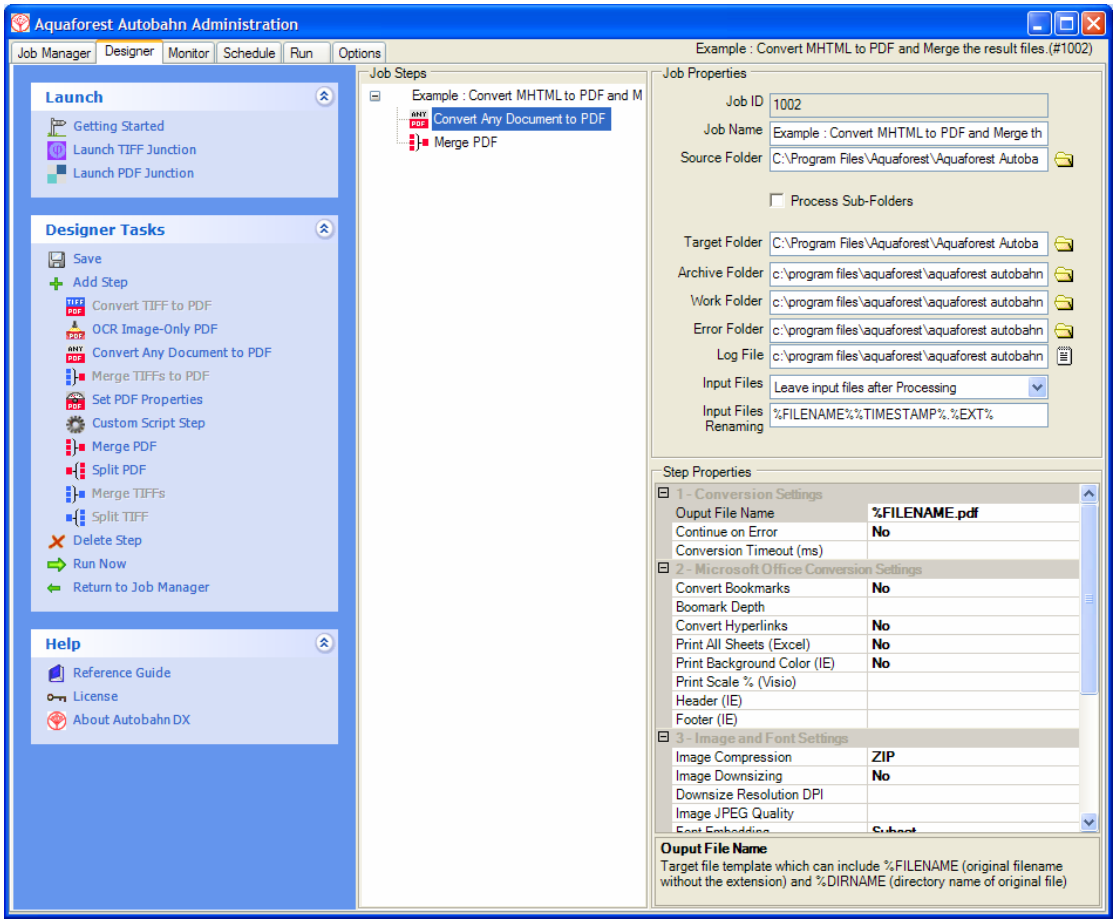
This section of the application displays a list of all current Job Definitions and the set of operations to be performed shown in the menu table below.



Menu Item	Action
Create New	Creates a new, skeleton job definition from jobtmplt.xml and then switches display to the Job Designer.
Edit	Switches the view to Edit on the selected Job.
Copy	Creates a copy of the selected job. The job name will be Job %JOBID% (Copy of %JOBNAME%)
Delete	Deletes the selected job definition after confirmation.
Run Now	Executes the selected job via the interface in the Run tab window.
Schedule	Switches the view to the schedule window for the selected job.
View Log File	Displays the most recent log file for the job.
Explore Output	Launches windows explorer on the job output folder.

5.2 Job Designer

This allows definition and editing of a job definition, using a tree-list type model coupled with a Visual Studio – style property list. Each step type will have its own icon. Drag and drop can be used to allow reordering of steps.



5.2.1 Menu Items

Menu Item	Action
Save	Validates the current job and if valid, saves the current job definition to %JOBID%.xml in the %JOBDEFDIR% directory.
Add Step (Expands)	Expands to show a list of steps that are valid to be placed after the currently selected tree node (or as the first node if none is currently highlighted). Which steps are valid is determined by matching the outputtype of the previous step with the validinputtype of the possible steps.
Delete Step	Deletes the currently selected step node.
Run Now	Performs a save (as above) and then executes the job. The output is displayed in the Run tab screen.
Return to Job Manager	Returns to the job manager with the current job selected.

5.2.2 Fields

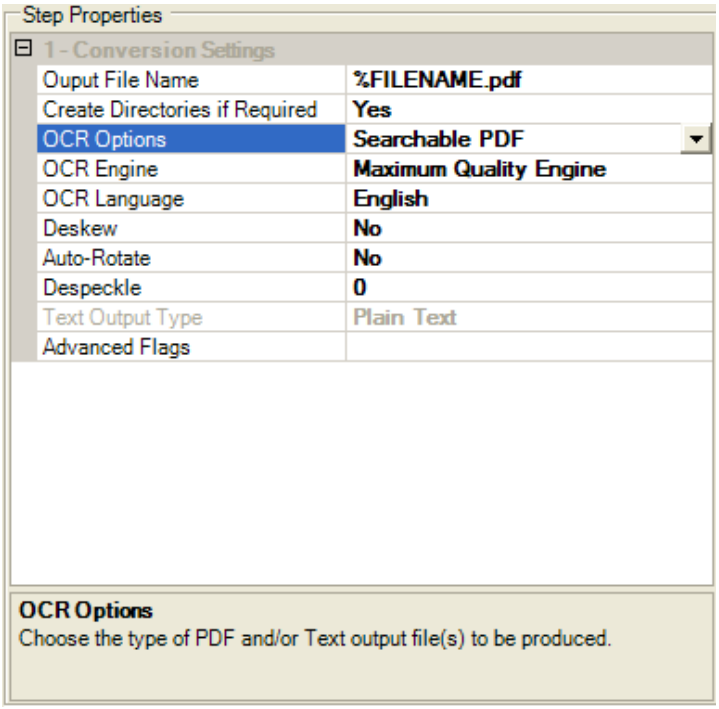
Menu Item	Description
Job ID	A sequential Job ID is allocated for the Job by Autobahn DX. This cannot be changed.
Job Name	A descriptive title for the job.
Source Folder	The folder containing the documents to be processed.
Process Sub-Folders	If checked, all sub-folders will be recursively processed.
Target Folder	The folder where the processed files will be placed if "Move input files to target folder after processing" is chosen.
Archive Folder	The folder where the processed files will be placed if "Move to archive after processing." is chosen.
Work Folder	The folder where files will be temporarily stored during conversion and processing.
Error Folder	May be used by custom job steps to place documents in error.
Log File	Path of the job log file. This can include %TIMESTAMP%, which is the timestamp of the job start time, and will therefore give a new log file for each run.
Input files	This option determines what happens to the input files once processing has been completed. The options are : *Leave input files after processing*: Files are left in the Source Folder. *Move to archive after processing*: Files are moved to the Archive Folder. *Move input files to target folder after processing:* Input files are placed in the same folder as the output files. *Delete input files after successful processing*: Input files are deleted.
Input Files renaming	This determines how input files will be renamed when moved to the Target or Archive folder. The default is : %FILENAME%%TIMESTAMP%.%EXT%

5.2.3 Step Types

Step Type	Description	Underlying Functionality Provided by
Convert TIFF to PDF	Converts a TIFF file to PDF. The TIFF compression must be one of the support formats : Group 4, Group 3, LZW, JPEG (type 6), RLE, Uncompressed (bitonal). Depending upon the Step Type Properties chosen, the PDF file may be searchable or image-only, and a separate text or HTML file may be produced from the OCR process.	TIFF Junction
OCR Image-Only PDF	Creates a searchable PDF file from the set of images from an image-only PDF file. Depending upon the Step Type Properties chosen, a separate text or HTML file may be produced from the OCR process.	TIFF Junction
Convert any document to PDF	This converts any printable document to PDF, such as Microsoft Word, Excel, PowerPoint, HTML etc subject to the native application being available on the server.	PDF Junction
Merge TIFFs to PDF	This merges a folder of TIFF files together to produce a PDF file.	TIFF Junction
Set PDF Properties	This is used to set PDF Metadata properties (such as Author, Title etc), Security settings and Document Display properties.	PDF Junction
Custom Step	This can be used to support a custom scripted step in the process. See section 6 for more details.	-
Merge PDF	Merges a folder of PDF files into a single file.	PDF Junction
Split PDF	Splits each input PDF file into a set of files, either a single page per file or by page ranges.	PDF Junction
Merge TIFF	Merges a folder of TIFF files into a single file.	TIFF Junction
Split TIFF	Splits each input TIFF file into a set of files, either a single page per file or by page ranges.	TIFF Junction

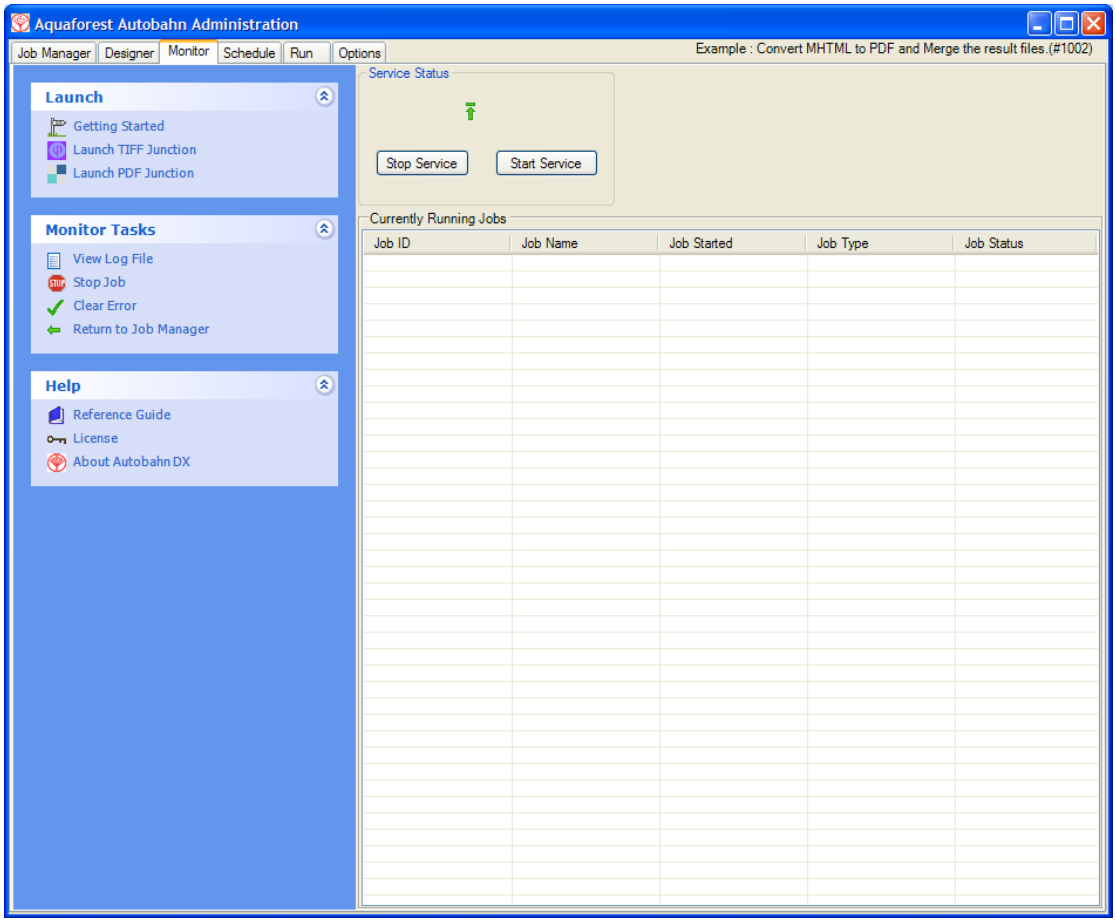
5.2.4 Step Type Properties

Each of the Step Types referred to in the previous section will have a set of properties such as that shown below for “TIFF to PDF”. Each property has a description associated with it which is displayed when the property is highlighted. Further information can be obtained by reviewing the reference guide of the underlying product that provides the functionality. For example, the TIFF Junction guide can be viewed by using “Launch TIFF Junction” and then Help | Reference Guide.



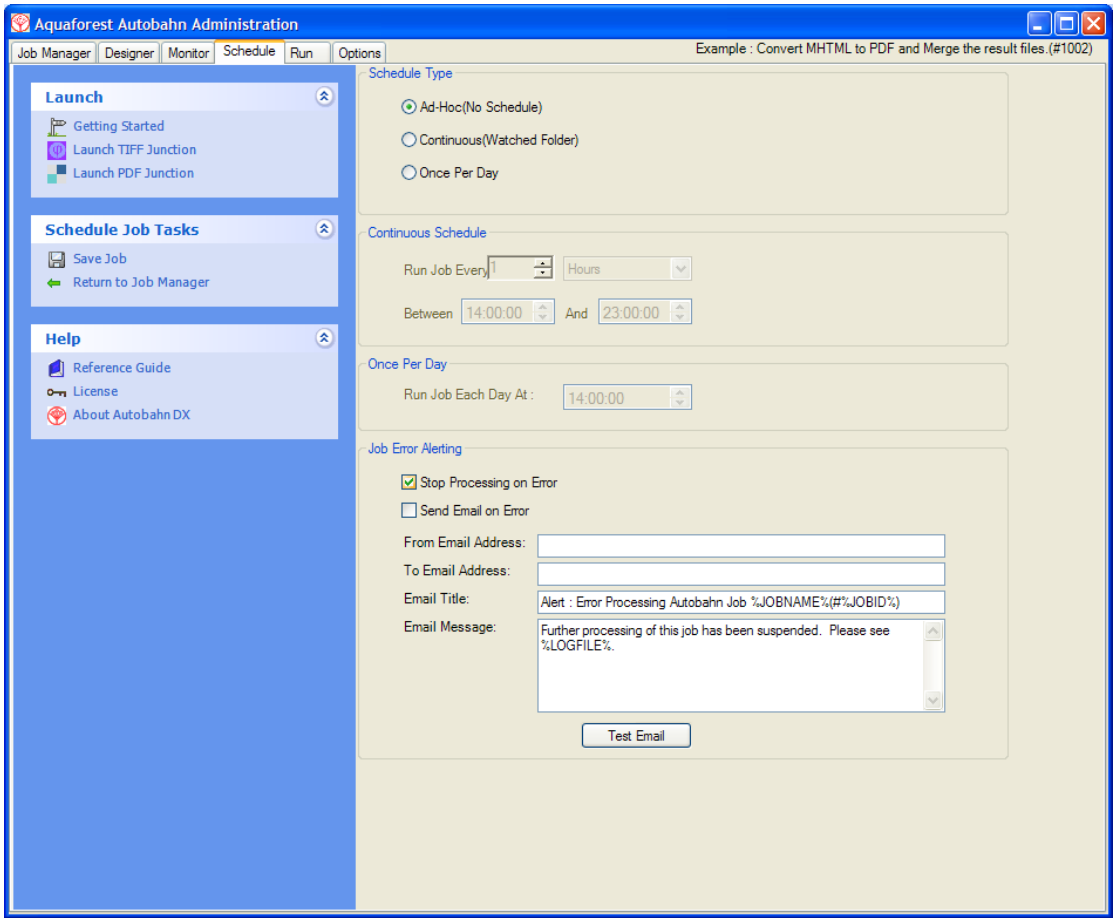
5.3 Monitor

This provides an interface to stop and start the Autobahn service, and to see which Jobs are currently running.



Menu Item	Action
View Log File	View the log file of the currently selected job.
Stop Job	Halt execution of the selected job.
Clear Error	Clears an error flag associated with a job, allowing it to be rerun.
Stop Service	Stop the Autobahn service.
Start Service	Start the Autobahn service.

5.4 Job Scheduling



Menu Item	Action
Save	Saves the schedule for the current job
Stop Processing on Error	If checked, the job will stop if it returns an error, and will not run again until the error is cleared from the Monitor screen.
Send Email on Error	If checked, Autobahn DX will send an email if an error occurs with a job. The email server configuration should be setup on the “options” screen – see section 5.5 for more information.
From Email Address	The “from” email address that will be used for the message.
To Email Address	The email address that the message will be sent to.
Email Title	The title of the email.
Email Message	The body of the email.

The product supports three types of scheduling which are implemented via the Autobahn DX service :

5.4.1 Ad-Hoc

This means that the job does not have any fixed schedule, but may be explicitly run via the management GUI or via one of the API methods.

5.4.2 Watched Folder / Continuous Scheduling

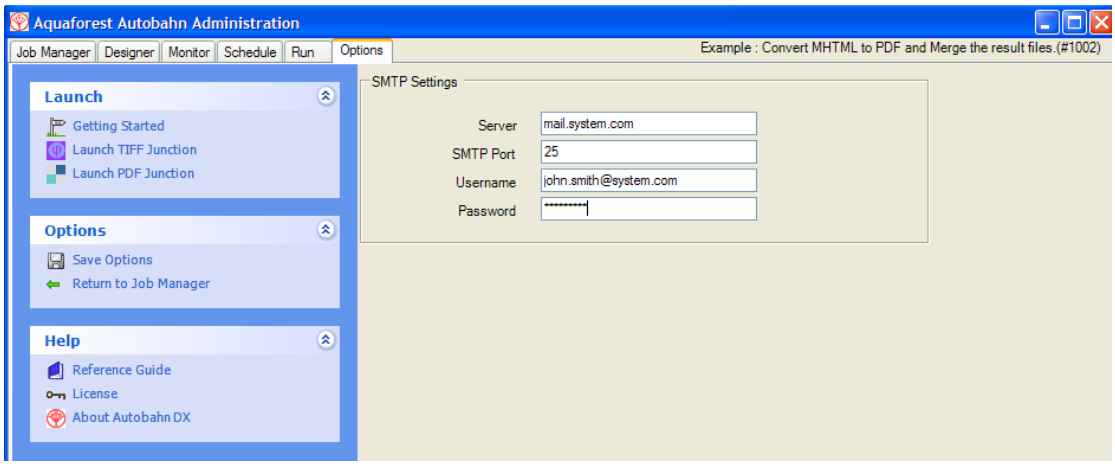
This allows the job to be scheduled to run periodically between a start time and end time each day. The periods may be seconds, minutes or hours. For example, a job may be specified to run every 30 seconds between 9:00 and 17:00.

5.4.3 Daily Scheduling

This allows the job to be scheduled to run at a specified time each day.

5.5 Email Options

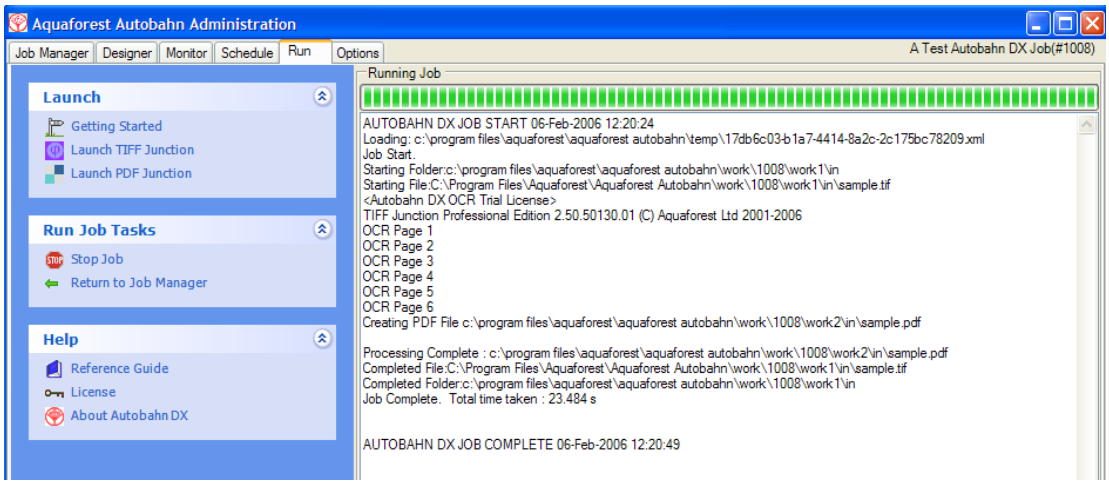
The options screen allows email server information to be defined. This is used to support the "Send Email on Error" functionality defined in the previous section. The information entered is help in Autobahn.config in the config directory. The password is held in an encrypted form.



Field	Description
Server	Address of the Server hosting the SMTP server.
SMTP Port	SMTP Server port. Default is 25.
Username	Username for authentication by the server.
Password	Password for the username.

5.6 Running Jobs from the Admin Tool

This executes the currently selected job definition and all output is placed in the text area window, as well as the Job log file.



Menu Item	Action
Stop	Stops the currently running job.
Return to Job Manager	Returns to the job manager with the current job selected.

5.7 Launching TIFF Junction and PDF Junction

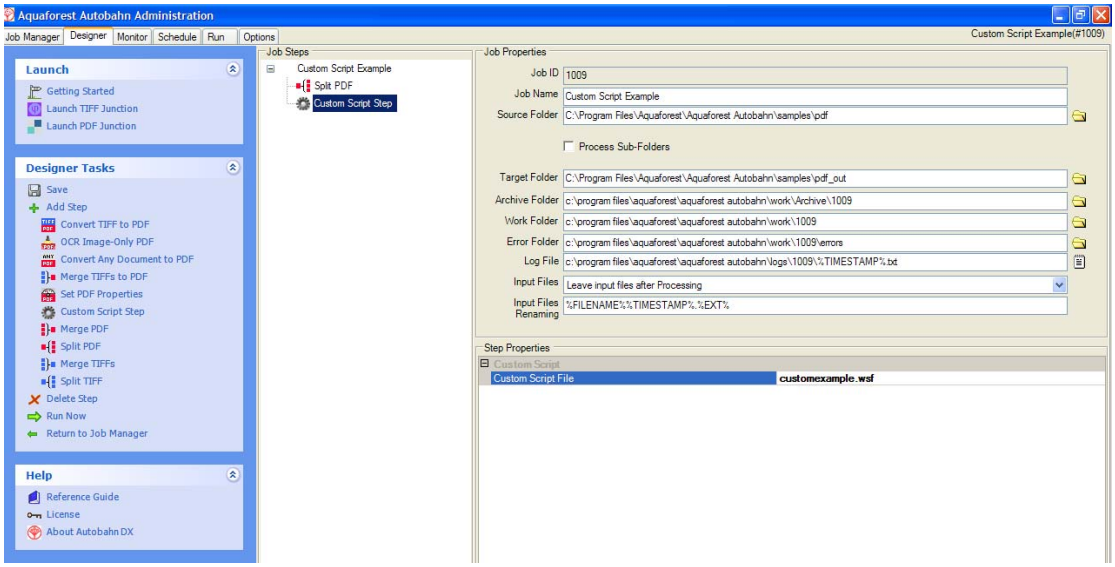
Autobahn includes both the TIFF Junction and PDF Junction products to provide document processing functionality. Both products can be launched directly from the menu. This can be helpful for testing conversion options before creating a Job Definition.

6 SCRIPTING CUSTOM STEPS

Autobahn DX allows custom job steps to be defined that can be included in a Job Definition in the same way as any other step type. The custom step itself can be any form of executable file or script. A template wsh script is included which demonstrates calling a command-line executable from within the script.

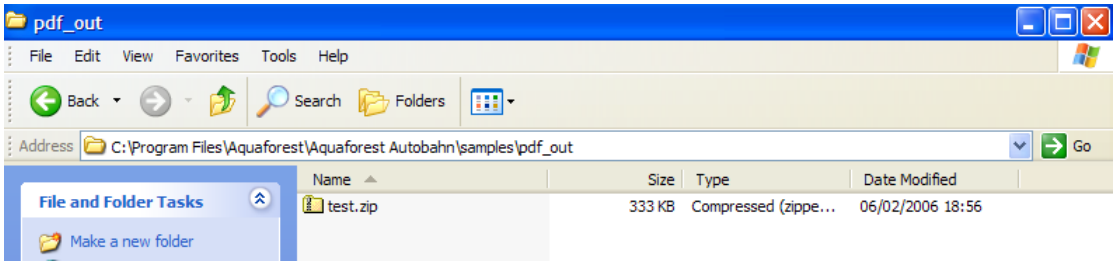
6.1 Custom Script Example

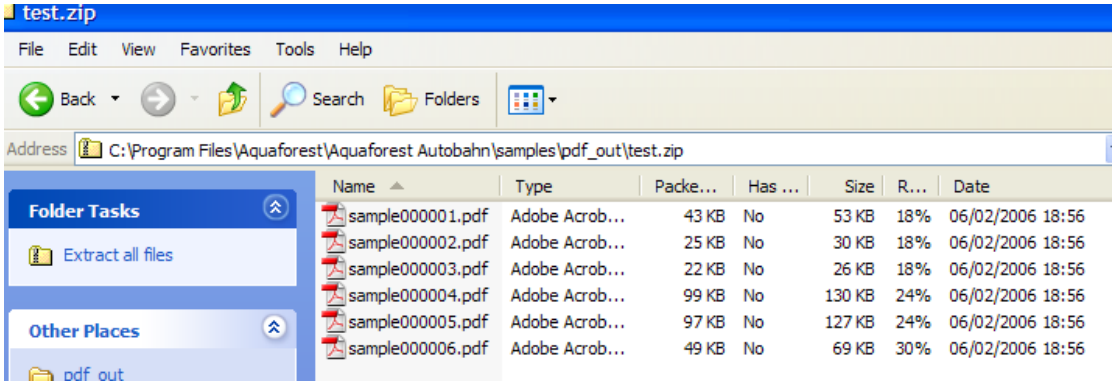
Included with the product is an example Job Definition that will split a PDF file into single pages and then create a single ZIP file that contains the pages. The example makes use of the Freebyte Zip application which you can download from here http://www.freebyte.com/download/fbzpack.exe , run and unpack into C:\Program Files\Aquaforest\Aquaforest Autobahn\custom.



Custom script files must be placed in C:\Program Files\Aquaforest\Aquaforest Autobahn\custom. In this case the custom script file is customexample.wsh.

The results of the Job Definition will be a zipped file as shown :





6.2 Custom Example Script

The contents of Customexample.wsf are shown below. This file can be used as a template for other custom job steps.

```
<job>
<runtime>
</runtime>
<script language="JScript">

/*****************************************************************
**
**      (C) Aquaforest Limited 2006
**
**      FILE
**      customexample.wsf
**
**      DESCRIPTION
**      Example Custom Script for Autobahn DX
**
*****************************************************************/

var WshShell = WScript.CreateObject("WScript.Shell");

/*****************************************************************
       Arguments
*****************************************************************/

var inFolder="";
var outFolder="";

if(WScript.Arguments.Named.Exists("in"))
{
       inFolder=WScript.Arguments.Named("in");

}
if(WScript.Arguments.Named.Exists("out"))
{
       outFolder=WScript.Arguments.Named("out");
}

doOut("In Custom Script Example! /in="+inFolder+" /out="+outFolder);

doCmd('C:\\Program Files\\Aquaforest\\Aquaforest Autobahn\\custom\\fbzip.exe -a
"'+outFolder+'\\customexample.zip" "'+inFolder+'" ');

/* Write Output */

function doOut(s)
{
       WScript.StdOut.WriteLine(s);
       if(loggingOn)
       {
              logOut.WriteLine(s)
       }
}

/* Execute Command */
```

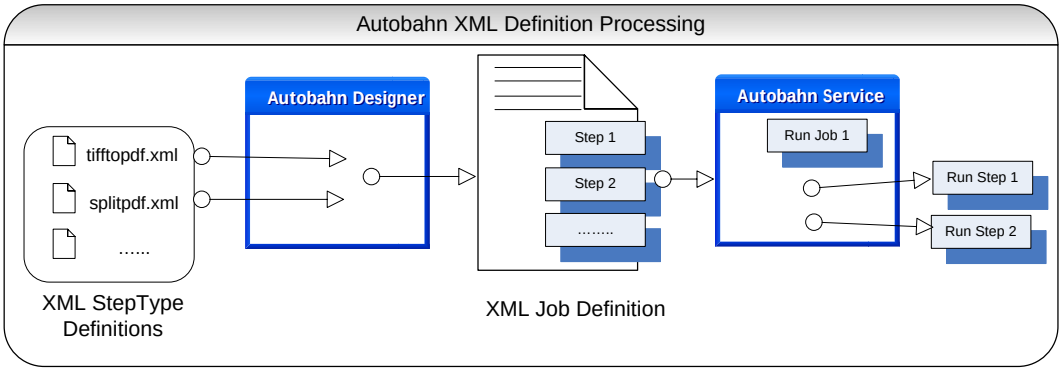
```
function doCmd(cmd)
{
	doOut(cmd);
	var oExec    = WshShell.Exec(cmd);
	while(oExec.Status == 0)
	{
		WScript.Sleep(100);
		if(verbose)
		{
			while(!oExec.StdOut.AtEndOfStream)
				doOut(oExec.StdOut.ReadLine())
		}
		else
		{
			while(!oExec.StdOut.AtEndOfStream)
				oExec.StdOut.ReadLine()
		}
		while(!oExec.StdErr.AtEndOfStream)
			doOut(oExec.StdErr.ReadLine())
	}

	if(verbose)
	{
		while(!oExec.StdOut.AtEndOfStream)
			doOut(oExec.StdOut.ReadLine())
	}
	else
	{
		while(!oExec.StdOut.AtEndOfStream)
			oExec.StdOut.ReadLine()
	}
	while(!oExec.StdErr.AtEndOfStream)
			doOut(oExec.StdErr.ReadLine())

	if(oExec.ExitCode==0)
	{
	}
	else
	{
		doOut("Task failed with error code : "+oExec.ExitCode);
		WScript.Quit(1);
	}
}
</script>
</job>
```

7 JOB DEFINITION XML FILES

7.1 Overview of Job Definition Creation and Processing



Autobahn Job Definitions are represented as XML files which can be created "by hand", or more likely by using the Autobahn Administrator (see section 5). A job definition file contains certain standard pieces of information (source folder for example) and also has a "StepType" Definition for each step in the process. It is these step definitions that are executed via the Autobahn service.

7.2 Job IDs

Autobahn uses a sequential integer job id that starts at 1001. The "next job id" value is held in the config/next_job_id/xml file and is updated each time a new job is created or copied using the administration tool.

The initial contents of the file are :

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<next_job_id>1001</next_job_id>
```

7.3 Sample Job Definition File (Excluding Step Details)

Below is a simple example of a job definition which is designed to continuously (every 30 seconds) monitor a directory (c:\faxes) for new .tif files and convert them to searchable PDF, and place the resulting files in c:\processed faxes.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<autobahnjob>
       <jobid>1002</jobid>
       <jobname>Watch Incoming Faxes</jobname>
       <scheduletype>continuous</scheduletype>
       <scheduleevery>30<scheduleevery>
       <scehduleeveryunits>seconds<scheduleevery>
       <schedulefrom>00:00<schedulefrom>
       <scheduleto>23:59<scheduleto>
       <scheduleat></scheduleat>
       <joblogfile></joblogfile>
       <jobsteps>1</jobsteps>
       <jobsourcetype>folder</jobsourcetype>
       <jobsource>c:\faxes</jobsource>
       <jobtarget>c:\processedfaxes</jobtarget>
       <joberrors>c:\errorfaxes</joberrors>
       <jobdeleteonsuccess>yes</jobdeleteonsuccess>
       <jobwork>C:\autobahnwork\1002</jobwork>
       <jobstep>
              <stepsequence>1</stepsequence>
              <steptype>splitpdf</steptype>
              <stepdetails>
              ……(See 6.5 Below)
              </stepdetails>
       </jobstep>
</autobahnjob>
```

XML Element	Description
jobid	Job ID number (see 3.2 above).
jobname	Job Description. Default is “Job %JOBID%”
scheduletype	Ad-hoc, continuous or onceperday
scheduleevery	For continuous, interval between runs
scheduleeveryunits	For continuous, unit of interval between runs
schedulefrom	For continuous, start time for runs
scheduleat	For continuous, end time for runs
scheduleto	For onceperday, the time at which to run
joblogfile	Location of the job logfile. By default the logfile is logs/%JOBID%/%TIMESTAMP%.txt
jobsteps	The number of jobsteps
jobsourcetype	File, folder or tree
jobsource	The source file or folder
jobtarget	The target folder
Joberrors	Folder for job errors. If not specified files that cannot be processed will be placed in jobwork/errors (this is also the default).
jobdeleteonsuccess	If “yes”, when a job has successfully completed, all work files (hence input files) are deleted.
jobwork	The root of the temporary work directories used by the job. The work directories themselves are named work1, work2 etc
jobstep	Contains the definition of a job step
stepsequence	Step sequence number starting at 1.
steptype	The steptype identifies which job element defines the step (see section 7.4 below)
stepdetails	Holds additional step definitions which will be used as the step definition file.

7.4 Sample Step Details

The details of the step are provided here and provide sufficient information for the Autobahn service to execute the step, in conjunction with information in the StepType definitions. Files are moved from the jobsource directory into the work directory specified by <sourcefiles> and the result files are placed in the <target> directory. Upon completion of all the steps, the service will move the files from the final work directory to the <jobtarget> directory.

```
<stepdetails>
       <operation>pdf</operation>
       <sourcetype>folder</sourcetype>
       <sourcefiles>C:\autobahnwork\1002\work1</sourcefiles>
       <target>C:\autobahnwork\1002\work2</target>
       <errors>c:\errorfaxes</errors>
       <joboptions>-d  -k 1  -h 0 -o %FILENAME.pdf -p </joboptions>
       <advancedflags></advancedflags>
       <metadata></metadata>
       <security></security>
       <docoptions></docoptions>
       <custom_script></custom_script>
       <custom_script_pre_job>N</custom_script_pre_job>
       <custom_script_post_job>N</custom_script_post_job>
       <custom_script_pre_folder>N</custom_script_pre_folder>
       <custom_script_post_folder>N</custom_script_post_folder>
       <custom_script_pre_file>N</custom_script_pre_file>
       <custom_script_post_file>N</custom_script_post_file>
       <custom_script_on_error>N</custom_script_on_error>
       <logfile></logfile>
</stepdetails>
```

XML Element	Description
operation	The operation (eg split). This is defined in the step definition file for the steptype.
sourcetype	folder, file or tree.
sourcefiles	Source file or folder.
target	Target folder.
errors	Files that cannot be processed are placed in this directory. Inherited from the Job definition.
joboptions	These are steptype-specific parameters that are derived from the options selected in the Job Designer.
advanced Flags	Additional advanced steptype-specific parameters that can be entered manually only.
metadata	PDF file metadata settings derived from the options selected in the Job Designer.
security	PDF file security settings derived from the options selected in the Job Designer.
docoptions	PDF file open options derived from the options selected in the Job Designer.
custom_script	If not left blank, specifies the name of a custom windows script file in %PDFJUNCTIONDIR%\custom or %TIFFJUNCTIONDIR%/custom that will be called in accordance with the custom_script_* settings below.
custom_script_pre_job	If =Y, the custom script will be called at the start of the job
custom_script_post_job	If =Y, the custom script will be called at the end of the job
custom_script_pre_folder	If =Y, the custom script will be called at the start of processing each folder
custom_script_post_folder	If =Y, the custom script will be called at the end of processing each folder

custom_script_pre_file	If =Y, the custom script will be called at the start of processing each file
custom_script_post_file	If =Y, the custom script will be called at the end of processing each file
custom_script_on_error	If =Y, the custom script will be called when a processing error occurs.
logfile	If specified, output will be logged to a file with this name in %PDFJUNCTIONDIR%\logs or %TIFFJUNCTIONDIR%\logs

Notes

- Note that step-level custom_scripts referred to above as supported by TIFF Junction and PDF Junction are different from “custom scripts” that may be used as a step.

- The step-level log files as referred to above as supported by TIFF Junction and PDF Junction are separate from the overall log file used by Autobahn, and can only be specified manually.

8 EXTENDED STEP TYPES : STEP TYPE DEFINITION XML FILES

Autobahn is designed to be extensible, and as such is driven by StepType Definitions which are used to define each of the possible steps that can be included in a job definition.

A step type definition will include information such as :

- Type of Document Accepted (eg TIFF, Word, PDF)
- Supported Source Types (eg Folder, Email..)
- Type of Document Output (eg PDF)
- Process to Call
- Command Line Parameters

8.1 Sample StepType Definition

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<steptype>tifftopdf</steptype>
<command>cscript %AUTOBAHNROOT%/tj/bin/tiffjunction.wsf
/jobid=%AUTOBAHNTEMP%/%TEMPJOBFILE%</command>
<operation>pdf</operation>
<joboptions>-p</joboptions>
<wrapper>tiffjunction_job</wrapper>
<multithread>yes</multithread>
<validinputtype>tif</validinputtype>
<outputtype>pdf</outputtype>
<iconfile>tifftopdf.bmp</iconfile>
<attributes>
       <attribute>
              <attributeid >SPDF</ attributeid >
              <attributename>Searchable PDF</attributename>
              <attributetype>option</attributetype>
              <attributedefault_value>Yes</attributedefault_value>
              <attributeoptions>
                     <option>
                            <value>Yes</value>
                            <joboptions>-k 3</joboptions>
                     </option>
                     <option>
                            <value>No</value>
                            <attributejoboptions></attributejoboptions >
                     </option>
              </attributeoptions>
       <attribute>
       <attribute>
              <attributeid>ofn</attributeid>
              <attributename>Ouput File Name</attributename>
              <attributetype>value</attributetype>
              <defaultvalue>%FILENAME.pdf</defaultvalue>
              < attributejoboptions >-o%ATTRIBUTEVALUE%</attributejoboptions
>
       </attribute>
<attributes>
```

XML Element	Description
steptype	The name that uniquely identifies the step type
command	The command line to be executed.
operation	The “internal” step type name used by the command
joboptions	Setting for the <joboptions> element in <stepdetails>. This value

	may be added to by attribute settings.
wrapper	Wrapper element to be used when creating temp step job file.
multithread	Yes or no – determines whether multiple instances of the steptype may run concurrently.
validinputtype	Type(s) of files accepted by the step. (NB this is NOT the same as a file extension). * means all types. Multiple values may be separated by commas.
outputtype	Type of file output by the step. (NB this is NOT the same as a file extension)
iconfile	Bitmap file with the steptype icon. The file is assumed to be in the %STEPTYPEDEFDIR% directory.
attribute	The definition of an attribute which the user may set a value for
attributeid	Unique internal name for the attribute
attributename	Display name for the attribute
attributetype	Option or value
attributeoptions	Only applies where attributetype=value. Note that the default option is the first listed
value	Display value
valuejoboptions	String to be appended to <joboptions>
defaultvalue	Default value (for value types)

9 AUTOBAHN DX .NET API

An API is provided which allows a .Net application to allow execution of existing job definitions and creation "on the fly" of new jobs from XML Job Ticket files. The API DLL is located in C:\Program Files\Aquaforest\Aquaforest Autobahn\bin.

9.1 API Example

```
using System;
using System.Threading;
using Aquaforest.Autobahn.API;

namespace ConsoleApplication2
{
	class Class1
	{
		[STAThread]
		static void Main(string[] args)
		{
			IJob job = new Job(1015);
			job.Start();
			job.WaitForExit();

			Console.WriteLine("Job execution finished");
			Console.WriteLine("Exit code is:" + job.ExitCode());
			job.Dispose();

			Console.ReadLine();
		}
	}
}
```

Constructors	public Job(int jobid) *Create Job object using an existing jobid.*
	public Job(string jobdeffile) *Create Job object using a temporary job definition file.*
Properties	public int ExitCode() *Returns the exit code from the Job.*
Methods	public Start() *Starts the Job.*
	public void WaitForExit(int ms) *Waits for the job to exit, for up to ms milliseconds.*
	public void WaitForExit() *Waits indefinitely for the job to compete.*
	public virtual void Dispose() *Disposes of the resources associated with the Job.*

10 PRODUCT FOLDER STRUCTURE

Directory	Contents
Api	Sample API Visual Studio 2003 Project.
bin	Executables, DLLs and WSH files.
config	Next_job_id.xml Jobtmplt.xml Autobahn.config
Custom	Custom Script Files
logs	Default location for log files is logs/%JOBID%/%TIMESTAMP%.txt
jobdef	Contains the Job Definition files.
steptype	Contains the StepType definition files.
Samples	Sample documents
work	Default work directory. Subdirectories are created as required underneath this directory and named 9999/work1, 9999/work2 etc where 9999 is the jobid.
temp	Default directory for temporary files.
pj	PDF Junction Install
tj	TIFF Junction Install
license	Contains key.txt which holds the license key
docs	Includes the Autobahn reference guide.