

Express Server

Watched Folder

- The simplest way to set up Express Server
- Express Server will monitor the input folder at a specific scan interval
- Converted documents will be placed in the output folder
- Folder locations can be local, or on the network

E-mail Processing

- Used to monitor Outlook folders or an Exchange account for input documents
- The Output will be e-mailed back to the sender
- We can limit the attachment size
- POP/SMTP must be used when running Express Server as a service
- Inbox/Outbox requires Outlook to be running at all times

Scripting

- Pre-Scan scripts are executed prior to processing Input Files or Job Tickets for:
 - Document Analysis to drive logic with successive workflow stages
 - On-the-fly XML Job Ticket creation or modification
- Post-Scan scripts are executed after processing Input Files or Job Tickets are completed for:
 - Checking documents into a Repository
 - Completing Document Analysis and Job Ticket Creation for multiple stage processing
- Scripts can also perform file system operation such as renaming

Job Tickets

- A Job Ticket is a control file that is written in XML format and conforms to a DTD that is provided with Express Server
- DTD stands for Document Type Definition and is essentially a list of all the settings that can be included in a Job Ticket
- Any job ticket setting will overwrite settings in the User Interface
- A user will not write Job Tickets. Rather, an application will generate a job ticket based on some user trigger such as clicking a convert button in a custom application



Job Ticket Sample – Merge

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?AdlibExpress applanguage="USA" appversion="3.0.0" dtdversion="2.0.0" ?>
<!DOCTYPE JOBS SYSTEM "C:\Program Files\AdLib\Express\DTD\AdLibExpress.dtd">
<JOBS xmlns:JOBS="http://www.adlibsoftware.com" xmlns:JOB="http://www.adlibsoftware.com">
<JOB>

<JOB:DOCINPUTS>
  <JOB:DOCINPUT FILENAME="Document1.doc" FOLDER="C:\Adlib Express\test" />
  <JOB:DOCINPUT FILENAME="Document2.pdf" FOLDER="C:\Adlib Express\test" />
  <JOB:DOCINPUT FILENAME="Document3.tif" FOLDER="C:\Adlib Express\test" />
  <JOB:DOCINPUT FILENAME="Document4.xls" FOLDER="C:\Adlib Express\test" />
  <JOB:DOCINPUT FILENAME="Document5.htm" FOLDER="C:\Adlib Express\test" />
</JOB:DOCINPUTS>

<JOB:DOCOUTPUTS>
  <JOB:DOCOUTPUT FILENAME="test.pdf" FOLDER="C:\Adlib Express\output" />
</JOB:DOCOUTPUTS>
</JOB>
</JOBS>
```

Job Ticket Sample – Stamping

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?AdlibExpress applanguage="USA" appversion="3.0.0" dtdversion="2.0.0" ?>
<!DOCTYPE JOBS SYSTEM "C:\Program Files\AdLib\Express\DTD\AdLibExpress.dtd">
<JOBS xmlns:JOBS="http://www.adlibsoftware.com" xmlns:JOB="http://www.adlibsoftware.com">
<JOB>

<JOB:DOCINPUTS>
<JOB:DOCINPUT FILENAME="test1.doc" FOLDER="C:\\"/>
</JOB:DOCINPUTS>

<JOB:DOCOUTPUTS>
<JOB:DOCOUTPUT DOCTYPE="PDF" FILENAME="test1.pdf" FOLDER="C:\\"/>
</JOB:DOCOUTPUTS>

<JOB:SETTINGS>
  <JOB:FOOTERS ENABLED="Yes">

<JOB:FOOTER ENABLED="No" FONTNAME="Helvetica-Bold" FONTSIZE="10" FONTCOLOR="0" LAYER="Foreground" MARGINLEFT="0.25"
MARGINRIGHT="0.25" MARGINVERTICAL="0.5" OPACITY="75" ORIENTATION="Page" TEXTCENTER="&[Date]- &[Time]"
TEXTLEFT="&[File]&[CR]&[PageID]" TEXTRIGHT="&[Page] of &[Pages]" PAGES="odd"
PAGENUMBERINGMODE="MergedDocument" />

<JOB:FOOTER ENABLED="Yes" TEXTLEFT="Differt footers on specified pages" PAGES="Even" OPACITY="70" ORIENTATION="Text" />
</JOB:FOOTERS>

<JOB:HEADERS ENABLED="Yes">

<JOB:HEADER ENABLED="Yes" FONTNAME="Courier" FONTSIZE="10" FONTCOLOR="6447714" LAYER="Automatic" MARGINLEFT="0.25"
MARGINRIGHT="0.25" MARGINVERTICAL="0.25" OPACITY="70" ORIENTATION="Page" TEXTCENTER="Header Center Line 1&[CR]&Header
Center Line 2" TEXTLEFT="&[File]&[CR]&[PageID]" TEXTRIGHT="Header Right Line 1&[CR]&HeaderRight Line 2"
PAGES="odd" PAGENUMBERINGMODE="MergedDocument" />
</JOB:HEADERS>

</JOB:SETTINGS>
</JOB>
</JOBS>
```



Job Ticket Sample – Printer Output

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<?AdlibExpress applanguage="USA" appversion="3.0.0" dtdversion="2.0.0" ?>
<!DOCTYPE JOBS SYSTEM "C:\Adlib Express\DTD\AdlibExpress.dtd">
<JOBS xmlns:JOBS="http://www.adlibsoftware.com" xmlns:JOB="http://www.adlibsoftware.com">
  <JOB>
    <JOB:DOCINPUTS>
      <JOB:DOCINPUT FILENAME="Adlib Test.doc" FOLDER="C:\Adlib Express\Input" />
    </JOB:DOCINPUTS>

    <JOB:DOCOUTPUTS>
      <JOB:DOCOUTPUT FILENAME="adlib test.pdf" FOLDER="C:\Adlib Express\output" DOCTYPE="PDF">
        <JOB:DESTINATION>
          <JOB:DESTPRINTERS>
            <JOB:DESTPRINTER PRINTER="HP LaserJet 4100 Series PS" NUMBEROFCOPIES="1" />
          </JOB:DESTPRINTERS>
        </JOB:DESTINATION>
      </JOB:DOCOUTPUT>
    </JOB:DOCOUTPUTS>
  </JOB>
</JOBS>
```

Job Ticket Sample – Job Ticket templates

- A Job Ticket template is a listing of settings that can be referenced and applied to each job ticket
- The Job Ticket will need to have a reference to this template
- The advantage is that you can have a series of template that apply specific conversion rules
- Then, the business application can decide which template to apply based on user input, or document properties

Sample Job Ticket With Template reference

```
<JOBS xmlns:JOB="http://www.adlibsys.com" xmlns:JOBS="http://www.adlibsys.com">
  <JOB>
    <JOB:DOCINPUTS>
      <JOB:DOCINPUT FILENAME="test_generic.doc" FOLDER="C:\Adlib Express\Input"/>
      <JOB:DOCINPUT FILENAME="test_generic.xls" FOLDER="C:\Adlib Express\Input"/>
    </JOB:DOCINPUTS>
    <JOB:DOCOUTPUTS>
      <JOB:DOCOUTPUT DOCTYPE="PDF" FILENAME="test_generic.pdf" FOLDER="C:\Adlib Express\output">
      </JOB:DOCOUTPUT>
    </JOB:DOCOUTPUTS>
    <JOB:SETTINGS MERGE="Yes">

    </JOB:SETTINGS>
  </JOB>
  <JOBTICKETTEMPLATES
    ENABLED="Yes">
    <JOBTICKETTEMPLATE
      ENABLED="Yes"
      MODE="Merge"
      PATH="C:\Adlib Express\Templates\MSWORD_TEMPLATE.xml"/>
  </JOBTICKETTEMPLATES>
</JOBS>
```

Sample Template

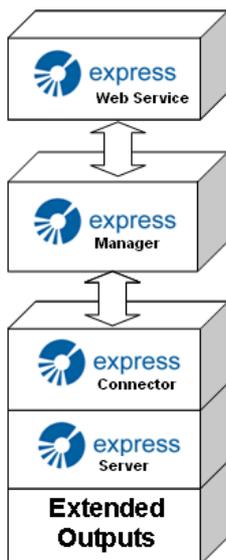
```
<?xml version="1.0" encoding="UTF-16"?>
<?AdLibExpress applanguage="USA" appversion="3.0.0" dtdversion="2.0.0" ?>
<JOBS xmlns:JOBS="http://www.adlibsys.com" xmlns:JOB="http://www.adlibsys.com">
<JOB>
  <JOB:SETTINGS>
    <JOB:NATIVEAPPSETTINGS>
      <JOB:MSWORD_ALLOWA4LETTERPAPERRESIZE="NO" USEPRINTERMETRICS="NO" USEHTMLPARAGRAPHAUTOSPACING="NO" />
    </JOB:NATIVEAPPSETTINGS>
  </JOB:SETTINGS>
</JOB>
</JOBS>
```

Express Web Services

Advantages

- Cross-platform accessibility allows jobs to be submitted from different platforms and applications
- Multi-threaded application for efficient job management
- Can process multiple documents at the same time
- Fault tolerant – job recovery allows files assigned to a failed server to be reassigned
- Incorporates new Express Servers seamlessly with no changes required to your workflow applications
- Rapid communication of job status from Express Server back to the Web Service layer
- Allows job prioritization – which can be altered on the fly

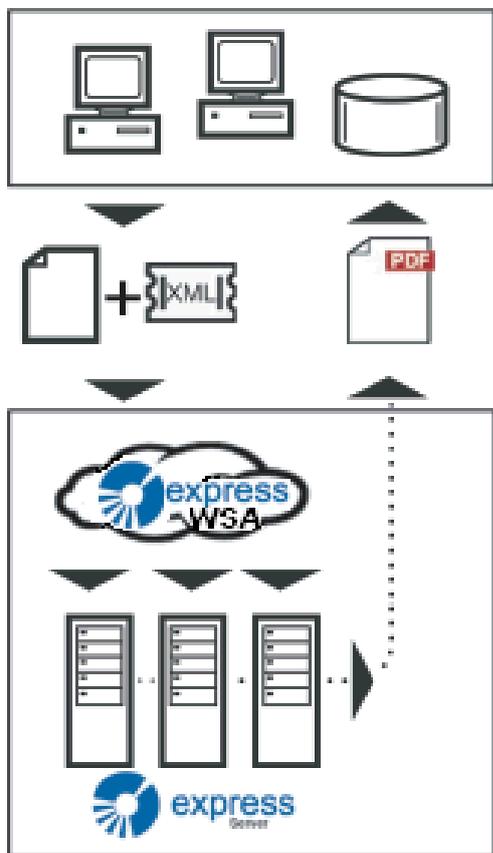
Express Web Services consists of the following 3 Components:



- Express Web Service Adaptor (WSA)
 - Collection of functions available for use by other programs over the Internet
 - Communication with the Web Service is done using standardized XML messaging
- Express Manager
 - Receives jobs from the Express WSA and manages the distribution of jobs to Express Servers for processing
 - Tracks the status of each Job as it is processed by Express Server
- Express Connector
 - Runs on each machine that runs Express Server
 - Receives Jobs from the Express Manager when Express Server is available for processing
 - Monitors Express Server status

How it works in the real world

- An IT Application such as Documentum, Relay or Livelink creates an XML Job Ticket & Job Info based on business rules
- Job Ticket is submitted to the Express Web Service
- Job Info contains Job Priority & Server Group
- Express Web Services delivers the job to an available Express Server for processing
- Job Status is delivered back through Express Web Services to the Business Application
- Express Server processes the job and delivers the resulting files (PDF) to the desired destination (Network, Web, FTP etc.)



ActiveX API

- Used to interface with Express Server directly
- Can only process one document at a time
- Need to develop a lot of code to endure that multiple conversion requests do not occur at the same time
- No fault tolerance or redundancy
- Unsuitable for web based applications