Net-It Enterprise 7.0 SR1

ADMINISTRATION GUIDE

LAST UPDATED, FRIDAY, APRIL 22, 2011
# Table of Contents

**Net-It Enterprise 7.0 SR1 Administration Guide** .............................................................. 1

- Introduction ......................................................................................................................... 1
- Product Version Information ......................................................................................... 1

**Quick Table References** ......................................................................................................................... 2

**Chapter 1: Planning Considerations** ................................................................................................. 3

- Chapter 1 Overview ........................................................................................................... 3
- Importance ......................................................................................................................... 3

- Component Overview ....................................................................................................... 4
  - The Net-It Enterprise .NET Queue Server ...................................................................... 4
  - The Net-It Enterprise .NET Job Processor ...................................................................... 4
- New Features ..................................................................................................................... 4

- System Requirements ...................................................................................................... 5
- Operating Systems ........................................................................................................ 5
- Microsoft Office Versions ............................................................................................ 6
- Scaling Options ................................................................................................................ 6
- Fault Tolerance ............................................................................................................... 7
- Bandwidth ....................................................................................................................... 7

- Firewall and Port Considerations .................................................................................... 8
  - Windows Firewalls and Net-It Enterprise ................................................................. 8
- Supported File Types ..................................................................................................... 8
  - Input File Types (Pre-published Formats) ................................................................. 9
  - Output File Types (Published Formats) ....................................................................... 9

**Chapter 2: Installation** ...................................................................................................................... 10

- Chapter 2 Overview ....................................................................................................... 10

- Installing the Enterprise Components ........................................................................ 11
  - Run the Net-It Enterprise Installation ........................................................................ 11
  - IGC Writer Installation ............................................................................................... 14

- Net-It Enterprise Queue Server and Job Processor Installation ................................ 17

- Installing Autodesk Inventor View for Inventor Support ........................................... 24
- Installation Verification .................................................................................................. 25
  - Job Processor List ..................................................................................................... 25
  - Configuration Check .................................................................................................. 25

- Licensing ......................................................................................................................... 25

- Configuring when Brava! Enterprise .NET is Installed ............................................... 26
  - Configure the products Queue Servers to share the same IIS installation ........... 26
  - Configure the Job Processors to talk to more than one Queue Server ............... 26

**Chapter 3: Usage** .............................................................................................................................. 28

- Chapter 3 Overview ...................................................................................................... 28
INTRODUCTION

This guide provides an overview of the Net-It ® Enterprise product, information about installation and configuration, as well as troubleshooting tips to assist administrators, integrators, and other IT professionals with advanced customizable Net-It Enterprise features. A certain amount of advanced technical knowledge is needed to effectively use this information guide.

PRODUCT VERSION INFORMATION

Version 7.0 SR1
Copyright 2011. All rights reserved.

If you need information on integrating and customizing Net-It Enterprise or if you experience any problems or have general inquiries, please feel free to contact us.

Informative Graphics Corp
4835 E. Cactus Road, Suite 445
Scottsdale, Arizona 85254
Phone: (602) 971-6061
E-mail: info@infograph.com
Support: http://www.infograph.com/support/
Web: http://www.infograph.com
QUICK TABLE REFERENCES

Optional Job Input Parameters (long)
Quick Reference Input Parameters (short)
Job Output Parameters
CDL Error Code Table
Job Processor Parameters
Output Page Size Table
Banner Settings and Banner Macro Table
CHAPTER 1: PLANNING CONSIDERATIONS

CHAPTER 1 OVERVIEW

Net-It® Enterprise (NIE) is a web-based component that allows integrators to publish files and monitor directory structures in an automated way.

IMPORTANCE

Integrators interact with Net-It Enterprise via URL and they are notified of success via HTTP.

- **Publishing files**: Files can be published to XDL (a multi-file, IGC format), CSF (a single file, secure IGC format), PDF, DWF, or TIFF.
- **Monitoring directories**: Net-It Enterprise can be configured to monitor directory structures and publish all files as they are created or changed.
- **TIFF manipulation**: Net-It Enterprise .NET allows users to manipulate TIFF files (rotate, split, merge, or mirror pages).

This chapter contains these sections:

- Planning Considerations
  - Component Overview
  - New Features
  - System Requirements
  - Firewall Considerations
  - Supported File Types
COMPONENT OVERVIEW

Net-It Enterprise .NET consists of two major components: 1) the Queue Server and 2) the Job Processor. The Queue Server holds jobs for publishing and serves those jobs to awaiting Job Processors. The Job Processor does the actual work of publication, and is responsible for calling a notification URL upon job completion. A typical installation consists of a single Queue Server and one or more Job Processors.

THE NET-IT ENTERPRISE .NET QUEUE SERVER

The Queue Server consists of two parts: a set of IIS ASPX pages (ASP.NET) and a Win32 Service (written in .NET 3.5). This guide will refer to the IIS Pages as the "Queue Server", and the Win32 Service as the "QueueService". Both parts are installed when you choose to install the Queue Server.

THE NET-IT ENTERPRISE .NET JOB PROCESSOR

The Job Processor connects to a Queue Server via a port (default, 8890) that is opened by the QueueService. It requests jobs according to the configuration specified in jobprocessor.config. After publishing, it calls the Notification URL (specified in the job) to indicate that the publishing job is complete.

NEW FEATURES

New to 7.0 SR1

- Ability to format extracted text with carriage returns separating each line. (New input param textformatcrlf).

- DExportAllLayers output parameter is replaced with ExportLayerState

New to 7.0

- Added Enhanced Directory Monitoring, making it easy to immediately use Net-It Enterprise features without writing custom integrations and without having to submit "URL" job submissions.

- Added the ability to cancel jobs that are in the queue.

- Support for HTTP POST, HTTP MULTIPARTPOST, as well as HTTP GET for calls the Queue Server and JP notification calls.
Chapter 1: Planning Considerations

- Ability to configure unique publishing output directories.
- Ability to control the font size and style per line in HTML banners.
- New search modes, including the use of REGEX in search strings.
- New job processor parameters
  - PersistentQueues - Addition of persisted publishing queues and timeout settings allows a document, once loaded, to be re-used for multiple jobs
  - PersistTimeout – Timeout for persisted documents
- New optional input parameters (see Input Parameters):
  - TextFormatUTF8 - Support for UTF-8 output format
  - Priority - Addition of Priority Queue allows integer-based job publishing priority
  - OutputColorMode - Forces all content in the document (rasters, vectors, and markups) are set to the chosen color mode: full color, grayscale, or monochrome
  - TiffCompressionType - Ability to specify the TIFF compression algorithm
  - CTBFile – Ability to specify a custom CTB file when publishing Autocad files
  - ExportPageExtents and ExportPageExtentsFile - Ability to export page extents as a separate file.

SYSTEM REQUIREMENTS

OPERATING SYSTEMS

Installing Net-It Enterprise software on certified operating systems allows installers to optimize performance. There are two sets of system requirements related to Net-It Enterprise: Queue Server and Job Processor

Important: UAC (User Account Control) must be disabled on the machine(s) where the Job Processor(s) and Queue Server are installed.

**Queue Server**

The Queue Server requires IIS running on Windows Server 2003, Windows 2008 (32 or 64 bit), or Windows R2 2008. The .NET 3.5 Framework is installed if not detected, and IIS is configured to run ASP.NET applications.

**IIS requirement:** Windows 2008 and 2008 R2 systems need to have the following Role Services enabled for IIS 7:

- IIS 6 Metabase Compatibility
Job Processor

The Job Processor requires Windows XP, Windows 2003, Windows 2008 (32 or 64 bit), or Windows R2 2008. The account under which you run the Job Processor service must have access to the source and target destinations for publication, as well as permission to call the notification URL. The Job Processor installation includes installation of the IGC Writer Printer drivers, necessary for publishing.

Publishing requirement: In order to publish files on the server, the Job Processor(s) must have the .NET Framework 4.0 installed.

MICROSOFT OFFICE VERSIONS

Microsoft Office must be installed on the Job Processor machine to publish DOC, DOCX, XLS, XLSX, PPT, and PPTX formats. The following Office versions are supported in this release:

Office 2010 (32/64 bit)
Office 2007 SP2 (32 bit)
Office 2003* with Compatibility pack for Office 2007 (32 bit)

*Note that although Office 2003 is supported, some Job Processor features, such as XPS publishing, are not available with this version.

If using Microsoft Office 2007 or 2007 SP1, you should have SaveAsXPS installed on your system prior to installing the Net-It Enterprise Job Processor. To do this, you can download the Microsoft “Save As XPS” add-in from: http://www.microsoft.com/downloads/en/details.aspx?familyid=BCE8F991-F0A4-47A0-866B-2FD84A329E02&displaylang=en or update your Office version to 2007 SP2 or Office 2010

SCALING OPTIONS

Net-It Enterprise can be scaled to handle additional publishing loads in two ways. Each Job Processor can be configured to take advantage of more powerful hardware and multiple Job Processors can be installed to handle additional load via more hardware.
SIMULTANEOUS PUBLISHING

Each Job Processor can be configured to perform a certain number of simultaneous conversions for each Queue for which it is responsible. The Job Processor can be configured to take advantage of additional CPU speed, cores and memory. The Job Processor scales almost linearly with number of CPUs/Cores. Additional memory or faster CPU will increase publish speed. Generally speaking, the number of simultaneous conversions should be set to no more than twice the number of processors (or cores) on the job processor machine. You can have more simultaneous processing than number of cores so that the conversion process doesn’t become I/O bound and cause un-used cycles.

NOTE: IGC Writer print-based conversion process users cannot use multiple conversions per machine. In this case, adding job processors is the only scalability option.

ADDING MORE JOB PROCESSORS

Adding more job processors (on dedicated hardware) will increase total throughput somewhat linearly, until network saturation is reached between the Job Processors and the source and target destinations. To handle additional load, more Job Processors can be installed to talk to a single Queue Server. Each Job Processor can be configured to handle the same types of conversion jobs; or multiple Job Processors can each be configured to handle different kinds of publishing jobs. Additional Job Processors can be added to a single Queue Server, up to about 10 Job Processors.

FAULT TOLERANCE

The Net-It Enterprise Job Processor is designed to perform each job in isolation from all other jobs. Thus, if the job has an unrecoverable error (due to malformed input, for instance), then that error does not interrupt processing of other jobs. The Job processor will report the error to the integrator when such an error occurs.

Furthermore, installation of multiple Job Processors on multiple machines provides additional fault tolerance in the case of network failures or hardware failure.

BANDWIDTH

If the files to be published are not on the Job Processor machine, then the IGC Driver technology will generally read the entire file once across the network. All published output is written locally and then copied to the destination specified in the published job. Therefore, bandwidth is directly proportionate to output file size.

The Job Processor will notify, as directed, a URL when publication is complete. These notifications are HTTP calls, and typically less than 2 kb.
FIREWALL AND PORT CONSIDERATIONS

WINDOWS FIREWALLS AND NET-IT ENTERPRISE

Windows (2008 in particular) comes out of the box with the Firewall turned on. This will almost certainly block the Net-It Enterprise system from working properly, especially if the Job Processor is installed on one box and the Queue Server is on another. The following ports, therefore, must be open on any firewall that is installed on Job Processor or Queue Server machines.

QUEUE SERVER:
Allow inbound connections on Port 80 (by default) to allow access to the Queue Server web pages. This port is controlled by the IIS settings for the Queue Server IIS virtual directory.
Allow inbound connections on Port 8890 (by default) to allow the Job Processors to pop jobs from the Queue Server Service. This port is controlled in Server.properties, the default being Pop.Processor.port=8890.

JOB PROCESSOR:
Allow inbound connections on port 7070 to allow access to the Job Processor configuration and status web pages. Controlled in Jobprocessor.config, Pop.Processor.port=7070
Allow outbound connections to port 8890 on the Queue Server so the Job Processor can request jobs from the Queue Server. Controlled in Jobprocessor.config, queue.server.pop.port.0=8890

SUPPORTED FILE TYPES

The following input and output file types are supported by Net-It Enterprise. Note that conversion of Password protected PDF files is not supported on the server.
Chapter 1: Planning Considerations

### INPUT FILE TYPES (PRE-PUBLISHED FORMATS)

<table>
<thead>
<tr>
<th>Type</th>
<th>DOC, DOCX</th>
<th>XLS, XLSX</th>
<th>PPT, PPTX</th>
<th>PDF</th>
<th>000</th>
</tr>
</thead>
<tbody>
<tr>
<td>906</td>
<td>907</td>
<td>BMP</td>
<td>CAL</td>
<td>CCZ</td>
<td></td>
</tr>
<tr>
<td>CG4</td>
<td>CGM</td>
<td>CIT</td>
<td>CM1</td>
<td>MI</td>
<td></td>
</tr>
<tr>
<td>DCX</td>
<td>DFT</td>
<td>DG</td>
<td>DGN</td>
<td>DGN7</td>
<td></td>
</tr>
<tr>
<td>DIF</td>
<td>DWF</td>
<td>DWG</td>
<td>DX</td>
<td>DXF</td>
<td></td>
</tr>
<tr>
<td>EDC</td>
<td>EDM</td>
<td>FTK</td>
<td>G3</td>
<td>G4</td>
<td></td>
</tr>
<tr>
<td>GP4</td>
<td>IGS</td>
<td>ISO</td>
<td>JPG</td>
<td>JPEG</td>
<td></td>
</tr>
<tr>
<td>MIL</td>
<td>MOT</td>
<td>MVS</td>
<td>PCX</td>
<td>PLT</td>
<td></td>
</tr>
<tr>
<td>PRT</td>
<td>PS</td>
<td>RLC</td>
<td>RLE</td>
<td>RNL</td>
<td></td>
</tr>
<tr>
<td>SLDDRW</td>
<td>TIF</td>
<td>TIFF</td>
<td>DSN</td>
<td>TXT</td>
<td></td>
</tr>
<tr>
<td>PNG</td>
<td>HGL</td>
<td>IDW*</td>
<td>APX</td>
<td>GIF</td>
<td></td>
</tr>
<tr>
<td>CC</td>
<td>PSD</td>
<td>HTM</td>
<td>HTML</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


### OUTPUT FILE TYPES (PUBLISHED FORMATS)

<table>
<thead>
<tr>
<th>Type</th>
<th>CSF</th>
<th>XDL</th>
<th>PDF</th>
<th>TIFF</th>
<th>DWF</th>
</tr>
</thead>
</table>

CHAPTER 2: INSTALLATION

CHAPTER 2 OVERVIEW

This topic covers the basic installation steps to install the .NET 3.5 Runtime, Queue Server, IGC Writer, and Job Processor. Not all installations require all components, and exceptions are detailed here.

The installation steps are outlined in an InstallShield Wizard for convenience and ease of installation. However, a complete understanding of the installation choices affords optimal setup.

This chapter contains these sections:

Running the InstallShield Wizard
- Queue Service and Job Processor Service Account Information
- IGC Writer Printer Driver Installation
- Queue Server and Job Processor Installation
- Installing Inventor View for IDW support
- Installation Verification
- Licensing
- Configuring when Brava! Enterprise .NET is Installed
INSTALLING THE ENTERPRISE COMPONENTS

RUN THE NET-IT ENTERPRISE INSTALLATION

1. Double-click on the self-extracting zip file that you downloaded. A Net-It Enterprise installation browser launches automatically. Click the **Install Net-It Enterprise 7.0** button to launch the InstallShield Wizard.

![Net-It Enterprise Installation Screen](image)

2. Setup will extract the MSI file to prepare for installation:

   ![InstallShield Wizard](image)

3. On the Welcome screen, click **Next**.
4. Read the *End user License Agreement* and select "I accept..." if you have read, understand, and agree to the terms of the Licensing Agreement. Click **Next** to continue.
5. Note that Microsoft .NET Framework 3.5 is a requirement of the Net-It Enterprise Server and .NET Framework 4.0 is a requirement of the Net-It Job Processor. One or both of these will appear in the prerequisite list if not found on your machine. These installations will run if not detected, and a system and setup restart will be required.

![Image of installing .NET Framework](image1.png)

To run this application, you must install one of the following versions of the .NET Framework:

- v4.0

Contact your application publisher for instructions about obtaining the appropriate version of the .NET Framework.

6. After these steps, you may choose to install a Queue Server, a Job Processor, or both components from the **Setup Type** screen.

![Image of setup type](image2.png)

You must install one Queue Server and at least one Job Processor. When installing the Job Processor, the IGC Writer Printer Driver is also installed to support print-capture based publishing. The simplest installation for evaluation purposes is to install the Queue Server and Job processor on the same machine at the same time.
The following account information is needed for the Queue Service and Job Processor installations:

**Queue Service**

The Queue Service should be installed to run as an administrator account. Security settings in the `web.config` and `QueueService.exe.config` are preconfigured to allow the Queue Service and the ASP pages to communicate properly.

**Job Processors**

You should select or create a domain user for your Job Processor logon. This account must have access to the source and destination directories that are referenced in the jobs submitted for publishing.

**DIRECTORIES**

By default, the installation will create the following directories (`Program Files` on 32 bit machines and `Program Files (X86)` on 64 bit machines):

- `C:\Program Files\IGC\Net-It Enterprise .NET`, which contains the common configuration file, `Server.Properties`, shared by the QueueServer and the QueueService. User guides are also installed to this directory.
- `C:\Program Files\IGC\Net-It Enterprise .NET\QueueServer`, which contains the ASP.NET based QueueServer.
- `C:\Program Files\IGC\Net-It Enterprise .NET\QueueService`, which contains the QueueService.
- `C:\Program Files\IGC\Net-It Enterprise .NET\JobProcessor`, which contains the Job Processor and associated support files.

**IGC WRITER INSTALLATION**

1. If the setup did not detect the correct version, the IGC Writer installation begins to install the IGC Writer printer driver. On the **Welcome** screen, click **Next >**, then click **Yes** on the license agreement screen after you have read and agreed to the terms of use.
2. In the *Choose Destination Location* screen, set the desired destination folder or accept the default location. (IGC strongly recommends accepting all the defaults.) Click **Next**.
3. In the *Ready to Install the Program* screen, click **Install**. The Net-It Enterprise files are installed and the Printer Driver is automatically configured.
4. When installation has completed, click **Finish** at the *InstallShield Wizard Complete* screen.

**Notes:**

- By default, the IGC Writer is installed under the same user account as the Job Processor. If you are running the Job Processor under a different account, you may have to manually set the CSF Writer properties for print-publishing to work correctly, but this setup is not recommended, nor has it been tested.

- When installing to a Japanese environment only, please follow the instructions provided through this link for switching Office IME 2007 and Legacy MS IME: [http://support.microsoft.com/kb/932104/ja](http://support.microsoft.com/kb/932104/ja)

---

**NET-IT ENTERPRISE QUEUE SERVER AND JOB PROCESSOR INSTALLATION**

1. In the *Choose Destination Location* screen, select a location for the Net-It Enterprise files or click **Next** to accept the default location and continue.
2. In the **Setup Type** screen, select Yes if you would like the install to setup and configure the Enhanced Directory Monitoring feature. The installer creates a default Enhanced Directory Monitoring setup on the QueueServer which can be used immediately. If you select No, you may configure this feature at any time after installation completes.
3. If you have selected to install this feature, the *Enhanced Directory Monitoring Share* screen displays. Enter the path to a unique shared directory that you would like to use for the Enhanced Directory Monitoring feature. **Projects names (the final directory in the full path) MUST BE UNIQUE across the entire system.**

![Enhanced Directory Monitoring Share](image)

4. In the *Enter machine information* screen, enter the name of web site that you would like Net-It Enterprise to use and click **Next**.
5. In the Enter machine information screen, enter the name of the virtual directory that the Net-It Enterprise Queue Server will use and click **Next**.

- Enter the port number that your web server uses and click **Next**. If you are using the default value of port 80, simply click **Next**.
6. Click **Install** on the *Ready to Install the Program* screen and the program files are installed.
7. You are given the option to enter your service login information if you plan to run the Job Processor or Queue Server as a service

If you choose Yes, enter your Log On user name and password that will be used to start the service and click Next.

1) If you install the Job Processor as a service (called Net-It Enterprise .NET) **YOU MUST ENTER A USER ACCOUNT THAT HAS ADMINISTRATOR RIGHTS ON YOUR MACHINE** in order to avoid permissions errors after the installation. The Net-It Enterprise service cannot run correctly as the Local System Account because the service must interact with other processes and components. Attempting to run the Net-It Enterprise service as the Local System Account will cause unexpected and undesirable behaviors with Net-It Enterprise.

2) If the Job Processor is NOT installed as a service, it must be started manually through your Windows Start menu and be running on the desktop in order for publishing to occur. If you log out of the server, this service will stop.
8. You are asked if you would like to start the Net-It Enterprise Queue Server and/or Job Processor Service. Select **Yes** to start the services or **No** to start the services manually. If you select **No**, you will need to restart the machine to automatically start the service.

9. On the *Installation Complete* screen, click **Finish** to complete the Net-It Enterprise installation.
You must install Autodesk Inventor® View in order for the Job Processor to publish Inventor files. Net-It Enterprise Job Processor supports Inventor natively using the IDW2DL.dll driver with Autodesk Inventor View 2008 and you do not have to use the Inventor Application print driver to publish the .IDW file format.

1. Log on to the machine (Windows 2003 Server or Windows 2008 Server) where the Job Processor is installed.
4. Install Autodesk Inventor View 2008, accepting all of the default values.
5. If you have one or more Job Processors installed on different servers, repeat steps 1 through 4 for each Job Processor server.

*Note that if installing on a Windows XP Job Processor machine, Inventor View 2009 is supported.
After completing the initial installation, you can verify the installation by locating a new set of Menu Items under the Start Menu. Launch All Programs -> IGC -> Net-It Enterprise .NET 7.0 -> Net-It Enterprise .NET Status to display a copy of your default web browser launches and the Net-It Enterprise .NET Status Page.

Below the Queue Statistic block of information you should see a list of your installed Job Processors. If you don’t see an installed Job Processor in this list, then it is not communicating with the QueueService.

Check the configuration in the JobProcessor.config file. Verify that the value for queue.server.address points to the Queue Server machine and that the port specified by queue.server.pop.port is accessible across your network.

When the status of each of your Job Processors displays as "UP" on the Net-It Enterprise .NET Status page, the system is ready for use.

By default, the Net-It Enterprise product installs a 5-user 30 day evaluation key. You will receive a license file called IGCKey.lic from Informative Graphics when you license your installation. Once you obtain this permanent license key, you must copy the file to the following installation directories to replace the 30 day evaluation license.

C:\Program Files (x86)\IGC\Net-It Enterprise\JobProcessor

Note that the default locations given above may be different in your environment if the installation location is changed.
A restart of the Job Processor is not necessary after updating the license file.

**CONFIGURING WHEN BRAVA! ENTERPRISE .NET IS INSTALLED**

Many customers desire to run both Brava! Enterprise .NET (BEN) and Net-It Enterprise together on the same machine: BEN for review, markup, and collaboration, and Net-It Enterprise for batch processing both at the beginning and at the end of a workflow. You may run both of these Informative Graphics products together on the same machine, however, doing so requires the following configuration updates be made to the servers and Job Processors.

There are two general steps involved in configuring BEN and Net-It Enterprise to run on the same machine. Each of these steps is discussed in the following sections:

- Configure the products Queue Servers to share the same IIS installation
- Configure the Job Processors to talk to more than one Queue Server

**CONFIGURE THE PRODUCTS QUEUE SERVERS TO SHARE THE SAME IIS INSTALLATION**

1. You can install the BEN Server and the Net-It Enterprise Queue Server in either order and they may even use the same Web Site.
2. By default, BEN and Net-It Enterprise use the same port (8890) to submit jobs to the Job Processor. One of these ports must be changed. In the following example, we change the Net-It Enterprise port to 8891, although any port you wish may be used.
   a. Open `server.properties` in the root installation folder of Net-It Enterprise (by default `C:\Program Files (x86)\IGC\Net-It Enterprise`)
   b. Locate and edit the `Pop.Processor.port` setting from 8890 to 8891.
3. Restart the Queue Server Service and IIS after saving your changes.

**CONFIGURE THE JOB PROCESSORS TO TALK TO MORE THAN ONE QUEUE SERVER**

1. Stop the Job Processors and open `jobprocessor.config` in a text editor.
2. Locate and copy the following lines:
   ```
   jobgetter.classname.0=IGC.JobProcessor.PopProcessorJobGetter
   ```
3. On the next line, insert (paste) a copy of these lines into the config file and edit the pasted lines as follows:

```
jobgetter.classname.1=IGC.JobProcessor.PopProcessorJobGetter
queue.server.address.1=localhost
queue.server.pop.port.1=8891
```

Note that we have changed the port to be 8891 (matching what was entered in `server.properties`), and the .1 notation (replacing what was .0) tells the JP to have a second set of job getters. In many production environments, a single Job Processor may not be adequate to handle the traffic load coming from both servers. You may need to set up multiple Job Processors to meet your specific load requirements.

4. Verify that the threads are set up appropriately for BEN as shown below. (BEN requires more queues than Net-It Enterprise.) Note that the number for each thread may vary.

```
thread.drw=3
thread.pdf=3
thread.single=3
thread.prq=1
thread.doc=1
```

5. Save the `jobprocessor.config` file and start the Job Processor.
CHAPTER 3: USAGE

CHAPTER 3 OVERVIEW

Information presented in this topic will help you properly set up and use the Net-It Enterprise product to publish files as intended, to monitor directories as desired, and to more efficiently address security needs using search strings and various scripts. You will be able to identify efficiencies and manage your jobs successfully using the information covered in this topic.

This chapter contains these sections:

- **Publishing a File**
  - Sample Job
  - Cancelling a Job
- **Monitoring a Directory**
  - Read and Write Access
  - Configuring Unique Output Directories
- **Receiving Results via a Notifier**
  - Using `ResultFilterSettings.txt`
- **Enhanced Directory Monitoring**
  - How the system works
  - Anatomy of a task
  - Paired Files
  - Setting up and Managing Projects and Tasks
- **Manipulating TIFF Files**
- **Security XML Details**
  - Specific Attribute Notes
  - ISO Banner and Watermark Settings
  - Banner Macros table
Chapter 3: Usage

PUBLISHING A FILE

To publish files with Net-It Enterprise, jobs are submitted to the Queue Server. Jobs are the basic unit of work for Net-It Enterprise and each job represents a request to publish a file, to monitor a directory for files to publish, or to manipulate a TIFF file.

For jobs that publish a file, the job must contain the following information:

- **Source**: The full path to the source file.
  (For multi-doc publishing, you can set multiple source files as sourceN to publish as one export file)

- **Target**: The target directory where the publishing rendition of the file will be placed after publishing is complete.

  Note that only network UNC paths are supported for Target and Source parameter path values. Local paths (but not mapped drives) can be used for single box testing only.

- **OutputFormat**: The type of output you would like; this format can be one of "XDL", "CSF", "PDF", "TIFF" or "None" if the desired outcome is only for Thumbnails or artifacts.

- **NotificationUrl**: A URL (in the format http://Server:port) that will be called upon completion of the publishing job. It will contain details of any errors that may have occurred, along with all of the original parameters sent with the job.

In addition, there are many optional job options which can be added to the basic job and are detailed in Chapter 4, Job Options.

Also, see the Enhanced Directory Monitoring topic for information on how end users can easily configure and publish jobs through the Windows file system without any programming knowledge.

SAMPLE JOB

To submit a job, call the push.aspx page on the Queue Server, and pass the parameters as Query String Arguments

Given a sample job with the following parameters:

- `source=\server\sourcefiles\netit\906\AIRPLANE.906`
- `target=\server\outputfiles\destinationdirectory`
- `outputformat=pdf`
- `notificationurl=http://localhost:9999`

It will look like this when pushed:

```
http://servername/queueserver/push.aspx?Source=\server\sourcefiles\netit\906\AIRPLANE.906&target=\server\outputfiles\destinationdirectory&outputformat=pdf&notificationurl=http://localhost:9999
```
Upon completion of this job, the output published by Net-It Enterprise will be a file named *airplane.pdf*, located in the directory `\server\outputfiles\destinationdirectory`. The notification URL (`http://localhost:9999`) will be called with the following items on the QueryString:

```
source=\server\sourcefiles\netit\906\AIRPLANE.906
target=\server\outputfiles\destinationdirectory
outputformat=pdf
notificationurl=http://localhost:9999
threadid=drw5
type=drw
jobid=1278684777549834871
filename=AIRPLANE.906
starttime=3/14/2006 3:09:35 PM
tempdir=<InstallPath>\JobProcessor\bin\release\tempdir\1278684777556084752
ext=906
targetfile=\<InstallPath>\JobProcessor\bin\release\tempdir\1278684777556084752\AIRPLANE.pdf
endtime=3/14/2006 3:09:37 PM
totaltime=00:00:02.3264386
mainfile=AIRPLANE.pdf
```

This query contains information about how long the job took (the "totaltime" parameter), and some information about the temporary files used by Job Processor. The presence of the "mainfile" parameter is the key indication of success, and is found in the directory pointed to by the "target" parameter. If errors or warnings are generated, then they will be described in the additional error details parameters (error 0, error 1, warning 0, warning1, etc), if available.

### CANCELLING A JOB ON THE QUEUE SERVER

If you wish to submit a job that you may wish to cancel, submit a job to the queue server with a *cancelid* that is set to a unique string. You will need to keep track of these unique string values in order to cancel the job successfully. For example:

```
cancelid=<unique string>
```

*Cancelid* is not required; however, the value must be unique among all the jobs in the queue. If a job is submitted with duplicate *cancelid*, the queue server will return the error: “Unable to submit job: Failed to queue publishing request. Attempt to submit two jobs with the same *cancelid*”

In order to cancel a job, use the *cancel.aspx* entry point. Call with the *cancelid* parameter set to the value established when the job was enqueued. For example;

Cancel.aspx returns HTTP OK if the job is successfully cancelled or returns an error message if the cancelid does not exist. Once a job is cancelled, it is completely removed from the queue server.

---

**MONITORING A DIRECTORY**

**MONITORING FOR CHANGES**

Net-It Enterprise can be configured to monitor one or more directories for changes to files. When a directory is monitored, any file that gets created or changed will be submitted for publishing. To monitor a directory, submit a job to the push.aspx page (described above) with the following parameters:

- sourcedir=<a directory to monitor>
- outputdir=<a directory (that exists) where output is placed)
- outputformat=pdf
- notificationurl=http://localhost:9999
- register_dir=True

Any of the optional job parameters can be added and will be applied to all documents "submitted" to the monitored directory. This list above shows the minimum needed for a job.

To stop monitoring a directory, submit a job with the same sourcedir and outputdir values and register_dir=False.

Because the sourcedir and outputdir must be accessible to all Job Processors in the system, they are typically specified as UNC Paths, not absolute drives.

---

**READ AND WRITE ACCESS**

The account used by the QueueService must have read and write access to the directory specified by the sourcedir parameter. The account used by the Job Processor must have read access to the directory specified by the sourcedir parameter and it must have read and write access to the directory specified by the outputdir parameter.

Directory monitoring is recursive. If you monitor a top level directory, then changes and creations in subdirectories will be published. The output directory will contain a mirror of the same directory structure.

**Note:** You shouldn’t publish to XDL when monitoring a directory unless you configure a unique output directory (see next section). XDL is a multi-file format, and name collisions can happen if you publish multiple files in the same directory. You can use XDL as an output file type if you monitor a directory where each file exists exclusively in its own directory.
Since the Queue Service performs directory monitoring, changes will result in files inserted into the queue and published by the next available job processor.

When monitoring a directory, the job processor will send the notification URL for each file that gets published.

**CONFIGURING UNIQUE OUTPUT DIRECTORIES**

The Queue Server, when in Directory Monitor mode, can be configured to allow generation of a unique output directory per published file. This allows directory monitoring to work with all features, including thumbnails and text files, without overwriting published files.

To use this feature, integrators submit the directory monitor job request with the parameter `dirmonuniquedirectory=true`. If this parameter is set to true, then the Queue Server will create a unique directory for each job it publishes, while still maintaining the hierarchy of the monitored directory tree.

Example:

If monitoring `\mydirectory\monitor` and publishing to `\mydirectory\output` and a file named `motor.dwg` is created in `\mydirectory\monitor`, then the output directory will be something like:

`\mydirectory\output\motor.dwg <time>\motor.pdf`

If a file named `lawnmower.dwg` is created in `\mydirectory\monitor\subdir`, then the output directory will be something like:

`\mydirectory\output\subdir\lawnmower.dwg <time>\lawnmower.pdf`

<Time> is UTC and formatted as Year-Month-Day Hour_Minute_Second, for example:

`2009-12-18 15_41_33Z`

If collisions occur, an additional -1, -2, etc will be appended to the time string.

See Also: [Enhanced Directory Monitoring](#)

**RECEIVING RESULTS VIA A NOTIFIER**

Using the same example job we discussed earlier, this section will discuss what the Job Processor does when a job is completed.
The Job Processor calls the URL http://yetanotherserver/netitnotifier.html with the http verb GET, POST, or MULTIPARTPOST (as determined by the value of parameter notificationverb), and the results of the job in the query string, just like the call to push. The results are always name=value pairs, and all of the parameters submitted to push.aspx are included in the results, along with other information about the job, such as how long it took and what, if any, errors occurred.

The notificationurl can be set to either POST or MULTIPARTPOST. If you specify MULTIPARTPOST, then the content type is multipart/form-data, and each parameter is a separate data element. If you specify POST, then the content is application/x-www-form-urlencoded and the parameters of the reply are in post data stream.

Because HTTP Post calls can often have many more parameters than GET calls, the Queue Server can remove non-job related parameters from the job before submitting it to the Job Processors. These parameters are called noise words and are defined in web.config via the JOB_NOISE_WORDS parameter.

The default list of parameters removed is:

<table>
<thead>
<tr>
<th>parameter</th>
<th>ASP.NET_SessionId</th>
<th>ALL_HTTP</th>
<th>ALL_RAW</th>
<th>APPL_MD_PATH</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPL_PHYSICAL_PATH</td>
<td>CONTENT_LENGTH</td>
<td>GATEWAY_INTERFACE</td>
<td>HTTPS</td>
<td></td>
</tr>
<tr>
<td>INSTANCE_ID</td>
<td>INSTANCE_META_PATH</td>
<td>LOCAL_ADDR</td>
<td>PATH_INFO</td>
<td></td>
</tr>
<tr>
<td>PATH_TRANSLATED</td>
<td>QUERY_STRING</td>
<td>REMOTE_ADDR</td>
<td>REMOTE_HOST</td>
<td></td>
</tr>
<tr>
<td>REMOTE_PORT</td>
<td>REQUEST_METHOD</td>
<td>SCRIPT_NAME</td>
<td>SERVER_NAME</td>
<td></td>
</tr>
<tr>
<td>SERVER_PORT</td>
<td>SERVER_PORT_SECURE</td>
<td>SERVER_PROTOCOL</td>
<td>SERVER_SOFTWARE</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>HTTP_CONNECTION</td>
<td>HTTP_HOST</td>
<td>AUTH_TYPE</td>
<td></td>
</tr>
<tr>
<td>AUTH_USER</td>
<td>AUTH_PASSWORD</td>
<td>LOGON_USER</td>
<td>REMOTE_USER</td>
<td></td>
</tr>
<tr>
<td>CERT_COOKIE</td>
<td>CERT_FLAGS</td>
<td>CERT_ISSUER</td>
<td>CERT_KEYSIZE</td>
<td></td>
</tr>
<tr>
<td>CERT_SECRETKEYSIZE</td>
<td>CERT_SERIALNUMBER</td>
<td>CERT_SERVER_ISSUER</td>
<td>CERT_SERVER_SUBJECT</td>
<td></td>
</tr>
<tr>
<td>CERT_SUBJECT</td>
<td>CONTENT_TYPE</td>
<td>HTTPS_KEYSIZE</td>
<td>HTTPS_SECRETKEYSIZE</td>
<td></td>
</tr>
<tr>
<td>HTTPS_SERVER_ISSUER</td>
<td>HTTPS_SERVER_SUBJECT</td>
<td>HTTP_CONTENT_LENGTH</td>
<td>HTTP_CONTENT_TYPE</td>
<td></td>
</tr>
<tr>
<td>HTTP_EXPECT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

As an integrator, you must write code to handle this call to be able to use the parameters appropriately.
USING RESULTFILTERSETTINGS.TXT

An event system is in place on the Job Processor that allows administrators to filter completed jobs and take certain actions when the jobs meet specified criteria.

**General notes:**

- Filters are set up on the Job Processor via a file called `resultfiltersettings.txt`
- Administrators can match job data names or values via Regular Expression. If any of a job's names (keys) or value's match a filters search pattern, the notification action is taken.
- Notifications can either be via email, or an event can be placed in the application log on the Job Processor machine.
- Notifications can be configured to be sent once in a certain time period (specified in seconds)
- By default, only licensing warnings are configured to be sent to the application log, and they're configured to do this once a day.

The configuration file, `resultfiltersettings.txt`, contains:

```plaintext
SMTPServer=
SMTPReplyTo=

One or more Filter Entries. Each Filter Entry must contain:

- **FilterNameN** - the unique name of the file
- **DestinationN** - the destination for events, either Application for the application event log or a semicolon separated list of email addresses
- **FrequencyN** - the frequency, in seconds, to set this event

Either **NameN**, **ValueN**, or both. The regular expressions used to match the event

If you wish to use email notifications, then the **DestinationN** line can contain multiple email addresses, separated by semicolons. If you use email, you must configure the **SMTPServer** and **SMTPReplyTo** lines in `ResultsFilterSettings.txt` file:

```plaintext
# SMTP Server for email
SMTPServer=mailserver.company.com

# Address to set into emails as a reply to
SMTPReplyTo=William.Shakespeare@company.com
```

Entries must be numbered starting at zero and increasing one number at a time.
SAMPLE FILTER ENTRY IN RESULTFILTERSETTINGS.TXT:

# Entry 0. Put all errors in the Application event log.
FilterName0=Error in Publish Job
Name0=error.*
Destination0=Application
Frequency0=0

The above sample means:

<table>
<thead>
<tr>
<th># Entry 0. Put all errors in the Application event log.</th>
<th>This is a comment, comments begin with a # sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>FilterName0=Error in Publish Job</td>
<td>This is the name of the filter. Filter names must be unique</td>
</tr>
<tr>
<td>Name0=error.*</td>
<td>This line means match any line of job data with a name that begins with &quot;error&quot;</td>
</tr>
<tr>
<td>Destination0=Application</td>
<td>This will put a filter entry in the Application event log</td>
</tr>
<tr>
<td>Frequency0=0</td>
<td>This means put an entry in the application log every second</td>
</tr>
</tbody>
</table>

Following is an example file with multiple filters configured:

# SMTP Server for email
SMTPServer@mailserver.BirnamWood.com
# Address to set into emails as a reply to
SMTPReplyTo=Macbeth@BirnamWood.com

# Entry 0. Email license warnings to an interested party
FilterName0=License Warning
Name0=LicenseWarning
Destination0=Macduff@BirnamWood.com;ladymcb@birnamwood.com
Frequency0=1440

# Entry 1. Put all errors in the Application event log.
FilterName1=Error in Publish Job
Name1=error.*
Destination1=Application
Frequency1=0

# Entry 2. If we publish a file named Airplane, note it.
FilterName2=Airplanes
Value2=.*[Aa]irplane.*
Destination2=Application
Frequency2=0
MANIPULATING TIFF FILES

Net-It Enterprise .NET has the ability to rotate, split, merge, and mirror TIFF files. This allows you to submit jobs to the system as desired, including rotating a scanned image, merging two or more tiffs, splitting large files into smaller ones, or mirroring images. Tiff manipulation jobs are submitted to the push.aspx interface, as above, and contain the following parameters:

Parameters required for all Tiff manipulation:

- **task=imageop**
- **operation** = <one of split, merge, rotate, or mirror>
- **NotificationUrl** = http://yetanotherserver/netitnotifier.html

Each operation has a required set of parameters. They are:

- **split** - Splits multi-page source file at page

  Required Parameters:

  - **task=imageop**
  - **operation=split**
  - **source=<path to file to be processed>**
  - **targetfile1=<path to first output file>**
  - **targetfile2=<path to second output file>**
  - **page=<page number to split from, first page is 1>**

- **merge** - Merges two files into one

  Required Parameters:

  - **task=imageop**
  - **operation=merge**
  - **source1=<path to first file>**
  - **source2=<path to second file>**
  - **targetfile=<path to outputfile>**

- **rotate** - Rotates image the number of degrees passed in

  Required Parameters:

  - **task=imageop**
  - **operation=rotate**
  - **source=<path to file to be processed>**
  - **targetfile=<path to output file>**
page=<page(s) to apply change to comma delimited list. Set to "all" for all pages>
degrees=<number of degrees to rotate file>

**mirror** - mirror image on x or y axis

Required Parameters:
- `task=imageop`
- `operation=mirror`
- `source=<path to file to be processed>`
- `targetfile=<path to output file>`
- `page=<page(s) to apply change to comma delimited list. Set to "all" for all pages>`
- `axis=<the axis to use, "y" or "x">`

### SECURITY XML DETAILS

Security settings for published CSF, PDF, and TIFF files are specified in the value (XML file path) set for the `securityxmlfilename` parameter. An example XML rights file is provided in the Job Processor installation directory as `security.xml`. The security XML file Visual Rights and password apply only to CSF output and disabling (false) any Visual Rights via a Security XML file for any other output type will cause publishing to fail. If a password or any other Visual Right is set on a published file, the file's thumbnail image will display with a default CSF icon and not a viewable thumbnail image of the document.

![CSF Icon](image)

### SPECIFIC ATTRIBUTE NOTES

There are two attributes that you can specify for the `DateExpired` attribute element of published CSF files. Note that if you set the `relativedays` attribute, it then takes precedence over the year, month, and day (absolute) date attributes. The `relativedays` attribute takes the current date (at the time of the publishing) and adds the specified number of days.

Within the `<RightFlags>` element, the right corresponding to the element `AuthorAndReviewMarkups` encompasses the `ReviewMarkups` right. In other words, if `AuthorAndReviewMarkups` is set to false, it means the settings takes precedence if the `ReviewMarkups` right is also set to false.
ISO BANNER SETTINGS

The ISO banners are strings of specific information (date, time, page number, user name, etc.) assigned to a location on the document header and footer. The watermark is a semi transparent character string that stretches from the lower left corner to the upper right corner of the printed or on screen document.

Watermarks and publish banners can be used to support ISO 9000 and QS 9000 quality standards. They can contain text strings, token values, or, optionally (through integrations API), metadata fields from document management. With 12 possible banner locations allowing 10 lines of data per location, you can add as much text as you need to provide sufficient traceability and status.

Note that when running the product under a Demo license, the watermark cannot be configured and will only display and publish the string "Evaluation".

SECURITY XML BANNER FONT SETTINGS

You can specify the font, color, style, height, and other attributes for the print and publish ISO banners and watermark through use of the IsoBanners attributes of the Security.xml file. Publish banners are those that will display on the published output file of a selected format (electronic).

```xml
<IsoBanners>
  <IsoBannerFont string="Arial"/>
  <IsoBannerFontHeight string="14"/>
  <IsoBannerFontStyle string="0"/>
  <ScreenWaterMark string="ScreenWatermark"/>
  <ScreenBanner string="ScreenBanner"/>
  <PublishBanners>
    <IsoBannerFont string="Arial"/>
    <IsoBannerFontHeight string="14"/>
    <IsoBannerFontStyle string="0"/>
    <IsoBannerColor string="0,0,0"/>
    <WaterMark string="Publish Watermark" editable="true"/>
    <WaterMarkOpacity string=".25"/>
    <TopLeft string="Publish Top Left" editable="true"/>
    <TopCenter string="Publish Top Center" editable="true"/>
    <TopRight string="Publish Top Right" editable="true"/>
    <LeftTop string="Publish Left Top" editable="true"/>
    <LeftCenter string="Publish Left Center" editable="true"/>
    <LeftBottom string="Publish Left Bottom" editable="true"/>
    <RightTop string="Publish Right Top" editable="true"/>
    <RightCenter string="Publish Right Center" editable="true"/>
    <RightBottom string="Publish Right Bottom" editable="true"/>
</PublishBanners>
</IsoBanners>
```
**Note:** If double byte characters are used, the Security XML file should be saved using UTF-8 encoding. Specifying banner information here will force the banners for all users. They will not be able to modify them unless the *editable* flag is set to true.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsoBannerFont</td>
<td>Specifies the default font name to use for banners.</td>
</tr>
<tr>
<td>IsoBannerFontHeight</td>
<td>Specifies the default font size to use for banners.</td>
</tr>
<tr>
<td>IsoBannerFontStyle</td>
<td>Specifies the default font style to use for banners.</td>
</tr>
<tr>
<td></td>
<td>Example to specify Italic underline style: <code>IsoBannerFontStyle string=&quot;6&quot;</code></td>
</tr>
<tr>
<td></td>
<td>Valid values are:</td>
</tr>
<tr>
<td></td>
<td>Normal=0</td>
</tr>
<tr>
<td></td>
<td>Bold=1</td>
</tr>
<tr>
<td></td>
<td>Italic=2</td>
</tr>
<tr>
<td></td>
<td>Underline=4</td>
</tr>
<tr>
<td></td>
<td>These values can be added together as the following examples illustrate:</td>
</tr>
<tr>
<td></td>
<td>Bold italic=3 (1 + 2)</td>
</tr>
<tr>
<td></td>
<td>Bold underline=5 (1 + 4)</td>
</tr>
<tr>
<td></td>
<td>Bold italic underline=7 (1 + 2 + 4)</td>
</tr>
<tr>
<td></td>
<td>Italic underline=6 (2 + 4)</td>
</tr>
<tr>
<td>IsoBannerColor</td>
<td>Specifies the default color to use for the banner font. Format: r,g,b:</td>
</tr>
<tr>
<td></td>
<td>range of 0 to 255. Example: <code>IsoBannerColor string=&quot;255,0,0&quot;</code></td>
</tr>
<tr>
<td>Watermark</td>
<td>Specifies the default text string to use for the watermark. Macro</td>
</tr>
<tr>
<td></td>
<td>commands may be used here (see next table)</td>
</tr>
<tr>
<td></td>
<td>Example: <code>Watermark string=&quot;Confidential&quot;</code></td>
</tr>
<tr>
<td>WatermarkOpacity</td>
<td>Specifies the opacity of the watermark string. Valid values are decimal</td>
</tr>
<tr>
<td></td>
<td>integers from 0 to 100. Example: <code>WatermarkOpacity string=&quot;.75&quot;</code></td>
</tr>
<tr>
<td><strong>Editable</strong></td>
<td>True or False. Determines if user is able to edit the banners and watermark (default is True). If the published output is CSF, the editable flag is false.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>
 or \n (or carriage return)</strong></td>
<td>Determines new line in string (line breaks). Example: <code>&lt;TopCenter string=&quot;First line\nSecond line&quot;/&gt; or </code>&lt;TopRight=&quot;Title:%Title\nPage:%Page&quot;/&gt; or enter with carriage return &lt;Enter&gt;: `&lt;TopCenter string=&quot;First line;Second line&quot;/&gt;</td>
</tr>
<tr>
<td>**ScreenWatermark</td>
<td>ScreenBanner**</td>
</tr>
</tbody>
</table>

## BANNER MACRO COMMANDS

Net-It Enterprise supports the Macro Commands listed here for the watermark and banner text strings. Insert one of these commands in the banner or watermark string value and the string will be dynamically inserted at print or publish time. If a banner or watermark is editable, end-users can enter a percent symbol (%) in the edit field of a print/publish banner or watermark line to summon a list of available tokens. Banners and the watermark can contain text strings or one of the available token values:

<table>
<thead>
<tr>
<th>Macro Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>%Date (%daydate, %d, or %D)</code></td>
<td>Inserts the date the print was spooled. If the tags are viewed on screen, the date at which the screen was last refreshed displays.</td>
</tr>
<tr>
<td><code>%Time(or %t)</code></td>
<td>Inserts the time the print was spooled based on a 12 hour clock (AM/PM). If the tags are viewed on screen, the time on which the screen was last refreshed displays.</td>
</tr>
<tr>
<td><code>%MilTime</code></td>
<td>Inserts the time the print was spooled based on a 24 hour clock.</td>
</tr>
<tr>
<td><code>%Title</code></td>
<td>Inserts the name of the document. If the Title macro is used, the title must be set via the <code>documenttitle</code> parameter</td>
</tr>
<tr>
<td><code>%Page</code></td>
<td>Inserts the page number.</td>
</tr>
<tr>
<td><code>%TotalPages</code></td>
<td>Inserts the total number of pages in macro output.</td>
</tr>
</tbody>
</table>
| `%TotalOriginalPages` | Inserts the total number of pages in the original file (as opposed to the number of pages you are currently exporting, which can be a
%OriginalPageNum | Inserts the number of the page in the original document. (as opposed to its position in the export).
---|---
%Login (or %User) | Inserts the user name of the person who issued the print. If the user or Login macro is used, the job must set the parameter publishusername.
%batespgno(x) | Bates Number This tag is used to indicate the starting page number and the number of digits to use. For example, %batespgno(0002) would place 0002 on the first page, 0003 on the next page, etc.
A Bates Number is considered to be both the prefix and the page number WITH leading zeros. To create a bates ID, you can add text preceding the tag. For example, Smith vs. Jones%batespgno(00001) would result in Smith vs. Jones000001 being printed on the first page, Smith vs. Jones000002 printed on the second page, etc.

Example: <RightTop string="%Date %Time" editable="true"/>

Result in printed/published output:

7/7/2010 09:02 AM

---

**CONTROLLING FONT SIZE AND STYLE IN MULTI-LINE BANNERS VIA CDATA FORMAT**

This HTML formatting option can be used in the XML file as a banner value, only if the published file you are creating is intended for viewing in the Flash or Silverlight viewers. The HTML banner format is not supported by the Brava ActiveX viewers. Note that this HTML format can only be used in the banner strings and not with screenbanner, screenwatermark, or watermark.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDATA</strong></td>
<td>Format: &lt;![CDATA[format and banner text data as described in the following cells]]&gt; This tag is used to contain the text data properties for the specified banner location.</td>
</tr>
<tr>
<td><strong>&lt;font&gt;&lt;/font&gt;</strong></td>
<td>All text between these tags will have the specified font, size, and color properties. Text will revert to default font, size, and color settings when inserted after the &lt;/font&gt; tag Example: &lt;font face='arial' size='14' color='0,255,0'&gt; banner text &lt;/font&gt;</td>
</tr>
<tr>
<td><strong>face</strong></td>
<td>The truetype font name</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>size</strong></td>
<td>Point size of the font used. For example, a 14 pt font = .20 of an inch on 11&quot; page</td>
</tr>
<tr>
<td><strong>color</strong></td>
<td>Format: r,g,b: range of 0 to 255. Example:  color='255,0,0'</td>
</tr>
<tr>
<td><code>&lt;b&gt;</code></td>
<td>All text between these tags will be bold</td>
</tr>
<tr>
<td><code>&lt;i&gt;</code></td>
<td>All text between these tags will be italic</td>
</tr>
<tr>
<td><code>&lt;u&gt;</code></td>
<td>All text between these tags will be underlined</td>
</tr>
<tr>
<td><code>&lt;br /&gt; or &amp;#10;</code></td>
<td>Line feed (break). Each line in the HTML banner string can contain its own set of properties, allowing you to apply different formatting to several lines of text in the same region.</td>
</tr>
</tbody>
</table>

**Example:**

```html
<TopLeft><![CDATA[<u><font face="Times New Roman" size='18' color='00,255,00'>18 pt Roman Green</font><br /><font face="Arial" size='14' color='255,0,0'>14 pt Arial Red</font><br /><font face="Arial" size='10' color='0,0,255'>10 pt Arial Blue</font>]]></TopLeft>
```

The example above produces the following 3-line banner in the top left corner of the published document.

18 pt Roman Green
14 pt Arial Red
10 pt Arial Blue
integrations and without having to submit "URL" job submissions. Administrative pages are available on the QueueServer to allow quick and easy configuration of directory monitoring.

The installer creates a default Enhanced Directory Monitoring setup at installation time (if you choose to install this feature) which can be used immediately. Workflows and integrations can use this feature to integrate with Net-It Enterprise.

Enhanced Directory Monitoring supports:

- File publishing to PDF, TIFF, secure CSF, and XDL
- Create Thumbnails
- Burn-in annotation, stamp, and watermark/banners
- Apply security settings (for CSF)

Publishing is controlled by the existence of templates, annotation files, and security files that are applied to or created in a task subdirectory. These files can be applied to all files contained in the source directory or via "pairing" by file name association. Process scheduling is optional and users can immediately publish files upon installation.

Enhanced Directory Monitoring setup and administration is done via the Queue Server status pages. End users can create new tasks and easily publish files and specify job options via drag and drop through the familiar Windows file system. No programming knowledge is needed, and folder permissions and security is defined by Windows file system security.

CONCEPTS

**Project** - A Project for Enhanced Directory Monitoring is the top level directory that is monitored by the Queue Server. Associated with the Project directory is:

- The output format of the files that are published
- An optional schedule of when to publish.
- A unique directory per task.

If the schedule is omitted, files are published as they appear or change in each categories Input directory. The schedule is simply a single time of day (like 00:00:00 for midnight, 03:00:00 for 3 in the morning, 17:00:00 for 5 in the evening), local to the queue server, when files in the input directory are published. **Projects names (the final directory in the full path) MUST BE UNIQUE across the entire system.**

Note that files are not placed in the Project's Directory. Rather, this directory merely contains zero or more "Task" directories.

**Tasks** - Tasks are directories in a Project directory where users configure publication options, place source files, and receive outputs. Tasks directories can be created ad-hoc and as users desire. When the Queue Server sees a directory appear in a Project, it then considers that directory a "Task" and creates, under the Task directory, a set of directories in which users manage publication options. All tasks in a given Project obey the same output type and schedule.
ANATOMY OF A TASK

A Task's directory contains three folders: input, output, and job:

End-users can simply drag and drop files into a task's input folder and the file is instantly (or per scheduled time) published with the job options specified in the job folder, and published to the output folder. Whenever a task directory is created, the directory structure is automatically created for the folder. The names of these folders cannot be changed. Default job settings files are provided with the installation that can be edited (for content) by users (stampposition.txt) and again, these filenames cannot be changed.

All files that are added to a task's input folder are published using the exact same settings, unless paired publication is being used (see paired file information). If users want to publish different files with different settings, he can simply create a new task directory and edit the job settings files located in the job folder. Files dropped into the new task directory's input folder will be published with the newly defined settings.

Tasks and/or projects provide very good security boundaries. Administrators should configure Windows file system permissions as appropriate, locking down projects or folders to meet their particular security needs. For example, locking down the Security folder will prevent users from editing banners and watermarks that are used for production of public documents, etc.
**input** - The folder where users place files for publication. It is also a location where users can place "paired" files (see below).

**output** - The folder where the final published output is placed.

**job** - The job folder has several subdirectories. Users place files into these folders to affect publishing (such as burning in markups, applying stamps, and setting other miscellaneous job options). The subfolders of the job folder are:

- **errors** - Contains the logs for any jobs that fail to publish (used by system).
- **logs** - Contains the log files for all successful publish jobs and the heartbeat files for all publish requests (used by system).
- **markups** - Users may place zero or more markup files in this directory to be burned in to the final publication.
- **miscellaneous options** - Users may place zero or more text files in this directory. Each text file is expected to contain `name=value` pairs of additional options to apply at publish time. A sample file is provided in the folder.
- **raster stamp** - Contains a file (called `stampposition.txt`) which can be edited to position a raster stamp during publication. Users place one additional PNG or JPG file into this directory to serve as the raster stamp. If more are added, no stamps are applied.
security - This folder may contain one security.xml file (with any name) that is applied at publication time. A sample file is provided which may be modified and renamed.

stamp template - Like raster stamp above, this directory contains a stampposition.txt file and may contain one stamp template (XSP file designed in Brava! ActiveX or Redact-It Desktop) to be applied at publication time.

working dir - Used by the system for temporary files during job publication.

---

## PAIRED FILES

### PAIRED FILES PRECEDENCE

In addition to setting up stamps, scripts, and templates in the "job" directories, a user can place files with specifically formatted names in the input directory that will then be applied to individual publish jobs. The rules for precedence are as follows:

- A paired file can be one of one or more markup files, a single raster stamp, a raster stamp position, a single stamp template, a stamp template position, and/or a single security file. Miscellaneous job options can also be set in paired files.

- Paired files are files that have names similar to the input file they will apply to, and specific suffixes to indicate their purpose. For example:
  - If a user is publishing a file named airplane.dwg, then he can add a security.xml file to that file by creating a file named airplane.dwg_security.xml in the input directory.
  - If three markups files existed that needed to be applied to the airplane.dwg file during publish time, then those files would be named airplane.dwg_markup_0.xrl, airplane.dwg_markup_1.xrl and airplane.dwg_markup_2.xrl.
  - If only a single markup file needed to be applied to airplane.dwg, then the user could create airplane.dwg Markup.xrl.

- Paired files MUST be created before the target file (airplane.dwg in the case above) is placed in the input directory. Once the system sees airplane.dwg in the input directory, it will publish the job based on the state of the directory at that time, therefore, any paired files that appear after airplane.dwg will not be applied to the publish job.

### PAIRED FILES PRECEDENCE AND COMBINATIONS:

In most cases, job options (such as markups and stamps) that are in the job directory will be combined with the paired files during publication. Thus, if a markup file is paired and placed in the input directory, it will be applied in addition to any markups in the job/markups directory.

The exception to this rule is the security.xml file, the raster stamp file, and the stamp template file. Since only ONE of these files may be applied at publication time, the file in the task's job directory will ALWAYS override any paired files.
PAIRED FILE SUFFIXES

Following is a list of the suffixes that can apply to a file in the input directory. Note that any file that ends with these suffixes will NOT be published when it appears in the input directory, since it is assumed to be a pair to some other file.

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>_markup.xrl</td>
<td>A single markup file to be applied at publish time</td>
</tr>
<tr>
<td>_markup_N.xrl</td>
<td>One of multiple markup files to be applied at publish time, start</td>
</tr>
<tr>
<td></td>
<td>numbering at N=0</td>
</tr>
<tr>
<td>_security.xml</td>
<td>A single security.xml file to be applied at publish time</td>
</tr>
<tr>
<td>_raster_stamp.png</td>
<td>A single raster stamp file to be applied at publish time, PNG format</td>
</tr>
<tr>
<td>_raster_stamp.jpg</td>
<td>A single raster stamp file to be applied at publish time, JPG format</td>
</tr>
<tr>
<td>_raster_stamp_position.txt</td>
<td>A text file describing where to put the raster stamp (see</td>
</tr>
<tr>
<td></td>
<td>application of stamps below)</td>
</tr>
<tr>
<td>_stamp_template.xsp</td>
<td>A single stamp template to apply during publication</td>
</tr>
<tr>
<td>_stamp_template_position.txt</td>
<td>A text file describing where to put the stamp template (see</td>
</tr>
<tr>
<td></td>
<td>application of stamps below)</td>
</tr>
<tr>
<td>_misc_options.txt</td>
<td>A text file of name=value pairs to set miscellaneous job options</td>
</tr>
</tbody>
</table>

BRAVA! DESKTOP MARKUP FILES

Paired publishing supports Brava Desktop style markup files. By default, Brava Desktop saves a markup file for a source file as follows:

- Given a source file, such as `drawing.dwg`, the markup file will be named `drawing_dwg.xrl`.
- Enhanced directory monitoring recognizes this pattern and will publish the file `drawing.dwg` with `drawing_dwg.xrl` if they are both in the input directory at job creation time.
- If more than one markup file is required, the files are named as indicated in the table above.

POSITIONING STAMP TEMPLATES AND RASTER STAMPS

Stamp Templates and Raster Stamps are positioned by placing a text file along side the stamp template or raster stamp file to describe where to put the stamp. This file may contain comment lines beginning with a pound sign (#). It must contain a single line that is the stamp position parameter, as defined in the user's guide. This line is `Left|Top|Scale|Pages`. 
In the case of stamps defined in the job\Stamp template or job\Raster Stamp, the file must live in the appropriate directory and be called StampPosition.txt. A default file is created when the task is created, so it may be simply edited.

In the case of paired templates, the position file must follow the suffix noted above.

---

**SETTING UP AND MANAGING PROJECTS AND TASKS**

**STEPS TO ESTABLISHING AND USING AN ENHANCED DIRECTORY MONITORING:**

**Administrator steps:**

1. Create a shared folder on a file server to hold all of the Projects.
2. Establish the Project (top level file directory monitored on the QueueServer)
3. Set the output format (TIFF, PDF, or CSF)
4. Set the optional time schedule for publishing
5. Set appropriate folder securities

**User steps:**

1. Create a task.
2. The system automatically sets up every Task directory to have three sub-directories (input, output, job directories)
3. A user copies desired templates, markups, misc. options, and security file xml into the appropriate "job" directories.
4. A user copies desired source files into the "input" directory
5. The system creates the resulting published files and saves them to the "output" directory.

---

**ADMINISTRATIVE DETAILS**

Administrators can use the web based management tool, edmsettings.aspx, to create new projects and add tasks to projects. Integrators can use the EDM API (see below) to add projects and tasks to projects.

In typical usage, outlined above, an administrator will create top level projects, and users or administrators will create tasks for projects by simply creating sub-directories in the project directory. Once the Queue Server sees a new task directory under a project, it will populate it with all the required subdirectories.

Tasks may be deleted by simply deleting the directory from the project.

Administrators can stop monitoring a projects using the Web UI (edmsettings.aspx) shown below. If an administrator stops monitoring a project publishing will cease, however, the files and directories are NOT deleted and remain on the file system until deleted manually.
Chapter 3: Usage

EDM SPECIFIC PARAMETERS

Project specific

`enhdirpubtime` - The time, in 24 hour format, when the system will publish all source files in the input directory. If omitted, then publishing happens as soon as a file is changed in the input directory.

Project and Task specific

The parameters below may be configured for a project, or specifically for an individual task. To apply these to a specific task, enter them in a file in the "miscellaneous options" directory.

`enhdirmovesourceonsuccessdir` - A directory name. If specified, the system will attempt to move source files to this directory upon successful job completion, removing the file from the input directory.

`enhdirmovesourceonfailedir` - A directory name. If specified, the system will attempt to move source files to this directory upon job failure, removing the file from the input directory.

`enhdiruniqueoutputdir` - Boolean. If set to "true", the system will create a unique directory under the job’s output directory for each published file. Unique directory names are a combination of the source file name and the time of publication.

EDM API FOR CREATING PROJECTS AND TASKS

`EDMCreateProject.aspx` - This API allows integrators to programmatically add a project to the system. It has the following parameters (specified as query string or headers to the html call).

`projectbasedir` - Required. The full path to the project base directory. This should always be a UNC path.
outputformat - Required. The type of output this project will generate. Either PDF, TIFF, CSF, XDL, or DWF. pdf, tiff, csf, xdl or dwf

createuniqueoutputdir - Optional. If true, all publication in this project will create a unique directory. Required to be set to true if the outputformat is XDL.

publishtime - Optional. If set to a pars-able 24 hour time, create a project that publishes on a schedule. Otherwise, publish as soon as source files appear or change.

**EDMCreateTask.aspx** - This API allows integrators to programmatically add tasks to already created projects. It has the following parameters:

  - **projectname** - Required. The name of the project in which to create the task. (Not the full path, simply the project name.)
  - **taskname** - Required. The name of the task to create in the project.

**EDM SETUP DEFAULT**

The Net-It Enterprise Installer creates default Projects named "Publish to PDF", "Publish to CSF", and "Publish to Tiff". The installer also creates a default Task, called "Default" in each of those directories (enabled by the server.properties setting; enhanced.directory.monitor.create.default).

These projects are set up to publish as soon as a file changes and to NOT publish to a separate directory for each file.
CHAPTER 4: JOB OPTIONS

CHAPTER 4 OVERVIEW

This topic covers information about configuring the Net-It Enterprise optional parameters to publish jobs. The job options covered in this topic detail how to tell the system what you want it to do. They are to Net-It Enterprise what all of the menus, buttons and commands are to a desktop application. Information is presented in a series of tables for quick and convenient referencing.

Integrators use job options to tell the system what to do, like publishing a file, and then what to do with the file once it is published (like output format, where to put the results, etc.).

This topic contains these sections:

- Optional Job Input Parameters
- Job Output Parameters
- Thumbnail Output
- CDL Error Codes
This section describes the various parameters in the `JobProcessor.config` file, located in your installed Net-It Enterprise Job Processor directory. This file controls different variables in the Net-It Enterprise Job Processor software product. The default values are shown in the given examples. Changing any of these values requires a restart of the Job Processor.

### JOB INPUT PARAMETERS

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
<td>The complete path to the source file to publish. The Job Processor must have access to this file. Not valid for directory monitoring job. Please note that only <strong>network paths</strong> are supported for <code>Source</code> parameter path values. Local paths (but not mapped drives) can be used for single box testing only.</td>
</tr>
</tbody>
</table>
| **SourceN**     | With multi-doc publishing, you can combine multiple source documents to publish as one exported file. To do so, set multiple source file paths using `SourceN` instead of `Source` (you cannot specify both). The various source files do not have to be of the same format and will all publish to the format specified by `OutputFormat`. For example:  
  
  source0=\computer\share\Airplane.dwg  
  source1=\computer\share\AirplaneSpecs.doc  
  source2=\computer\share\AirplaneBrochure.pdf  
  
  **Notes:**  
  - CSF files are not supported as input types  
  - XDL files are supported as input types  
  - TIFF Manipulation does not support the multi-doc feature |
| **Target**      | The target directory, where the published artifacts will be placed. With directory monitoring, published output is specified in the directory monitoring job. Please note that only **network paths** are supported for `Target` parameter path values. Local paths (but not mapped drives) can be used for single box testing only. |
| **OutputFormat**| The format of the output file. Valid values are either XDL, CSF, PDF, DWF, TIFF, or NONE. When publishing to TIFF, or when markups are included in the publishing request, it is highly recommended that a page size is set through the `OutputPageSize` parameter to avoid huge file sizes. If set to NONE, there will be no export of the published file, but other... |
Chapter 4: Job Options

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotificationUrl</td>
<td>The URL to call when the job is completed.</td>
</tr>
<tr>
<td>NotificationVerb</td>
<td>Http method to use for making calls to the Job Processor. Valid values are GET, POST, or MULTIPARTPOST. The default value is GET. If notificationverb is set, then the Job Processor will call notificationurl with the set HTTP verb.</td>
</tr>
<tr>
<td>MarkupFilename</td>
<td>A single file name (XRL) that contains an IGC formatted markup to be burned into the published file. It must be appropriate to the file specified in source. This parameter does nothing if the output format is XDL.</td>
</tr>
<tr>
<td>MarkupListFilename</td>
<td>A text file containing a list of markup files, one per line.</td>
</tr>
<tr>
<td>ExportMarkupFilename</td>
<td>Specify the name of a file that will receive all of the markups created from markup files. If you choose this option, the resulting published file WILL NOT have markups burned in. Rather, they will be placed in the file for review. The markup file applies to either the published XDL or CSF rendition of the file, or to the original file when opened via IGC viewing technology.</td>
</tr>
<tr>
<td>SecurityXmlFilename</td>
<td>Security settings for publication including published banners and watermarks. See Security XML Details.</td>
</tr>
<tr>
<td>ExportChangemarkSummary</td>
<td>If set to true, a Changemarks summary page is published along with the output PDF file.</td>
</tr>
<tr>
<td>PdfExportMarkupsAsAnnotations</td>
<td>If set to true, markups (see exceptions detailed below) on PDF output files are saved as PDF Annotations. If false (the default), markups are burned into the PDF output and cannot be removed by users. <strong>Changemarks Details:</strong> Regardless of the state of this flag, IGC Changemarks Entities are ALWAYS annotations in PDF output files and IGC Redactup Entities are ALWAYS burned in and cannot be edited.</td>
</tr>
<tr>
<td>FitWithinBanners</td>
<td>If set to true, the printed documents pages are scaled to fit within any banners that might be set.</td>
</tr>
</tbody>
</table>
| CSFName                       | Use this parameter to specify the output filename with extension. If specified, the full path to the output file is a concatenation of the Target and CSFName parameters. If omitted, the output file name is the source file name with a new extension based on output format. You must specify the file extension when using this parameter, and it must be the same as OutputFormat.  
Example:  
CSFName=document.pdf |
| PublishPageList               | A list of page numbers to publish. Only the specified pages will be published into the final format. This can be a list of single pages (separated by the pipe character, |) or a set of page ranges (also
### HeartBeat

The value of this parameter is the full path to the file name. The path is not automatically created, and if not specified, heartbeat is disabled. When specified on a job, the Job Processor periodically writes to the specified file at points during the publish process. The file can be a unique file for each job (recommended - a file in the Target directory is a good choice here). Multiple jobs can use the same file, but they may incur performance penalties when trying to write to the same file simultaneously.

The file is encoded UTF-8, and the Job Processor must have access to the file. UNC paths are recommended.

Each line in the file is formatted as:

```
[<Date Time>] [<Source File Name>] Msg
```

An example line:

```
[03/30/09 03:15:41] [1ST FLOOR PLAN.DWG] Starting Job
```

Following is the list of possible messaging information that may appear in a job:

- Starting Job
- Auto recognizing File <<source file>>
- Loading file <<source file full path>>
- Applying markup <<markup file name>>
- Applying Banners and Watermarks
- Exporting as <<export type>>
- Generating text dump
- Publishing Complete
- Saving markups as file
- Generating thumbnails
- Job has an error
- Copying results
- Notifying <<URL>>
- Job Done
Chapter 4: Job Options

- Notification Response <<response from notification URL>>
- Final job parameters <<key pairs, one per line>>
- Critical error in job <<error and stack information>>

### StepTimeout

The maximum number of seconds allowed for any one step of the conversion process. It corresponds to the maximum amount of time allowed between heartbeat messages. The heartbeat does NOT need to be enabled in order to use this parameter. The default value is 600 seconds. This value may need to be increased if you are publishing very large documents via print-publishing. The default may be set for all jobs via the `jpconsole.exe.config` file and `jpservice.exe.config` file, via the `DefaultStepTimeoutInSeconds` value. The value of `steptimeout` in each job always overrides the value in the config file.

### StampImage

The Stamp feature allows users to specify a JPG or PNG image to apply as a stamp to all or certain pages of a 2D file when converted. This parameter contains the Job Processor accessible path to the image file to stamp. It must be a UNC path to a JPG or PNG file.

**Example:**

```
StampImage=\\mymachine\sampleimages\completed.png
```

### StampPositionAndSize

This parameter is used in conjunction with `StampImage` and contains multiple values, separated by the pipe character (|), that specify the size and position of the image file (stamp) specified by the value of `StampImage`, and on which pages it will be applied.

The string parameter format is:

```
StampPositionAndSize=left|top|scale|page[|page][|page]...
```

- **Left** – position of the image, as a percentage of page width
- **Top** – position of the image, as a percentage of page height
- **Scale** – size of the image, as a percentage of page width. This means that the image is resized to take up scale percentage of scale width. -1 can be specified to tell the system not to scale at all.
- **Page** – page on which to apply the stamp. It can be -1 to apply to all pages, or a sequence of page numbers, pipe character (|) separated and one-based on which to apply the stamp. To apply the stamp to only the first page, set this value to 1.

**Example:**

```
Source=\\mymachine\files\DWG\layout.dwg
target=C:temp\PublishedOutput
Outputformat=pdf
StampImage=\\mymachine\sampleimages\completed.png
StampPositionAndSize=10|10|-1|-1
```
| **StampTemplateFileName** | The Stamp Template feature allows users to specify a Stamp Template (created in IGC’s Brava Desktop) to apply to all or some pages of a 2D file when converted. This parameter contains the Job Processor accessible path to the Stamp Template file. It must be a UNC path to a Stamp Template file (XSP).

**Example:**
StampTemplateFileName=\\mymachine\\samplefiles\\mystamp.xsp |

| **StampTemplatePositionAndSize** | This parameter, used in conjunction with **StampTemplateFileName**, contains multiple values, separated by the pipe character (|), that specify the size and position of the stamp template specified by the value of **StampTemplateFileName**, and on which page(s) it will be applied.

The string parameter format is:
StampTemplatePositionAndSize=left|top|scale|page[|page ][|page]...[|page]

**Left** – position of the image, as a percentage of page width

**Top** – position of the image, as a percentage of page height

**Scale** – size of the image, as a percentage of page width. This means that the image is resized to take up scale percentage of scale width. -1 can be specified to tell the system not to scale at all.

**Page** – page on which to apply the stamp. It can be -1 to apply to all pages, or a sequence of page numbers, pipe character (|) separated and one-based on which to apply the stamp. To apply the stamp to only the first page, set this value to 1.

**Example:**
Source=\\mymachine\\files\\DWG\\layout.dwg
target=C:\\temp\\PublishedOutput
Outputformat=pdf
StampTemplateFileName=\\mymachine\\samplefiles\\mystamp.xsp

**StampTemplatePositionAndSize=10|10|--1|--1**

notificationurl=http://mymachine:9999

This example would apply the template mystamp.xsp to all pages of the published PDF document, starting 10% of the width from the left of the page and 10% of the page height down from the top of the page. On each page, the image would not be scaled. This feature is useful for
applying stamp templates to a large volume of documents of the same type and size (for example, 8.5 x 11 PDF files, or same-size CAD files)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TiffBpp</strong></td>
<td>You can specify color depth for TIFF output with this parameter. This parameter, along with setting TIFF dpi, gives control over output file size. Valid values are 2, 4, 8, 24, or 32. If omitted, the system will default to 8 bpp. If omitted, AND the input contains rasters, then the system may switch to 24 bpp if needed for fidelity.</td>
</tr>
<tr>
<td><strong>TiffDpi</strong></td>
<td>For use when the output type of a conversion is TIFF only. This sets the DPI in the output TIFF file. The default is 300 DPI. If the input type is also a TIFF and this parameter is not set, then the output DPI is set to the maximum DPI of any page in the source file. If this parameter is set, it overrides any source file DPI settings.</td>
</tr>
<tr>
<td><strong>TiffCompressionType</strong></td>
<td>Determines the TIFF compression algorithm type to use in the output TIFF file. Valid values are LZW, JPG, CCITT, or PACKBITS. Note that CCITT is also known as &quot;G4&quot;. The default value is LZW.</td>
</tr>
<tr>
<td><strong>GetText</strong></td>
<td>When GetText is set to TRUE, instructs the server to produce a set of text files with all of the text from the published document extracted as plain text. This parameter is useful for indexing. If the publishpagelist value is specified, both values are honored and only pages listed in publishpagelist will have their text extracted. Default value is false.</td>
</tr>
<tr>
<td><strong>TextFormatUTF8</strong></td>
<td>If set to true, then the output format produced from GetText is UTF-8 encoded. If omitted or set to false, the output format is UTF-16 encoded.</td>
</tr>
<tr>
<td><strong>txtformatcrlf</strong></td>
<td>By default, text files extracted by the Job Processor (when gettext=true) have their lines delimited by a single line feed (Unix style). If you wish to have each line separated by a carriage return/line feed pair (DOS/Windows style), you can instruct the Job Processor to do so by adding the parameter textformatcrlf=true to the job.</td>
</tr>
<tr>
<td><strong>DocumentTitle</strong></td>
<td>Specifies the title of the document, used when the %title macro is specified for an ISOBanner in security.xml.</td>
</tr>
<tr>
<td><strong>AddMetadataToTextOutput</strong></td>
<td>If both this and the GetText parameter are set to True, then any metadata present in the source file will be included in the text output pages. The values are inserted in the first page's text output and are only available from DWG Files. Metadata includes, but is not limited to: Date created Author of file Date modified Last updated Title Object</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ExportFilePerPage</td>
<td>Net-It Enterprise can export a single file per each page in the source document if this parameter is set to true. These pages will be named FILENAME_PAGE_NN.EXPORTTYPE. This will be a feature available for PDF and TIFF Export ONLY. For example, if you have an N page document, called building.pdf, then the output will be building_page0.pdf, building_page1.pdf, ... building_pageN-1.pdf, etc. When exporting to single file per page, the total page count macro, the Bates number, and the page number macro will increment with each file. Therefore, file one will be page 1, file two page 2, etc. Total pages will be set based on total pages in the export request (which equals all pages in the source file unless you are only exporting selected pages). For example: When publishing pages 2 and 3 of a six-page PDF file, one file per page, the total page count will be 2 at export time.</td>
</tr>
<tr>
<td>ExportPageExtents</td>
<td>When ExportPageExtents is set to true, a file's page extents will be exported from the Job Processor as a separate file. The system will place a file, specified by ExportPageExtentsFile in the target directory containing the page extents in IGC units (floating point) suitable for conversion of markup files. Example: #PageNum,Height,Width (format key) 0, 10750.000000, 8250.000000 1, 10750.000000, 8250.000000 2, 10750.000000, 8250.000000 Note that if the PublishPageList parameter is specified, then only the extents for those pages will be published into the file and available for export.</td>
</tr>
<tr>
<td>ExportPageExtentsFile</td>
<td></td>
</tr>
<tr>
<td>PublishUserName</td>
<td>Specify the user name who is publishing a file, used when the %user or %login macro is specified for an ISOBanner in security.xml.</td>
</tr>
<tr>
<td>PdfExportA1b</td>
<td>If set to true, then the PDF output will be PDF/A-1b archive compatible. It will likely be a larger file if it contains many rasters or transparencies.</td>
</tr>
<tr>
<td>ForceTiffMonochrome</td>
<td>If set to true, all markups, vectors, and rasters are rendered monochrome in the output TIFF file. If true, this parameter value overrides the value set by OutputColorMode</td>
</tr>
<tr>
<td>OutputColorMode</td>
<td>When set, all content in the document (rasters, vectors, and markups) are set to the chosen color mode. This parameter does not affect banner color, which can be set through the security.xml file as appropriate. Valid values are as follows: fullcolor - all colors for all entities are exported as seen in the source document(s) and markups. grayscale (or greyscale) - all colors for text, rasters, vectors, and</td>
</tr>
</tbody>
</table>
**Chapter 4: Job Options**

Markups are rendered in 256 shades of grey.

**Monochrome** - all text, rasters, vectors, and markups are rendered in 2-bits, either black or white. Raster images (such as embedded JPG), appear solid black.

**Notes:**
- If the output format is TIFF, and the color mode is set to *monochrome*, then the output compression type will be CCITT Group 4 (FAX).
- CSF output only supports *fullcolor* mode. A warning message displayed if set to any other value when publishing as CSF.

<table>
<thead>
<tr>
<th><strong>ExportLayerState</strong></th>
<th>This parameter determines the layer state for exporting all document types. Valid values are:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>Export all layers to PDF, DWF, CSF, or XDL (or visible layers flattened for TIFF)</td>
</tr>
<tr>
<td><strong>Visible</strong></td>
<td>Export all visible layers only to PDF, DWF, CSF, or XDL (or flattened for TIFF)</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>Export only visible layers, flattened (or as a single layer in DWF)</td>
</tr>
</tbody>
</table>

If the original document does not contain layers, this option has no effect and a warning message displays. TIFF documents do not support layers and are recognized as single layer visible or flattened documents. When exporting any format to TIFF, you only get the visible layers flattened as output, regardless of this option setting.

| **OutputPageSize** | You can set the page size by specifying the *outputpagesize* for output of type PDF, TIFF and DWF. It is highly recommended to set this parameter when *OutputFormat* is set to TIFF or if markups are included in the publishing request to avoid creating a huge output file size. Refer to the [table of defined sizes](#). |

**Performance recommendation** - when publishing CAD files (and other file types with very large page sizes) to TIFF, the output file can become quite large, since the page dimensions can be very large. Therefore, if you are publishing CAD files and are concerned with file size, IGC strongly recommends that you set the page size to control file size.

| **CreatePageSizesTextFile** | For use with XDL output only, this parameter allows users to create a *Pagesizes.txt* file. If set to true, then *Pagesizes.txt* will be created in the file’s output directory. |

<p>| <strong>CreatePageSizesTextMetric</strong> | For use with XDL output only, this parameter determines the type of measurement information displayed in the <em>Pagesizes.txt</em> file (when <em>createpagesizestextfile</em> is set to true). If this parameter is set to true, then the system will output metric page size information. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timeout</strong></td>
<td>The number of seconds to allow a job to process. The default value is 1200 (20 minutes). If a job takes longer than this set time to publish, processing is terminated and a time out error is returned.</td>
</tr>
<tr>
<td><strong>ShowLineWeights</strong></td>
<td>When set to true, AutoCAD Print Line Weights are supported when publishing to PDF. The default is false.</td>
</tr>
<tr>
<td><strong>ShowThinLinesOnly</strong></td>
<td>Set to true to show only thin lines. Default is false.</td>
</tr>
<tr>
<td><strong>Priority</strong></td>
<td>Priority indicates the order that jobs are given to a requested Job Processor thread. This optional parameter is a string representation of a 32-bit signed integer. Jobs are sent to individual queues by their extension (DRW, DOC, PDF), and within those queues, they are ordered by the integer value set by Priority. A higher number receives higher priority and equivalent numbers will be inserted in order of first in, first out. Jobs that do not have a priority value set will be processed after all jobs in the queue that have a priority value are processed. The range of properties is any integer from -1000 to 1000, inclusive. If the range is out of bounds of -1000 to 1000, it will be adjusted to the closest value. If the range is invalid (cannot be parsed as an integer), then the priority will be ignored. In both cases, the server adds a warning to the job. So, for example, if 5 jobs are inserted into the DRW queue, using the following calls in the following order: push.aspx?name=job1,...,priority=10,... push.aspx?name=job2,..., push.aspx?name=job3,...,priority=0,... push.aspx?name=job4,...,priority=10,... push.aspx?name=job5,...,priority=-5,... Then the jobs will be served up to the JPs in the following order: Job1, Job4, Job3, Job5, Job2 Note that jobs may not finish in that order if multiple Job Processors are running simultaneously.</td>
</tr>
<tr>
<td><strong>ctbfile</strong></td>
<td>Full path to the custom CTB file to use when publishing autocad files. To enable this feature, you must also edit the Job Processor's mydrv.ini file. In the [DWG2DL] section of the file, add or modify the line: UsePlotStyleColors=true Driver file DWG2DL.dll version 1.5.9.1 or higher is required to use this feature.</td>
</tr>
<tr>
<td><strong>intloutputstringfile</strong></td>
<td>This parameter allows each job to override the strings used by the Job Processor during markup and redaction burn-in and export. The default</td>
</tr>
</tbody>
</table>
values for these strings are:
MARKUP_GHOSTTEXT="<Text>"
CMPRINT_PAGE = "Page "
CMPRINT_CMS = " - Changemarks ( "
CMPRINT_NOTES = " Note )"
CMPRINT_RD = " - Redaction ( "
CMPRINT_CHANGEMARK_REPORT_FONT = "Arial"
CMCOPY_BOOKMARK_PAGE_TITLE = "List of Comments"
CMCOPY_BOOKMARK_PAGE_TITLE_LINK_ID = "List"
CMCOPY_BOOKMARK_PAGE_TITLE_LINK_TEXT = "(Back to List)"
CMCOPY_COMMENT_BY = "Comment by "
CMCOPY_ON = "on "
CMCOPY_AT = "at "
CMCOPY_SEE_ALSO = "See also: "
REDACTED_MSG = "(Redacted)"
CMPRINT_INSTANCE = "instance"
CMPRINT_INSTANCES = "instances"
NO_REDACTION_REASONS = "<<Unspecified redaction code>>"
NO_REDACTIONS = "No redactions applied"
NO_CHANGEMARKS = "No changemarks applied"

You can override these parameters by creating a text file (using UTF-16 Little Endian byte order). Each line in the file can contain a definition as shown above. Each line must be one of the keys above (like MARKUP_GHOSTTEXT, CMCOPY_AT, etc) followed by an equal sign, and then the string to use. The string to use may not contain an equal sign. The file's first two bytes must contain a UTF-16LE byte order mark (0xFFFE).

A sample file is provided in the installed Job Processor directory (called sample_intloutputstrings.txt,) that can be modified or referenced.

These parameters control **Thumbnail publishing** options:

<table>
<thead>
<tr>
<th>Input Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThumbFormat</td>
<td>Specifies the format to use for exported thumbnails. Valid values are <strong>JPG</strong>, <strong>PNG</strong>, or <strong>smallest</strong>.</td>
</tr>
<tr>
<td><strong>ThumbName</strong></td>
<td>The name of the thumbnail file, without an extension (thumbnail, for example). The &quot;jpg&quot; or &quot;png&quot; will be appended to the file name. Thumbnails are placed in the directory specified by the target parameter.</td>
</tr>
</tbody>
</table>
| **ThumbSizes** | The size of the thumbnail (in pixels) to create, as x,y.  
For example: thumbsizes=300,200  
More than one size thumbnail can be exported in a given print job. For example:  
thumbsizes= W1,H1,W2,H2,W3,H3,…,Wn,Hn  
Results in n thumbnails per page requested.  
If ThumbSizes is set, ThumbWidths and ThumbHeights parameters are ignored. The precedence for thumb sizes is ThumbSizes, ThumbWidths, ThumbHeights. |
| **ThumbWidths** | Allows thumbnail requests by width or height only. When set, the system calculates the other dimension for each page being published. Both of these parameters can be a list of values. If specified as a list, the number of values entered will display per page. For example,  
ThumbWidths=100,300,1000  
Results in three thumbnails being created for each page. One at 200 pixels wide, a second at 300 pixels, and the third at 1000 pixels. Height is determined by constrained proportion.  
You can only specify either heights or widths. If both are specified, then heights is ignored (according to the precedence stated in ThumbSizes above). |
| **ThumbPages** | The pages to create thumbnails for. This is only valid for 2D output.  
Valid options are:  
A - creates a thumbnail for all the pages in the document  
F - creates a thumbnail for the first page only  
&page number> - creates a thumbnail for the specified page. Only
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThumbQuality</td>
<td>The quality of the thumbnail JPEG or PNG image. Enter a number from 1 (lowest quality) to 100 (highest quality).</td>
</tr>
<tr>
<td>JpegName</td>
<td>The name of the thumbnail file, without an extension (thumbnail, for example). The &quot;.jpg&quot; will be appended to the file name. Thumbnails are placed in the directory specified by the target parameter. If set, this parameter takes precedence over ThumbName.</td>
</tr>
<tr>
<td>JpegSizes</td>
<td>The size of the thumbnail (in pixels) to create, as x,y. For example: jpegsizes=300,200. More than one size thumbnail can be exported in a given print job. For example: jpegsizes= W1,H1,W2,H2,W3,H3,…,Wn,Hn. Results in n thumbnails per page requested. If JpegSizes is set, JpegWidths and JpegHeights parameters are ignored. The precedence for JPEG sizes is JpegSizes, JpegWidths, JpegHeights. This parameter, if set, takes precedence over ThumbSizes.</td>
</tr>
<tr>
<td>JpegWidths</td>
<td>Allows thumbnail requests by width or height only. When set, the system calculates the other dimension for each page being published. Both of these parameters can be a list of values. If specified as a list, the number of values entered will display per page. For example, JpegWidths=100,300,1000. Results in three thumbnails being created for each page. One at 200 pixels wide, a second at 300 pixels, and the third at 1000 pixels. Height is determined by constrained proportion. You can only specify either heights or widths. If both are specified, then heights is ignored (according to the precedence stated in JpegSizes above). These parameters, if set, take precedence over ThumbWidths and ThumbHeights.</td>
</tr>
<tr>
<td>JpegHeights</td>
<td></td>
</tr>
<tr>
<td>JpegPages</td>
<td>The pages to create thumbnails for. This is only valid for 2D output. Valid options are: A - creates a thumbnail for all the pages in the document F - creates a thumbnail for the first page only &lt;page number&gt; - creates a thumbnail for the specified page. Only one page number can be specified.</td>
</tr>
</tbody>
</table>
If you specify A, Net-It Enterprise will append the page number to the published JPEG name. Page numbers start at 0.

This parameter, if set, takes precedence over ThumPages.

<table>
<thead>
<tr>
<th>JpegQuality</th>
<th>The quality of the thumbnail JPEG. Enter a number from 1 (lowest quality) to 100 (highest quality).</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This parameter, if set, takes precedence over ThumbQuality.</td>
</tr>
</tbody>
</table>

You can send additional input options (set as name=value pairs in the query string of the push) to the QueueServer and they will be echoed, verbatim, by the Job Processor when it notifies you of the completed job. With this option, you can set additional information in the job to track progress, store results, and coordinate sent and received jobs. The only requirement is that the name of the parameter does not conflict with the names listed in the previous table, or with the list of reserved parameter names shown here:

| igc_* | mainfile | tempdir |
| jobid | targetfile | endtime |
| type | filename | starttime |
| threaded | ext | totaltime |

**JOB OUTPUT PARAMETERS**

The output parameters listed in this section are returned from Net-It Enterprise after a job is complete. The output parameters let you deal with the results of the job, including errors.

<table>
<thead>
<tr>
<th>Output Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mainfile</td>
<td>The main output file. This parameter will be present only if publishing was successful. It will be the first file name for a multi-file output, or the name of the single file for the output of a single file (like PDF). If you choose to publish each page of a document to a separate file, this will be the name of the first file.</td>
</tr>
<tr>
<td>publishedpagecount</td>
<td>Returns the number of pages successfully published, if publish succeeds.</td>
</tr>
<tr>
<td>sourcepagecount</td>
<td>Returns the total number of pages in the source file.</td>
</tr>
</tbody>
</table>
### Warning

Any warnings that occur during the conversion are presented here.

### Error

Any errors during conversion are detailed here. See the CDL Errors table in the next section for code descriptions.

#### error 0,1, ... N

*warning 0, 1...N*  
Additional information, as required, relating to errors or warnings presented in the "Error" parameter.  
Example: *error 1= mismatched widgets*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cdlversion</td>
<td>The version of CDL (internal IGC information) used during conversion.</td>
</tr>
<tr>
<td>starttime</td>
<td>The time the job started getting processed by the Job Processor.</td>
</tr>
<tr>
<td>endtime</td>
<td>The time the job finished processing by the Job Processor.</td>
</tr>
<tr>
<td>publishernname</td>
<td>The Job Processor that published the job, along with the Job Processor version.</td>
</tr>
</tbody>
</table>

### THUMBNAIL OUTPUT

If you request thumbnail files during publishing, they are placed in the directory that you specify via the target parameter.

Note that if any Visual Right is set on a published CSF file, including expiration dates, the file's thumbnail image will display with a default CSF icon and not a viewable thumbnail image of the document.

When you publish files and request one or more thumbnails, the thumbnail file names that the system assigns to the thumbnail files are formatted as follows:

```plaintext
<jpegname>_XxY_page_N.jpg
```

Where:

- `<jpegname>` is the value you specify in the `jpegname` parameter
- `X` is the X dimension you specify in the `jpegsizes` parameter
- `Y` is the Y dimension you specify in the `jpegsizes` parameter
CDL ERROR CODES

Certain conversion errors and warnings include numerical codes to describe the error, and are reported by the Job Processor as either:

Critical CDL Error 0x####
CDL Warning 0x####

The number codes and descriptions are listed here:

<table>
<thead>
<tr>
<th>Code (decimal value)</th>
<th>Code (Hex value)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0x0000</td>
<td>Success</td>
</tr>
<tr>
<td>27</td>
<td>0x001B</td>
<td>Can't find support file</td>
</tr>
<tr>
<td>28</td>
<td>0x001C</td>
<td>Can't find Xref</td>
</tr>
<tr>
<td>32769</td>
<td>0x8001</td>
<td>Out of memory</td>
</tr>
<tr>
<td>32771</td>
<td>0x8003</td>
<td>Out of stack</td>
</tr>
<tr>
<td>32772</td>
<td>0x8004</td>
<td>Can't write output</td>
</tr>
<tr>
<td>32773</td>
<td>0x8005</td>
<td>Can't open file</td>
</tr>
<tr>
<td>32774</td>
<td>0x8006</td>
<td>Can't find file</td>
</tr>
<tr>
<td>32775</td>
<td>0x8007</td>
<td>Can't create file</td>
</tr>
<tr>
<td>32776</td>
<td>0x8008</td>
<td>Can't read file</td>
</tr>
<tr>
<td>32777</td>
<td>0x8009</td>
<td>Can't write file</td>
</tr>
<tr>
<td>32778</td>
<td>0x800A</td>
<td>Corrupt file</td>
</tr>
<tr>
<td>32779</td>
<td>0x800B</td>
<td>Unsupported format</td>
</tr>
<tr>
<td>32780</td>
<td>0x800C</td>
<td>Unlicensed format</td>
</tr>
<tr>
<td>32781</td>
<td>0x800D</td>
<td>Can't create image</td>
</tr>
<tr>
<td>Error Code</td>
<td>Error Code Value</td>
<td>Error Description</td>
</tr>
<tr>
<td>------------</td>
<td>------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>32782</td>
<td>0x800E</td>
<td>Can’t insert entity</td>
</tr>
<tr>
<td>32783</td>
<td>0x800F</td>
<td>Can’t register layer</td>
</tr>
<tr>
<td>32784</td>
<td>0x8010</td>
<td>Can’t register linestyle</td>
</tr>
<tr>
<td>32785</td>
<td>0x8011</td>
<td>Invalid limits</td>
</tr>
<tr>
<td>32786</td>
<td>0x8012</td>
<td>Invalid page</td>
</tr>
<tr>
<td>32787</td>
<td>0x8013</td>
<td>Invalid version</td>
</tr>
<tr>
<td>32788</td>
<td>0x8014</td>
<td>Invalid layer</td>
</tr>
<tr>
<td>32789</td>
<td>0x8015</td>
<td>Invalid entity</td>
</tr>
<tr>
<td>32790</td>
<td>0x8016</td>
<td>Application callback error</td>
</tr>
<tr>
<td>32791</td>
<td>0x8017</td>
<td>3rd party library error</td>
</tr>
<tr>
<td>32792</td>
<td>0x8018</td>
<td>Exception thrown</td>
</tr>
<tr>
<td>32793</td>
<td>0x8019</td>
<td>No 2D entities</td>
</tr>
<tr>
<td>32794</td>
<td>0x801A</td>
<td>Can’t load loader</td>
</tr>
<tr>
<td>32797</td>
<td>0x801D</td>
<td>Previous loader is still active</td>
</tr>
<tr>
<td>32798</td>
<td>0x801E</td>
<td>Page is out of range</td>
</tr>
<tr>
<td>32799</td>
<td>0x801F</td>
<td>No document open</td>
</tr>
<tr>
<td>32800</td>
<td>0x8020</td>
<td>Function not implemented</td>
</tr>
<tr>
<td>32801</td>
<td>0x8021</td>
<td>Output directory is not set</td>
</tr>
<tr>
<td>32802</td>
<td>0x8022</td>
<td>Document summary allocation</td>
</tr>
<tr>
<td>32803</td>
<td>0x8023</td>
<td>DLGenerator allocation</td>
</tr>
<tr>
<td>32804</td>
<td>0x8024</td>
<td>Semaphore allocation</td>
</tr>
<tr>
<td>32805</td>
<td>0x8025</td>
<td>Total pages not known</td>
</tr>
<tr>
<td>32806</td>
<td>0x8026</td>
<td>Thread failure</td>
</tr>
<tr>
<td>32807</td>
<td>0x8027</td>
<td>File access error</td>
</tr>
<tr>
<td>32808</td>
<td>0x8028</td>
<td>Loader error</td>
</tr>
<tr>
<td>32809</td>
<td>0x8029</td>
<td>Can’t load DLGenerator</td>
</tr>
<tr>
<td>32810</td>
<td>0x802A</td>
<td>Invalid argument</td>
</tr>
<tr>
<td>Code</td>
<td>Value</td>
<td>Message</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>32811</td>
<td>0x802B</td>
<td>AutoRec failed</td>
</tr>
<tr>
<td>32812</td>
<td>0x802C</td>
<td>Path is too long</td>
</tr>
<tr>
<td>32818</td>
<td>0x8032</td>
<td>Can't load page</td>
</tr>
<tr>
<td>32819</td>
<td>0x8033</td>
<td>Can't open document</td>
</tr>
<tr>
<td>32820</td>
<td>0x8034</td>
<td>Software error</td>
</tr>
<tr>
<td>32828</td>
<td>0x803C</td>
<td>CSF conversion failed</td>
</tr>
<tr>
<td>32829</td>
<td>0x803D</td>
<td>CSF parameters are missing</td>
</tr>
<tr>
<td>32831</td>
<td>0x803F</td>
<td>CSF invalid password</td>
</tr>
<tr>
<td>32835</td>
<td>0x8043</td>
<td>User cancel</td>
</tr>
<tr>
<td>32836</td>
<td>0x8044</td>
<td>Dot NET 3.0 is not installed</td>
</tr>
<tr>
<td>32837</td>
<td>0x8045</td>
<td>Native file is protected</td>
</tr>
<tr>
<td>32838</td>
<td>0x8046</td>
<td>Loader not licensed</td>
</tr>
<tr>
<td>62</td>
<td>0x003E</td>
<td>CSF found 3D entities</td>
</tr>
<tr>
<td>64</td>
<td>0x0040</td>
<td>File contains 2D entities</td>
</tr>
<tr>
<td>65</td>
<td>0x0041</td>
<td>File contains 3D entities</td>
</tr>
<tr>
<td>66</td>
<td>0x0042</td>
<td>File contains 2D and 3D entities</td>
</tr>
<tr>
<td>71</td>
<td>0x0047</td>
<td>BiPrint NIN license</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently not used for server products. Free NIN will prompt a separate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>error message.</td>
</tr>
<tr>
<td>72</td>
<td>0x0048</td>
<td>BiPrint unspecified parameters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Currently not in use</td>
</tr>
<tr>
<td>73</td>
<td>0x0049</td>
<td>BiPrint module location not found</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The location of BiPrint.dll is expected to be found in the registry path</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specified by &quot;HKEY_LOCAL_MACHINES\Software\IGC\Net-It Now&lt;version&gt;&lt;Install Directory&gt;</td>
</tr>
<tr>
<td>74</td>
<td>0x004A</td>
<td>BiPrint unknown extension</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When a source document has an invalid extension (dll, exe, ncp, zip, mcf),</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or if set to print using DDE, the file extension cannot be looked up in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the Registry</td>
</tr>
<tr>
<td>75</td>
<td>0x004B</td>
<td>BiPrint drag and drop print</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A print/printto action is not defined for this file format</td>
</tr>
<tr>
<td>Code</td>
<td>Error Description</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td></td>
</tr>
</tbody>
</table>
| 0x004C | BiPrint printer not installed  
The IGC Writer is not found and needs to be installed |
| 0x004D | BiPrint DDE conversation  
User has canceled DDE conversion |
| 0x004E | BiPrint DDE connect  
An error has occurred in DDE connection |
| 0x004F | BiPrint DDE conversion  
Generic error in CCE conversion |
| 0x0050 | BiPrint perform shell execute  
This is a specific error in DDE conversion when using DDE printing. Some applications, such as Office documents, use DDE to do print/printto actions. File formats, such as txt or HTML, use the command line to print (not using DDE) |
| 0x0051 | BiPrint reg application  
An error occurred finding the registered application to print for DDE conversion |
| 0x0052 | BiPrint reg command  
An error occurred finding the registered command for DDE conversion |
| 0x0053 | BiPrint main exception error  
As with Office automation errors, this thread execution error occurs when there is a problem writing into NINFile.xml. The NINFile.xml is a temporary file created to communicate the source document information to Net-It Now, launched after printing. Net-It Now updates this file with GRP and EMF files and BiPrint returns it to BI2DL. |
| 0x0054 | BiPrint timeout  
This timeout error displays when any of the following actions exceed their maximum time allowance: printing, exporting to XPS, SaveAs message to Doc file, or waiting on NINFile.xml to be updated with all GRP and EMF files. |
| 0x0055 | BiPrint NIN file read - error reading NINFile.xml |
| 0x0056 | BiPrint matching source document  
Unable to match the source document name with the job title in the printed GRP file. |
| 0x0057 | BiPrint attachment enabled  
This error occurs if the "Print attached Files. Attachments will print to the default printer only" option is enabled in the Outlook Print dialog. This |
<table>
<thead>
<tr>
<th>Option</th>
<th>Error Code</th>
<th>Description</th>
</tr>
</thead>
</table>
| 88     | 0x0058     | BiPrint create mutex failed  
Mutex is used to implement mutual exclusion so that no more than one process is accessing a common resource at the same time. |
| 89     | 0x0059     | BiPrint office automation printing failed |
| 90     | 0x005A     | BiPrint abort secondary document processing (Outlook attachments)  
Used for Outlook messages that have attachments and determines whether the main document processing should continue or not based on the secondary document processing. When the IgnoreSecondaryDocumentError setting in BIPrint.ini is set to NO, this error is returned on the main document for any processing error that occurs on attachments. |
| 91     | 0x005B     | BiPrint temp folder not found  
Unable to find the system TEMP folder (typically %SystemRoot%\temp). The system TEMP folder path is read from the registry entry: "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Session Manager\Environment\TEMP" and if ControlSet001 is missing from the registry and any key in the path is missing, this error is returned. |
| 92     | 0x005C     | BiPrint temp folder write failed  
Unable to write to the system TEMP folder |
| 93     | 0x005D     | BiPrint Outlook item contains delegates  
Unable to process Outlook message/meeting request because it contains delegates (meeting request is sent on behalf of someone). |
| 94     | 0x005E     | BiPrint Office Automation Save As XPS failed |
| 95     | 0x005F     | BiPrint Outlook item read receipt request error  
This print error occurs when unable to process Outlook messages when a read receipt is requested. |
| 96     | 0x0060     | BiPrint MSG file non printable properties  
This print error occurs when processing Outlook messages using MAPI API fails. |
| 42767  | 0xA70F     | Unspecified error |

**Publishing Error Notes:**
The BiPrint/Net-It Now publishing module performs print publishing using either Office automation or DDE. Unless mentioned specifically, the error messages apply to both.

**BIPrint.ini**
The IGC Writer configuration file is installed to `C:\Program Files\IGC Writer\BIPrint.ini` by default. Please refer to the `BIPrint_userguide.pdf` document located in this same directory for detailed
information about the default publishing settings contained in this file.

**Notable parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **WordRevisionMode** | This setting determines whether Microsoft Word displays revisions in balloons in the margin, or in line with the document’s text.  
0= Balloon revisions  
1= Inline revisions |
| **OutlookMessageFormat** | Determines the format to convert to for Outlook messages.  
0= Keeps the existing MSG format (default)  
1= Force Plain Text Format (converts to plain text format, removing links and text formatting; speeds processing)  
2= Force Rich Text Format (converts to Rich Text format, retaining links and text formatting) |
| **OutlookMessageFilterOnPrintableProperties** | Determines whether non printable properties (e.g., delegates, read receipts) should be filtered from Outlook messages in order for printing to succeed.  
Yes= Create a copy of the original msg file and filter non-printable properties. (default)  
No= Return an error if the msg file contains no printable properties. |
CHAPTER 5: JOB PROCESSOR CONFIGURATION

CHAPTER 5 OVERVIEW

This chapter covers the Job Processor configuration options. Several tables are presented here to provide you with the details you will need when configuring your Job Processors for Net-It Enterprise.

Job Processor configuration allows the administrator to configure the Job Processor performance and its communication with the Queue Server. Proper configuration is key to optimizing performance.

This chapter covers the following:

- Setting up Multiple Job Processors for one Queue Server
- Job Processor Parameter Settings
- Process Monitor
- Page Size Settings
It’s possible (and often preferred) to have multiple Job Processors set up to provide publishing capabilities to a single installation of the Queue Server. This section will help you to successfully distribute publishing responsibility.

BACKGROUND

Although often referred to as the “Job Processor”, this implies an installation of the JobProcessor and its associated Queue Server. Each Queue Server manages publish requests, and any number of Job Processors query the Queue Server for available publishing “jobs” they can handle. This architecture allows publishing power to be increased to accommodate very small to very large systems with very little administrative responsibility.

An added feature of the JobProcessor component is that it can be configured to handle only specific types of jobs. This can allow more complicated arrangements where, for example, a few very fast computers with very fast network connections are set up to handle PDF publishing, while a few slower computers can handle the relaxed requirements of other file types. Or, it can be used to direct types that require Microsoft Office to be installed to only a small subset of computers for which that product has been purchased, while using other machines to handle other types.

PREREQUISITES

Before creating multiple job processors to work with one Queue Server, the Queue Server must be set up. These can normally be installed at the same time, using the same installation program, and most often on the same computer. Once a Queue Server is installed and running, a Job Processor can be installed elsewhere and directed to connect to the Queue Server.

An important requirement that must be satisfied for everything to work is that all Job Processors must be UNC-accessible to and from the Queue Server. Although requests are handled via HTTP, actual file transfers occur using the UNC mechanism.

SETUP

Perform the following steps for each Job Processor that will communicate with a single Queue Server.

1. Using the installation program, install the JobProcessor as described in Chapter 2. This can be on any computer system that has both HTTP and UNC access to the machine on which the Net-It Server is installed.

2. Verify the installation by checking to see that the program now resides on the machine C:\Program Files (x86)\IGC\Net-It Enterprise .NET\JobProcessor.
3. In the JobProcessor directory, locate the `JobProcessor.config` file. Open it and acquaint yourself with the following sections:
   a. `thread.drw,thread.pdf,thread.doc,thread.prq`
   b. `queue.server.address`

4. To direct the new job processor to connect to the existing Queue Server, you will change the `queue.server.address` property.

5. To define which types of files this job processor can handle, you will change the `drw, pdf, doc, and prq thread properties`:
   To allow only a certain group of file types, list only that group in the `jobprocessor.config` file, and comment out all the other thread properties (start the line with a `'#'`), leaving the ones that match the thread properties you want to keep.

   For example, suppose you want this job processor to handle only prq and drw file groups. The thread properties would look like this:
   ```plaintext
   thread.drw=5
   #thread.pdf=1
   #thread.doc=1
   thread.prq=1
   ```

   It is not recommended under normal circumstances to change the number of threads listed for each file group.

   Note that your specific license restrictions may prevent the option of configuring threads.

---

**JOB PROCESSOR PARAMETER SETTINGS**

This table summarizes the Job Processor configuration settings (set in `JobProcessor.config`).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dataport</strong></td>
<td>The port that the Job Processor opens to provide configuration and log web pages. Point your browser at <a href="http://JPMachine:7070/config">http://JPMachine:7070/config</a> to see configuration information.</td>
<td>7070</td>
</tr>
<tr>
<td><strong>log</strong></td>
<td>Enables or disabled job logging by the Job Processor. The log file location is displayed in the Job Processor Console window and is typically C:\Documents and Settings&lt;JP USER ID&gt;\local settings\application data\Informative Graphics Corp\ or C:\Users&lt;JP USER ID&gt;\AppData\Local\Informative Graphics</td>
<td>Off</td>
</tr>
</tbody>
</table>
### Chapter 5: Job Processor Configuration

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>log.size</td>
<td>The maximum number of jobs that are written to the log file at any time. Older jobs are removed from the log as newer jobs are processed. If performance is an issue, logging may be turned off by setting the <code>log=off</code> parameter.</td>
<td>200</td>
</tr>
<tr>
<td>thread.drw</td>
<td>The number of jobs from the DRW queue to concurrently publish.</td>
<td>5</td>
</tr>
<tr>
<td>thread.pdf</td>
<td>The number of jobs from the PDF queue to concurrently publish.</td>
<td>1</td>
</tr>
<tr>
<td>thread.doc</td>
<td>The number of jobs from the DOC queue to concurrently publish.</td>
<td>1</td>
</tr>
<tr>
<td>persistentqueues</td>
<td>A comma separated list of <code>(thread.*)</code> queues that you wish to be persisted. This setting allows a document, once loaded, to be re-used for multiple jobs if that document will be processed multiple times.</td>
<td>doc,pdf</td>
</tr>
<tr>
<td>persisttimeout</td>
<td>The amount of time (in seconds) a persistent document will be kept in memory before being recycled.</td>
<td>10</td>
</tr>
<tr>
<td>queue.server.address</td>
<td>The hostname of the Server which has an internal publish request queue. Used by the IGC.JobProcessor.PollingHttpJobGetter.</td>
<td>mymachine.int.foobar.com</td>
</tr>
<tr>
<td>jobgetter.classname</td>
<td>The name of the Job Getter used to retrieve jobs for the Job Processor to process. The JobProcessor.config file supports multiple job getters. To add multiple job processors, use the following syntax:</td>
<td><code>IGC.JobProcessor.PollingHttpJobGetter</code> <code>IGC.JobProcessor.DirectoryWatcherJobGetter</code> <code>IGC.JobProcessor.PopProcessorJobGetter</code></td>
</tr>
<tr>
<td>job.file.dir</td>
<td>The directory where the DirectoryWatcherJobGetter retrieves jobs. See the Net-It SDK guide for more information.</td>
<td><code>\mymachine\jobs</code></td>
</tr>
<tr>
<td>dir.watcher.sleep.time</td>
<td>The amount of time (in milliseconds) that the DirectoryWatcherJobGetter spends looking for new jobs</td>
<td>250</td>
</tr>
</tbody>
</table>
before returning ‘no job present’. See the Net-It SDK guide for more information.

<table>
<thead>
<tr>
<th><strong>tasks.config</strong></th>
<th>The location of the task definition file. DO NOT MODIFY.</th>
<th>./tasks.config</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>maxmessagecount</strong></td>
<td>The maximum number of warnings or error messages written to the log file per job.</td>
<td>20</td>
</tr>
</tbody>
</table>

### PROCESS MONITOR

Methods are available to further manage the processes created by print-publishing. These systems are enabled in JobProcessor.config. Note that this management code is selective though aggressive in terminating non-responsive processes, and should only be used on machines with dedicated Job Processors.

### MONITORING PROCESSES DURING PRINT PUBLISHING

This mechanism takes a snapshot of the processes running on the system before print-publishing begins. After print-publishing is complete, it takes another snapshot of the system to compare running processes. Any programs (from an administrator-defined list) that are found to be running after print-publishing completes (but were not running before print-publishing began) are closed.

To enable this system, configure the ProcessesToMonitor parameter in JobProcessor.config to monitor the desired processes. For example:

```
ProcessesToMonitor=winword.exe,excel.exe
```

This command would monitor only for instances of Microsoft Word and Excel. Additional applications can be added to the comma separated list. **The list is case-sensitive.**

### SCANNING FOR STALE AND ORPHANED PROCESSES AND LEFT OVER FILES

An additional system is available in the Job Processor to monitor the Job Processor machine for stale processes and files. Due to the nature of print-publishing, there is always a chance that processes will hang or fail to clean up temporary files. This system in the Job Processor exists to periodically clean up these print-publishing artifacts.

Configuration of this system requires a handful of lines to be configured in the JobProcessor.config file. The section is shown below which includes instructions for each entry.

```
##############################################################
# Process cleanup Settings, to monitor processes by polling
```
Chapter 5: Job Processor Configuration

EnableProcessMonitor=true

# Log file
ProcessMonitorLogFile=%temp%\Informative Graphics Corp\IGCMonitorServiceLog.txt
ProcessMonitorLogFileMaxSizeInMB=1

# timeout in seconds to kill automation launched processes
# Default = 14400 seconds (4 hours, 4 * 60 * 60) --> EnableProcessMonitor=false
StaleProcessTimeout=14400

# Processes to kill if launched by automation and idle for "timeout" seconds, comma separated list. This list IS CASE SENSITIVE
ProcessesToMonitor=excel.exe, winword.exe, powerpnt.exe, excelcnv.exe, EXCEL.EXE, WINWORD.EXE, POWERPNT.EXE, EXCELCNV.EXE

# Polling frequency, in seconds. How often to check for orphaned or stale automation processes (default 300 seconds)
ProcessMonitorPollingFrequencyInSeconds=300

# The parent service for processes to kill. We find the service process with this DisplayName and kill only children of that process
ProcessMonitorParentProcessName=DCOM Server Process Launcher

# Set KillOrphans to true if you wish to kill any found processes (listed in ProcessesToMonitor) that have a parent process that no longer exists.
ProcessMonitorKillOrphans=true

# List the directories you wish to clean up here. Additional entries may be added by increasing the integer 0, 1, 2, etc. You may not skip entries.
ProcessMonitorCleanupDir0=c:\windows\temp
ProcessMonitorCleanupDir1=%temp%
ProcessMonitorCleanupDir2=C:\Documents and Settings\<USER>\Local Settings\Temp

# What files to delete (cmd style expressions)
ProcessMonitorFileDeletePattern0=DL_*
ProcessMonitorFileDeletePattern1=*.EMF
ProcessMonitorFileDeletePattern2=*.GRP
ProcessMonitorFileDeletePattern3=*.GRF.XML

# What directories to delete. ALL FILES in these subdirectories of the temp directories above will be deleted. Every file in the directory must be older (access time) than TempFileMaxAge
ProcessMonitorDirectoryDeletePattern0=DL_*

# Maximum age for temp files (from last access time)-->
# Default = 14400 seconds (4 hours, 4 * 60 * 60)-->
TempFileMaxAgeInSeconds=14400

---

**PAGE SIZE SETTINGS**

**Outputpagesizetable**: This table summarizes the Job Processor page size configuration settings (set in `Pagesizes.config`).

The file can be extended if desired. To change the file, you must restart the Job Processor. The page sizes are listed without respect to orientation. ISO, JIS, ARCH, and ANSI sizes are listed with smaller dimension first. Letter through Executive are listed in width by height order for the expected (portrait/landscape) orientation.

```
outputpagesize=ISO A1
outputpagesize=Letter
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Orientation</th>
<th>Units</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO A0</td>
<td>Landscape</td>
<td>mm</td>
<td>1189</td>
<td>841</td>
</tr>
<tr>
<td>ISO A1</td>
<td>Landscape</td>
<td>mm</td>
<td>841</td>
<td>594</td>
</tr>
<tr>
<td>ISO A2</td>
<td>Landscape</td>
<td>mm</td>
<td>594</td>
<td>420</td>
</tr>
<tr>
<td>ISO A3</td>
<td>Landscape</td>
<td>mm</td>
<td>420</td>
<td>297</td>
</tr>
<tr>
<td>ISO A4</td>
<td>Landscape</td>
<td>mm</td>
<td>297</td>
<td>210</td>
</tr>
<tr>
<td>ISO A5</td>
<td>Landscape</td>
<td>mm</td>
<td>210</td>
<td>148</td>
</tr>
<tr>
<td>ISO A6</td>
<td>Landscape</td>
<td>mm</td>
<td>148</td>
<td>105</td>
</tr>
<tr>
<td>ISO A7</td>
<td>Landscape</td>
<td>mm</td>
<td>105</td>
<td>74</td>
</tr>
<tr>
<td>ISO A0</td>
<td>Portrait</td>
<td>mm</td>
<td>841</td>
<td>1189</td>
</tr>
<tr>
<td>ISO A1</td>
<td>Portrait</td>
<td>mm</td>
<td>594</td>
<td>841</td>
</tr>
<tr>
<td>ISO A2</td>
<td>Portrait</td>
<td>mm</td>
<td>420</td>
<td>594</td>
</tr>
</tbody>
</table>
# Chapter 5: Job Processor Configuration

<table>
<thead>
<tr>
<th>Format</th>
<th>Orientation</th>
<th>Unit</th>
<th>Width</th>
<th>Height</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO A3</td>
<td>Portrait</td>
<td>mm</td>
<td>297</td>
<td>420</td>
</tr>
<tr>
<td>ISO A4</td>
<td>Portrait</td>
<td>mm</td>
<td>210</td>
<td>297</td>
</tr>
<tr>
<td>ISO A5</td>
<td>Portrait</td>
<td>mm</td>
<td>148</td>
<td>210</td>
</tr>
<tr>
<td>ISO A6</td>
<td>Portrait</td>
<td>mm</td>
<td>105</td>
<td>148</td>
</tr>
<tr>
<td>ISO A7</td>
<td>Portrait</td>
<td>mm</td>
<td>74</td>
<td>105</td>
</tr>
<tr>
<td>ISO B0</td>
<td>Landscape</td>
<td>mm</td>
<td>1414</td>
<td>1000</td>
</tr>
<tr>
<td>ISO B1</td>
<td>Landscape</td>
<td>mm</td>
<td>1000</td>
<td>707</td>
</tr>
<tr>
<td>ISO B2</td>
<td>Landscape</td>
<td>mm</td>
<td>707</td>
<td>500</td>
</tr>
<tr>
<td>ISO B3</td>
<td>Landscape</td>
<td>mm</td>
<td>500</td>
<td>353</td>
</tr>
<tr>
<td>ISO B4</td>
<td>Landscape</td>
<td>mm</td>
<td>353</td>
<td>250</td>
</tr>
<tr>
<td>ISO B5</td>
<td>Landscape</td>
<td>mm</td>
<td>250</td>
<td>176</td>
</tr>
<tr>
<td>ISO B6</td>
<td>Landscape</td>
<td>mm</td>
<td>176</td>
<td>125</td>
</tr>
<tr>
<td>ISO B7</td>
<td>Landscape</td>
<td>mm</td>
<td>125</td>
<td>88</td>
</tr>
<tr>
<td>ISO B0</td>
<td>Portrait</td>
<td>mm</td>
<td>1000</td>
<td>1414</td>
</tr>
<tr>
<td>ISO B1</td>
<td>Portrait</td>
<td>mm</td>
<td>707</td>
<td>1000</td>
</tr>
<tr>
<td>ISO B2</td>
<td>Portrait</td>
<td>mm</td>
<td>500</td>
<td>707</td>
</tr>
<tr>
<td>ISO B3</td>
<td>Portrait</td>
<td>mm</td>
<td>353</td>
<td>500</td>
</tr>
<tr>
<td>ISO B4</td>
<td>Portrait</td>
<td>mm</td>
<td>250</td>
<td>353</td>
</tr>
<tr>
<td>ISO B5</td>
<td>Portrait</td>
<td>mm</td>
<td>176</td>
<td>250</td>
</tr>
<tr>
<td>ISO B6</td>
<td>Portrait</td>
<td>mm</td>
<td>125</td>
<td>176</td>
</tr>
<tr>
<td>ISO B7</td>
<td>Portrait</td>
<td>mm</td>
<td>88</td>
<td>125</td>
</tr>
<tr>
<td>JIS B4</td>
<td>mm</td>
<td></td>
<td>364</td>
<td>257</td>
</tr>
<tr>
<td>JIS B5</td>
<td>mm</td>
<td></td>
<td>182</td>
<td>257</td>
</tr>
<tr>
<td>Letter</td>
<td>in</td>
<td></td>
<td>8.5</td>
<td>11.0</td>
</tr>
<tr>
<td>Legal</td>
<td>in</td>
<td></td>
<td>8.5</td>
<td>14.0</td>
</tr>
<tr>
<td>Ledger</td>
<td>in</td>
<td></td>
<td>17.0</td>
<td>11.0</td>
</tr>
<tr>
<td>Tabloid</td>
<td>in</td>
<td></td>
<td>11.0</td>
<td>17.0</td>
</tr>
<tr>
<td>Executive</td>
<td>in</td>
<td></td>
<td>7.25</td>
<td>10.55</td>
</tr>
<tr>
<td>Arch C Size Sheet</td>
<td>in</td>
<td>24.0</td>
<td>18.0</td>
<td></td>
</tr>
<tr>
<td>Arch D Size Sheet</td>
<td>in</td>
<td>36.0</td>
<td>24.0</td>
<td></td>
</tr>
<tr>
<td>Arch E Size Sheet</td>
<td>in</td>
<td>48.0</td>
<td>36.0</td>
<td></td>
</tr>
<tr>
<td>Ansi C Size Sheet</td>
<td>in</td>
<td>22.0</td>
<td>17.0</td>
<td></td>
</tr>
<tr>
<td>Ansi D Size Sheet</td>
<td>in</td>
<td>34.0</td>
<td>22.0</td>
<td></td>
</tr>
<tr>
<td>Ansi E Size Sheet</td>
<td>in</td>
<td>44.0</td>
<td>34.0</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER 6 OVERVIEW

This chapter includes a collection of tips for addressing common issues you may encounter while installing and configuring your Net-It Enterprise software.

Every installation and configuration is unique based on your requirements, third-party software, and equipment. You will find the information in this topic helpful when addressing setup, optimization and maintenance issues.

This chapter contains information on the following topics:

- Uninstalling
- Anti-Virus Software
- Workaround Macros
- Publishing Errors
UNINSTALLING

If Terminal Services is installed on the Job Processor, and you wish to uninstall and/or reinstall the Job Processor, you must also manually uninstall the IGC Writer Print Driver through the Control Panel Add/Remove Programs dialog. This process can be done either before or after uninstalling Net-It Enterprise.

If you have uninstalled a previous version of Net-It Enterprise and are receiving -21 errors when attempting to publish a document, then try uninstalling the IGC Writer Drivers and clean up the associated files:

1. Uninstall the containing product (i.e. Net-It Enterprise)
2. Uninstall the IGC Writer Driver if it is listed in the Control Panel's "Add or Remove Programs" panel (listed as "IGC Writer").
3. Reinstall Net-It Enterprise.
4. If the three above steps do not solve the problem, repeat steps 1 and 2 and continue with the following steps.
5. Unzip the BlackIceCleaner.zip file located in the JobProcessor directory.
6. Copy the resulting BClean.exe file to your machine’s C: drive.
7. Boot the machine into safe mode and launch BClean.exe.
8. Reboot your machine, and then reinstall Net-It Enterprise.

WORKAROUNDS

MICROSOFT OUTLOOK

To publish MSG files, Microsoft Outlook needs to be installed and initially set up on the Job Processor machine with no personal mail account information set and no Exchange server information set. After this initial setup, close Outlook. Processing MSG files should now launch in Outlook and successfully print-publish.

If you do not perform this initial setup, MSG files do not publish and a critical error message is logged. Since the IGC Writer may forcefully close Outlook during the print publishing process, there should be no email accounts configured in the Outlook application on the Job Processor machine because those
accounts can become corrupted. If the Job Processor is having problems converting MSG files, do the following:

1. Shut down the Job Processor.
2. Make sure IGC Writer and Outlook processes are not running (go to the Windows Task Manager and end the processes if they show up there).
3. Manually attempt to open Outlook and make sure it can launch without reporting errors.
5. Restart the Job Processor.

---

**ANTI-VIRUS SOFTWARE**

The Job Processor is not supported on machines running anti-virus software. If you have this configuration and you are experiencing problems, turn the anti-virus software off on the Job Processor machine and restart the Job Processor.

If you must run virus protection software on the Job Processor server, the following directories should be excluded from scanning:

- C:\Windows\system32\spool\PRINTERS
- C:\Windows\Temp
- C:\Documents And Settings\{user}\Local Settings\Temp
- C:\Program Files (x86)\IGC\Brava!\JobProcessor

If you are experiencing issues with excluding C:\Windows\Temp, you can update the system output path of the IGC Writer Print Driver. The steps are:

1. In the Control Panel "Printers and Faxes" panel, right-click on the "IGC Writer" printer and select Properties.
2. Select the Advanced tab.
3. Click the "Printing Defaults..." button.
4. Select the "Filename Generation" tab.
5. Change the Output Directory path to the following:
   - On the Queue Server machine (which is typically the same as the Job Processor machine), the suggested path is:
     C:\igcprq
   - On the Net-It Server machine, the suggested path is:
     C:\Program Files (x86)\IGC\Net-It Enterprise
6. Click OK to save your settings.
Chapter 6: Tips and Troubleshooting

PUBLISHING ERRORS

- **Running a Job Processor on a Windows 2008 Server**: Windows 2008 Server places restrictions on services running in session zero. Informative Graphic's print-publishing technology requires access to session zero and the Job Processor service is configured (upon installation to a Windows 2008 server) to access session zero. Windows 2008, however, ships with a service (called "Interactive Services Detection") that may cause unnecessary and annoying dialog boxes to pop up on the Job Processor machine. Therefore, it is recommended that you disabled the Interactive Services Detection service on the Job Processor machines.

- Some instances of Net-It Enterprise ship with licenses that restrict the number of queues that the Job Processor can process. If more than the number of allowable threads are attempted to be created (as defined in the license), warnings will be created in the event log and the system will continue to run as licensed.

- If you have upgraded your Internet Explorer version from IE 6.0 on the Job Processor machine and are experiencing problems publishing HTML files, you may need to edit your registry for print publishing to use command line printing instead of DDE printing. (Be sure to back up your registry before performing any edits.) To do this:
  1. Through regedit, browse your registry to locate "HKEY_CLASSES_ROOT\htmlfile\shell"
  2. There should only be one key (folder) listed in the printto folder called command (that is set with Type REG_SZ for IE 6,0 or REG_EXPAND_SZ for IE 7.0 to run rundll32.exe)
  3. If an additional key (folder) exists called ddeexec, you need to delete this DDE key (folder) so that print publishing will only use command line printing with HTML files.
  4. Alternatively, you can turn off DDE printing by going to Windows explorer->Tools->Folder Options->File Types Tab->Registered File Types->html,htm->printo action->Edit->Deselect DDE. Sometimes when editing through this dialog, DDE is re-selected, so please confirm that DDE is not being used by checking the registry.

- If attempting to publish an Office 95 compliant file with Office 2010, print/XPS publishing may fail if the file block protection settings are enabled. This setting may be altered through your Office application file options:
  
  From your Office application, go to:
  File->Options->Trust Center->Trust Center Settings->File Block Settings
  
  Any file checked as 'open' will fail
The reference table below is provided as a quick reference for input job options, grouped in order of function. For full descriptions, please see the table in Chapter 4:

<table>
<thead>
<tr>
<th>Publishing Parameter / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source</strong></td>
</tr>
<tr>
<td><strong>SourceN</strong></td>
</tr>
<tr>
<td><strong>Target</strong></td>
</tr>
<tr>
<td><strong>OutputFormat</strong></td>
</tr>
<tr>
<td><strong>NotificationUrl</strong></td>
</tr>
<tr>
<td><strong>NotificationVerb</strong></td>
</tr>
<tr>
<td><strong>SecurityXmlFilename</strong></td>
</tr>
<tr>
<td><strong>FitWithinBanners</strong></td>
</tr>
<tr>
<td><strong>CSFName</strong></td>
</tr>
<tr>
<td><strong>OutputColorMode</strong></td>
</tr>
<tr>
<td><strong>PublishPageList</strong></td>
</tr>
<tr>
<td><strong>HeartBeat</strong></td>
</tr>
<tr>
<td><strong>StepTimeout</strong></td>
</tr>
<tr>
<td><strong>DocumentTitle</strong></td>
</tr>
<tr>
<td><strong>ExportFilePerPage</strong></td>
</tr>
<tr>
<td><strong>PublishUserName</strong></td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
</tr>
<tr>
<td><strong>Priority</strong></td>
</tr>
<tr>
<td><strong>ctbfile</strong></td>
</tr>
<tr>
<td><strong>intloutputstringfile</strong></td>
</tr>
<tr>
<td>TIFF Parameter / Description</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>ForceTiffMonochrome</td>
</tr>
<tr>
<td>TiffBpp</td>
</tr>
<tr>
<td>TiffDpi</td>
</tr>
<tr>
<td>TiffCompressionType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PDF Parameter / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PdfExportA1b</td>
</tr>
<tr>
<td>ExportLayerState</td>
</tr>
<tr>
<td>ShowLineWeights</td>
</tr>
<tr>
<td>ShowThinLinesOnly</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Page Size Parameter / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OutputPageSize</td>
</tr>
<tr>
<td>CreatePageSizesTextFile</td>
</tr>
<tr>
<td>CreatePageSizesTextMetric</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Markup Parameter / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarkupFilename</td>
</tr>
<tr>
<td>MarkupListFilename</td>
</tr>
<tr>
<td>ExportMarkupFilename</td>
</tr>
<tr>
<td>ExportChangemarkSummary</td>
</tr>
<tr>
<td>PdfExportMarkupsAsAnnotations</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thumbnail Parameter / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThumbFormat</td>
</tr>
<tr>
<td>ThumbName</td>
</tr>
<tr>
<td>ThumbSizes</td>
</tr>
<tr>
<td>ThumbWidths</td>
</tr>
<tr>
<td>ThumbHeights</td>
</tr>
<tr>
<td>ThumbPages</td>
</tr>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>--------------------</td>
</tr>
<tr>
<td>ThumbQuality</td>
</tr>
<tr>
<td>JpegName</td>
</tr>
<tr>
<td>JpegSizes</td>
</tr>
<tr>
<td>JpegWidths</td>
</tr>
<tr>
<td>JpegHeights</td>
</tr>
<tr>
<td>JpegPages</td>
</tr>
<tr>
<td>JpegQuality</td>
</tr>
</tbody>
</table>

**Get Text Parameter / Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetText</td>
<td>Instructs the server to produce a set of text files with all of the text from the published document extracted as plain text.</td>
</tr>
<tr>
<td>AddMetadataToTextOutput</td>
<td>If both this and the GetText parameter are set to True, then any metadata present in the source file will be included in the text output pages.</td>
</tr>
<tr>
<td>TextFormatUTF8</td>
<td>If true, then the output format produced from GetText is UTF-8 encoded.</td>
</tr>
<tr>
<td>txtformatcrlf</td>
<td>Set to true if you would like text output to be formatted with carriage returns separating each line.</td>
</tr>
</tbody>
</table>

**Stamp Parameter / Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StampImage</td>
<td>Path to the image file to stamp, using the Stamp feature. It must be a UNC path to a JPG or PNG file.</td>
</tr>
<tr>
<td>StampPositionAndSize</td>
<td>Multiple values that specify the size and position of the stamp image (XSP file). The string parameter format is: StampPositionAndSize=left</td>
</tr>
<tr>
<td>StampTemplateFileName</td>
<td>The UNC path to a Stamp Template file.</td>
</tr>
<tr>
<td>StampTemplatePositionAndSize</td>
<td>Multiple values that specify the size and position of the stamp template. The string parameter format is: StampTemplatePositionAndSize=left</td>
</tr>
</tbody>
</table>
This software includes third party component software distributed by IGC to you pursuant to specific third party license agreements, whose terms and conditions are as set forth in your license agreement with IGC and/or the Terms and Conditions of Embedded Products.

Below please find the applicable copyright notices required by such third party licenses:

- Portions Copyright © 1988-1997 Sam Leffler. All Rights Reserved.
- Portions Copyright © 1991-1997 Silicon Graphics, Inc. All Rights Reserved.
- Portions Copyright © 1995-2005 Jean-loup Gailly and Mark Adler. All Rights Reserved.
- Portions Copyright © 2004, 2006-2007 Glenn Randers-Pehrson. All Rights Reserved.
- Portions Copyright © 2003 Boost Software. All Rights Reserved.
- Portions Copyright © 2011 John Resig. All Rights Reserved.

Acknowledgements:

- This software is based in part on the work of the Independent JPEG Group. (www.iijg.org)
- This software includes LibTiff software developed by Sam Leffler while working for Silicon Graphics. (www.sgi.com)
- This software includes ZLib software developed by Jean-loup Gailly and Mark Adler (www.zlib.net/zlib_license.html)
- This software includes LibPng software developed by Glenn Randers-Pehrson and Cosmin Truta (http://www.libpng.org/pub/png/src/libpng-LICENSE.txt)
- This software includes Boost libraries developed by Boost Software. (www.boost.org/LICENSE_1_0.txt)
- This software includes DWF Toolkit developed by Autodesk, Inc. (http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=5522878)