

WINDWARD STUDIOS

---

Windward Reports 7.0

# System Administrator's Guide for .NET

---

WINDWARD REPORTS

# System Administrator's Guide for .NET

---

© 2002 – 2009 Windward Studios, Inc.  
2945 Center Green Court South  
Boulder, CO 80301  
Phone 303.499.2544 • Fax 303.554.5636

Last revised 6 February 2009  
Revised through version 7

This document Copyright © 2002 – 2009 by Windward Studios, Inc.  
All Rights Reserved

---

# Table of Contents

Windward Reports	2
About This Guide	2
Providing Feedback	3
Installing Windward Reports	4
System Requirements	4
Configuration File: app.exe.config	4
The License Key	5
Log4net	5
Ready to Run	5
Testing the Installation	6
ADO Data Sources	7
ADO .NET Connection Drivers	7
Views	7
F.A.Q.	8
Windward Studios	9
Location	9
Sales	9
Technical Support	9
Appendices	10
Licensing F.A.Q.	10
About the Licensing Copy Protection	10
WindwardReports Configurations	12
Locale and Resources	14
Index	16

---

# Windward Reports

*An introduction to Windward Reports.*

**W**indward Reports™ is the only .NET report engine that uses Microsoft Word™, Excel™ or PowerPoint™ as its layout tool -- so creating reports is fast & easy, for both technical and non-technical users.

With the robust, plug 'n' play Windward Reports engine, you've got an elegant, flexible and very affordable reporting solution.

It's as easy as 1-2-3:

1. Design your report template using MS-Word (or any RTF-capable editor), Excel or PowerPoint. Templates can be created in RTF WordML (XML), DOCX, XLSX, or PPTX.
2. Send your XML and/or SQL data source and the report template to the Windward Reports engine.
3. Specify your desired output format: PDF, DOCX, XLSX, PPTX, RTF, HTML, TXT, WordML (Word's XML format), SpreadsheetML (Excel's XML format), and XLS.

That's it! There's no learning curve or added expenses for your layout tool, and the intuitive JSP-like formatting syntax lets you create everything from simple everyday reports to complex, enterprise-wide analyses.

## About This Guide

This guide is for the system administrator responsible for installing and maintaining Windward Reports on a server. This document lists various methods of installing the program; however, the system administrator should check with the development group to see if they need a specific configuration.

## Providing Feedback

We have tried to make our documentation as clear and complete as possible. But only you can tell us what we missed. So if you are new to Windward Reports, please email [docs@windward.net](mailto:docs@windward.net) with your suggestions for improvement... especially when you find yourself wishing for something that would save you a lot of time.

Thank you for your feedback, – The Windward Documentation Team

## Installing Windward Reports

For quick installation instructions, see the *Getting Started Guide*. This section contains additional detail.

### System Requirements

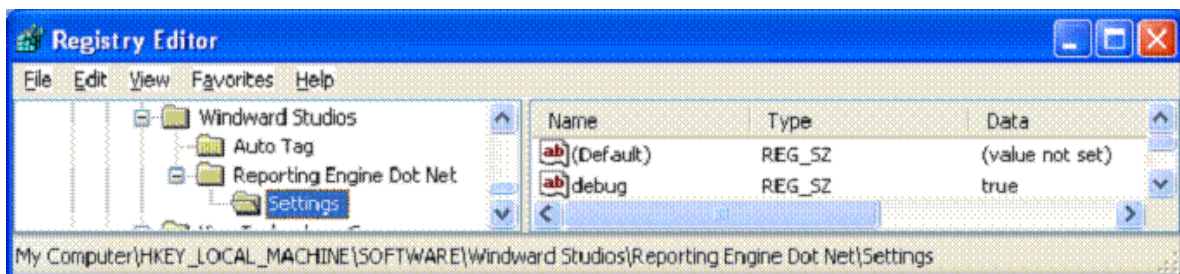
Windward Reports is a .NET library that runs under .NET 2.0 or later. During setup, a number of .DLL files are installed, including WindwardReports.dll and others.

### Configuration File: app.exe.config

All configuration information goes in the App.exe.config file for the application calling Windward Reports. There is a sample WindwardReports.dll.config file to show the Windward settings – but this file is not used. You must copy this to your App.exe.config.

If a property is not found in the app.exe.config file, the program then looks in the registry to see if the property has been set there. This is done for each property, looking in app.exe.config first, and then in the registry so you can set some values in the config file and some in the registry. The config file is always checked first.

All properties are stored in HKEY\_LOCAL\_MACHINE\SOFTWARE\Windward Studios\Reporting Engine Dot Net\Settings and all properties should be stored as a String value (even numbers). Below is an example of setting the debug property in the registry.



## The License Key

A license key is required to run Windward Reports. This key holds the information that allows Windward Reports to run on your computer, including:

- How many reports a day your system may run.
- If it is a development server (only one report at a time).

Paste your license key into the app.exe.config for the application calling Windward Reports file.

As with all configuration entries, use the standard .NET name/value format when entering your license key. An example is provided in the WindwardReports.dll.config file, and is also shown below for your reference.

```
<add key="license" value="cHtzLjSa8TwnWysGZbDtvf1YB8eKHSxNWb54T0m
MoG+d/qztTfI9xdwyRx98vGk2dpjpPHYSRQOoZmE0J8o7NWYSRQWpZ220dhDTwnWysGZbDtv
j2A7T53uzsdJzqxHO1Zm61kZukHZqt/NSs5aaaThis/is/not/a/valid/key===" />
```

This can be entered as a single line, or it can be entered on multiple lines as long as each line is just the key (no spaces before or after). (The uuencode above is not a legitimate key.)

If your key is not a legitimate key, Windward Reports prints the key it loaded. Look to make sure it got all of the key and that there were no spaces in it.

## Log4net

Windward Reports uses log4net for its logging. The log4net configuration is placed in the app.exe.config file. Please refer to <http://logging.apache.org/log4net/> for information on configuring log4net.

We strongly recommend you set logging for net.windward.\* to either warn or info. If you set it to debug it will slow Windward Reports down a lot and the additional logging will be of no use to you.

**Note:** There is a log4j.dll in the set of Windward Reports dlls. This dll translates log4j calls to log4net. Windward Reports is not using log4j in the .net version.

## Ready to Run

Windward Reports is now ready to run. You do not need to reboot your system. You may need to reopen your console windows to view the environment changes in a console window.

## Testing the Installation

*But it works here...*

To test that the installation is okay, Windward Reports includes a program that generates reports from the command line. The easiest way to test the installation is to go to the directory where you installed Windward Reports, and then go to the `test` sub-directory.

Before you run the test, make sure you have placed your license key in the calling application's config file! (In this case, `RunReport.exe.config`.)

In `test`, run this command:

```
run.bat
```

This uses the data file `order.xml` and the template file `template.rtf` to create the report `report.htm`. If it creates the file, then Windward Reports is installed and running.

## ADO Data Sources

*Using a SQL database as the report generation data source.*

If your users need a SQL database for the data source of their reports, read this chapter. Otherwise, you may skip it.

## ADO .NET Connection Drivers

If you need help determining the connection string for your ADO .NET driver, a good source of information is <http://www.connectionstrings.com>.

## Views

Windward Reports creates temporary views with the name WR\_1234ef where the 1234ef is a random combination of up to 6 hexadecimal digits. Windward Reports normally deletes these views when it completes a report.

However, if an exception is thrown, and in the cleanup of the exception the calling program does not close the Windward Reports connection, then the view is not deleted. It is safe to delete these views when Windward Reports is not running.

## F.A.Q.

*Everything else is here.*

**Q: What network ports does Windward Reports use?**

A: Windward Reports uses port 1707, which is assigned to Windward Studios by the Internet Assigned Names Authority (IANA). This is used for internal purposes (this cannot be changed) and by its server (this can be changed). Please note that *AutoTag* (the Windward Reports Word add-in) also uses this port for both internal purposes as well as to communicate with the server.

When you encounter a problem, even if you solve it yourself, please let us know about it. We can then add the issue here or in the support forum for future users.

## Windward Studios

### *Finding Us*

#### Location

Windward Studios, Inc.  
2945 Center Green Court South  
Boulder, CO 80301  
U.S.A.  
01-303-499-2544  
[www.windwardreports.com](http://www.windwardreports.com)

#### Sales

On-line sales can be made 24 hours a day at our online store.

<https://store.windward.net/login.aspx>

The sales department can be reached from 8:00 AM to 5:00 PM, Mountain Time at:  
01-303-499-2544

#### Technical Support

Online technical support can be reached 24 hours a day on our support forums.

- Please go to <http://www.windwardreports.com/ohana/> to access the support forums.

The following two support options are available only to users who purchased a support contract:

- E-mail technical support can be reached from 8:00 AM to 5:00 PM, Mountain Time. E-mail questions are replied to within 30 minutes. Please send e-mail to [support@windward.net](mailto:support@windward.net).
- Technical Support can be reached from 8:00 AM to 5:00 PM, Mountain Time at:  
1-303-499-2544

# Appendices

## Licensing F.A.Q.

Here are the questions we normally get about our licensing. If you have any questions about the number of licenses you need to have, please contact [sales@windward.net](mailto:sales@windward.net).

- You must have a separate license for each computer that executes Windward Reports code.
- You do not need a client license for machines whose browser will view generated reports, only for the machine the reports are created on (usually a central server).
- The "Standard Server" runs an unlimited number of reports daily. It is **not** a license to install it on an unlimited number of machines. A single license for an unlimited server allows it to be run on one machine only.
- Where the WindwardReports.dll file is located is irrelevant— a license is needed for each machine executing code that uses WindwardReports.dll. So if WindWardReports.dll is located on a file server but never executed on the file server, the file server does not need a license. But each machine that executes code using that dll on the file server does need a license.
- You do not need a separate license for a cold backup system as long as the cold backup system has the same IP address as the hot server.
- One license per machine is required, regardless of number of users, CPUs or CPU speed.
- You must purchase support and upgrades for either all licenses or none.

## About the Licensing Copy Protection

This section contains detailed information about running the Windward Reports products on multiple machines or servers. It is provided to help the system administrator and third-party developers understand how the Windward licensing works.

AutoTag and the reporting engines (both java and .NET) have a copy protection mechanism that checks for license overuse. All three systems work the same but are independent of each other. In other words, AutoTag checks for AutoTag overuse but does not check reporting engine use. The java engine checks java engine overuse but does not check the .NET engine, etc.

In each case, when the program “starts,” it does a net multicast asking all other systems running using the “same” license to identify themselves. Each system that is already running will respond to that request, identifying itself. This uses port 1707 which is assigned to Windward by ICANN.

A license specifies the number of machines the license is valid for. If that number of machines responds to the system that just started, then the license is in full use and the system just starting will throw a `LicenseException` and will not generate a report. A `LicenseException` will not be thrown by any of the existing running engines.

AutoTag starts when Word starts. It ends when all copies of Word on a user's system exit.

The reporting engine starts when `Report.Init()` is called or the first time a report is created. It ends when the CLR it was called in exits. So if `Report.Init()` is called and no report is generated for 2 weeks but the virtual machine is still running, that system is still using the license and will respond. If `Report.Init()` is not called and the first report is generated 2 weeks after the server starts, that is when the system will determine if there is an available license for that server.

One license is good for as many virtual machines on a physical server as you wish to use.

This system is purposely set up so licenses can move between machines (unlike Microsoft's licensing which locks each license to a specific machine). Keep in mind that if you want to stop using a license on a specific server, you must stop all virtual machines on the server that called the engine.

Including the engine with your code and running your code does not cause the license to be used on a given machine. It is only if the engine code is executed. This is by design. So if you have one license and four servers, you can have the same program on all four machines including Windward Reports. You must then have a flag in your properties (true in 1, false in 3) so that only one of the servers actually calls the engine.

Each license has a unique ID. Two systems have the same license only if they have the same ID and the same settings such as product, number of reports, etc. If you purchased 3 server licenses, and then later 2 server licenses, you probably have 2 different licenses and each will have a different ID and will only check for other use of its license. In this case, if you would prefer a single license good for 5 systems, contact us and we will swap licenses giving you a single license good for 5 systems.

If a virtual machine throws the `LicenseException`, it will continue to do so on subsequent calls even if other engines have exited, making a license available. The virtual machine must be restarted for it to try again.

The java and .NET client code does not use a license. (They do indirectly because they need to call the java engine running as a server and that engine requires a license.)

The program `ListServers` will list all systems running the reporting engine by IP address. In AutoTag the About box will list the IP addresses of all systems running AutoTag by IP address. This finds all systems regardless of license ID and settings and the display does not display any of the license settings, just the IP address and server name. Each does limit itself to the same product, i.e AutoTag only shows systems running AutoTag.

## WindwardReports Configurations

The following values can be set in the config file that calls Windward Reports. Most system level settings require a restart of the system to take effect. Most report formatting level settings take effect as soon as they are made. Enter these in the config file for the application that calls Windward Reports. Use the standard format for .NET name/value pairs.

- `license=uuencoded_license_key`

Required - The license key for the program.

Restart the system to implement the changes.
- `system.locale=de_CH` or `system.locale=de`

Set the locale for system messages. If not set, the system default locale is used. This example sets it to German Switzerland and standard German, respectively.

Restart the system to implement the changes.
- `base.directory=c:/images`

Sets the default value of `ProcessReport.setBaseDirectory()`. This can be overridden for any specific `ProcessReport` object by calling `setBaseDirectory()` on that object. If this is not set the base directory is null and the default directory will be used.
- `default.image.dpi=96`

Some bitmap formats do not store the Dots Per Inch (DPI) of the image. In various situations Windward Reports must use the image size and DPI to determine the displayed size of the image. If this is not set, it will use the Windows default of 96. (Note the Apple and graphics world default is 72 so you may need to set this to 72.)
- `font.files=c:/windows/fonts/*.ttf;c:/windows/fonts/*.ttc;`

A series of paths, separated by semicolons, to where the true-type fonts are on the system. The only wildcard recognized is the \* (i.e., no regular expressions).

Restart the system to implement the changes.
- `line.break=internal`

This uses Windward's internal engine to determine line breaks and text layout instead of using Windows font metrics API. This should never be needed. This does not handle word breaks other than a space character (0x20) and a couple of other special cases.

This does not handle Thai and has not been tested with Arabic or Hebrew.

Restart the system to implement the changes.
- `report.embed.info=off`

Turns odd the report system info that is embedded as a comment in generated reports.

- `html.full_file=true`

The default value for `ProcessHtml.setFullFile()`. You must use `true` or `false`. This can still be overridden with a programmatic call to `ProcessHtml.setFullFile()`.

- `html.html_type=1`

The default value for `ProcessHtml.setSpec()`. You must use the number of the static final int specified in `ProcessHtml`. This can be overridden with a programmatic call to `ProcessHtml.setSpec()`.

- `pdf.compression_level=1`

The default value for `ProcessPdf.setCompressionLevel()`. You must use the number of the static final int specified in `ProcessPdf`. This can be overridden with a programmatic call to `ProcessPdf.setCompressionLevel()`.

- `pdf.font_level=2`

The default value for `ProcessPdf.setFontLevel()`. You must use the number of the static final int specified in `ProcessPdf`. This can be overridden with a programmatic call to `ProcessPdf.setFontLevel()`.

- `report.charset=1`

The default value for `ProcessReport.setCharset()`. You must use either a Windows codepage number (U.S. English = 1252) or the number of the static final int specified in `ProcessReport`. This can be overridden with a programmatic call to `ProcessReport.setCharset()`.

- `report.debug_mode=false`

The default value for `ProcessReport.setDebugMode()`. You must use `true` or `false`. This can be overridden with a programmatic call to `ProcessReport.DebugMode()`.

- `report.locale=de_CH` or `report.locale=de`

The default locale for `ProcessReport.setLocale()`. This can be overridden with a programmatic call to `ProcessReport.setLocale()`. This setting affects:

- the default locale in the output report (rtf only)
- display the print time field
- the *out* tag when the **type** attribute is set.

- `rtf.copy_unknown=false`

The default value for `ProcessRtf.setCopyAllTags()`. You must use `true` or `false`. This can be overridden with a programmatic call to `ProcessRtf.setCopyAllTags()`.

- `txt.break_page =false`

The default value for `ProcessTxt.setBreakPage()`. You must use true or false. This can be overridden with a programmatic call to `ProcessTxt.setBreakPage()`.

- `txt.font_name=Courier`

The default value for `ProcessTxt.setFont()`. This is the font used for all formatting of a TXT report. This can be overridden with a programmatic call to `ProcessTxt.setFont()`. The default for this is “Courier New” under windows, “Courier” in any other operating system.

- `txt.font_size=240`

The default value for `ProcessTxt.setFontSize()`. This is the size of the font used for all formatting of a TXT report. This is set in twips (20<sup>th</sup>s of a point) so a value of 240 is 12 points. This can be overridden with a programmatic call to `ProcessTxt.setFontSize()`.

- `txt.show_headers =false`

The default value for `ProcessTxt.setShowHeaders()`. You must use true or false. This can be overridden with a programmatic call to `ProcessTxt.setShowHeaders()`.

- `txt.use_margins=false`

The default value for `ProcessTxt.setUseMargins()`. You must use true or false. This can be overridden with a programmatic call to `ProcessTxt.setUseMargins()`.

- `word.version=2007`

Can be set to 2000, 2002, 2003, or 2007(default). The template will be parsed using the default values for that version of Word.

- `write.as.fields=true`

When a report writes out unused tags (primarily used by `RunCleanup`), determines if the tags will be written as fields or text. You must use true or false. The default is true.

## Locale and Resources

Windward Reports has resources for numerous locales. It loads the resources for the system locale. This can be overridden with the `system.locale` setting in the configuration file that calls Windward Reports.

In the case of Windows, when displaying in a DOS box, the default is for DOS to display codepage 437, which is the old IBM PC OEM codepage with the line drawing characters in the upper 128. If you want to display your local text correctly in a DOS box while running Windward Reports, here's how:

1. Set the DOS box to use a True Type font - usually Lucida Console is the only choice, preferably bold.
2. run `chcp 1252` - where 1252 is the Western Europe code page. Valid codepage numbers are:

- 874 (Thai)
- 932 (Japanese Shift-JIS)
- 936 (Simplified Chinese GBK)
- 949 (Korean)
- 950 (Traditional Chinese Big5)
- 1250 (Central Europe)
- 1251 (Cyrillic)
- 1252 (Latin I - Western Europe)
- 1253 (Greek)
- 1254 (Turkish)
- 1255 (Hebrew)
- 1256 (Arabic)
- 1257 (Baltic)
- 1258 (Vietnam)

# Index

Data sources

SQL, 7

F.A.Q.s, 8

Installation

testing, 6

J2EE, 2

JDBC Drivers, 7

License Key, 5

Licensing F.A.Q.s, 10

Locale and Resources, 15

Log4j, 5

network ports, 8

output formats, 2

ports, 8

Providing Feedback, 3

Running Windward Reports, 5

Sales

Windward Reports, 9

SQL data sources, 7

System Requirements, 4

Technical Support, 9

Testing the Installation, 6

Views, 7

Windward Reports Sales, 9

WindwardReports.properties, 12