.NET REPORT ENGINE USER MANUAL

Table of Contents

Getting Started with .NET Report Engine 4

.NET Report Engine API Reference 5

.NET Report Engine Example Code 6

All About Windward Studios Licensing 7

Getting Started with .NET Report Engine 18

What's New in Version 16.1? 20

FAQs 23

How Do I Enable Error Handling in .NET Report Engine? 24

How Do I Inject My License Key Into the .NET Engine? 29

How Do I Insert Input Parameters in a v16 .NET Engine Application? 30

How Do I Install .NET Report Engine? 31

How Do I Use Clean Templates For .NET Engine? 37

How Do I Use Microsoft Office to Generate PDF Output with .NET Report Engine? 39

Subscription License FAQ 40

How to Access Tags through .NET API Call? 43

How Do I Fail-Over an Engine to a New Server 44

How Do I learn what input parameters a template needs programmatically with the .NET Engine? 45

Reference 46

.NET Report Engine app.exe.config Settings 47

.NET Report Engine Configuration File Reference 60

.NET Report Engine Data Source Connectors Reference 64

Differing Versions of Report Designer and the Engine 67

Output Limitations 69

Windward Input to Output Formats 74

20.0.0 Windward Input to Output Formats 75

16.7.0 Windward Input to Output Formats 79

Miscellaneous 83

How Do I Upload Files to Mega.nz? 84

List of .NET Report Engine v15.2 DLLs and JARs 90

List of .NET Report Engine v16.1 DLLs and JARs 94

Product End of Life Policies 99
Sending a Test Template Using Sample Data 104
Support Tickets - What to Expect 105
Upcoming Release Schedule 109
Version 16 Perpetual License Update 110
Windward Software Installers 111
Windward Performance Analysis 112
Windward Studios Privacy Policy 121

Getting Started with .NET Report Engine

.NET Report Engine API Reference

The current v16 online .NET Report Engine API documentation is here.

The v15 online .NET Report Engine API documentation is here.

.NET Report Engine Example Code

Example source code for .NET Report Engine applications can be found at github.

All About Windward Studios Licensing

This article covers license keys for the various Windward Studios products; common licensing software errors; licensing restrictions; license key corruption issues; and other licensing tips.

Please use these links to jump to your section of interest.

How Do I Retrieve My License Keys?

When Must I Update My License Keys?

How Do I Update My License Keys?

Important License Information

License Key Tips

License Key Exceptions

License Key Corruption

Internal Communication Protocols and Ports

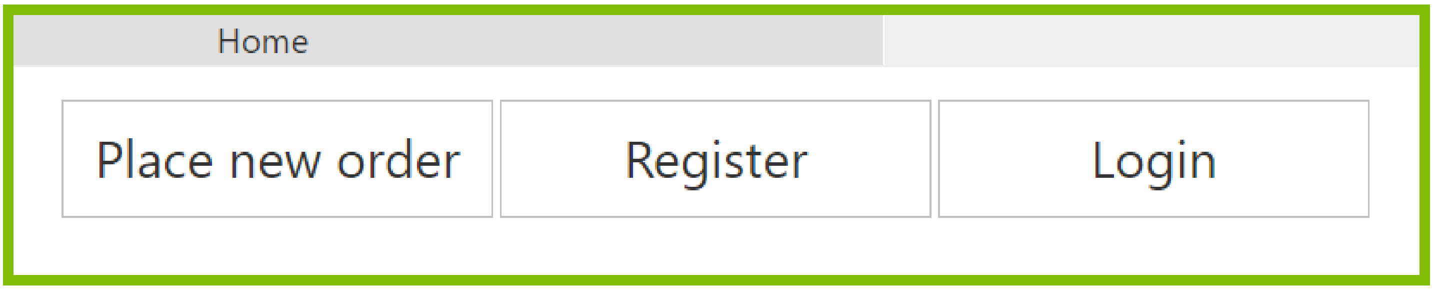
Licensing Restrictions

License Copy Protection

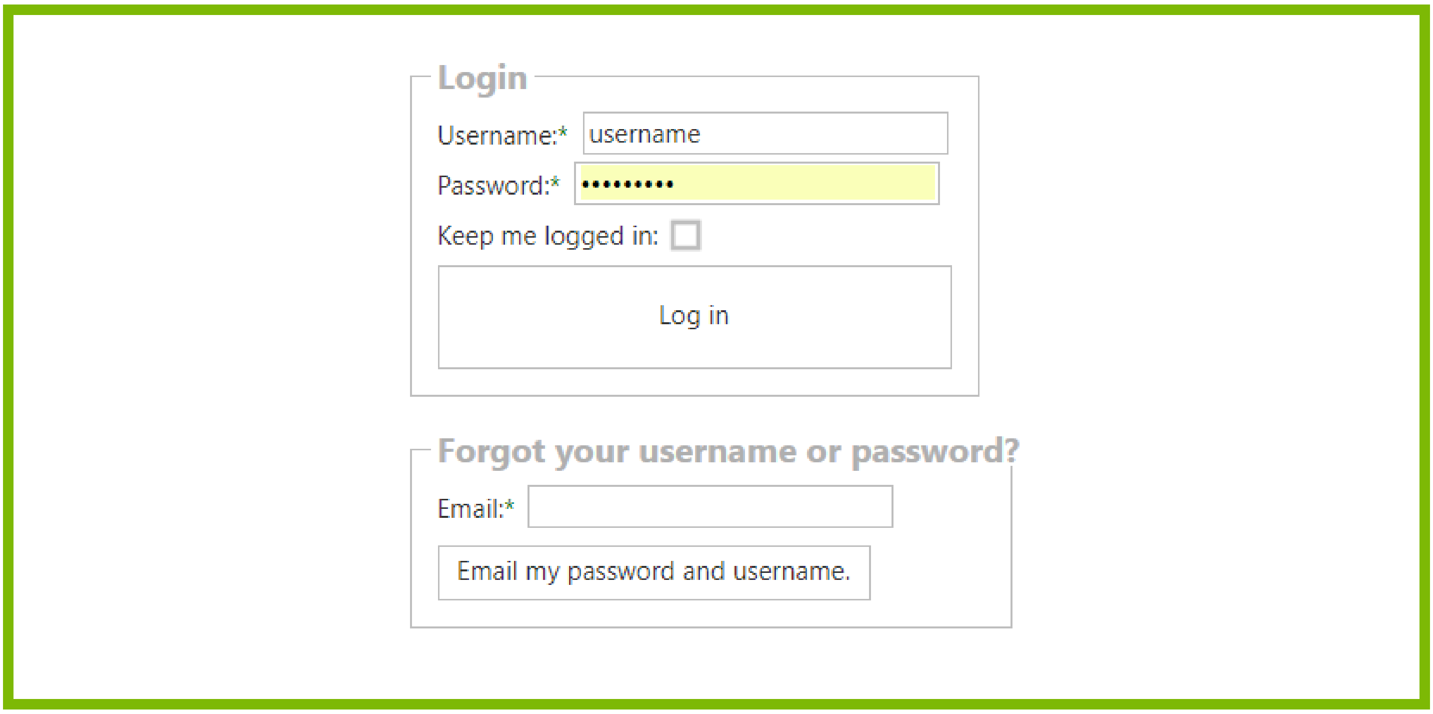
How Do I Retrieve My License Keys?

If you have not yet purchased keys or need to upgrade them, please email sales@windwardstudios.com.

1. Point your browser to https://store.windward.net.
2. Click on 'Login'



3. Enter your login info. If you've forgotten your information you can request it via this screen.



Login

Username:* username

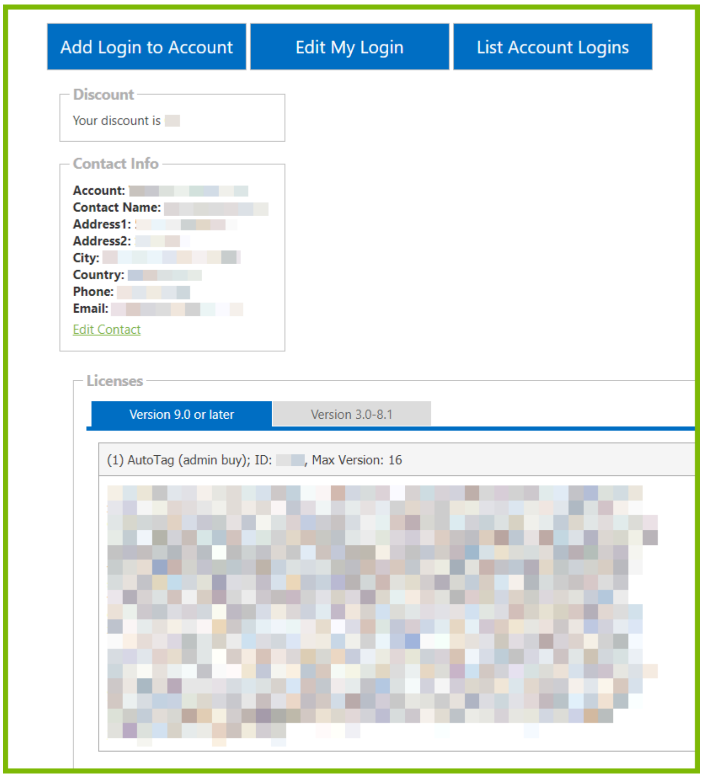
Password:* ••••••••••

Keep me logged in: ☐

Forgot your username or password?

Email:*

4. Copy your keys from your browser then paste them to the appropriate place for your product (see [How Do I Update My License Keys?]() below).



Discount

Your discount is [redacted]

Contact Info

Account: [redacted]
Contact Name: [redacted]
Address1: [redacted]
Address2: [redacted]
City: [redacted]
Country: [redacted]
Phone: [redacted]
Email: [redacted]

Edit Contact

Licenses

(1) AutoTag (admin buy); ID: [redacted], Max Version: 16

[redacted]

Keys are listed in two sections: version 9.x and higher and version 8.x and lower.

When Must I Update My License Keys?

When you purchase a maintenance agreement in conjunction with your Windward Studios product license, you are entitled to install all Support Releases for your version and earlier versions of the software, as long as your maintenance contract is current.

As a general rule, the license key you are assigned will work with the current version of a Windward Studios product; will also work for previous versions; but will *not* work for subsequent releases. However, if your maintenance contract is up to date, you simply retrieve a new license from the Windward store (see How Do I Retrieve My License Keys?).

If you have upgraded any of your Windward Studios products (a Report Designer or a Report Engine), you may receive a license error similar to this:

> **"This license is only valid up to version 15.*.*"**

If you see this error, your keys are only valid up to the version current at the time you downloaded them (in this case version 15). To fix this, login to the Windward Store and retrieve your current license keys.

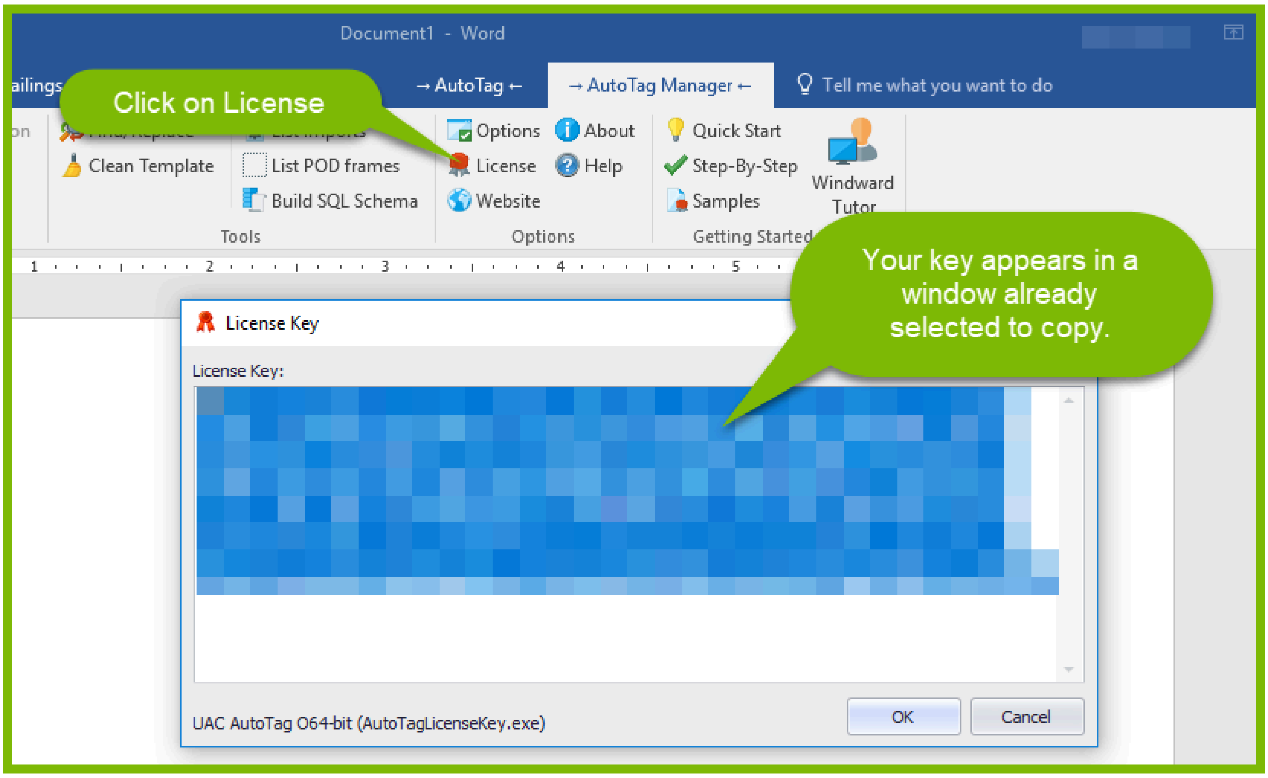
When we make any changes to your current keys, such as changing the length or date of your maintenance contract; adding new products; or changing license restrictions (e.g. extending a Report Engine license from two to four cores); you must go to the Store to download your updated keys.

How Do I Update My License Keys?

How to update your license keys varies depends upon the Windward Studios product you're licensing.

Licensing for Report Designer Office Edition

For the Report Designer, to see the key you have installed, open MS Word, Excel, or PowerPoint. In the ribbon, click the AutoTag Manager tab and click the License button. (You may see a prompt asking if you want to allow the program to make changes to your device - just click 'OK'.) A window will appear with your license key.



Click on License

Your key appears in a window already selected to copy.

License Key

License Key:

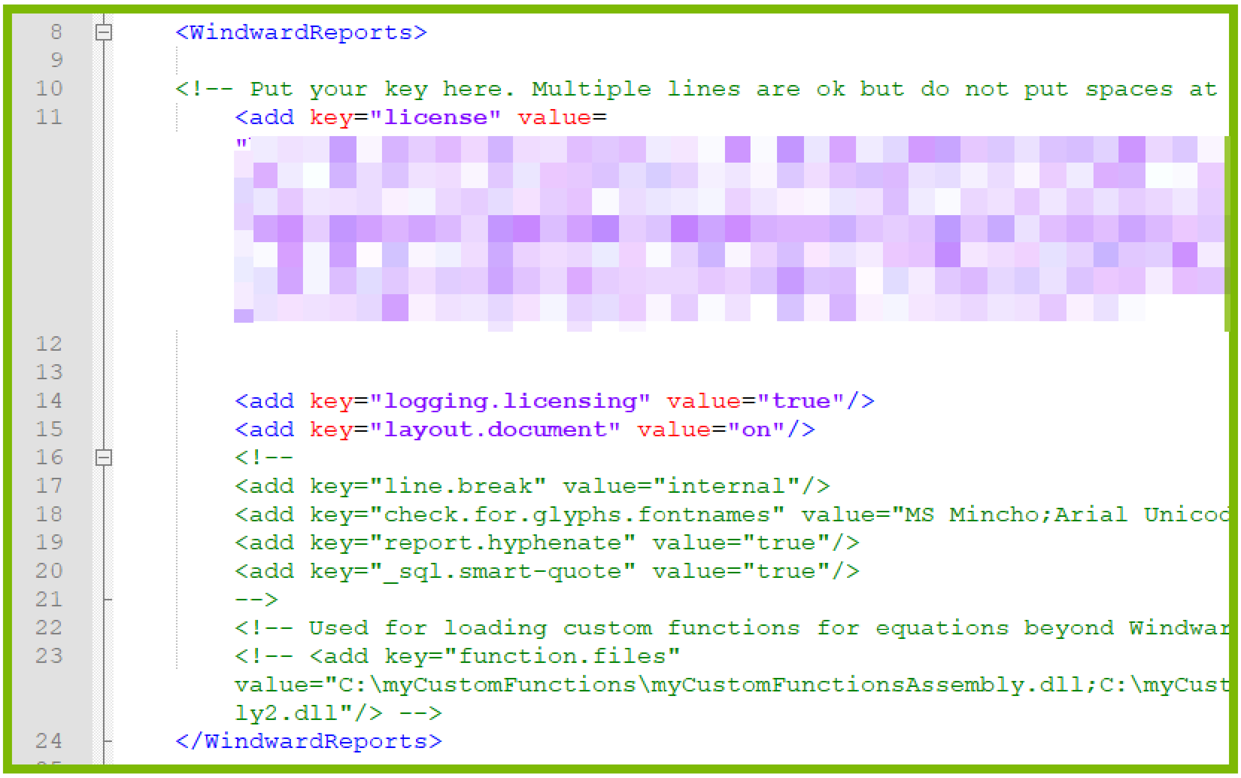
[redacted]

UAC AutoTag 064-bit (AutoTagLicenseKey.exe)

Licensing for .NET Report Engine

The license key must be visible to the Report Engine when it starts. Your Report Engine license key should be entered into

- the <app>.exe.config file (where <app> is the name of your embedding application)
- the Web.config file of your embedding ASP.NET application

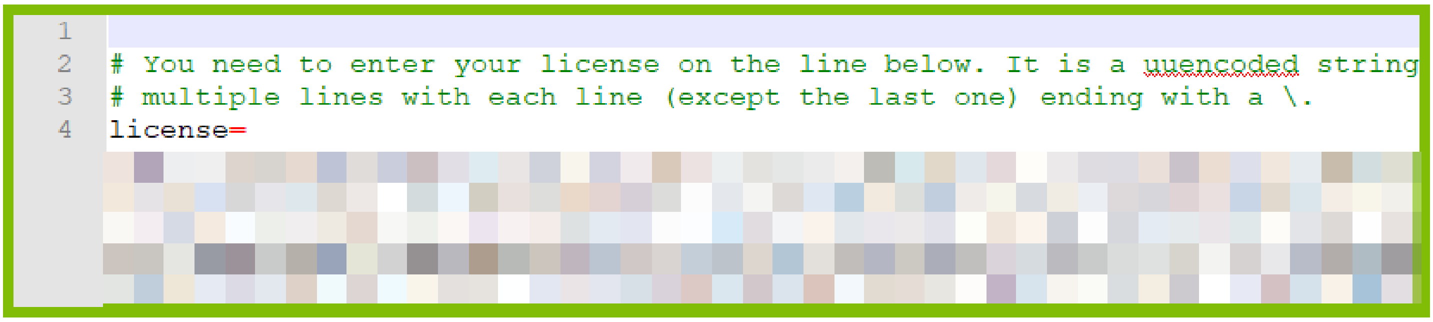


```
<WindwardReports>

<!-- Put your key here. Multiple lines are ok but do not put spaces at
    <add key="license" value=
    "
    [redacted]


    <add key="logging.licensing" value="true"/>
    <add key="layout.document" value="on"/>
    <!--
    <add key="line.break" value="internal"/>
    <add key="check.for.glyphs.fontnames" value="MS Mincho;Arial Unicod
    <add key="report.hyphenate" value="true"/>
    <add key="_sql.smart-quote" value="true"/>
    -->
    <!-- Used for loading custom functions for equations beyond Windwar
    <!-- <add key="function.files"
    value="C:\myCustomFunctions\myCustomFunctionsAssembly.dll;C:\myCust
    ly2.dll"/> -->
</WindwardReports>
```

You cannot use WindwardReports.dll.config. And the config file must be in the correct directory - the same as where the <app>.exe file is for applications and in the IIS website root directory for Web.config.

Licensing for Java Report Engine

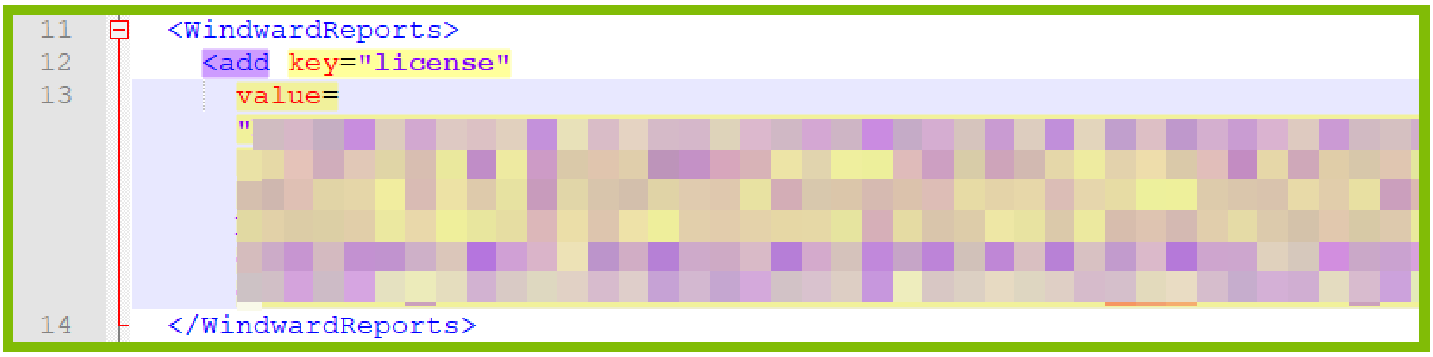
The key must be visible to the Report Engine when it starts. The Report Engine will look in the Java application properties for your application for the property ('license') with the license value. By default this is in the WindwardReports.properties file. However, you can use any appropriate means to insert the license key into your application's Java application properties.



```

# You need to enter your license on the line below. It is a uuencoded string
# multiple lines with each line (except the last one) ending with a \.
license=
[redacted]
```

Licensing for Report Engine for RESTful

The license key must be visible to the Report Engine when it starts. Your Report Engine license key should be entered into the Web.config file in your Report Engine IIS website's home directory.



```
<WindwardReports>
  <add key="license"
    value=
    "[redacted]
</WindwardReports>
```

Important License Information

- You must have a separate license for each system where a Windward Studios product is installed and executed.
- You don't need a client license for machines running a browser used to view generated output, only for the machine the output was generated.
- One license per machine is required.
- You must purchase support and upgrades for either all licenses or none.
- The "Standard Server" license runs an unlimited number of reports daily. A single license for a server allows it to be run on one machine only. It is not a license to install the product on an unlimited number of machines.
- Where the WindwardReports.jar or WindwardReports.dll file is located is irrelevant— a license is needed for each machine where a Windward Studios product is installed and executed.
- 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.
- Your Report Engine version should always be equal to or newer than the version of the Report Designer you're using. This is because newer versions of Report Designer may use newer features that older versions of the Report Engine don't recognize.

License Key Tips

Here are some tips on using Windward Studios licenses:

- When changing a key in a Report Engine, you must restart the application or service embedding that Report Engine for the new license key to take effect.
- In the Java Report Engine's WindwardReports.properties file, you can enter a multi-line key but you need a "\" (backslash) at the end of every line except the last line.

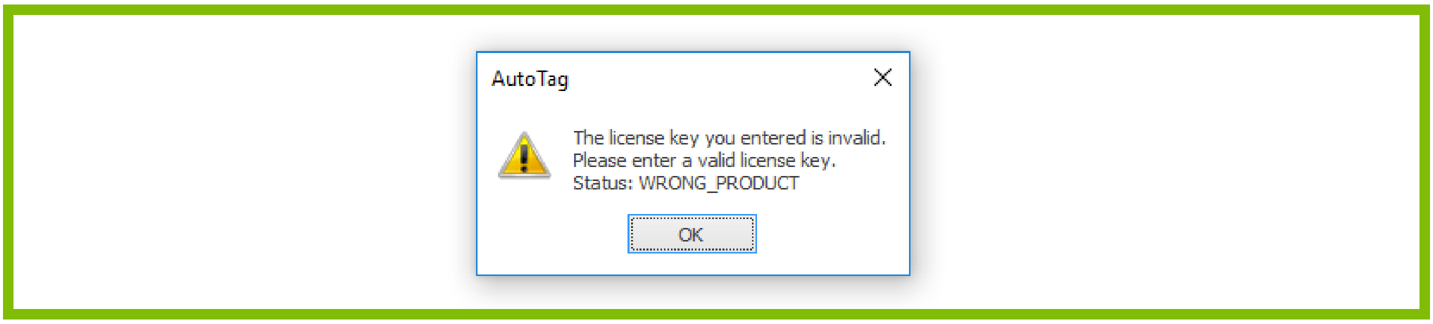
- In the .NET Report Engine's <app>.exe.config file, ensure the entire key is in double quotes, e.g. "BDjfsrtf".

License Key Exceptions

If a restriction of a license key (e.g. number of machines, number of cores, number of threads, etc.) is exceeded you will see these license key exceptions.

Wrong Product

If your Report Designer is given a license key for another Windward Studios product, this pop up will appear:



AutoTag

The license key you entered is invalid.
Please enter a valid license key.
Status: WRONG_PRODUCT

If your Reporting Engine is given a license key for another Windward Studios product, you will see an exception similar to this:

```
************************************************************************
* Exception occurred: Your license is for a different Windward product 1
************************************************************************
Error: Your license is for a different Windward product 1
```

Too Many Machines

If your Reporting Engine is running on more than the licensed number of machines, you will see an exception similar to this:

```
net.windward.util.LicenseException: Running on more machines than licensed for. This server:xxx.xxx.xxx.xxx; Running: 2, Licensed: 1 Addr: yyy.yyy.yyy.yyy (hostname1, hostname2)
```

Too Many Threads

If your Reporting Engine is being executed by more than the licensed number of threads simultaneously, you will see an exception similar to this:

net.windward.util.LicenseException: Running on more threads than licensed for. License good for maximum of 2 threads

Too Many Cores

If your Reporting Engine is running on a machine with more than the licensed number of cores, you will see an exception similar to this:

net.windward.util.LicenseException: License violation License violation - licensed for 4 cores (across all CPUs), this computer has 8 cores.

Developer License Exceptions

Developer licenses allow one thread at a time and a maximum of 250 reports a day. If the thread limit is exceeded you will see an exception similar to this:

License Error: Only allowed 1 request at a time using this server

If the 250-report-a-day limit is exceeded, you will see an exception similar to this:

```
****************************************************************
* only 250 reports a day in development & limited version servers *
****************************************************************
WARN (?:?) - only 250 reports a day in development & limited version servers.
WARN (?:?) - Error processing request from /xxx.xxx.xxx.xxx:yyy
net.windward.util.LicenseException: Surpassed the limited server daily limit
```

License Key Corruption

If you see a license exception similar to **Invalid license key (BAD_XML)**, your license key may have become corrupted while forwarding it in email.

If someone who received an HTML email from Windward Studios with the proper license key then forwards that key as a plain text email to another recipient, the email system may corrupt the uuencoded license key string in the original HTML email. Because of this, it is recommended to retrieve your keys from the Windward Store as described above. But if license keys must be forwarded via email, save the keys from the Windward Studios email to a text file, then forward the text file as an email attachment. (Windward Studios doesn't initially deliver keys using this method because the email may be blocked by a firewall.)

Internal Communication Protocols and Ports

The Windward Studios licensing mechanism communicates using both TCP and UDP network protocols on the ICANN-registered **port 1707**.

During the Report.Init() process the license key file is read and the following license key properties are broadcast:

- license key number
- total number of machines on which the key is valid to run
- total number of threads or cores per machine for which the license key is valid

A running total is kept of the following values, each of which are updated when a broadcast is received on a system where a Windward Studios product is running:

- current total number of machines on which the license key is used
- current number of threads or cores allocated for the reporting process on each machine where the license key is used

For additional details about how these requests operate refer to the License Copy Protection section below.

Licensing Restrictions

Watermarks

Report Engines licensed with a Developer or Test license will print one of the following watermarks at the bottom of the last page of output:

- "Report created by Windward Reports test version - www.windward.net"
- "Report created by Windward Reports Developer System - www.windward.net"
- "Created by demo version of Windward Reports - www.windward.net"

Password-protected Output

To prevent license abuse, test and development keys generate password-protected (read-only) output, where the password is not made public. This applies to output types DOCX, PPTX, XLSX and PDF.

Restricted by Computer Host Name or IP Address

Some licenses are restricted to running on specific computers by IP address and/or host name. If a license key has this restriction and is running on a computer that does not match the allowed systems, it will throw a license exception and list the disallowed IP address(es).

Restricted by Number of Calling Threads

Some licenses are restricted by the number of threads that can call a Report Engine simultaneously. If this license restriction is in effect, the licensing mechanism increments a counter each time a report is started (by calling ProcessReport.setupProcess) and decremented each time it completes (by calling ProcessReport.completeProcess() or ProcessReport.close()).

License keys also have the maximum number of cores restriction. If the system running a Windward Studio product has the same or fewer cores than the core limit in the license, then the number of threads limit is ignored. The number of threads limit only takes effect on systems that have more cores than the core limit in the license.

For example, if you have a license key for four cores and four threads, and the Report Engine is running on an eight-core machine, as long as no more than four threads call the Report Engine simultaneously, there is no license violation. But if a fifth thread calls the Report Engine while four threads that called the Report Engine are running, the fifth thread will fail with a license exception.

Windward Studios products themselves are single threaded. The thread count referred to in license keys is the number of threads that call a Report Engine simultaneously to generate output. This thread limit is counted across all virtual machines (VM), Java Virtual Machines (JVM), and .NET Common Language Runtimes (CLRs) on a physical server. If you have two VMs, one with three JVMs and the other with four CLRs, and all are running applications embedding Report Engines, all seven environments count against the total. If those environments each have four threads generating output, the total is 28 threads. *But the total is 28 only if all 28 threads are generating output at the same time.* If the Report Engines all generate output at different times (yes, unlikely – even with semaphores), then you only need a license that allows one thread.

Because the count is not decremented until a Report Engine's output is completed, if output is interrupted by an exception, or you decide to not complete it, you must call ProcessReport.close() to decrement the count. Otherwise that thread remains counted. If you abort a JVM while output is being generated, all other JVMs will retain the count for all Report Engines generating output.

License Copy Protection

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

All editions of the Report Designer and Report Engines have a copy protection mechanism that checks for license violations. The licensing mechanism in each product works the same but are independent of each other. In other words, the Report Designer checks for Report Designer license violations, but does not check Report Engine license violations. The Java Report Engine checks Java Report Engine license violations but does not check for .NET Report Engine, etc.

In each case, when the product starts, it asks all other systems running using the same license to identify themselves. Each system that is already running will respond to that request, identifying itself.

A license specifies the number of machines for which the license is valid. If that number of machines responds to the Report Engine that just started, then the license is in full use and the

Report Engine just starting will fail with a license exception and not generate output. A license exception will not be thrown by any of the existing running Report Engines.

The Report Designer starts when Office (Word, Excel or PowerPoint) starts. It ends when all copies of Office on a user's workstation exit.

A Report Engine starts when ProcessReport.init() is called or the first time output is generated. It ends when the JVM or CLR which called it exits. So if ProcessReport.init() is called and no output is generated for two weeks, but the VM is still running, that system is still using the license and will be counted against that license. If ProcessReport.Init() is *not* called and the first output is generated two weeks after the Report Engine starts, *that* is when the Report Engine will determine if it complies with its license restrictions for its server.

One license is good for a single VM running under a single set of credentials. If you have two VMs on a single physical machine, they require two licenses. If on a single VM you run two applications embedding a Report Engine under two sets of credentials, they require two licenses.

The licensing mechanism is designed so licenses can move between machines. Keep in mind that if you want to stop using a license on a specific server, you must stop all VMs on that server that called a Report Engine.

Each license has a unique ID. Two systems have the same license if and only if they have the same ID and the same restrictions such as product, number of reports, etc. If you purchased three server licenses, and then later two server licenses, you probably have two different licenses and each will have a different ID and will only check for use of itself. In this case, if you would prefer a single license good for five servers (or any number of valid licenses), contact us and we can swap those licenses for a single license good for your five servers.

If a Report Engine running within a VM fails with a license exception, it will continue to do so on subsequent calls even if other Report Engines have decremented the machine count by exiting. *The VM must be restarted for the Report Engine running within it to try again.*

The license exception is a runtime exception and therefore does not require there be try/catch blocks for it. This decision was made because a license exception will only be thrown when the Report Engine starts, and if there is a license problem, it will always be thrown. There are a large number of methods that can throw this exception, so it would have required additional try/catch blocks around almost all calls to Report Engine APIs. While your program does not require try/catch blocks for license exceptions, you may implement them if you wish.

The class net.windward.tools.ListServers (in the Java Report Engine Utilities jar) will list all machines running the Report Engine by IP address. In the Report Designer the About box lists the IP addresses of all machines running Report Designer. The utility finds all systems regardless of license ID and settings and only reports IP address and host name.

Getting Started with .NET Report Engine

This article gives you the basic information you need to install and test .NET Report Engine.

Requirements

- .NET Framework 4.5 or higher (.NET Framework 4.6.1 for v16)
- J# Redistributable Package 2.0 REQUIRED ONLY FOR VERSIONS PRIOR TO 12.5 for x86, x64 or IA64
- A .NET Report Engine license key (request a trial license key if you don't have one)

Installing .NET Report Engine

ℹ For detailed instructions, see How Do I Install the .NET Report Engine?

Installation

Go to Windward Studios Downloads and click on the link .NET Engine Installer – (EXE)

Once the download is complete, double-click on the installer. Follow the steps of the Installation Wizard. You may enter your license key during the installation or afterward.

Entering Your License Key

All configuration information goes in the App.exe.config file for the application calling the .NET Report Engine. By default, an example WindwardReports.dll.config file is located in C:\Program Files (x86)\Windward Studios\Windward .NET Engine. If you chose not to enter your license key during the installation, after the installation you must copy and paste your license key into to your App.exe.config.

As with all configuration entries, use the standard .NET '<add key="xxx" value="xxx"/>' format when entering your license key. An example is provided in the WindwardReports.dll.config file, and is also shown below for your reference (but is not a valid key).

```
<add key="license" value="cHtzLjSa8TwnWysGZbDtvf1YB8eKHSxNWb54T0m MoG+d/
qztTfI9xdwyRx98vGk2dpjpPHYSRQQoZmE0J8o7NWYSRQWpZ220dhDT
wnWysGZbDtvj2A7T53uzsdJzqxHO1Zm61kZukHZqt/NSs5aaaThis/is/not/a/valid/key==="/>
```

The license key string can be entered as a single line, or it can be entered on multiple lines as long as each line is just the license key (no spaces before or after).

Testing the Installation

The .NET Report Engine test directory (by default C:\Program Files (x86)\Windward Studios\ Windward .NET Engine\test) contains a test template and data source to verify your installation is working correctly. Navigate to that directory and execute run.bat. If the output file "testreport.pdf" is created, then .NET Report Engine is installed and running.

What's Next

You have now verified your .NET Engine installation. You have a few choices for how to proceed from here:

- Open the installed Catapult sample software application and look through the sample projects. A wide variety are installed with the .NET Report Engine to get you started.
- Just need to the see the code? See the .NET Report Engine API Reference.

What's New in Version 16.1?

Here is a summary of new features in all Windward Studios products for v16.1. There are also links to What's New in previous versions.

Click here to see information about new features in previous versions.

System Requirements

You must have the following installed in order for Windward Designer version 16.1 to run:

- Windows
- Office
- .Net 4.6.1 or later

All Windward Products

What Changed?	Description
Time Zones Conversion	A new Windward macro has been added to help with converting dates and times from UTC time to the local time zone. Use TOLOCALTIME() to convert to your computer time zone. (From a customer new feature request.)
‘Shrink to Fit’ in Excel	The ‘Shrink to Fit’ cell property is now output to XLSX output. This will shrink the contents of a cell until it fits the defined cell size. (From a customer new feature request.)

Report Designer for Office Edition

What Changed?	Description
Reorganized Ribbon	The ribbon menus have been reorganized and consolidated to improve the Report Template design workflow.
Boolean Conditional Wizard	The new Boolean Conditional Wizard helps

What Changed?	Description
	you create complex conditional statements. Before, conditional statements could only be written manually. Now they can also be built using our intuitive Wizard interface.
Data Bin Search	The Data Bin can now be searched to find a table, column, node or other piece of data without having to scroll through all of it. (From a customer new feature request.)
Stored Procedure Wizard	The new Stored Procedure Wizard helps you create a query for a stored procedure. This works for all Tag types that are connected to a SQL-based data source (Microsoft SQL Server, Oracle, MySQL, PostgreSQL, DB2 or ODBC).

.NET Report Engine and Java Report Engine

What Changed?	Description
New Output Format	PostScript has been added as an output format. This format is commonly used with printers and printing companies. (From a customer new feature request.)

Report Designer for Office Edition, .NET Report Engine and Java Report Engine

What Changed?	Description
New and Improved Datasets	Datasets have been rewritten from scratch to be more powerful and easier to use. Unlike our previous Datasets (which required being saved in a POD file to utilize in the Report Engines), new Datasets can be used like all other Datasources. This makes Datasets easier to manage and deploy in an application.

Previous Versions

Here are links to What's New in previous versions:

[Version 15]

[Version 14]

[Version 13]

[Version 12]

[Version 11]

[Version 10]

[Version 9]

[Version 8]

[Version 7]

[Version 6]

[Version 5]

[Version 4]

FAQs

How Do I Enable Error Handling in .NET Report Engine?

The .NET Report Engine API includes a facility for collecting and displaying warnings and errors that occur while it is generating output. This article describes what the verify and error handling features do, with some examples of the warnings and errors they capture.

This article also describes the API calls used to enable error handling, and provides example code based on one of our Catapult C# example command-line applications.

The relevant API calls are the TrackErrors property and GetErrorInfo().

What Does It Do?

The *verify* feature provides you with actionable information on non-fatal errors, warnings, and other issues contained within a Report Template that occur when generating output with .NET Report Engine.

Error handling is a Tag property that can be set to catch errors and exceptions that occur in a Tag's query. Depending on the **error-handling** Tag property setting, otherwise fatal errors can be captured and ignored so output can be completed despite the error. For more details about the **error-handling** Tag property, see the Out Tag Reference.

Example Warnings and Errors

Here are some examples of the warnings and errors captured when the verify and error handling features are used:

Data Type Issues

warning: ‘Buchanan’ is String while NUMBER expected: <wr:out select=’SELECT dbo.Emplayees.LastName FROM dbo.Employees’ type=’NUMBER’ nickname='TypeError’ error-handling=‘Ignore type error;Ignore formatting error;Ignore select error;Node must exist’ datasource=’MSSQL’/>

warning: ‘Buchanan’ is String while NUMBER expected: <wrout select=’SELECT dbo.Emplayees.LastName FROM dbo.Employees’ type=’NUMBER’ nickname='TypeError’ error-handling=‘Ignore type error;Ignore formatting error;Ignore select error;Node must exist’ datasource=’MSSQL’/>

Formatting Issues

warning: format(Buchanan,) is not allowed: <wr:out select=’SELECT dbo.Employees.LastName FROM dbo.Employees’ type=‘NUMBER' nickname=TypeError’ error-handling= Ignore type error;Ignore formatting error;Iqnore select error;Node must exist’ datasource=’MSSQL’/>

warning: Could not parse: Adam: <wr:out select='/Formats-to-test/Author/Firstname’ type=‘DATE’ format=‘categary-date-type:0-format:m/d/yyyy’; nickname=’FORMAT_PROBLEM' error-handling=‘Ignore formatting error;Ignore select error’ datasource='TestingDatabase'/>

warning: format(Buchanan,) is not allowed: <wr:out select=’SELECT dbo.Employees.LastName FROM dbo.Employees’ type=’NUMBER’ nickname=’TypeError’ error-handling=’Ignore type error;Ignore formatting error;Iqnore select error,Node must exist’ datasource=’MSSQL’/>

warning: Could not parse: Adam: <wr:out select='/Formats-to-test/Author/Firstname’ type=‘DATE’ format=‘categary-date-type:0-format:m/d/yyyy’; nickname=’FORMAT_PROBLEM' error-handling=‘Ignore formatting error;Ignore select error’ datasource='TestingDatabase'/>

Bad Select Statement

warning: Invalid column name ‘dbo’. Final select: select dbo from dbo.Employees: <wr:out select=‘select dbo from dbo.Employees’ nickname=‘DNE1' error-handling=’Ignore select error’ datasource=‘MSSQL’/>

warning: Invalid column name ‘dbo’. Final select: select dbo from dbo.Employees: <wr:out select=‘select dbo from dbo.Employees’ nickname=‘DNE1' error-handling=’Ignore select error’ datasource=‘MSSQL’/>

Node Does Not Exist

error: The tag returned no nodes: <wr:out select='/Formats-to-test/Nada’ nickname=‘Nadal' error-handling='Node must exist’ datasource=’TestingDatabase’/>

error: The tag returned no nodes: <wr:out select='/Formats-to-test/Nada’ nickname=‘Nadal' error-handling='Node must exist’ datasource=TestingDatabase’/>

Node Is Null

warning: The tag is empty: <wr:out select=‘select ShipRegion from dbo.Orders where OrderID=10248’ nickname=‘[ShipRegion]’ error-handling=‘Node must not return NULL’ datasource=‘MSSQL’/>

warning: The tag is empty: <wr:out select=‘select ShipRegion from dbo.Orders where OrderID=10248’ nickname=‘[ShipRegion]’ error-handling=‘Node must not return NULL’ datasource=‘MSSQL’/>

Verification Issues

warning: Tag uses undefined data source ‘OData’: <wr:set select=‘Employees?$select=Nada&amp;$top=1' var='varName3’ nickname=‘DNE8' error-handling=‘Ignore select error’ datasource='OData’/>

warning: Tag uses undefined data source ‘OData’: <wr:set select=‘Employees?$select=Nada&amp;$top=1' var='varName3’ nickname=‘DNE' error-handling=‘Ignore select error’ datasource='OData'/>

TrackErrors

TrackErrors is used to enable error handling in your .NET Report Engine application:

```
/// Enable or disable the error handling and verify features. If they are enabled you
can call GetErrorInfo() to
/// get a list of errors encountered during the report generation.
/// By default these features are turned off (disabled).
/// Use a combination of ERROR_HANDLING bits to set a value for the property (see below)
public int TrackErrors{...}
```

TrackErrors is declared in the net.windward.xmlreport.errorhandling namespace.

TrackErrors Flags

Here are the values to which the TrackErrors property can be set:

```
/// The settings for the TrackErrors property.
/// These map to ProcessReportApi.ERROR_HANDLING_ values.
[Flags]
public enum ERROR_HANDLING
{
/// <summary>
/// Handle errors as indicated by a tag properties.
/// </summary>
TRACK_ERRORS = 0x01,

/// <summary>
/// Perform verification during a report generation.
/// </summary>
VERIFY = 0x02,

/// <summary>
/// The bitmask that turns the complete error handling functionality off.
/// </summary>
NONE = 0x00,

/// <summary>
/// The bitmask that turns the complete error handling functionality on.
/// </summary>
ALL = 0x03
}
```

GetErrorInfo()

GetErrorInfo() is used to process any warnings and errors that occur while .NET Report Engine is running, after TrackErrors is used to turn on error handling:

```
/// Get an object containing a set of errors encountered during the report generation.
To enable the errors tracking
/// you have to turn this feature on by setting TrackErrors.
/// returns an instance of the ErrorInfo implementation.</returns>
public ErrorInfo GetErrorInfo() {...}
```

GetErrorInfo() is also declared in the net.windward.xmlreport.errorhandling namespace.

RunReportXml.cs Example

Here is the source file RunReportXml.cs from our Catapult C# command-line examples, with error handling added and NEW COMMENTS IN ALL CAPS:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using net.windward.api.csharp;
using WindwardReportsDrivers.net.windward.datasource;
using WindwardInterfaces.net.windward.api.csharp;
using System.IO;
using net.windward.xmlreport.errorhandling; // USED FOR ERROR
HANDLING


namespace RunReportXml
{
    class RunReportXml
    {
        static void Main(string[] args)
        {
            // Initialize the engine
            Report.Init();

            // Open template file and create output file
            FileStream template = File.OpenRead("../../../Samples/Windward Trucking 2 -
Template.docx");
            FileStream output = File.Create("../../../Samples/Xml Report.pdf");
            // Create report process
            Report myReport = new ReportPdf(template, output);

            // Open a data object to connect to our xml file
            string url = Path.GetFullPath("../../../Samples/Windward Trucking 2 - Data.
xml");
            string xsd = null;
            IReportDataSource data = new SaxonDataSourceImpl(string.Format("Url={0}",
url), xsd);

                // TURN ON ERROR HANDLING
            myReport.TrackErrors = (int)Report.ERROR_HANDLING.ALL;

            // Run the report process
            myReport.ProcessSetup();

            // The second parameter is "" to tell the process that our data is the
default data source
            myReport.ProcessData(data, "sax");
            myReport.ProcessComplete();

                // PROCESS ANY ERRORS
            ErrorInfo errors = myReport.GetErrorInfo();
            if (errors.hasErrors())
            {
                java.util.List list = errors.getErrors();
                for (int i = 0; i < list.size(); ++i)
                    Console.Out.WriteLine(((Issue)list.get(i)).getMessage());
            }

            // Close out of our template file and output
            data.Close();
            output.Close();
            template.Close();

            // Opens the finished report
            string fullPath = Path.GetFullPath("../../../Samples/Xml Report.pdf");
            System.Diagnostics.Process.Start(fullPath);
        }
    }
}
```

How Do I Inject My License Key Into the .NET Engine?

In 16.3.0 a method is added to the API to specify a license key for the engine at run time.

A new method is added to the .NET API in version 16.3.0:

```
Report.ApplyLicenseKey(String licenseKey)
```

Calling this method before calling `Report.Init()` will use the specified license key instead of whatever key is specified in the app.config file. Calling this method after the object instantiation will not apply the new license.

How Do I Insert Input Parameters in a v16 .NET Engine Application?

In v16.1, the Input Parameters have been moved from the data source object to the report object. Instead of setting IReportDataSource.Map, please set Report.Parameters.

For more details, see the [v16 .NET Engine API Reference](#).

Here is an example code snippet for .NET Engine v16 and later.

```
using (AdoDataSourceImpl datasource = new AdoDataSourceImpl("System.Data.SqlClient",
dbConnectionString))
{
    report.ProcessSetup();

    // set the variables to provide list results for each var
    report.Parameters = adHocVariables;

    // run the datasource
    report.ProcessData(datasource, connTypeAndName);

}
```

How Do I Install .NET Report Engine?

This article gives detailed instructions for installing the .NET Report Engine.

Requirements

- .NET Framework 4.5 or higher (.NET Framework 4.6.1 or higher for v16)
- J# Redistributable Package 2.0 REQUIRED ONLY FOR VERSIONS PRIOR TO 12.5 for x86, x64 or IA64
- A .NET Report Engine license key (request a trial license key if you don't have one)

Installation

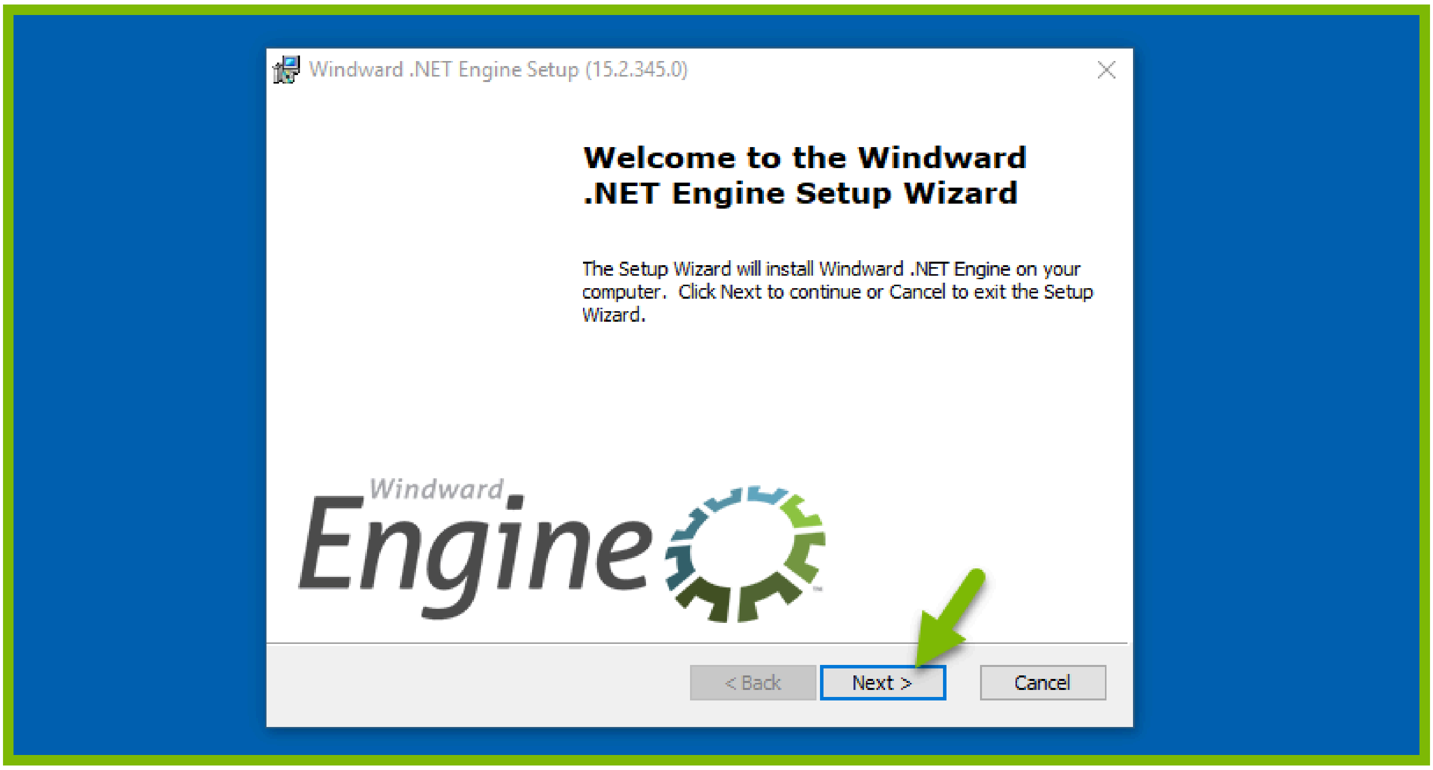
Download

Go to Windward Studios Downloads and click on the link .NET Engine Installer – (EXE).

Install

Once the download is complete, double-click on the installer.

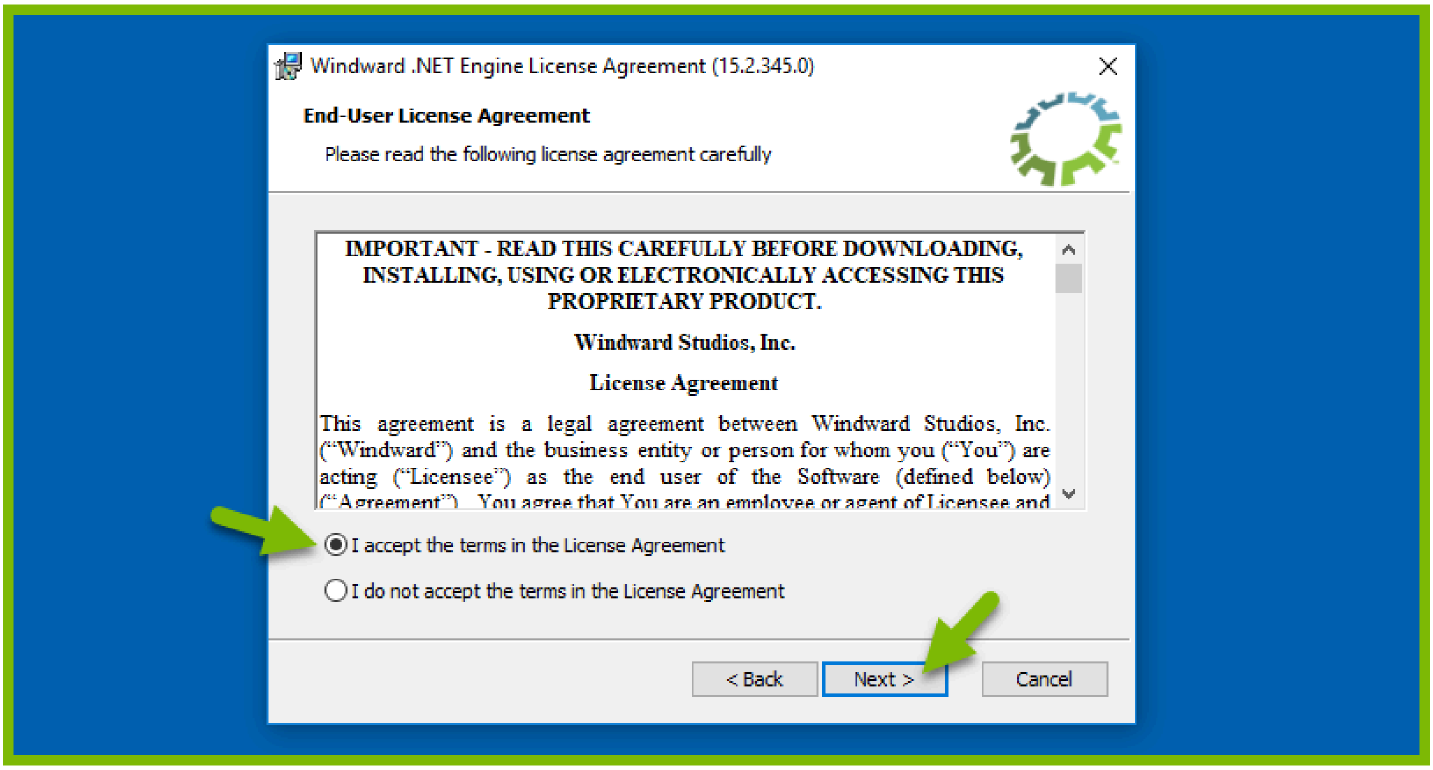
The Installation Wizard will open; click on Next.



Welcome to the Windward .NET Engine Setup Wizard

The Setup Wizard will install Windward .NET Engine on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Accept the license agreement, and click on Next.



End-User License Agreement

Please read the following license agreement carefully

IMPORTANT - READ THIS CAREFULLY BEFORE DOWNLOADING, INSTALLING, USING OR ELECTRONICALLY ACCESSING THIS PROPRIETARY PRODUCT.

Windward Studios, Inc.

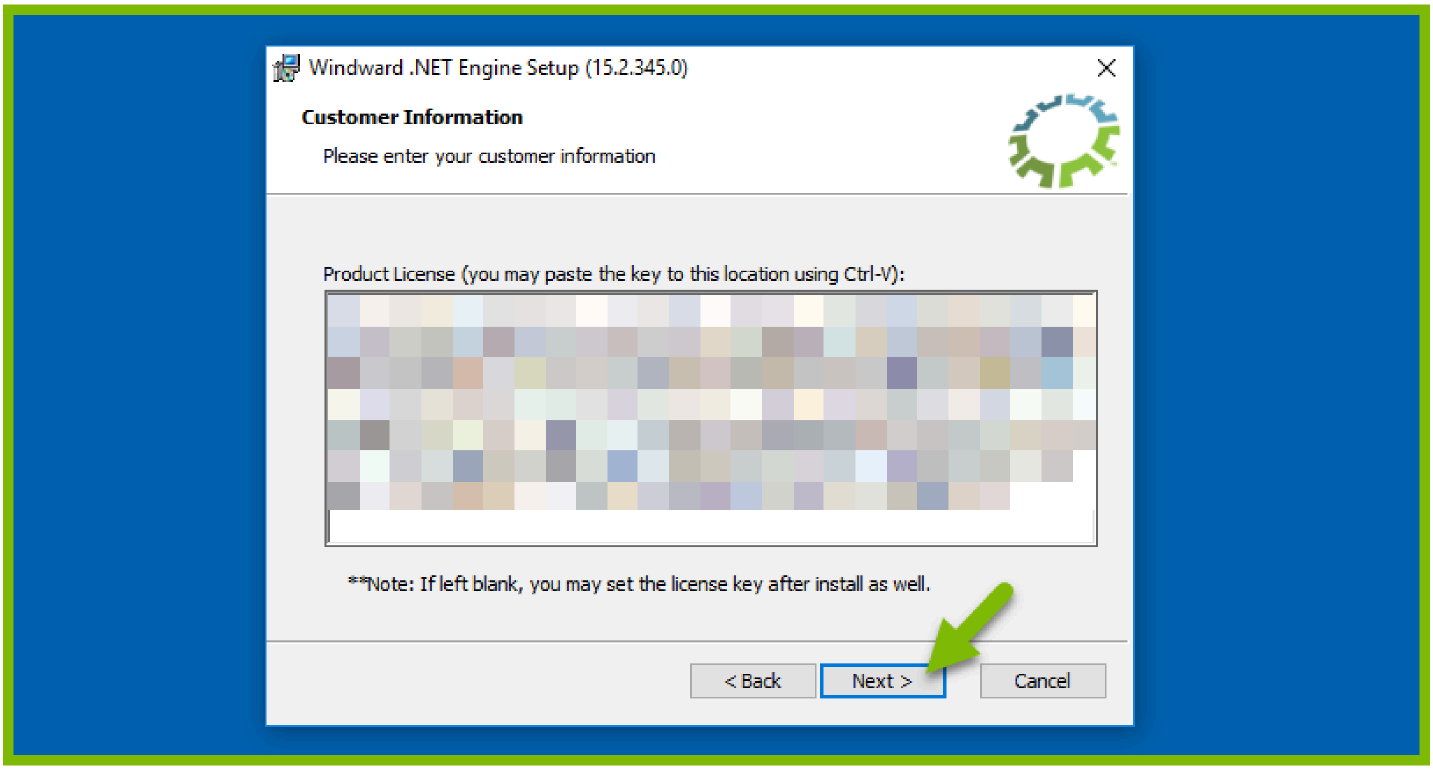
License Agreement

This agreement is a legal agreement between Windward Studios, Inc. (“Windward”) and the business entity or person for whom you (“You”) are acting (“Licensee”) as the end user of the Software (defined below) (“Agreement”). You agree that You are an employee or agent of Licensee and

● I accept the terms in the License Agreement

○ I do not accept the terms in the License Agreement

Enter your license key and click on Next. You may also enter your license key after installation.



Customer Information

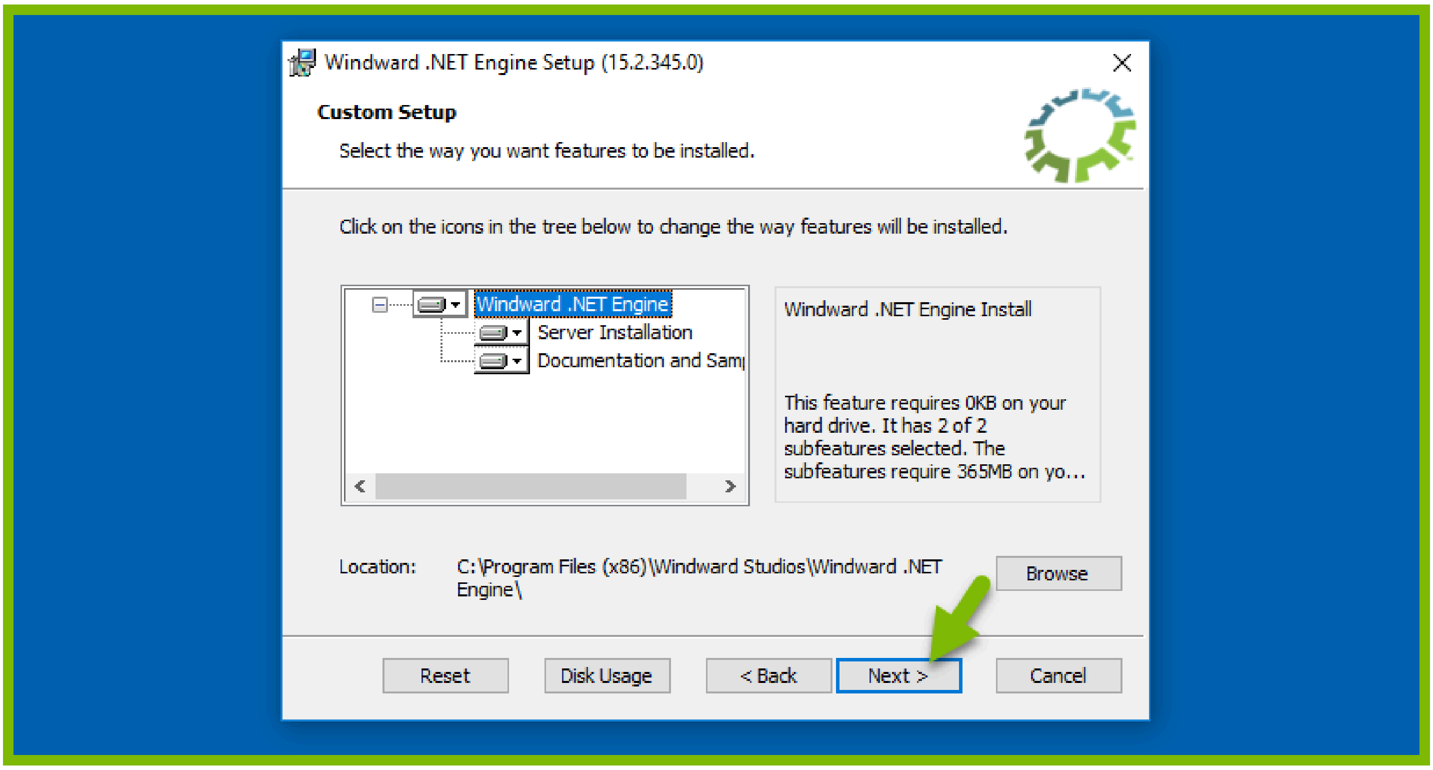
Please enter your customer information

Product License (you may paste the key to this location using Ctrl-V):

[redacted]

**Note: If left blank, you may set the license key after install as well.

By default, all necessary files are installed. Click on Next to continue.



Custom Setup

Select the way you want features to be installed.

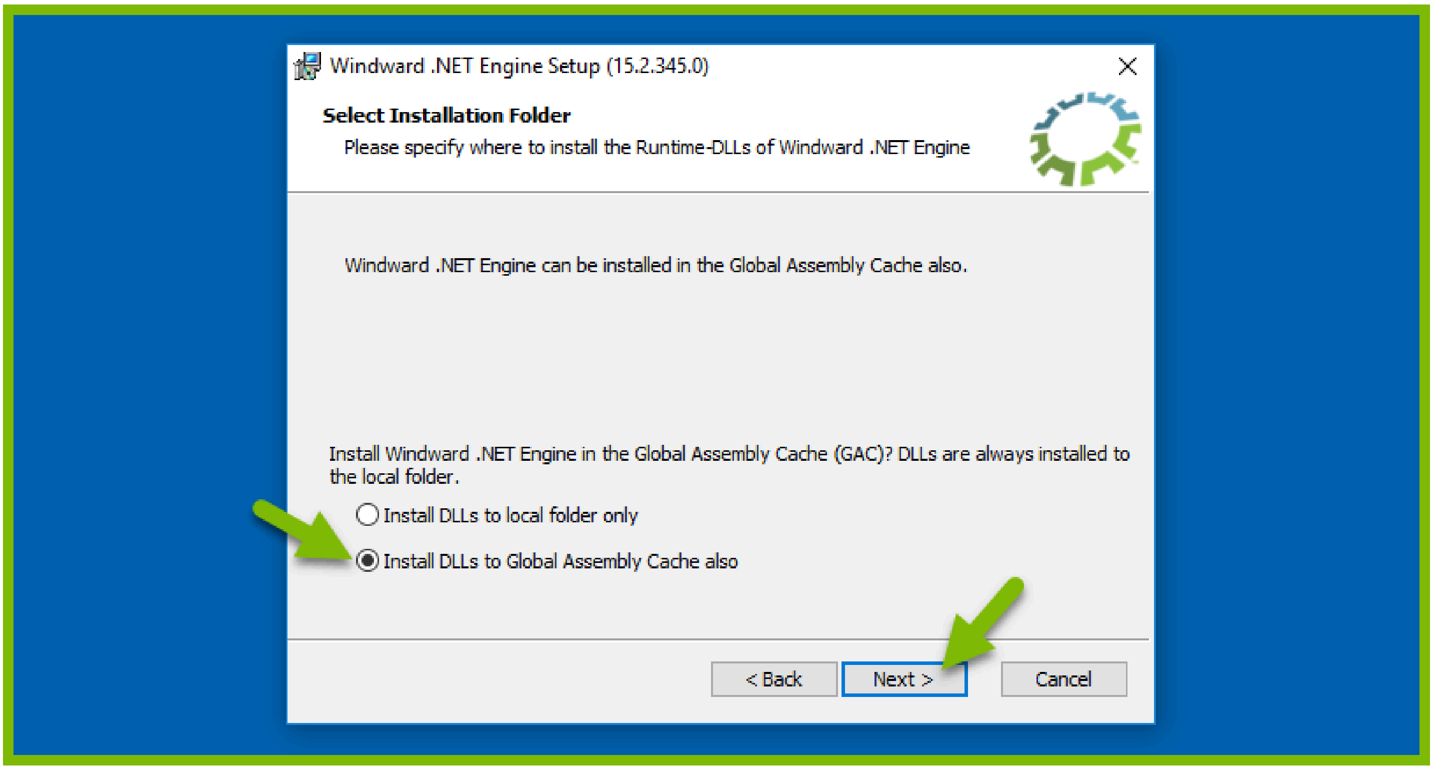
Click on the icons in the tree below to change the way features will be installed.

Windward .NET Engine Install

This feature requires 0KB on your hard drive. It has 2 of 2 subfeatures selected. The subfeatures require 365MB on yo...

Location: C:\Program Files (x86)\Windward Studios\Windward .NET Engine\

Select whether to install .NET Report Engine into the Global Assembly Cache; click on Next to continue.



Windward .NET Engine Setup (15.2.345.0)

Select Installation Folder

Please specify where to install the Runtime-DLLs of Windward .NET Engine

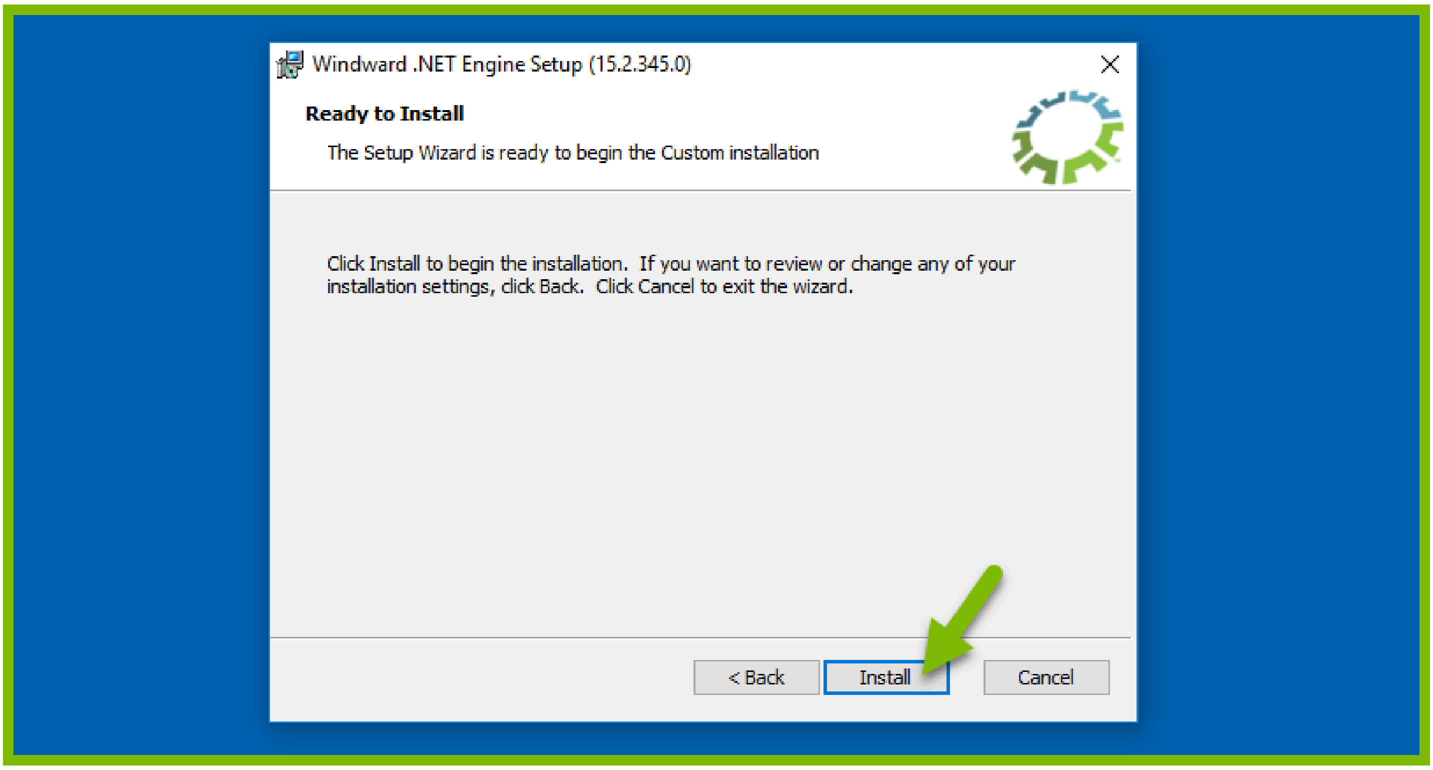
Windward .NET Engine can be installed in the Global Assembly Cache also.

Install Windward .NET Engine in the Global Assembly Cache (GAC)? DLLs are always installed to the local folder.

○ Install DLLs to local folder only

● Install DLLs to Global Assembly Cache also

Click on Install to begin the installation.



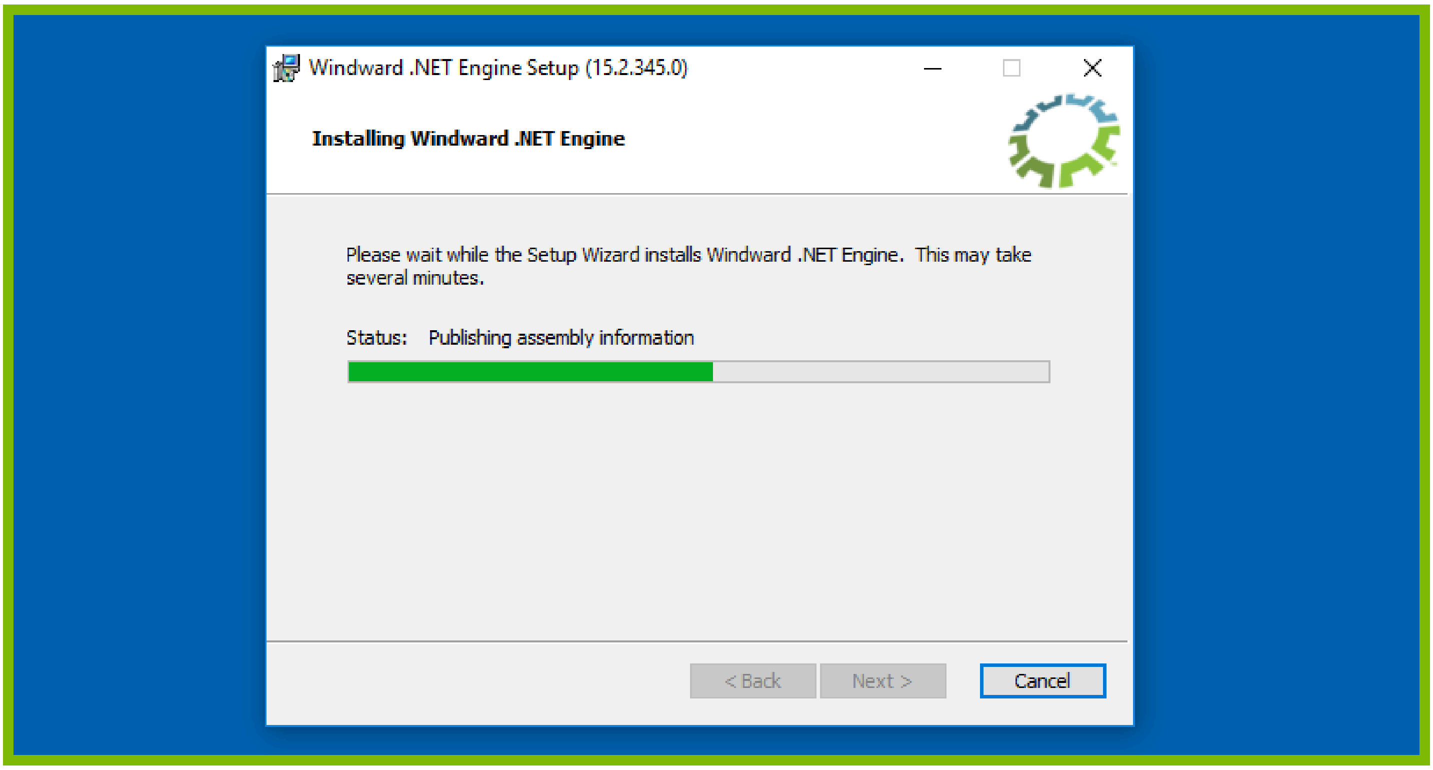
Windward .NET Engine Setup (15.2.345.0)

Ready to Install

The Setup Wizard is ready to begin the Custom installation

Click Install to begin the installation. If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

After the installation is complete, information about .NET Report Engine appears. Click on Next.

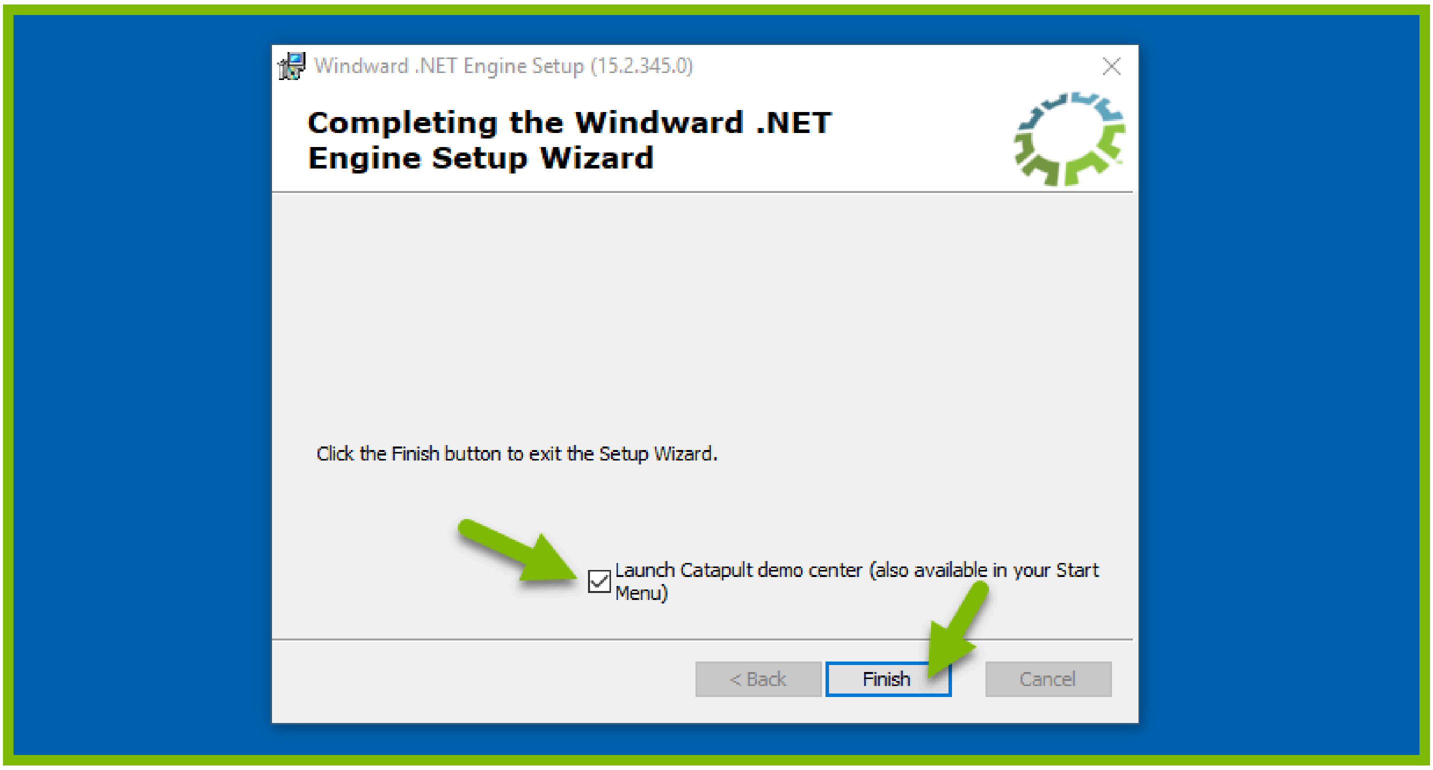


Installing Windward .NET Engine

Please wait while the Setup Wizard installs Windward .NET Engine. This may take several minutes.

Status: Publishing assembly information

Ensure the "Launch Catapult..." box at the bottom of the window is checked on to open the Catapult Help Menu. Click on Finish.



Completing the Windward .NET Engine Setup Wizard

Click the Finish button to exit the Setup Wizard.

☑ Launch Catapult demo center (also available in your Start Menu)

Validation

1. Open the installation directory of your .NET Report Engine, by default "C:\Program Files (x86)\Windward Studios\Windward .NET Engine".
2. Open the “test” folder.
3. Run the run.bat contained in the folder to run the test sample.
4. If you have not yet entered your license key, you must now add it to the RunReport.exe.config file.

The output file testreport.pdf file should appear in the folder. If it does, then you've successfully installed the .NET Report Engine.

How Do I Use Clean Templates For .NET Engine?

CleanTemplates.exe is a batch tool that installs with the Engine (in, e.g., C:\Program Files (x86)\Windward Studios\Windward .NET Engine) which allows you to clean a group of templates in order to convert them to a newer version of Windward.

Details

Always make a back up copy of your templates before using this tool.

This will update all Tags to the mode desired, adjust selects to match the new auto-detect of select/value/evaluate, and remove the quotes on '${var}' if you have switched to set parameter (suggested).

The template file(s) is a path where the filename component can have wildcards. So it can be C:\templates\*.rtf. The output directory is the directory to place the cleaned templates in. It can be the same directory only if the file extension is changing, i.e. you cannot overwrite the template with a cleaned template.

If you do not list a data source the selects against that data source will not be changed.

usage [options] template_file(s) output_directory.

Options are:

```
  -ps Substitution parameter mode.
  -pp Parameter Plus parameter mode. This is the default.
  -po Parameter Only parameter mode.
  -sq Smart quote OFF - remove quotes around ${vars}. The default is ON. Must be before
-sql/xml.
  -ru Remove Unused styles, etc. OFF. The default is ON.
  -tt Text tags.
  -tf Field tags (macro in Excel).
  -tp Field Plus tags (macro in Excel). This is the default.
  -tc Content Control tags (macro in Excel).
  -v9 Use this option if the template you are cleaning was created in version 9 or
earlier of AutoTag. Do not use this option if the template was created in version 10 or
later. Setting/not setting this parameter correctly is very important. (What this does
is tell clean template if the tags use select/evaluate/value (version 9 and earlier) or
```

```
the tags use select only (version 10 and later.)
```

Data sources are:

```
  -sql:name
  -xml:name
```

From the Command Line

Here's and example of what you might type to make adjustments from the command line: CleanTemplate.exe -ps -pp -po c:\full\path\to\template.docx c:\full\path\to\template1.docx c:\full\path\to\template2.docx c:\outputDir

How Do I Use Microsoft Office to Generate PDF Output with .NET Report Engine?

Because of vagueness in the PDF specification, using Microsoft Office to generate PDF output from report templates rather than the Windward PDF renderer can sometimes improve the fidelity of the output.

To allow the flexibility of using Office to generate PDF output, Report Designer and the Report Engines have a toggle to turn on using the Office PDF renderer.

The use.external.output.builder Property

To turn on the use Office for PDF output feature in .NET Report Engine, set the external.output.builder property in your application configuration file:

```
…

<WindwardReports>
…
   <add key="use.external.output.builder" value = "on"/>
…
</WindwardReports>

…
```

Use Office to Generate PDF Output with Report Designer

See How Do I Use Microsoft Office to Generate PDF Output in Report Designer?

Use Office to Generate PDF Output with the Java Report Engine

See How Do I Use Office to Generate PDF Output with Java Report Engine?

Use Office to Generate PDF Output with the Report Engine for RESTful

See **[How Do I Use Office to Generate PDF Output with Report Engine for RESTful]**

Subscription License FAQ

In this article we answer some common questions about the new Windward subscription licensing model.

When does Designer contact the License Server?

When it starts, each time it processes a template, and when it exits.

What is meant by the Designer starting/exiting?

When you start or exit Microsoft Word, Excel or PowerPoint. Even if you never use the Designer in Office, running Office is running the Designer.

When does Designer or a Processing Engine contact the License Server?

At the start of processing a template to verify that your license is active. And at the end of processing a template to update the License Server with the number of pages generated.

Can the License Server be unavailable?

The License Server is globally redundant running on Azure data-centers in the U.S., Europe and Australia. It is highly unlikely that all three regions would go down at the same time.

What if my connection to the Internet is down?

Then you cannot process templates. If the Designer has already started you can continue to design templates, but you cannot generate test output.

ℹ The PRO subscription (not scale out) can run for awhile without a connection to the license server.

How can I test my connection to the License Server?

Go to https://license.windwardscout.net/about and you will see a status message from the License Server if everything is OK. This does not test your license key, just that your computer can communicate with the License Server and the License Server is running properly.

With multiple servers (and databases), and multiple requests to process templates, can the data Windward has be out of date?

Yes, but to the advantage of our customers. If you process a template that hits or exceeds your scale out or overages limit, but that information is not yet written to the database, then you can process an additional template. When the additional numbers are written to the database, you are now over the maximum charge you set. In this case your bill will be the maximum and Windward throws away the additional charges.

What data do the Designer and Processing Engine pass to the License Server?

It passes your license key, the username the program is running under, the version of the program, MAC addresses for network interfaces on your machine, and the local IP address of the computer. On completion of processing a template it passes the number of pages generated. For a scale-out server it may pass the datetime of previous template processed.

Does the License Server see my template and/or data?

No.

How long is the License Server down when you do an upgrade?

We have a staging environment setup that we first deploy to. It is then swapped into production. Microsoft implements this swap into production with a "Virtual IP Swap." In effect, there will be no outages when we update the License Server. Even with no outage, we will notify customers before any update.

How long does it take for a license to be released, so I can give it to another user, or move it to another server?

For the Processing Engines, a subscription is released no more than 24 hours after it last contacted the License Server (see above).

How does fail-over work with Windward Pro licenses?

Windward has a lenient policy for users to fail-over Engines. Read more here.

How to Access Tags through .NET API Call?

This Article outlines how to extract tag data from a report object in the Windward .NET Engine using Report.getTagTree() API call

This functionality was added in version 20.0.0

In order to get the xml tag tree, use the following method:

```
using (FileStream output_file = new FileStream("[PATH TO XML OUTPUT FILE]", FileMode.
OpenOrCreate))
{
        Report.GetTagTree(input_template, output_file, license_key);
}
```

The input templates must be one of the following:

- .docx
- .xlsx
- .pptx

How Do I Fail-Over an Engine to a New Server

In many situations, customers need to change the server they are reporting on. Sometimes this can be planned, sometimes it can be spontaneous. This article covers how to fail-over an Engine to a new server.

How Windward Engine Fail-Over Works

Fail-over works very simple with Windward: simply stop reporting on your old Engine server, and start reporting on your new Engine server.

What is the Upper Limit of Supported Fail-Over

Fail-over is lenient with Windward to cover many different cases. To look at the most extreme case of fail-over allowed, where a user needs to fail-over ever reporting server they have licensed simultaneously:

1. Users are able to fail-over every Engine server they have licensed simultaneously.
2. Users can overlap outputting reports from the old Engine server and the new Engine server while the fail-over takes place for any single server.
3. Users must stop reporting from the old Engine server within a reasonable amount of time of performing the fail-over for any single server.

For this most extreme case, at one given time the user can at maximum be outputting from every old instance of Engine server and every new instance of Engine server simultaneously.

Limitations of to Fail-Over

Users are not allowed to continue reporting on the old Engine server for an unreasonable amount of time after performing fail-over to a new server. Users who continue to output on both servers for an unreasonable amount of time will have their case assessed on an individual basis, and may have all fail-over capabilities removed from their account.

When conducting fail-over, please avoid starting any new output on your old server to avoid any violations of Windward fail-over.

How Do I learn what input parameters a template needs programmatically with the .NET Engine?

For some context, this is so users can edit and upload templates, and then you can provide a dynamic “generate document” UI based on the template that the user selects.

For the .NET Engine use the GetMetrics Method

Reference

.NET Report Engine app.exe.config Settings

This article describes the optional properties that can be placed in your .NET Report Engine application's app.exe.config file.

Place these properties into the "WindwardReports" section of the app.exe.config file, e.g.:

```
<configuration>
    <configSections>
        <section name="WindwardReports" type="System.Configuration.
NameValueSectionHandler" />
    </configSections>
    <WindwardReports>
        <add key="license" value="EnterKeyHere" />
    </WindwardReports>
</configuration>
```

Properties Categories

The below property settings fall into three general categories:

1. Read in and set when the Report Engine starts. You need to restart the Report Engine for new values to be read and used. These properties are categorized as **Global**.
2. Read in and used every time a output is generated from a Report Template. When these are changed, the next report will use the new settings. These properties are categorized as **Local**.
3. Default values for properties you can set in a (Process)Report object. When you change these the next instantiation of the report object will use those new values as the defaults. You can then explicitly set these properties in the object. These properties are categorized as **Report**.
4. Other properties that don't fall into the above categories are categorized as **Miscellaneous**.

Use these links to jump forward to a properties category:

Global Properties

Local Properties

Report Properties

Miscellaneous Properties

Global Properties

Name	Allowed Values	Default	Description
asian.support	*true, false*	*true*	If set to *false,* the server assumes the files needed for output of Asian text is not available and will fall-back to latin text only.
default.chart.dpi	any number	300	When charts are rendered to a bitmap (for printers), they are rendered at this resolution.
default.image.dpi	any number	96	If a bitmap does not have it's DPI set, this is the value used.
expand.out.text	*ignore, newline, paragraph*	*paragraph*	What to do if an Out Tag has a <CR> and/or <TAB> in the text. Choices are *ignore*, *newline*, or *paragraph*. <TAB> is handled for both *newline* and *paragraph*. *newline* is a line break (like <br/> in html) and *paragraph* is a paragraph break.
font.files	fontname; fontname;...	none	Example: C:/windows/fonts/*.ttf;c:/windows/fonts/*.ttc; A series of pathnames, separated by semicolons, of true type fonts installed on the system. The only wildcard recognized is the "*" (i.e. no regular expressions). The Report Engine will look in the usual directories on Windows & Linux; this setting is just to include non-standard locations.

Name	Allowed Values	Default	Description
font.map	font_replace1= font_substitute1;...	none	Example: Arial=Courier New;Times New Roman=Verdana Map a font onto another. In the example above any use of Arial in the generated output will instead use Courier New. Any use of Times New Roman will instead use Verdana. Very useful when a template uses fonts that do not exist on the system the engine is running on.
force.bitmap.format	*bmp, gif, jpeg, png, & tiff*	none	This will force all bitmaps in the Report Template output to be converted to the requested format. The default is no conversion.
ignore.tag.colors	*true, false*	*false*	Set to *false* to remove foreground colors from Tags if the color of the Tag in the template matches the Tag's assigned color.
layout.document	*on, off*	*on*	If set to *off*, then page layout for output of type DOCX, XLSX, PPTX and HTML in non pages mode, will not be performed. The advantage o turning this *off* is that document production will be faster. This setting is ignored for other output types.
license	a valid Windward license	none	This is required for the .NET Report Engine – it is your license to run the program. I is a uuencoded string and can be in a single line or in multiple lines. If multiple lines, do not put spaces at the beginning/end of the lines.

Name	Allowed Values	Default	Description
logging.filename	*off*, a log4j properties filename	none	The location of your log4j properties file. If you use log4j for your application, put the properties file location here. The value of *off* means you are not using log4j. There is no need to set this if log4j is initialized before Windward Reports is called.
odata.max-message-size	Any number	10000000	Set the maximum message size in the OData provider.
pdf.font_mapping	*auto*, postscriptFont1= displayFont1;...	none	Can be set to *auto* or ArialMT=Arial;Arial-BoldMT=Arial,Bold; *auto* will set the font names to the names specified in the Report Template, overwriting the true postscript name. The series of "postscriptFont=displayFont;" is an explicit mapping and you can have as many as you wish.
pdf.makeLineToFit	*true, false*	*true*	Set to *true* to force PDF text to fit within the Report Engine-calculated width. Set to *false* to fit the text within the width Acrobat calculates.
pdf.rgbTransparencyBlendingSpace	*true, false*	*false*	Set to *false* to turn off using the RGB space for image color blending in PDF output
system.locale	any locale such as *de* or *de_CH*	the system locale	Set the locale for system messages.

Local Properties

Name	Allowed Values	Default	Description
check.for.glyphs	*on, off*	*on*	Checks the font specified for text and changes the font if it does not have glyphs for some of the text. If a font has normal glyphs, but not bold/italic and those are needed, it will use the normal font and widen/skew the rendered glyphs.
check.for.glyphs.fontnames	font name 1;font name 2;...	MS Mincho;Arial Unicode MS	The list of fonts to consider when the specified font does not have the glyphs needed. The font with the largest number of needed glyphs is used. If more than one font has the largest number, the earliest font in the list is used.
check.for.ole.images	*true, false*	*true*	If *true,* the Report Engine will check all images brought in via an Out Tag for an OLE header (used in Northwind & Access databases), and will strip it off. Non-OLE images are not affected.
html.embed_images	*true, false*	*false*	If set to *true*,

Name	Allowed Values	Default	Description
			embeds images in HTML output. If set to *false*, separate image files are output.
html.split_pages	*true, false*	*false*	Set to *true* to output multiple files, one HTML file per document page. Set to *false* to create one continuous HTML page.
line.break	*internal, external*	*internal*	Set to *internal* to use the Windward Reports internal line breaker. Set to *external* to use the Java or .NET runtime line breaker. The Windows default is *external*; the default for other operating systems is *internal*. Use *internal* for hyphenation, small caps, font scaling and font spacing for PDF and printer output.
report.hyphenate	*0*, *1*, or *2*	*1*	Use to hyphenate generated output. *0*: do not hyphenate; *1*: hyphenate if auto-hyphenate is set in the Report Template; *2*: hyphenate.
report.remove-unused-	*true, false*	*false*	Set to *true* to

Name	Allowed Values	Default	Description
formats			remove unused formats (Word only).
report.timeout	any number (in seconds) greater than 0	none	If a report takes longer than this time (+/- 15 seconds) to generate, it will throw an exception.
report.trim	none, *end.whitespace*, and *end.blank*	none	Use this to trim whitespace at the end of a report. *end.whitespace* will remove empty paragraphs and section breaks. *end.blank* will also remove blank text.
report.write-fields-as-fields	*true, false*	*true*	If *true*, DOCX form fields and content controls are written to PDF and HTML output as form fields. If *false*, they are written as text.
sql.smart-quote	*true, false*	*false*	If set to *true,* a '${var}' will be changed to ? instead of the regular substitution of '?'.
sql.timeout	any number 0 or greater	Whatever the underlying data connector (ADO.NET) has as a default.	Set the timeout on SQL selects in the underlying SQL connector. A value of 0 is infinite and is NOT recommended.
trim.tables	*aggressively*	*aggressively*	For Excel only - it will remove empty

Name	Allowed Values	Default	Description
			rows at the bottom of the table and empty columns on the right side of rows. It will remove formatted cells, but it will not remove cells with content or that are merged with other cells.
txt.font_name	any valid font name	Courier (Courier New on Windows)	Use this font instead of Courier to calculate text placement in Report Template output.
txt.font_size	*6 - 256*	*12*	Use this font size instead of 12pt to calculate text placement in Report Template output.
use.decimal	*true, false*	*false*	Set to true to use BigDecimal instead of Double in the Windward functions. Useful in cases when String to Double rounding errors occur.
word.version	*2007, 2010, 2013, 2016*	*2016*	The Report Template will be parsed using the default values for this version of Word.
write.tags	*0, 1, 2*	*2*	When generating output with Tags without a data source applied, it will output the Tags

Name	Allowed Values	Default	Description
			in the following format: *0*: text; *1*: field (Word), Report Designer macro (Excel), text (PPT); *2*: field, bitmap/chart for Tags that produce a bitmap or chart (Word only).

Report Properties

Name	Allowed Values	Default	Description
csv.cell_separator	any string	,	In CSV reports, this changes the sepa from "," to the specified string.
csv.quote_char	any string	"	In CSV reports, changes the quote de this string.
csv.newline	any string	*\r\n*	In CSV reports, changes the new line *n* (carriage return, line feed) to this st
csv.trim_whitespace	*true, false*	*false*	In CSV reports, if set to *true*, this will t at the end of cell contents in all cells.
html.full_file	*true, false*	*true*	The default value for `ProcessOpenXmlAPI.getFullFil` overridden with a programatic call to `Report.setFullFile()`.
html.html_type	see setSpec()	see setSpec()	The default value for `ProcessOpenXmlAPI.getSpec()`. overridden with a programatic call to `Report.setSpec()`.
openxml.copy-drawings	*true, false*	*true*	The default value for `ProcessOpenXmlAPI.getCopyDra` be overridden with a programatic call `setCopyDrawings()`.
openxml.copy-embedded-objects	*true, false*	*true*	The default value for `ProcessOpenXmlAPI.getCopyEmb`

Name	Allowed Values	Default	Description
			This can be overridden with a progra `Report.setCopyEmbeddedObject`
openxml.copy-metadata	see [setCopyMetadata()]()	*2*	The default value for `ProcessOpenXmlAPI.getCopyMeta` be overridden with a programmatic c `Report.setCopyMetadata()`.
openxml.copy-shapes	*true, false*	*true*	The default value for `ProcessOpenXmlAPI.getCopySha` be overridden with a programmatic c `Report.setCopyShapes()`.
openxml.recalculate	*true, false*	*true* for XLSX; *false* for DOCX and PPTX	The default value for `ProcessOpenXmlAPI.getRecalcu` This can be overridden with a progra `Report.setRecalculateOnOpen(`
openxml.write-chart-as-chart	*true, false*	*true*	The default value for `ProcessOpenXmlAPI.getChartsA` can be overridden with a programmatic `Report.setChartsAsCharts()`.
pdf.compression_level	*0, 1*, or *2*	*1*	The default value for `ReportPdf.getCompressionLeve` overridden with a programmatic call `ReportPdf.setCompressionLeve`
pdf.font_level	*0, 1*, or *2*	*2*	This is the default value for `ProcessPdf.setFontLevel()`. Yo number of the static final int specifie This can still be overridden with a pro `ProcessPdf.setFontLevel()`.
pdf.key_length	*40* or *128*	*40*	The default value for `ReportPdf.ge` This can be overridden with a progra `ReportPdf.setKeyLength()`.
pdf.owner_password	any string	none	The default value for `ReportPdf.getOwnerPassword()` overridden with a programmatic call `ReportPdf.setOwnerPassword()`
pdf.PDF_A	*on, off*	*off*	The default value for `ReportPdf.is`

Name	Allowed Values	Default	Description
			be overridden with a programmatic c `ReportPdf.setPdfA()`.
pdf.security	see [getSecurity()]()	*0*	The default value for `ReportPdf.ge` This can be overridden with a prograı `ReportPdf.setSecurity()`.
pdf.user_password	any string	none	The default value for `ReportPdf.getUserPassword()`. overridden with a programmatic call `ReportPdf.setUserPassword()`.
report.hidden.mode	see [Report.setHiddenMode()]()	*0*	The default for `Report.getHidden` be overridden with a programmatic c `Report.setHiddenMode()`.
report.locale	any locale string	the system locale	The default value for `Report.setLo` must use either a language (en) or lar (en_US). This can be overridden with call to `Report.setLocale()`. This i when generating output for things lik formatting of a currency amount.
report.locale_asian	any locale string	the system locale for East Asian content	Similar to report.locale except used w Tag is marked as East Asian.
report.locale_bidi	any locale string	the system locale for bidi content	Similar to report.locale except used w Tag is marked as bi-directional, e.g. *a*
txt.break_page	*true, false*	*false*	The default value for `ProcessText.getBreakPage()`. T overridden with a programmatic call `ReportPdf.setBreakPage()`.
txt.show_headers	*true, false*	*false*	The default value for `ProcessText.getShowHeaders()` overridden with a programmatic call `ReportPdf.setShowHeaders()`.

Name	Allowed Values	Default	Description
txt.use_margins	*true, false*	*false*	The default value for `ProcessText.getUseMargins()` . overridden with a programmatic call t `ReportPdf.setUseMargins()` .
txt.use_soft_eoln	*true, false*	*false*	The default value for `ProcessText.getUseSoftEoln()` overridden with a programatic call to `ReportPdf.setUseSoftEoln()` .

Miscellaneous Properties

Name	Allowed Values	Default	Description
import.use.child.styles	*true, false*	*true*	(Version 13.x only. For applying/importing parent styles in versions 14.x and later, se Import Tag Reference.) When an importec template's style settings don't match the master document this determines if defau settings in the imported document will match the imported style (*true*) or the master style (*false*).
sql.parameters	*substitution*, *parameters* or *parameters.always*	*parameters*	Sets the default value in `DataSourceProvider.setParamMode(`
use.parent.format	*true, false*	*false*	(Version 13.x only. For applying/importing parent styles in versions 14.x and later, se Import Tag Reference.) This sets the defau value for the Import Tag **use-parent-format** property. The default is *false* and does nothing. If set to *true*, then imported templates will have their first paragraph a character formatting set to that of the paragraph in the parent template where t Import Tag is located.

ImportFileClass

Use the ImportFileClass if you have a method for accessing files and/or security protocols that is not supported by Windward. See that attached file for a sample implementation

In the app.exe.config file you can set the property ImportFileClass. You set it to the classname of the class you implement. For example, you can set:

ImportFileClass=net.windward.env.StandardImportFile

If that class exists, it will be used instead of the default class (which is net.windward.env.StandardImportFile). This class must be available to the Report Engine so it must be loaded with your application. The class must implement ImportFile. It has a single method that is passed the information of the file you wish to import and it returns the imported file.

SampleImportFile.java

.NET Report Engine Configuration File Reference

The article details how to name the .NET Engine configuration file, and its contents.

app.exe.config

In the .NET environment, we are encouraged by Microsoft to use a configuration file rather than the Windows Registry for settings and values. The advantage to this is you can use config files to copy all files to a new location without requiring the migration of a bunch of different settings and values in the Registry.

The name and location of these files is set by Microsoft and cannot be overridden (it is different for Office Add-Ins such as Report Designer). The schema for the config (XML) file is also specified by Microsoft.

For an ASP.NET application, the config file is named ‘web.config’ and goes in the root directory of the website.

For a program named "RunReport.exe" the file is labeled "RunReport.exe.config". This must be in the same directory as RunReport.exe. If you rename "RunReport.exe" to "MyCoolApp.exe", then you need to also rename the config file "MyCoolApp.exe.config". The common shorthand for the config file is "app.exe.config", but that is not its actual name (unless your application is named "app.exe", which is unlikely).

Sample app.exe.config File

We provide a sample config file named ‘WindwardReports.dll.config’ with .NET Report Engine, but a config file with that name is not used by .NET Report Engine. Please do not try to use a file with this name.

Below is a sample config file with the minimum required settings for .NET Report Engine. At a minimum, your license key must go in this file. .NET Report Engine will not look in the registry for the license key; it must be in the config file. Generally (definitely in the case of an ASP.NET app), you will have additional sections in this file.

Although not required, we've included a logging section in the sample app.exe.config, since logging is used so often by embedding-application developers and Support for troubleshooting.

```
<?xml version="1.0" encoding="utf-8" ?>
    <configuration>
        <configSections>
```

```
            <section name="WindwardReports" type="System.Configuration.
NameValueSectionHandler"/>
            <section name="log4net" type="log4net.Config.
Log4NetConfigurationSectionHandler,log4net, Culture=neutral,
PublicKeyToken=669e0ddf0bb1aa2a" />
        </configSections>
    <WindwardReports>
    <!-- Put your key here. Multiple lines are ok but do not put spaces at the
beginning/end of the lines -->
        <add key="license" value="the/license/goes/here"/>
    </WindwardReports>

   <log4net debug="false">
      <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
         <!-- This specifies where the file will be written. Make sure your app has
write permissions to this folder! -->
         <param name="File" value="C:\temp\LoggingExample.log" />
         <param name="AppendToFile" value="true" />
         <!-- When a file is filled it is renamed LoggingExample.log.1 and a new
LoggingExample.log is created. -->
         <param name="MaxSizeRollBackups" value="10" />
         <!-- MaxSizeRollBackups * MaximumFileSize determines how much recent logging
is retained. -->
         <param name="MaximumFileSize" value="50KB" />
         <param name="RollingStyle" value="Size" />
         <param name="StaticLogFileName" value="true" />
         <layout type="log4net.Layout.PatternLayout">
               <!-- When sending a log to Windward, we need all of these values, but
you can change the order. -->
               <param name="ConversionPattern" value="%date [%thread] %level %logger -
%message%newline%exception" />
         </layout>
      </appender>
      <root>
         <!-- Normally set to info or warn. When providing to Windward set to debug. -->
         <level value="debug" />
         <appender-ref ref="RollingFileAppender" />
      </root>
   </log4net>
</configuration>
```

Log File Not Created?

Having problems with your log file not being created? To address that:

If you configured logging in your app.exe.config file and you find that there is no log file created, **make sure you have permissions to write to the location you specified in app.exe.config.**

If that doesn't resolve the issue, try putting the following code in your application before your logging code: `log4net.Config.XmlConfigurator.Configure();`

You can define it in Global.asax:

```
void Application_Start(object sender, EventArgs e)
{
    // Code that runs on application startup
    // Initialize log4net.
    log4net.Config.XmlConfigurator.Configure();
}
```

You can also add the following line (either mentioning your config file name or not):

`[assembly: log4net.Config.XmlConfigurator]`

Or

` [assembly: log4net.Config.XmlConfigurator(ConfigFile = "Web.config", Watch = true)]`

Additional Logging Info

If you would like additional logging info with your exceptions such as the InnerException and the StackTrace, you can use Windward's custom log format layout. To use this, replace the current <layout> block within the <log4net> block in app.config and replace it with this.

If you would like to use the ExceptionFormatter class in other applications, [you can get it here](#).

```
<layout type="Kailua.net.windward.utils.ExceptionFormatter" >
        <param name="Pattern" value="%date [%thread] %level %logger -
%exception%newline" />
</layout>
```

To have log4net see this class, you need to initialize it in your code as follows:

```
// by loading this, log4net sees it
var formatter = new ExceptionFormatter();
XmlConfigurator.Configure();
```

Using the <log4net> section from above, the modified version would look like

```
<log4net debug="false">
      <appender name="RollingFileAppender" type="log4net.Appender.RollingFileAppender">
```

```
         <!-- This specifies where the file will be written. Make sure your app has
write permissions to this folder! -->
         <param name="File" value="C:\temp\LoggingExample.log" />
         <param name="AppendToFile" value="true" />
         <!-- When a file is filled it is renamed LoggingExample.log.1 and a new
LoggingExample.log is created. -->
         <param name="MaxSizeRollBackups" value="10" />
         <!-- MaxSizeRollBackups * MaximumFileSize determines how much recent logging
is retained. -->
         <param name="MaximumFileSize" value="50KB" />
         <param name="RollingStyle" value="Size" />
         <param name="StaticLogFileName" value="true" />
         <layout type="Kailua.net.windward.utils.ExceptionFormatter" >
                 <param name="Pattern" value="%date [%thread] %level %logger -
%exception%newline" />
         </layout>
      </appender>
      <root>
         <!-- Normally set to info or warn. When providing to Windward set to debug. -->
         <level value="debug" />
         <appender-ref ref="RollingFileAppender" />
      </root>
   </log4net>
```

Optional Settings

See .NET Report Engine app.exe.config Settings for all of the optional settings that can be added to app.exe.config.

.NET Report Engine Data Source Connectors Reference

For SQL data sources, such as Microsoft SQL Server, Oracle, etc., .NET Report Engine requires an ADO.NET connector be installed so the Engine can connect to the data source. The ADO.NET connectors are provided by the SQL data source vendor.

This article details where to download, and how to install, the connectors for each supported SQL data source.

Microsoft SQL Server

Microsoft SQL Server includes its ADO.NET connector as part of the .NET Framework, so no additional software must be installed.

Oracle Database

To install the ODAC (ODP.NET) Oracle connector (aka Oracle Managed Client), and only the drivers needed, do the following:

1. Go to the 'Oracle Data Access Components' page: http://www.oracle.com/technetwork/database/windows/downloads/utilsoft-087491.html
2. Click the 'Accept License Agreement' radio button.
3. Find the 'ODP.NET, Managed Driver Xcopy version only' section.
4. Click the 'ODP.NET_Managed_ODAC...zip' link (http://download.oracle.com/otn/other/ole-oo4o/ODP.NET_Managed_ODAC12cR4.zip)
5. Oracle will request you sign in before download (if you don't have an account, you must create one).
6. After signing in, the necessary .zip file will be downloaded.
7. Close all Office files.
8. Once you have downloaded the .zip file, extract all the files to your Downloads folder.
9. Open a command prompt **as Administrator** and navigate (cd) to the extracted Downloads folder (i.e. ODP.NET_Managed_ODAC12cR4).
10. Run the command: `install_odpm.bat c:\oracle both true`
11. You should see several "The operation completed successfully" prompts.
12. You have installed the necessary drivers to connect to your Oracle data source.

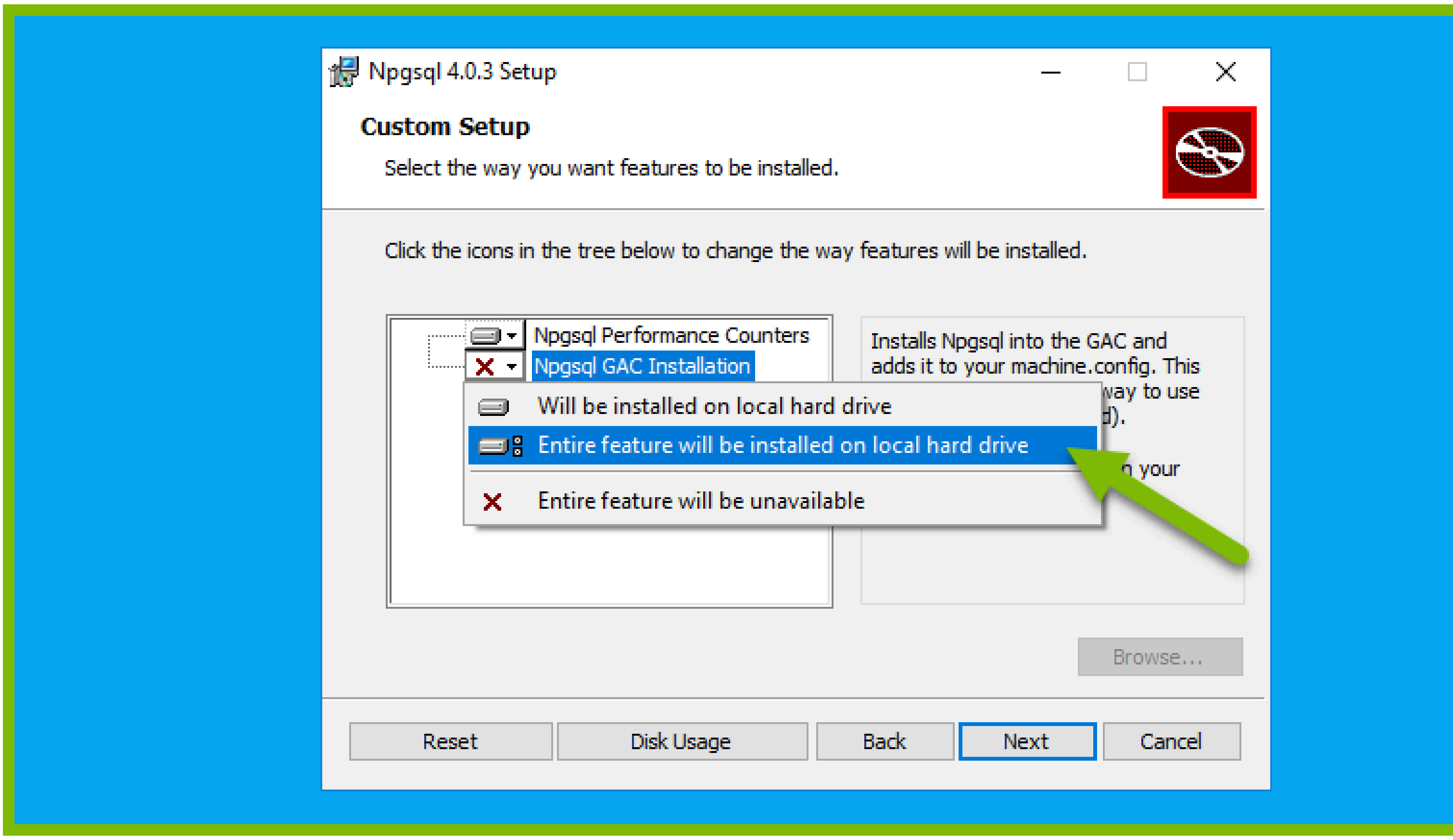
MySQL Server

Download this [ADO.NET connector](#) , then execute the installer.

PostgreSQL

To download and install the PostgreSQL Npgsql ADO.NET connector:

1. Download the latest release of Npsql from [github](#).
2. Double-click on the .msi file to begin the installation.
3. Accept the EULA.
4. When the Custom Setup dialog appears, click on "Npgsql GAC Installation" and select "Entire feature will be installed on local hard drive".



Npgsql 4.0.3 Setup

Custom Setup

Select the way you want features to be installed.

Click the icons in the tree below to change the way features will be installed.

Installs Npgsql into the GAC and adds it to your machine.config. This

Npgsql Performance Counters
Npgsql GAC Installation
Will be installed on local hard drive
Entire feature will be installed on local hard drive
Entire feature will be unavailable
Browse...
Reset
Disk Usage
Back
Next
Cancel

5. Click on Next
6. Click on Install
7. Click on Finish

IBM DB2

To download and install the IBM DB2 ADO.NET connector:

1. Go to this [IBM Fix Central page](#).

2. In the 'Product selector' input box type "IBM Data Server Client Packages".
3. In the 'Installed Version' input box select the latest version.
4. In the 'Platform' input box select your PC OS type. Example: "Windows 64-bit, x86".
5. Click on Continue.
6. On the 'Identify fixes' page, choose the radio button next to 'Individual fix IDs' and in the input box type "*dsdriver*FP002".
7. Click on Continue.
8. On the 'Select fixes' page, check the fix pack: "DSClients-ntx{bitness}-dsdriver-{version#}-FP002". Where {bitness} is 32 or 64 and {version#} is the latest version you chose. *You don't want the Merge Modules so be sure the fix pack doesn't have "mm" before the version #.*
9. Click on Continue.
10. You may have to create an IBMid. Just fill out the form and you'll receive an email with a code that allows you to complete your new user registration.
11. On the 'Download options' page, choose "Download using your browser (HTTPS)" option.
12. Click on Continue.
13. On the 'Download files using HTTPS' page, download the "{version#}fp2_ntx{bitness}_dsdriver_EN.exe". Where {bitness} is 32 or 64 and {version#} is the latest version you chose.
14. After the download completes, run the "{version#}fp2_ntx{bitness}_dsdriver_EN.exe" program and select the defaults.
15. After you have followed the prompts from the IBM installer, you have completed the driver install.

ODBC

Differing Versions of Report Designer and the Engine

How using different versions of Report Designer and the Engine works.

Resolution

The version of your Engine must be equal to or greater (newer) than the version of Report Designer you are using.

The reason for this is that newer versions of Report Designer use newer features that older versions of the Engine don't recognize.

The best practice is to keep your version of Report Designer below or equal to the version of the Engine. The problem with using a new version of Report Designer, and an older version of the Engine is that you could possibly create features that the Engine is not aware of in your template. It will run fine in Report Designer, but when the Engine encounters a new or altered feature, it will not be able to process it or it will process it differently than expected.

Examples

Best Method - Engine and Report Designer are the same version	
Report Designer	16.2.0.188
Engine	16.2.0.188

OK - Engine has a greater version than Report Designer	
Report Designer	16.1.108.0
Engine	16.2.0.188

NOT OK - Engine version is less than Report Designer version	
Report Designer	16.2.0.188

NOT OK - Engine version is less than Report Designer version	
Engine	16.1.108.0

Output Limitations

This article covers output format limitations that can occur when generating output from Report Templates created in Microsoft Office (Word, Excel, or PowerPoint). These limitations can cause unexpected results when your output is generated. Windward Studios recommends familiarizing yourself with these design limitations so you can create templates that are accurately reflected when you generate output.

All Formats

Vertical Absolute Position

Absolutely-positioned objects can cause problems as they don't move when data is merged into a Report Template's output. These problems can be compounded by the fact the merged-in data needs to adjust other objects and take into account their placement. For more about how Report Templates work see **[Template Layout Best Practices]**.

The normal (unTagged) text in your Report Template will appear in the same place in the output. However, Tags can produce objects that may expand or shrink depending on the Tags' select statements. ForEach Tags will always expand and produce more rows of text, while the If, Else, Switch and Case Tags may add or remove text from a Report Template's output.

When Tag objects expand they push the text and objects located below further down in the Report Template's output. In the case where Tag objects shrink, the text and objects below will move up in the output. This can cause page breaks to occur unexpectedly. Using hard and soft page breaks can ensure your text appears as you desire in the Report Template's output.

When you absolutely position an item, you are telling Microsoft Office that item should always remain in the same position in the output. You can think of this as a large rock extending out from a waterfall causing the water to crash against the rock before it hits the waterfall's basin. The water's flow is disrupted by the rock during its descent. Similarly, text and other objects that are not absolutely positioned will flow smoothly and dynamically with the layout; however, items that are fixed will have text written over them producing undesired output. Windward Studios generally recommends not using absolutely-positioned objects, but if you must use them, here are a few tips to assist you:

- Objects with a vertical absolute position relative to the top of the page or margin are problematic as Windward has to guess how to place them based on the output objects generated prior to the object. The most commonly absolutely-positioned items are text boxes, but this also happens with images and tables.
- Absolute vertically-positioned items based on the paragraph or line is fine.
- Absolute horizontal positioning is fine.

Crossing Formats

When generating output in one format, from a Report Template created in another format, there may be inconsistencies due to the nature of the program used to create the Template. For example, output generated from Excel Templates may not produce the exact same output as that generated from PowerPoint Templates, because PowerPoint does not understand the concept of tables, rows and formulas. For this reason, Windward doesn't recommend and may not support input and output formats which are problematic. Please see Input to Output Format considerations.

OpenXML Limitations

ℹ OpenXML formats are DOCX, PPTX and XLSX

- Report Templates other than DOCX, PPTX, and XLSX have limited functionality, due to the limited functionality of older formats:
  - RTF interpretation varies from different Word versions and even service packs between versions. It is for this reason Windward Studios doesn't support new feature development for RTF Report Templates.
  - PPT, CSV and XLS Report Templates were not created using the OpenXML-based file formats, and are more difficult to interact with.
  - DOC is not a supported Report Template format.

- Output to HTML, PDF, and matching OpenXML formats are fully supported to the extent possible for that format.
- PPTX output can only be generated from PPTX templates.
- Crossing formats (like DOCX to XLSX) is strongly discouraged and the generated output will often be a poor match because the file format settings are so different.
- Chart Tag Limitations
- **[Embedded objects limitations]**
- In output from and to dissimilar OpenXML formats, such as from XLSX to DOCX, Windward will remove the following:
  - SmartArt
  - Shapes
  - Textboxes
  - Embedded Objects

Unsupported Features

These features will not be applied to output from any Report Template:

- Tracking Changes
- Proofing marks
- Phonetic markers/guides
- Gauges

Format-Specific Limitations

Word

See **[Document (Word) Limitations]**, which describes Word format limitations in detail.

Excel

See Excel Report Template Output Limitations, which describes Excel format limitations in detail.

PowerPoint

See Support for PowerPoint PPTX Format, which describes PowerPoint format limitations in detail.

Office 365 (Word Online, Excel Online)

Report Designer Online Edition (Report Designer) can't be used in a browser-only version of Office. When you download Office 365, you are installing Office 2016, and Report Designer will function normally. See **[Limitations of Office 365]** for more details.

PDF

Fonts

When generating PDF output, the system running Report Designer or a Report Engine *must have the fonts used in the Report Template installed* (this is generally an issue on Unix-based systems).

- If the requested font is not installed, the system will use the *closest match* - and that can be quite a bit different.
- When the system has to substitute a font, it will list the font substitution in the log file.

Tables and Borders

Due to a PDF limitation, when drawing lines in Adobe Acrobat (for table and paragraph borders), each line can be joined to the previous line only if the line width and style do not change. Therefore, in PDF output:

- Table Borders - The outer box around a table is drawn as one line so the outer box will join up. Each inner row and column divider is drawn as a separate line. These inner lines are set to end on the outer box line, but Acrobat cannot be told to join the inner lines to the outer box.
- Line Continuity - If the outer box or inner line changes style or width part-way around, one line ends and another starts. Again, Acrobat does not know these two lines are supposed to be joined and will not draw them contiguously.
- Thick Borders - If borders and lines look thicker when zoomed out in the document, zoom in and you will see the lines are redrawn. When the document is printed it will appear as expected.

Conformance levels and versions

Windward currently supports PDF/A-1b version output. Should there be a request for additional conformance levels, it is recommended to suggest this on our ideas page: Ideas Page.
This is where people can request new functionality to be added to upcoming releases.

HTML

- There are no tabs in the HTML language. Tabs are approximated but cannot be matched exactly. We recommend you use tables instead.
- Absolute positioning is *not supported*. Absolute-positioned objects are treated as inline.
- If you are using a Report Engine to generate HTML for emails, we recommend:
  - Setting the **html.html_type** property to *1*
  - Or setting the **cssType property** in the ReportHtml (.NET) and ProcessHtml (Java) classes to *CSS_NO*, because Outlook does not support CSS very well.

- We only support a subset of Shapes and Smart Art at this time. This includes but is not limited to pictures and text inside of Shapes and Smart Art.

HTML templates

- All HTML limitations listed above
- Template support is designed to provide a means to pass basic formatting, not to handle complex layout via HTML. For complex layout, please use DOCX.
- For supported Tags and Style settings, see HTML and CSS Tags and Attributes Supported By Out Tags.

Printer

Printer output is basically the same as PDF output in Report Designer and the Report Engines, except for PDF-specific settings and settings that are not applicable to printed output.

Windward Input to Output Formats

Here are the input and output file formats supported by Windward. Some differ based on the version

20.0.0 .NET Report Engine Image Output Reference

16.7.0 Windward Input to Output Formats

20.0.0 Windward Input to Output Formats

Here are the input and output file formats supported by Windward. Some differ based on the version.

Windward Input Formats

- Word DOCX documents, the basic Microsoft Word document.
- Excel XLSX documents, the format for Excel spreadsheets for versions 2007 and higher.
- PowerPoint PPTX documents, Microsoft's professional presentation document.

Windward Input Format Limitations

Here are some limitations to consider when creating Report Templates.

Importing Tables

- Importing an XLSX Report Template into a DOCX Report Template is not supported.
- Importing a DOCX Report Template with tables into an XLSX Report Template imports the entire Template into the single cell that contains the Import Tag. The table becomes just text - it is not a table as Excel does not support tables within a cell.
- Importing an XLSX Report Template into an XLSX Report Template imports the entire table into the single cell that contains the Import Tag. The table becomes just text - it is not a table as Excel does not support tables within a cell.

Importing Excel Pivot Tables

- Output to PDF is not supported.
- A pivot table is non-movable and expandable. This means it can’t be placed in a worksheet containing other dynamic contents, e.g. a ForEach Tag loop. So, place pivot tables in separate worksheets.
- A data source for a pivot table can’t contain more than Z (26) columns.
- A pivot table can use only by-row-expandable ForEach Tag loops. By-column expansion of a ForEach Tag loop is not supported.

Windward Output Formats

- Word DOCX documents, the basic Microsoft Word document. (We do not support DOC.)

- Excel XLSX documents, the format for Excel spreadsheets for versions 2007 and higher. (We do not support XLS.)
- PowerPoint PPTX documents, Microsoft's professional presentation document. (We do not support PPT.)
- PDF, or Portable Document Format, allows you to create a formatted document and preserve its layout no matter which program or software was used to create it. This allows you to share your document with a large number of people, because the PDF reader software is free and ubiquitous.
- HTML web page. We support a variety of HTML formats, such as with or without CSS. Note: Outlook does not support CSS in HTML documents very well. (HTML was substantially improved in version 13.)
- Printer output is where a document is sent directly to your available printer with the inserted data intact.
- RTF, or Rich Text Format, is compatible with almost all word processors (Word, WordPad, Open Office Writer, Google documents, etc.). It allows you to exchange files between word processors for different operating systems. RTF has been around for a while and so is available for earlier versions of word processors.

Windward Input to Output Formats

With Report Designer, you can go from one supported Report Template (input) format to a different output format. For instance, you can go from a DOCX Report Template to an HTML file.

But this does not hold true in every case. For example, PPTX is not an output option for Word and Excel Report Templates, and PowerPoint Report Templates can only be output to PDF, printer and PPTX formats. Supported outputs will appear as active (i.e. not grayed out) in the output options for a given Report Template.

This table shows you which output formats are available for which input formats, with important notes below the table:

	Input: DOCX	**Input: XLSX**	**Input: PPTX**	**Input: HTML**[1]
Output: PDF	Yes	Yes	Yes	Yes[1]
Output: printer	Yes	Yes	Yes	Yes[1]
Output: HTML	Yes[2]	Yes[2]	No	Yes[1,2]
Output: DOCX	Yes	No	No	Yes[1]
Output: XLS[3]	Yes	Yes	No	Yes[1]
Output: PPTX	No	No	Yes	Yes[1]

	Input: DOCX	Input: XLSX	Input: PPTX	Input: HTML[1]
Output: TXT	Yes	Yes	No	Yes[1]
Output: PS	Yes	Yes	Yes	Yes
Output: SVG	Yes	Yes	Yes	Yes
Output: ESP	Yes	Yes	Yes	Yes
Output: BMP	Yes	Yes	Yes	Yes
Output: PNG	Yes	Yes	Yes	Yes
Output: JPG	Yes	Yes	Yes	Yes
Output: GIF	Yes	Yes	Yes	Yes
Output: TIFF	Yes	Yes	Yes	Yes

1. HTML input is for simple template use only. Many advanced features will not carry across to other formats in the output. See HTML and CSS Tags and Attributes Supported By Out Tags.

2. HTML output may not display format-specific features exactly as intended.

3. XLS output is limited to basic formatting and does not include charts and images. We recommend using XLSX instead.

Windward Image Outputs

The following functionality was added in version 20.0.0

There are a number of different image output types Windward supports in our Java and .NET Engines. The following table will highlight those image output types as well as the commands that correspond to the output in the two different engines.

Image Type	Java Command	.NET Command
ESP	HtmlImage.RENDER_EPS	ReportImage.FORMAT.EPS
SVG	HtmlImage.RENDER_SVG	ReportImage.FORMAT.SVG

Image Type	Java Command	.NET Command
BMP	HtmlImage.BITMAP_BMP	ReportImage.FORMAT.BMP
GIF	HtmlImage.BITMAP_GIF	ReportImage.FORMAT.GIF
JPG	HtmlImage.BITMAP_JPG	ReportImage.FORMAT.JPG
PNG	HtmlImage.BITMAP_PNG	ReportImage.FORMAT.PNG
TIFF	HtmlImage.BITMAP_TIF	ReportImage.FORMAT.TIF

16.7.0 Windward Input to Output Formats

Here are the input and output file formats supported by Windward. Some differ based on the version.

Windward Input Formats

- Word DOCX documents, the basic Microsoft Word document.
- Excel XLSX documents, the format for Excel spreadsheets for versions 2007 and higher.
- PowerPoint PPTX documents, Microsoft's professional presentation document.

Windward Input Format Limitations

Here are some limitations to consider when creating Report Templates.

Importing Tables

- Importing an XLSX Report Template into a DOCX Report Template is not supported.
- Importing a DOCX Report Template with tables into an XLSX Report Template imports the entire Template into the single cell that contains the Import Tag. The table becomes just text - it is not a table as Excel does not support tables within a cell.
- Importing an XLSX Report Template into an XLSX Report Template imports the entire table into the single cell that contains the Import Tag. The table becomes just text - it is not a table as Excel does not support tables within a cell.

Importing Excel Pivot Tables

- Output to PDF is not supported.
- A pivot table is non-movable and expandable. This means it can’t be placed in a worksheet containing other dynamic contents, e.g. a ForEach Tag loop. So, place pivot tables in separate worksheets.
- A data source for a pivot table can’t contain more than Z (26) columns.
- A pivot table can use only by-row-expandable ForEach Tag loops. By-column expansion of a ForEach Tag loop is not supported.

Windward Output Formats

- Word DOCX documents, the basic Microsoft Word document. (We do not support DOC.)

- Excel XLSX documents, the format for Excel spreadsheets for versions 2007 and higher. (We do not support XLS.)
- PowerPoint PPTX documents, Microsoft's professional presentation document. (We do not support PPT.)
- PDF, or Portable Document Format, allows you to create a formatted document and preserve its layout no matter which program or software was used to create it. This allows you to share your document with a large number of people, because the PDF reader software is free and ubiquitous.
- HTML web page. We support a variety of HTML formats, such as with or without CSS. Note: Outlook does not support CSS in HTML documents very well. (HTML was substantially improved in version 13.)
- Printer output is where a document is sent directly to your available printer with the inserted data intact.
- RTF, or Rich Text Format, is compatible with almost all word processors (Word, WordPad, Open Office Writer, Google documents, etc.). It allows you to exchange files between word processors for different operating systems. RTF has been around for a while and so is available for earlier versions of word processors.

Windward Input to Output Formats

With Report Designer, you can go from one supported Report Template (input) format to a different output format. For instance, you can go from a DOCX Report Template to an HTML file.

But this does not hold true in every case. For example, PPTX is not an output option for Word and Excel Report Templates, and PowerPoint Report Templates can only be output to PDF, printer and PPTX formats. Supported outputs will appear as active (i.e. not grayed out) in the output options for a given Report Template.

This table shows you which output formats are available for which input formats, with important notes below the table:

	Input: DOCX	**Input: XLSX**	**Input: PPTX**	**Input: HTML**[1]
Output: PDF	Yes	Yes	Yes	Yes[1]
Output: printer	Yes	Yes	Yes	Yes[1]
Output: HTML	Yes[2]	Yes[2]	No	Yes[1,2]
Output: DOCX	Yes	No	No	Yes[1]
Output: XLS[3]	Yes	Yes	No	Yes[1]
Output: PPTX	No	No	Yes	Yes[1]

	Input: DOCX	Input: XLSX	Input: PPTX	Input: HTML[1]
Output: TXT	Yes	Yes	No	Yes[1]
Output: PS	Yes	Yes	Yes	Yes
Output: SVG	Yes	Yes	Yes	Yes
Output: ESP	Yes	Yes	Yes	Yes
Output: BMP	Yes	Yes	Yes	Yes
Output: PNG	Yes	Yes	Yes	Yes
Output: JPG	Yes	Yes	Yes	Yes
Output: GIF	Yes	Yes	Yes	Yes
Output: TIFF	Yes	Yes	Yes	Yes

1. HTML input is for simple template use only. Many advanced features will not carry across to other formats in the output. See HTML and CSS Tags and Attributes Supported By Out Tags.

2. HTML output may not display format-specific features exactly as intended.

3. XLS output is limited to basic formatting and does not include charts and images. We recommend using XLSX instead.

Windward Image Outputs

The following functionality was added in version 20.0.0

There are a number of different image output types Windward supports in our Java and .NET Engines. The following table will highlight those image output types as well as the commands that correspond to the output in the two different engines.

Image Type	Java Command	.NET Command
ESP	HtmlImage.RENDER_EPS	ReportImage.FORMAT.EPS
SVG	HtmlImage.RENDER_SVG	ReportImage.FORMAT.SVG

Image Type	**Java Command**	**.NET Command**
BMP	HtmlImage.BITMAP_BMP	ReportImage.FORMAT.BMP
GIF	HtmlImage.BITMAP_GIF	ReportImage.FORMAT.GIF
JPG	HtmlImage.BITMAP_JPG	ReportImage.FORMAT.JPG
PNG	HtmlImage.BITMAP_PNG	ReportImage.FORMAT.PNG
TIFF	HtmlImage.BITMAP_TIF	ReportImage.FORMAT.TIF

Miscellaneous

How Do I Upload Files to Mega.nz?

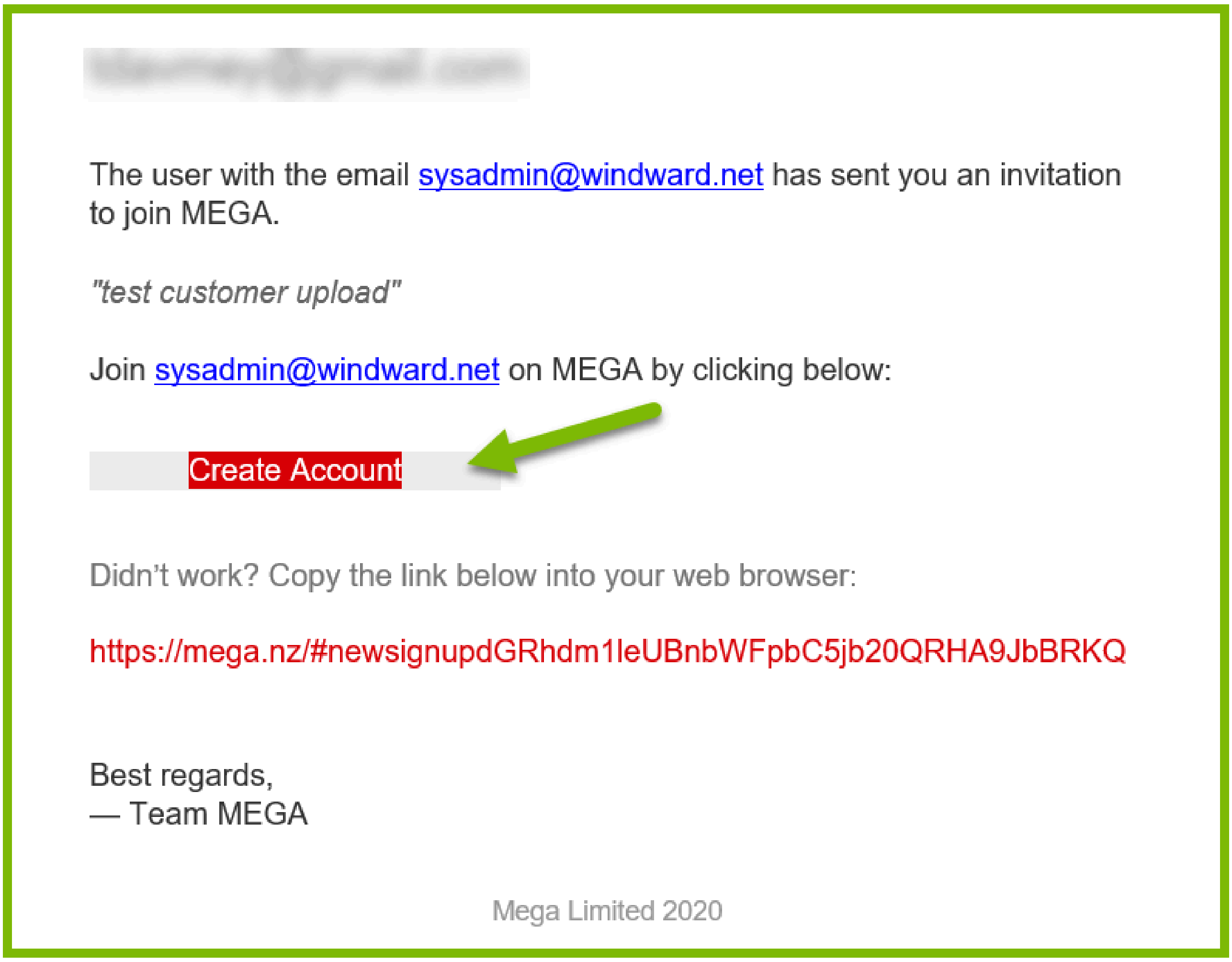
When customers must share files with Windward Support that are too large to attach to support tickets, we use mega.nz for securely uploading those files.

This article describes how to upload files to mega.nz when asked to by Support.

Create Mega.nz Account

If you already have a mega.nz account, skip to [Accept Directory Sharing Request...]()

After Support creates your upload directory, you will receive an email from mega.nz to create an account, click on "Create Account."



[redacted]

The user with the email sysadmin@windward.net has sent you an invitation to join MEGA.

"test customer upload"

Join sysadmin@windward.net on MEGA by clicking below:

Didn't work? Copy the link below into your web browser:

https://mega.nz/#newsignupdGRhdm1leUBnbWFpbC5jb20QRHA9JbBRKQ

Best regards,
— Team MEGA

Mega Limited 2020

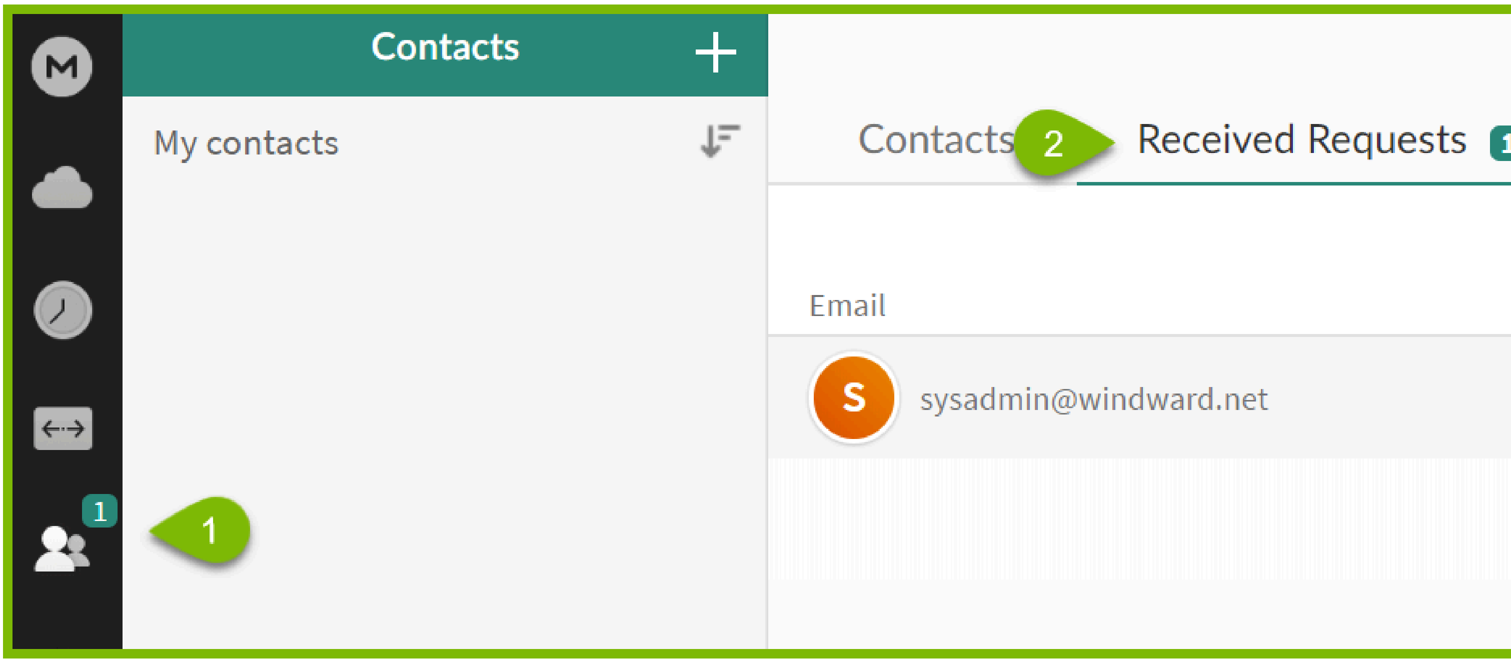
Complete the information for your free account. You will be asked to confirm the new account via email.

Notify Windward Support after you've created your new account, with the email address you used to create the account.

Accept Directory Sharing Request from Windward Support

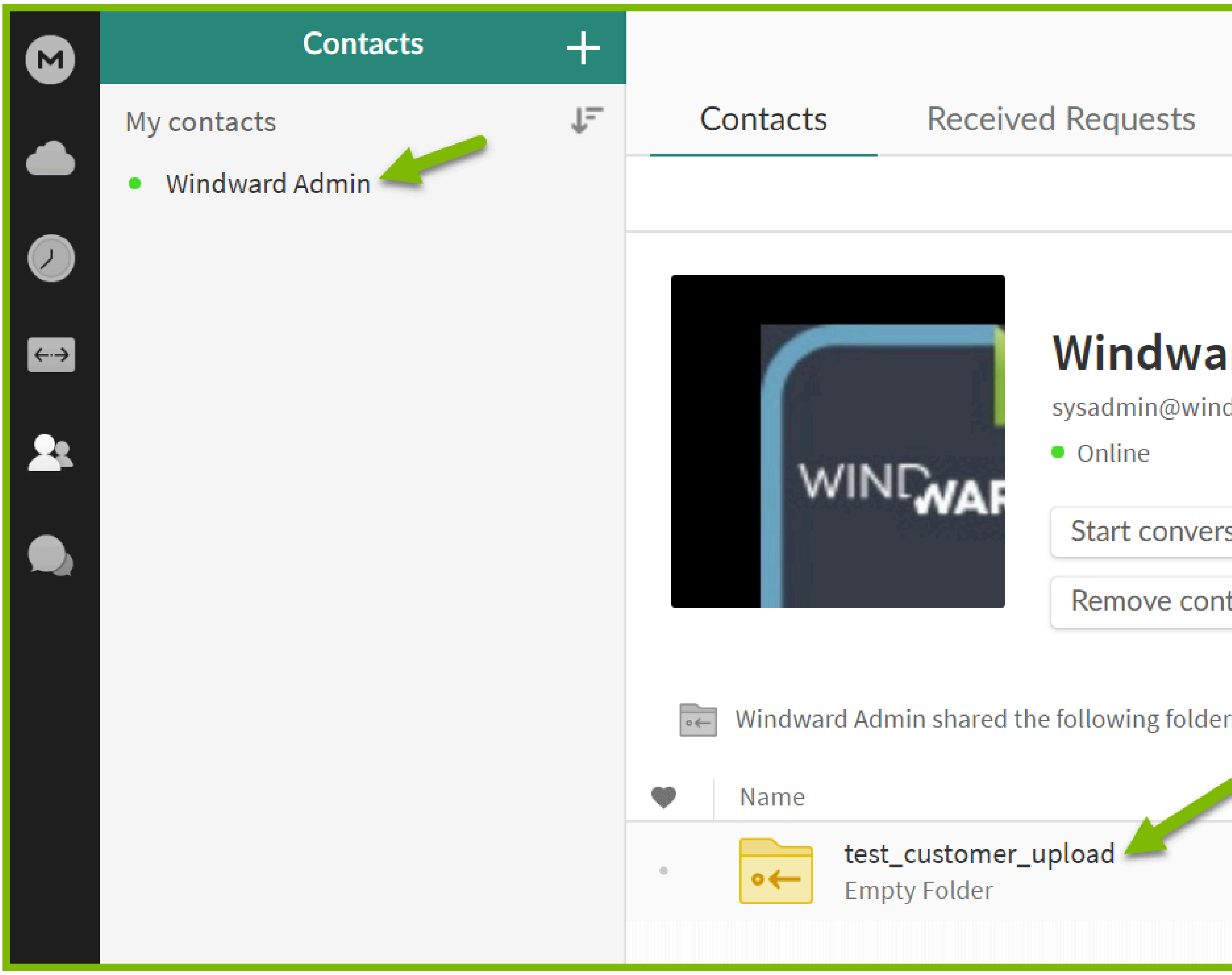
After logging in to your mega.nz account:

1. Click on the “Contacts” icon in the toolbar on the left-hand side of the page
2. Click on the “Received Requests” tab
3. Click on the “Accept” button

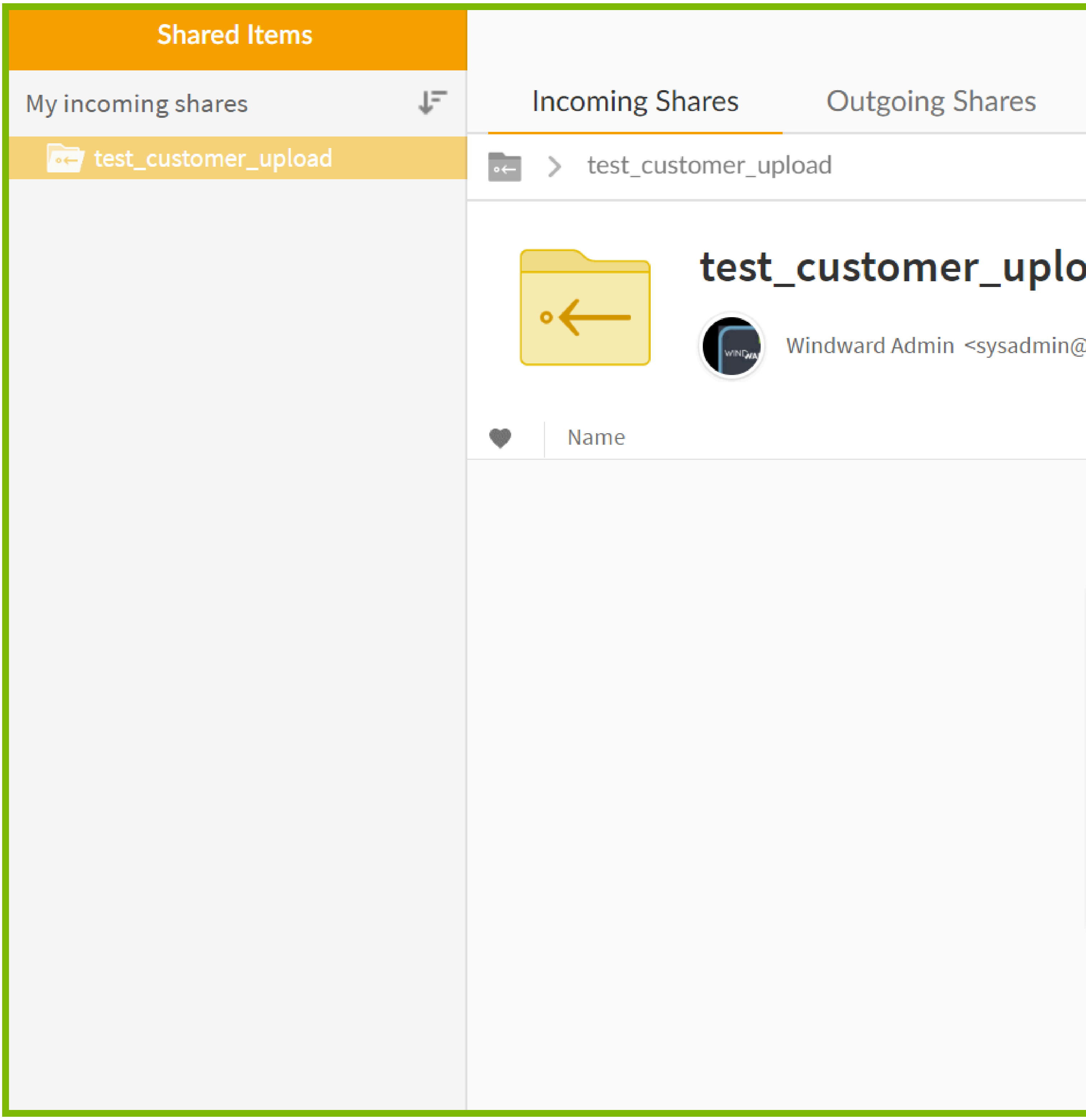


At this point, log out, then log back into mega.nz before proceeding.

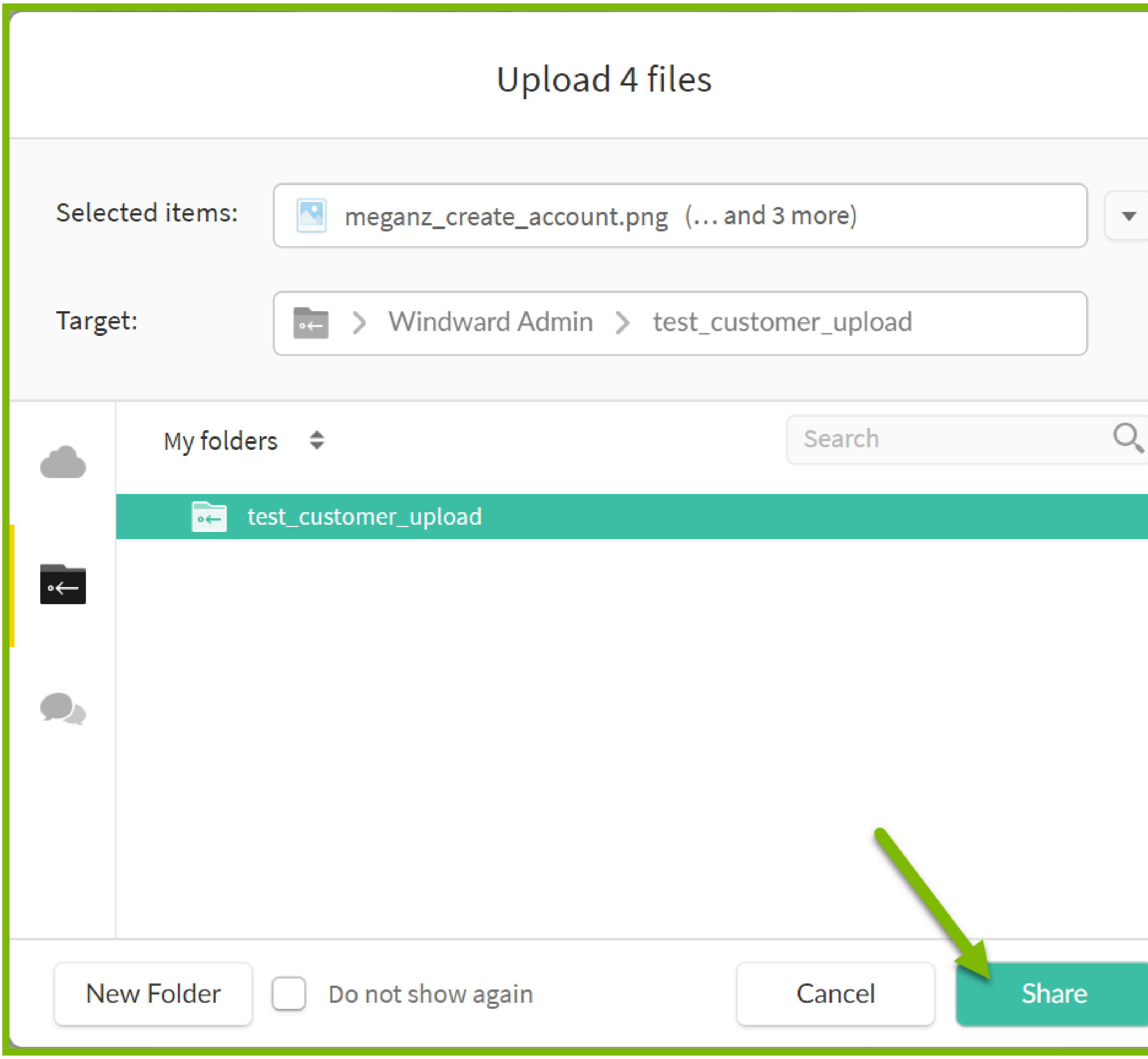
In the “Contacts” list sysadmin@windward.net will appear. Double-click on the directory Windward Support shared with you.



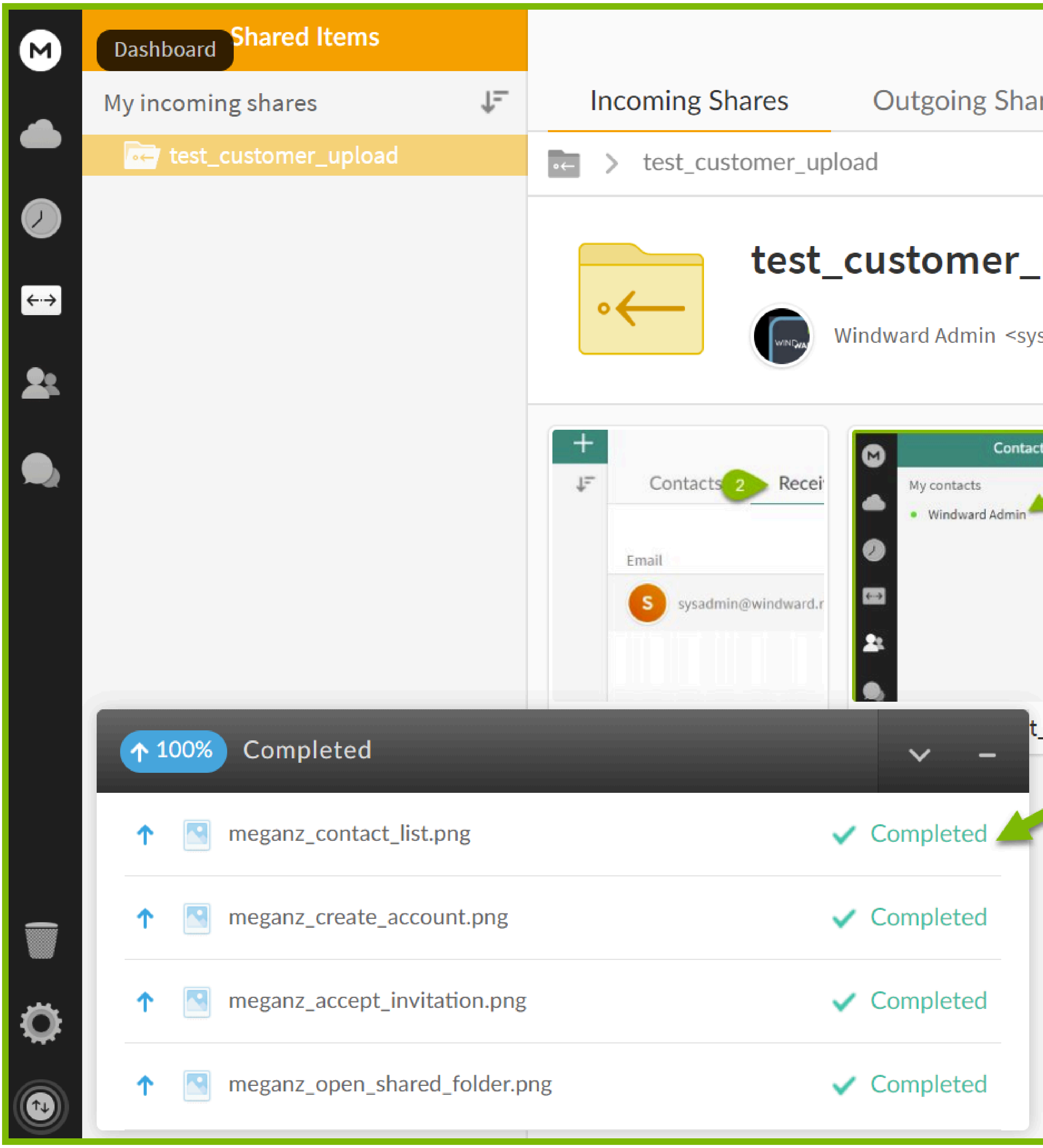
Drag and drop your file(s) to the Empty Folder icon.



When prompted to confirm the upload, click on "Share."



You will see thumbnails of your files appear in the file upload area, and a "Completed" window.



100% Completed

meganz_contact_list.png Completed

meganz_create_account.png Completed

meganz_accept_invitation.png Completed

meganz_open_shared_folder.png Completed

Now that your file(s) are securely uploaded, notify Support.

List of .NET Report Engine v15.2 DLLs and JARs

Here is a list of the DLLs and JARs installed with .NET Report Engine v15.2, and their versions.

This list was updated 10/2/2018 (15.2.403.0).

Windward Studios DLLs

Filename	Description	Version
Kailua.dll	Data metadata	Engine version
Morph.dll	Windward transform	Engine version
OfficeOutputBuilder.dll	Output using Office	Engine version
SharePointDataSourceDriver.dll	SharePoint list driver	Engine version
WindwardCustomFunctions.dll	Sample custom macros	Engine version
WindwardInterfaces.dll	Windward Interfaces	Engine version
WindwardReports.dll	Windward Engine	Engine version
WindwardReportsAPI.dll	Windward Engine	Engine version
WindwardReportsDrivers.dll	Windward data drivers	Engine version

3rd Party DLLs

Filename	Description	Version
DevExpress.Data.v15.1.dll		15.1.6
DevExpress.Office.v15.1.Core.dll		15.1.6
DevExpress.RichEdit.v15.1.Core.dll		15.1.6
DevExpress.Utils.v15.1.dll		15.1.6

Filename	Description	Version
DevExpress.XtraBars.v15.1.dll		15.1.6
DevExpress.XtraEditors.v15.1.dll		15.1.6
DevExpress.XtraLayout.v15.1.dll		15.1.6
DevExpress.XtraPrinting.v15.1.dll		15.1.6
DevExpress.XtraRichEdit.v15.1.dll		15.1.6
DevExpress.XtraTreeList.v15.1.dll		15.1.6
ICSharpCode.SharpZipLib.dll		0.86.0.518
IKVM.AWT.WinForms.dll		7.2.4630.5
IKVM.OpenJDK.Beans.dll		7.2.4630.5
IKVM.OpenJDK.Charsets.dll		7.2.4630.5
IKVM.OpenJDK.Corba.dll		7.2.4630.5
IKVM.OpenJDK.Core.dll		7.2.4630.5
IKVM.OpenJDK.Jdbc.dll		7.2.4630.5
IKVM.OpenJDK.Management.dll		7.2.4630.5
IKVM.OpenJDK.Media.dll		7.2.4630.5
IKVM.OpenJDK.Misc.dll		7.2.4630.5
IKVM.OpenJDK.Naming.dll		7.2.4630.5
IKVM.OpenJDK.Remoting.dll		7.2.4630.5
IKVM.OpenJDK.Security.dll		7.2.4630.5
IKVM.OpenJDK.SwingAWT.dll		7.2.4630.5
IKVM.OpenJDK.Text.dll		7.2.4630.5
IKVM.OpenJDK.Tools.dll		7.2.4630.5
IKVM.OpenJDK.Util.dll		7.2.4630.5
IKVM.OpenJDK.XML.API.dll		7.2.4630.5
IKVM.OpenJDK.XML.Bind.dll		7.2.4630.5

Filename	Description	Version
IKVM.OpenJDK.XML.Crypto.dll		7.2.4630.5
IKVM.OpenJDK.XML.Parse.dll		7.2.4630.5
IKVM.OpenJDK.XML.Transform.dll		7.2.4630.5
IKVM.OpenJDK.XML.WebServices.dll		7.2.4630.5
IKVM.OpenJDK.XML.XPath.dll		7.2.4630.5
IKVM.Reflection.dll		7.2.4630.5
IKVM.Runtime.dll		7.2.4630.5
IKVM.Runtime.JNI.dll		7.2.4630.5
log4net.dll		1.2.15.0
Microsoft.Data.Edm.dll		5.6.3.62017
Microsoft.Data.OData.dll		5.6.3.62017
Microsoft.OData.Core.dll		6.9.0.62068
Microsoft.OData.Edm.dll		6.9.0.62068
Microsoft.Spatial.dll		6.9.0.62068
netchartdir.dll		6.0.0.0
Newtonsoft.Json.dll		6.0.8.18111
saxon9ee.dll		9.8.0.14
saxon9ee-api.dll		9.8.0.14
System.Spatial.dll		5.6.3.62017

These JAR files are converted using IKVM and packaged in WindwardReports.DLL.

Filename	Description	Version
bcmail-jdk14-138.jar		1.38
bcprov-jdk14-138.jar		1.38
commons-codec-1.9.jar	Used for uuencode & uudecodeing	1.9
commons-	Image processing	1

Filename	Description	Version
imaging-1.0-SNAPSHOT.jar		
commons-lang-2.6.jar	Used by jfreechart	2.6
commons-io-2.1.jar	Used for HTML output	2.1
commons-logging-1.2.jar	Logging bridge	1.2
httpclient-4.5.2.jar	Read http files (datasources & import tag)	4.5.2
httpcore-4.4.4.jar	Read http files (datasources & import tag)	4.4.4
javax.mail.jar		1.4.4
javax.servlet.jsp.jar		2.2.5
javax.servlet.jsp-api.jar		2.2.1
javax.servlet-api.jar		3.0.1
jetty-6.1.26.jar		6.1.26
jetty-util-6.1.26.jar		6.1.26
jfreesvg-3.1.jar	Creates the SVG files for HTML reports	3.1
json-smart-2.2.jar	JSON dependency	2.2
json-path-2.5.1.jar	JSON query library	2.5.1
servlet-api-2.5-20081211.jar		2.5
slf4j-api-1.7.5.jar	JSON dependency	1.7.5
WindwardReports.jar	Windward Engine	Engine version
xmlgraphics-commons-2.2.jar	Image output formats	2.2

List of .NET Report Engine v16.1 DLLs and JARs

Here is a list of the DLLs and JARs installed with .NET Report Engine v16.1, and their versions.

This list was updated 10/2/2018 (16.1.60.0).

Windward Studios DLLs

Filename	Description	Version
Kailua.dll	Data metadata	Engine Version
Morph.dll	Windward transform	Engine Version
OfficeOutputBuilder.dll	Output using Office	Engine Version
SharePointDataSourceDriver.dll	SharePoint list driver	Engine Version
WindwardCustomFunctions.dll	Sample custom macros	Engine Version
WindwardInterfaces.dll	Windward Interfaces	Engine Version
WindwardReports.dll	Windward Engine	Engine Version
WindwardReportsAPI.dll	Windward Engine	Engine Version
WindwardReportsDrivers.dll	Windward data drivers	Engine Version

3rd Party DLLs

Filename	Description	Version
DevExpress.Data.v15.1.dll		15.1.6
DevExpress.Office.v15.1.Core.dll		15.1.6
DevExpress.RichEdit.v15.1.Core.dll		15.1.6
DevExpress.Utils.v15.1.dll		15.1.6

Filename	Description	Version
DevExpress.XtraBars.v15.1.dll		15.1.6
DevExpress.XtraEditors.v15.1.dll		15.1.6
DevExpress.XtraLayout.v15.1.dll		15.1.6
DevExpress.XtraPrinting.v15.1.dll		15.1.6
DevExpress.XtraRichEdit.v15.1.dll		15.1.6
DevExpress.XtraTreeList.v15.1.dll		15.1.6
ICSharpCode.SharpZipLib.dll		0.86.0.518
IKVM.AWT.WinForms.dll		8.5.0.2
IKVM.OpenJDK.Beans.dll		8.5.0.2
IKVM.OpenJDK.Charsets.dll		8.5.0.2
IKVM.OpenJDK.Cldrdata.dll		8.5.0.2
IKVM.OpenJDK.Corba.dll		8.5.0.2
IKVM.OpenJDK.Core.dll		8.5.0.2
IKVM.OpenJDK.Jdbc.dll		8.5.0.2
IKVM.OpenJDK.Localedata.dll		8.5.0.2
IKVM.OpenJDK.Management.dll		8.5.0.2
IKVM.OpenJDK.Media.dll		8.5.0.2
IKVM.OpenJDK.Misc.dll		8.5.0.2
IKVM.OpenJDK.Naming.dll		8.5.0.2
IKVM.OpenJDK.Nashorn.dll		8.5.0.2
IKVM.OpenJDK.Remoting.dll		8.5.0.2
IKVM.OpenJDK.Security.dll		8.5.0.2
IKVM.OpenJDK.SwingAWT.dll		8.5.0.2
IKVM.OpenJDK.Text.dll		8.5.0.2
IKVM.OpenJDK.Tools.dll		8.5.0.2

Filename	Description	Version
IKVM.OpenJDK.Util.dll		8.5.0.2
IKVM.OpenJDK.XML.API.dll		8.5.0.2
IKVM.OpenJDK.XML.Bind.dll		8.5.0.2
IKVM.OpenJDK.XML.Crypto.dll		8.5.0.2
IKVM.OpenJDK.XML.Parse.dll		8.5.0.2
IKVM.OpenJDK.XML.Transform.dll		8.5.0.2
IKVM.OpenJDK.XML.WebServices.dll		8.5.0.2
IKVM.OpenJDK.XML.XPath.dll		8.5.0.2
IKVM.Reflection.dll		8.5.0.2
IKVM.Runtime.dll		8.5.0.2
IKVM.Runtime.JNI.dll		8.5.0.2
log4net.dll		1.2.15.0
Microsoft.Data.Edm.dll		5.6.3.62017
Microsoft.Data.OData.dll		5.6.3.62017
Microsoft.OData.Core.dll		6.9.0.62068
Microsoft.OData.Edm.dll		6.9.0.62068
Microsoft.Spatial.dll		6.9.0.62068
netchartdir.dll		6.0.0.0
Newtonsoft.Json.dll		6.0.8.18111
System.Spatial.dll		5.6.3.62017

These JAR files are converted using IKVM and packaged in WindwardReports.DLL.

Filename	Description	Version
asm-5.0.3.jarasm-analysis-5.0.3.jar	Saxon dependency	5.0.3
asm-commons-5.0.3.jar	Saxon dependency	5.0.3

Filename	Description	Version
asm-tree-5.0.3.jar	Saxon dependency	5.0.3
asm-util-5.0.3.jar	Saxon dependency	5.0.3
bcmail-jdk14.jar		1.38
bcprov-jdk14.jar		1.38
bctsp-jdk14.jar		1.38
commons-codec.jar	Used for uuencode & uudecodeing	1.9
commons-imaging.jar	Image processing	1.0
commons-lang.jar	Used by jfreechart	2.6
commons-io.jar	Used for HTML output	2.1
commons-logging.jar	Logging bridge	1.2
httpclient.jar	Read http files (datasources & import tag)	4.5.2
httpcore.jar	Read http files (datasources & import tag)	4.4.4
icu4j-59_1.jar	Saxon dependency	59.1
javax.mail.jar		1.4.4
javax.servlet.jsp.jar		2.2.5
javax.servlet.jsp-api.jar		2.2.1
javax.servlet-api.jar		3.0.1
jetty.jar		6.1.26
jetty-util.jar		6.1.26
jfreesvg.jar	Creates the SVG files for HTML reports	3.3
json-smart.jar	JSON dependency	2.3
json-path.jar	JSON query library	2.5.1
mail.jar	Saxon dependency	1.4
servlet-api.jar		2.5

Filename	Description	Version
tagsoup-1.2.1.jar	Saxon dependency	1.2.1
saxon9ee.jar	XPath 2.0 client	9.8
slf4j-api.jar	JSON dependency	1.7.25
slf4j-nop.jar	JSON dependency	1.7.25
WindwardReports.jar	Windward Engine	Engine Version
xerces.jar	Saxon dependency	N/A
xmlgraphics-commons.jar	Image output formats	2.3

Product End of Life Policies

This article details the past versions of Windward Studios products we support, and to what extent we support them. Please remember that our first recommendation is *always* to upgrade to the newest version of a product. If you are current on Support and using the latest version, you don't need to worry about any of this - this page is only relevant if you are using an older version of a Windward Studios product. If you have any questions, please don't hesitate to contact us.

Definitions

- Security issues: a bug that exposes a security vulnerability in your system
- Failures: an issue that causes an exception or program failure, and we cannot find a workaround
- Formatting and layout: your output does not match the template

Support

End of life dates are listed below for all versions except the current version. Our guidelines for the dates set are:

- formatting and layout bugs are handled for six months after the release date of the subsequent version
- exception and failure bugs are handled for 18 months after the release date of the subsequent version
- security issues are handled for five years after the release date of the subsequent version

For issues in an older version, we strongly recommend upgrading to the latest version. Version changes are generally not a significant change in existing functionality but more of a calendar event so an upgrade from version 11.1.37.0 to 11.1.38.0 is not much different from an upgrade from 11.1.38.0 to 12.0.0.0. You get new features while the existing features are unchanged. (The upgrade from version 9 to version 10 is an exception - that has significant changes in how select statements are specified.)

If you have a current Support contract and are using a no-longer-supported version, you can still submit Support requests to our Helpdesk or support phone line if the question is valid for newer versions as well. However, we will not answer questions specific to that version.

12.5 is a special case; at that release we eliminated using J# in .NET Report Engine. Therefore it is treated as a major version change.

Version	Formatting and Layout	Exceptions and Failures	Security Issues
1	1 November 2003	1 February 2004	1 August 2008
2	1 April 2005	1 July 2005	1 January 2010
3	1 April 2006	1 July 2006	1 January 2011
4	1 December 2007	1 March 2008	1 September 2012
5	1 September 2008	1 December 2008	1 June 2013
6	1 March 2009	1 June 2010	1 December 2013
7	1 September 2009	1 December 2010	1 August 2014
8	1 November 2010	1 February 2012	1 August 2015
9	1 March 2011	1 June 2012	1 December 2016
10	1 April 2012	1 July 2013	1 January 2017
11	1 October 2013	1 October 2014	1 April 2018
12.0	1 March 2014	1 March 2015	1 September 2018
12.5	1 December 2014	1 December 2015	1 June 2019
13	1 October 2015	1 October 2016	1 April 2020
14	1 February 2017	1 February 2018	1 August 2021
15	1 April 2019	1 April 2020	1 October 2023
16	1 October 2020	1 October 2021	1 April 2025
there is no version 17, 18, or 19 (now by year - version 20 is for 2020)			
20	present release		

End of life for version 20 will be announced when version 21 is released.

Product-Specific Notes

Report Designer Office Edition

Starting with version 16, .NET Framework 4.6.1 or later is required.

Java Report Engine

- Starting with version 12.5, Java Report Engine requires Java 1.6 or later
- Starting with version 6, Java Report Engine requires Java 1.4 or later
- Starting with version 16, Java Report Engine requires Java 1.8 or later

.NET Report Engine

- For version 12.5, the .NET Report Engine API was upgraded to use generics instead of arrays (see **[Upgrading to the 12.5 API]**)
- Starting with version 9.0 .NET Report Engine is now built with .NET 3.5 instead of .NET 2.0
- Support for .NET 3.5 ended in version 15.1
- Starting with version 16, .NET Framework 4.6.1 or later is required

Report Engine for RESTful

Starting with version 16, .NET Framework 4.6.1 or later is required.

Microsoft Office

Microsoft drops support for Office 10 years after the release date, e.g. Office 2003 support was ended by Microsoft in 2014. We add one year to that for our support; hence end of life for supporting Report Designer Office Edition on a version of Office is:

Version	Report Designer End of Support	Final Report Designer Version
Office 2000	1 January 2012	11.1
Office 2002	1 January 2014	12.5
Office 2003	1 January 2015	13.1
Office 2007	1 January 2019	15.2

Version	Report Designer End of Support	Final Report Designer Version
Office 2010	1 January 2022	
Office 2013	1 January 2025	
Office 2016	1 January 2028	

Earlier Version Requirements (v12 and Earlier)

Report Designer for Office Edition

In version 12.5 Windward removed the requirement for J# and has replaced it with IKVM. In Versions 12.5 and later, J# is no longer required.

.NET Report Engine

In version 12.5 Windward removed the requirement for J# and has replaced it with IKVM. In Versions 12.5 and later, J# is no longer required.

- See more details on the conversion at **[Upgrading to the 12.5 API]**
- Version 9.0 and later .NET Report Engine is compiled under .NET 3.5 instead of .NET 2.0.

.NET Report Engine requires these drivers for all versions 12.0 and earlier:

- .NET Framework 3.5 - (You probably already have this installed)

Java Report Engine

Java Report Engine requires:

- Java 1.4 or later (For Java Report Engine v6 and later, the Java Report Engine is compiled for Java 1.4, and later versions of Java.)
- Oracle JAI Image I/O library (Only if you are using Java 1.5 or earlier)
  - Linux JRE Install
  - Solaris SPARC JRE Install
  - Solaris x86 JRE Install
  - Windows JRE Install
  - JAI Image I/O download page

J# Information

Both Report Designer and .NET Report Engine require the J# 2.0 redistributable package from Microsoft for versions 12.0 and earlier. This must be installed after .NET 3.5 is installed as it is an extension of .NET 3.5. It must be installed before installing Report Designer or .NET Report Engine.

You must install J# version 2.0 - Second Edition, released May 2007.

Download only the J# appropriate for your O/S

- J# redist 2.0 (x86) - 32-bit Windows (for Windward version 12 and earlier only)
- J# redist 2.0 (x64) - 64-bit Windows (for Windward version 12 and earlier only)
- J# redist 2.0 (IA64) - Intel Itanium Windows (if you have this, you know it -- also for Windward version 12 and earlier only)

Sending a Test Template Using Sample Data

If you have trouble with a Template that you would like Windward Support Staff to troubleshoot, please send us a simplified version to test. Sending us a simple Template can help us resolve your problem quickly.

How to Simplify Your Test Template

Try the following in a copy of your original Template:

- Delete all pages and Tags except for the ones that you know are causing trouble.
- Highlight or otherwise mark the Tag(s) you think are causing trouble so we can quickly locate them. Use simple things like Bold, Underline, text color, and highlighting.

What About Data Sources?

If you are using a data source that we can't access (like a SQL server, or a data source that has private data that you cannot share), try the following:

- Send us a version of your data source using test data. Don't forget to include login credentials if those are needed.
- Use one of Windward's public data sources to reveal the problem. Here is a list of data sources to use - our public data source credentials are listed in the articles per data source.
- In each of your Tags, select approximate data from our public data source that replace your organization's data

Tell Us Which Product Versions You're Using

We will test the template in the *version of Report Designer* and the *version of the Report Engine* with which you are experiencing trouble.

- We will compare the results with tests that we run in our most up-to-date versions, or in the version in which you are specifically experiencing trouble.
- In some cases we may suggest that you upgrade to a newer version.

Unsupported Versions

We do not support the these product versions.

Support Tickets - What to Expect

Your questions are extremely important to you, so they are important to us!

- Please read through "What our Support Team needs from You" below, to speed up your resolution time.
- If you have other questions, please refer to the FAQs section below.

What our Support Team Needs from You

If you have submitted a Support ticket, we may not be able to begin testing your Template until we have more information, or a simple version of your template to test. To reduce your resolution time, please make sure your ticket has the following information in the body or the text or as an attachment.

Your License Key (Existing Customers)

If you're not sure how to find your license key, please see All About Windward Licenses.

Your Product Versions

- The version of your Report Designer
- The version of your Report Engine

A Simple, Testable Template

- It is helpful if we are able to open and generate output from the Template immediately without having to find and replace Tags (e.g. Import Tags) that link to items not available in our environment.
- If our Support staff have to make many adjustments to your Template in order to generate output in our environment, we may provide hints on simplifying Templates and ask you to send us another version.

An Error Log

Logging for Report Designer

Navigate to the Options section of the AutoTag manager Tab in your Office ribbon. In the Options popup, go to the Advanced tab and click the "Create Log File" button. See this article for more information.

Logging for the Report Engines

.NET Report Engine

Java Report Engine

[Report Engine for RESTful]

One Topic Per Ticket

To ensure our Support doesn't overlook any customer concerns while working on a Support ticket, if your ticket contains more than one issue or question, we will create new tickets for those additional topics. We will make sure to answer all of your questions in the correct ticket.

FAQs

What Kind of Help is Provided by the Windward Support Team?

On the Support Desk, we can provide guidance, reference material, and respond to exceptions in the software. However, designing your Template for you, and helping you to write code or queries, falls outside the scope of Support, and requires a separate Professional Services Contract. If you would like to arrange a Professional Services contract, let us know and we can put you in touch with your Account Manager.

Why do You Need My License Key?

We provide support to many different kinds of groups. You may be the customer of a customer, or you may be working with a third party solutions contractor. Because of contractual obligations, we must make sure that we align your ticket with the correct parent group, and only you can tell us who you're working with.

Why do You Need to Know the Versions of Windward Products I'm Using?

The version of the Report Engine must be equal to or newer than the version of Report Designer you're using. Otherwise, you could design a Template with a feature available in Report Designer that an older version of the Report Engine would not know how to interpret, with unpredictable results.

When Will My Ticket be Answered?

We do our damnedest to answer tickets in as timely a manner as possible, in the order they are received. Your ticket may be delayed because we are waiting on more information from you, but we will work on your ticket as soon as we can.

How Long Will It Take?

That's entirely dependent on the complexity of the Template, how much time our Support staff spends trying to get your Template to generate output in our environment, and whether we find something that requires a fix. Some tickets are solved within minutes. If you have a really tricky question that we need to ask Development, your ticket may be delayed due to Development's schedule.

How Do I Get Updated License Keys?

In order to use the latest version, you will need to log on to the Windward Store and retrieve your updated licenses. See this article for details.

If you have forgotten your user name and password, there is a button on the login page that you can click to have the store email your user name and password.

Why Am I Asked to Separate Questions in My Ticket(s)?

There are two basic reasons, and the bottom line for both is to answer your questions quickly and effectively:

- The first is that we can lose track of each request if there are many listed in one ticket; we want to make sure that all of your questions are answered effectively.
- The second is that it allows us to assign questions to specific developers if needed. If there are many issues in one ticket, the areas in question can be unrelated and need to be addressed by different developers in different departments. Imagine that you had questions that seemed related, but upon inspection, we find that one is related to Template layout and

another is related to your Report Engine -your ticket could be in limbo for days after waiting to be addressed by one department, reassigned to Support, reassigned to another department, and back to Support again.

Upcoming Release Schedule

Here is a schedule for our upcoming releases.

Version Release	Target Release Date	State	Functionality
20.3.0	December 15, 2020	In Progress	20.3.0 Targeted Development (Unreleased
20.2.1	November 10, 2020	Released!	Version 20.2.1 Features
20.1.1	August 6th, 2020	Released!	Version 20.1.1 Features
20.0.0	Released March 31, 2020	Released!	Version 20.0.0 Features
16.7.2	Released January 27, 2020	Released!	
16.6.0	Released on November 4, 2019	Released!	
16.5.0	Released on September 30, 2019	Released!	Version 16.5.0 Features

Version 16 Perpetual License Update

Beginning with Version 16, there are changes to PDF output generated with Development and Test perpetual licenses. This article documents those changes.

Perpetual License Types and Restrictions

Here are the different license types and license restrictions by type:

❗ Note that PDF output generated with Test and Developer licenses is locked, but printable.

Standard Engines

- Limited by cores/threads; can be adjusted
- Produce unlimited reports
- No Watermark

Test Engines

- Limited by cores/threads; can be adjusted
- Produce unlimited reports
- Watermarked

Developer Systems

- Default four cores; can be adjusted as needed
- Limited to 250 reports/day
- Watermarked

Developer Systems with Report Designer

- Same as above except Report Designer is included
- A report produced from the attached Report Designer will be watermarked.

Report Designer

- No watermark
- All Report Designers are the same - they are not customizable

Windward Software Installers

Bellow are links to the current Windward Core release and previous Windward core releases, Support Releases for each version and Release Notes for each version.

The Release Notes for each version are posted in the directory for that version.

The most recent Windward Core release is 20.2.1.0

Version 20

Version 16

Version 15

Version 14

Version 13

Version 12

Version 11

Version 10

Version 9

Version 8

Version 7

Version 6

Version 5

Version 4

Version 3

Version 2

Version 1

Javelin

Windward Performance Analysis

This article describes how to run performance tests using either our .NET or Java Report Engines. Windward's engines are fast, very fast. The number one thing that we hear from our clients moving to Windward from competing solutions is the performance gain they receive. We realize that running performance tests under ideal conditions are not always reflected in real environments. We routinely update the performance PDF and this article with the latest versions of our Report Engine benchmarks. However, we also provide this performance test as a part of our Report Engine software package in both demo and production versions. We encourage you to test the Windward Report Engine in your own environment to see how it performs before you purchase.

The steps below will outline how we test our product internally and how you can run these tests for yourself in your own environment. As always, we are curious to know your results and are always interested to see how our product performs in different environments. If you would like assistance running these tests and are willing to share your results then please feel free to contact us at support@windwardstudios.com and our Marketing team will reach out to you.

Download the Full Report here: Windward Engine Performance.pdf

What does this Article Cover?

- **The Numbers** - the latest performance analysis report for the current version of the Windward Engine
- **The Measurement System** - Here we will explain how we run our tests including hardware specifications, template size, output size and data source details.
- **How do I run the Tests?**

The Numbers

The below tests all use the files in WindwardPerformance.zip.

- The **OData** Server contains the Northwind database. The database server is located in the cloud.
- The **JSON** file is a 1.27M file which contains a dump of the SQL Server Northwind database. The file is on the local machine.

- The **SQL** Server contains the Northwind SQL database. The database server is located on the test machine.
- The **XML** file is a 1.35M file which contains a dump of the SQL Server Northwind database. The file is on the local machine.
- The template is 2 pages long and the report it generates is 17 pages long and is run 50 times.

The Measurement System

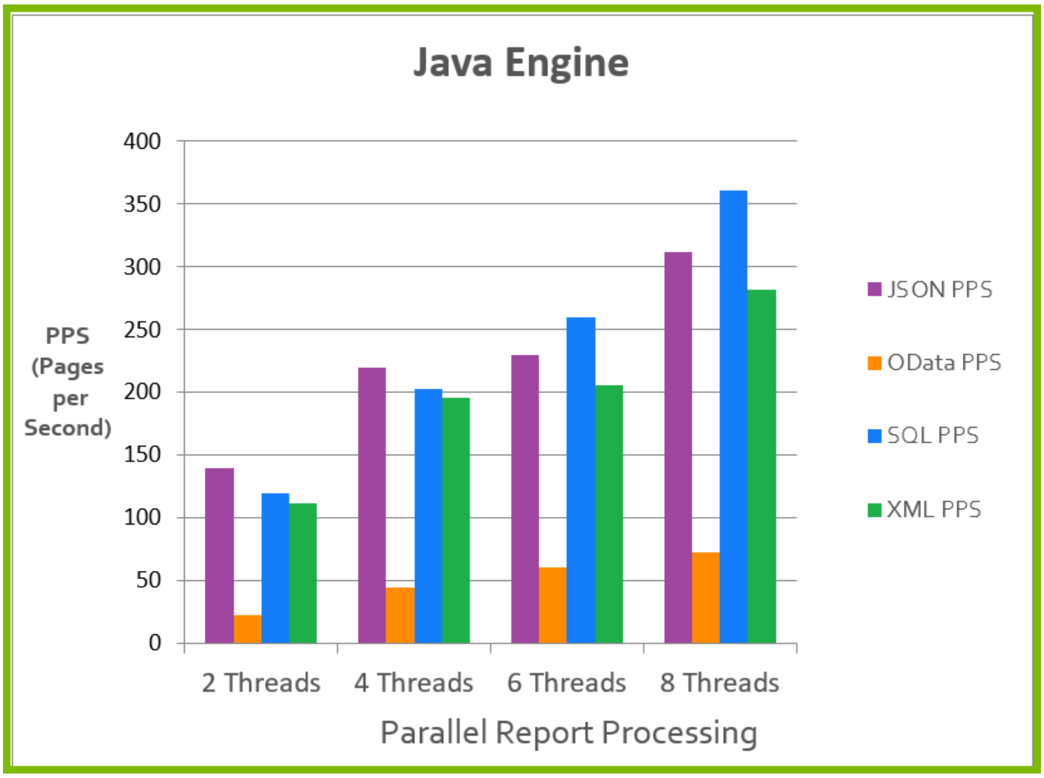
For this latest version of the Windward Report Engine, we tested the performance by creating a report 50 times and averaging the result. Performance results (number of pages created per second [PPS]) depended on which engine is used, what sort of data source is used, and the number of simultaneous threads called by the Windward application.

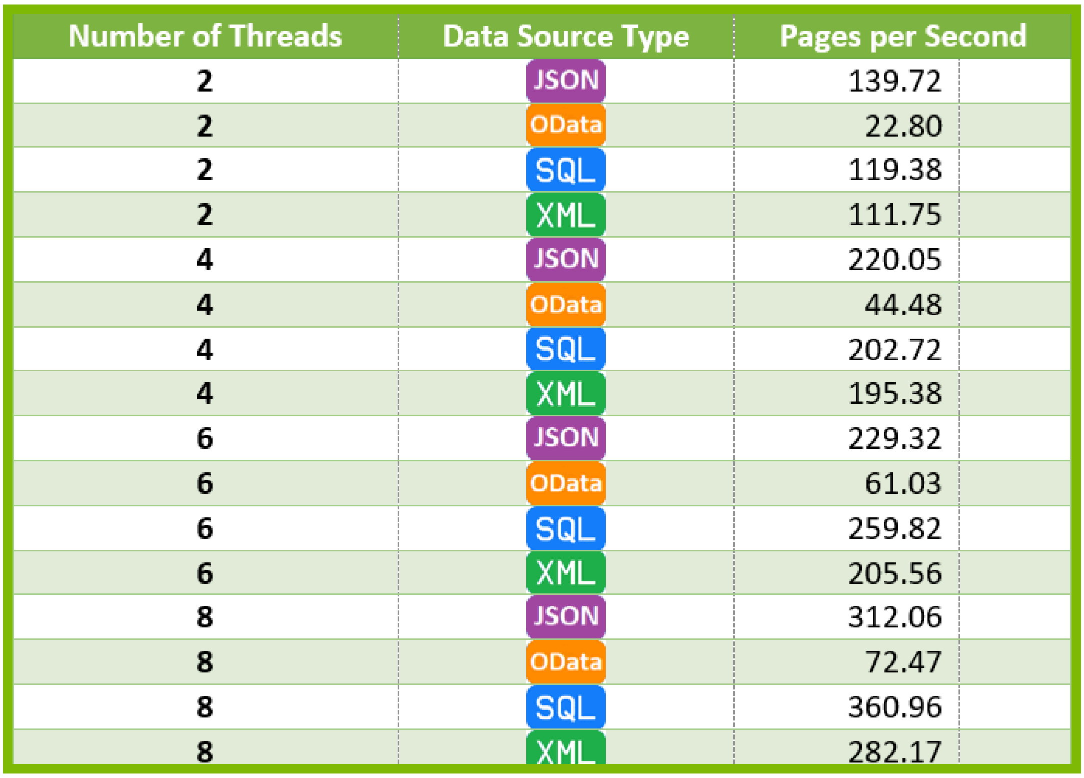
- **Processor**: Intel® Xeon® E5520 @ 2.27GHz 8 virtual processors
- **Memory**: 32.0 GB
- **Operating system**: Windows Server 2016 Standard 64-bit
- The Windward AutoTag template is a Microsoft® Word document two pages long. It contains text, images, headers, footers, multiple loops, and equations.
- The generated report is a 17-page Word document (DOCX). The number of pages generated per second is not affected by the output format. Both DOCX and PDF reports were generated in the same amount of time. The report is run 50 times for each test.

The Windward Java Report Engine

For each test of the Java Report Engine, we ran with four data sources (JSON, OData, SQL, and XML) and four different thread counts (2, 4, 6 and 8 threads). The exact results are listed in the table below.

*Running on an Intel® Xeon® E5520 @ 2.27GHz with 32 GB RAM on Windows Server 2016 Standard (64-bit)



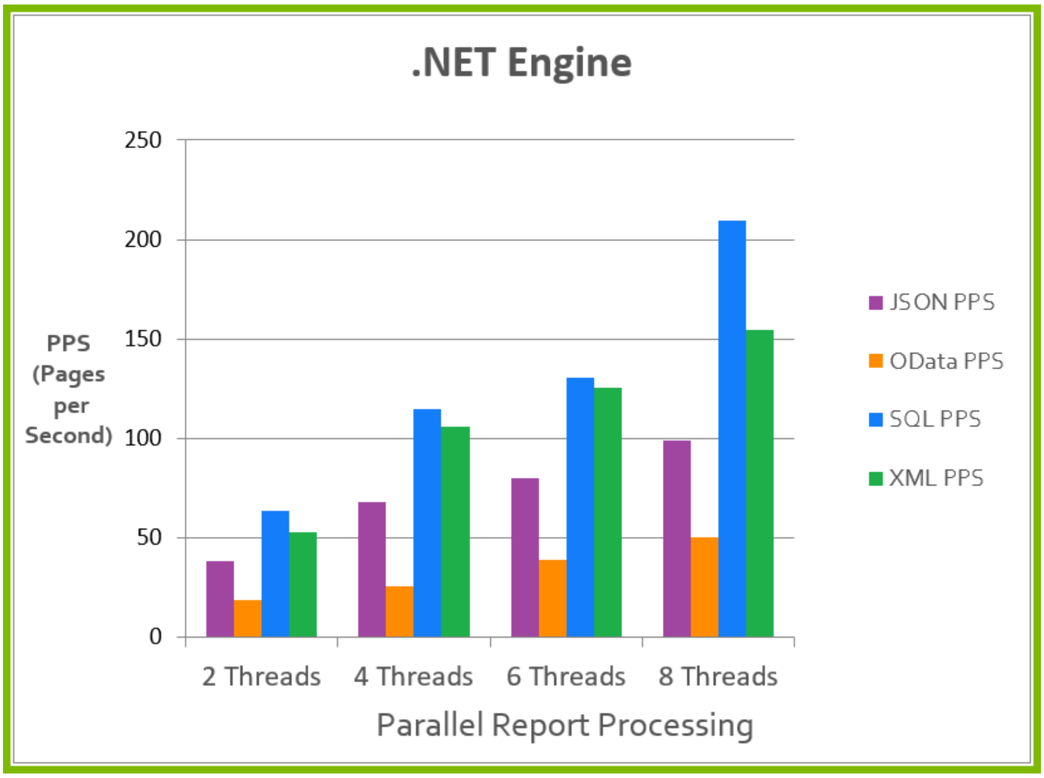


Number of Threads	Data Source Type	Pages per Second
2	JSON	139.72
2	OData	22.80
2	SQL	119.38
2	XML	111.75
4	JSON	220.05
4	OData	44.48
4	SQL	202.72
4	XML	195.38
6	JSON	229.32
6	OData	61.03
6	SQL	259.82
6	XML	205.56
8	JSON	312.06
8	OData	72.47
8	SQL	360.96
8	XML	282.17

The Windward .NET Report Engine

For each test of the .NET Report Engine, we ran with four data sources (JSON, OData, SQL, and XML) and four different thread counts (2, 4, 6 and 8 threads). The exact results are listed in the table below.

*Running on an Intel® Xeon® E5520 @ 2.27GHz with 32 GB RAM on Windows Server 2016 Standard (64-bit)



Number of Threads	Data Source Type	Pages per Second
2	JSON	38.39
2	OData	18.92
2	SQL	63.73
2	XML	53.04
4	JSON	67.73
4	OData	25.81
4	SQL	114.67
4	XML	105.81
6	JSON	80.22
6	OData	39.05
6	SQL	130.76
6	XML	125.70
8	JSON	98.89
8	OData	50.22
8	SQL	209.70
8	XML	154.53

Testing Notes

These Report Engine performance tests *assume the worst case performance* in several aspects, so your performance could easily be faster.

In this test:

- We open the template each time the Engine generates a report. When you run the same template in bulk through the Windward Engine, it scans the template one time and caches it. This results in a speed increase as the time to read the template are removed from each successive run.
- The data sources are loaded again each time, which may not be necessary in your use case.
- Both DOCX and PDF reports were generated in the same amount of time.

A Note about processor thrashing

We ran these tests on a 4 core system. Typically we recommend that you do not exceed 2 times the number of total cores on your target system. You will notice in most of the tests above that performance starts to plateau after 8 threads (2 x 4 cores = 8 threads). This is a result of a phenomena called processor thrashing which is a state where the processor performs less *productive* work and more work *swapping* between all *simultaneously running processes or threads.* This yields diminishing returns over time as the processor is spending more time trying to manage all simultaneous processes than actually processing them.

As always, Windward recommends that you run our performance tests with different thread values in order to find the sweet spot for your system. The 2 times total system cores as threads is more of a guideline than a rule but we see this proven above in our standard testing.

How Do I Run the Tests?

The *-performance:1234* option is available in the sample RunReport application shipped with both the .NET and Java engines. We strongly recommend that you run this program using your sample sets of templates and data on the server you will use.

One of the key factors to the performance of generating reports is based on the hardware being used. The greater the processing power and the larger memory your system has will translate into improved performance of generating reports. That being said, here are some important stats to know in regards to generating reports.

There are a couple of generalizations we feel safe making about Windward Reports:

- Its performance time is linear. It will take 10x longer to run 100,000 reports than it takes to run 10,000 reports.
- For every CPU core, you should generally have a maximum 2 threads generating reports.
- Its memory usage is constant. Running 1,000 reports uses the same amount of memory as running 100,000 reports.
- DOCX, XLSX, PPTX, PDF, HTML, and TXT reports all take about the same time.

- File I/O is a big part of the time required.
- Logic commands are the most expensive computationally. IF/ELSE and SWITCH/CASE statements, when used excessively, will slow down your report generation (this is only noticeable if you have more than 150 tags of this type in a single template)

Performance Analysis Commands

Java Engine

1. Navigate to the directory where you installed the Java Engine

- **Windows**: C:\Program Files (x86)\Windward Studios\Windward Java Engine\
- **Unix**: path where you extracted the Windward JAR files

2. Make sure all Windward JAR files are present in your CLASSPATH

- **Windows**: this is done automatically, no further action needed
- **Unix**: you will have to alter your CLASSPATH to include all JAR files in the Windward /lib directory

3. Change to the "test" directory

- **Windows**: C:\Program Files (x86)\Windward Studios\Windward Java Engine\test
- **Unix**: /Your_Windward_Path/test

4. Run the command below

Command

```
net.windward.XmlReport.RunReport TEMPLATE_NAME OUTPUT_NAME DATA_CONNECTION
–performance:NUM_REPORTS –threads:NUM_THREADS
```

.NET Engine

Navigate to the directory where you installed the .NET Engine (C:\Program Files (x86)\Windward Studios\Windward .NET Engine\)

Change to the "test" directory

In this directory is the RunReport.exe program, a precompiled version of a command line implementation of the .NET Engine

Run the command below

Command

```
RunReport.exe TEMPLATE_NAME OUTPUT_NAME DATA_CONNECTION
–performance:NUM_REPORTS –threads:NUM_THREADS
```

Example

This example will run the template 10,000 times using 4 simultaneous threads to generate the reports in parallel.

```
RunReport.exe "Looping Variable Invoice Sample - XML.docx" testreport.docx -xml:XML
"Southwind - Data.xml" -performance:10000 -threads:4
```

RunReport.exe command options

When executing the RunReport program from the command line of the Windward Engine with no parameters you will see the following output below will all options that can be used.

```
usage: RunReport template_file output_file [-basedir path] [-xml xml_file | -sql
connection_string | -sforce | -oracle connection_string | -ole oledb_connection_string]
[key=value | ...]

    The template file can be a docx, pptx, or xlsx file.

    The output file extension determines the report type created:

      output.csv - SpreadSheet CSV file

      output.docm - Word DOCM file

      output.docx - Word DOCX file

      output.htm - HTML file with no CSS

      output.html - HTML file with CSS

      output.pdf - Acrobat PDF file

      output.pptm - PowerPoint PPTM file

      output.pptx - PowerPoint PPTX file

      output.prn - Printer where "output" is the printer name

      output.rtf - Rich Text Format file

      output.txt - Ascii text file

      output.xhtml - XHTML file with CSS

      output.xlsm - Excel XLSM file

      output.xlsx - Excel XLSX file

    -basedir - will set the base directory to this.

    -data filename.xml - will write data.xml to this filename.

    -embed - will embed data.xml in the generated report. DOCX, PDF, PPTX, & XLSX only.

    -launch - will launch the report when complete.

    -performance:123 - will run the report 123 times.

       output file is used for directory and extension for reports
```

-cache - will cache template & datasources, will write output to memory stream. Only used with -performance.

-threads:4 - will create 4 threads when running -performance.

-verify:N - turn on the error handling and verify feature where N is a number: 0 (none) , 1 (track errors), 2 (verify), 3 (all). The list of issues is printed to the standard error.

-version=9 - sets the template to the passed version (9 in this example).

encoding=UTF-8 (or other) - set BEFORE datasource to specify an encoding.

locale=en_US - set the locale passed to the engine.

pod=pod_filename - set a POD file (datasets).

username=user password=pass - set BEFORE datasource for database connections.

The datasource is identified with a pair of parameters

-json filename - passes a JSON file as the datasource

filename can be a url/filename or a connection string

-odata url - passes a url as the datasource accessing it using the OData protocol

-sforce - password should be password + security_token

-sharepoint filename - passes an xml file as the datasource reading it with the SharePoint FBA protocol

-xml filename - XPath 2.0 passes an xml file as the datasource

-xml xmlFilename=schema:schemaFilename - passes an xml file and a schema file as the datasource

filename can be a url/filename or a connection string

-xpath filename - uses the old XPath 1.0 datasource.

-xml xmlFilename=schema:schemaFilename - passes an xml file and a schema file as the datasource

filename can be a url/filename or a connection string

-db2 connection_string ex: server=localhost;database=SAMPLE;Uid=test;Pwd=pass;

-excel connection_string ex: Provider=Microsoft.ACE.OLEDB.12.0;Data Source=c:\test1.xlsx;Extended Properties="Excel 12.0 Xml;HDR=YES"

-mysql connection_string ex: server=localhost;database=sakila;user id=test;password=pass;

-odbc connection_string ex: Driver={Sql Server};Server=localhost;Database=Northwind;User ID=test;Password=pass;

-oledb connection_string ex: Provider=sqloledb;Data Source=localhost;Initial Catalog=Northwind;User ID=test;Password=pass;

-oracle connection_string ex: Data Source=localhost:1521/HR;Persist Security Info=True;Password=HR;User ID=HR

-sql connection_string ex: Data Source=localhost;Initial Catalog=Northwind;Integrated Security=SSPI;

-postgresql connection_string ex: HOST=localhost;DATABASE=pagila;USER ID=test;PASSWORD=test;

if a datasource is named you use the syntax -type:name (ex: -xml:name filename.xml)

You can have 0-N key=value pairs that are passed to the datasource Map property

If the value starts with I', F', or D' it parses it as an integer, float, or date(yyyy-MM-ddThh:mm:ss)

If the value is * it will set a filter of all

If the value is "text,text,..." it will set a filter of all

Windward Studios Privacy Policy

Attached is our Windward Studios Privacy Policy.

PDF privacy-policy-pdf-english.pdf