



(800) 833-4459 | www.wireready.com | sales@wireready.com

StormReady32 Report Generator

WireReady NSI now has an external program that looks at the StormReady32 database, and generates printed, CSV or SQL database reports and updates.

Needed files:

- StormReadyReportGenerator.exe
- StormReadyReport.ini
- The StormReady32 database

The program uses the INI file to generate a report and then terminates. The report columns, widths, and search parameters for each search you want to support are all stored in the "stormreadyreport.ini" file that is located in the same directory you run the StormReadyReportGenerator.exe from.

The main feature of this system is that users can have single click printed reports, without needing to operate the StormReady32 program from their desktop. For automated tasks, such as generating files for stations or the web site, the program can be setup as a shortcut and set to run through Windows Task Scheduler.

In order to use this program it MUST be run from a shortcut on the desktop, the Run command, or setup as a job in Task Scheduler. You cannot double click on the EXE itself. That is because you must include the name of the report whenever you run the program. If the program can't find a report name, it pops a warning screen telling you the correct format.

The format to run this in a short cut is:

[drive]:\path\StormReadyReportGenerator.exe [report name]

The INI can have every configured report as a group, all contained in the same INI file. You can distribute the EXE to separate computers or drive locations and they in turn can each have their own INI, or you can have one INI, one EXE and let people call it from the network. WireReadyNSI recommends that you backup the INI file to preserve it in case a user edits it.

The StormReadyReport.ini that we have included has reports set up for both sending to a printer and to create a CSV file.

The below listed groups exist in the included ini file. The group heading is the name of the report that will need to be included in the target path to generate that report.

```
[PRINT-ALL-CLOSINGS]
[PRINT-ALL-CLOSINGS-NO-REPEAT]
[PRINT-ALL-CLOSINGS-TODAY]
[PRINT-ALL-CLOSINGS-NO-REPEAT-TODAY]
[PRINT-ALL-CLOSINGS-TOMORROW]
[PRINT-ALL-CLOSINGS-NO-REPEAT-TOMORROW]
[CSV-ALL-CLOSINGS]
[CSV-ALL-CLOSINGS-TODAY]
[CSV-ALL-CLOSINGS-TOMORROW]
[SQLServerWeb]
```

An example shortcut to run one of the reports is:

```
W:\wire\stormreadyreportgenerator.exe print-all-closings
```

The stormreadyreport.ini lines are:

Group Name The name of the report, enclosed in square brackets. Spaces are not allowed in the group name.

DataPath = The location of the StormReady32 database, including drive and directory location

OutputFilename = The path and filename of the file to be created when using CSV as the Output. Drive letter and "/" paths are allowed.

Output = Where the report will be sent to: Printer, CSV, or SQLserver.

WHERE = Standard access/SQL search query syntax. Only one WHERE clause is supported per report

ReportField1 = The database field name for the first column of the output report.

ReportFieldLength1 = The number of characters to include in the first field. If the field is shorter than the data, the data will be truncated to fit.

ReportFieldSort1 = Indicates if the first field is sorted (True/False)

ReportFieldRepeat1 = Indicates if the value in the first field will be repeated (True/False)

ReportFieldStringValue1 = Indicates if the field will be surrounded with quotes (True/False). This line defaults to True if not present, and only needs to be added to suppress the quotes.

DestinationDBConnectionString = The path, IP, user, password, and table names that match with the web server hosting the database. Only used with reports generated to SQLserver.

You can configure as many columns of data in a report as are needed. The format of the lines is *ReportField#*, the # being an ordinal starting with 1. The first column is 1, the second is 2, etc. Don't skip a number or use the same number twice. Each ReportField has a *Length*, *Sort* and a *Repeat* value that correspond to that number.

When the Output line is set to PRINTER, a print dialog window opens.

When it is set to CSV, the program will generate the CSV with the path and filename entered.

When it is set to SQLserver, it talks to the SQL database server specified in the connection string.

The *ReportFieldRepeat* line is useful for printing. In CSV format, you generally want to repeat. But if someone is printing, the report is often easier to read if all the consecutive closures don't say CLOSED line after line. The same is true for the City/Location field if you are including that as a column.

Sequel

There are hundreds of SQL comparators and calculators you can use to generate reports. The INI file provided uses "DateDiff" and the "d" operator for Day. Also used is the "NOW()" reserved name which gets the system date/time for the comparison.

DateDiff can calculate based on seconds through years. The Keys are:

- yyyy Year
- q Quarter
- m Month
- y Day of year
- d Day
- w Weekday
- ww Week
- h Hour
- n Minute
- s Second

To calculate anything that was more than a day old, the line would read:

```
WHERE = DateDiff('d',[Event Kill Date], Now()) > 1
```

To calculate anything that was more than an hour old:

WHERE = DateDiff('h',[Event Kill Date], Now()) > 1

To calculate anything that was for tomorrow:

WHERE = DateDiff('d',[Event Kill Date], Now()) = 1

To calculate anything that was for today:

WHERE = DateDiff('d',[Event Kill Date], Now()) = 0

Each one of these expressions can be **AND** or **OR** in the same WHERE clause. Only one WHERE clause is supported per report.