Analysis of report framework for Play’n GO
Abstract

Play’n GO uses a reporting framework where their customers (operators) can access game related and financial information. These reports are very important for the operators since they use them all the time for statistics, marketing and financial settlement. The reporting framework is now getting old and needs to be updated or replaced.

This report will handle the analysis of the current framework, an updated framework, new frameworks and the option of creating a framework by themselves.

The result was that it would be easier for them to make a framework of their own, rather than to buy and try to adapt an already existing one.

Keywords: Report framework, Analysis, Integration of framework

Sammanfattning

Play’n GO använder ett rapporteringsramverk där deras kunder (operatörer) kan komma åt spelrelaterad och ekonomisk information. De här rapporterna är väldigt viktiga för operatörerna då de hela tiden används för statistik, marknadsföring och ekonomiska beslut. Rapporteringsramverket håller nu på att bli gammalt och behöver uppdateras eller bli ersatt.

Den här rapporten kommer att beröra analysen av det nuvarande ramverket, ett uppdaterat ramverk, helt nya ramverk och alternativet att skapa ett ramverk själva.

Resultatet blev att det skulle vara enklare för dem att göra ett eget ramverk, hellre än att köpa och försöka anpassa ett redan existerande.

Nyckelord: Rapporteringsramverk, Analys, Integration av ramverk
Preface

This report was written in collaboration with Play’n GO and Linnaeus University and I would like to thank Play’n GO for the opportunity of writing my thesis with them.
Appendix C

C.1 How to create a report in Syncfusion

Appendix D

D.1 How to create a report with Windward
1 Introduction

A report framework is used by companies for many different reasons. It is most common to use its capabilities for datamining in a company’s private database and generate different reports from that information.

1.1 Background

The current report framework that Play’n GO are using is still working, but it is old and do not have a lot of the new modern features that new programs have. The company has their own web component, called GATweb, that they would like to use instead. In the component, they want a specific place for all the report handling. Creating a new report should be easy and fast.

1.2 Previous research

The company has started to build a new site for their customers and have a working dashboard, but at the moment no reports. On the internet there are however plenty of guides to “finding your new report framework”. There are a lot of different Business Intelligence options out there for many different needs.

1.3 Problem formulation

Play’n GO uses an old reporting framework and want to change to a new and more updated version. The job will be to analyze existing frameworks and see if any of them will fit the company’s needs or if it’s more efficient for them to build a framework of their own.

They want the framework to be able to handle:

- To be hosted in the same environment as future products (Windows)
- Auto deploy with future products
- Both visualizing, tables and filter possibilities
- Export to well-known formats (excel, csv, pdf, etc...)
- A simpler way to create reports
- Easy to maintain reports and the framework itself
- Integration in products and then with similar look and feeling

If they will make a framework by themselves, how should it be done?

- What components should be used?
- What choice of architecture and design?
- Cost?
- How would it be integrated with other products?
• How much maintenance would the code require?

1.4 Motivation
Play’n GO sells their product to operators. The operators in their turn sell their products to their customers. The operators that sells a product wants to know how it is received by the buyer. This is done by directly asking them or by gathering data on how the products are used. When it comes to gathering data, the company wants a good report framework to handle all the data. The framework should be able to handle big amounts of data, should be able to sort it in various different ways and be able to show graphics. Companies are developing more and more for the web and need report framework that works and is using the full potential.

1.5 Research Question

<table>
<thead>
<tr>
<th>RQ1</th>
<th>Is there any report framework currently on the market that lives up to Play’n GO’s demands (see Problem formulation) and in that case, can it be implemented and integrated into the current system?</th>
</tr>
</thead>
<tbody>
<tr>
<td>RQ2</td>
<td>If not – How would be the best way of implementing one by themselves?</td>
</tr>
</tbody>
</table>

The thesis will give the company a good analysis of what their options are and a recommendation.

1.6 Scope/Limitation
This thesis will handle an analysis of report frameworks for Play’n GO. Since the company only wants a report tool, all complete Business Intelligence programs are ignored unless it is possible to only buy the report part. Also no Big Data analysis programs has been looked into.

From the alternatives left, the analysis has been on; one update on the current framework, three commercial products and a suggestion on how to implement one by themselves. The analysis has been made with the written question in mind to keep a steady frame for the thesis.

1.7 Target group
The analysis could be useful for all companies that use a report framework, if they are thinking about changing the program that they currently are using or change to a web based solution.
1.8 Outline

Chapter 2 is Method and will go through the details of how the analysis was done. After that the results will be presented in chapter 3. Here the result of the analysis of the different frameworks are presented and has sub-titles 3.1-3.6.

Chapter 4 will present an analysis of the thesis and Discussion will follow in chapter 5. Chapter 6 will be the conclusions of the thesis and after that the list of references are presented.

The thesis will end with Appendix A-D that presents guides on how to create reports with the different tools that have been under investigation.
2 Method

In the following sections there will be an explanation on how the analysis has been done.

2.1 Scientific Approach

In this thesis an inductive approach will be used and all the data will be empirical data. An analysis on different report frameworks will be made with a qualitative method.

2.2 Method Description

The analysis will be made with the help of the different company’s tutorials, trial versions and documentation. Small implementations may be necessary and will in that case be made with the tools offered by the company that is currently being analyzed.

2.3 Reliability and Validity

An analysis that had more time would have had the opportunity to look into many more frameworks and to make a much more detailed analysis of those that have been selected. The conclusions and results are taken from the information that answers the research questions.
3 A report framework

This chapter will explain in general how a report framework works and how it can be used. Reporting is usually a part of a complete Business Intelligence (BI) package where the program also makes for example spreadsheets, data mining, dashboards and data warehousing. [1] This thesis is about the reporting parts only.

3.1 What is a report?

A report is a document used to present important information in an objectively way. It can show financial information, sales, costs and basically anything a company can gather data on. This is presented in a way that, usually, is easy to understand by everyone. Graphs and tables are often used to make the data more graphic.

3.2 What are reports used for?

A report is used to show different people how different parts of the company or the services are doing over a specific period of time. This can later be used to see if they need to change something, what is working and what they can do better. A report is a very important part of making new business decisions and in what direction a company could go.

3.3 What does the framework do?

To get all the information that is needed to present a report, an analysis of the information in the database needs to be made. Assuming a medium to large company, to do this manually would be a tedious and slow job. The information would probably be old and slightly irrelevant by the time the report was done.

That is where the framework comes in. The framework is there to help with all the database and presentation work. Connect the program to a database. The program will in most cases have a wizard to help with the setup and adding all the connection strings that is needed. After the connection is done, the layout job begins. The user then creates the template for the report, with all the graphs, tables and text that is required in the specific case. The frameworks then use the connection to the database and automatically adds the requested data in the correct fields. The report is then done and can be printed.

Common features in the framework is:

- Database connection (a must in one way or another)
- Exports to different formats
- Graphs and tables
- Templates
- Integrations
- To a program (for example Visual Studio)
- Code to integrate to a company’s own applications

Scripts

The framework should be easy to use and in the market now, there is a trend of drag-and-drop frameworks. This is so that the people working with reports should not need to be able to code to make the reports work. If a graph is needed, just drag a “graph” to the correct location and add the correct data to it and the program will handle the rest.
4 Results

This chapter will go through the current report framework and the different new frameworks that has been up for study during this thesis. The demands of the company, found in Problem formulation, will be the foundation of the analysis and the research questions will be answered in the discussion. In the cost for all these alternatives, the mandays needed for the implementation is just an estimation and they can be red in the discussion. One manday is 8 hours and the estimations has been done with an employee who has worked a lot with the GATWeb.

4.1 Telerik Reporting - Current report framework

The company is currently using Telerik Reporting Q2 2012 (6.1.12.611) and C# version 2.0.50727. The reports are created in Visual Studio, using C#, and are then used in their own program Game Account Toolkit (GAT). GAT, see image 3.1 below, brings information from the database when a request is made and displays it as the predefined report. The documentation on old versions on Telerik is very hard to get, which makes the process of learning difficult.

Image 4.1 GAT

The product is used in Windows but cannot be used online with the current versions. It has tables, filter opportunities and visualization during creating. The most common formats to export to is covered, such as pdf, csv, Excel, Rich Text Format, tiff and Webb Archive.
A new report is created if a customer has a special need that is currently not covered. The process is plain C# coding, but code from previous reports can often be reused. A new Stored Procedure (SP) to the database might need to be created for the new report.

Telerik offers a free program that can be used to create reports, Telerik Report Designer. The old version of the program could not be tested, since the newest version is the only one available now. A comparison will however be made in part 3.2, since the versions of Telerik Report Designer and Telerik Reporting then is compatible.

The maintenance is currently nonexistent and reports are only changed if absolutely necessary. All the logic calculations are placed in the database that is updated regularly.

GAT is not responsive in any way, so reports that are created in the program are difficult to look at on smaller screens since they do not adapt to the screen size.

4.2 Telerik Reporting - Updated framework

The updated framework is Telerik Reporting R3 2016 SP1 (10.2.16.1025) [2]. This framework needs at least .NET 4, Windows XP and Visual Studio 2010 to work correctly, beta versions will not work.

It has all the old features and updated new ones. The visualization is better with lines and images. There are more graphs to choose from and the overall feeling is easier. Now Visual Studio has a Telerik tab at the top for easy access to three explorers, as can be seen in image 3.2. There are also a Wizard to help generate a report fast and easy, see Appendix A.

![Image 4.1. Reporting tab in Visual Studio](image)

The toolbox is also updated with a Telerik tab and has 26 items that can be dragged and dropped to the report for easy use, see image 3.3.
The reports can be put in a WindowsFormApplication and launched to a webpage [3]. It is not possible to interact with the report on the webpage and it has a hard time adapting nicely to different screens. Integrating to a specific page could be hard, but this has not been tested during the analysis.

When the update to the new version was made, the reports stopped working. The previous code is old and some of it is removed and replaced in different ways in newer versions. The jump between versions became too big and it would take quite some time to fix all the reports, since it has to be done one at the time. The program can still export to pdf, csv, Excel, Rich Text Format, tiff and Webb Archive.

The Report Designer program is easy to use and can now be moved over to Visual Studio to receive code. The code can then be put in a WindowsFormApplication as previously stated. The program is easy to use and learn, and a report can be made in less than an hour without any previous knowledge. The report is then saved in trdx-format. In images 3.4 and 3.5
below, a template for a report is shown. The first one is made in Visual Studio by a programmer, the second one is a replica made in Report Designer.


 Telerik reporting costs $599 per developer and per year, with possibilities for volume discount, and $299 for an early renewal when the license is about to expire [4]. Telerik should be updated with every release, two to three times a year, to stay modern and up to date. The reports should be checked and evaluated at these times as well.

4.3  jsreport

jsreport is hosted both on your computer (Windows) and on the web with a simple login [5]. The installed program is easy to use and the documentation is somewhat lacking since it has not been updated to the latest version of the program. In both the installed version and on the web, all the code is written by a programmer in JavaScript and html. In image 3.6 below, a report example is shown in the program. In the left box, the JavaScript code is visible and in the right box the preview of the report is shown.

Image 4.6. A report example in jsreport

The program uses Node.js, that is a JavaScript runtime built on Chrome’s V8 JavaScript engine [6]. With JavaScript there is a lot of different libraries that can be added and get the visualization that is desirable, but nothing is built into the product. With the use of JavaScript and JSON the program can access the database that is needed and retrieve data. Since it is written in JavaScript, it can be integrated to the GATWeb.

The report can be exported to many different formats, such as xlsx, text, html and pdf. The program also has html-with-browser-client, fop-pdf, html-to-xlsx and phantom-pdf.

To create a new template for a report is simple. The menu is located to the left in the program and can be found in image 3.7 below, you press the plus sign and can start the coding. See Appendix B for a step by step to create a new report.
The site does not say how often the upgrades will come, so the maintenance is hard to predict. The template needs to be kept up to date and so does the queries that are used.

jsreport is a mix of a commercial product and an open source project and can be found on GitHub. There are also several additions to the program that can be found there [7].

The original jsreport is free for up to five templates. No license key is needed and there is access to a fully featured server. For $295 per year you have no limit on report templates, updates are included and development servers are for free. You can also go up to $995 per year and will then have the program royalty-free, infinite amount of copies, included updates and it is possible to distribute jsreport along with your single application [8].

4.4 Syncfusion

Syncfusion is a solution built on the open Report Definition Language(RDL) and the Report Platform was investigated during this study [9]. The Platform was very similar to Telerik’s Report Designer and worked in a similar way. It can be installed and works good in Windows. In image 3.8 below the starting page of a new report is shown and the creation of a report is built on a drag-and-drop idea.
The designer handles the most common visual graphics and can show data in lists, charts, gauge, maps, databars, sparklines, indicators, images and textboxes. These are easy to place and format to get the layout that is desired. When the report is done, it can easily be exported to Excel-, Word-, pdf-, or html-format.

The drag-and-drop approach makes it easy to create new reports, see Appendix C. To create templates that can be used for more than one report is not possible, so each report needs their own template to use. The reports are created and finished in the program itself and if they are hosted on the Syncfusion Report Server, they can be embedded in an application. If the report server is not used, this is not possible. The report server can be hosted in the local network or on the cloud, so the data stays in the company servers.

Using stored procedures is tricky and might not always work, depending on the database. The program wants the information in a specific way, which makes it hard to use sometime.

Updates need to be made to the templates in all the reports every time a new version comes out. An overlook of the reports should be done at the same time.

Report Platform costs $1995 per year and will then come with the Report Designer, Report Server (Windows), Report Server (Microsoft Azure) and Report Platform SDK [10].

4.5 Windward

Windward solutions are a Document Generation Platform that is working together with Microsoft Office Word, Excel and PowerPoint. It consists of
two parts; the Windward AutoTag and the Windward Engine [11]. The idea, that can be seen in picture 3.9 below, is that you create a template with the help of AutoTag in any of the three programs. An example of how to create a report can be seen in Appendix D.

The Engine is embedded in the application and will, with the help of the template name and data, get the template, add the data and then generate a report in the desired format.

The product is hosted in Windows and can use all the different charts and graphs that Word, Excel or PowerPoint are equipped with. The reports can be generated directly from the different programs if a database is connected and the output could be for example a PDF or html report.

The different programs can give a slightly different output. Word can generate reports in docx, html, pdf, rtf and directly to a printer. Excel has the outputs html, pdf, xlsx and a printer and PowerPoint can generate to pdf, pptx and a printer. When imbedded the engine is capable to “generate any output format you need” [12].

A report template was created to test the program and a fairly good result was achieved, see image 3.10. To a person that is used to work in these programs the templates should be easy to create. Windward is generous when giving example templates and code to help with the integration to the application [13].

A Windward license costs between $12 462-$23 292 depending on how much functionality that is wanted. A company can also contact them for license that is specifically made for their needs. The price is a onetime fee and after that a fee of 20% a year is charged for maintenance [14].

To fit the Company, the reports might have to change in appearance to be more adaptable in the chosen programs. In the image below, the A4 page is lying down to be able to fit the entire table.
4.6 Creating one by themselves

When making a report tool by themselves, one way is to use the same languages as is used to make the GATWeb, which is JavaScript, html and CSS. If the solution is created with these technologies, everything is compatible and will be easily integrated.

Here is an example of how it could be done. As can be seen in image 3.11, the user chooses the dates of which the report should cover. Then what kind of report is wanted and what template to use for the data.

Depending on what Data is chosen, a number of custom questions will appear to make sure all the information is there for the query, see image 3.12.
The request is sent to the database to get the information and to the templates. The information is then added to the chosen template and returned to the user as requested.

Play’n GO could create about 5-10 different templates that can be chosen by the user when creating a report. They should be dynamic and adapt to the amount of data that is taken from the database. The template should only say that “here is a chart” and “here is a table”. The data from the database will be handled and the chart and table will adapt from that. A simple example can be seen in image 3.13 below.

![Image 4.13. Example of a template image.](image)

The templates will get information from the request on what type of chart is wanted and will add the data from the database when it is available and return a full report. The “create report”-tab should stay at the top of the page so the user can easily change the values and generate a new report.

Tables can use AngularJS and the NgTable API to create the dynamic tables that is needed [15] [16]. The columns can be added dependent on how many titles the data has and the values will be added after that. Other examples can be seen in image 3.14 and 3.15. The first one has a more visual
design with an optional table if the “more”-link is clicked. It should be good if different charts from the same data is desired [17]. What data to put in the charts can be a good thing to ask in the custom questions that will pop up after the template is chosen.

![Image 4.13. A second report template with more charts.]

In image 3.15 there is no graphs at all, only tables. It would work in almost the same way as 3.14, with different tables showing different information. Important when creating the templates is that a good variety is shown. Adding pictures and/or logos would also be a good thing to add to a template.
This way of making a report will make all the reports look similar and create an overall same feeling, even when exported. The implementation should have export support for pdf, csv, excel, rich text format, tiff and web archive, like the old version. If a customer later wants a custom report, it is easy to create a new template or change the stored procedure for that user.

When making a report tool, it will demand more of the company when it comes to the maintenance. A responsible person needs to keep track of when new updates to the languages and the libraries are made. The reports need to be updated as well to be able to function when new code is implemented.

The cost for this alternative will be the mandays that is required to create it. Updates needs to be done regularly to make sure the code is up to date and an evaluation of the current templates should be done at least once a year to keep them updated and relevant.
5 Analysis

It is very easy to see that the old version of Telerik Reporting does not work for the needs of the company. With about 45 different reports, it is hard to maintain and the current way of creating a report is hard and a time-consuming activity.

The updated version covers many of the parts that the company wants, but the reports are still created in Visual Studio or Telerik Report Designer, one by one. To create a report is much easier in the updated version with the new Telerik tab, however it still takes some coding skills for it to be done exactly as they want it and it still lacks the capability to integrate nicely to the web.

jsreport is a nice program with a fast learning curve. The program does require good knowledge about JavaScript to be used, so it is not for beginners in that area. The product is the only one which is open source, but only up to 5 templates. This makes it a little difficult to use, since the limitation closes the alternative of ever making new templates after that.

Syncfusion Report Platform was easy to use and learn. It does however have a problem with the database when adding a Data Source. To do that correctly, changes has to be made in the database every time since it does not seem to be able to handle different conditions when retrieving data. So every stored procedure that has if-else statements will not work.

Windward is integrated into Microsoft Word, Excel and PowerPoint without problems and is ready to go, which is a big plus. If the developer is used to these programs, creating a report is very simple and demands very little from the user. The integration of the Windward engine does require some programming knowledge however, but it can be done with 14 lines of code. Templates are also used in Windward, but since the tables are not dynamic there might be a need of making a lot of different templates.

If they were to make their own framework, they would have to look up good ways of implementing it and maybe even have to hire a new developer to handle the programing, if not the skills are in the company already.

The programs that have been under analysis during this thesis have all been really good candidates for Play’n GO. The demands that the company had on the new program was good, but after working and talking to the people it was mentioned that some more features were desirable, such as responsiveness and drill-downs. The analysis has still been made with the written question in mind to keep a steady frame for the thesis, but a glance has been made to the responsiveness of the finished reports. Telerik is the only company that mentions responsiveness in their webpage, so an assumption is made that the others are not.

All the programs in the analysis met the original requirements. Syncfusion was however removed as an alternative do to the problems with connecting to the database and get correct information back. That leaves
Telerik Reporting, jsreport and Windward that all three meets the requirements.

The integration for the different programs are estimated between 1½-2 weeks and the work of making templates and making them work with GATWeb is estimated in about 3-4 weeks. Telerik Reporting has an estimation on about 10 mandays for the integration itself and another 20 mandays for making report templates and making them work online. jsreport is estimated to about 7 mandays for integration and 15 mandays for making the templates and getting them to work.

Windward seems to be the easiest to integrate with about 4 mandays. It is however estimated to 20 mandays to make all the templates and making them work. When making a framework by themselves there is not really any talk about integration, but the estimate to get a functional report. About 10 mandays is estimated to make the framework in the GATWeb and another 10 mandays to make the templates and be up and running.
6 Discussion

If I would start over with this analysis, I would probably have the same planning as before. The amount of time spent analyzing the different options was good. It would however have been good with a couple of more weeks. As can be seen in Future Research, there are a couple of more frameworks that would have been interesting to investigate.

Something that I would have done differently would be the way I choose the different frameworks. Every time I would start a new analysis I sat down and searched for a framework that I thought looked interesting. If I would do it again, I would have done all the searching and decided the frameworks in the beginning. That seems like a more well thought plan to work by.

I do not believe however, that the result would have been different if I had used that method. The analysis that was made of the programs was thorough. For every framework that was analyzed, the research questions have been answered and the questions were the foundation of the analysis when it was ongoing.
7 Conclusion

Even though three of the four analyzed frameworks meet the official requirements of Play’n GO, the conclusion is still that it would be best for them to make a framework of their own. This is based on the fact that none of the frameworks integrate to the web in a good and responsive way.

The use of AngularJS and the NgTable API makes it possible to make really good templates that could be used for many different reports, which again promotes making their own. When the framework is done, only a couple of templates needs to be made and maintained. If they would choose to use Telerik, for example, they would have to do all the 45 reports and keep all of them updated on a regular basis.

The mandays calculation concludes that by that estimation, making a product of their own would be the fastest option for the company. Since there are no license costs or anything for this alternative, the cost is also the smallest.

7.1 Future Research

To actually make a report in JavaScript would be one of the things that would be good to do. Another is to look into exactly how the programs were to be integrated, not only that it can be done.

Windward has many options that is not covered in this thesis, since it demanded integration to the system to be able to use them.

During the analysis in this thesis many frameworks has been overlooked when a choice was made. Some frameworks that could be interesting to investigate is Seal Report [18], myDBR [19], FusionCharts [20] and Stimulsoft Reports.js [21].
References


[Använd 23 November 2016].


Appendix A

A.1 How to create a report with Telerik Reporting

This guide will demonstrate how to make a report in the program Telerik Report Designer. In image A.1 you can see the opening page of the program. The wizard is an easy and good way to make a foundation of the report that you want.

![Image A.1. Start page of Telerik Report Designer.](image)

When clicking “next” on the wizard, it is time to add the data source. Previously added data sources will be visible in a list, or a new one can be created, as seen in image A.2 below.
When a data source is set, the type of the report is chosen. There are two types to choose from, a standard report or a label report. See image A.3.


Image A.3. Choose a type for the report.
The design of the layout is an easy drag-and-drop method that uses the values from the connected data source. The fields have a clear explanation in the boxes on the right side in image A.4. In the box on the left, the values from the data source will be shown when it is correctly connected.

![Image A.4. Design the data layout of a report with easy drag-and-drop.]

After the data layout, it is time for the report layout. There are three standard layouts that the program offers. Stepped, Outline and Left Aligned. All of them have explanations and a small preview, as can be seen in image A.5.
The last step is to choose a style for the report. The wizard will offer seven different style sheets. They come with a description and a preview, see image A.6.

Image A.5. Choose the report layout.

The last step is to choose a style for the report. The wizard will offer seven different style sheets. They come with a description and a preview, see image A.6.

Image A.6. Choose the style of the report.
The program will then develop a report based on all the data that has been entered and the work for a more custom report can begin. The image A.7 is not a good representation, since no data source or data has been added.

*Image A.7. Finished report after the wizard, no data is connected to it.*
Appendix B

B.1 How to create a report in JsReport

When JsReport is opened, a page with all the templates, data sources and scripts are shown. When you add more things, like images for example, they will show up here to. Image B.1 below shows the starting page of the program.

In the menu to the left, click on the “plus” next to “templates” to create a new template. A window will pop up where you name the template. This will create an empty page where you can write JavaScript code. In the same manner, data is added. Both the template and the data are empty pages that is up to the user to fill in. The name boxes for adding a template and data can be seen in images B.2 and B.3.
The template is now finished and the user can start coding the actual template. In image B.4 the picture will show a heading in code and in the prevue.
Image B.4. Overview of a new template with a heading.
Appendix C

C.1 How to create a report in Syncfusion

Syncfusion has an easy drag-and-drop function to creating the reports. In image C.1, the start page of the program can be seen. The tab “view” will decide what you see on the start page and not. The tab can be seen in image C.2.

 Begin by adding a data source, by either clicking on the plus or right click on the label. A new window will open where you add the properties, connect to the server and test the connection. See images C.3 and C.4 below.
After that you add the dataset in the same way and another window with properties will pop up.

When the program is connected to a data source, the buttons on the “insert”-tab is used to create the report that is wanted, see image C.5.
Image C.6 is an example how it could look when a template is finished. No data is connected to the template at this point.

Image C.6. Finished report example.
Appendix D

D.1 How to create a report with Windward

Windward reporting tool is integrated into Microsoft Word, Excel and PowerPoint and looks the same in all three programs. Two tags are added to the top bar, AutoTag and AutoTag Manager, and they are ready to start using. In the images D.1 and D.2 the two tabs with their functionality are shown.

![Image D.1. The AutoTag tab is open.](image1)

![Image D.2. AutoTagManager tab is open.](image2)

To start the report, open the program of choice and start a new project. Open the AutoTag Manager tab and click on Data Sources, to the far left. Image D.3 below will show the possible connections that Windward can do to different data sources.
When the connection to a database is made, create the report as wanted. To add charts, click on Tags under the AutoTag-tab and choose Chart. All the options under Tags are shown below in image D.4.


Image D.4. Tags options.
When a chart is chosen, a new window pops up to specify what data is wanted in the image. There are a lot of options for the charts, as can be seen in image D.5.

![Image D.5. Add data to a chart.](image)

If a table is wanted, the “ForEach” and “End ForEach” is used to tell the program where to start and where to stop the data. There is also if-else statements and switch-case statements. Image D.6 shows an example of how a report template could look when finished, no data is connected to the program.
The tables are created in the same way as a normal table in word, and then the “ForEach” tab is set in the beginning (first orange line), the “out” tag follows and are connected to what data should be there (the blue ones) and then the “End ForEach” is set at the end (last orange line).