sWOIS Reporting System Development

Thomas Damsten

Bachelor's Thesis

Degree Programme in Electrical and Automation Engineering

Vaasa 2021

BACHELOR'S THESIS

Author: Thomas Damsten
Degree Programme: Electrical and automation Engineering
Specialization: Automation
Supervisor(s): Matti Anttila – Wärtsilä Finland Oy
Joachim Böling – Novia University of Applied Sciences

Title: sWOIS reporting system development

Date: May 27, 2021	Number of pages: 51	Appendices: 1

Abstract

This Bachelor's thesis was done on behalf of the Plant Automation department of Wärtsilä Finland Oy. The goal of this thesis was to create instructions and training material for the use of the reporting solution Dream Report for industrial and process automation, in conjunction with Wärtsilä's server based Wärtsilä Operator Interface System (sWOIS) and AVEVA Historian, from now on Historian, in a virtualized environment.

First, introductory courses for Historian and Dream Report were attended to get a basic understanding of the solutions in use for sWOIS related to this project. To further extend the knowledge of the reporting aspects, manuals for sWOIS reporting with Historian and Dream Report were reviewed. As the foundation of the different software applications were set and familiar, individual use of Dream Report was conducted to start creating report examples. As the workflow progressed, a firm grip of the reporting solution and its possibilities with Historian and sWOIS was established.

This thesis covers the sWOIS structure, data management using Historian, the reporting solution Dream Report on its functionality, studio modules and use. The process of creating a report using Dream Report's simulation driver for real-time data is documented and can be used as training material in Wärtsilä's courses as an introduction to reporting with Dream Report.

As a result of the thesis, a suggestion of an engine maintenance report created on a virtual machine using Historian as a data source is presented.

Language: English	Key words: reporting, sWOIS, Historian, Dream Report

EXAMENSARBETE

Författare: Thomas Damsten
Utbildning och ort: El-& automationsteknik, Vasa
Inriktningsalternativ/Fördjupning: Automation
Handledare: Matti Anttila – Wärtsilä Finland Oy
Joachim Böling – Yrkeshögskolan Novia

Titel: Utveckling av sWOIS rapporteringssystem

Datum: 27.5.2021 Sidantal: 51 Bilagor: 1

Abstrakt

Detta examensarbete blev gjort på uppdrag av Plant Automation, Wärtsilä Finland Oy. Målet med detta examensarbete var att skapa instruktioner och utbildningsmaterial för användningen av rapporteringslösningen Dream Report för industriell och processautomation. Detta gjordes i samband med Wärtsiläs server baserat Wärtsilä Operator Interface System (sWOIS) och AVEVA Historian, i fortsättningen Historian, i en virtualiserad miljö.

Arbetet påbörjades med att delta i introduktionskurser för Historian och Dream Report för att få en grundlig förståelse om lösningarna i användning av sWOIS som relaterar till detta examensarbete. För att ytterligare utöka kunskapen om rapporteringsaspekterna, granskades manualer för sWOIS-rapportering med Historian och Dream Report. När grunden till programvaruapplikationerna var satt och bekant, påbörjades den praktiska användningen av Dream Report för att skapa exempel av rapporter. I samband med att arbetet framskred, erhölls en uppfattning om rapporteringslösningen och dess möjligheter med Historian och sWOIS.

Detta examensarbete behandlar strukturen av sWOIS, datahantering med Historian, rapporteringslösningen Dream Report om dess funktionalitet, programmoduler och användning. Arbetsprocessen för att skapa en rapport med Dream Reports simuleringsdrivrutin för data i realtid är dokumenterad och kan användas som utbildningsmaterial i Wärtsiläs kurser som en introduktion till Dream Report.

Resultatet av examensarbetet presenterar ett förslag av en underhållsrapport av en motor. Rapporten blev skapad på en virtuell dator där Historian användes som datakälla.

Språk: engelska Nyckelord: rapportering, sWOIS, Historian, Dream Report

---

OPINNÄYTETYÖ

Tekijä: Thomas Damsten
Koulutus ja paikkakunta: Sähkö- ja automaatiotekniikka
Suuntautumisvaihtoehto: Automaatiotekniikka
Ohjaaja(t): Matti Anttila – Wärtsilä Finland Oy
Joachim Böling – Yrkeshögskolan Novia
Nimike: sWOIS raportointijärjestelmän kehittäminen

Päivämäärä: 27.5.2021 Sivumäärä: 51 Liitteet: 1

Tiivistelmä

Tämä opinnäytetyö tehtiin Wärtsilä Finland Oy:n Plant Automation -osastolle. Opinnäytetyön tavoitteena oli luoda ohjeita ja koulutusmateriaalia teollisuus- ja prosessiautomaatioon tarkoitetun Dream Report -raportointiratkaisun käyttöön. Tämä tehtiin yhteistyössä Wärtsilän palvelimen Wärtsilä Operator Interface Systemin (sWOIS) ja AVEVA Historianin, jatkossa Historian, kanssa virtualisoidussa ympäristössä.

Työ aloitettiin osallistumalla Historianin ja Dream Reportin perehdyttämiskursseille, jotta saatiin perusteellinen käsitys ratkaisuista käyttäessä sWOIS-järjestelmää, joka liittyy tähän opinnäytetyöhön. Tarkastamalla käsikirjat sWOIS-raportoinnista Historianin ja Dream Reportin avulla saatiin lisää tietoa raportointinäkökulmista. Kun ohjelmistosovellusten pohja oli tehty ja siihen oli tutustuttu, aloitettiin Dream Reportin käyttöä käytännössä luomalla raporttiesimerkkejä. Työn etenemisen myötä saatiin käsitys raportointiratkaisusta ja sen mahdollisuuksista Historianin ja sWOIS:in kanssa.

Tämä opinnäytetyö käsittelee sWOIS:in rakennetta, tiedonhallintaa Historianin kanssa, Dream Reportin raportointiratkaisun toiminnallisuutta, ohjelmamoduuleja ja niiden käyttöä. Työprosessiraportin luominen Dream Reportin reaaliaikaisen tiedon simulointohjelman avulla on dokumentoitu ja sitä voidaan käyttää koulutusmateriaalina Wärtsilän kursseilla johdatuksena Dream Reportiin.

Opinnäytetyön lopputulos esittää ehdotuksen moottorin huoltoraportista. Raportti luotiin virtuaalisella tietokoneella, jossa Historiania käytettiin tietolähteenä.

Kieli: englanti Avainsanat: raportointi, sWOIS, Historian, Dream Report

---

Table of Contents

List of Figures and Tables

Abbreviations

sWOIS = Server based Wärtsilä Operator’s Interface System

HMI = Human-Machine Interface

PLC = Programmable Logic Controller

SCADA = Supervisory Control and Data Acquisition

RTU = Remote Terminal Unit

ICS = Industrial Control System

LAN = Local Area Network

WAN = Wide Area Network

UPS = Uninterrupted Power Source

NTP = Network Time Protocol

VM = Virtual Machine

ITAA = InTouch Access Anywhere

OPC = Open Platform Communications

OPC HDA = OPC History Data Access

ODBC = Open Database Connectivity

OLE-DB = Object Linking and Embedding Database

AST = Automatic Statistical Table

RMC = Runtime Management Console

VM = Virtual Machine

1 Introduction

This thesis was done on behalf of the Plant Automation department of Wärtsilä Finland Oy, with the purpose of creating documentation and training material with instructions for using the reporting program called Dream Report. Dream Report is an automated reporting and data analysis software designed for industrial automation applications.

Wärtsilä's own server-based Operator Interface System uses the Dream Report software solution and industrial reports can be created with the program. The demand for additional reporting material is motivated by the internal and external demand from Wärtsilä's customers. The purpose of this thesis is also a study of possibilities regarding potential report suggestions that can be implemented in the Energy Business sector. As Wärtsilä Energy Business is the targeted audience for the report suggestions, the most relevant data in that field is processed in the reports.

From a broader perspective, this thesis is based on Wärtsilä's own industrial control system and its applications. These applications include a process historian for data storage, a company-created SCADA and HMI system for control and monitoring purposes, and finally a software used for creating reports based on the data gathered from the processes controlled by the system.

The goal of this thesis is to create a suggestion for a new type of report, for a Wärtsilä power plant asset management by using Historian as a database and presenting gathered data with Dream Report in Wärtsilä's own server-based virtualized environment called sWOIS. This documentation will later be used as training material for conducting reporting simulations with Dream Report.

The importance of reporting processes in facilities and plants cannot be understated, as it is essential that companies can provide valid reports related to production and asset management. High-quality reports contribute to better decision making among operators and managers, as accurate decisions related to device and system lifecycles and production leads to optimized operations within an organization.

1.1 Organization

Wärtsilä is a global leader in smart technologies and complete lifecycle solutions for the marine and energy markets. With major emphasis on sustainable innovation, data analytics and total efficiency, the performance of both the economic and environmental aspects of vessels and power plants, are maximized for the customers. The Wärtsilä brand identity reflects the corporate values of the company, which are Energy, Excellence and Excitement. Wärtsilä has approximately 18,000 employees and the company har operations in over 200 locations in more than 70 countries around the world. [1]

1.2 Energy business

Wärtsilä Energy Business is leading the transition towards a 100% renewable energy future. To accelerate the transition towards renewable energy, Wärtsilä contributes with their leading market understanding, capability to create technology solutions and lifecycle services. With capable energy system integrations, the customers' assets are being made even smarter and more competitive on their path towards a renewable energy future. [2] Some of Wärtsilä's offerings include engine and hybrid power plants, energy storage optimization and flexible lifecycle solutions. [1]

1.3 Plant Automation

Plant automation is the team that I was a part of when I was working on my thesis. This team operates by maintaining technical knowledge of their products and services, distributing software development tools and backups, along with conducting field tests, surveys and trainings. This technical knowledge and support are developed for plant automation products and is shared both internally and among Wärtsilä's customers. Other than knowledge maintenance and development, the team also aims to improve the ways of distributing this knowledge and provide sales support of their products. [3]

1.4 Project Background

The need for this thesis became apparent when the plant automation team decided that further training material is needed to learn how to utilize Dream Report more efficiently, discover more possibilities of the solution and to provide Wärtsilä's customers additional viable options of high-quality generated reports.

The need for high-quality reports is essential for any business in the industrial sector and an easy-to-use software makes the process even more appealing. Wärtsilä have already implemented the use of Dream Report in conjunction with their own virtualized server environment and my task is to provide further training material. To summarize, the purpose of this thesis is to create documentation material and instructions for the use of the reporting solution Dream Report, in conjunction with the server based Wärtsilä Operator's Interface System (sWOIS) and using Historian as a data source.

1.5 Disposition

This thesis is divided into different chapters. Each chapter will explain the different areas of theory and practice that are relevant to this work. At the stage where the reader has reached the practical section of the thesis, they should have a good understanding of the different aspects that are involved in the final reporting process.

The first chapter is introducing the reader to the projects background and presenting Wärtsilä as an organization, along with the operation department and team that I worked for.

The second chapter gives a basic overview of an industrial control system and the different instrumentations and applications that are usually a part of a control system. The third chapter explains the Wärtsilä Operator Interface System (WOIS), what hardware and software that the system is built of and why it is used in the organization.

The fourth and fifth chapters explains the theory of the Historian and the reporting solution Dream Report. Here the reader will find information about the different aspects of data retrieval and aggregation that are used for generating industrial reports, as well as the different features that are available with the reporting software. When creating reports for process and industrial automation, it is important to know what types of data that can be used and why it is beneficial to use a process historian to help manage that data.

The sixth chapter covers the result of the project and in the seventh chapter, the reader will find a concluding discussion about the result of the thesis, the problems that were encountered and what was successful regarding the work. The final chapter contains the references used in the thesis.

2 Industrial Control Systems

This chapter will introduce the reader to the concept of an industrial control system (ICS) and the several types of systems and instrumentations that are associated with it for industrial process control. In the case of devices in a Wärtsilä plant or facility, the system is controlled and monitored with their own application of a SCADA and HMI system called sWOIS. This interface system will be further explained in Chapter 3 of this thesis. Before explaining sWOIS, it is relevant to explain the basic architecture and principle of an ICS to better help the reader in the transition from a general ICS to a version specifically developed for a company.

2.1 SCADA

Supervisory control and data acquisition (SCADA) systems incorporates PLCs, HMIs and network communication systems into a complete integrated system that allows an operator to monitor, control and gather measurement information of a plant or facility assets. Essentially, SCADA is a collection of hardware and software components. SCADA systems are widely used in modern-day utility industries for purposes such as monitoring and local or remote controlling of processes. Typical utility industry processes are electric power transmission and distribution, water and gas distribution. Other than utility industries, SCADA systems also support manufacturing and industrial processes. [4, p. 237]

The evolution of SCADA architecture has developed through new needs of the industry and as early automation systems and technologies were replaced, newer computers, microprocessors, standards and communication technologies were discovered to aid in developing a modern supervisory system. As the system developed, supervisory personnel of a plant could start monitoring and control the processes remotely instead of having to monitor the production on site. This became hugely beneficial in terms of flexibility and cost reduction of routine visits to monitor operations in facilities or plants. Fewer visits to the plant site were more common because the control and monitoring operations could be done remotely. [4, p. 237], [5, pp. 1-2]

The structural design of a standard SCADA system is a combination of subsystems. These subsystems include, the central host computer, a field-located remote measurement and control equipment called remote terminal units (RTU) and/or PLCs, a communication system to connect all the equipment and an operator interface that provides the user with access to the system. These are the critical subsystems of the architecture that have not changed much compared to the technologies and operational use of the SCADA system that has been evolved with improved technology and topology. [4, p. 237]

Core functionalities of a SCADA system are acquisition of data from field devices through PLC and/or RTU and perform remote control of field devices. Field data can be processed to detect changes in a process, providing a consistent database of process information about the plant or facility. Process data can be generated through a graphical user interface to display alarms, trends and reports, performing system monitoring and taking appropriate actions to abnormalities. The field devices capture data from the field and send it to the central host computer in response to message requests. The message contains information from the controlled device, such as configuration data, process measurements, control commands, and so on. The communication system enables communication between the central host computer and the field devices so that messages can be sent and received. [4, p. 238]

The modern SCADA system supports a whole lot more features and functions than the ones discussed in this chapter, but the ones covered here are the most basic set of functions that represents the foundation of every SCADA system architectural design. In addition to the functions and features already mentioned, SCADA systems also support industry-specific applications with customized interfaces for the unique requirement of the individual market division. [4, p. 247]

To better understand the process and importance of generating industrial reports, it is essential to be aware of how the data gets from point-a to point-b and how it is ultimately presented in an understandable format from which the data can be interpreted, and necessary actions can be taken accordingly. As the SCADA system is a central part of this process, it is appropriate that the reader gets a basic view of this system and the role it plays in the world of automation technology.

2.2 PLC

The programmable logic controller (PLC) is considered as the foundational block of which the automation systems are built of. The PLC is a special purpose computer that uses a programmable memory to store instructions and perform functions to control machines and processes. The many different functions that a PLC can implement are, as an example, logic, timing, sequencing and counting. Primarily the programming is done with the goal of performing logic and switching operations, as an example, if A and B occurs switch on C and if A or B occurs switch on D. Input and output devices in a system are connected to the PLC and instructions can be entered into the memory of the PLC by the operator and the signals are then monitored and controlled according to the programmed rules. The programming and the rules established is easily configurable by altering the instructions. This results in a flexible and cost-effective system that does not require complex rewiring of the control system, compared to the old relays that were replaced by PLCs. [4, p. 37]

The PLC design is optimized for control tasks and the industrial environment. Thus, the PLCs are rugged and designed to withstand tougher conditions compared to a regular computer, such as, temperature, vibrations and humidity. [4, p. 38] The functional components of a PLC are a processor unit and memory, power-supply unit, programming device, input/output (I/O) and communications interface. The power supply is used to convert the mains AC voltage into DC output voltage that powers all the other modules associated with the PLC. Important note is that the power supply does not provide power for field devices, only the PLC modules. The I/O modules are connected to the field devices such as, switches, valves and lamps. The processor consists of the central processing unit (CPU) and memory.

This part makes decisions needed to observe and operate the field devices connected to the I/O modules. The memory stores data representing the field devices conditions and what actions to perform based on the made observations. [4, p. 38-41] Finally, the programming device is usually a laptop or a desktop computer that hosts the decision-making programs that are created for the PLC. An example of a programming PLC software is Siemens TIA portal.

The programming language of the PLC is based on the international standard for programmable controller programming languages, called IEC 61131-3 (formerly 1131-3), established by the International Electrotechnical Commission (IEC). This standard consists of two textual languages, Instruction List (IL) and Structured Text (ST), along with two graphical languages, Ladder Diagram (LD) and Function Block Diagram (FBD). Companies can choose to implement this standard to improve the creation and maintainability of a control project. [4, p. 52], [7], [8]

2.3 Communication Network

The communication network is an important part of the power plant. Many tasks are done through the network, such as production and process data collection, monitoring and controlling of devices and device configurations. The basic principle behind a communication network used in power plants is to provide plant-wide information accessible to the user, exchange of information through interactive workstations, integrity and security of data for reliable plant operations. The structured network can be split into logical segments to maintain better configuring and commissioning, also, if the facility experiences a security breach, the threat can be isolated into one small section of the control network and thus protecting other segments. [9] This form of security maintenance can be achieved with the use of routers and firewalls. There are different types of network topologies that facilitate data exchange among units and stations in a plant, some of the most common topologies are star, ring, mesh, and bus network. Usually, the data communication networks are functionally divided into a Local Area Network (LAN) or a Wide Area Network (WAN). The networks are separated by their different geographical applications, where a LAN is usually confined to a single room or building, whilst a WAN extends a long distance in an area and connects to different LANs used in cities. [6, pp. 238-240]

3 WOIS Overview

This chapter will look at the general information about Wärtsilä Operator's Interface System (WOIS), why the system is used, as well as explaining what components the system is built upon.

3.1 sWOIS in General

The term "sWOIS" is an internal name and stands for server WOIS, which separates it from the traditional WOIS. The system is used for monitoring essential data and the status of a power plant, manufacturing facility or marine vessel, all while simplifying the work of the operator through simple visualization of process information. The facility or power plant in use is controlled and supervised from sWOIS, and all the key data and actions necessary for optimal operations are accessible and executed via sWOIS operator stations, using a mouse, keyboard and display. The operator station is based on thin client hardware and is equipped with the graphical operator interface specifically customized for Wärtsilä power plant solutions to provide a reliable and efficient operator interface. The sWOIS system includes an integrated historian for historical data storage and archiving. For reporting purposes, the sWOIS system is also equipped with a reporting solution for generation of production reports. This solution cooperates with Dream Report, which is the reporting solution mainly covered in this thesis. [10]

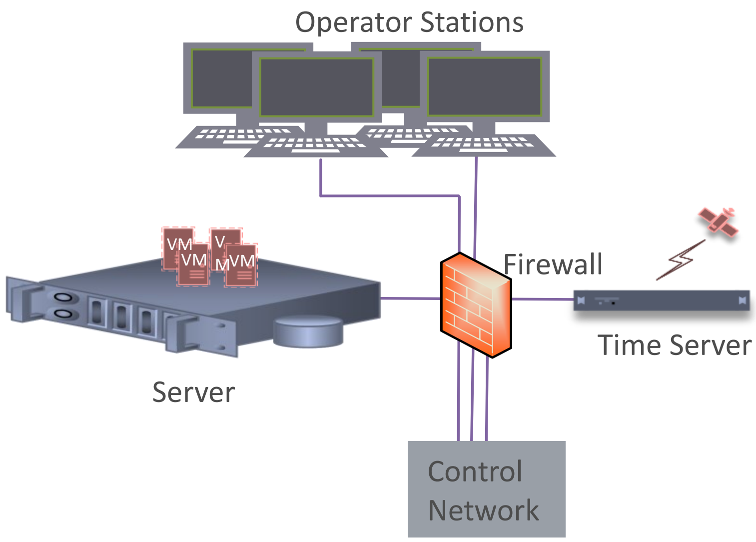


Figure 1: Basic sWOIS Setup. [12]

3.2 sWOIS Architecture

This section will explain the key components of which the sWOIS is built upon. The sWOIS system is server based, running on server hardware in a virtualized environment. The hardware connects to the WOIS server(s) via the plant network. [10]

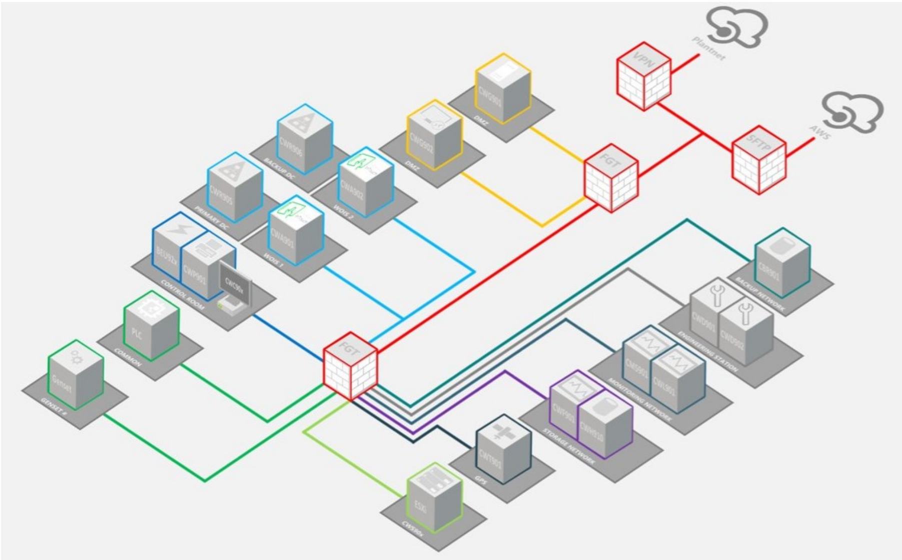


Figure 2: sWOIS System Logical Overview. [13]

Figure 2 illustrates a logical system overview of the sWOIS architecture using color connections to separate the different parts of the system. The system is connected to the cloud plant network though VPN and Fortigate firewalls (FGT), and data can also be extracted through a File Transfer Protocol (SFTP) to a cloud service platform (AWS). [13] I created Table 1 to summarize the different parts in Figure 2.

Table 1: Summarized table of parts and their functions from Figure 2

Color	Part	Function
Red	Firewall	Protects the system from potential security threats
Yellow	Demilitarized Zone (DMZ)	"isolates" the system from the internet
Blue	Operator station and domain controller	Virtual machine (VM) for Domain Controller
Dark Blue	Operator's terminal, UPS and printer	Serves as the control room
Green	Genset and PLC modules	Modules that connect to plant/facility devices
Light Green	Virtualization platform ESXi	Hosting and coordinating multiple operating systems
Black	Time server (GPS)	Synchronizes the time on network clients through NTP
Purple	Storage network	VM for reporting (Dream Report) and archiving (Historian)
Dark Blue	Monitoring network	Monitors health status of sWOIS server hardware
Gray	Engineering station	Maintenance and management system
Turquoise	Backup network	Creates, validates and stores backups on dedicated server

sWOIS Hardware:

This section covers the different hardware that is used in sWOIS. Table 2 shows the hardware components and their functions.

Table 2: sWOIS hardware and their functions. [10]

Hardware	Functions
Server rack	cabinet containing most of WOIS components.
UPS	Provide emergency power to a load when the main power source fails.
NTP Server	Clock synchronization for servers via the network.
Server	Services for clients, sharing data or resources.
Firewall	Monitors and controls incoming and outgoing network traffic.
Thin client with monitor, keyboard and mouse	Components serve as the operator station. Connects to the server and presents the user interface to the operator

sWOIS Software:

This section covers the different software that are present in the sWOIS system. Table 3 shows the software components and their functions.

Table 3: sWOIS software and their functions. [10]

Software	Functions
VMWare ESXi	Virtualization platform
Windows server 2016	Operating system, user management
Windows AppLocker	Malware prevention
AVEVA InTouch	HMI, WOISApplication
AVEVA Historian Clients	Data storage, archiving, trend component and Query tool
AVEVA Application Server	Components serve as the operator station. Connects to the server and presents the user interface to the operator
Alarm Client	Viewer for Alarms and Events in long term storage
Ocean Data Dream Report	Reporting
AVEVA InTouch Access Anywhere (ITAA)	Remote monitoring/operation option with Mobile HMI
AVEVA ITAA Secure Gateway	Connection point option in DMZ for ITAA

3.3 sWOIS Purpose

To summarize the system, sWOIS is made to simplify the work of the operator by providing access to essential information and status of a Wärtsilä power plant. This is made possible through a high-quality graphical user interface designed for monitoring purposes. [10] In today's industrial society it is essential that the process data and status of a power plant is easily accessible and visualized. Not only is it important that the data is visualized, but also that the interface system provides the accurate methods and actions available for controlling the monitored facility through an interface that is easy to use for the operator.

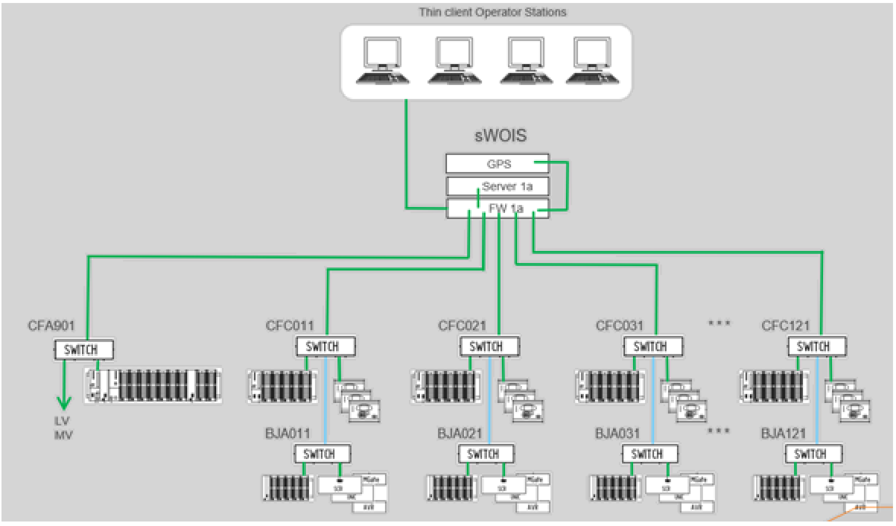


Figure 3: Plant automation layout with sWOIS. [12]

4 Historian

Historian is a high-performance process historian, capable of handling the large amount of data that today's industrial facilities and businesses generate. Historian can deliver the generated data with speed and efficiency for storage and retrieval purposes. It utilizes the flexibility of Microsoft SQL server and leverages some of the server features, such as database security, replication and backups. The Historian uses advanced storage and compression techniques to ensure open access to process, alarm and event data needed for analysis and reporting. [14] Process data is considered relevant data that answers business questions, these data types are real-time, historical, summary, event and configuration data. This enables fully informative decisions from the operators, as the data provides the operational performance of the facilities. Data analysis from process data can help improve a business from a performance, quality or cost perspective. [4, pp. 583-584]

4.1 SQL

SQL is a flexible computer sublanguage that is the most implemented tool for communicating with a relational database. The language works with the user typing and sending a request to the database and the database management system (DBMS) handles the request to achieve the most optimal outcome. The communication between the user and the database happens through an interaction called a *query*. A query is a question that that the user asks the database, and if the database has valid data in response to the query, SQL retrieves that data. Besides retrieving data, SQL queries can also create, delete, insert and update database tables and records. SQL has been accepted as a standard query language for relational databases by ANSI (American National Standards Institute) and ISO (International Standards Organization). [15, pp. 21-24]

SQL Server is a relational database management system (RDBMS) developed by Microsoft. The system allows the user to manage databases through a management studio, with SQL being the sublanguage that operates on the system. Since SQL is a sublanguage, it must be combined with a procedural language to be used in an application. A procedural language is such programming languages as C, C++, Java and Python. Because of the way that SQL is structured, it has some strengths and weaknesses, and these are complementing the strengths and weaknesses of a procedural language. This means that if the two languages are combined, powerful applications can be built with a large range of capabilities. A good strength in SQL are the many tools of data retrieval that are available with the sublanguage. Whether a table in a database has a hundred or a thousand rows, SQL can simply retrieve data in a table without having to deal with each row or column individually. Comparing this strength with a typical weakness in a procedural language, such as the design for operating in one row at a time, it is noticeable that the two types of languages complement each other. [15, pp. 333-338]

As mentioned earlier in the above section, the two languages can be combined. The most common method of combining SQL with a procedural language is called *embedded* SQL. [15, p. 336-337] As Chapter 4 introduces the reader to Historian, it is important to note that AVEVA has combined data collection with time series extensions to an embedded MS (Microsoft) SQL Server to optimize both the storage and retrieval performance of Historian. [14]

4.2 Historian Architecture

The Historian architecture is based on a client/server system. This computing model distinguishes different client applications from server applications where a client can communicate with the server. In these types of systems, the number of clients can change in time, but the number of servers is finite. [15, pp. 47-49] The configuration data of the Historian is stored on a MS SQL Server database, tables and real-time data is stored in proprietary files of the Historian called history blocks. The database communicates to the server as a response to requests from the client. [16] The chapters 4.2.1 to 4.5.2 will cover more information about the Historian client and server, along with the different data storage and retrieval modes that are associated with the Historian.

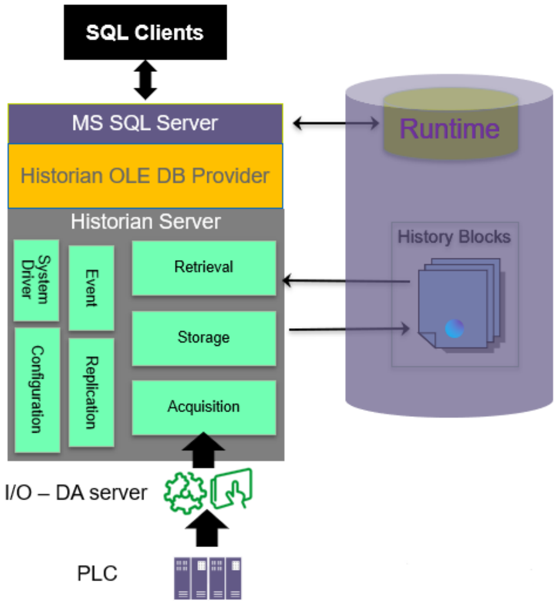


Figure 4: Historian architecture. [16]

4.2.1 Historian Server

Historian Server is a software component in the client/server system that interprets SQL requests and commands coming from the client and translates these requests into actions in the database. The server software also sends back retrieved commands from the client, with the result being formatted. To summarize the job of the server, it acquires real-time plant data, compresses it, stores and reports it in response to SQL requests that is sent to it across a network of clients. [16]

4.2.2 Historian Client

Historian Client software operates with the intention to maximize the value of data stored in the Historian Server. The applications presented in the Historian Client allows the user to explore and trend process data with a graphical interface, analyze the data to produce relevant information and through a point-and-click query tool execute queries against any Historian Server data. Historian Client also offers the function to create reports, however this function is not used for this thesis. [16]

4.3 System Tags

Any data related to running a process successfully, is stored and acquired by the Historian as tags. The term *tag* refers to any hardware input or output signal and any internal software point. [5, pp. 51-52] Each tag identifies the corresponding device or software point to the Historian and represents a parameter or a plant data point. As an example, a pump might have two tags – one for the volume meter and one for the speed. This tag-system allows the operator to easily distinguish between different devices and parts of a plant and what their values are, as they are recorded by the system tags. The data corresponding to each tag is stored by the Historian Storage subsystem, and then retrieved by the Data Retrieval subsystem. [17, p. 7] The storage and retrieval subsystems are explained further in Chapters 4.4 and 4.5.

4.4 Data Storage

Historian uses many different structures to store data. The main structure that is used is the SQL Server database. This database is created by default when Historian is installed and is called Runtime. Configuration information and classic event data is stored in this database. The configured information is relatively static and is not often altered, such as added or deleted tags, descriptions and engineering units. The Runtime database is transferred with a set of standard database entities, such as tables, views, stored configuration data and procedures for a typical industrial facility. Data that reflects a factory environment can be configured in the System Management Console for the

Historian, and the configured data can later be added to the Runtime database. Processing data, including alarm and event data is stored in proprietary files called history blocks. The history block is a folder containing data files and if configured, subfolders. There can be gaps for a certain time period in the history blocks, called block gaps. This occurs if no data is acquired, or if a block is deleted. The communication is handled with OLE-DB to access the time-based data in the history blocks, which is stored separately from the Runtime database. The Runtime database can be managed in SQL Server Management Studio.

Historian supports four storage modes called delta, cyclic, forced and no storage. These modes tell the Historian to store data only if certain conditions are met. I created Table 4, which will show the different modes and their conditions. [18, pp. 147-148], [4, p. 588]

Table 4: Historian storage modes.

Storage Mode	Condition
Delta storage	Only changed data values are stored
Cyclic storage	Only data values specified at a certain time interval is stored
Forced storage	All collected data values are stored
No storage	No data values are stored

4.5 Data Retrieval

There are different retrieval modes that allows the user to retrieve data in different ways. This is important based on the purpose that the data is used for. As an example, if data is retrieved over a long time period, it is optimal to only retrieve a few hundred evenly spaced points of data to minimize the response time. The same principle goes for a shorter time period, it is more optimal to retrieve all values to get more accurate results. [19, p. 27]

Like the Data Storage subsystem, Historian also features a Data Retrieval subsystem. It is through the retrieval subsystem that the Historian receives SQL queries from clients, locates the requested data and processing that data to be returned as a result. There are two repositories that the Historian can store data in, SQL Server database and history blocks. There are different methods of data retrieval for both repositories, and it is important to know the difference between them.

The retrieval of configuration and event data stored in SQL Server database tables is done with SQL queries, as this is the data sublanguage that the server responds to. The historical data from history blocks must first be retrieved and sent to clients as if it is stored in SQL Server tables. When the data from the history blocks is retrieved, it is presented to the OLE-DB provider as “virtual” history tables, which can then be read and interpreted by the client that requested the data. The history blocks serve as a remote data source for SQL Server.

Like the Data Storage modes, there are different retrieval modes available as well for Historian. In chapter 4.5.1, the cyclic retrieval mode is explained, as it is used in conjunction with sWOIS reporting with Dream Report. [19, pp. 11-12], [4, p. 588]

4.5.1 Cyclic Retrieval

The cyclic retrieval mode retrieves stored data for a given time period based on a specified cyclic retrieval resolution, regardless of changes in the tag(s) value. In this mode, a virtual row set is produced, and one row is returned for each “cycle boundary”, which may or may not correspond to the actual data rows stored on the Historian. The user can specify either several cycles or a time resolution, for which the cyclic retrieval mode operates in. The cyclic resolution is calculated by dividing the specified time period by the number of cycle boundaries. If a user specifies a resolution, the number of cycles is calculated by dividing the time period by the resolution.

Cyclic retrieval works by retrieving data from a table in cycle boundaries over a defined time period. If no data value is stored specifically at a cycle boundary, the mode returns the last value before the boundary. The cyclic retrieval mode is used because it is a fast way to retrieve data and consumes little server resources. Figure 5 illustrates how values are returned for cyclic retrieval. [19, pp. 27-28]

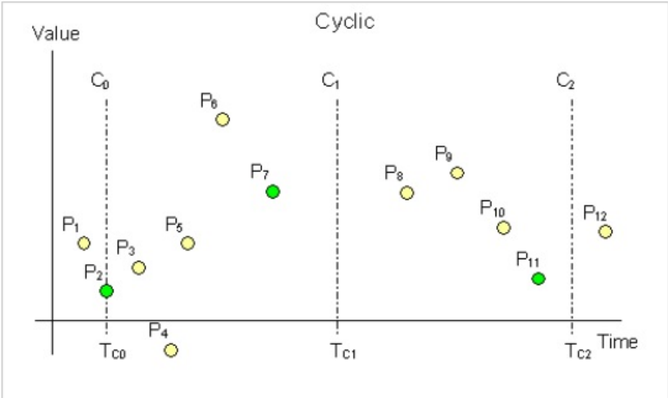


Figure 5: Cyclic retrieval function. [19, p.28]

Data is retrieved using three cycle boundaries over a defined time period. The retrieval mode returns the data points at these cycles based on the resolution that was set. The cycle boundaries are defined as T_{C0}, T_{C1} and T_{C2}. Each dot in the image represents an actual data point stored in the historian, but only the green dots (P_2, P_7 and P_{11}) are returned values.

4.5.2 Delta Retrieval

Another type of retrieval mode that is available with Historian, is the delta retrieval mode. This mode retrieves data only based on exceptions, which are only the changed values for tag(s) for a specified time period. Duplicate values are not returned in this mode. Compared to the cyclic retrieval mode, the delta retrieval mode does not create a virtual row set that may or may not correspond to the actual row set in the historian. Instead, this mode produces a row set comprised of rows that are indeed stored in the historian. The delta query returns all physical rows for the specified tag(s) over the specified time period and any duplicate values are not returned.

Delta retrieval mode works by returning data points of changed values inside a specified time period. If there is no actual point of data at the start time of the specified time period, the last data point before the time period is returned instead. Figure 6 illustrates how values are returned for delta retrieval. [19, p. 30]

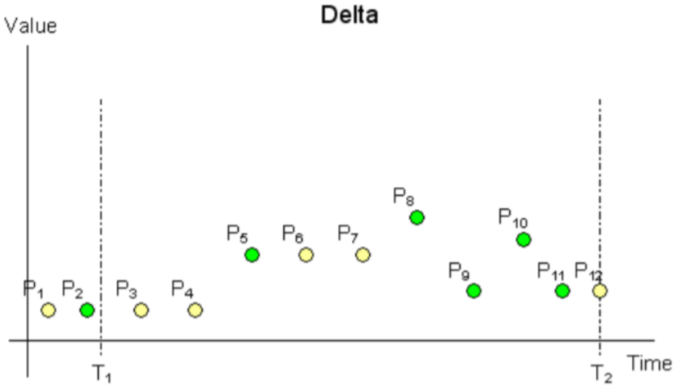


Figure 6: Delta retrieval function. [19, p. 30]

Like the cyclic retrieval illustration in Figure 5, the delta retrieval illustration also returns only the values of the green dots in Figure 6. These returned data points are changed values, which is why there are no green dots on the same horizontal level right after each other.

4.6 Historian Purpose

The use of database systems such as Historian is essential for many automation topologies. There are many benefits with using a historian database when it comes to maintaining and improving the quality of manufactured products for any organization. Some of the major benefits of a historian is, the huge data storage capacity, which allows for appropriate data management from multiple sites/plants as the ability to store huge data is a key factor to improving and maintaining product quality. Options for data capturing improves the potential of reducing size of data captured over a certain time period, as a result, data capture can be adjusted according to the physical memory available and thus reducing the possibility of data being lost or corrupted. The ability to store data for a longer time period is important for decision making, as it enables users to find patterns of the system behavior and a performance analysis of the system can be done. Other options available with Historian is statistical tools for analyzing parameters, such as trending and analysis tools. To summarize, a historian is mainly used for better data management and analysis rather than controlling any process in a plant or facility. [4, pp. 583-584]

Regarding why Wärtsilä chose AVEVA as the vendor of choice, it was the result of an intensive evaluation process between different product vendors. According to Development Manager Joakim Kullas (e-mail 14.4.2021) the choice of using Historian for data storage was made based on the AVEVA vendor's performance in the evaluation process, where many aspects were considered, such as lifecycle and maintenance aspects, cost analysis and other possible complementary solutions for their products. To summarize, Historian was chosen because it was evaluated to be the best performing native option for long term data storage.

5 Dream Report

Dream report is a professional automated reporting and data analysis software designed specifically for industrial and process automation. The software is created by Ocean Data Systems. It supports real-time data and alarm acquisition, optional logging to a database, as well as direct connectivity to third-party archives, such as process historians. Dream report allows the user to create reports for projects in a very user-friendly environment without programming or scripting, as such, no software development skills are required to design reports in the dream report studio. The user can design and configure reports using templates, graphical objects and functions to calculate and present data in different reporting styles. The reports can be triggered automatically on a scheduled or event-based generation, and they can be published in many formats and distributed to various destinations. Dream report also has the web-portal feature, which allows automatically generated reports, as well as dynamically generated reports by a user to be viewed in the web-portal. The web-portal is accessible on both desktop and mobile devices.

Dream Report offers access from any system interface, whether open or proprietary protocols are used by the supplier. The solution integrates a data logger module to log data to any standard database when data is not already archived. Data can be extracted using graphical objects in a studio environment, and the report can be designed and customized in an intuitive graphical editor. Finally, the reports can be generated and distributed manually or automatically and in various formats. The solution is a good fit for many markets and especially the field of industrial and process automation. Dream Report is available in 14 languages and is sold all around the world. [21], [20, p. 4]

5.1 Dream Report Architecture

The architecture of Dream Report can be described as the structured system of steps and workflow of which the user goes through when creating a project in Dream Report. This chapter will cover the different stages that are encountered when a user is working with Dream Report. [20, pp. 6-7]

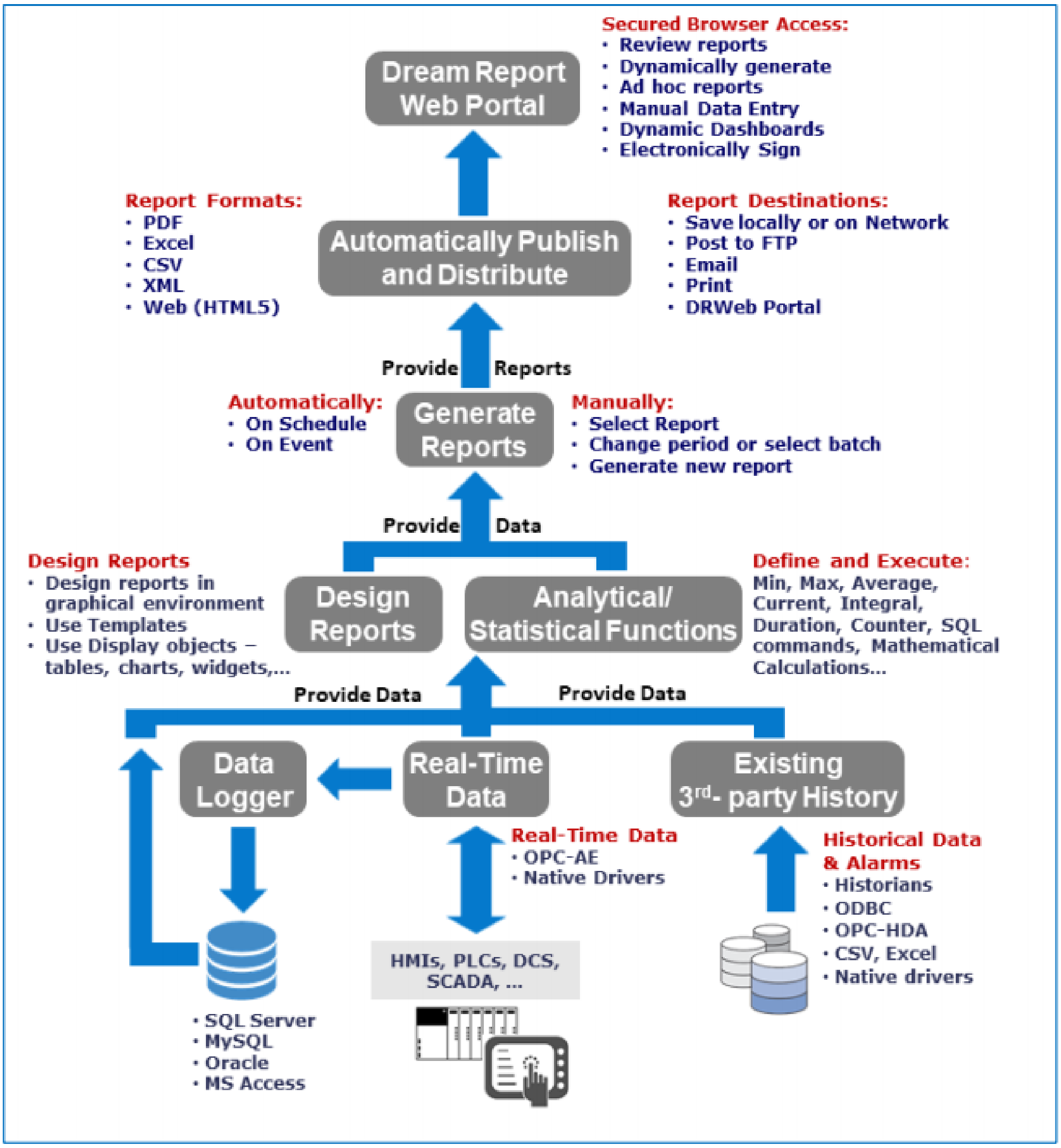


Figure 7: Dream Report Architecture. [20, p. 6]

5.1.1 Data Connection

As previously mentioned in the introductory chapter to dream report, the software can connect to many different data sources. These data sources can be real-time data and alarms, existing third-party archives and Dream Report also offers an option of logging real-time data into any open database, such as SQL Server, Oracle, Microsoft Access etc. The real-time data and alarms source allow for connectivity to SCADAs, HMIs, PLCs, etc. The last class of data sources are the existing third-party archives, these include process historians, open communication protocols such as OPC HDA, as well as any automation open database which supports ODBC and OLE interface.

The connection is established to the different sources through communication drivers that Dream Report has available in their program. There are different communication drivers available for different manufacturers to improve the flexibility of connection methods. Multiple communication drivers can be configured if connection to multiple data sources needs to be established. [21]

5.1.2 Data Logging

The Data Logging function is an optional method of providing data to Dream Report. If the user is connecting to a real-time data source, data can be logged to Dream Reports internal history database. By default, an MS Access database is created when starting a new project in Dream Report, however, this type of database is not recommended to use in a production environment, due to the storage limitations of the database. Natively, SQL Server is preferred but other databases like Oracle and MySQL can also be used through ODBC.

The Data Logger application can benefit many smaller applications if data is not already stored in repositories accessible by Dream Report. Usually, larger applications will implement the use of a data historian for improved data management. If an application is using a data historian supported by Dream Report, the Data Logging function is not necessary. This is because the data has already been logged/archived to Dream Report through the historian. [21], [20, pp. 10-12]

5.1.3 Design and Configurations

After data has been provided from a source, the report can then be designed and configured in the user-friendly environment of Dream Report studio. The studio offers a wide range of integrated graphical objects for which to display important data for analysis, these include charts, tables, widgets and images. There are built-in statistical functions in the studio which allow for raw data to be presented in the context of the report type, as an example, the function called “counter”, which can be used to count consumption (water, electricity etc.). It is heavily encouraged to use the options for page and report templates that the studio provides, as they make it easier for the user to apply already created templates and reuse them in new projects and even the same project that the template was originally created for. [21]

5.1.4 Report Management

The reports can be triggered on any schedule, whether it must be generated at specific times of the day, on a calendar basis or on a periodic schedule. Reports can also be generated on event, based on any real-time value acquired through a connection to an HMI or PLC. Lastly, there is also an option to generate reports manually, through a runtime management console, dynamic report generator or a remote application. The default output format of Dream Reports are PDF documents. Optionally the reports can be distributed and published in various other formats, such as Excel-files and CSV-files. Dream Report also allows for the output format of HTML5 webpages, where the report can be browsed to and interacted with using the web portal. [21]

5.1.5 Web Portal

The last part of the Dream Report architecture is the web portal. This web portal is self-configuring and allows the users to access and browse a list of any report in the currently running Dream Report project. Other features of the web portal include the dynamic generation of reports, where the user can enter manual values of data into the Dream Report database or set parameters for filtering for new reports. The web portal is accessible through any modern desktop browser or mobile device, such as android or iOS. The web portal is also optimized for a mobile layout. User security authentication can be setup in the web portal to limit the accessibility of reports only to specific users. [21], [20, pp. 26-28]

5.2 Data Aggregation

In order to create a wide range of high-quality industrial reports, all data from ongoing processes must be gathered. The data can later be aggregated for presentation in a summarized format. This is crucial for industrial operations, because an accurate presentation of high-quality data provides better decision making and thus leading to more successful and optimized operations.

As mentioned earlier, Dream Report offers a various number of options when it comes to collecting data. It leverages all real-time, historic and business data sources. The Dream Report software can connect to one or multiple data sources and collect their data and alarms, either through real-time database connections, historical archives or supported OPCs. [21]

As there are many different types of data that can be acquired with Dream Report, it is important when working with the software to be aware of what separates the different sources of collected data. Each type of data presents different methods of connecting to the reporting solution. These next few headlines will explain some of the different types of data available for aggregation a bit further.

5.2.1 Real-Time Data

Real-time data is information that is viewable immediately when it is available and delivered directly to an end user after it is collected. It presents data as it is “right now”. This does, however, not mean that the data gets delivered instantly, there are still things that can delay the data, such as network bandwidth between users. Real-time data is helpful for analytical projects, monitoring and presenting updated information quickly. [21]

5.2.2 Historical Data

Historical data is the opposite of Real-time data and is considered data from the past, this can be a minute, hour or a week of past data. This data cannot be altered, as the value is no longer being updated but it is stored actively as historical data is always generated. It is essentially data captured in real-time to be used for historical reference. For businesses, it is of major importance that historical data is maintained and managed properly. Historical data is good for analyzing past performances and noticing trends. This can be helpful to predict future performance rates. [21], [4, p. 123]

5.2.3 Alarm and Event Data

Alarm and event data is typically gathered from an alarm or event log in a control system. Alarm data is an alarm that has been set up to be triggered when a defined limit of a certain value is exceeded, while event data is a defined activity of an occurrence in the system. The triggered alarm warns the operator about a condition that could potentially cause a problem. It is important that alarms are managed properly to improve system performance and prevent a problematic situation from escalating further. There is an alarm management life-cycle concept provided in (Management of Alarm Systems for the Process Industries) ANSI/ISA 18.2. This concept offers defined analysis methods and solutions to help manage abnormal process events and alarm reporting, based on data generated from a control systems alarm and event logs. The ISA 18.2 functionality in Dream Report is provided with specific drivers that are built for many alarm repositories to aid in alarm management. [21]

5.3 Runtime Management

Configured project and reports in Dream Report is run by a module called Runtime Manager (RTM). This module can be run either as an application or a as a service by setting up Runtime Configuration, which can be setup for a specific Dream Report project. These configuration settings involve start up options for the RTM, message/error reporting to the Dream Report log file, default time period definition for Dynamic Report Generator (DRG), adjustable priority of runtime tasks to balance resources on a machine running multiple processes and parallel report generation, which enables the user to specify the number of reports that can be run simultaneously.

There is a feature for report generation prioritization called Report Generation Priority, where the user can specify how Dream Report manages priorities when generating reports. This priority-based feature is relevant both when generating reports in different modes and the same mode. The generation priority is by default based upon a list with five different priority levels and defined descriptions for each level. The numeric ID of the priority level cannot be changed but the user can specify their own description of each priority level. An important note is this priority mechanism only manages the report queue, this means that any report that is currently being generated will be completed before the generation of a new report can commence, regardless of its priority level. Below is a table of the default list of priorities for generation modes. [21], [20, pp. 23-25]

Table 5: Generation modes in Runtime Manager (RTM). [22, p. 408]

Generation Mode	Priority
On Schedule	Low
On Event	Medium
Runtime Management Console (RMC)	Medium
Dynamic Report Generator (DRG)	High
Web	Top

5.3.1 Runtime Management Console

The operations related to the Runtime Manager can be controlled with the Runtime Management Console (RMC). The RMC can be connected to a project created with Dream Report and various operations and functionalities for the reports within the project can be managed with this application. Projects can be managed in runtime, i.e., operations can be executed while the project is running. These operations include reloading and shutting down projects and reports, on demand managing of reports, enabled access to Dream Report help documentation, license info and different project options. Internet Information Services (IIS) can be configured through the RMC to access the Dream Report web portal for the currently running project. It is important to note that IIS needs to be separately enabled and configured with specific options in Windows before the installation of Dream Report. [22, pp. 409-412]

5.3.2 Dynamic Report Generator

The Dynamic Report Generator (DRG) is a module available in the RMC that enables predefined reports for the currently running project to be generated on demand by selecting a specific time period and output formats for the report generation. The dynamic modifications made will not be saved in the report layout. This enables the original report to continue to be generated according to its time definitions as before any dynamic modifications were made to its properties. Reports can also be dynamically generated from a command line and there is a separate section in the RTM that describes this function. This module is implemented as a standard application in Windows as well as an add-on component that is added as Dream Report is installed, this component can later be embedded in an HMI system. [21], [22, pp. 413-414]

5.4 Simulation Process

When the gathering of information about Dream Report was completed, it was time to start conducting simulations to see how the report generating process works in practice and if appropriate data can be presented. The simulations were conducted without the use of historian database, as Dream Report has its own simulation drivers available for project testing without connecting to “real” data sources. These simulation drivers were used as they offered the easiest method of connecting to a data source that is using the default MS Access database. The Dream Report version used in this process is version 5.0.

In this chapter, the report creation process will be explained along with the description of the simulations that were conducted. The chapter is split into the major steps that are done when creating a project in Dream Report. The images in this chapter are screenshots of the configurations made during the process. Here, when buttons and checkboxes in the images are referred to, quotation marks are used. When windows, sections and tabs in the images are referred to, the words are written in italic style.

5.4.1 Create a New Project

The very first thing that the user encounters when starting Dream Report, is the project window. This window lets the user choose between creating a new project, open an existing project, access a remote project, or to run a project wizard. In this window, the *New Project* was selected, and the project was given the name "Simulation".

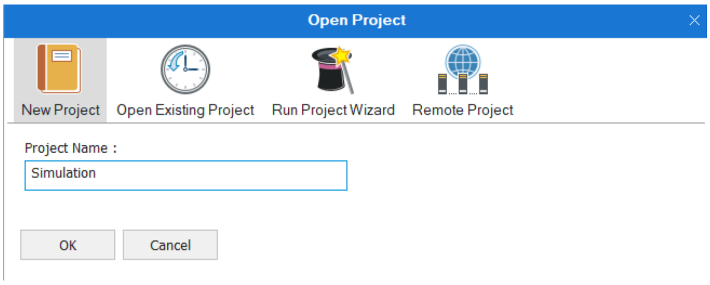


Figure 8: Project selecting window.

5.4.2 Configure Communication Driver(s)

When the project is created, the Dream Report *Communication Configuration Wizard* will open automatically and a connection to a data source can be established by selecting the appropriate communication driver and configuring it. In this case, the "Dream Report Simulation Driver for real-time values" will be used. The driver is selected by browsing to the Internal folder and Simulation Driver subfolder. After selecting the driver, it needs to be configured and given a logical name. The configuration is made by clicking the "Configure" button and selecting "DataSource_A", this data source is a text file called "dummy.cfg" where the data configuration of the driver can be made to better suit the simulation. The file can be accessed by browsing to the system folder of the Dream Report installation. When a name is selected and configurations are made, the driver is added to the *Defined Drivers List*. [22, pp. 604-605]

In this simulation, I named the data source “Simulator”, and this will always be chosen as a data source in this process.

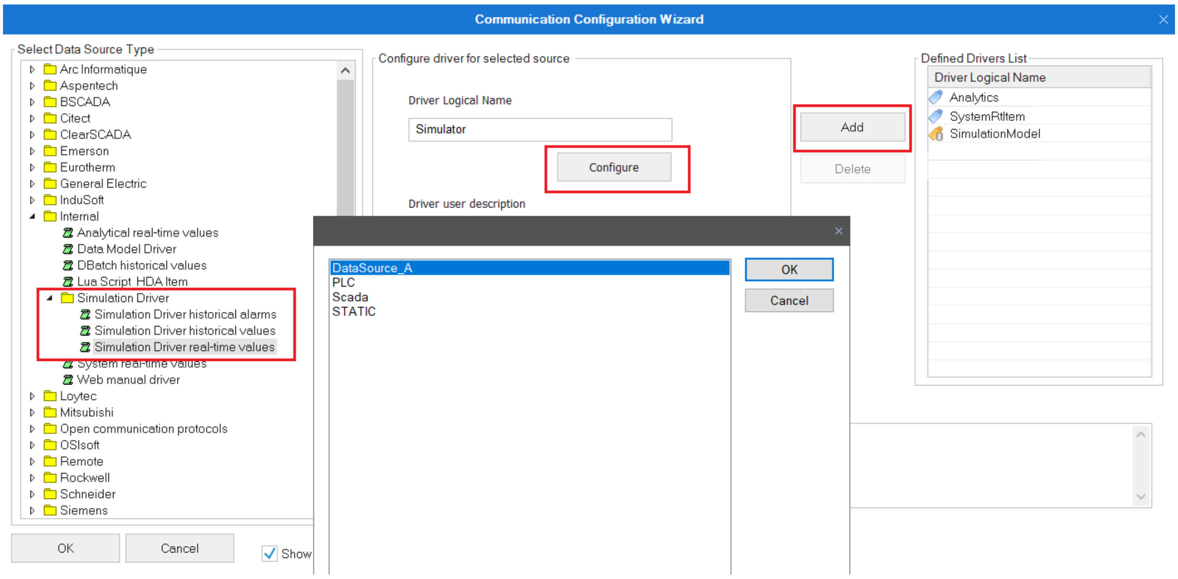


Figure 9: Communication driver configuration.

With the added driver properly set up, it is relevant to also select the appropriate database for the project. In the case of this simulation, the projects MS Access database is used. The database is configured by browsing to the *Project* tab and selecting *Database Definition*.

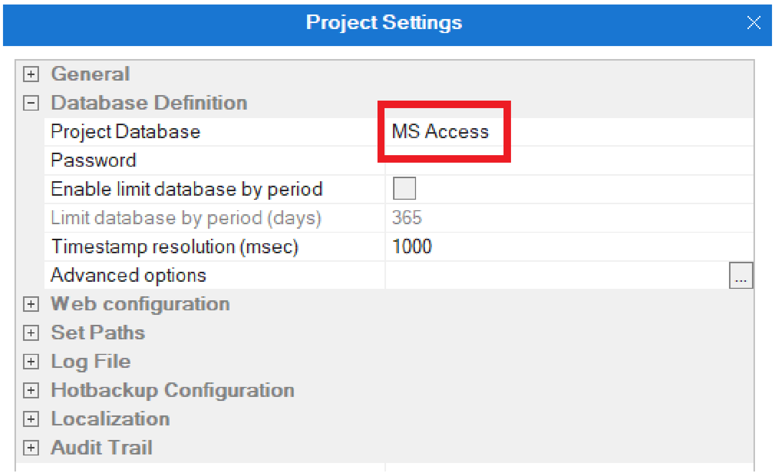


Figure 10: Project database settings.

Verify that *MS Access* is used as the project database in the *Database Definition* section of the *Project Settings* window. It should be selected as a default when creating a new project.

5.4.3 Set up Logging Group(s)

Because real-time data is being used, a logging group for the data must be set up.

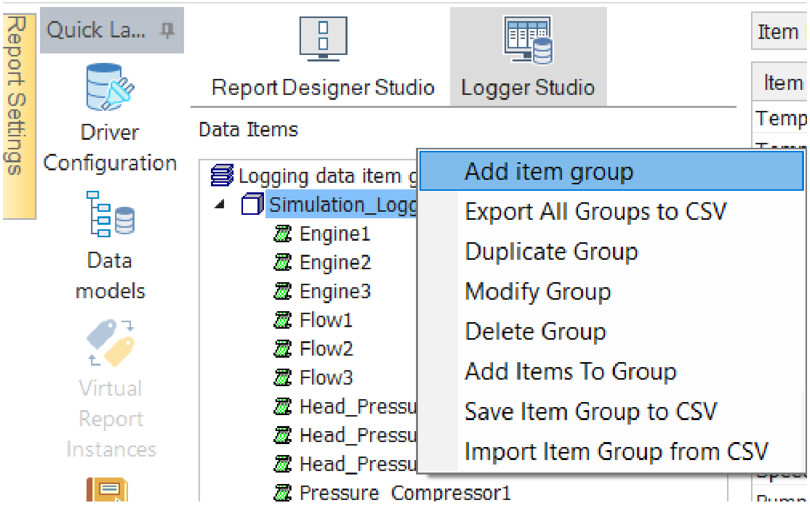


Figure 11: Adding of data logging group.

- Add a new item group in the logger studio by right clicking on the "Logging data item group" and selecting "Add item group".

When logging data into Dream Reports history database, logging groups of the data items can be created with separate logging conditions. Here the logging groups can be separated into groups such as process data and energy data. Separate logging modes and conditions for each group can be set up and then, items from that group are only logged if the specified conditions are met. Typical logging conditions can be on change, on event, and on time. The communication between the data source and Dream Report can be verified by clicking on the "Start Monitoring" or the "Force Read" button. This will indicate whether the data status is good and if the values are visible in Dream Reports internal database. [22, pp. 98-99]

The *Add item group* window will appear. In the *General tab*, a logical name must be given to the item group and the data source must be defined in the *Group Source* section. Tags are added by clicking on the "Add items" button and selecting the tags to be logged. The *Record Rate* and *Record Conditions* window can be used to define certain logging rates and conditions of the group, such as data only being logged when a change in value occurs. However, these options are not used in this simulation.

- Define a logical name for the logging group in the *Group Name* field.
- Select the data source "Simulator" in the *Group Source* section and open the *Select Data Items* window by clicking on the "Add items" button.

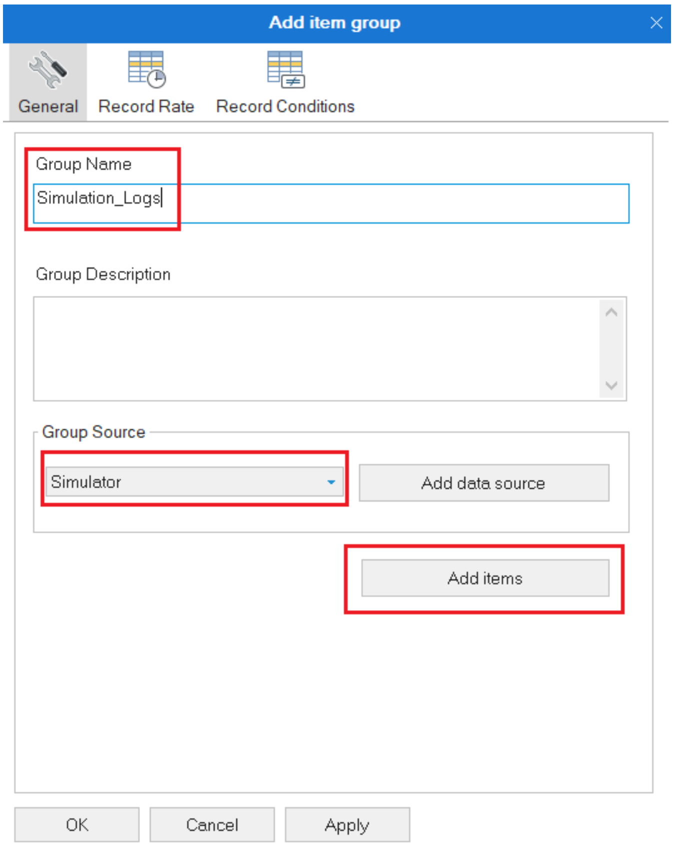


Figure 12: Data source definition of logging group.

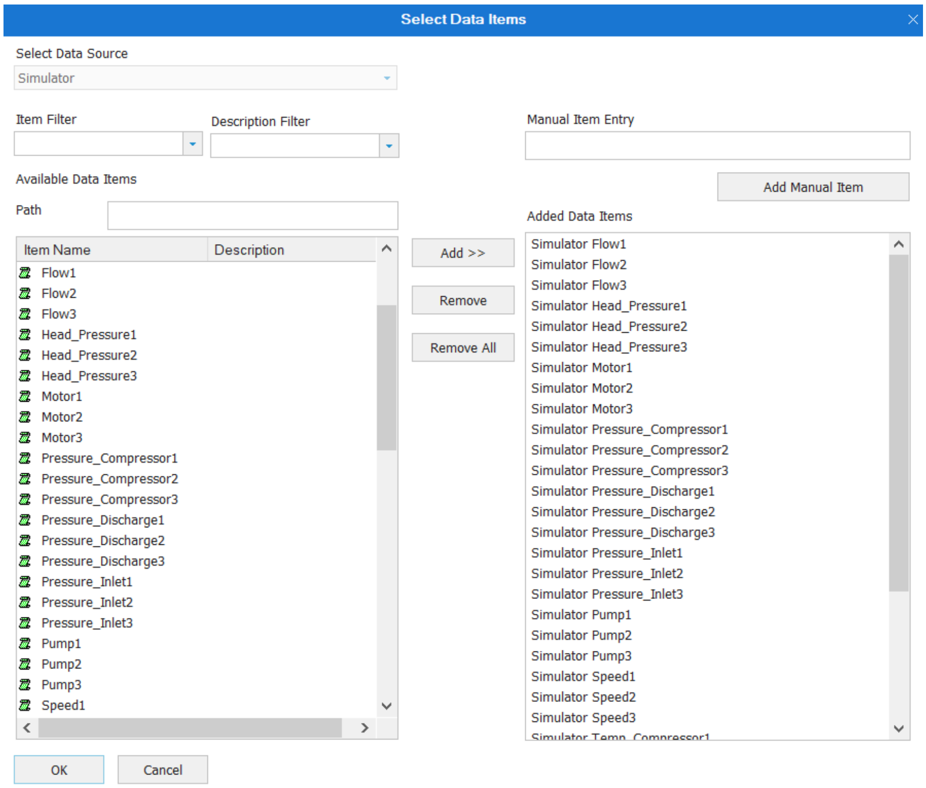


Figure 13: Available and added tags from data source.

- Select all the tags that are going to be used in the report and click on the "Add" button to add the selected tags to the logger studio.

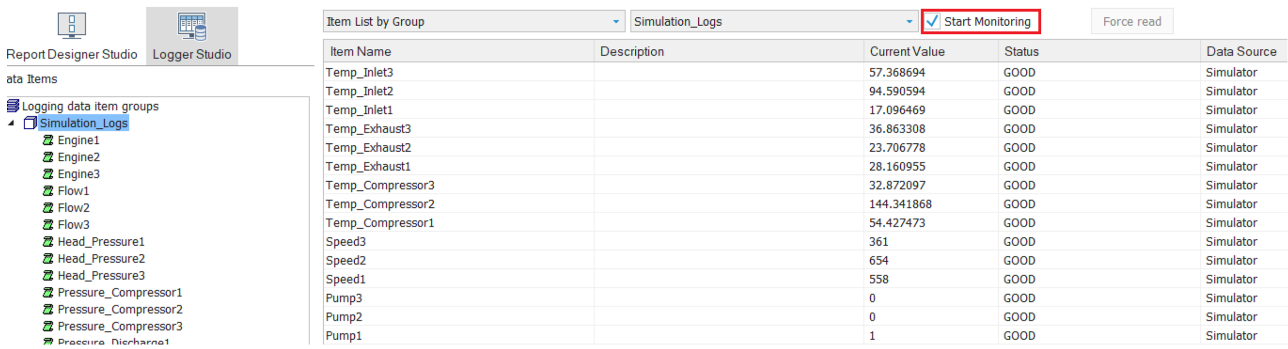


Figure 14: Monitoring of added tags in logger studio.

- Check the "Start Monitoring" box to confirm that the values of the selected tags are being logged into Dream Reports internal database. Here can the current values, status and data source of the tags be viewed.

5.4.4 Create a New Report

After a data source connection has been set up along with a logging group, a new report can be created in the designer studio. Create a new report by right-clicking on the "Project Reports List" folder and selecting "New Report".

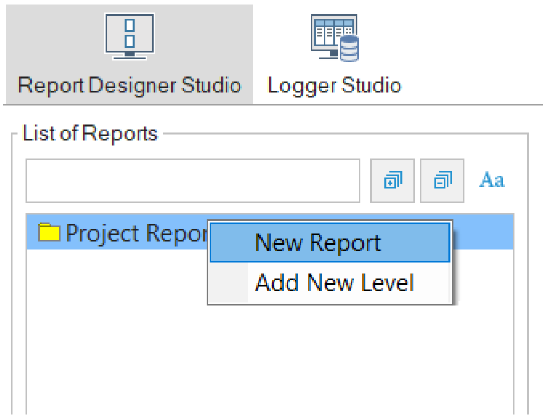


Figure 15: Creation of a new report.

The *Report Settings* window will appear, and the report name can be written in the *General* tab. Continue to browse to the *Report File Format* tab and select "Web" and ".PDF**"** as report targets. PDF can also be selected to be opened after the report has been generated.

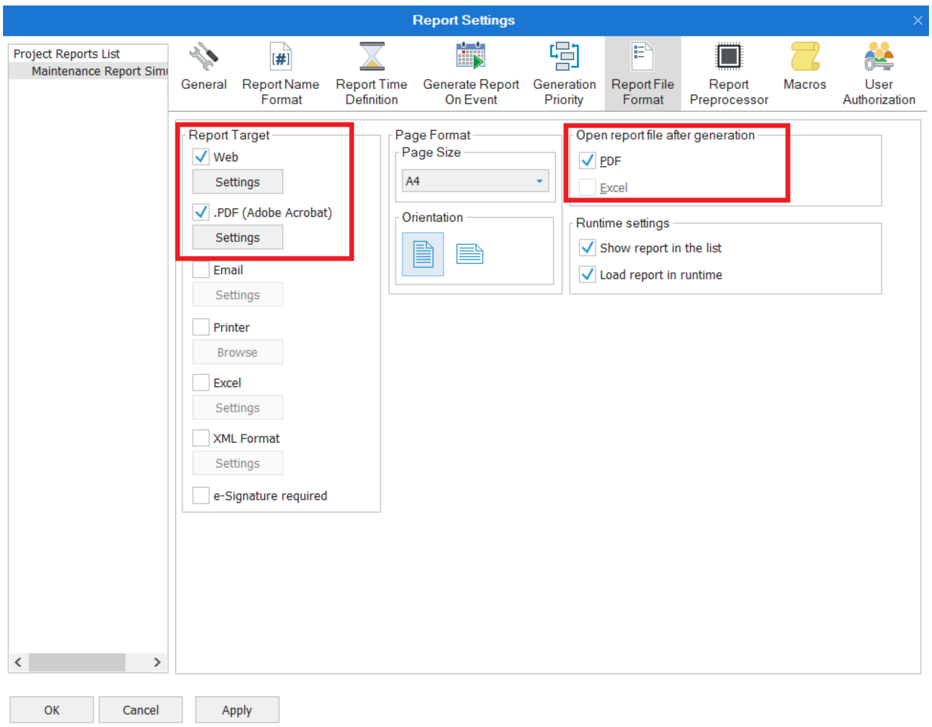


Figure 16: Report file format settings.

Additionally, a report time definition and generation conditions can be set up in the tabs called *Report Time Definition* and *Generate Report on Event*. However, this set up is not made with this simulation.

5.4.5 Report Configuration

The configuration part is easily the longest part of the report creation process. This is where most configurations are made and the most common are, selecting objects to display data with appropriate settings/filters and defining a time period from which the data is being gathered. From the *Home* tab, there are general display options available for text, lines, static or dynamic images and fonts can also be adjusted, such as colors and thickness. [22, p. 115] In the *Elements* toolbar on the right side in the designer studio, the user can choose amongst many different dynamic objects to display data with. An important note is that in order to display any data from a database, the data source and associated tag(s) must be selected for the object. To place objects on the report, the user must simply "draw" a rectangle on the page for the configuration window to appear for the selected object.

For the simulation, a page template is created so that it can be added to a different report without having to draw and place the same objects again. A "Dynamic Text Object" is placed on the report and the report name is selected as the dynamic data to be displayed. Additionally, the "Text Box" object is selected and "Generated on:" is written in the box, along with a "Date and Time Object" where the *option for dynamic generation* is selected as "report generation time".

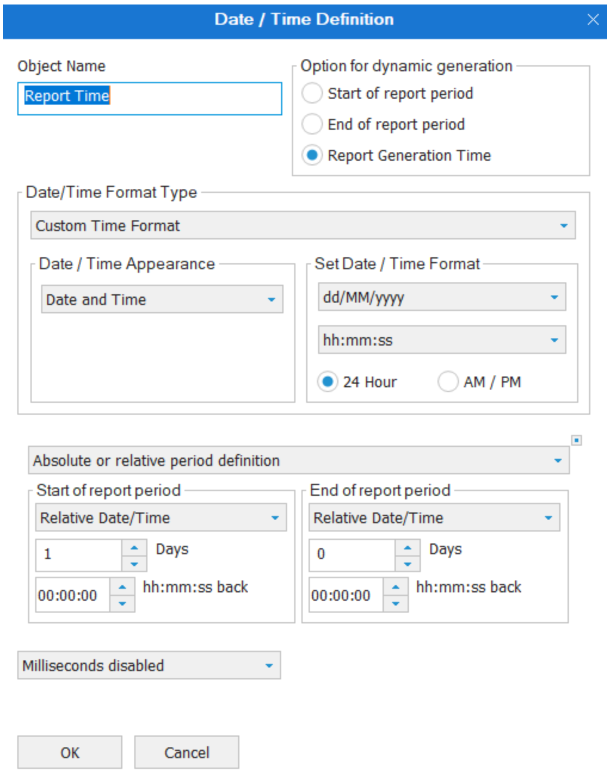


Figure 17: Report date/time definition.

To finalize the simulation template, a horizontal line is drawn to separate the text, image and date/time object from the area where the tables will be displayed. To draw the horizontal line, click on the "Line" icon in the *Drawing* section in the upper part of the designer studio. The width of the line can also be adjusted by clicking on the "Line Width" icon, which is in the same section. The green arrow markers on the vertical ruler can be moved to adjust the space of the page that will be used by the objects and tables created.

Figure 18: Finalized report template.

As the template is done, the tables can be added to the report. The first table to be used is the “Step Table”. The step table creates a table over a time period, where each step (row) in the table can have a predefined time with calculations being made over each step period. For the simulation, 6 columns are selected as the number of columns the table will contain. The time period is defined as a fixed period of 1 hour and the step period is selected as 10 minutes. The table will then display 6 columns where all columns represent the total time period of 1 hour and each step of the table represents 10 minutes of data gathered over the last hour. [20, p. 20]

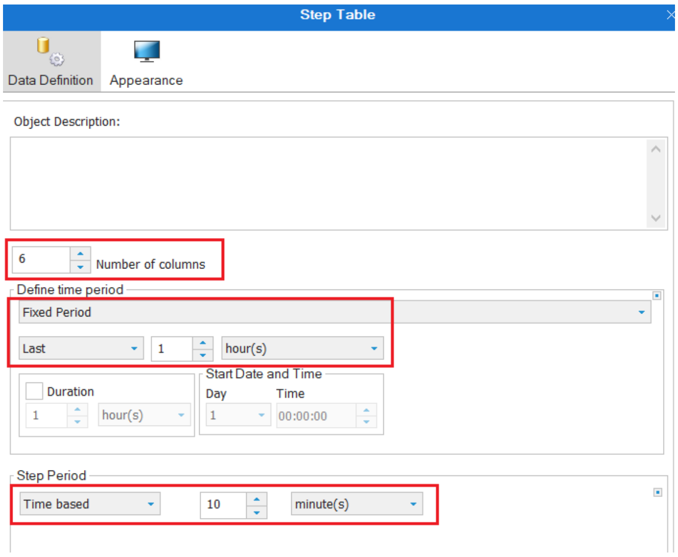


Figure 19: Step table data definition.

After the data definition is done, the appearance of the table can be configured by browsing to the *Appearance* tab next to the *Data Definition* tab. The “Display Table Name” box must be checked for a table name to be written. As the number of columns were already selected, the rest of the column configurations are done by defining the column type and caption. The caption can be specified by double-clicking on the caption name in the *Display Columns* section and typing a logical name. If no name is specified, the name of the statistical function used for the column will be displayed instead. The column type is configured by selecting the column that the user wants to modify and click on the box with the three dots, it is important to select the column first as the box will not be visible otherwise.

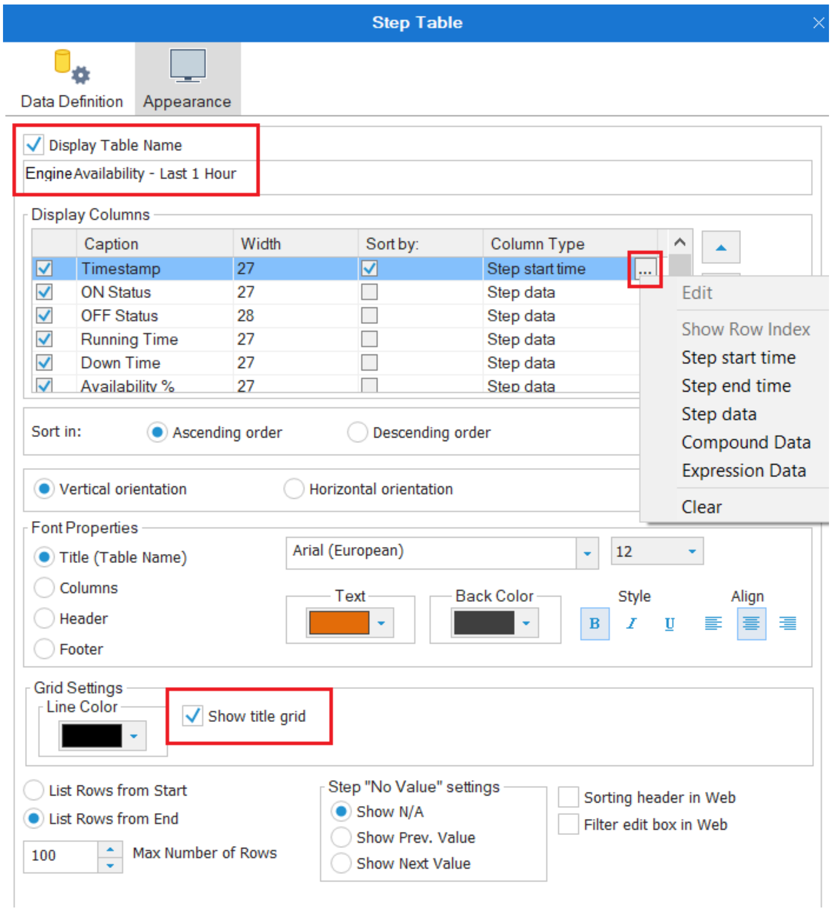


Figure 20: Step table appearance configurations.

The first column is the timestamp, and it is configured by selecting "Step Start Time" as the *Column Type*. The *Custom Time Format* is selected with the default settings. For the rest of the columns in the table, configuring is done by selecting "Step Data" as the column type. The *Single Data Definition* window will appear, and an object name must be defined in order to configure the column. For the table to display any data, a data source and associated tag(s) must be specified. First, a definition of how historical data is gathered must be defined in order to select the appropriate data source and tag(s). As data is being logged into the internal Dream Report history database, the checkbox "Dream Report History" is selected. With the database specified, the data source and the appropriate tag can be selected. This is done by clicking on the box with three dots and selecting "Simulator" as the data source and adding "Engine1" as the tag. A statistical function must be added for the calculation to be correct, in this case the "ON Counter" function is selected. This function calculates how many times the value changed from zero to a non-zero value, during the defined time period. [22, p. 152]

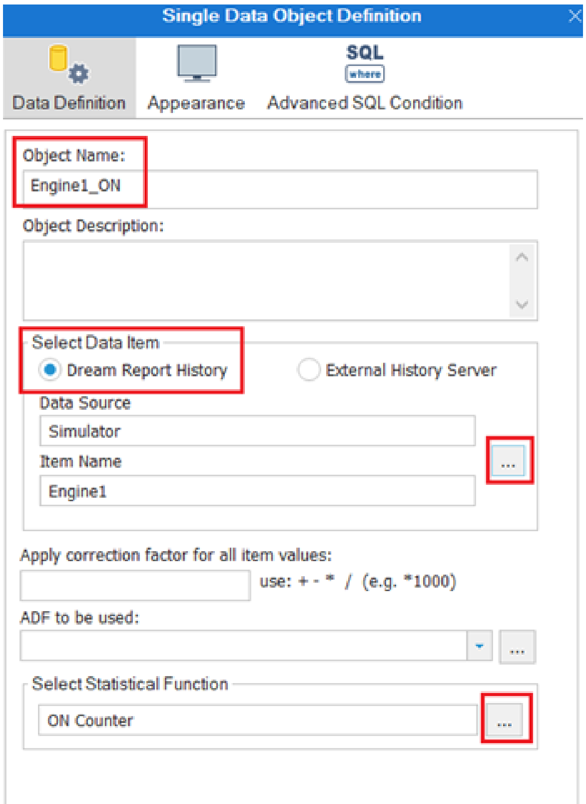


Figure 21: Single data object definition.

Similar steps are to be performed for the rest of the columns in the table. The only difference is a new object name and a different statistical function that must be defined.

- **Column3:** define "Engine1_OFF" as object name and "OFF Counter" as statistical function. The Off Counter function calculates how many times the value changed from a non-zero value to a zero value, during the defined time period. [22, p. 152]

- **Column4:** define "Engine1_RT" as object name and "Running Time" as statistical function. The Running Time function calculates the time period when the value was recorded as a non-zero value into the database. [22, p. 153]

- **Column5:** define "Engine1_DT" as object name and "Down Time" as statistical function. The Down Time function calculates the time period when the value was recorded into the database as a zero value. [22, p. 153]

- **Column6:** define "Engine1_Availability" as object name and "System availability" as statistical function. The System availability function calculates which part of the total defined time period the system was available (the item had a non-zero value) [22, p. 153]

The font and text configurations are made in the *Appearance* tab and the table is named "Engine Availability". The result is a completed step table with parameters for an engine performance analysis.

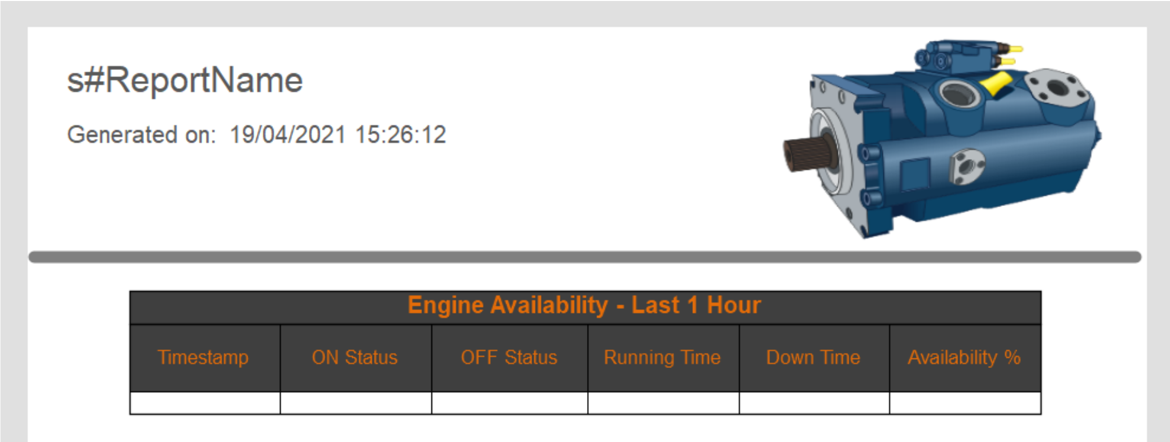


Figure 22: Finalized step table on the report template.

The report template in Figure 22 has been altered a little bit compared to the original template in Figure 18. The changes that were made was the color of the "Dynamic Text Object", "Text Box", and the "Date and Time Object". A picture of an engine was also added to the template to add a little visual flair. Dream Report has pre-stored pictures in the image folder that is created when the program is installed. The pictures available from Dream Report can be found by browsing to ODS > Dream Report > ClipArt > Images. From there, the user can choose any industry specific image that could be appropriate for the report. [22, p. 116]

A bar graph is configured to display availability over a different step period compared to the step table. The bar graph object is placed under the step table that was recently created and its data definition is configured first. In this configuration the data source and tag(s) must be selected for each bar that is added to the graph and the time period must be defined in conjunction with the bar period. Multiple tags can be displayed here with separate bars. When the data is defined, the color of the bar(s) can be selected and a legend to display for the selected bar can be written.

- Define *Chart Based On* to be set as "Value" in the combo box, define the time period to be fixed for the last 1 hour and each bar period to be time based for the last 10 minutes. The same configuration is made for all bars that are added.
- Check the box for "Dream Report History" in the *Get Data From* section in order to be able to select the tags for the data being logged. Select "Simulator" as data source and pick the "Engine1" tag. Pick the "Engine2" and "Engine3" tags for the next two bars to be added.
- Select the "System availability" calculation to be done in the *Process Value* section. The same calculation is made for each bar.
- Write the name of the tag used in its selected bar in the *Legend* field and select the preferred color in the color box. Repeat this for the two added bars.

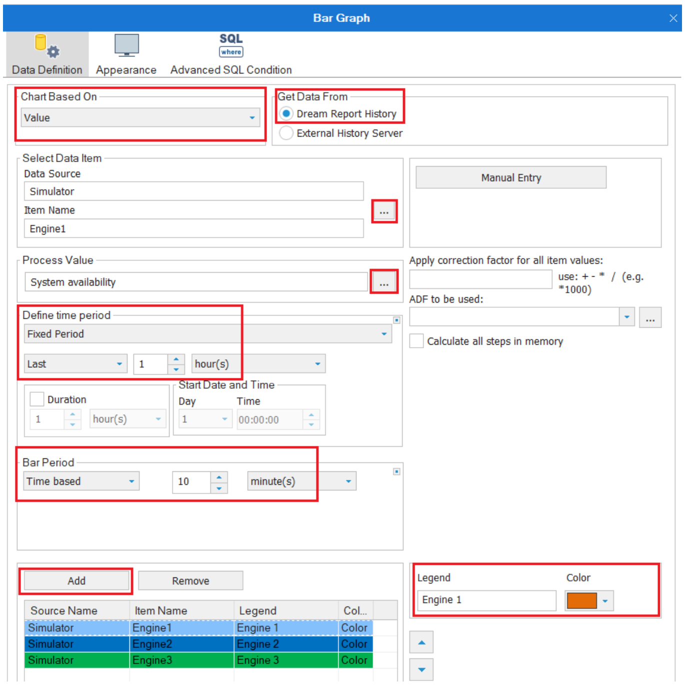


Figure 23: Bar graph data definition.

After the data has been defined, the appearance of the bar graph can be altered in the *Appearance* tab. The graph is given a name to indicate what type of information that is being displayed. The "Display Legend" box is checked as it was written for each tag in the data definition window. The *Scale limit mode* is set to "Manual or item-based" with the values ranging from 0 to 100. This is done to optimize the appearance of the graph, as availability is calculated in percentage. Text, font and color alterations can be made according to preference in the *Display Style* section.

- Give the graph a logical name.
- Check the "Display Legend" box in the *Display Options* section
- Select "Manual or item-based" in the combo box and set "from 0 to 100" in the *Scale limit mode* section.

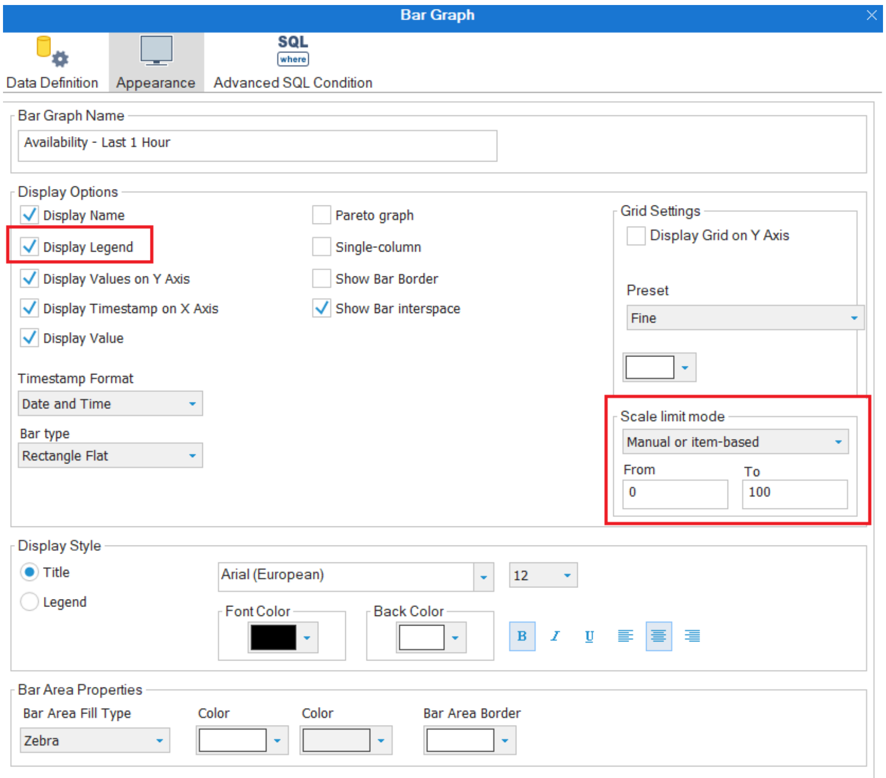


Figure 266: Bar graph appearance definition.

The last table to add to the report is an "Automatic-Statistical Table" (AST). The AST summarizes large amount of data in a grid, with the user only having to specify a time period and what statistics to be calculated and displayed. Multiple tags can be added along with different calculations in each row. Basic calculations for a maintenance report can be current, minimum, maximum and average values. [20, p. 20]

- Select "Dream Report History" in the *Get Data From* section, select the tags to be used by clicking on the "Edit List" button. Select the "Simulator" data source and add the appropriate tags.

- Define the time period to be "Fixed Period" and "Last 1 hour" in the *Define time period* section.

- Check the boxes for each function in the *Select Function* section. Select "Current Value", "Maximum", "Minimum" and "Average" functions.

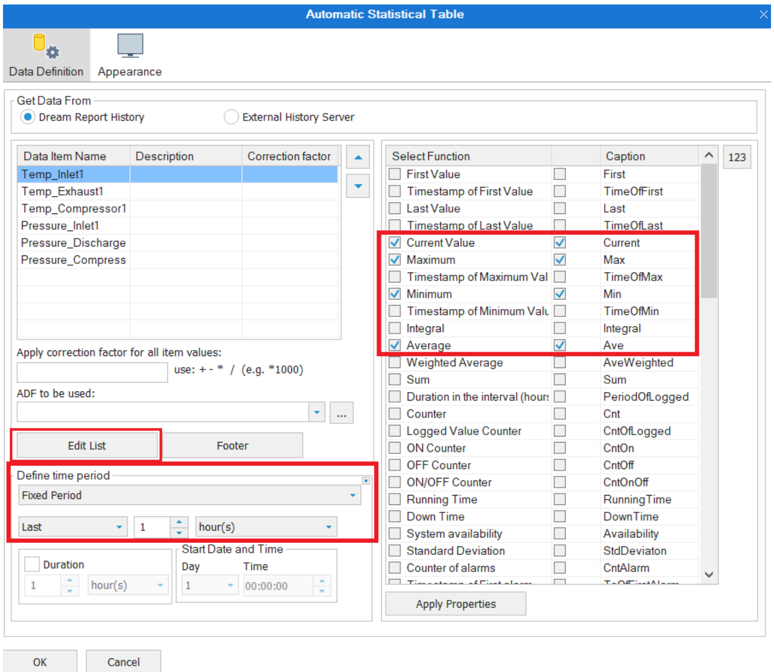


Figure 267: AST data definition.

In the *Appearance* tab the table name is defined, and the text and fonts can be adjusted as well according to preference.

- Define a logical name for the automatic statistical table.
- Apply preferred fonts, colors and text.

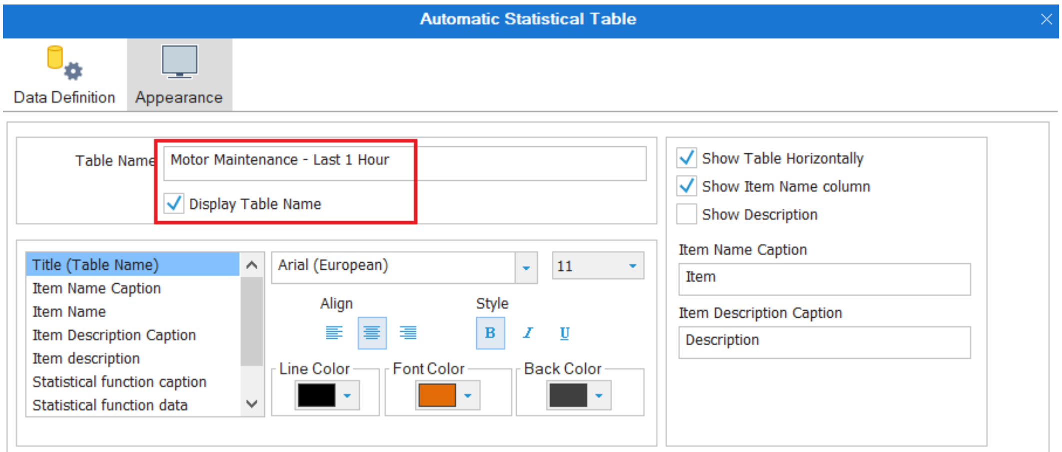


Figure 268: AST appearance configuration**.**

The "Automatic Statistical Table" or AST allows the user to quickly create and modify a "summary" table. As a list of tags is selected for the table, one or more predefined calculations for the tags can be set. The tags can have their own unique set of calculations or share the same calculations. The options of calculations available are many, as there are over 40 predefined functions to choose from. Additionally, there are options available for specifications to be set for each calculation, such as displaying units of the result and precising the result after decimal point. [22, pp. 203-205]

To get the best representation of data in an AST, or any table or graph in general, it is important to select a common time period of the calculations made in a Dream Report table. This is to prevent confusion when interpreting the data from a report, as data calculated over different time periods in a single table is bound to be inconvenient for analysis and the report might be misinterpreted.


s#ReportName

Generated on: 18/04/2021 21:35:04

Engine Availability - Last 1 Hour					
Timestamp	ON Status	OFF Status	Running Time	Down Time	Availability %

Engine Maintenance - Last 1 Hour				
Item	Current	Max	Min	Ave
Temp_Inlet1	#Temp_Inlet1_Current	#Temp_Inlet1_Max	#Temp_Inlet1_Min	#Temp_Inlet1_Ave
Temp_Exhaust1	#Temp_Exhaust1_Current	#Temp_Exhaust1_Max	#Temp_Exhaust1_Min	#Temp_Exhaust1_Ave
Temp_Compressor1	#Temp_Compressor1_Current	#Temp_Compressor1_Max	#Temp_Compressor1_Min	#Temp_Compressor1_Ave
Pressure_Inlet1	#Pressure_Inlet1_Current	#Pressure_Inlet1_Max	#Pressure_Inlet1_Min	#Pressure_Inlet1_Ave
Pressure_Discharge1	#Pressure_Discharge1_Current	#Pressure_Discharge1_Max	#Pressure_Discharge1_Min	#Pressure_Discharge1_Ave
Pressure_Compressor1	#Pressure_Compressor1_Current	#Pressure_Compressor1_Max	#Pressure_Compressor1_Min	#Pressure_Compressor1_Ave

Figure 269: Finalized report in designer studio.

When the tables and the graph has been configured, the report should look like the one in Figure 27. The report displays two tables, a step table for engine availability and an AST for engine maintenance parameters. The bar graph is also added to get a visualization of the availability for multiple engines.

The original time period that was intended for the step table was a daily time interval where each step would represent an hour of the day. The bar graph also had another intended interval, where data over a month would be displayed as a weekly availability. Because the simulation drivers were used, data would have had to be logged into Dream Reports internal database for 24 hours for the step table and for a month for the bar graph. For this to work as intended, I would have had to keep my computer constantly running with Dream Report being active and logging the data. That is why the interval over an hour was used instead of a weekly interval.

5.4.6 Report Generation

When the report has been created, the project must be run for the report to be generated. This is done by selecting “Run Project” and the *Runtime Management Console* window will appear after the project has finished loading.

- Select the created report in the *Available Reports* section.
- Check the “PDF” box in the *Reports* window to select the file format that the report will be generated with.
- Click on the graph button to generate the report.

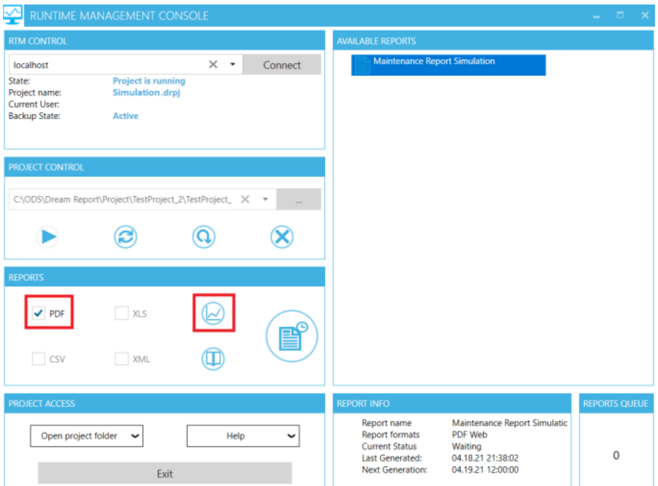


Figure 270: Runtime management console.

The generated report of this simulation is presented as an appendix under Chapter 8 in this thesis. In the appendix, a report of a pump is also presented with a similar layout as the engine report. The only difference is that the tags are changed from engine to pump tags, and the bar graph is replaced by a flow chart.

The reason why the pump report is not documented, is because Wärtsilä did not have any corresponding pump tags available with Historian. The pump report was simply made with the goal of educating myself further to develop my knowledge of reporting with Dream Report.

6 Result

The goal of this thesis was to present a report suggestion of a maintenance report using Dream Report and Historian as the data source. There are not many differences in the configuration compared to the simulation, the main difference is the use of Historian as a data source.

The report for the result was done in Wärtsilä's virtualized server-environment on a virtual machine (VM) that is running Dream Report. The VM is also connected to sWOIS and Historian.

First, the Historian communication driver must be set up.

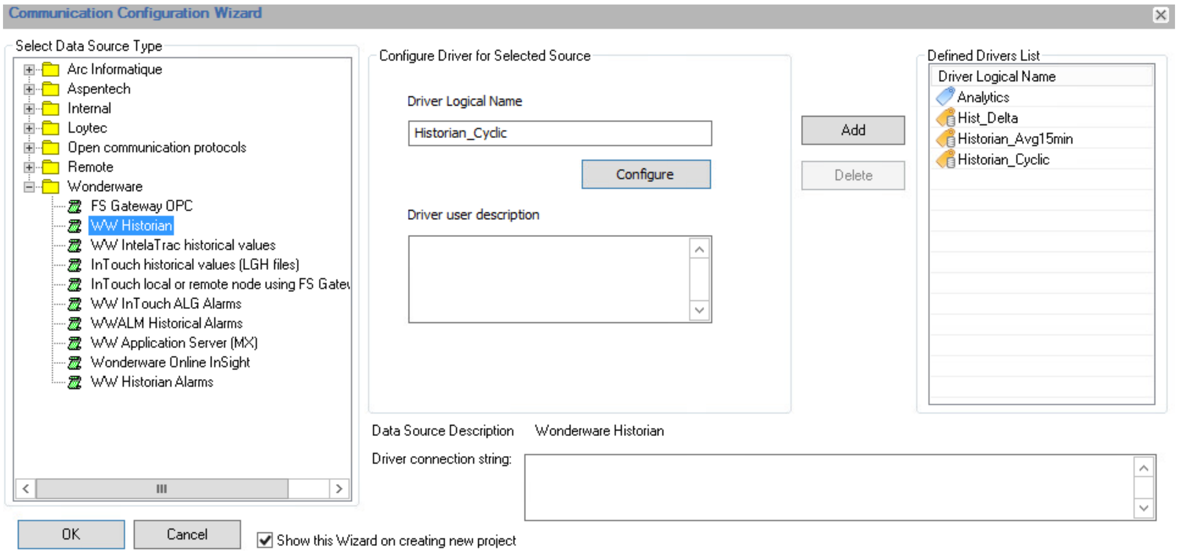


Figure 271: Historian driver selection.

The appropriate Historian driver is found by browsing to the Wonderware folder and selecting "WW Historian" as the driver to be used. An important note here is that Wonderware was the previous name of AVEVA, and therefore the driver and folder is named Wonderware. The Dream Report version on this VM was an older version compared to the 5.0 version used in chapter 5.4.

The configuration of the driver is done by defining the IP-address of the Historian computer and the name of the database to connect to. If there is any user authentication, credentials must be verified as well. Connection is tested by clicking on the "Test Connection" button.

The retrieval mode is selected as cyclic, as it is the retrieval mode of choice when generating daily reports. The value of the resolution is set to 3600000 milliseconds to represent 1 hour. If any other time resolution is desired, then the appropriate value of that time must also be set in milliseconds.

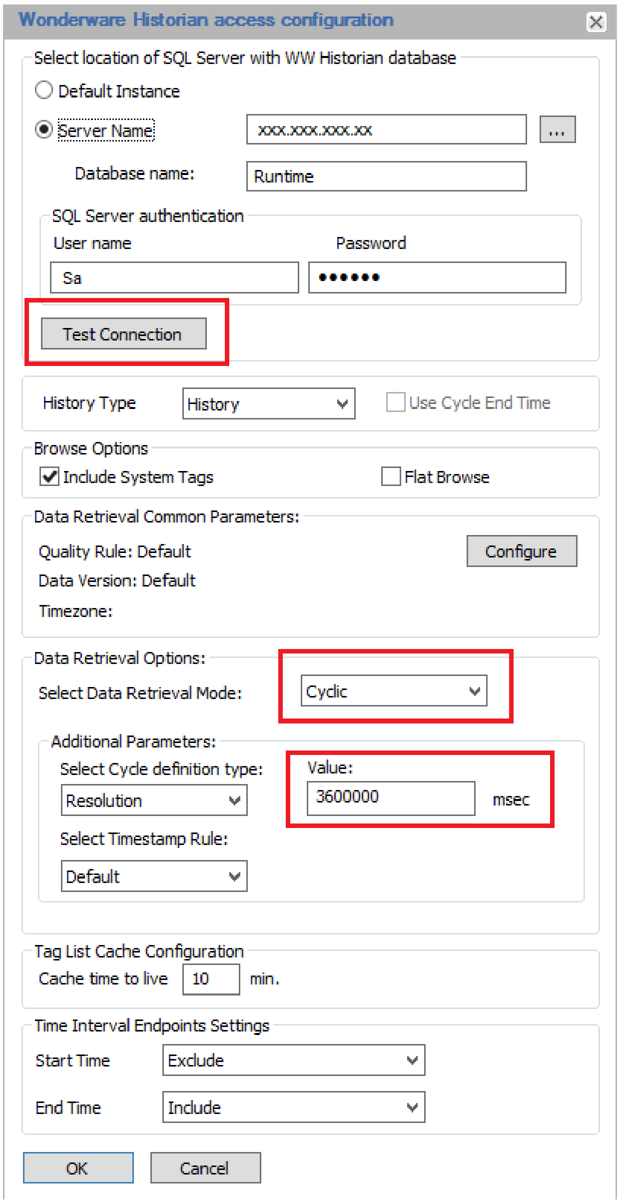


Figure 272: Historian driver configuration.

The project database must be defined as MS SQL Server and the computer hosting the Historian database must be defined as well. User credentials for the Historian must be entered before the connection can be verified. The connection is verified by clicking on the box with three dots in the *Test Connection* field.

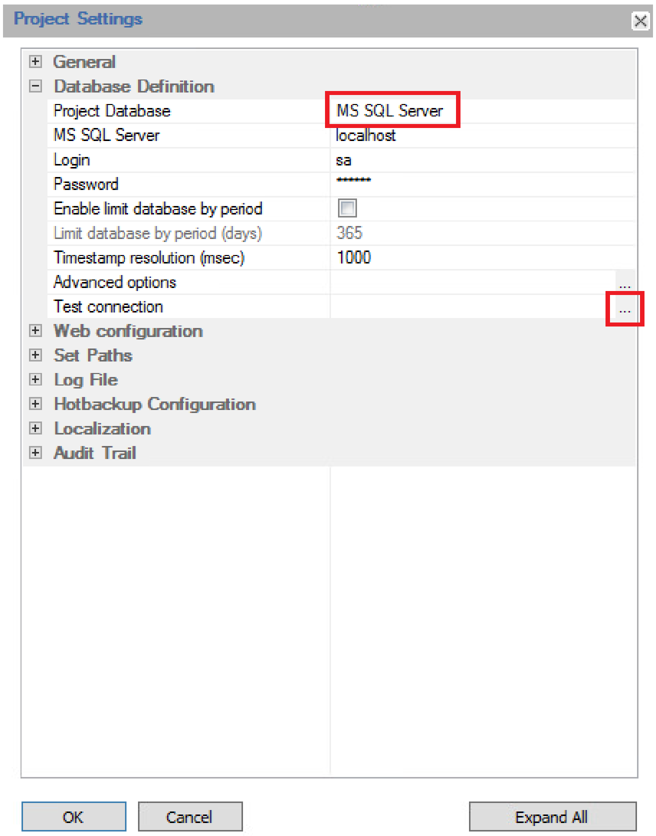
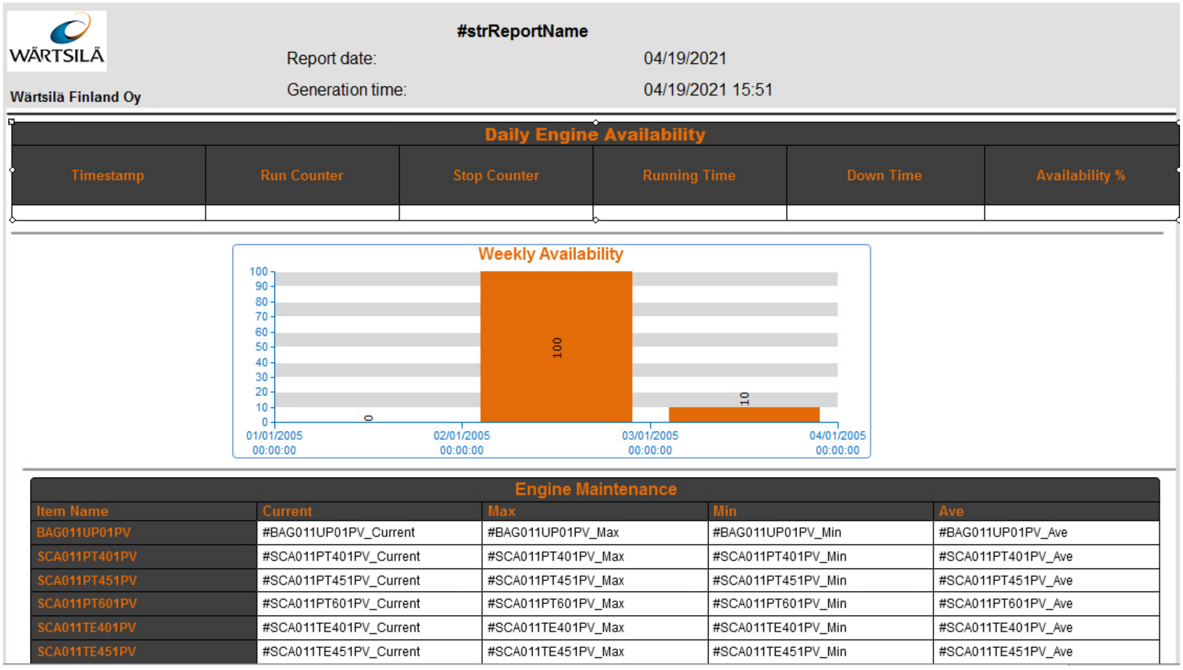


Figure 273: Project database definition.

When configuring the data definition for the objects used in the report, it is important to note that when using an external data source (historian), the “External History Server” checkbox must be selected instead of “Dream Report History” in the *Get Data From* and *Select Data Item* sections. Refer to Figures 21 and 23 for data definition using Dream Report’s internal database and note how the configuration would change if an external data source were used.

After a valid connection to Historian was established, the report could be configured in the Dream Report studio. The tables and the graph were configured in the same manner as presented in the "Simulation Process" chapter of this thesis. A major difference besides the communication driver, was the selected time period that is used in the report settings and the reporting objects. An appropriate time period was already calculated in Wärtsilä's report templates for production and asset management, these calculations were not to be modified.



Item Name	Current	Max	Min	Ave
BAG011UP01PV	#BAG011UP01PV_Current	#BAG011UP01PV_Max	#BAG011UP01PV_Min	#BAG011UP01PV_Ave
SCA011PT401PV	#SCA011PT401PV_Current	#SCA011PT401PV_Max	#SCA011PT401PV_Min	#SCA011PT401PV_Ave
SCA011PT451PV	#SCA011PT451PV_Current	#SCA011PT451PV_Max	#SCA011PT451PV_Min	#SCA011PT451PV_Ave
SCA011PT601PV	#SCA011PT601PV_Current	#SCA011PT601PV_Max	#SCA011PT601PV_Min	#SCA011PT601PV_Ave
SCA011TE401PV	#SCA011TE401PV_Current	#SCA011TE401PV_Max	#SCA011TE401PV_Min	#SCA011TE401PV_Ave
SCA011TE451PV	#SCA011TE451PV_Current	#SCA011TE451PV_Max	#SCA011TE451PV_Min	#SCA011TE451PV_Ave

Figure 274: Result in the Dream Report designer studio on VM.

The result of this thesis was a report suggestion of an engine maintenance report made using Historian as a data source and Dream Report as the reporting software in Wärtsilä's server-based virtualized environment.

The report using Historian is designed in a similar manner as the engine maintenance report presented in the appendix. The noticeable difference between the two reports is that the bar graph only displays the availability of one engine in the report using Historian, compared to the simulated report which displays the availability of three engines in the bar graph. Additionally, the Wärtsilä logo and design of the standard page template already in use for Wärtsilä's daily and monthly asset reports, were implemented in this report suggestion.

7 Conclusion

This project was concluded with a presentable template of an engine maintenance report. The concluded report displays the essential parameters related to engine maintenance and availability. When reviewing this report, a performance analysis can be done.

The tags that were chosen for the report were based on engine genset 1. However, these tags did not have any data available at the moment when this project was conducted, which means that when the report is generated, the values are presented as “N/A” (Not Available). Although the data from the tags was not available, the report template can still be used for the demand that was set for the report, as was shown with the working simulation in Dream Report. Tags in Historian that were not related to the engine maintenance report were tested to confirm that the connection was working, these tags were PLC parameters for current and voltage values. I would have wanted the engine related tags to be working to better present the resulting report, but despite them not having any value as this project was concluded, I am still happy with the outcome of the thesis result.

The most challenging part about this project was to identify why the report sometimes displayed values as “N/A”. There were many potential sources of this issue and often the issue could not be identified immediately, but only after ruling out other contributing factors. A good thing about using the Data Logger function of Dream Report, is that any connection issues to the data source can quickly be ruled out by monitoring the logger studio to see if there is any data available. If data is available and the status is “good”, then the issue is most likely in the object configurations of the report. Usually, the time period defined in objects could be wrong, and it is important to first let the data be logged for the time that it is meant to be displayed in the report when using the Data Logger.

A valuable lesson that I have learned during this thesis, is the importance of managing and presenting data correctly. Not only is it important that the correct data is extracted and managed, but also that it is presented properly so that everyone understands the report.

As a final note, I would like to express my gratitude towards Matti Anttila and the team at TS Plant Automation for giving me the opportunity to write my thesis for Wärtsilä and complete my bachelor’s degree in engineering.

8 References

[1] Wärtsilä. (Accessed January 18, 2021). wartsila.com [Online]. Available: https://www.wartsila.com/about

[2] Wärtsilä. (Accessed March 6, 2021). Becoming Wärtsilä [PDF].

[3] Wärtsilä. (Accessed March 10, 2021). Plant Automation [Intranet]

[4] B. R. Mehta & Y. J. Reddy, *Industrial Process Automation Systems: Design and Implementation.* Elsevier, 2014. ProQuest Ebook Central. Accessed: April 13, 2021. [Online]. Available: https://ebookcentral.proquest.com

[5] S. G. McCrady, *Designing SCADA Application Software: a Practical Approach.* Elsevier, 2013. ProQuest Ebook Central. Accessed on: April 10, 2021. [Online]. Available: https://ebookcentral.proquest.com

[6] K. L. S. Sharma, *Overview of Industrial Process Automation,* 2nd ed. Boston, Joe Hayton, 2016. ProQuest Ebook Central. Accessed: April 18, 2021. [Online]. Available: https://ebookcentral.proquest.com

[7] *Programmable controllers – Part 3: Programming languages*, IEC 61131-3, 2013

[8] PLCopen, *Status IEC 61131-3 standard*, (n.d.). Accessed April 12, 2021. [Online]. Available: https://plcopen.org/status-iec-61131-3-standard

[9] Wärtsilä. (Accessed April 18, 2021). sWOIS Network Implementation [Intranet]

[10] Wärtsilä. (Accessed March 3, 2021). WOIS – Wärtsilä Operator's Interface System [Intranet]

[11] Wärtsilä. (Accessed March 3, 2021). WOIS User's Manual [PDF]

[12] Wärtsilä. (Accessed May 5, 2021). Advanced Automation Training sWOIS [PPT]

[13] Wärtsilä. (Accessed May 5, 2021). sWOIS Release Presentation [PPT]

[14] AVEVA. (Accessed April 21, 2021). Datasheet AVEVA Historian [PDF]. Available: https://www.aveva.com/content/dam/aveva/documents/datasheets/Datasheet_AVEVA_Historian_04-20.pdf.coredownload.inline.pdf

[15] Allen G. Taylor, *SQL for Dummies, 8th edition*. John Wiley & Sons, Inc. 2013. Accessed May 20, 2021. [Online]. Available: https://ebookcentral.proquest.com

[16] Schneider Electric. (Accessed April 9, 2021). Historian Server & Client Presentation [PPT]

[17] AVEVA. (Accessed April 10, 2021). Historian Concepts Guide [Online]. Available: https://cdn.logic-control.com/docs/aveva/historian/HistorianConcepts.pdf

[18] AVEVA. (Accessed April 10, 2021). Historian Administration Guide [Online]. Available: https://cdn.logic-control.com/docs/aveva/historian/HistorianAdmin.pdf

[19] AVEVA. (Accessed April 10, 2021). Historian Retrieval Guide [Online]. Available: https://cdn.logic-control.com/docs/aveva/historian/HistorianRetrieval.pdf

[20] Ocean Data Systems (Accessed March 22, 2021). Dream Report Getting Started Tutorial [Online]. Available: https://dreamreport.net/wp-content/uploads/Dream-Report-Getting-Started-2019.pdf

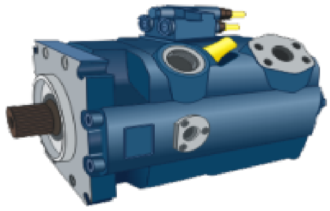
[21] Ocean Data Systems (Accessed March 22, 2021). Dream Report [Online]. Available: https://dreamreport.net

[22] Ocean Data Systems (Accessed March 25, 2021). Dream Report User's Guide [Online]. Available: https://downloads.softwaretoolbox.com/demodnld/295/Documentation/Dream%20Report%20user%20manual.pdf

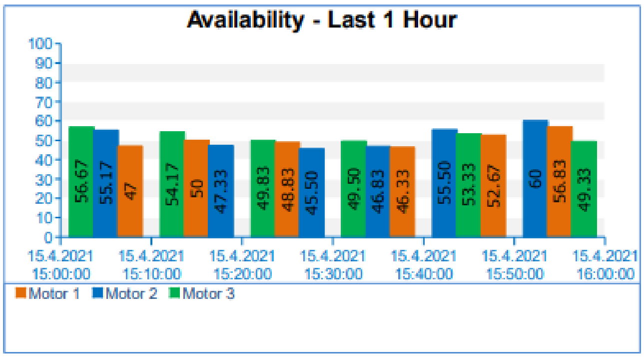
Appendix

Maintenance Report

Generated on: 15/04/2021 16:15:17



Motor Availability - Last 1 Hour					
Timestamp	ON Status	OFF Status	Running Time	Down Time	Availability %
15/04/2021 15:00:00	75	74	0:04:42	0:05:18	47 %
15/04/2021 15:10:00	77	77	0:05:00	0:05:00	50 %
15/04/2021 15:20:00	74	75	0:04:53	0:05:07	49 %
15/04/2021 15:30:00	69	69	0:04:38	0:05:22	46 %
15/04/2021 15:40:00	73	73	0:05:16	0:04:44	53 %
15/04/2021 15:50:00	77	76	0:05:41	0:04:19	57 %

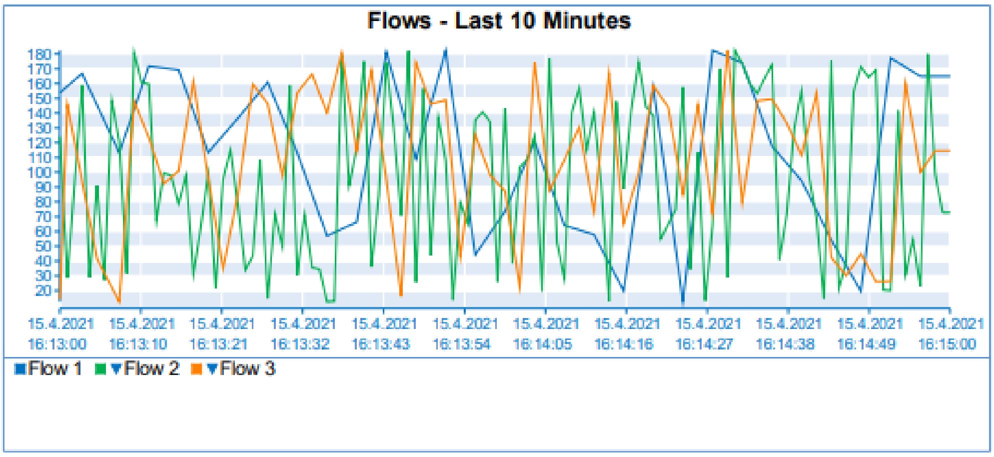


Motor Maintenance - Last 1 Hour				
Item	Current	Max	Min	Ave
Temp_Inlet1	3.06 °C	100.00 °C	0.22 °C	49.96 °C
Temp_Exhaust1	1.53 °C	49.87 °C	0.05 °C	24.52 °C
Temp_Compressor1	2.29 °C	74.89 °C	0.22 °C	37.23 °C
Pressure_Inlet1	1.83	59.99	0.02	30.68
Pressure_Discharge1	1.83	59.98	0.02	29.43
Pressure_Compressor1	1.53	49.92	0.06	25.51

Maintenance Report

Generated on: 15/04/2021 16:15:17

Pump Availability - Last 1 Hour					
Timestamp	ON Status	OFF Status	Running Time	Down Time	Availability %
15/04/2021 15:00:00	154	154	0:04:53	0:05:07	49 %
15/04/2021 15:10:00	142	141	0:04:53	0:05:07	49 %
15/04/2021 15:20:00	140	141	0:04:57	0:05:03	50 %
15/04/2021 15:30:00	144	143	0:05:05	0:04:55	51 %
15/04/2021 15:40:00	147	147	0:04:55	0:05:05	49 %
15/04/2021 15:50:00	160	161	0:05:01	0:04:59	50 %



Pump Maintenance - Last 1 Hour				
Item	Current	Min	Max	Ave
Flow1	251.53	2.53	199.30	98.29
Flow2	222.45	0.80	299.38	152.80
Flow3	203.06	2.26	396.31	200.21
Speed1	61.00	4.00	1985.00	1067.95
Speed2	53.00	4.00	1748.00	918.29
Speed3	45.00	0.00	1499.00	779.04
Head_Pressure1	3.00	0.00	99.00	49.26
Head_Pressure2	2.00	0.00	79.00	40.76
Head_Pressure3	2.00	0.00	69.00	35.19