STORIA DELLE VERSIONI

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rev** | **Date** | **Document** | **Issued by** | **Checked by** | **Approved by** |
| 1 | 14/06/2023 | File Exporter | Ugur Akcelik |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

DOCUMENTI RICEVUTI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Titolo | Descrizione | Rev | Data | Ricevuto Tramite |
|  |  |  |  |  |

DOCUMENTI ALLEGATI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Titolo | Descrizione | Rev | Data | Inviato Tramite |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

INDICES

[1. Introduction 3](#__RefHeading___Toc451794242)

[2. Scope 3](#__RefHeading___Toc451794242)

[3. Key Functionalities 4](#__RefHeading___Toc451794243)

[4. Components 5](#__RefHeading___Toc451794244)

[5. Software Architecture Use Case 8](#__RefHeading___Toc451794245)

# Introduction

The File Exporter System is a feature designed to provide users with an efficient way to export files from a widget to remote FTP (File Transfer Protocol) servers. This document serves as an architecture guide for the development and implementation of the FTP Exporter System, outlining the purpose, the key functionalities, interfaces, and the benefits it offers to users.

The primary goal of the FTP Exporter System is to enable users to transfer files securely and reliably to FTP servers, facilitating an integration with external systems and streamlining data exchange processes. By leveraging the power of FTP, this feature empowers users to export files to remote locations, facilitating data sharing and collaboration across different platforms and networks.

# Scope

The implementation aims to create a cohesive software solution that automates the process of exporting files from a source system and transferring them to a remote system using File Transfer Protocol (FTP). It involves integrating the back-end FTP node with the front-end widget, allowing users to configure FTP server credentials, time frequency settings and device selection. The software ensures secure and reliable file uploads by handling authentication, establishing secure connections, and providing error handling mechanisms. It serves as an efficient solution for organizations and individuals who need to automate file transfers, enabling them to streamline their workflows and improve productivity.

Users have the flexibility to select specific devices or choose to transfer files from all devices. This feature allows users to tailor their file exports based on their specific needs and requirements. By supporting device selection, the software empowers organizations and individuals to efficiently manage their file transfers, optimizing their workflows and enhancing productivity.

The scope of the file exporter with FTP feature encompasses the design, development, and implementation of a software component specifically focused on automating file transfers using the File Transfer Protocol (FTP).

The scope of the file exporter with FTP feature does not include the implementation of the underlying FTP server or the management of the remote system. Instead, it focuses solely on the design and development of the software component responsible for initiating and managing file transfers using the FTP protocol.

# Key Functionalities

The implementation of the file exporter with FTP feature, utilizing a generator to send user attributes to the FTP node on a specified frequency, includes the following key functionalities:

* File Export and Transfer: The file exporter enables users to select and export files from a source system and transfer them to a remote system using FTP. The exporter specifically uploads files based on device names, allowing users to organize and transfer files associated with different devices.
* FTP Node Integration: Incorporating the FTP node into the software architecture to handle file transfers. Establishing secure connections with FTP servers using the provided user attributes. Authenticating with the FTP server using the provided username and password.
* User Attribute Generation: Implementing a generator to dynamically commit user attributes for the FTP node. Generating attributes such as URL, port, username, password, optional device selection and schedule attributes. Providing the necessary user attributes for establishing the connection and scheduling file transfers.
* Data Format and Content: The exporter supports .csv files containing specific data fields, including date, device ID, device name, and telemetries. Software provides users with the capability to view logs and access detailed information regarding the telemetries.
* FTP Protocol Implementation: The exporter incorporates the necessary functionalities to establish secure and authenticated connections with FTP servers. It follows the FTP protocol standards for reliable file transfer, ensuring data integrity and confidentiality.
* Scheduled File Transfers: Enabling the software to schedule file transfers based on the specified frequency and time. Supporting different schedule methods (e.g., daily, weekly, monthly) as specified by the scheduleMethod attribute. Allowing users to define the specific hour and minute for the scheduled file transfers using the scheduleHour and scheduleMinute attributes.
* Error Handling and Retry Mechanisms: The software incorporates robust error handling mechanisms to detect and handle transfer failures or interruptions. It provides automatic retry options to ensure successful delivery of files, even in the presence of network disruptions or temporary issues.
* Logging and Monitoring: Incorporating logging mechanisms to capture relevant information about the file transfers. Logging the user attributes, transfer details, and any errors or exceptions that occur during the process. Providing monitoring capabilities to track the status and progress of the file transfers.

By implementing these key functionalities, the software enables the generator to dynamically fetch user attributes and sends them to the FTP node. This integration facilitates scheduled and secure file transfers based on the provided schedule attributes. The logging and monitoring features provide visibility into the transfer process and assist with troubleshooting and auditing.

# Components

## Backend FTP Node

The backend FTP node consists of several components that facilitate the functionality and methods required for efficient file transfer operations. These components include:

**Schedule Manager**

Purpose: Responsible for managing scheduled file uploads to the FTP server.

Methods:

* scheduleUploadFtp(): Allows scheduling of file uploads to the FTP server at specified intervals using the provided parameters.
* cancelScheduleUploadFtp(): Cancels a scheduled file upload, preventing it from executing at the specified interval.

**FTP Uploader**

Purpose: Handles the actual uploading of files to the FTP server.

Methods:

* uploadFtp(): Performs the upload of files to the FTP server, using the provided credentials and file data.
* testFtpConnection(): Tests the FTP server connection to ensure it is accessible and functional.

**Data Converter**

Purpose: Converts data into the CSV format required for file transfer.

Methods:

* convertDataToCSV(): Converts the provided data into the CSV format, ensuring compatibility for file transfer.

These components work together to enable the backend FTP node to effectively schedule and execute file uploads to the FTP server. The Schedule Manager allows for the configuration of scheduled uploads at specific intervals, while the FTP Uploader handles the actual transfer of files using the FTP protocol. The testFtpConnection() method ensures the FTP server connection is valid and accessible. The Data Converter component ensures that data is properly converted to the CSV format, facilitating smooth and accurate file transfers.

By leveraging these components, the backend FTP node provides the necessary functionality to automate and manage file transfers to the FTP server, ensuring secure, efficient, and reliable data exchange.

## FTP Widget

The FTP widget consists of various components that facilitate the configuration and management of FTP server credentials and scheduling settings. These components include:

**Input Fields**

Purpose: These fields capture user input for configuring FTP server details, scheduling settings, and device selection.

Components:

URL: Represents the FTP server URL.

Port: Represents the port number for the FTP server.

Username: Captures the username for authentication with the FTP server.

Password: Captures the password for authentication with the FTP server.

Tries: Indicates the number of attempts to make for file uploads.

Schedule Method: Specifies the schedule method for file uploads (daily, weekly, monthly).

Device: Allows the user to select a specific device or leave it blank to indicate that all devices should be included in the file transfer.

The Attribute Service component handles the saving of these attributes associated with the specific entity (e.g., CUSTOMER), including the device selection. Success and failure handling callbacks are implemented to provide appropriate feedback based on the attribute saving status.

By utilizing these components, the FTP widget allows users to configure and save FTP server credentials, scheduling settings, and specify the device(s) for file transfers.

**Attribute Service**

Purpose: Handles the saving of FTP server attributes associated with the entity (e.g., CUSTOMER).

Methods:

* saveEntityAttributes(): Saves the FTP server attributes for the given entityId and server scope.

**Success and Failure Handling**

Purpose: Provides success and failure callbacks for handling attribute saving status.

Methods:

* success(): Executes when the saving of FTP server attributes is successful.
* fail(): Executes when the saving of FTP server attributes fails.

The Input Fields component collects the necessary information such as URL, port, username, password, tries, schedule method, schedule hour, and schedule minute. The AttributeService component handles the saving of these attributes associated with the specific entity (e.g., CUSTOMER). Success and failure handling callbacks are implemented to provide appropriate feedback based on the attribute saving status.

By utilizing these components, the FTP widget allows users to configure and save FTP server credentials and scheduling settings for efficient and automated file transfers.

# Software Architecture Use Case

# A picture containing text, diagram, line, technical drawing Description automatically generated

* User interacts with the File Transfer Widget, providing the data.
* The File Transfer Widget sends the user-provided data to the Customer Attributes Node.
* The Customer Attributes Node receives the data and saves it as customer attributes associated with the user or entity (e.g., CUSTOMER).
* The Generator component triggers the retrieval of customer attributes, including FTP server details and scheduling settings.
* The Customer Attributes Node receives the request from the Generator component and retrieves the relevant customer attributes.
* The Customer Attributes Node sends the retrieved attributes, such as URL, port, username, password, tries, schedule method etc. to the FTP Node.
* The FTP Node receives the customer attributes from the Customer Attributes Node, which include the FTP server details and scheduling settings.
* The FTP Node processes the received attributes and uses them for establishing a connection with the FTP server, scheduling file transfers, and executing the file transfer operations.