

Managed File Transfer Protocol

Naveen Muppa*

10494 Red Stone Dr Collierville, Tennessee, USA

Citation: Naveen Muppa. SAVIYNT IGA Solution for Simplifying and Streamlining the IT Operations. *J Artif Intell Mach Learn & Data Sci* 2023, 1(1), 242-241. DOI: doi.org/10.51219/JAIMLD/Naveen-muppa/77

Received: 02 November, 2023; **Accepted:** 18 November, 2023; **Published:** 20 November, 2023

***Corresponding author:** Naveen Muppa, 10494 Red Stone Dr Collierville, Tennessee, USA

Copyright: © 2023 Muppa N. Enhancing Supplier Relationships: Critical Factors in Procurement Supplier Selection.. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ABSTRACT

This paper is to let new users to get familiar with Power BI and be able to create jobs in GoAnywhere.

Keywords: GoAnywhere is centralize and secure file transfers tool with ease, streamline processes. It provides centralized control and improves the quality of your file transfers and help the organization to comply with data security policies and regulations. It is a managed file transfer and encryption tool used by Mid-sized companies, large enterprises, and government entities to protect sensitive data.

1. Introduction

This tool is used for any function to be performed with files. A Business Process which involves Ftp'ing of files to an outside server or uploading of a document onto another server, Encrypt or Decrypt a file, create a trigger file, we make use of GoAnywhere.

GoAnywhere (GA) practically has interrelationship with across applications being an important tool to deal with file operations. These jobs either be scheduled to run at a particular time or could be called on to run during a Informatica Feed execution.

Since the Job deals with files movement across applications in the system, it holds the critical place in the business.

2. Functional Overview

Resources are the names and connection properties of the servers, and other data sources, which GoAnywhere can interact with. Users with the Resource Manager role can create and edit Resources. The defined Resources can be used within GoAnywhere Projects by choosing the Resource names from drop-down lists.

Listed below are the Resource types that have been defined in GoAnywhere.

Azure Blob Storage: Azure Blob Storage provides organizations with affordable and scalable cloud storage provided by Microsoft. GoAnywhere supports Block type blob storage.

Network Shares: The names and connection properties of shared file locations on a network to which GoAnywhere can connect for accessing files.

Database Servers: The names and connection properties of the database servers to which GoAnywhere can connect. Database servers supported include DB2 for IBM i (iSeries), DB2, Oracle, SQL Server 2000 and later, Sybase, Informix, PostgreSQL, and MySQL.

FTP Servers: The names and connection properties of the FTP servers to which GoAnywhere can connect for sending and receiving files.

FTPS Servers: The names and connection properties of the FTPS servers (FTP over SSL) to which GoAnywhere can connect for sending and receiving files.

SSH Servers (SFTP/SCP/SSH): The names and connection

properties of the SFTP servers (FTP over SSH) to which GoAnywhere can connect for sending and receiving files.

SMTP Servers: The names and connection properties of the SMTP servers (Mail Servers) to which GoAnywhere can connect for sending emails.

OpenPGP Key Rings: The names of the OpenPGP Key Rings that can be used in GoAnywhere to encrypt and decrypt files using the OpenPGP standard.

Permissions (authorities) for a Resource can be granted to individual users and Groups of Users. The following three types of permissions can be granted:

Permission Description: Read - Allows any user with the Resource Manager role to view the settings for the Resource.

Write: Allows any user with the Resource Manager role to change the Resource settings or permissions, or to delete the Resource.

Use: Allows users to utilize the Resource when executing a Project.

For instance, you may have a FTP server resource that only certain users should be able to utilize (connect to). If you additionally do not want these users to change the settings on the FTP server, then you would give the users the permission of Use only to that Resource.

Module: A Module is a logical grouping of one or more Tasks. For instance, a module may be defined with three Tasks to be executed in sequential order. The first Task in the Module may be to create a Work Space, second task calculate end time and third task to connect to source path and create a file list. Multiple modules can be defined in a Project. A Module can pass control to another Module based on certain conditions.

For instance, you could have a main Module to perform a series of tasks. If any errors are encountered in the main Module, you could have control passed to another module that sends an error notification to an email address or perform some other Tasks. The Call Module task can execute another Module in a Project to perform a sub-routine of tasks based on specific criteria.

IF Condition: An IF Condition controls if a block of tasks will run if a condition is met. The IF Condition is like a Yes/No decision point in a process flow chart.

Loops: Loops are complex components that repeat a set of Tasks on the data or files produced in a Project.

For instance, the data in a spreadsheet needs to be read into a database. A Loop contains the Tasks that will read a row of data, evaluate and process it and then iterate to the next row of data and repeat the Loop.

Task: A Task is a discrete business process to perform. For instance, a Task may write data to an XML document, send an email, import data from an Excel file, or FTP a file. GoAnywhere includes dozens of different Tasks to choose from when building a Project. There is no logical limit to the number of Tasks that can be defined in a Project.

Element: A Task is made up of one or more Elements, which describes the types of work to perform within the Task. For example, the FTP task has an Element to put a file on a FTP server, an Element to get a file from a FTP server, an Element to

rename a file on a FTP server, and so on. As another example, the Send E-Mail task has an Element to indicate the attachments to send.

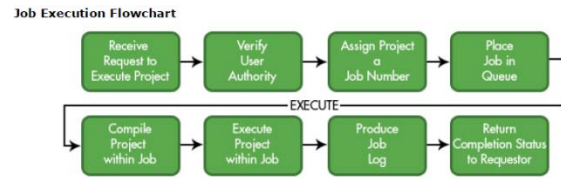


Figure 1: Flow.

3. Technical Overview

We have within the GA GUI, which are Resource, Projects, scheduler, Active Jobs, Completed Jobs, Users, and groups. The Development team has access to Projects, scheduler, Active Jobs, Completed Jobs.

Resources are the section wherein the resource utilized within the project are being set up. Projects are the section wherein you will be able to find all the existing projects we have. This is the section where we develop any new projects required to support the business. Active and Completed section are the areas where we can view the jobs that are in execution or the once that have completed their execution.

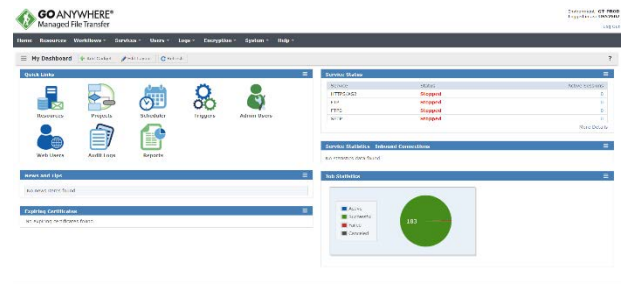


Figure 2: Dashboard.

Pub/Sub Pattern:

Publication Process: The publication process includes retrieving the data from the publisher, running the publication mapping, and writing the data to the relevant topic in the publication repository. After the publication process ends, each subscriber consumes the published data according to the schedule and the filter that you define when you create the subscription. In addition, in our process after every publication a Mapping task subscription will be triggered to load the Topic table's data into Datastore.

**As part of this solution to increase reusability and to mitigate any future risks if we add/remove any attributes – Mappings are considered instead of DSS tasks.

The following flow shows the main stages of the publication process for publications:

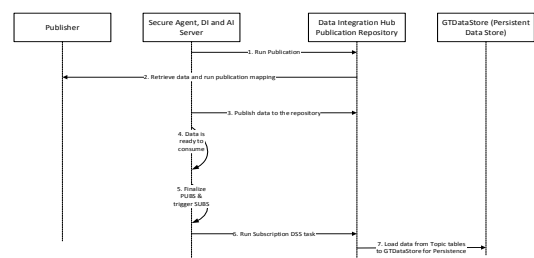


Figure 3: publication process.

Subscription Process: The subscription process includes retrieving the required data from the Data Integration Hub

publication repository, running the subscription mapping, and writing the data to one or more subscriber targets. Data Integration Hub keeps the data in the publication repository until the retention period of the topic expires.

As soon as all the Topics and topic tables are published, all the data from these topics and topic tables will be subscribed into the Data Store, a persistent database on Azure.

For all the applications that are being configured to run the APIs for Subscription, the tasks will retrieve data from Datastore but not from Data Hub.

The following image shows the main stages of the subscription process for each subscription:

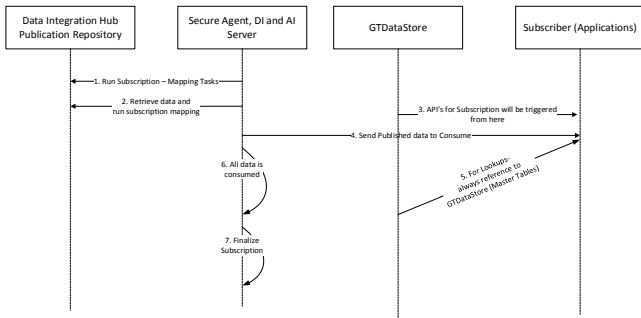


Figure 4: Processes for Subscription.

Integration Tasks without Topics/Topic Tables (ETL Pattern): Informatica Data Integration or Application integration tasks (mappings, task flows, etc.) will be developed and configured to run on the source data and the target data is staged in the *Data Store Stage* database on Azure for further processing. This route will be taken for application data which are not used downstream, and which are only for application specific.

The following flow shows the main stages of how the DT Data Store Stage is integrated within the process:

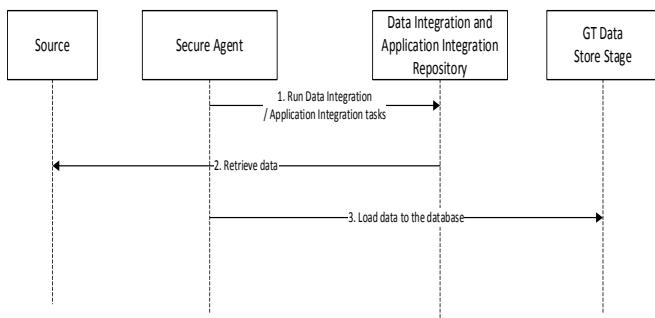


Figure 5: Processes for topic.

Integration Tasks for Real Time applications: For all the real time feeds, Informatica Data Integration or Application integration tasks (mappings, task flows, etc.) are considered, but Power Exchange CDC comes into consideration as all the real time feeds are dependent on the condensed files for capturing few attributes, which will be consumed by other applications.

4. Handling CDC & Full Load Process

- All CDC logic will be built from Source to CDC Topics in their Publication.
- Bound Subscription will subscribe that data through a Mapping task from Topics to Datastore
- Bound Subscription will subscribe the data through mapping task from Topics to the Applications.
- A Publisher for Full load will be loaded from Datastore to Full Load Topics

- Unbound Subscription will subscribe that data from Topics.

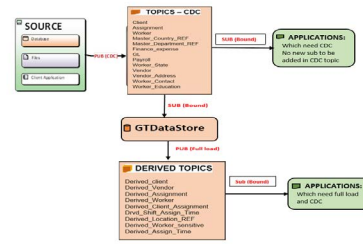


Figure 6: Processes for full load topics.

5. Initialize and Generate Automatic Parameter File

- This task takes the process name and application name parameter and executes the batch job to call the stored procedure, which in turns does the following activities.
- While inserting a record in Process table, it checks for the below conditions and picks up the respective values to insert:
 - Checks for IsOverrideETLProcessID (default is 0), for each task from TaskParameter table.
 - If IsOverrideETLProcessID=0, then Identity column in Process Table: ProcessID= ETLProcessID in Process table.
 - If IsOverrideETLProcessID>0, it picks up the mentioned ETLProcessID from the TaskParameter table.
 - Checks for the override parameter value (IsOverride) for each task from TaskParameter table (default is always 0, 1 means override)
 - If IsOverride=0, then the latest TaskStartDate for that Task from Task Table where Status='Success' is updated into TaskParameter Table as SourceExtractStartDate. And Then, SourceExtractEndDate=getdate ().
 - If IsOverride=1, it picks up the mentioned SourceExtractStartDate and SourceExtractEndDate from the TaskParameter table.
- Generates the dynamic parameter file by merging static file and the output of TaskParameter table
- For PUB/SUB MCT's, the variables are set at the mapping level itself. (SourceExtractStartDate, SourceExtractEndDate, ModifiedDate, ETLProcessID)

6. Delta Detection

When the data in a source system is frequently updated, it is necessary to capture the updated information to the target extracts. However, due to high volume and load window, it is desirable to consider only the updated delta information, rather than reloading the entire source table. Usually, there is a Modified timestamp column in the source table. This column can be used to filter the source records, based on the last source extract end time of that task.

This is implemented using a common framework component to generate a dynamic parameter file for each task flow, based on the previous successful execution status in the Task table, Stored procedure picks the SourceExtractEndDate for that task and updates the TaskParameter table SourceExtractStartDate = Task. SourceextractendDate and SourceExtractEndDate = GETDATE.

7. Folder Structure

The below Project structure facilitates the coordination

and implementation of data hub using different services in IICS. Its main reason is to create an environment that fosters interactions among the team members with a minimum number of disruptions, overlaps and conflict.

The structure of the directories created in the repositories always follows the below hierarchy.

- Application Name
- API (Process, Process Objects, etc.)
- ETL (Mappings, Mapping tasks, DSS tasks, etc.)
- Orchestration
- Publisher (All mappings that are in the Publications for the application)
- Subscriber (All mappings that are in the Subscriptions for the application)

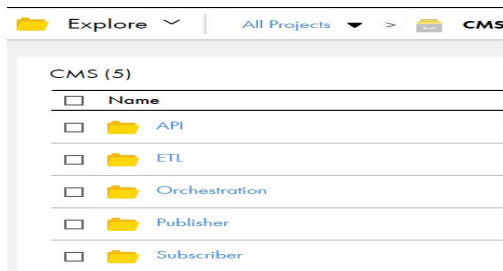


Figure 7: Folder structure

Folder structure in integration hub: In Integration hub, assets like Application, Topic, Publisher, and subscribers are managed.

Folders in CIH are predefined based on Applications, Topics, Publications and Subscriptions. There is no provision to update folder structure manually.

8. Version Control

Version control is not available in Informatica Cloud. All developers must save their changes religiously and always make sure to take backups regularly.

Run team is working on a solution for Version Control which is out of scope for this project.

The below is the proposed solution for code management. Even though it is not in scope of this project, below is the future roadmap for handling code version management in IICS as the tool doesn't provide any version control.

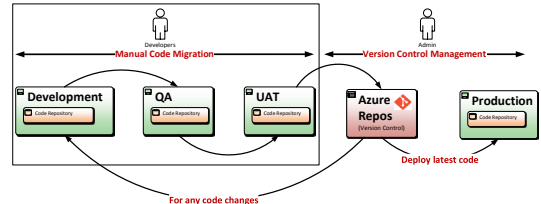


Figure 8: Deployment flow.

9. References

1. <https://www.informatica.com/blogs/welcome-to-informatica-intelligent-cloud-services.html>
2. <https://now.informatica.com/IICS-Cloud-Data-Integration-Services-onDemand.html>