Contents

[VinSolutions\_FTP 1](#_Toc76987484)

[SSIS Package 1](#_Toc76987485)

[Overview 1](#_Toc76987486)

[Package Sections 1](#_Toc76987487)

[File Processing Errors 1](#_Toc76987488)

[Checking the Process 2](#_Toc76987489)

[Overview 2](#_Toc76987490)

[Troubleshooting Failed Runs 3](#_Toc76987491)

[Troubleshooting Errors 3](#_Toc76987492)

[Notes 4](#_Toc76987493)

# VinSolutions\_FTP

## SSIS Project

### Overview

The SSIS package is scheduled to run every hour at the 45-minute mark. It is designed to process any files in the target folder that has its hidden attribute set to false. This way if a file is missed, gets placed into the folder after a pass, or throws an error during a pass (see errors below), the package will attempt to process it again.

The package was created in Visual Studio 2015 with the SSIS tools installed separately. When you open the package in VS you will be prompted to enter a password. The password is “users”. It was necessary to give the project a password as SSIS is security oriented. It wouldn’t allow me to create the project on one machine and install it on the server without a password.

### Package Sections

#### Main (Main.dtsx)

* Reads in DealerIDs.Txt for dealer Id folders to loop through
* Reads in DatabaseTables.txt for database table folders to loop through
* Calculates current date folder name (Cloudstore starts their new folder after 8pm the day before the folder name) So after 8pm on 7/7 a 20210708 folder was created and the rest of the files – as well as 7/8 files – are place in it.
* Traverses the folder structure (DealerID/TableName/CurrentDate)
* Zip files that are in the folder and are not processed (have the Hidden attribute set to true) are unzipped and the csv file name as well as the db table name are passed to the CSVFileProcessing package

#### CSV File Processing (CSVFileProcessing.dtsx)

* This package basically sends the csv file name to the appropriate file processing package based on the database table. It’s effectively one big if…then…elseif… structure

#### File Processing Packages (Appointment.dtsx to UserAccess.dtsx)

* Each package handles the files for a different database table
* The file connection is specific to the file structure and maps it to the database table column definitions (this is why there are so many individual packages).
* The sequence is:
  + Receive the file name to process
  + Import the csv file
    - Open the csv file and read it in as a flat file
    - Send each record in the file to the table’s stored procedure
      * Stored procedure updates the existing record, or inserts a new record
* If the import fails, it deletes the csv file (leaving it to be potentially processed later). Or, if successful, it changes the related zip file’s hidden attribute to true so it won’t be processed later.

### Configuration Variables

There are four variables that need to be configured each time the package is installed/updated

1. DBTablesFile: the location of the DBTables.txt file.
2. DealerIDsFile: location of the DealerIDs.txt file
3. FTPFolder: the base folder where the files to be processed reside.
4. UnZipExecutable: the location of the 7zip executable (only necessary when the default installation path isn’t being used. Currently, it is installed at the default path.)

These can be found in SSMS on the server by drilling down Integration Services Catalog/SSISDB/VinSolutionFTP/Project/VinSolutionFTP. Right-click on the project and select “Configure…”

Database connection configuration is under the Connection Managers tab

### Reinstalling the Package

Any time the package is changed it has to be reinstalled on the server. I usually delete the old project before installing the new one (I don’t really have a reason to do it. I just started that way and haven’t done it any other way). To do this:

1. Browse to the project’s Bin directory
2. Under that, in the appropriate folder you should find a file named VinSolutionFTP.ispac
3. Copy that file and place it somewhere on the server
4. Right-click the file and select OpenWith -> SQLServer Integration Services Deployment Wizard
5. Follow the prompts
   1. Under Select Source make sure the project deployment file is selected. Click Next
   2. Enter the password
   3. Enter/Select the server name (JJFDATASERVER)
   4. For the path, click the Browse button and click on the VinSolutionFTP folder
   5. Click OK. Click the Next button
   6. Review the selections, click Deploy

### File Processing Errors

#### SSIS Errors

1. If a column experiences a truncation error
2. All other errors

There ways of handling these errors:

1. Halt all processing
2. Ignore the error (in this case the column that throws the error will have its value set to null)
3. Skip the row that threw the error and continue processing the rest of the file.

* Currently, the VinSolutions package is set up to throw an error when a truncation error occurs and skip the file (in which case the package will attempt to process it again in the next pass).
* For other errors it skips the record and continues to process the rest of the file. The errors that I saw causing this error were usually bad ID numbers (IDs with alpha and other characters in them)

#### Handling Truncation Errors

Truncation errors occur when the data in the csv file is too long for the database column. In this case, I’ve mostly increased the size of the db column to accommodate the data.

Increasing a DB table column size requires the following changes:

1. The database table needs to be altered (obviously)
2. The Stored Procedure that inserts/updated the table parameter for the column
3. The file connection in the database table named package (for example, Appointment.dtsx)
4. The new column size needs to be propagated to the flat file processing (step named Read CSV File) column mappings
5. The column size needs to be propagated to the column mappings in the Execute Appropriate SP step of the package.

#### Zip file Errors

Lately, in the past 6 weeks or so, we’ve periodically had issues with zip files having a 0kb size in the folders. I’m not sure what is causing this. Whether it’s a download issue or an the file just doesn’t have any data. Either the process will record it as an error with unzipping a file and continue processing the rest of the files.

The failure will continue to be flagged until it is taken care of. In the past, I’ve just set the offending files hidden attribute to true, but now I’m thinking it might be better to just delete it and have Cloudstore try to download it again. If it fails to download it properly the second time, then I’d set the hidden attribute to true.

## Checking the Process

### Overview

I check the process at least once a day when I get in. I do this by looking at the reporting for the VinSolutionFTP project.

To do this:

1. Open SSMS
2. Browse to the project
   1. Expand the Integration Services Catalog folder
   2. Expand SSISDB folder
   3. Expand VinSolutionFTP
   4. Expand projects
   5. Right-click on VinsolutionFTP. Select Reports/Standard Reports/All Executions
3. The report will display the results of the last ~150 runs
4. Look for any failed runs (there may be several in a run – this is intentional as the process always attempts to process a zip file even if it has failed before)

### Troubleshooting Failed Runs

1. Click on the “Overview” link for one of the failed runs (again there will probably be several in a row)
2. Above the list of messages there is a link labeled “Filter”. Click that.
3. Click on the Result dropdown and select “Failed”
4. Click on the links in the Execution Path column to view the details of the errors
5. I’ve seen two types of errors
   1. Truncation
   2. Unzip

### Troubleshooting Errors

Each link in the Execution Path column from the previous section usually displays a few different messages. The key is to identify the type of error and the file that caused the error

#### Truncation Errors

Truncation error messages will give you the file that threw the error, the column in the file and the row.

Search for the file in the folder structure. I tend to copy the file and paste it onto my machine to work with (I just don’t like to mess with the data on production until I know what I want to change).

I find it best to open a new Excel spreadsheet, click on the Data tab, then import the unzipped csv file into Excel. (Just opening the file in Excel usually everything in the same column.)

Find offending row and column. Identify the problem (usually that the text is too long to fit the column) Most of the time I’ve changed the db column definition to accommodate the extra length as I’ve found that if there’s one record that creates this error, there will likely be more.

Follow the directions under [Handling Truncation Errors](#_Handling_Truncation_Errors) to lengthen the column definition and alter the package.

#### Unzip Errors

As I said above, these errors usually show up as a failure of the 7zip program. I’ve only seen this happen when the zip file has a 0-byte size. In the past, I’ve just set the hidden attribute to true on the zip file to cause the process to ignore the file in the future.

It may make sense to delete the file instead. The download process should attempt to download the file again which would cause the VinSolution package to attempt to process it again.

#### Notes About Errors

1. As I said before, if a file throws an error, it will throw it again until one of two conditions are met:
   1. The issue is addressed
   2. The end of the downloading day (~8pm). The first download after 8pm each day goes into a folder with the next day’s date and the package process changes it’s focus to the new folder.
2. It is not unusual to see several failed processed in a row. It’s usually one error propagated over the following sessions. But make sure that’s the case and there aren’t other issues as well.
3. If you fix an error in a file that was downloaded to the previous day’s folder. It WILL NOT get processed unless you copy it to the current day’s folder.

## Upgrading to v5.0.0

I’ve updated the package to be able to process the 5.0.0 changes. Here are the steps necessary to upgrade the VinSolution\_FTP tables/stored procedures:

### Table Definition Changes

These columns have to be added to the VinSolutions\_FTP tables

* Appointment
  + AppointmentReasonName varchar(50)
  + AppointmentLocationTypeName varchar(50)
  + AppointmentSubject varchar(500)
  + AppointmentMemo varchar(500
* CRMSoldTransaction
* CRMSaleID  int
* Customer
* PreferredContactMethodName varchar(50)
* Lead
* CoBuyerCustomerID int
* LeadTradeInVehicle
* Memo 500
* ShowroomVisit
  + VisitLocationTypeName varchar(100)

### Stored Procedure Changes

The following stored procedures need to be changed to accommodate the table changes:

* usp\_upsert\_Appointment
* usp\_upsert\_CRMSoldTransaction
* usp\_upsert\_Customer
* usp\_upsert\_Lead
* usp\_upsert\_LeadTradeInVehicle
* usp\_upsert\_ShowroomVisit

The easiest way to update these is to copy the code from the versions in the jjstest database (these stored procedures have already been upgraded)

### VinSolutionFTP Package

Deploy the VinSolutionFTP.ispac file that can be found in the \bin folder of the VinSolutionFTP\_5 project. Follow the instructions [here](#_Reinstalling_the_Package) to redeploy the package.

### Installation Caveats

I’m pretty sure that you can’t make the database changes until you’re ready to use the 5.0 package as the stored procedures will fail against the older version tables.

Right now Cloudstore is putting the v5 files under the updates\5.0.0 folder. After August 15th, they will be putting the files directly into the update folder (and stop putting them in the 5.0.0 folder).

## Notes

1. It’s important to remember that only files In the most recently created folder will be processed. To process a file in an older folder, it has to be copied into the current (most recent) folder. Be careful when doing this: if you move a file with older data in it, it will overwrite the newer data)
2. If you’re testing a change to the project and receive errors stating that there’s not enough memory.
   1. That’s usually a Visual Studio issue and not an issue with the package. Visual Studio uses a lot of memory to display the processing of the data and that’s usually the cause.
   2. Make sure that there are no other issues
      1. If there are, examine them and fix them before running again
      2. If there aren’t, run the package again. It will pick up where it left off and finish the processing (unless it runs out of memory again).
   3. You can also limit which dealers and database tables get processed by adding/removing them from their respective files.
   4. I haven’t seen this kind of error when running the project on the server
3. It is possible to configure the package to run any combination of dealers and database tables by editing the DealerIDs.txt and DatabaseTables.txt files before running the package. So If you wanted to skip processing a particular table you could just remove it from the DatabaseTables.txt file. This should only be done under rare circumstances as you might forget to replace the tables/dealerids you removed when you’re done.

## Updates

### July 19, 2021

I’ve added a new project level parameter (ManualCurrentDate). Project level parameters can be configured outside of the packages by clicking on the project name (in this case VinSolutionFTP) and selecting “Configure…”.

* If the parameter is left blank the project will calculate the current date (which is not today’s date if it is after 8pm, [See calculating current date folder](#CalculateCurrentDateFolder)).
* If the parameter has a value it will use that value as the current date folder. Current dates are formatted as yyyymmdd.

I added this feature as a result of the issues we’ve been having with the download. In previous versions, only the calculated date was used and we had no way to (relatively) easily previous day’s data. Now we can do this although processing a full day’s update data takes a long time as there are 11(dealerships) \* 23 (database tables) \* 22 (hourly files – not all hours have downloads) = 5,566 files to process for a given day.